



**UNIVERSIDAD INTERNACIONAL DEL  
ECUADOR**

**FACULTAD DE CIENCIAS TÉCNICAS**

**ESCUELA DE INGENIERÍA MECATRÓNICA**

**DISEÑO Y SIMULACIÓN DE UN SISTEMA DE RECONOCIMIENTO  
FACIAL BASADO EN DEEP LEARNING**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
MECATRÓNICA**

**SEBASTIÁN ALEJANDRO PÉREZ JARRÍN**

**DIRECTOR: ING. VICTORIA MERA, M.Sc.**

**D. M. Quito,**

**2021**

## DECLARACIÓN

Yo PÉREZ JARRÍN SEBASTIÁN ALEJANDRO, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que se ha investigado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Internacional del Ecuador, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por normativa institucional vigente.



---

**PÉREZ JARRÍN SEBASTIÁN ALEJANDRO**

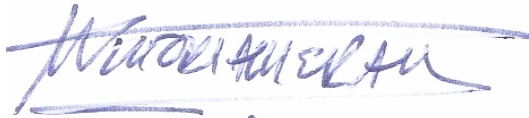
**C.C. 172174110-4**

## CERTIFICACIÓN

El Docente de la Facultad de Ciencias Técnicas Ingeniera Victoria Mera.

CERTIFICA QUE:

El proyecto de investigación “DISEÑO Y SIMULACIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING”, fue desarrollado por el Sr. PÉREZ JARRÍN SEBASTIÁN ALEJANDRO ha sido debidamente revisado y está en condiciones de ser entregado para que siga lo dispuesto por la Facultad de Ciencias Técnicas, correspondiente a la sustentación y defensa del mismo.



---

**ING. VICTORIA MERA, M.Sc.**

**C.C. 180400459-4**

**DIRECTORA DE PROYECTO**

*Dedico este trabajo a la memoria de mi Abuelito, quien siempre me guió hacia la excelencia mediante el esfuerzo, con el conocimiento de que este momento hubiera sido tan especial para él como lo es para mí. A mis padres, por ser el pilar de mi vida y demostrarme su cariño y apoyo incondicional. A mis hermanos, por la sabiduría que me han transmitido a lo largo de mi desarrollo con sus consejos y enseñanzas. A mi Mamichía, a quien quiero como una madre, por siempre estar dispuesta a escucharme, quererme y ayudarme en todo momento. A mi Mini, por darme todo el amor y motivación del mundo, sin ti no lo hubiera logrado. Y finalmente gracias a todas las personas que ayudaron directa e indirectamente en la realización de este proyecto*

# ÍNDICE DE CONTENIDOS

1.	Tema.....	1
2.	Objetivos.....	1
2.1.	General.....	1
2.2.	Específicos.....	1
3.	Problema .....	2
4.	Hipótesis.....	2
5.	Estudio Teórico.....	2
5.1.	Introducción .....	2
5.2.	Reconocimiento Facial .....	4
5.3.	inteligencia Artificial .....	5
5.4.	Machine Learning.....	6
5.5.	Deep Learning.....	7
5.6.	Redes Neuronales Artificiales.....	8
5.7.	Redes Neuronales Convolucionales .....	10
5.8.	Trabajos Relacionados.....	13
6.	Análisis y Selección de Alternativas .....	14
6.1.	Especificaciones Técnicas .....	14
6.2.	Análisis Funcional .....	14
6.3.	Alternativas .....	17
7.	Diseño Mecánico.....	28
7.1.	Consideraciones de diseño.....	29
7.2.	Dimensionamiento Base Externa .....	30
7.3.	Dimensionamiento Motor.....	39
7.4.	Dimensionamiento Eje.....	42
7.5.	Torque de Sujeción Base Exterior.....	46
8.	Algoritmo general de programación.....	48
8.1.	Base de Datos .....	50
8.2.	Red Neuronal Convolucional .....	53

8.3.	Reconocimiento Facial .....	54
9.	Diseño Electrónico.....	57
9.1.	Características del Ordenador .....	57
9.2.	Actuadores.....	58
9.3.	Controlador .....	60
10.	Pruebas y Resultados .....	65
10.1.	Prueba 1: Determinación del número de imágenes necesarias para la base de datos de Google .....	65
10.2.	Prueba 2: Determinación del número de imágenes necesarias para la base de datos obtenidas localmente (cámara de profundidad) .....	67
10.3.	Prueba 3: Determinación de funcionalidad clasificador " Cascade" e Interferencia de accesorios del usuario .....	68
10.4.	Prueba 4: Determinación de funcionalidad por interferencia de semejanza entre familiares.....	75
10.5.	Prueba 5: Determinación de luminosidad de imágenes de la Base de datos.....	76
10.6.	Prueba 6: Determinación de luminosidad para detección facial.....	78
10.7.	Prueba 7: Determinación de ángulo mínimo para detección de rostro80	
10.8.	Prueba 8: Determinación de factibilidad de diferentes ángulos dentro de la base de datos.....	82
10.9.	Prueba 9: Determinación de distancia máxima.....	84
10.10.	Prueba 10: Determinación de calidad de imágenes .....	85
10.11.	Prueba 11: Determinación de calidad de imágenes de la base de datos87	
10.12.	Prueba 12: Determinación de calidad de vídeo .....	88
10.13.	Prueba 13: Comparación de cámara de profundidad con cámara normal90	
10.14.	Prueba 14: Comparación del sistema con cámaras de seguridad en el mercado actual .....	92
11.	Conclusiones .....	97
12.	Recomendaciones.....	101

## ÍNDICE DE FIGURAS

1. Sistema biométrico facial [1].....	5
2. Relación entre neurona artificial y neurona biológica [2].....	9
3. Estructura de una red neuronal [3].....	9
4. Estructura de una red neuronal convolucional [4] .....	11
5. Convolución [5] .....	11
6. Pooling [6].....	12
7. Análisis funcional Nivel 0 .....	15
8. Análisis funcional Nivel 1 .....	15
9. Análisis funcional Nivel 2 .....	16
10. Vista General Prototipo .....	29
11. Diagrama de cuerpo libre viga en voladizo cargas puntuales .....	30
12. Diagrama de cuerpo libre fuerza de apoyo .....	31
13. Diagrama de cortante.....	32
14. Diagrama de momentos.....	32
15. Simulación Cargas Puntuales Viga en voladizo .....	34
16. Diagrama de cuerpo libre viga en voladizo carga distribuida .....	35
17. Diagrama de cortante.....	36
18. Diagrama de momentos.....	36
19. Simulación Cargas Distribuidas Viga en voladizo .....	37
20. Simulación Base Externa.....	37
21. Diagrama de cuerpo libre Eje .....	42
22. Simulación Flexión y Torsión Eje de 10mm.....	46
23. Simulación Torsión aplicada en perforaciones de sujeción .....	47
24. Algoritmo general.....	49
25. Imagen almacenada con clasificador " Cascade" en la base de datos .....	52
26. Imagen almacenada en la base de datos.....	53
27. Imagen resultado almacenado .....	55
28. Video resultado almacenado .....	56

29. Video resultado almacenado .....	57
30. Diagrama de Bloques del prototipo.....	57
31. Representación Gráfica Prueba 1 .....	66
32. Imagen resultado almacenado Prueba 1 .....	66
33. Representación Gráfica Prueba 2.....	67
34. Imagen resultado almacenado Prueba 2 .....	68
35. Representación Gráfica Prueba 3 con " Cascade" y con accesorio .....	69
36. Representación Gráfica Prueba 3 con " Cascade" y sin accesorio .....	69
37. Imagen dentro de la base de datos Prueba 3 con " Cascade" y con accesorio	70
38. Imagen dentro de la base de datos Prueba 3 con " Cascade" y sin accesorio	70
39. Imagen resultado almacenada Prueba 3 con " Cascade" y con accesorio .....	70
40. Imagen resultado almacenado Prueba 3 con " Cascade" y sin accesorio .....	71
41. Representación Gráfica Prueba 3 sin " Cascade" y con accesorio .....	72
42. Representación Gráfica Prueba 3 sin " Cascade" y sin accesorio .....	72
43. Imagen dentro de la base de datos Prueba 3 sin " Cascade" y con accesorio	73
44. Imagen dentro de la base de datos Prueba 3 sin " Cascade" y sin accesorio	73
45. Imagen resultado almacenado Prueba 3 sin " Cascade" y con accesorio .....	74
46. Imagen resultado almacenado Prueba 3 sin " Cascade" y sin accesorio .....	74
47. Imagen resultado almacenado Prueba 4 .....	75
48. Representación Gráfica Prueba 4.....	76
49. Rango de exposición de las imágenes base de datos Prueba 5.....	77
50. Representación Gráfica Prueba 5.....	78
51. Imagen resultado almacenado Prueba 5 .....	78
52. Rango de exposición de una imagen Prueba 6.....	79
53. Representación Gráfica Prueba 6.....	80
54. Fotografías en el rango de 0 a 90 grados Prueba 7.....	81
55. Representación Gráfica Prueba 7.....	81
56. Fotografías en el rango de 0 a 90 grados Prueba 8.....	82
57. Imagen resultado almacenado Prueba 8 .....	83
58. Representación Gráfica Prueba 8.....	84



59. Fotografías a diferentes distancias en metros Prueba 9 .....	84
60. Representación Gráfica Prueba 9.....	85
61. Imagen a diferentes resoluciones Prueba 10 .....	86
62. Representación Gráfica Prueba 10.....	87
63. Imágenes de la base de datos a diferentes resoluciones Prueba 11 .....	87
64. Representación Gráfica Prueba 11.....	88
65. Fragmento 1 Vídeo serie Glee a 720p Prueba 12.....	89
66. Fragmento 2 Vídeo serie Glee a 720p Prueba 12.....	89
67. Fragmento 3 Vídeo serie Glee a 720p Prueba 12.....	89
68. Imágenes de las bases de datos de cada cámara.....	90
69. Representación Gráfica distancias Prueba 13 .....	91
70. Representación Gráfica ángulos Prueba 13.....	92

## ÍNDICE DE TABLAS

1. Algoritmos de Machine Learning.....	6
2. Clasificación de cámaras según su forma [7] .....	17
3. Modelos Intel®RealSense™de Cámaras de profundidad [8] .....	20
4. Clasificación de servomotor por rotación [9].....	23
5. Clasificación de servomotor por uso y componentes [10] .....	24
6. Alternativas de Materiales.....	26
7. Observaciones Simulación base externa .....	38
8. Observaciones Simulación Torsión en perforaciones de sujeción.....	47
9. Modelos de servomotores de rotación continua .....	59
10. Modelos de microcontroladores .....	61
11. Prueba 1 Base de datos google .....	65
12. Prueba 2 Base de datos local.....	67
13. Prueba 3 Base de datos local con clasificador " Cascade" .....	68
14. Prueba 3 Base de datos local sin clasificador " Cascade" .....	71
15. Prueba 4 Análisis de coincidencia entre familiares .....	75
16. Prueba 5 Luminosidad en la Base de Datos.....	77
17. Prueba 6 Luminosidad externa.....	79
18. Prueba 7 ángulo mínimo para la detección de rostro.....	81
19. Prueba 8 Ángulo de la Base de Datos.....	83
20. Prueba 9 Distancia máxima de reconocimiento .....	84
21. Prueba 10 Calidad de imágenes .....	86
22. Prueba 11 Calidad de imágenes dentro de la base de datos.....	88
23. Prueba 13 Distancia máxima de reconocimiento con cámara normal.....	91
24. Prueba 13 Ángulo mínimo para la detección de rostro con cámara normal ....	91
25. Características Sistemas Hikvision .....	94

## ÍNDICE DE ANEXOS

Anexo A: Trabajos Relacionados.....	111
-------------------------------------	-----

Anexo B: Análisis de Alternativas ..... 140

Anexo C: Algoritmo de Control ..... 146

Anexo D: Planos de Construcción ..... 166

# CÁMARA DE SEGURIDAD CON DEEP LEARNING

## 1. Tema

El tema del proyecto planteado es el diseño y simulación de un sistema de reconocimiento facial basado en deep learning para seguridad en zonas internas con una cámara de profundidad y un movimiento de 360 grados.

## 2. Objetivos

### 2.1. General

Diseñar y simular un sistema de reconocimiento facial basado en Deep Learning para seguridad en zonas internas con una cámara de profundidad y un movimiento de 360 grados.

### 2.2. Específicos

- Investigar y documentar la teoría necesaria para la comprensión del método de Deep Learning, el reconocimiento facial y su respectiva aplicación dentro de esta.
- Identificar el método adecuado para los sistemas y mecanismos necesarios para el diseño de una cámara de seguridad tomando en consideración la implementación de una cámara de profundidad.
- Dimensionar y diseñar los elementos mecánicos; componentes móviles y electrónicos para el sistema de detección de rostros en zonas interiores.
- Implementar un software basado en Deep Learning en Python<sup>TM</sup> y Open CV que permita el reconocimiento facial en tiempo real.
- Simular el reconocimiento facial en tiempo real mediante la creación de una base de datos para la comprobación del correcto funcionamiento del sistema.

### **3. Problema**

La inseguridad existe a nivel mundial, el "60 % de las personas que viven en ciudades de países en vías de desarrollo han sido víctimas de la delincuencia por lo menos una vez en los últimos cinco años. La urbanización ha sido acompañada por un aumento en los niveles de delincuencia, violencia y desorden, convirtiendo a Latinoamérica en la región más afectada por la criminalidad en el mundo". [11]

En la ciudad de Quito desde el 1 de enero hasta el 6 de noviembre del año 2020 existieron 9118 denuncias de delitos en contra de transeúntes, automóviles, motocicletas, locales comerciales y viviendas. Esta es una reducción del 31 % en comparación con los datos del año anterior, lo cual es positivo pero también se debe considerar las circunstancias que rodean la pandemia mundial y que no todos los crímenes son denunciados.

El Municipio ha planteado aplicar planes para colaborar con la seguridad de la ciudadanía en la implementación de cámaras de seguridad de las cuales algunas cuentan con capacidad de identificación facial. El problema recae en que el software de estas cámaras con reconocimiento facial requieren una licencia con un costo elevado. [12]

### **4. Hipótesis**

El sistema de reconocimiento facial basado en Deep Learning para seguridad en zonas internas es de bajo costo y de fácil adquisición, reduce la cantidad de delitos de locales comerciales y viviendas dentro del Distrito Metropolitano de Quito.

### **5. Estudio Teórico**

#### **5.1. Introducción**

La violencia y la inseguridad exponen a la población a numerosas violaciones de los derechos humanos. En América Latina y Caribe se tienen las cifras de criminalidad más altas. Todo ello lastra poderosamente el desarrollo de los países a los que estas dinámicas golpean con mayor fuerza. [13]

Para hacer frente a esta situación, el Estado aplica mecanismos de seguridad en colaboración con la ciudadanía y otras organizaciones para mantener el orden público, la

erradicación de la violencia. Para asegurar la convivencia y su desarrollo mediante diferentes estrategias el Estado dispone de los poderes legislativos para generar diferentes leyes y normas así como la fuerza policial y el ejército. Las instituciones no gubernamentales también ofrecen servicios de seguridad privada. [14]

En la actualidad para la reducción de la inseguridad se han aplicado diferentes tecnologías tanto para la prevención como para la resolución de casos tales como: Videovigilancia, alarmas, sensores, cercos electrónicos, entre otros.

La Videovigilancia es uno de los mejores aliados para las fuerzas policiales y es un servicio comúnmente ocupado para seguridad privada de hogares y negocios. Estas son herramientas que ayudan a la prevención y atención de diversas emergencias y delitos disminuyendo de una manera notable la inseguridad. En los últimos años gracias al avance de la tecnología se han implementado a nivel mundial cámaras de seguridad con la capacidad de reconocer rostros. [15]

Dentro de las últimas décadas el reconocimiento visual artificial se ha posicionado como una aplicación crucial en ciertos campos de la ingeniería, especialmente en la robótica. Ya que tiene por objetivo dar a los sistemas robóticos una aproximación de la capacidad del sentido de la vista humana. El reconocimiento visual dentro de los sistemas robóticos busca generar detección en su entorno e interactuar con él según la información recogida de manera similar a como lo hace el cerebro humano. [16]

Las primeras investigaciones acerca de la visión por computador se iniciaron alrededor de los años 50. Desde aquella época han surgido diferentes técnicas a partir de las cuales se han ido mejorando los resultados obteniendo un margen de error más bajo.

Hoy en día las computadoras son muy eficientes resolviendo problemas que representan un mayor grado de dificultad para el ser humano. No obstante, al momento de resolver tareas intuitivas, sencillas e incluso automáticas como la identificación de objetos, las computadoras aun muestran falencias en comparación con la percepción humana en estas tareas. Siendo así la inteligencia artificial uno de los mecanismo por el cual una computadora puede lograr esto.

La inteligencia artificial tiene una infinidad de campos que continúan evolucionando dando la solución a una gran cantidad de problemas que siguen apareciendo cada día. Las

empresas cada vez invierten más en este campo para resolver de forma eficiente sus dilemas con mejores tolerancias, menos errores, más adaptabilidad, menor costo a largo plazo y muchas otras ventajas que da este mecanismo.

La inteligencia artificial se encuentra en auge en la sociedad por la curiosidad por entender cómo se trabajan las principales aplicaciones en este campo.

Deep Learning o Aprendizaje Profundo es cada vez más usado para resolver estas tareas sencillas para las personas pero difíciles para las computadoras. El Deep Learning es un nuevo paradigma del Machine Learning, otro conjunto de algoritmos de aprendizaje y un diferente tipo de redes neuronales. [17]

Con la técnica de Deep Learning se ha conseguido llegar a un nivel superior de tratamiento de datos comparado al Machine Learning, esto mediante las redes neuronales convolucionales.

La detección de objeto tiene una gran cantidad de aplicaciones tecnológicas como el reconocimiento facial, el reconocimiento de tráfico en automóviles auto-pilotados, la ingeniería biomédica, entre muchos otros, dando un gran avance hacia la inteligencia artificial dentro de la industria. [18]

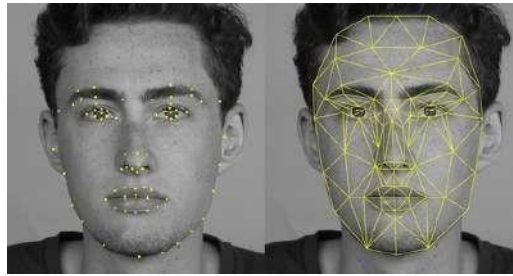
## **5.2. Reconocimiento Facial**

El Reconocimiento facial es un sistema basado en algoritmos de procesamiento de imágenes o vídeos, encargados de identificar a las personas de forma automática mediante el uso de cámaras. [19]

La tecnología de reconocimiento facial busca la verificación, autenticación, identificación o reconocimiento de caras a partir de comparaciones entre los atributos de la cara de la persona que se quiere identificar y los de millones de individuos con características similares contenidas en una enorme base de datos.

Para la obtención de los rasgos de la cara en general el reconocimiento facial se basa en un sistema biométrico que se fundamenta en la colocación de puntos sobre la imagen y el cálculo de sus distancias, esto para dar una característica única al rostro analizado como se puede ver en la Figura 1. El reconocimiento de patrones utiliza varias técnicas entre estas la de las redes neuronales; y puesto que el reconocimiento facial es un área

contenida en el reconocimiento de patrones, esta le permite emplear las mismas técnicas para su desarrollo. [20]



**Figura 1.** Sistema biométrico facial [1]

### 5.3. inteligencia Artificial

La inteligencia artificial es una rama de la Informática encargada de estudiar y diseñar algoritmos que le permiten a una máquina o sistema informático tomar decisiones y actuar según varios parámetros; el propósito de un sistema de inteligencia artificial es el aprendizaje tras recibir datos de un medio, realizando una acción según vaya recopilando más información y simulando de esta manera el comportamiento de la lógica humana.

Los métodos y técnicas de la inteligencia artificial se aplican para resolver problemas, de acuerdo a procesos y reglas metodológicas. El éxito de la tarea programada para resolver problemas depende de la dificultad de la misma y la capacidad de adquisición de datos, extracción de características y toma de decisiones, con el fin de maximizar y optimizar el tiempo y recursos empleados. [21]

Existen algoritmos en la inteligencia artificial que facilitan la búsqueda, comparación, clasificación y discretización de información, para posteriormente generar un proceso de aprendizaje. De manera general Nils John Nilsson sintetizó los 4 pilares de la inteligencia artificial en los siguientes procesos de algoritmos: Búsqueda del estado, Algoritmos genéticos, Redes neuronales artificiales y Razonamiento mediante una lógica formal. [22]

A partir de estos grupos se puede generar tareas más específicas que utilizan los algoritmos planteados, como por ejemplo el reconocimiento de patrones, que trata de la extracción de objetos físicos o abstractos lo que permite determinar las características y propiedades de los mismos. En el caso de este proyecto se utilizará la comparación de las diferentes características del rostro.



## 5.4. Machine Learning

Machine Learning que en español significa aprendizaje automático o de máquina es una técnica de análisis de datos que enseña a las máquinas a imitar la capacidad del ser humano para procesar la información. Los algoritmos de aprendizaje automático emplean métodos de cálculo para desarrollar la capacidad de generalizar y asociar directamente los datos sin depender de una ecuación predeterminada como modelo de manera precisa y exacta. Los algoritmos mejoran su rendimiento de forma adaptativa a medida que aumenta el número de muestras disponibles para el aprendizaje. [23]

Existen 3 tipos de algoritmos que se basan en Machine Learning, tomando en cuenta los datos que se dispongan para entrenar al sistema como se puede ver en la Tabla 1.

**Tabla 1.** Algoritmos de Machine Learning

<b>Aprendizaje Supervisado</b>	<b>Aprendizaje no Supervisado</b>	<b>Aprendizaje de Refuerzo</b>
Clasificación	Clustering	Procesos de decisión
Regresión	Asociación de Marketing	Sistemas de premios
Ranking	Segmentación	
	Reducción de dimensionalidad	

El aprendizaje supervisado, es un modelo predictivo donde los conjuntos de datos obtenidos son etiquetados con el fin de facilitar las tareas de detección de los patrones, luego de ello son nuevamente utilizados en el proceso de etiquetar un conjunto de datos recientes, permitiendo que la máquina aprenda explícitamente y que tengan la capacidad de predecir el futuro a partir de históricos, resolver problemas de clasificación y regresión.

El aprendizaje no supervisado, es un modelo descriptivo donde el conjunto de datos no se encuentra etiquetado, por ende, basan su clasificación a partir de diferencias y coincidencias o similitudes, permitiendo que la máquina entienda los datos de manera cualitativa o indirecta, por lo que la máquina con este aprendizaje solo tiene la capacidad de encontrar datos específicos mas no predecirlos.

Por último el aprendizaje de refuerzo, al igual que el anterior, el conjunto de datos tomados no cuentan con etiquetas, no obstante su particularidad se presenta cuando se hayan llevado a cabo una o varias acciones de etiqueta, el sistema de Inteligencia Artificial recibe realimentación a partir de los patrones conseguidos, por lo que la máquina aprende como actuar en un determinado entorno. [24]

## 5.5. Deep Learning

Deep Learning es un aprendizaje automático que se lo puede describir como una evolución del Machine Learning ya que de igual forma que este imita el aprendizaje humano pero de una manera más semejante al cerebro humano, ya que se basa en el uso de las redes neuronales y utiliza varias capas de procesamiento no lineal para aprender características directamente de imágenes, textos o señales. [25]

Una manera básica de visualizar su funcionamiento es con niveles jerárquicos, donde existen diferentes capas que analizan desde lo más simple a lo más complejo, comunicándose entre ellas para que el sistema aprenda.

El Deep Learning como se menciona anteriormente ha conseguido dar un paso más para el análisis de la información, ya que es de suma importancia en la actualidad el descubrimiento y reconocimiento de patrones para el desarrollo científico y tecnológico. Su potencial se evidencia en las aplicaciones industriales que brinda como:

- Utilización de imágenes en lugar de palabras clave para búsqueda de productos.
- Identificar marcas y logotipos de empresas en fotografías.
- Monitorización en tiempo real de reacciones durante el lanzamiento de productos.
- Orientación de anuncios y predicción de las preferencias de los clientes.
- Identificación y seguimiento de los niveles de confianza de los clientes.
- Identificación de clientes potenciales.
- Detección de fraudes.
- Recomendaciones a clientes.

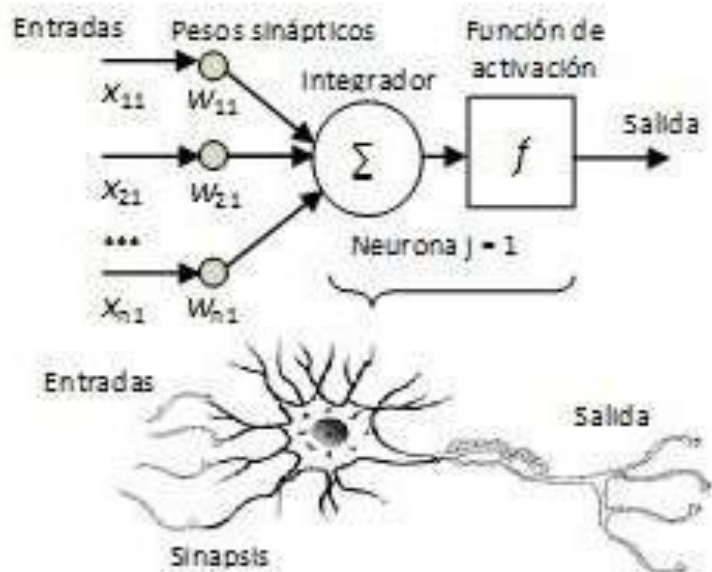
- Gestión de relaciones con los clientes.
- Mejor comprensión de enfermedades, mutaciones de enfermedades y terapias genéticas.
- Análisis de imágenes médicas, como radiografías y resonancias magnéticas.
- Detección, predicción y prevención de amenazas sofisticadas en tiempo real en el campo de la ciberseguridad.
- Localización de caras e identificación de emociones faciales.
- Reconocimiento facial.
- Reconocimiento de voz.
- Clasificación de vídeos. [26]

## **5.6. Redes Neuronales Artificiales**

Las Redes Neuronales Artificiales son un sistema de procesamiento de información mediante unidades básicas de procesamiento que se basan en el funcionamiento de la célula fundamental del sistema nervioso humano: la neurona.

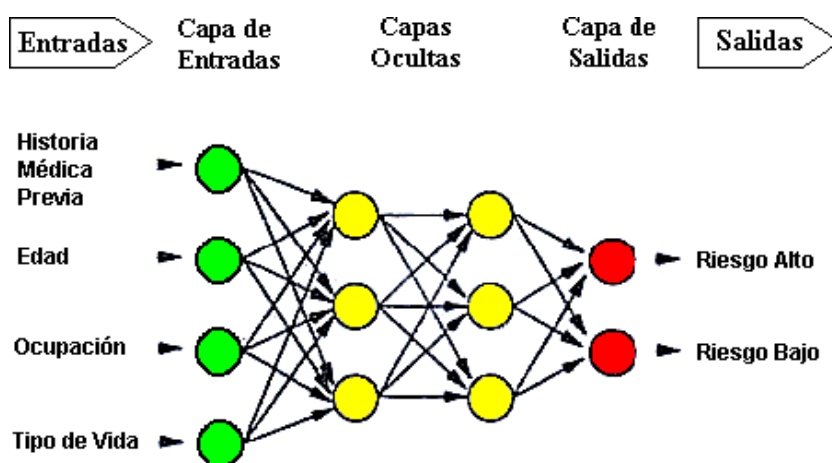
La neurona tiene la capacidad de interconectarse con otras neuronas para intercambiar información entre ellas y formar redes neuronales.

Las neuronas artificiales se comportan de la misma manera que una neurona que recibe una serie de señales de entrada, que harán activar la neurona según una determinada función de activación, generando una señal de salida que puede ser transmitida hacia otra neurona como se puede observar en la Figura 2. [27]



**Figura 2.** Relación entre neurona artificial y neurona biológica [2]

Dentro de las redes neuronales, las neuronas que la componen se encuentran organizadas en capas de neuronas, cada una de ellas procesa la información recibida de la capa anterior, obteniendo una salida que irá a la capa siguiente. Todas las redes neuronales tendrán tres tipos de capas: una capa de entrada, una capa oculta y una capa de salida como se observa en la Figura 3.



**Figura 3.** Estructura de una red neuronal [3]

La capa inicial de entrada recibe información que provienen de sensores o de bases de datos con un peso y los transmite a la primera capa oculta. Esta capa tendrá tantas neuronas como entradas tenga la red.

La capa intermedia denominada capa oculta realiza todo el procesamiento de la información, para finalmente obtener la respuesta al exterior que se desea. La complejidad de cada red neuronal vendrá determinada por el número de capas ocultas que tenga la misma, así como de la función de activación de las neuronas.

La capa final de salida contiene el resultado realizado anteriormente en la capa oculta. Su número depende de la cantidad de clasificaciones que se requiera.

El machine learning con su proceso de entrenamiento modifica los pesos de las conexiones para realizar el reconocimiento ideal, sin la necesidad de atributos específicos como sucede en otros métodos. Sin embargo existe un inconveniente en su implementación cuando se trata de un conjunto de datos de entrada de gran tamaño, ya que supone el hecho de que las neuronas estén completamente interconectadas entre sí. Para ofrecer una solución a este problema surgen las Redes Neuronales Convolucionales. [28]

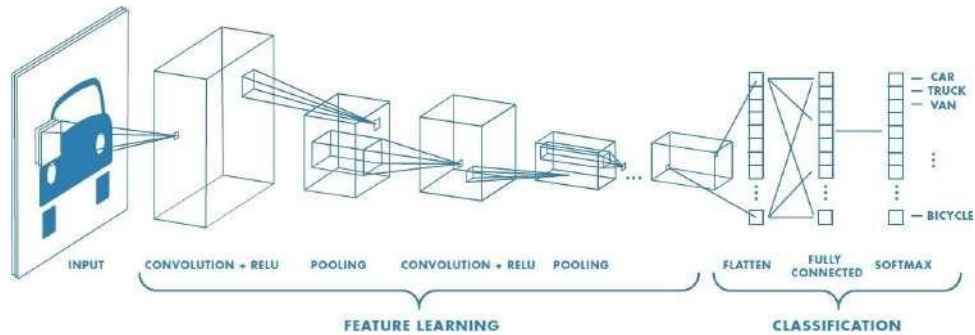
### **5.7. Redes Neuronales Convolucionales**

Una red neuronal convolucional (CNN), es una arquitectura de Deep learning, que se basa en el funcionamiento de la corteza visual cerebral.

Este tipo especial de Redes Neuronales, se diferencia de las demás por el hecho de que cada una de las neuronas de las capas que la componen no recibe conexiones entrantes de todas las neuronas de la capa anterior, sino sólo de algunas de ellas.

El algoritmo de detección se divide en dos etapas, la extracción de características del contenido de la imagen y la posterior búsqueda de objetos basada en dichas características para su correcta clasificación. [29]

Las redes neuronales convolucionales están formadas por una sucesión de capas convolucionales y agrupaciones (pooling), donde en una primera instancia se lleva a cabo la extracción de características, posteriormente una red neuronal totalmente conectada se encarga de calcular las puntuaciones obtenidas por la imagen de entrada para cada una de las clases o categorías definidas en el problema como se puede observar en la Figura 4.



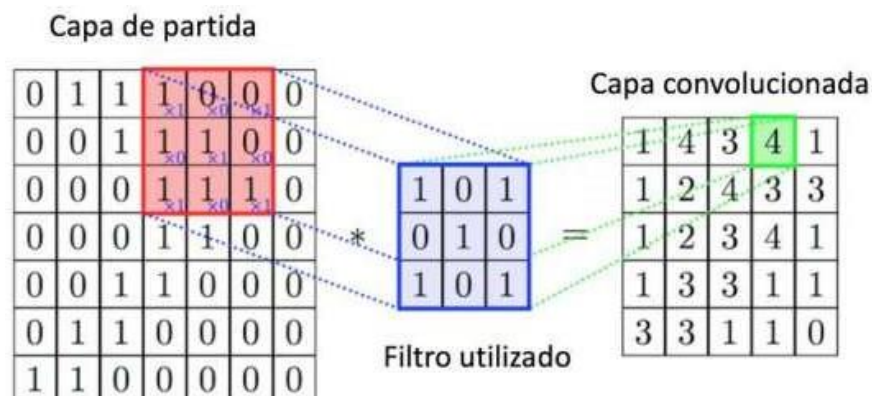
**Figura 4.** Estructura de una red neuronal convolucional [4]

En la capa de convolución de la Red Neuronal, se le aplica a la imagen una operación de convolución con un filtro determinado o kernel para cometidos como el desenfoco, el aumento de nitidez, acentuar el relieve o la detección de bordes en una imagen.

Cuando se habla de convolución en términos de aprendizaje automático, visión por computador o redes neuronales, hay tres conceptos principales a tener en cuenta: La primera es la entrada que generalmente es una imagen representada como  $I$ . La segunda es el kernel o filtro, el cual se representa con la letra  $K$ . Y la tercera es la salida, que es el mapa de características. La convolución de un kernel  $K$  de tamaño  $m \times n$  colocado en el píxel  $(i, j)$  de una imagen  $I$  se representa mediante la ecuación (1).

$$(I \times K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (1)$$

Las capas convolucionales de las redes normalmente aplican varias convoluciones en paralelo en diferentes puntos  $(i, j)$ , esto puede entenderse mejor visualizando un kernel como una ventana deslizante que recorre toda la imagen y calcula el mapa de características resultante en cada salto como se puede ver en la Figura 5.

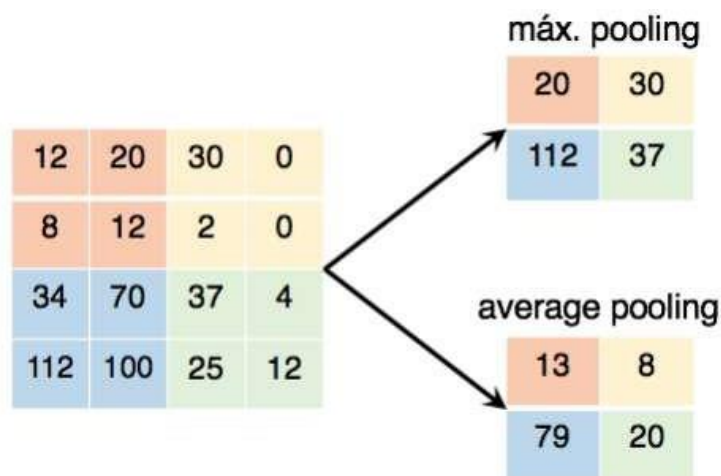


**Figura 5.** Convolución [5]

A la salida de una capa convolucional normalmente se aplica el ReLU que significa Rectified Linear Unit, la cual es una función de activación que se encarga de convertir los valores negativos a 0, con la finalidad de introducir la no linealidad en un sistema que ha calculado operaciones lineales durante la capa convolucional.

Una vez extraídas las características mediante la capa de convolución se continúa con la capa de Pooling, donde se agrupan las neuronas y se reducen las dimensiones para obtener las propiedades más relevantes de los datos de entrada y obviar datos superfluos que pueden inducir a error.

La capa de Pooling se puede dar de dos maneras: Max Pooling y Average Pooling. El Max Pooling es cuando se desliza un filtro de tamaño  $m \times m$  a través de la matriz de entrada y se selecciona el valor mayor de la matriz de entrada mientras que Average Pooling es cuando se desliza un filtro pero se selecciona la media de dicha matriz como se observa en la Figura 6.



**Figura 6.** Pooling [6]

Después de añadir varias capas sucesivas de convolución y pooling respectivamente, se añaden una o dos capas completamente conectadas (FC) que cumplen con el objetivo de la clasificación. Requiere de una adaptación Flatten. Finalmente, la capa Softmax se encarga de asociar una etiqueta a la imagen de entrada aplicando una probabilidad de semejanza como se puede observar en la Figura 4. [30]

## 5.8. Trabajos Relacionados

Existen infinidad de estudios y equipos creados con sistemas de Deep Learning, se realizó el análisis de treinta proyectos los cuales se pueden ver en el Anexo A. Este método tiene un gran potencial para la manipulación de un elevado número de datos sin saturar el sistema con demasiadas restricciones (como sucede en el Machine Learning), mediante diferentes métodos.

Entre estos se encuentra la investigación "FaceTime – Deep Learning Based Face Recognition Attendance System", realizado por Marko Arsenovic, Srdjan Sladojevic, Andras Anderla y Darko Stefanovic, presentada en la conferencia IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY).<sup>en</sup> el año 2017. En este proyecto se se propuso un sistema de reconocimiento facial mediante dos técnicas de aprendizaje profundo: Redes neuronales convolucionales (CNN) y redes neuronales convolucionales en cascada (CNN cascade) para su empleo práctico. El resultado que se obtuvo fue de 95.02 % es decir una precisión alta mediante el método propuesto con un sistema de asistencia y la combinación de métodos de aprendizaje profundo. [31]

Otra investigación que se puede mencionar es ".^ Fast Face Recognition System Based On Deep Learning", realizado por Xiujie Qu, Tianbo Wei, Cheng Peng y Peng Du, presentada en la conferencia "2018 11th International Symposium on Computational Intelligence and Design (ISCID).<sup>en</sup> el año 2018. En este proyecto se propuso un método de reconocimiento facial en tiempo real en FPGA, que mejora la velocidad y la precisión del reconocimiento facial. Este método es basado en el principio de la red neuronal convolucional (CNN). Como resultado, la velocidad de reconocimiento del sistema alcanza los 400 fps, superando con creces los resultados existentes. La tasa de reconocimiento fue del 99,25 %, superior a la del ojo humano. Además, posee una buena robustez para la luz del ambiente. [32]

Como conclusión del análisis de los sistemas existentes, el método de Deep Learning con mejores resultados es el de las redes neuronales convolucionales. No se encontró trabajos existentes que analicen el uso de una cámara de profundidad ni siquiera detallan la cámara ocupada.



## **6. Análisis y Selección de Alternativas**

### **6.1. Especificaciones Técnicas**

Se realiza el análisis de especificaciones técnicas a partir del estudio de los trabajos existentes y de las cámaras de seguridad en el mercado actual. De esta forma se diseña un proyecto de calidad. Los requerimientos son los detallados a continuación:

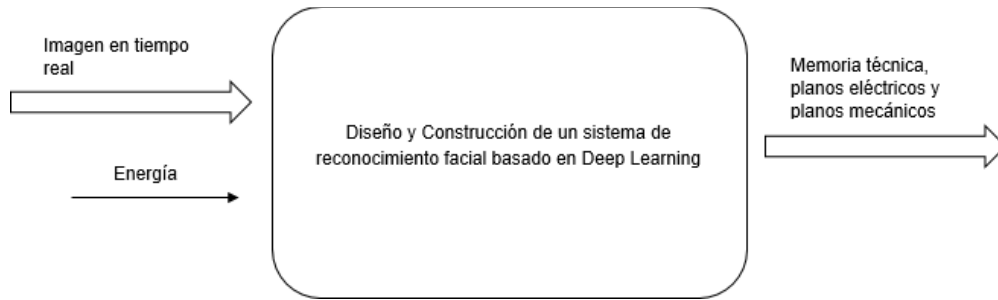
- Cámara de profundidad Intel®RealSense™.
- Movimiento horizontal y vertical 360 grados.
- Análisis de imagen en tiempo real mediante Deep Learning.
- Software Python™OpenCV.
- Cámara de seguridad para el uso en interiores.

### **6.2. Análisis Funcional**

Este análisis se lo realiza descomponiendo el sistema a diseñar en varios bloques funcionales, los cuales interactúan entre sí de forma que cumplan con el objetivo por el cual fue diseñada la máquina.

#### **6.2.1. Nivel 0**

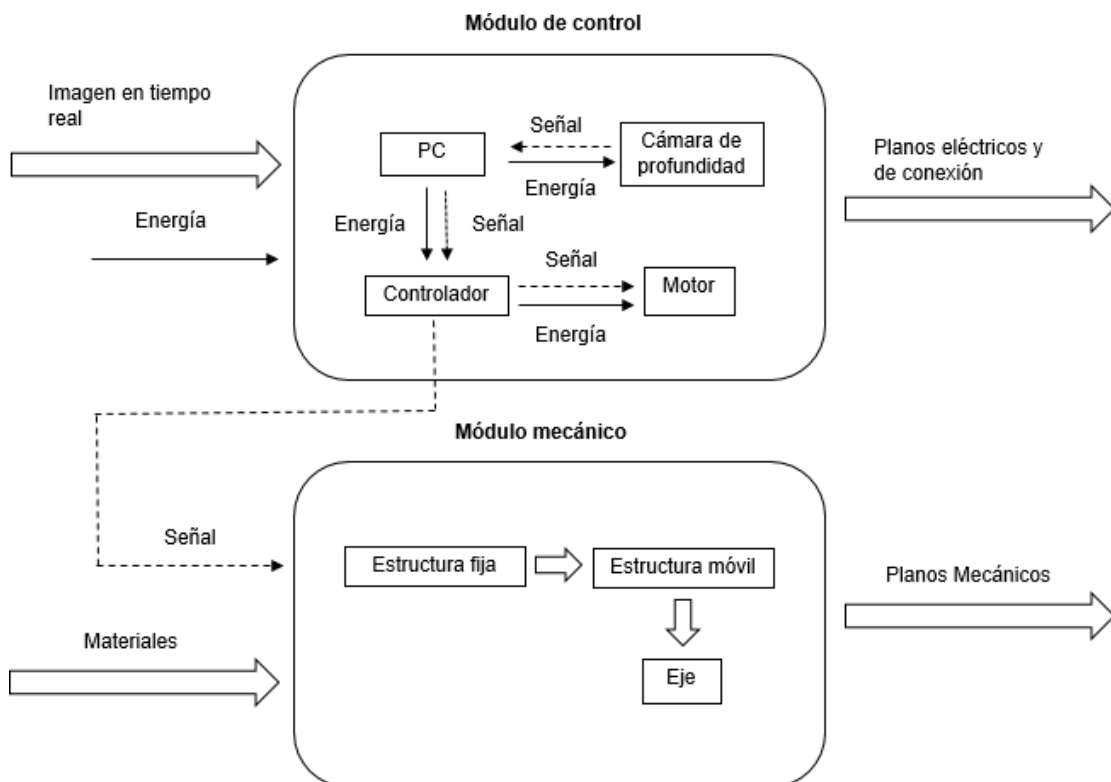
Se presenta el proceso de manera general, el cual se tiene en el ingreso de energía y señal, con lo que se busca generar el diseño de un sistema de reconocimiento facial basado en Deep Learning para obtener la memoria técnica, planos eléctricos, planos mecánicos y una programación basada en el mencionado método. En la Figura 7 se presenta el nivel 0.



**Figura 7.** Análisis funcional Nivel 0

### 6.2.2. Nivel 1

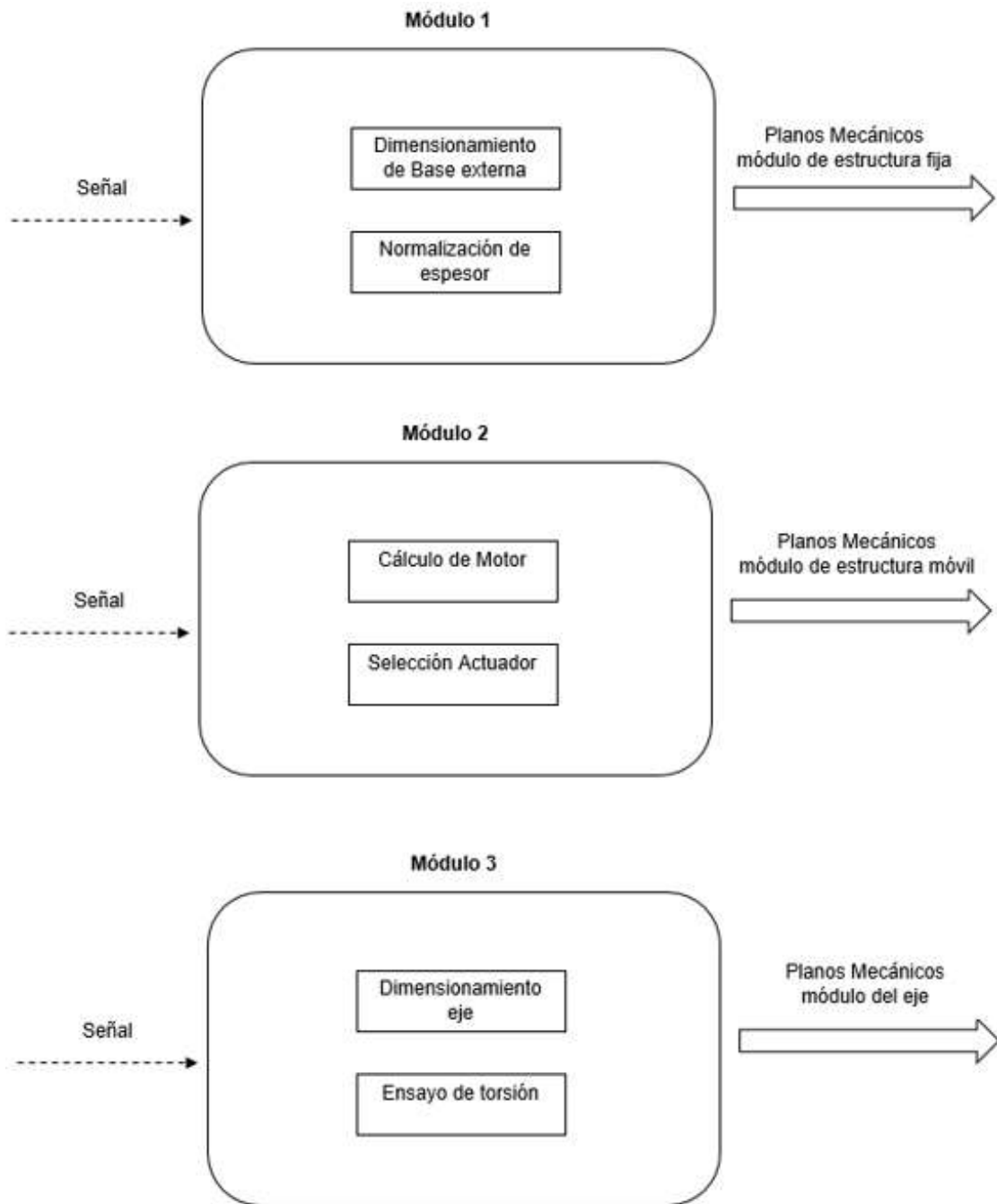
El Nivel 1 genera 2 módulos, uno de control y otro mecánico, en los cuales se representa todos los procesos involucrados en el diseño de la máquina. En la Figura 8 se presenta el Nivel 1.



**Figura 8.** Análisis funcional Nivel 1

### 6.2.3. Nivel 2

Una vez creado el Nivel 1 se procede a construir el Nivel 2, del cual se generan 3 módulos que son: el módulo 1 de estructura fija, módulo 2 de estructura móvil y el módulo 3 de eje. Estos módulos muestran de forma más detallada los procesos del Nivel 1. En la Figura 9 se presentan los cuatro módulos del Nivel 2.



**Figura 9.** Análisis funcional Nivel 2

### 6.3. Alternativas

#### 6.3.1. Cámara


En el mercado se pueden encontrar multitud de cámaras y sistemas de videovigilancia pero en general se pueden clasificar de acuerdo a su forma, aspecto y características. Cada cámara puede ser utilizada en alguna situación en particular, para brindarle a los usuarios una solución especial para su necesidad o requerimiento.

Para seleccionar la estructura del prototipo se necesita analizar las cámaras de seguridad en el mercado según su forma como se puede ver en la Tabla 2.

**Tabla 2.** Clasificación de cámaras según su forma [7]

Tipo de cámara	Descripción	Fotografía
Domo	Se gradúan manualmente. Facilidad de instalación. Generalmente visión nocturna. Económica. Sujeción directa a la superficie. Uso interno.	
Cubo	Mayor facilidad de instalación. Soporte. Transmisión de vídeo por internet. Conectividad inalámbrica. Uso interno.	

Tipo de cámara	Descripción	Fotografía
Robóticas	<p>Movimiento remoto.</p> <p>Almacenamiento de vídeo.</p> <p>Transmisión de vídeo por internet.</p> <p>Costosas.</p> <p>Uso interno y difícil uso externo.</p>	
Camuflada	<p>No llaman la atención.</p> <p>Desconocimiento de las personas que se les esta grabando.</p> <p>Acoplado en otros elementos como por ejemplo espejos, sensores, entre otros.</p> <p>Uso interno.</p>	
Bala	<p>Mayor capacidad de visión nocturna.</p> <p>No le afecta lluvia o sol.</p> <p>Mayor resistencia.</p> <p>Soporte.</p> <p>Generalmente son fijas.</p> <p>Uso externo.</p>	

Tipo de cámara	Descripción	Fotografía
Ciber Domo PTZ	<p>Combinación principio de funcionamiento cámaras tipo domo y robóticas.</p> <p>Movimiento 360 grados.</p> <p>Alto rango de calibración en cuanto a su ángulo de visión.</p> <p>Visión Nocturna.</p> <p>Más avanzadas tecnológicamente.</p> <p>Uso interno y difícil uso externo.</p>	

Tomando en cuenta la matriz de decisión de Pugh que se puede visualizar con mayor detalle en el Anexo B.1, se toma la decisión de optar por la alternativa de la estructura de la cámara tipo domo con los mecanismos de la tipo PTZ para obtener: una fácil instalación, sujeción directa a la superficie, protección de los elementos internos, capacidad de uso en zonas internas y lo más importante un movimiento de 360 grados.


### 6.3.2. Cámara de profundidad

Dentro de las especificaciones técnicas consta el uso de una cámara con sensor de profundidad de Intel® RealSense™. Esta tecnología tiene la capacidad de ver el entorno, comprenderlo, interactuar con él e incluso aprender de él.

Intel® ofrece un catálogo con los diferentes modelos los cuales serán detallados en la Tabla 3.

**Tabla 3.** Modelos Intel®RealSense™ de Cámaras de profundidad [8]

Modelo	Especificaciones	Fotografía
T265	<p>Costo \$200.</p> <p>V-SLAM.</p> <p>Intel®Movidius™Myriad™ 2.0 VPU.</p> <p>Dos lentes tipo ojo de pez con una combinación de 1635 Campo de Visión de Profundidad (FOV).</p> <p>BMI055 IMU.</p> <p>No tiene estructura externa.</p>	
D435i	<p>Costo \$200.</p> <p>BMI055.</p> <p>Uso interno o externo.</p> <p>Tiene estructura externa.</p> <p>IR stereo.</p> <p>FOV 873 x 581 x 953.</p> <p>Resolución 1280 x 720.</p> <p>90 cuadros por segundo (fps).</p> <p>Distancia de profundidad 0.1 m.</p> <p>Rango de 10 m.</p> <p>Obturador global.</p> <p>Sensor RGB 1920 x 1080 a 30 fps, 69.4 x 42.5 x 77 (+/- 3) FOV.</p> <p>Dimensiones 90 mm x 25 mm x 25 mm.</p> <p>Conector USB-C* 3.1 Gen 1.</p> <p>Montura perno 1/4-20 UNC.</p> <p>Requerimiento de procesador alta.</p>	

Modelo	Especificaciones	Fotografía
D435	<p>Costo \$180.</p> <p>Uso interno o externo.</p> <p>Tiene estructura externa.</p> <p>IR stereo.</p> <p>FOV 873 x 581 x 953.</p> <p>Resolución 1280 x 720.</p> <p>90 cuadros por segundo (fps).</p> <p>Distancia de profundidad 0.1 m.</p> <p>Rango de 10 m.</p> <p>Obturador global.</p> <p>Sensor RGB 1920 x 1080 a 30 fps, 69.4 x 42.5 x 77 (+/- 3) FOV.</p> <p>Dimensiones 90 mm x 25 mm x 25 mm.</p> <p>Conector USB-C* 3.1 Gen 1.</p> <p>Montura perno 1/4-20 UNC.</p> <p>Requerimiento de procesador media-alta.</p>	



Modelo	Especificaciones	Fotografía
D415	<p>Costo \$150.</p> <p>Uso interno o externo.</p> <p>Tiene estructura externa.</p> <p>IR stereo.</p> <p>FOV 652 x 401 x 722.</p> <p>Resolución 1280 x 720.</p> <p>90 cuadros por segundo (fps).</p> <p>Distancia de profundidad 0.3 m.</p> <p>Rango de 10 m.</p> <p>Obturador global.</p> <p>Sensor RGB 1920 x 1080 a 30 fps, 69.4 x 42.5 x 77 (+/- 3) FOV.</p> <p>Dimensiones 99 mm x 20 mm x 23 mm.</p> <p>Conector USB-C* 3.1 Gen 1.</p> <p>Montura perno 1/4-20 UNC.</p> <p>Requerimiento de procesador media.</p>	
SR300	<p>Costo \$70.</p> <p>Uso interno.</p> <p>Tecnología de Luz codificada.</p> <p>Sensor RGB 1920 x 1080 a 30 fps, 68 x 41.5 (2) FOV.</p> <p>FOV 693 x 542.</p> <p>Resolución 640 x 480.</p> <p>60 cuadros por segundo (fps).</p> <p>Distancia de profundidad 0.2 m.</p> <p>No tiene estructura externa.</p> <p>Rango de 1.5 m.</p> <p>Requerimiento de procesador baja.</p>	

Tomando en cuenta la matriz de decisión de Pugh que se puede visualizar con mayor detalle en el Anexo B.2, se toma la decisión de optar por la alternativa de la cámara Intel® RealSense™ modelo D415 porque satisface el rango requerido, tiene características aceptables en lo que respecta a su tecnología de profundidad, sus características ópticas y el requerimiento de software. Además este modelo tiene una estructura externa que permite la sujeción adecuada en el prototipo.

### 6.3.3. Motor

Dentro de las especificaciones técnicas se tiene el requerimiento de un movimiento vertical y uno horizontal para la visualización de 360 grados de la cámara, por lo que es necesario dos motores.

Para el prototipo se ocupa servomotores porque permiten controlar la posición del eje en un momento dado. Están diseñados para moverse determinada cantidad de grados y luego mantenerse fijos en una posición.

Se puede clasificar los servomotores por su rotación como se puede observar en la Tabla 4.




**Tabla 4.** Clasificación de servomotor por rotación [9]

Tipo	Descripción
Limitada	Rotación eje limitada a 180 grados.
Continua	Rotación completa 360 grados.

El prototipo al tener la necesidad de girar en su totalidad tanto verticalmente como horizontalmente requiere rotación continua de su motor.

Existen diferentes tipos de servomotores de rotación continua que se clasifican según su uso, también por el sistema mixto de piezas electromecánicas y electrónicas que los componen como se puede ver en la Tabla 5.

**Tabla 5.** Clasificación de servomotor por uso y componentes [10]

Tipo	Descripción	Fotografía
Servomotores de corriente continua	Comunes. Asequibles. PWM. Torque bajos - medios.	
Servomotores de corriente alterna	Mayor Torque.	
Servomotores de imanes permanentes	Torque alto. Velocidades altas. Más usados en la industria.	
Motor Paso a Paso	Campo robótica. No gira avanza en pasos. PWM. Torque medio.	


Tomando en cuenta la matriz de decisión de Pugh que se puede visualizar con mayor detalle en el Anexo B.3, se toma la decisión de optar por la alternativa de servomotores de


corriente continua porque es la que mejor satisface la necesidad del tamaño. Además este tipo de servomotor tiene un torque bajo - mediano, es asequible y se tiene experiencia en el manejo de este.

#### **6.3.4. Material**

Para la selección del material del prototipo se considera el que ocupan las cámaras en el mercado como se puede observar en la Tabla 6.

**Tabla 6.** Alternativas de Materiales

Tipo	Descripción	Fotografía
ABS [33]	<p>El ABS es uno de los materiales plásticos más usados en la industria, este es muy resistente al impacto, a la abrasión y a los elementos químicos. Es un material muy utilizado en automoción y otras aplicaciones tanto industriales como domésticas.</p> <p>Densidad: 1,03 a 1,38 g/cm<sup>3</sup>.</p> <p>Módulo de elasticidad (Young): 1.7 a 2.8 GPa.</p> <p>Elongación a la rotura: 3 % al 75 %.</p> <p>Módulo de flexión: 2.1 a 7.6 GPa.</p> <p>Resistencia a la flexión: 69 a 97 MPa.</p> <p>Temperatura de transición vítrea: 100°C.</p> <p>Temperatura de deflexión del calor (a 455 kPa): 84 a 118°C.</p> <p>Capacidad calorífica específica: 1080 a 1400 J/kg-K.</p> <p>Relación fuerza-peso: 31 a 80 kN-m/kg.</p> <p>Resistencia a la tracción (UTS): 33 a 110 MPa.</p> <p>Expansión térmica: 83 a 95 m/m-K.</p>	

Tipo	Descripción	Fotografía
<p>Policarbonato [34]</p>	<p>Resistencia al impacto extremadamente elevada.</p> <p>Gran transparencia.</p> <p>Resistencia y rigidez elevadas.</p> <p>Elevada resistencia a la deformación térmica.</p> <p>Elevada estabilidad dimensional, es decir, elevada resistencia a la fluencia.</p> <p>Buenas propiedades de aislamiento eléctrico.</p> <p>Elevada resistencia a la intemperie, con protección contra rayos ultravioleta.</p> <p>Resistencia media a sustancias químicas.</p> <p>Sensibilidad al entallado y susceptibilidad a fisuras por esfuerzos.</p> <p>Sensibilidad a la hidrólisis.</p> <p>Densidad 1.14 - 1.21 g/cm<sup>3</sup></p> <p>Cristalidad Amorfo.</p> <p>Módulo elástico 2 - 2.44 GPa</p> <p>Coefficiente de Poisson 0.391 - 0.408</p> <p>Resistencia mecánica a la compresión 69 - 86.9 MPa</p> <p>Resistencia mecánica a la tracción 60 - 72.4 MPa</p> <p>Tenacidad a fractura (K<sub>Ic</sub>) 2.1 - 4.6 MPa·m<sup>1/2</sup></p>	

Tomando en cuenta la matriz de decisión de Pugh que se puede visualizar con mayor

detalle en el Anexo B.4, se toma la decisión de optar por la alternativa de ABS ya que el policarbonato es para uso externo, entre otras consideraciones.

Para el domo protector industrialmente se realiza en vidrio pero al ser un prototipo de laboratorio se opta por un domo de acrílico que cumple con la misma función de proteger y tener la transparencia necesaria para la visualización de la cámara.

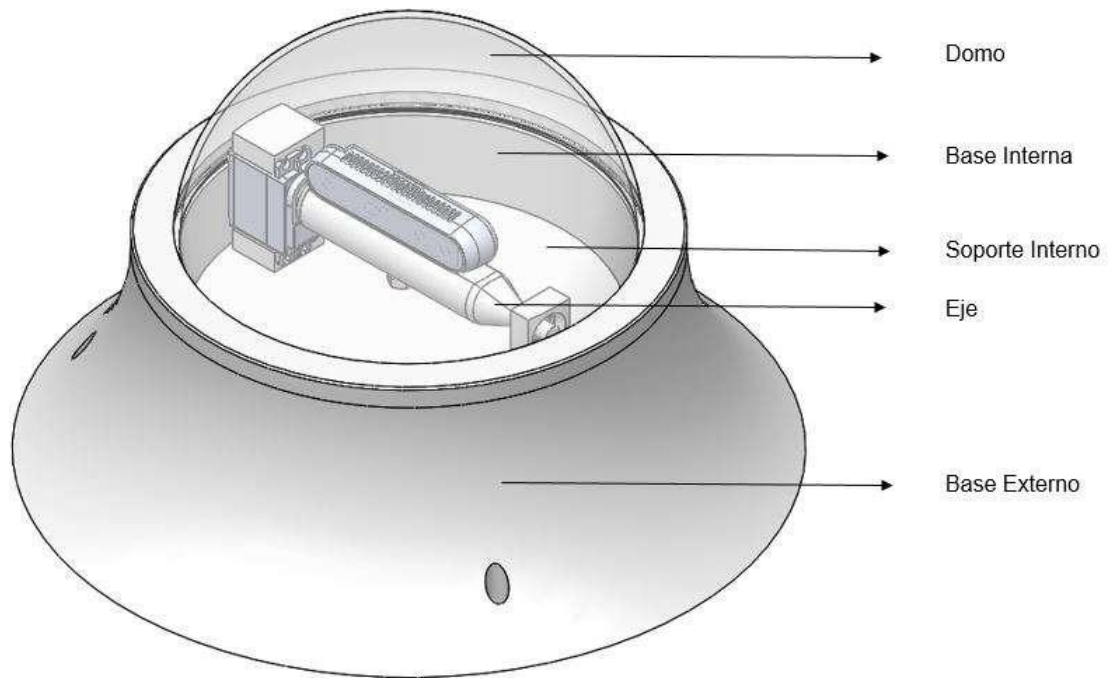
Propiedades Mecánicas del ABS [35]:

- Densidad de la masa  $\rho$ : 1210 kg/m<sup>3</sup>.
- Módulo de elasticidad  $E$ : 2900 N/mm<sup>2</sup>.
- Radio de Poisson  $\nu$ : 0.391.
- Conductividad Térmica  $\Lambda$ : 0.17 W/(m K).
- Calor específico  $c$ : 1920 J/(kg K).

Pero cabe destacar que las propiedades mecánicas de los polímeros cambian una vez impresas por diferentes factores como: el ambiente, la impresora 3D, entre otras. En consecuencia para el ABS se toma el valor de resistencia a la tracción  $S_{ut}$  de 30.483 MPa obtenidos del promedio de los diferentes ensayos realizados en la tesis realizada en la Escuela Politécnica Nacional [36].

## 7. Diseño Mecánico

El diseño de la estructura se realiza en el software de diseño SOLIDWORKS® Student Edition 2019-2020, el cual además de dar una vista previa de los elementos necesarios, ofrece facilidad en la creación de planos y vistas para los anexos del proyecto. En la Figura 10 se puede observar el prototipo con los elementos que le componen.



**Figura 10.** Vista General Prototipo

Los elementos a ser dimensionados son la base externa, el eje y los servomotores. Los cálculos fueron basados en el libro [37].

### 7.1. Consideraciones de diseño

- Sistema para zonas internas.
- El prototipo es realizado con el polímero ABS.
- El ABS es un material frágil.
- Tiene un altura máxima de 35 cm.
- El diámetro máximo de la base externa es de 35 cm.
- El servomotor tiene un peso máximo de 100 g.
- El microcontrolador tiene un peso máximo de 50 g.
- Debe soportar el peso de la cámara de profundidad el cual es de 250 g.
- Valor de gravedad ocupado  $9.8 \text{ m/s}^2$ .

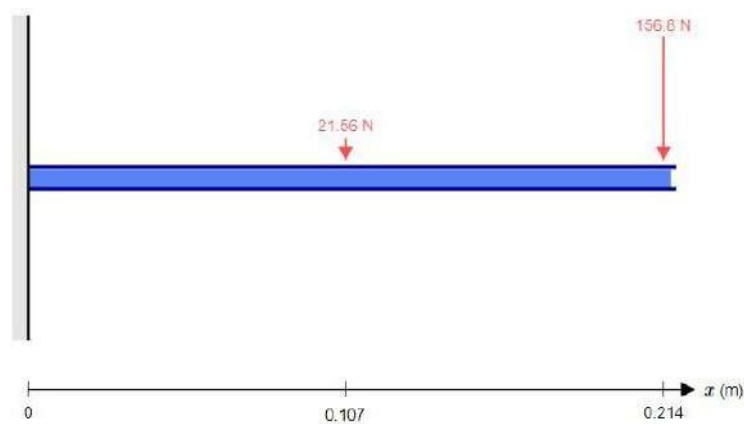


## 7.2. Dimensionamiento Base Externa

### 7.2.1. Cargas Puntuales

El cálculo de la base externa se realiza en voladizo como se observa en la Figura 11 con espesor global de 10mm para la obtención de las dimensiones y pesos de todos los componentes, los cuales son las cargas de este elemento.

El objetivo de este cálculo es la obtención de un espesor inicial ya que la forma de la base externa es compleja.

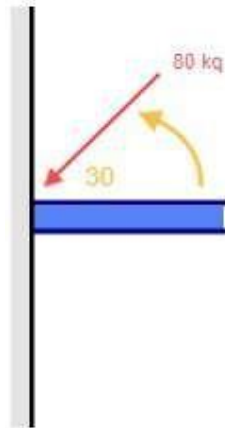


**Figura 11.** Diagrama de cuerpo libre viga en voladizo cargas puntuales

Datos iniciales:

- Peso Carga  $m$ : 2.2 kg.
- Resistencia a la Tracción  $S_{ut}$ : 30.483 MPa.
- Longitud  $b$ : 0.214 m.

Para poder determinar el valor de  $F_2$  se realizó un experimento para verificar la fuerza de apoyo de una persona de peso promedio (80 kg) [38] a 30 grados mediante una balanza como se puede ver en la Figura 12.



**Figura 12.** Diagrama de cuerpo libre fuerza de apoyo

Se comprobó que la fuerza de apoyo resultante es del 20 % del peso total de la persona, específicamente es de 16 kg.

A partir de los datos iniciales se calcula  $F_1$  mediante (2) y se procede a calcular el momento con (3).

$$F = m \times g \quad (2)$$

Donde

$F$  fuerza, en N;

$m$  peso, en kg;

$g$  gravedad, en  $m/s^2$ ;

$$M = \sum_m (F \times d) \quad (3)$$

Donde

$M$  momento, en Nm;

$F$  fuerza, en N;

$d$  distancia, en m;

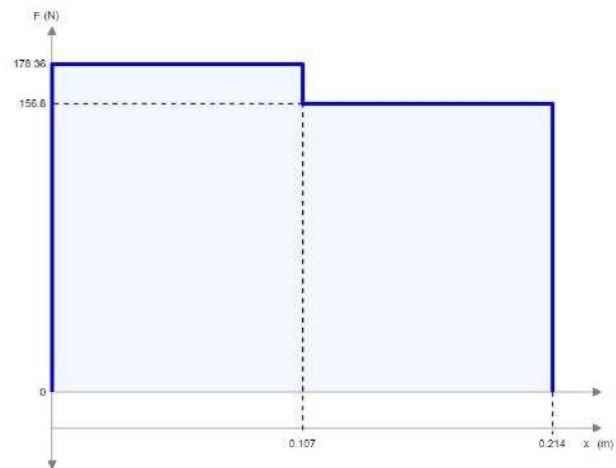
Se reemplaza en (2) para obtener  $F_1$  y  $F_2$  y se resuelve la sumatoria en (3).

$$F_1 = 2,2 \text{ kg} \times 9,8 \text{ m/s}^2$$

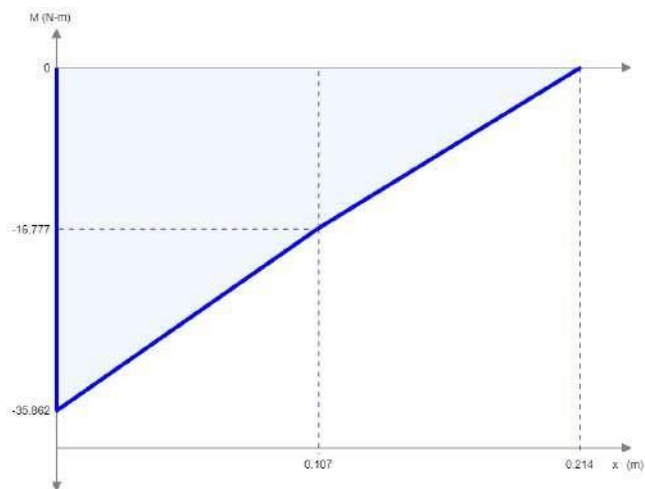
$$F2 = 16 \text{ kg} \times 9,8 \text{ m/s}^2$$

$$M = (2,2 \text{ kg} \times 9,8 \text{ m/s}^2)(0,214/2 \text{ m}) + (16 \text{ kg} \times 9,8 \text{ m/s}^2)(0,214 \text{ m})$$

$$M = 35,86 \text{ Nm}$$



**Figura 13.** Diagrama de cortante



**Figura 14.** Diagrama de momentos

Determinado el momento se procede a calcular el espesor mínimo necesario para la viga en voladizo por lo que se parte de (4) y de (5).

$$\sigma = \frac{M \times c}{I} \quad (4)$$

Donde

$\sigma$  esfuerzo, en Pa;

$M$  momento, en Nm;

- $c$  distancia al centro en este caso es la mitad del espesor, en m;  
 $I$  inercia, en  $m^4$ ;

$$\sigma = \frac{0,557Sut}{n} \quad (5)$$

Donde

- $\sigma$  esfuerzo, en Pa;  
 $Sut$  resistencia a la tracción, en Pa;  
 $n$  factor de seguridad;

Se iguala (4) y (5) tomando en consideración que la geometría de la viga es rectangular por lo que se ocupa (6) obteniendo (7) .

$$I_{rectangulo} = \frac{bh^3}{12} \quad (6)$$

Donde

- $I$  inercia, en  $m^4$ ;  
 $b$  longitud viga, en b;  
 $h$  altura en este caso es igual al espesor, en m;

Se despeja el espesor de (7) y se obtiene:

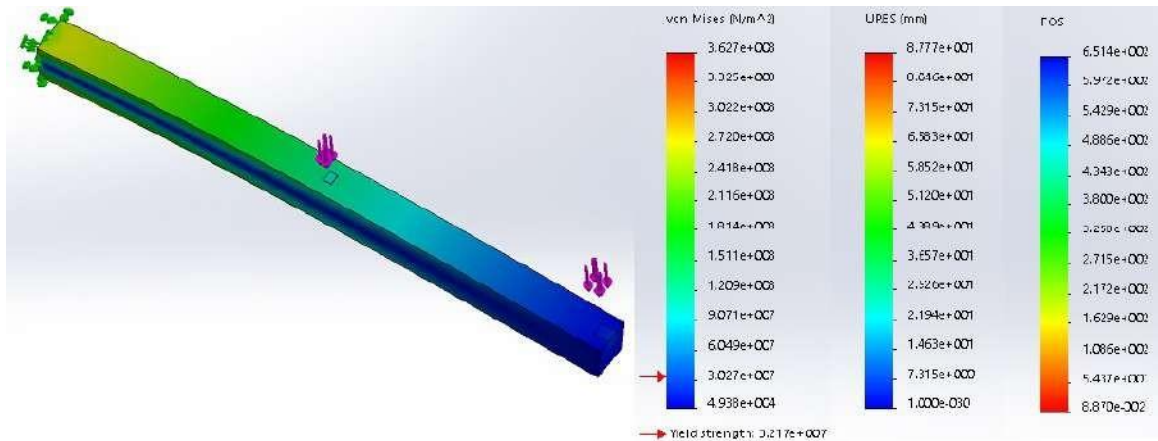
$$\frac{M(e/2)}{be^3/12} = \frac{0,577Sut}{n} \quad (7)$$

$$e = \frac{\sqrt[3]{\frac{6Mn}{0,577Sutb}}}{\sqrt[3]{\frac{6(35,86 \text{ Nm})(1)}{0,577(30,483 \times 10^6 \text{ Pa})(0,114 \text{ m})}}}$$

$$e = 10,359 \text{ mm}$$

## Simulacion

Dentro del Software SOLIDWORKS® se procedió a realizar la viga en voladizo con sus fuerza puntuales y se obtuvo lo que se puede visualizar en la Figura 15.



**Figura 15.** Simulación Cargas Puntuales Viga en voladizo

Comparando los valores analíticos con los valores obtenidos de la simulación (experimentales) son aceptables ya que el esfuerzo equivalente de Von Mises, factor de seguridad tiene menos de un 12 % de error y el desplazamiento se desprecia.

### 7.2.2. Cargas Distribuidas

El prototipo tiene una forma tipo cilíndrica curva y su comportamiento con las cargas es distribuida como se puede observar en la Figura 16 por lo que se procede a calcular mediante (8).

$$q = \frac{F1 + F2}{L} \quad (8)$$

Donde

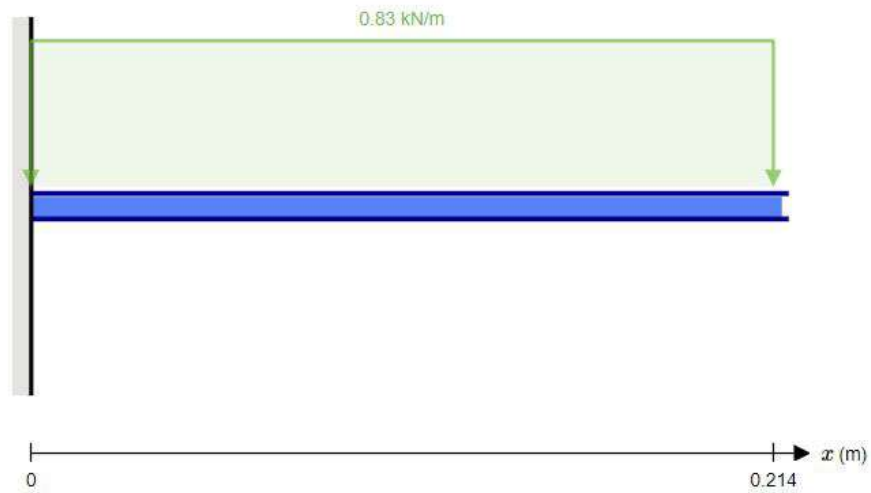
$q$  carga total, en N/m;

$F$  fuerza, en N;

$L$  longitud en m;

$$q = \frac{(2,2 \text{ kg} + 16 \text{ kg})(9,8 \text{ m/s}^2)}{0,214 \text{ m}}$$

$$q = 833,46 \text{ N/m}$$



**Figura 16.** Diagrama de cuerpo libre viga en voladizo carga distribuida

A partir de la carga distribuida obtenida se procede al cálculo de la fuerza puntual que representa el centro de la viga mediante (9).

$$F = q \times b \quad (9)$$

Donde

$F$  Fuerza, en N;

$q$  carga total, en N/m;

$b$  longitud de la viga, en m;

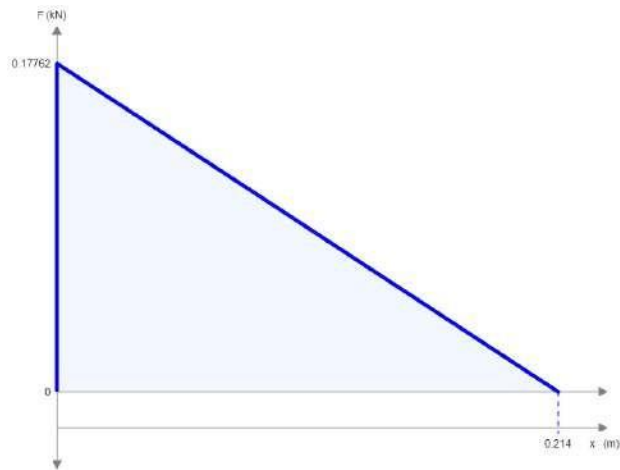
$$F = 833,46 \text{ N/m} \times (0,214 \text{ m})$$

$$F = 178,36 \text{ N}$$

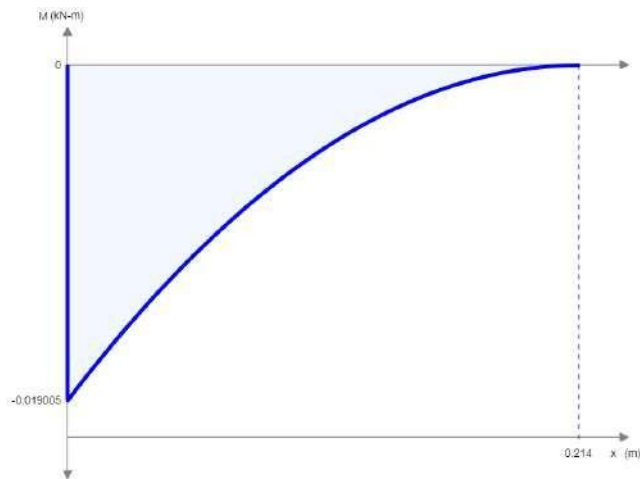
Obtenida la Fuerza puntual en el centro de la viga se procede a ocupar (3).

$$M = 178,36 \text{ N} \times 0,214/2 \text{ m}$$

$$M = 19,08 \text{ Nm}$$



**Figura 17.** Diagrama de cortante



**Figura 18.** Diagrama de momentos

Con el momento calculado se determina el espesor con (7) como se realizó anteriormente.

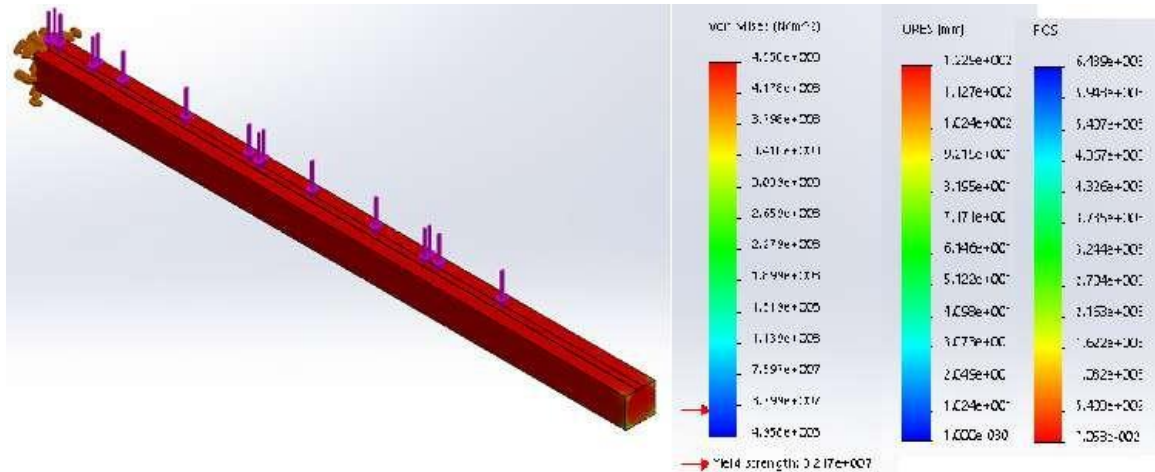
$$e = \frac{q}{0,577 \frac{6Mn}{Sutb}}$$

$$e = \frac{19,08 \text{ Nm}}{0,577(30,483 \times 10^6 \text{ Pa})(0,214 \text{ m})}$$

$$e = 5,515 \text{ mm}$$

## Simulación

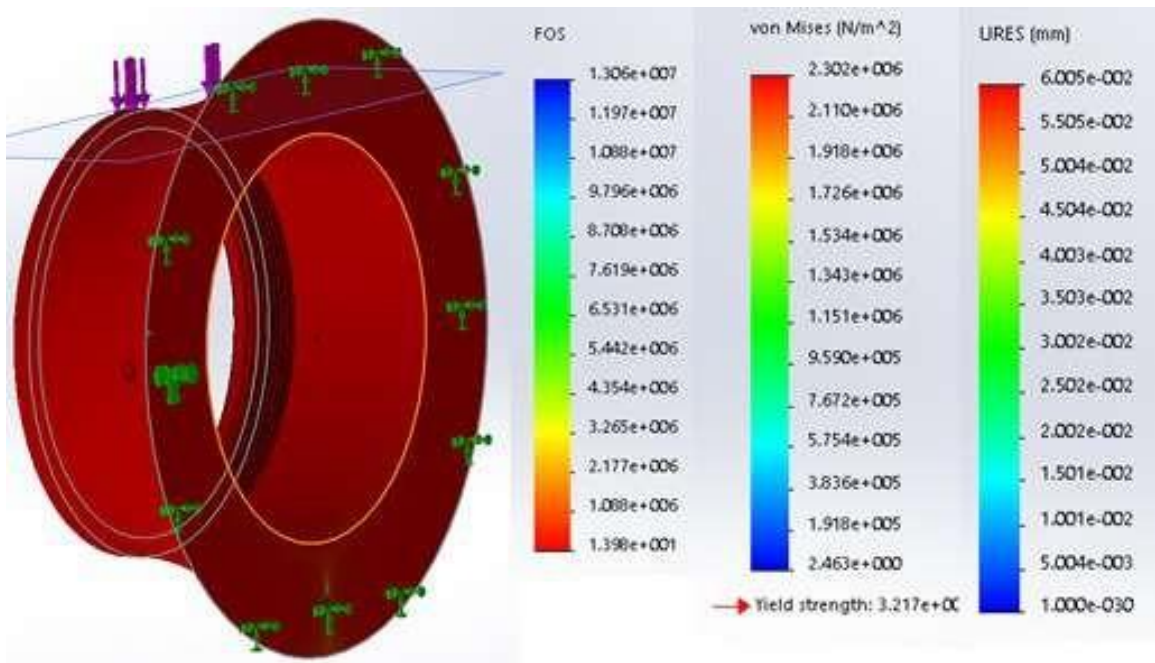
Dentro del Software SOLIDWORKS <sup>®</sup> se procedió a realizar la viga en voladizo con su fuerza distribuida y se obtuvo lo que se puede visualizar en la Figura 19.



**Figura 19.** Simulación Cargas Distribuidas Viga en voladizo

Comparando los valores analíticos con los valores obtenidos de la simulación (experimentales) son aceptables ya que el esfuerzo equivalente de Von Mises, factor de seguridad tiene menos de un 12 % de error y desplazamiento se desprecia.

Una vez comprobado los resultados en ambos casos se procede a simular en la geometría específica de la base externa como se puede observar en la Figura 20.



**Figura 20.** Simulación Base Externa

El valor obtenido en la simulación es bastante alto comparando los dos valores obtenidos analíticamente, ya que no se considera en su totalidad la geometría de la estructura sino se lo considera como una viga en voladizo y las cargas son puntuales o distribuidas. El valor

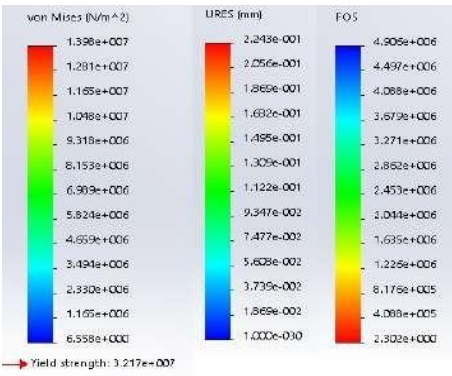


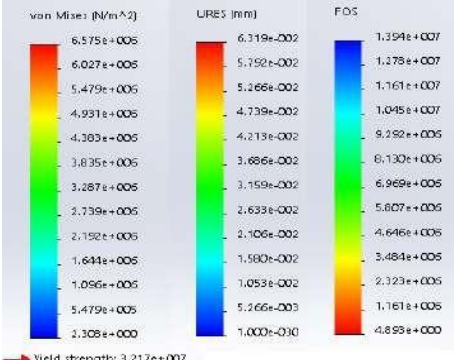
de desplazamiento es despreciable y el esfuerzo no supera la resistencia de tracción del prototipo. Por lo tanto el espesor obtenido por calculo solo es un punto de partida.

### 7.2.3. Selección Base Exterior

Dentro del Software SOLIDWORKS ® se procedió a realizar la simulación de las cargas sobre la base exterior que se puede visualizar en la Figura 20 para determinar el espesor global del prototipo. Se realiza mediante la variación del espesor considerando los valores calculados anteriormente como se puede observar en la Tabla 7.

**Tabla 7.** Observaciones Simulación base externa

Resultado	Observación
<p><b>e= 5 mm</b></p> 	<p>Con el espesor dado la base externa soporta las cargas es decir el esfuerzo de Von Mises no supera el del material, la deformación es despreciable y tiene un factor de seguridad de 2.3 el cual confirma que la geometría del prototipo se comporta diferente a una viga en voladizo, distribuye las cargas y su geometría soporta mas.</p>
<p><b>e= 7 mm</b></p> 	<p>Con el espesor ocupado se logra una mejora comparándola con la anterior pero no en una gran cantidad numérica.</p>

Resultado	Observación
<p><math>e = 10 \text{ mm}</math></p>  <p>Yield strength: 3.217e+007</p>	<p>Con el espesor ocupado se duplica la cantidad numérica en comparación a la primera.</p>

Realizadas las simulaciones se toma la decisión de seleccionar un espesor global de 5 mm.

### 7.3. Dimensionamiento Motor

Hay dos maneras de realizar el dimensionamiento de los motores, una tomando en cuenta la inercia y otra sin esta.

#### 7.3.1. Motor Eje

Para el primer cálculo se considera la inercia por lo que se ocupa un torque (10) y una inercia (11).

$$T = \gamma \times I \quad (10)$$

Donde

$T$  torque, en Nm;

$\gamma$  aceleración angular, en  $\text{rad/s}^2$ ;

$I$  inercia de rotación, en  $\text{kgm}^2$ ;

$$I = f \times m \times r^2 \quad (11)$$

Donde

$I$  inercia de rotación, en  $\text{kgm}^2$ ;

- $f$  coeficiente geométrico, en  $m/s^2$ ;  
 $m$  peso de la carga, en kg;  
 $r$  radio, en m;

Datos iniciales:

- Coeficiente geométrico  $f_{cilindro}$ :  $0.5 m/s^2$ .
- Peso  $m$ :  $141.2 \times 10^{-3} kg$ .
- Radio  $r$ :  $0.012 m$ .
- Aceleración angular de una cámara tipo domo  $\gamma_{domo}$ :  $36 \% - 32 \%$ . [39]

Para la cámara de tipo domo se ocupa un promedio de aceleración angular recomendada es decir de un  $34 \%$  que representa  $17\pi/90 rad/s^2$ . Con todos los valores se reemplaza (11) en (10) y se obtiene el torque requerido.

$$T = \left(\frac{17\pi}{90} rad/s^2\right)(0,5 m/s^2)(141,2 \times 10^{-3} kg)(0,012 m)^2$$

$$T = 6,03 uNm$$

Para el segundo cálculo no se considera inercia es decir se desprecia porque el motor no realiza un movimiento rotatorio grande por lo que se ocupa un torque (12).

$$T = F \times r \tag{12}$$

Donde

- $t$  torque, en Nm;  
 $F$  fuerza, en N;  
 $r$  radio, en m;

Reemplazando los valores iniciales y la ecuación (2) en (12) se obtiene el torque requerido.

$$T = (141,2 \times 10^{-3} kg)(9,8 m/s^2)(0,012 m)$$

$$T = 16,605 mNm$$

Para el torque real requerido se debe considerar la eficiencia del motor mediante (13) el cual en general es del 40 %.

$$T_r = \frac{T_t}{\eta} \quad (13)$$

Donde

$tT_r$  torque real, en Nm;

$T_t$  torque teorico, en N;

$\eta$  eficiencia del motor;

El torque real que se ocupa es el obtenido sin considerar inercia ya que es mayor.

$$T = \frac{16,605 \times 10^{-3} \text{ Nm}}{0,4}$$

$$T = 0,0415 \text{ Nm}$$

### 7.3.2. Motor Soporte Interno

Se realiza el mismo proceso que el anterior pero con los datos que corresponden a este elemento.

Datos iniciales:

- Coeficiente geométrico  $f_{\text{circulo}}$ : 0.4 m/s<sup>2</sup>.
- Peso  $m$ : 0.61 kg.
- Radio  $r$ : 0.093 m.
- Aceleración angular de una cámara tipo domo  $\gamma_{\text{domo}}$ :  $17\pi/90$  rad/s<sup>2</sup>.

Con todos los valores se reemplaza (11) en (10) y se obtiene el torque requerido considerando la inercia.

$$T = \left(\frac{17\pi}{90} \text{ rad/s}^2\right)(0,4 \text{ m/s}^2)(0,61 \text{ kg})(0,093 \text{ m})^2$$

$$T = 1,25 \text{ mNm}$$

Sin considerar inercia se reemplaza los valores iniciales y la ecuación (2) en (12) se obtiene el torque requerido.

$$T = (0,61 \text{ kg})(9,8 \text{ m/s}^2)(0,093 \text{ m})$$

$$T = 0,56 \text{ Nm}$$

Para el torque real requerido se debe considerar la eficiencia del motor mediante (13) el cual en general es de 40 %. El torque real que se ocupa es el obtenido sin considerar inercia ya que es mayor.

$$T = \frac{0,5 \text{ Nm}}{0,4}$$

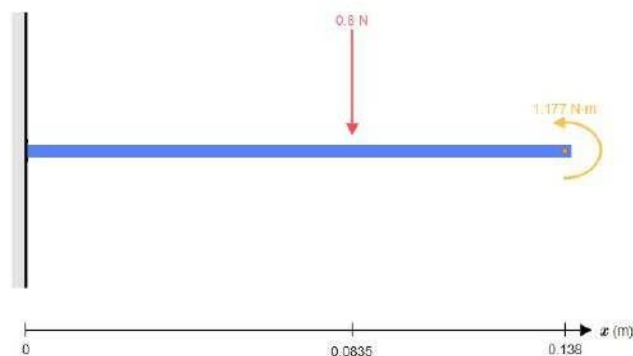
$$T = 1 \text{ Nm}$$

Para los siguientes cálculos se tomo como referencia el servomotor MG995 a rotación continua de 360 grados con un torque de 1.177 Nm [40].

#### 7.4. Dimensionamiento Eje

Para el dimensionamiento del eje, tomando en cuenta que el ABS es un material frágil, este se debe realizar un diseño estático y comprobarlo por medio de un ensayo destructivo de torsión para determinar el número de ciclos que soporta el eje.

El cálculo del eje estático se realiza como se observa en la Figura 21.



**Figura 21.** Diagrama de cuerpo libre Eje

Datos iniciales:

- Fuerza Carga  $F$ : 0.8 N.
- Torque  $Sut$ : 1.177 Nm.
- Diámetro  $d$ : 0.008 m.

Como se puede observar en la Figura 21 el eje se encuentra tanto en flexión por  $F$  como en torsión por  $T$ . Para poder determinar el factor de seguridad del eje con el diámetro dado, se procede a obtener el esfuerzo equivalente, pero como primer paso es necesario el momento que se calcula con (3).

$$M = (0,8 \text{ N})(0,0835 \text{ m})$$

$$M = 0,0668 \text{ Nm}$$

Obtenido el momento se calcula el esfuerzo normal causado por flexión mediante (14) con la inercia de un cilindro (15).

$$\sigma_A = \frac{M \times c}{I} \quad (14)$$

Donde

$\sigma_A$  esfuerzo normal, en Pa;

$M$  momento, en Nm;

$c$  radio, en m;

$I$  inercia, en m<sup>4</sup>;

$$I = \frac{\pi}{4} r^4 \quad (15)$$

Donde

$I$  inercia, en m<sup>4</sup>;

$r$  radio, en m;

Reemplazando (15) en (14) se obtiene un esfuerzo normal.

$$\sigma_A = \frac{(0,0668 \text{ Nm})(0,004 \text{ m})}{\pi/4(0,004 \text{ m})^4}$$

$$\sigma_A = 1,33 \text{ MPa}$$

A continuación se procede a calcular el esfuerzo cortante causado por torsión mediante (16) con el momento polar de inercia (17).

$$\tau_A = \frac{T \times c}{J} \quad (16)$$

Donde

- $\tau_A$  esfuerzo cortante, en Pa;
- $T$  torque, en Nm;
- $c$  radio, en m;
- $J$  momento polar de la inercia, en m<sup>4</sup>;

$$J = 2I \quad (17)$$

Donde

- $J$  momento polar de la inercia, en m<sup>4</sup>;
- $I$  inercia, en m<sup>4</sup>;

Reemplazando (15) en (17) y su resultante en (16) se obtiene un esfuerzo cortante.

$$\tau_A = \frac{(1,177 \text{ Nm})(0,004 \text{ m})}{\pi/2(0,004 \text{ m})^4}$$

$$\tau_A = 11,71 \text{ MPa}$$

Obtenido tanto el esfuerzo normal como el cortante se procede a aplicar Von Mises (18) y obtener el factor de seguridad con (5)

$$\sigma_T = \sqrt{\sigma_A^2 + 3\tau_A^2} \quad (18)$$

Donde

- $\sigma_T$  esfuerzo equivalente de Von Mises, en Pa;
- $\sigma_A$  esfuerzo normal, en Pa;
- $\tau_A$  esfuerzo cortante, en Pa;

$$\sigma_T = \sqrt{(1,33 \times 10^6 \text{ Pa})^2 + 3(11,71 \times 10^6 \text{ Pa})^2}$$

$$\sigma_T = 20,336 \text{ MPa}$$

$$n = \frac{0,577(32,171 \times 10^6 \text{ Pa})}{20,336 \text{ Pa}}$$

$$n = 0,91$$

Al obtener un factor de seguridad menor a 1 se concluye que el eje fallaría con un diámetro de 8 mm por lo que se procede a calcular el diámetro mínimo en el cual se tendría un factor de seguridad igual a 1. Para obtener diámetro mínimo se realiza en sentido inverso los cálculos anteriores.

Primero se obtiene el esfuerzo equivalente máximo en (5).

$$\sigma_T = \frac{0,577(32,17 \times 10^6 \text{ Pa})}{1}$$

$$\sigma_T = 18,56 \text{ MPa}$$

Para obtener el radio mínimo al esfuerzo normal (15) en y cortante en (16) se lo deja expresado en función del radio y se despeja el radio dentro de Von Mises en (18)

$$\sigma_A = \frac{0,0668 \text{ Nm}}{\pi/4r^3}$$

$$\tau_A = \frac{1,177 \text{ Nm}}{\pi/2r^4}$$

$$\sigma_T = \sqrt{\left(\frac{0,0668 \text{ Nm}}{\pi/4r^3}\right)^2 + 3\left(\frac{1,177 \text{ Nm}}{\pi/2r^4}\right)^2}$$

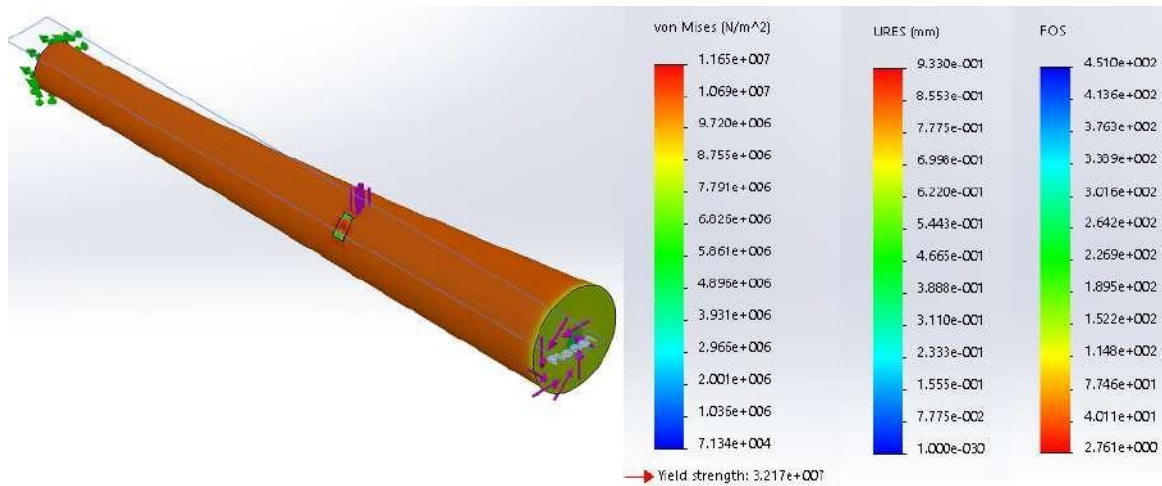
$$r = 4,13 \text{ mm}$$

Se obtuvo un radio mínimo de 4.13 mm por lo que se toma un radio de 5 mm es decir un diámetro de 10 mm y se procede a comprobar de la misma forma que se realizó de 8 mm obteniendo un esfuerzo equivalente de Von Mises de  $\sigma_T$  de 10.4 MPa y un factor de seguridad aceptable de 1.78.

## Simulación

Dentro del Software SOLIDWORKS ® se procedió a realizar el eje con su fuerza que causa flexión y torque que causa torsión y se obtuvo lo que se puede visualizar en la Figura 22.





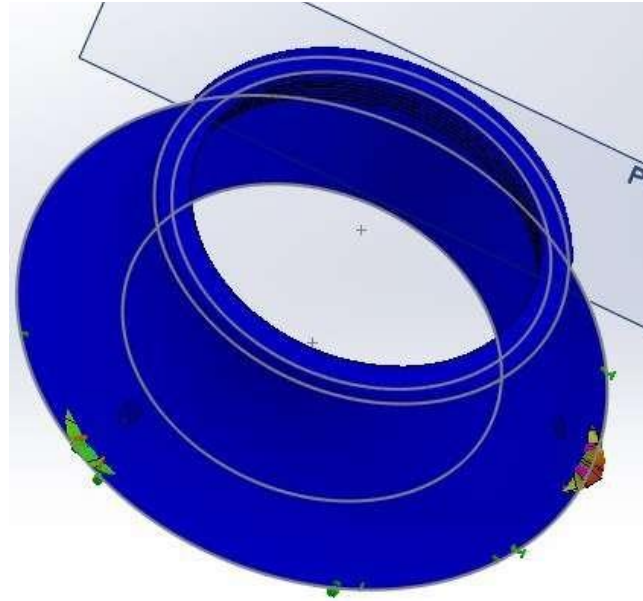
**Figura 22.** Simulación Flexión y Torsión Eje de 10mm

Comparando los valores analíticos con los valores obtenidos de la simulación (experimentales) son aceptables ya que el esfuerzo equivalente de Von Mises tiene un 12 % de error y el factor de seguridad es mayor a 1 y no tan lejano al teórico.

### 7.5. Torque de Sujeción Base Exterior

En el elemento de la base exterior existen tres perforaciones para la sujeción del prototipo en la superficie de apoyo que podría ser una pared o techo mediante pernos.

Dentro del Software SOLIDWORKS ® se procedió a realizar la simulación de torsión en cada perforación como se puede visualizar en la Figura 23 para determinar el torque máximo que se puede aplicar al perno. Se realiza mediante la variación del valor de torque. El proceso se puede visualizar en la Tabla 8.



**Figura 23.** Simulación Torsión aplicada en perforaciones de sujeción

**Tabla 8.** Observaciones Simulación Torsión en perforaciones de sujeción

Resultado	Observación
<p><i>Torque= 1 Nm</i></p>	<p>Con el torque dado la base externa soporta la torsión del perno.</p>
<p><i>Torque= 10 Nm</i></p>	<p>Con el torque dado la base externa falla por 50 MPa por lo que se debe reducir el torque.</p>

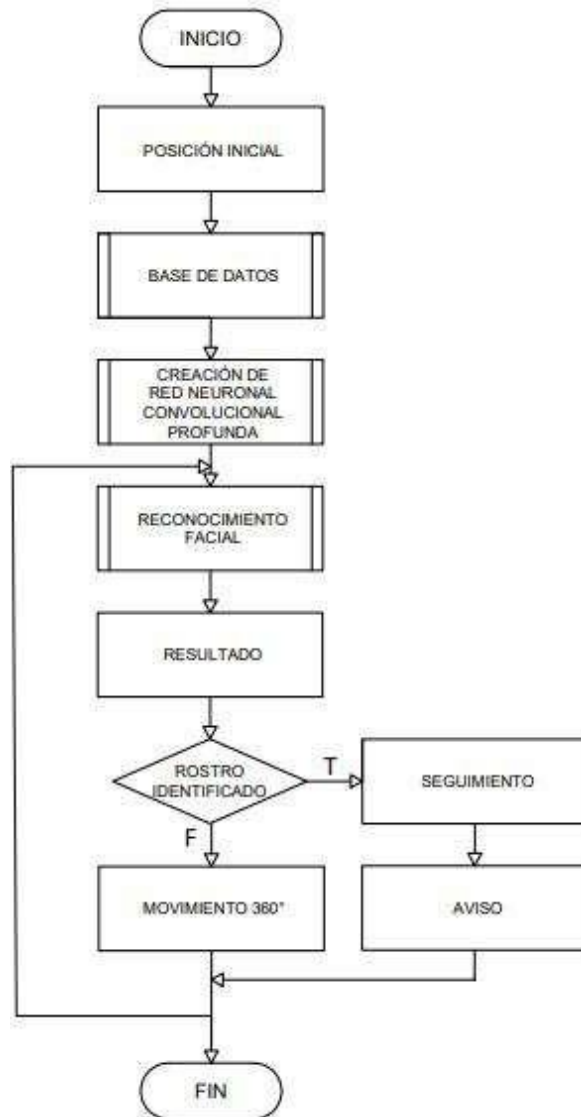
Resultado	Observación
<p><i>Torque= 3 Nm</i></p> 	<p>Con el torque dado la base externa soporta la torsión del perno.</p>
<p><i>Torque= 4 Nm</i></p> 	<p>Con el torque dado la base externa falla pero por la gráfica se concluye que el torque máximo es cercano a 4 Nm.</p>
<p><i>Torque= 3.91 Nm</i></p> 	<p>El torque máximo para las perforaciones de sujeción mediante pernos se encuentra entre 3.9 Nm y 3.91 Nm.</p>

Con el torque máximo obtenido se toma la decisión de seleccionar el perno milimétrico M5 de grado 4.6 el cual soporta un torque de ajuste de 2 Nm así se garantiza que el perno falla antes que el prototipo. [41]

## 8. Algoritmo general de programación

Cuando el prototipo se conecta a la computadora y se ejecuta el algoritmo de Deep Learning el sistema procede con su movimiento tanto horizontal como vertical y en el momento

que identifica un rostro de su interés el prototipo le sigue y envía un aviso de alerta. El algoritmo en su totalidad se encuentra en el Anexo C. La lógica de funcionamiento general se observa en la Figura 24 las subrutinas se encuentran en el Anexo D.



**Figura 24.** Algoritmo general

Para el desarrollo del algoritmo se utiliza el software Python<sup>TM</sup> en la versión 2.7 ya que en esta se encuentran la mayoría de librerías a diferencia de sus versiones superiores por lo que se ocupa en proyectos de investigación a nivel mundial. En esta versión se encuentra todo lo necesario para la ejecución de las diferentes funciones en conjunto con OpenCV para el reconocimiento facial.

El algoritmo se compone de tres partes necesarias para su adecuado funcionamiento. La primera parte es la base de datos la cual puede ser pre-hecha o puede ser creada

con imágenes de rostros que se desea identificar, la segunda parte consta de la creación de una red neuronal convolucional a partir de la base de datos y la tercera parte es el reconocimiento facial en tiempo real, de vídeos o de imágenes.

## **8.1. Base de Datos**

Existen dos opciones para obtener una base de datos, la primera se basa en la descarga de imágenes de google y la segunda en la obtención en tiempo real de imágenes mediante una cámara.

### **8.1.1. Base de datos de Google**

Una manera rápida y eficiente de obtener una base de datos es mediante la descarga de imágenes de rostros de la gigantesca memoria de google la cual asemeja un banco de imágenes pre-hecho.

Tomando en referencia el algoritmo de control que se puede visualizar en el Anexo C.1, el programa funciona mediante cuatro librerías, la primera es "requests" la cual contiene los paquetes http y todas las funciones necesarias para la descarga de la imagen, la segunda es "cv2", la tercera es "os" y por último la librería "time", las tres últimas se van a ocupar a lo largo de todo el proyecto ya que permite el uso del OpenCV, la creación de directorios y tomar el tiempo de ejecución respectivamente.

A continuación se encuentra los parámetros del programa y el inicio del cronómetro, entre los parámetros se encuentra el factor esencial el cual es la API key que es un identificador que sirve como el medio de autenticación de un usuario para el uso de los servicios proporcionados por google, en este caso para la descarga de imágenes. En esta sección se ubica el término a ser buscado, el número de búsquedas y el número de grupos de resultados.

Después se realiza la creación de directorios que es otro factor a tener en consideración ya que la base de datos debe estar en la misma carpeta que todos los programas y cada usuario (sus imágenes faciales) tiene su respectivo directorio dentro de este con su nombre.

Es muy importante considerar los posibles errores que se generan al momento de la búsqueda de imágenes por lo que se implementa excepciones HTTP de manera manual.

Una vez realizado lo anterior se procede con la búsqueda y descarga por lo que se implementa dos bucles, uno dentro del otro, el primero se encarga de la localización de la imagen obteniendo el URL y los parámetros de esta, el segundo bucle solicita la imagen para guardarlo en la base de datos, en este bucle se implementan excepciones por posibles errores en el formato de la imagen.

A lo largo de la ejecución del programa se podrá observar mensajes de verificación de cada imagen descargada o si falla y una vez finalizado se puede visualizar el tiempo de ejecución.

### **8.1.2. Base de datos Local**

Esta base de datos se puede obtener de dos maneras y mediante pruebas se determina cuál es la ideal para la obtención de rostros mediante una cámara en tiempo real.

La primera consiste en la realización de un tratamiento de imagen y un reconocimiento facial mediante el clasificador " Cascade" el que es un método tradicional de detección facial de python. Este clasificador se encuentra en las páginas oficiales de Python<sup>TM</sup> y su directorio se debe localizar en la misma carpeta que todos los demás programas.

Tomando en referencia el algoritmo de control que se puede visualizar en el Anexo C.2, el programa funciona mediante tres librerías: " cv2" , " os" y " time" las cuales son ocupadas por el programa anterior.

De igual forma este programa comienza con los parámetros y da inicio al cronómetro. Como primer parámetro mediante la función " VideoCapture" de OpenCV se selecciona la cámara con la que se va obtener las imágenes de los rostros deseados, a continuación de este se llama al clasificador " Cascade" .

Después se realiza la creación de directorios y se procede a la implementación de un bucle el cual depende de un contador que determina el número de imágenes de la base de datos. En este bucle se abre la cámara, se realiza el tratamiento donde se transforma la imagen a RGB, se detecta el rostro mediante el clasificador, se recorta la imagen enfocando donde se detecta el rostro y se almacena la imagen como se puede ver en la Figura 25.



**Figura 25.** Imagen almacenada con clasificador "Cascade" en la base de datos

La parte final consiste en la implementación de la condición para liberar la cámara y cerrar el programa en el caso de error de conexión. Una vez finalizado se puede visualizar el tiempo de ejecución.

La segunda opción a diferencia de la anterior no tiene un tratamiento de imagen ni una detección facial inicial.

Tomando como referencia el algoritmo de control que se puede visualizar en el Anexo C.3, el programa funciona con las mismas librerías y el primer parámetro del anterior. Al no tener un clasificador se debe colocar las especificaciones deseadas para la imagen, considerando las características de la cámara a ser ocupada, estas especificaciones son: ancho, alto, fotogramas por segundo (fps) y exposición. Para la exposición ideal del programa se realizan pruebas que son expuestas en la siguiente sección del documento.

Después de los parámetros el programa crea directorios, implementa un bucle y la parte final es de la misma forma que la anterior base de datos local, pero con la diferencia que en este simplemente se almacena la imagen sin ningún tratamiento de imagen o detección facial inicial como se puede ver en la Figura 26.



**Figura 26.** Imagen almacenada en la base de datos

## 8.2. Red Neuronal Convolutacional

La creación de la red neuronal convolutacional se da mediante una serie de procesos, primero se realiza una convolución y un pooling que permite la obtención de las características faciales, luego se realiza una identificación facial y una codificación con una red neuronal generando un vector de 128d categorizando cada persona dentro de la base de datos. La red neuronal fue entrenada con 3 millones de imágenes de rostros con una precisión de 99.38 %.

Tomando en referencia el algoritmo de control que se puede visualizar en el Anexo C.4, el programa funciona con seis librerías, la primera librería es "imutils" la cual permite operaciones de procesamiento de imágenes, la segunda es "face\_recognition" que tiene la red neuronal entrenada para la realización de la identificación y codificación, la tercera es "pickle", este módulo realiza un proceso de serialización es decir que convierte un objeto arbitrario en una serie de bytes, la red neuronal convolutacional se codifica en este formato, las últimas tres librerías son las mismas ocupadas anteriormente ("cv2", "os" y "time").

A continuación de las librerías se procede a colocar los parámetros necesarios, el primero se trata del método a ocupar, el segundo es el nombre de la base de datos para la obtención de su dirección (Path) y el tercero es el nombre que se desea de la red neuronal sin modificar su extensión (.pickle). La librería "face\_recognition" se puede ocupar median-



te dos métodos, histograma de gradientes orientados (HOG) o red neuronal convolucional (CNN), para este proyecto se ocupa el segundo.

Después de los parámetros se procede con la creación de dos listas, en la primera se almacena el nombre del directorio de la base de datos que es el nombre de la persona a quien pertenece el rostro y en la segunda se almacena la codificación correspondiente del nombre. Estas listas se llenan dentro de un bucle el cual es limitado según el número de imágenes dentro de la base de datos, dentro de esta se abre cada imagen de la base de datos, se realiza un tratamiento el cual consiste en transformar a RGB y en cambiar el ancho de la imagen a 1280 píxeles (calidad HD 720p) para la distinción de las características faciales y eficiencia del programa. Por último se obtiene las coordenadas del rostro y se codifica en un vector 128d que representa de manera numérica las características más importantes del rostro.

Finalizado el bucle se procede a almacenar la red neuronal con su extensión .pickle, se cierra el programa y permite visualizar el tiempo de ejecución del programa como los programas anteriores.

### **8.3. Reconocimiento Facial**

El reconocimiento facial funciona realizando una comparación de la codificación de la red neuronal convolucional con la imagen obtenida de la cámara en tiempo real, imagen o vídeo dando así un porcentaje de coincidencia.

#### **8.3.1. Imágenes**

Tomando en referencia el algoritmo de control que se puede visualizar en el Anexo C.5, el programa funciona con cinco librerías: " face\_recognition" , " pickle" , " cv2" , " time" e " imutils" las cuales han sido mencionadas anteriormente.

De la misma forma que los programas anteriores después de las librerías se colocan los parámetros para el funcionamiento de este y el inicio del cronómetro. Entre los parámetros se coloca el método de red neuronal convolucional (CNN), el nombre de la imagen que se desee analizar con su respectiva extensión, el número de imágenes dentro de la base de datos y el nombre de la red neuronal creada con su extensión .pickle.

Con todos los parámetros colocados se procede a la apertura de la imagen que se desea identificar: los rostros y la obtención de la codificación de la red neuronal. A continuación se crean dos listas, una se va a encargar del almacenamiento de los nombres y la segunda se encarga del almacenamiento del porcentaje de coincidencia. Estas listas se llenan dentro de un bucle donde se realiza una comparación entre la codificación de la red neuronal con la imagen seleccionada mediante índices, estos permiten contar coincidencias y obtener el nombre de las personas a quien corresponde el rostro. El porcentaje se calcula tomando el número de coincidencias dividido para el tamaño de la base de datos multiplicado por 100.

Una vez finalizado el bucle, es decir realizada la comparación de la imagen con toda la red neuronal convolucional, se visualiza y almacena una imagen con los resultados obtenidos como se puede ver en la Figura 27, para esto se realiza un bucle donde se coloca un rectángulo en las coordenadas donde se localiza un rostro y sobre este se puede visualizar su respectivo nombre y porcentaje de coincidencia. De igual forma que los anteriores programas se puede visualizar en el programa el tiempo de ejecución de la misma.

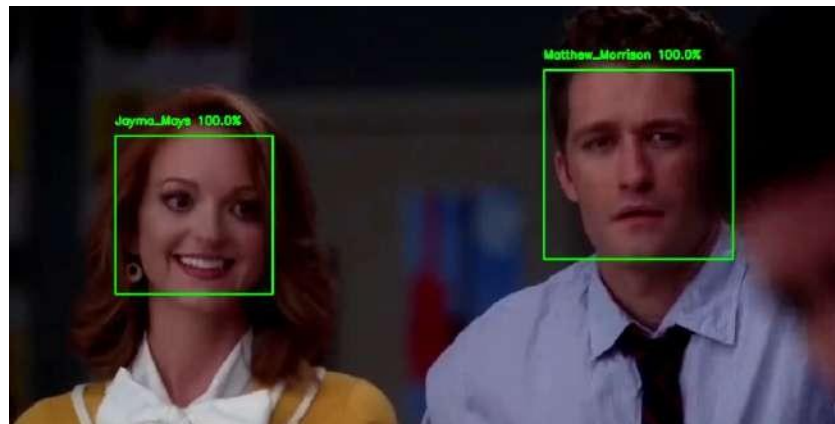


**Figura 27.** Imagen resultado almacenado

### 8.3.2. Videos

Tomando en referencia el algoritmo de control que se puede visualizar en el Anexo C.6, se puede visualizar que el programa tiene la misma lógica que el reconocimiento facial de imágenes y esto tiene sentido ya que un vídeo es una secuencia de imágenes. La diferencia

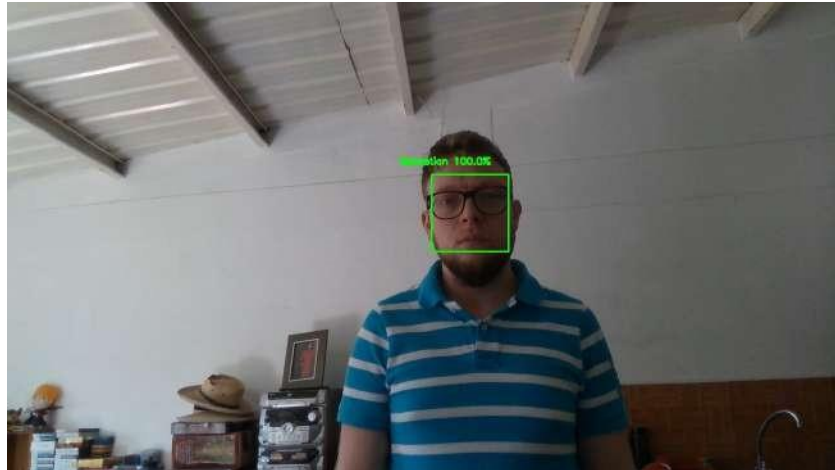
respecto al anterior programa esta en los parámetros, donde se debe poner el nombre del vídeo con su respectiva extensión no una imagen, se necesita también colocar un nombre con el que se desee almacenar el resultado y por último es diferente la manera de almacenar el resultado ya que necesita la función " VideoWriter" en imagen es la función " imwrite" . Se puede observar el resultado en la Figura 28 la cual es obtenida de un momento del vídeo resultado.



**Figura 28.** Video resultado almacenado

### 8.3.3. Tiempo Real

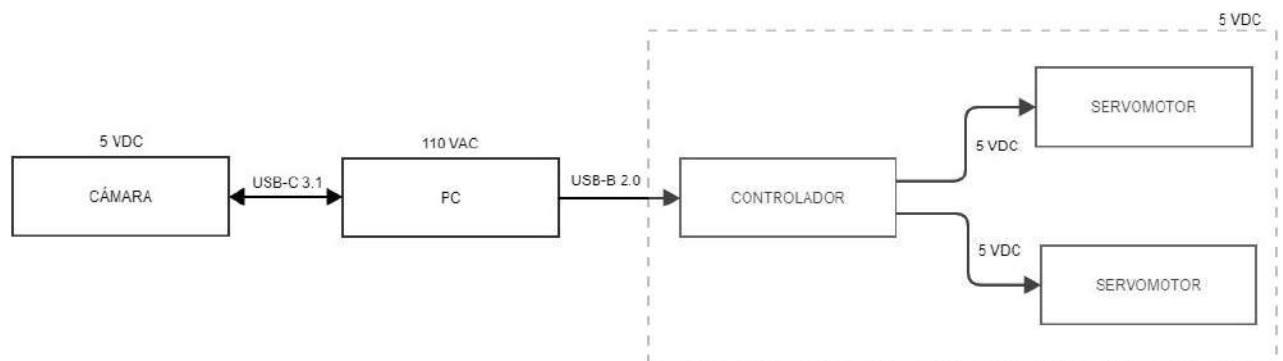
Tomando en referencia el algoritmo de control que se puede visualizar en el Anexo C.7, de la misma forma que el anterior comparten la misma lógica ya que este programa consiste en la obtención de vídeo en tiempo real mediante una cámara, en el caso del proyecto la cámara es la de profundidad. La diferencia respecto a la anterior es que no se necesita llamar a ningún archivo sea imagen o vídeo sino se debe abrir la cámara y obtener su información para su respectivo reconocimiento facial. Se puede observar el resultado en la Figura 29 la cual es obtenida de un momento del vídeo resultado.



**Figura 29.** Video resultado almacenado

## 9. Diseño Electrónico

Para el dimensionamiento electrónico se realiza un diagrama de bloques donde se especifican todos los elementos que posee el prototipo visualizado en la Figura 30.



**Figura 30.** Diagrama de Bloques del prototipo

El sistema consta de dos salidas digitales (controlador - servomotores) y una conexión USB.

### 9.1. Características del Ordenador

Las funciones que ejecuta la computadora son:

- Alimentación cámara.
- Alimentación microcontrolador - actuadores.
- Algoritmo para la creación de base de datos.

- Algoritmo de entrenamiento de red neuronal convolucional.
- Algoritmo de reconocimiento facial en tiempo real.

El proyecto se desarrollo y ejecutó en una computadora portátil y en una computadora de escritorio con diferentes especificaciones.

Las especificaciones de la computadora portátil son:

- Procesador: Intel ®Duo Core I3 3110 2.40 GHz.
- RAM: 16 Gb 1600 MHz.
- Tarjeta de video: Intel ®HD Graphics 4000.

Las especificaciones de la computadora de escritorio son:

- Procesador: Intel ®Quad Core I5 4440 3.10 GHz.
- RAM: 16 Gb 2400 MHz, 8 Gb 1600 MHz.
- Tarjeta de video: NVIDIA ®GeForce GTX 1650 4 Gb de memoria.

La diferencia entre las dos computadoras fueron el procesador, la memoria y la tarjeta gráfica donde la computadora de escritorio es de mayor gama en comparación a la portátil.


La tarjeta de vídeo es de suma importancia para el desarrollo de Deep Learning ya que el tiempo para el entrenamiento de la red neuronal convolucional es mucho menor, específicamente en la computadora de escritorio el entrenamiento de la red para una imagen en la base de datos es de 8.67 segundos a diferencia de la computadora portátil que se demora 290.83 segundos. Al momento de realizar una red con una base de datos de un gran tamaño, es decir con un número significativo de imágenes, existe la necesidad de tener una tarjeta gráfica y en el caso de ser de la marca NVIDIA ®se debe verificar que tengan compatibilidad para aplicaciones CUDA el cual permite acelerar procesos. En la página web de NVIDIA developer se puede encontrar el listado.

## 9.2. Actuadores

En el mercado se puede encontrar diferentes motores que tienen diferentes características mostradas en la Tabla 9.

**Tabla 9.** Modelos de servomotores de rotación continua

Modelo	Características	Fotografía
MG995	<p>Torque: 12 kg cm 1.177 Nm.</p> <p>Rotación: 360 grados.</p> <p>Velocidad de operación: 0.19 s / 60 grados.</p> <p>Voltaje de operación: 4.8 - 6.6 V.</p> <p>Tipo de engranaje: Metálico.</p> <p>Dimensiones: 54 x 20 x 47.2 mm.</p> <p>Corriente de operación: 100 mA.</p>	
43R SM-4315R	<p>Torque: 15.4 kg cm 1.51 Nm.</p> <p>Rotación: 360 grados.</p> <p>Velocidad de operación: 0.31 s / 60 grados.</p> <p>Voltaje de operación: 4.8 - 6 V.</p> <p>Tipo de engranaje: Metálico.</p> <p>Dimensiones: 54 x 20.5 x 47.2 mm.</p> <p>Corriente de operación: 150 mA.</p>	
DS04-NFC	<p>Torque: 5.5 kg cm 0.54 Nm.</p> <p>Rotación: 360 grados.</p> <p>Velocidad de operación: 0.22 s / 60 grados.</p> <p>Voltaje de operación: 4.8 - 6 V.</p> <p>Tipo de engranaje: Plástico.</p> <p>Dimensiones: 54 x 20 x 44 mm.</p> <p>Corriente de operación: 100 mA.</p>	

Modelo	Características	Fotografía
Futaba S3003	Torque: 4.1 kg cm 0.4 Nm. Rotación: 360 grados. Velocidad de operación: 0.19 s / 60 grados. Voltaje de operación: 4.8 - 6 V. Tipo de engranaje: Plástico. Dimensiones: 40 x 20.1 x 36.1 mm. Corriente de operación: 100 mA.	

Tomando en cuenta la matriz de decisión de Pugh que se puede visualizar con mayor detalle en el Anexo B.5, se toma la decisión de optar por el servomotor MG995 ya que sus características satisfacen todas las necesidades.


### 9.3. Controlador

Para la selección del controlador se toma en consideración las entradas y salidas tanto digitales como analógicas del sistema, en este caso se tienen dos salidas digitales las cuales deben enviar la señal a los actuadores.

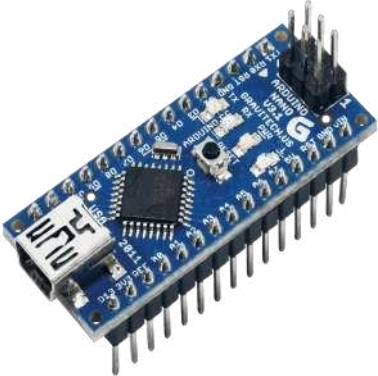
Los actuadores son dos servomotores MG995 de rotación continua los cuales tienen un consumo de 100 mA cada uno.


En el mercado existen sin número de microcontroladores con diferentes características los cuales son descritas en la Tabla 10


**Tabla 10.** Modelos de microcontroladores

Modelo	Características	Fotografía
<p>Arduino UNO</p>	<p>Microcontrolador: ATmega328P.            Voltaje de Operación: 5V.            Voltaje de entrada (recomendada): 7-12V.            Voltaje de entrada (Limite): 6-20V.            Pines digitales: 14 (6 PWM).            Pines analógicos: 6.            Corriente continua por pin IO: 40 mA.            Corriente continua en el pin 3.3V: 50 mA.            Corriente salida (Alimentación USB): 500 mA.            Memoria Flash: 32 KB (0.5 KB bootloader).            SRAM: 2 KB.            EEPROM: 1 KB.            Frecuencia reloj: 16 MHz.            Dimensiones: 6.8 x 5.3 cm.</p>	



Modelo	Características	Fotografía
<p>Arduino Nano</p>	<p>Microcontrolador: ATmega328.            Voltaje de Operación: 5V.            Voltaje de entrada (recomendada): 7-12V.            Voltaje de entrada (Limite): 6-20V.            Pines digitales: 14 (6 PWM).            Pines analógicos: 8.            Corriente continua por pin IO: 40 mA.            Corriente continua en el pin 3.3V: 50 mA.            Corriente salida (Alimentación USB): 200 mA.            Memoria Flash: 32 KB (2 KB bootloa-der).            SRAM: 2 KB.            EEPROM: 1 KB.            Frecuencia reloj: 16 MHz.            Dimensiones: 4.2 x 1.85 cm.</p>	

Modelo	Características	Fotografía
chipKIT Uno32	<p>Microcontrolador: PIC32MX795F512L.</p> <p>Voltaje de Operación: 3.3 V.</p> <p>Voltaje de entrada (recomendada): 7-15V.</p> <p>Voltaje de entrada (Limite): 20V.</p> <p>Pines digitales: 83.</p> <p>Pines analógicos: 16.</p> <p>Corriente continua por pin IO: 18 mA.</p> <p>Corriente salida (Alimentación USB): 90 mA.</p> <p>Memoria Flash: 512 Kb.</p> <p>Frecuencia reloj: 80 MHz.</p> <p>10/100 Ethernet MAC.</p> <p>Dimensiones: 11.176 x 6.096 cm.</p>	

Modelo	Características	Fotografía
NodeMCU ESP8266	Microcontrolador: Tensilica 32-bit RISC CPU Xtensa LX106. Voltaje de Operación: 3.3V. Voltaje de entrada (recomendada): 7-12V. Pines digitales: 16. Pines analógicos: 1. UARTs: 1. SPIs: 1. I2Cs: 1. Memoria Flash: 4 MB. SRAM: 64 KB. Frecuencia reloj: 80 MHz. USB-TTL. PCB Antenna. Dimensiones: 4.6 x 2.6 cm.	

Tomando en cuenta la matriz de decisión de Pugh que se puede visualizar con mayor detalle en el Anexo B.6, se toma la decisión de implementar el microcontrolador ATmega328P ya que es el que mejor satisface las necesidades del proyecto.

Las funciones que ejecuta el microcontrolador son:

- Movimiento horizontal.
- Movimiento vertical.
- Seguimiento de objeto.
- Alerta.

## 10. Pruebas y Resultados

Se realizaron catorce pruebas para poder obtener un sistema funcional y eficiente capaz de realizar el reconocimiento facial basado en Deep Learning para seguridad en zonas internas con una cámara de profundidad y su respectivo movimiento en tiempo real.

Para las diferentes pruebas se realizaron tres bases de datos: La primera se realizó mediante la descarga de imágenes de Google de la serie Glee ya que en esta se encuentra personajes con diferentes características faciales y también cabe destacar que los capítulos de la serie se pueden encontrar en una alta calidad de vídeo, la base de datos se compone de 16 personajes los cuales salen a lo largo de la segunda temporada (Amber Riley, Chris Colfer, Cory Monteith, Dianna Agron, Heather Morris, Idina Menzel, Jane Lynch, Jayma Mays, Jenna Ushkowitz, Jessalyn Gilsig, Kevin McHale, Lea Michele, Mark Salling, Matthew Morrison, Mike O'Malley y Naya Rivera). La segunda base de datos es realizada mediante la cámara de profundidad seleccionada y contiene imágenes de personas conocidas. La tercera base de datos se ocupó para la penúltima prueba con una cámara normal de 20 Megapíxeles.

### 10.1. Prueba 1: Determinación del número de imágenes necesarias para la base de datos de Google

La primera prueba consta del análisis del porcentaje de acierto, considerando el número de imágenes por personaje dentro de la base de datos, dentro de 5 imágenes de la serie Glee. Se realizó en un rango de 1 a 40 imágenes por cada personaje dando como resultado que el número óptimo para que exista un porcentaje de acierto mayor al 90 % es de 21 imágenes por personaje es decir 336 imágenes en su totalidad como se puede observar en la Tabla 11 y su representación gráfica en la Figura 31. En la Figura 32 se puede observar el resultado de 1 fotografía con la base de datos óptima.

**Tabla 11.** Prueba 1 Base de datos google

Número de imagen	Porcentaje de acierto
Imagen 1	90.476
Imagen 2	90.476

Número de imagen	Porcentaje de acierto
Imagen 3	85.714
Imagen 4	100
Imagen 5	100

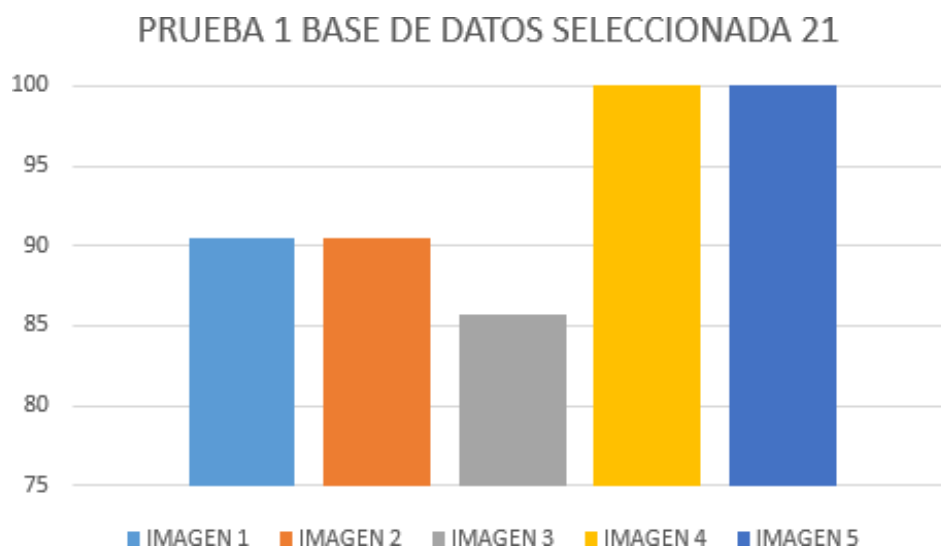


Figura 31. Representación Gráfica Prueba 1

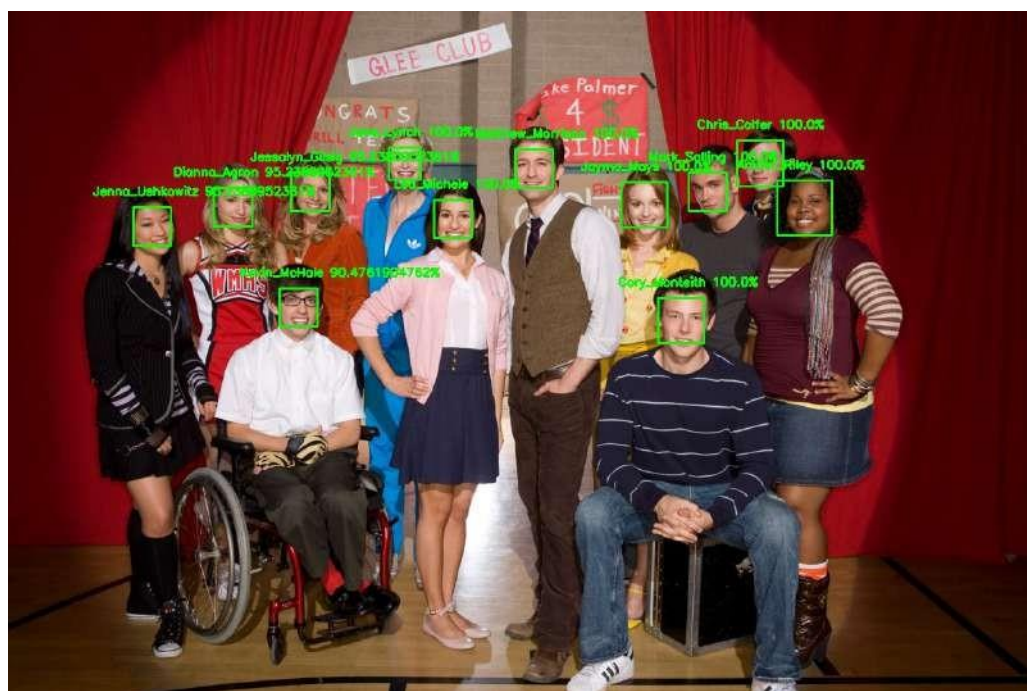


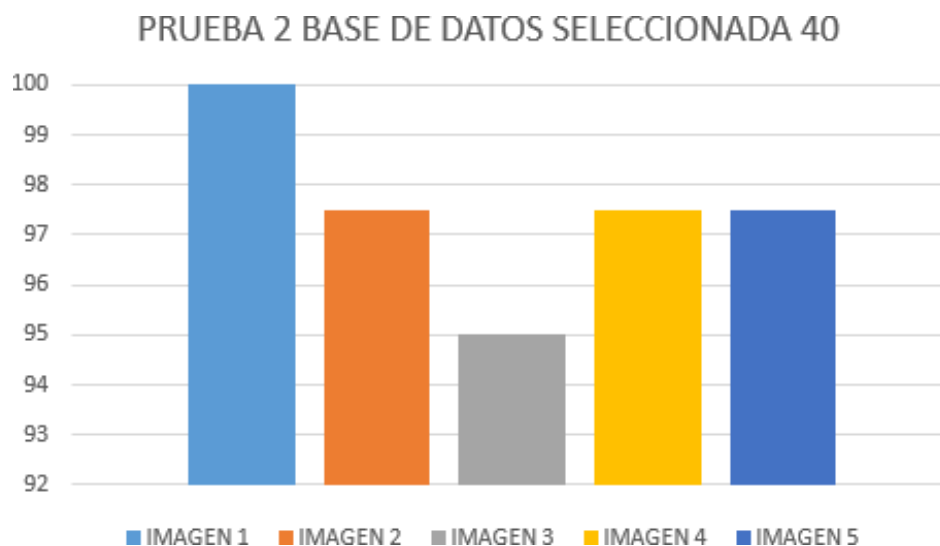
Figura 32. Imagen resultado almacenado Prueba 1

## 10.2. Prueba 2: Determinación del número de imágenes necesarias para la base de datos obtenidas localmente (cámara de profundidad)

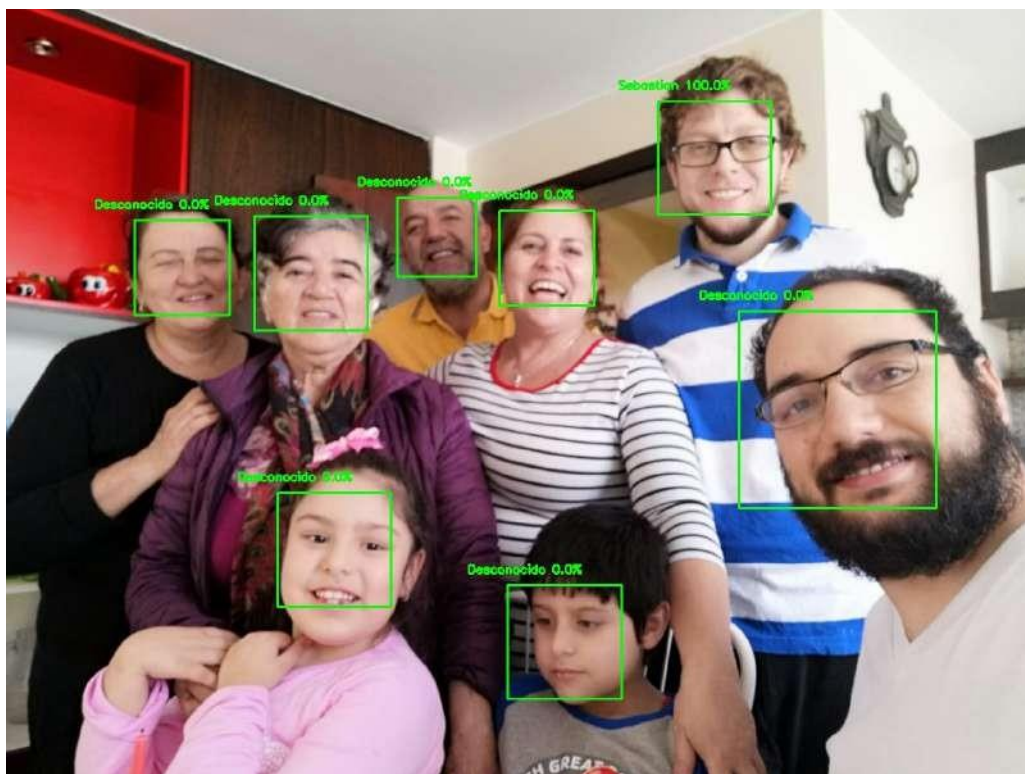
La segunda prueba consta del análisis del porcentaje de acierto considerando el número de imágenes de la base de datos local. Se analizó en un rango de 1 a 100 imágenes, dando como resultado que el número óptimo para que exista un porcentaje de acierto mayor al 90 % es de 40 como se puede observar en la Tabla 12 y su representación gráfica en la Figura 33. En la Figura 34 se puede observar el resultado de 1 fotografía con la base de datos óptima.

**Tabla 12.** Prueba 2 Base de datos local

Número de imagen	Porcentaje de acierto
Imagen 1	100
Imagen 2	97.5
Imagen 3	95
Imagen 4	97.5
Imagen 5	97.5



**Figura 33.** Representación Gráfica Prueba 2



**Figura 34.** Imagen resultado almacenado Prueba 2

### 10.3. Prueba 3: Determinación de funcionalidad clasificador "Cascade" e Interferencia de accesorios del usuario

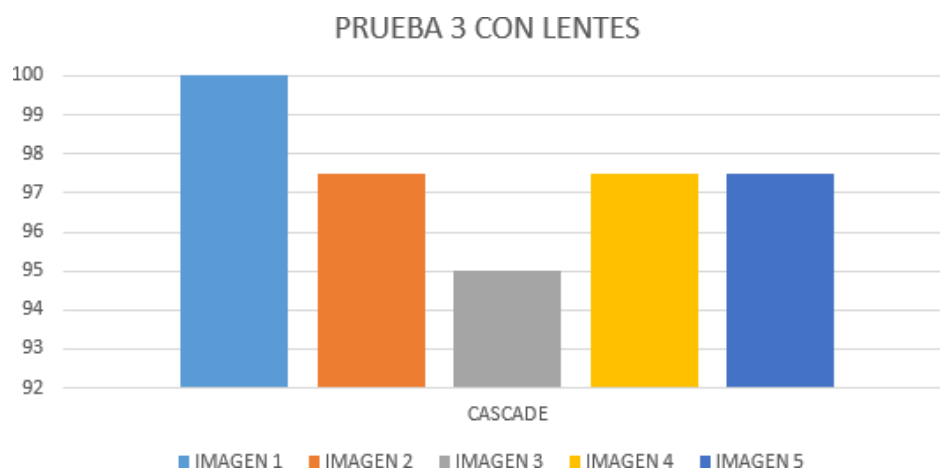
La tercera prueba es la comparación en la creación de la base de datos con y sin el uso del clasificador "Cascade" y accesorios. Se realizó esta prueba para decidir si es necesario el clasificador "Cascade" y si el reconocimiento facial es afectado por el uso de accesorios, en el caso de la prueba esta se realizó con el accesorio de los lentes.

En la Tabla 13 y su representación gráfica en las Figuras 35 y 36 se puede observar los resultados obtenidos con el uso del clasificador "Cascade" con y sin accesorios. Se puede verificar las imágenes dentro de la base de datos en las Figuras 37 y 38 respectivamente y su resultado en las Figuras 39 y 40.

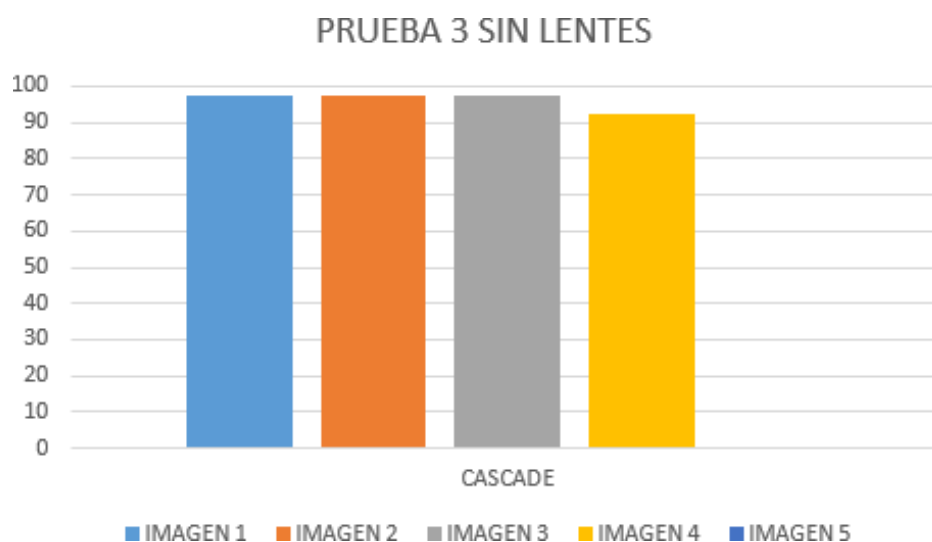
**Tabla 13.** Prueba 3 Base de datos local con clasificador "Cascade"

Número de imagen	Porcentaje de acierto con lentes	Porcentaje de acierto sin lentes
Imagen 1	100	97.5
Imagen 2	97.5	97.5

Número de imagen	Porcentaje de acierto con lentes	Porcentaje de acierto sin lentes
Imagen 3	95	97.5
Imagen 4	97.5	92.5
Imagen 5	97.5	0

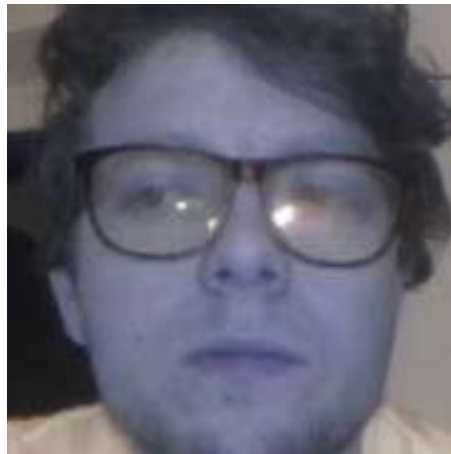


**Figura 35.** Representación Gráfica Prueba 3 con "Cascade" y con accesorio



**Figura 36.** Representación Gráfica Prueba 3 con "Cascade" y sin accesorio

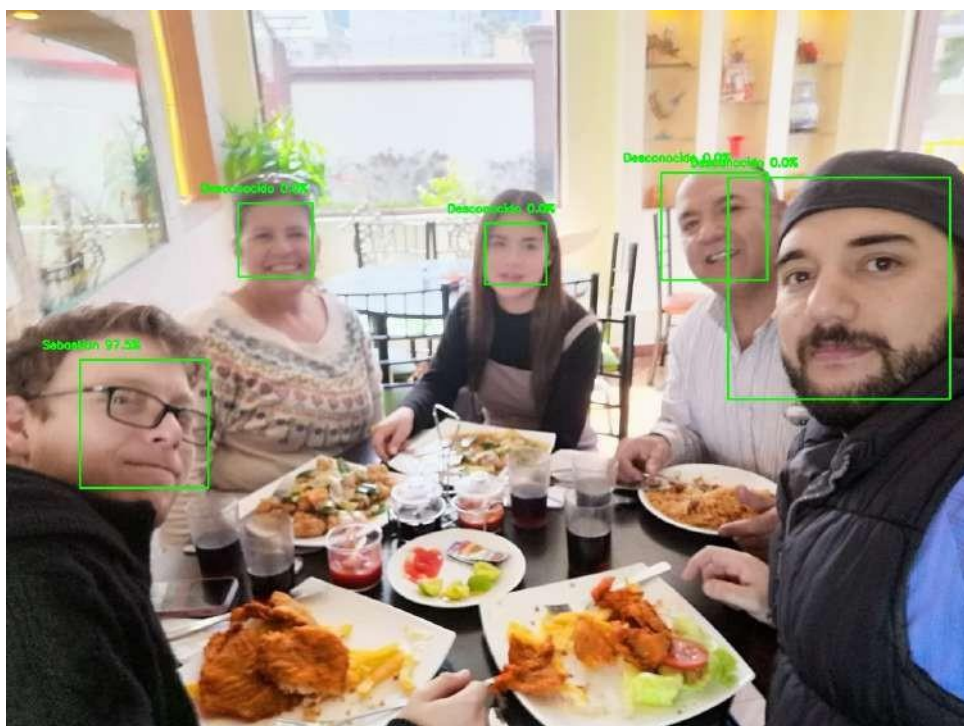




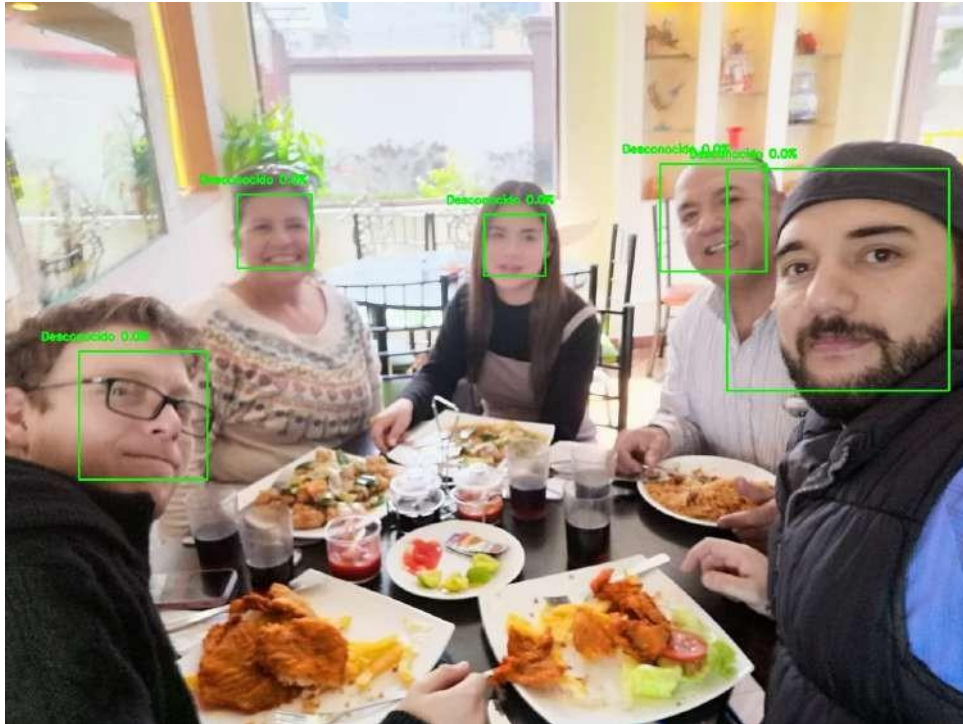
**Figura 37.** Imagen dentro de la base de datos Prueba 3 con "Cascade" y con accesorio



**Figura 38.** Imagen dentro de la base de datos Prueba 3 con "Cascade" y sin accesorio



**Figura 39.** Imagen resultado almacenada Prueba 3 con "Cascade" y con accesorio

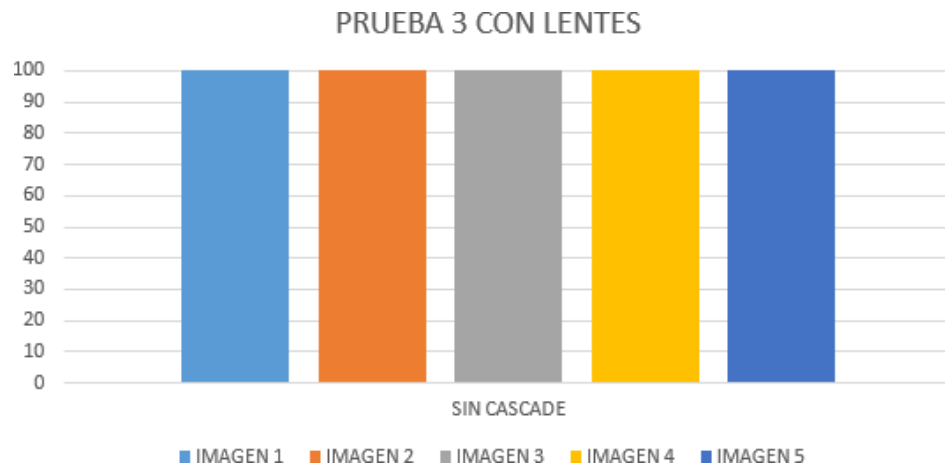


**Figura 40.** Imagen resultado almacenado Prueba 3 con "Cascade" y sin accesorio

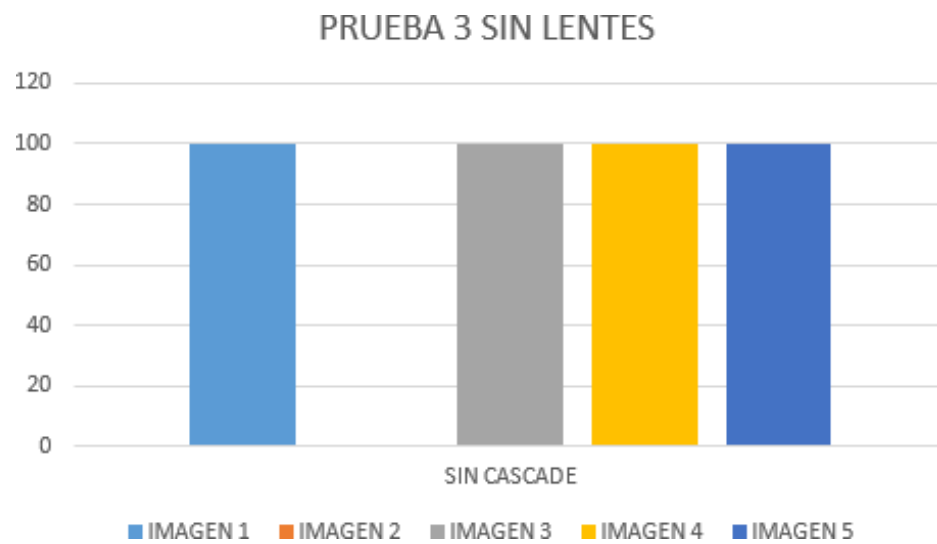
En la Tabla 14 y su representación gráfica en las Figuras 41 y 42 se puede observar los resultados obtenidos sin el uso del clasificador "Cascade" con y sin accesorios. Dando como resultado que no es necesario realizar un pre-procesamiento de imágenes mediante el clasificador "Cascade" y que los accesorios no afectan al reconocimiento facial, si se tiene accesorios de uso permanente como en este caso los lentes se recomienda que se ocupe al momento de realizar la base de datos. Se puede verificar los imágenes dentro de la base de datos en las Figuras 43 y 44 respectivamente y su respectivo resultado en las Figuras 45 y 46.

**Tabla 14.** Prueba 3 Base de datos local sin clasificador "Cascade"

Número de imagen	Porcentaje de acierto con lentes	Porcentaje de acierto sin lentes
Imagen 1	100	100
Imagen 2	100	0
Imagen 3	100	100
Imagen 4	100	100
Imagen 5	100	100



**Figura 41.** Representación Gráfica Prueba 3 sin "Cascade" y con accesorio



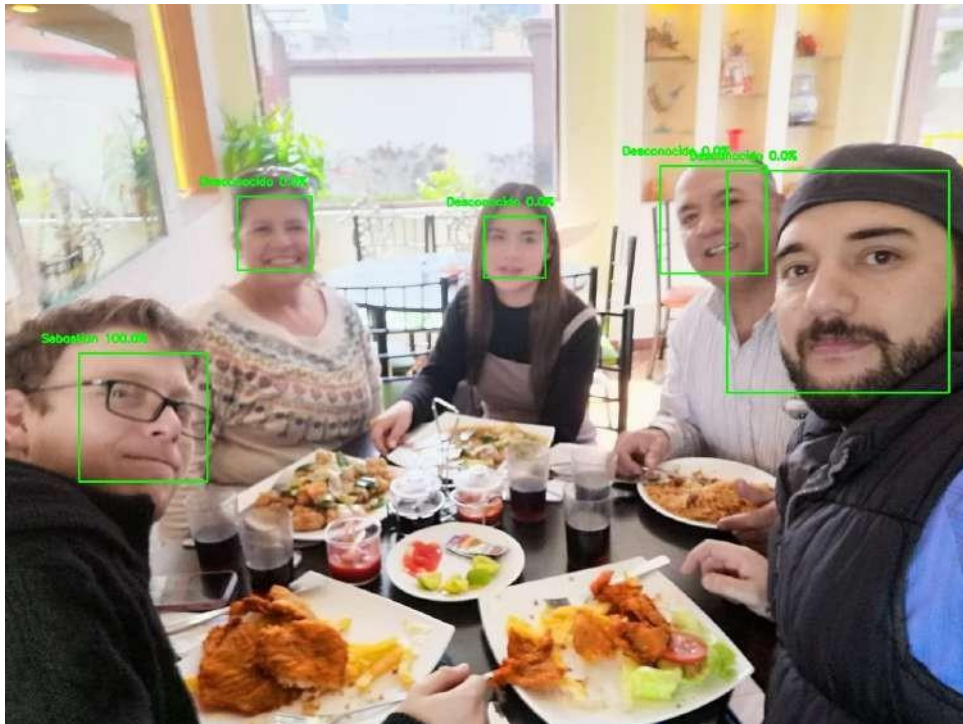
**Figura 42.** Representación Gráfica Prueba 3 sin "Cascade" y sin accesorio



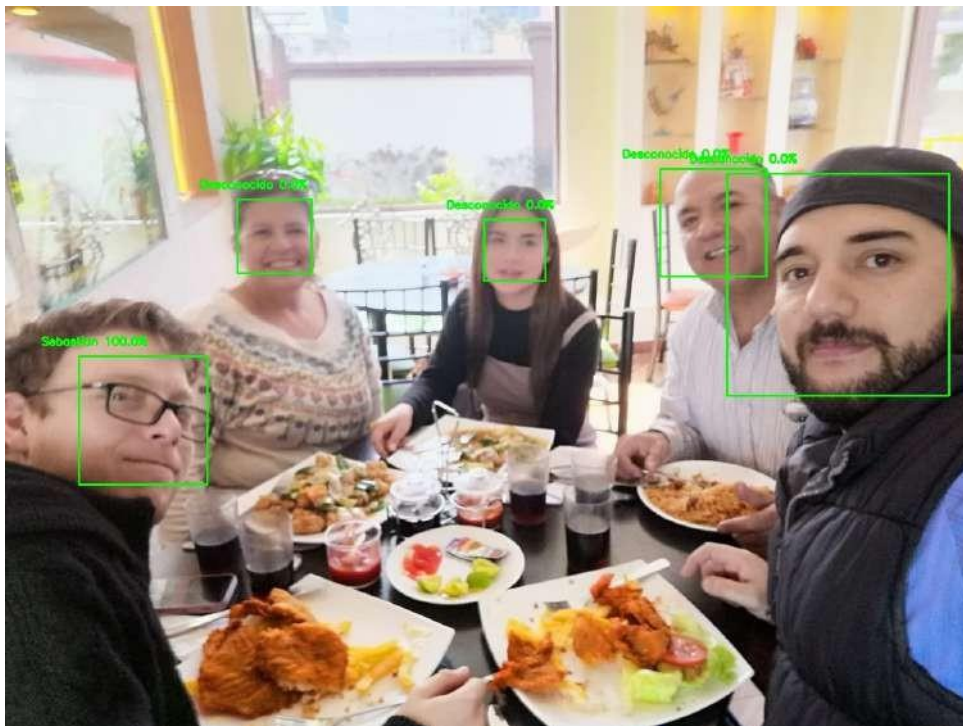
**Figura 43.** Imagen dentro de la base de datos Prueba 3 sin "Cascade" y con accesorio



**Figura 44.** Imagen dentro de la base de datos Prueba 3 sin "Cascade" y sin accesorio



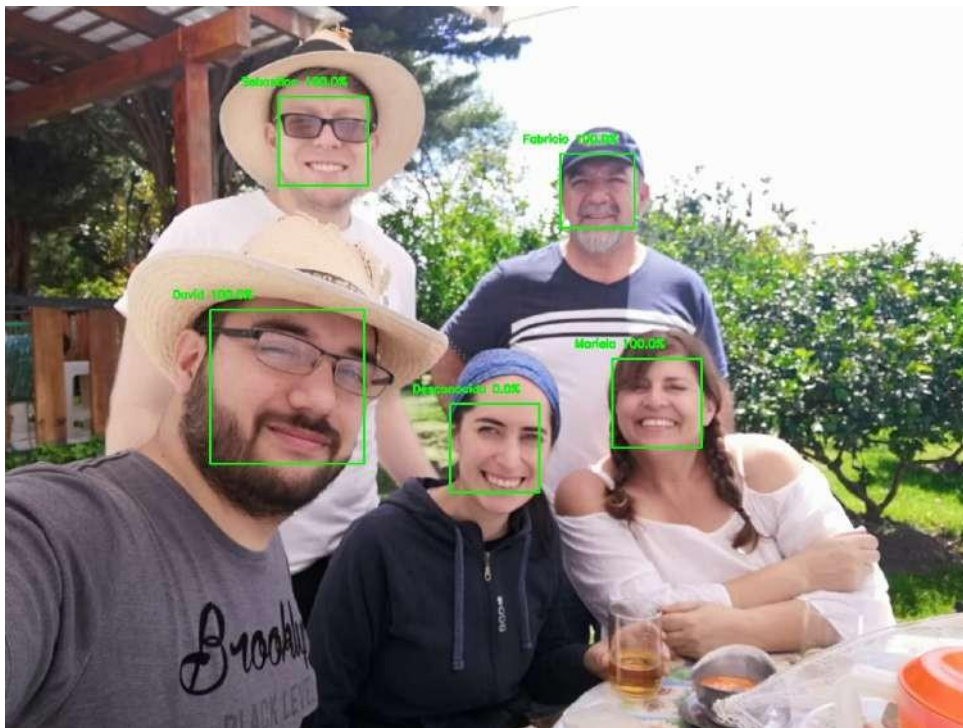
**Figura 45.** Imagen resultado almacenado Prueba 3 sin "Cascade" y con accesorio



**Figura 46.** Imagen resultado almacenado Prueba 3 sin "Cascade" y sin accesorio

#### 10.4. Prueba 4: Determinación de funcionalidad por interferencia de semejanza entre familiares

La cuarta prueba consta del análisis del porcentaje de acierto en 10 imágenes tomando como base de datos a todos los integrantes de una familia. El resultado fue que no existe dificultad / confusión en el caso que dentro de la imagen analizada en tiempo real se encuentre integrantes de la familia como se puede observar en la Figura 47, los resultados en la Tabla 15 y su representación gráfica en la Figura 48.



**Figura 47.** Imagen resultado almacenado Prueba 4

**Tabla 15.** Prueba 4 Análisis de coincidencia entre familiares

Número de imagen	Detección
Imagen 1	100
Imagen 2	100
Imagen 3	100
Imagen 4	100
Imagen 5	100
Imagen 6	100
Imagen 7	100

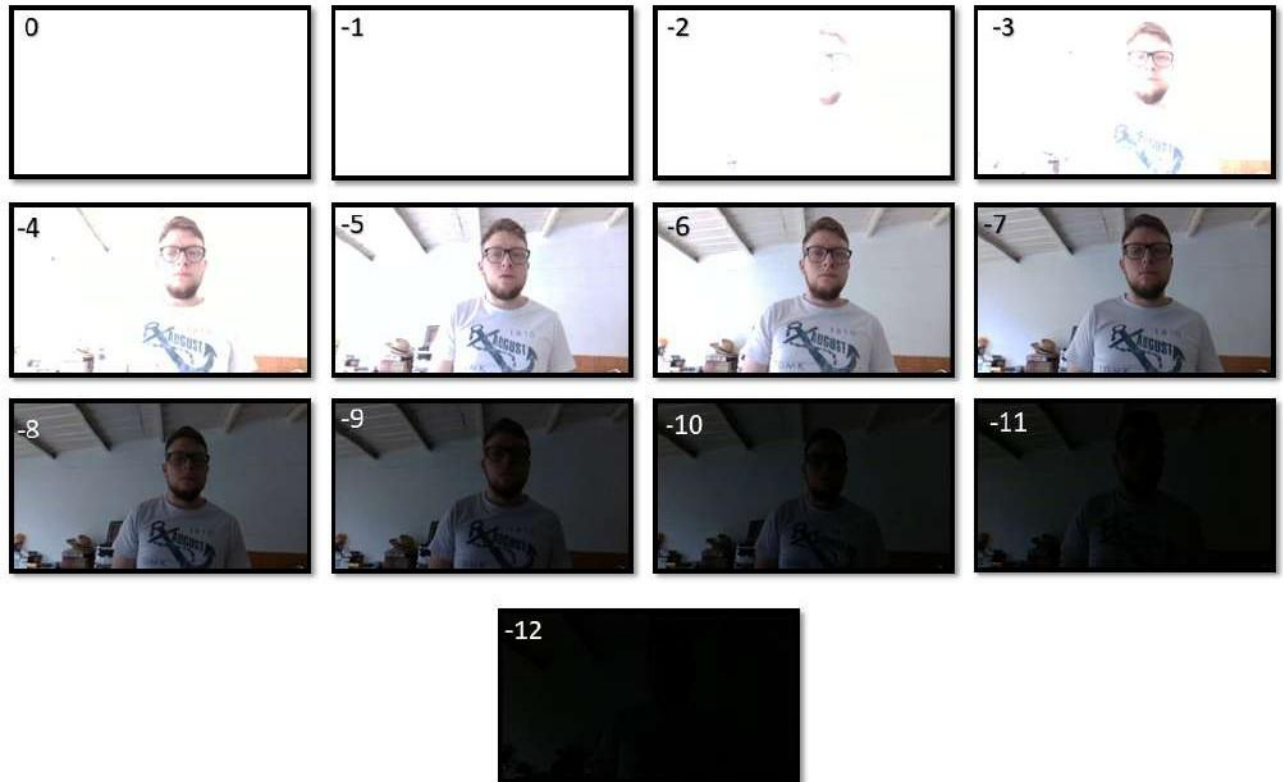
Número de imagen	Detección
Imagen 8	100
Imagen 9	100
Imagen 10	100



**Figura 48.** Representación Gráfica Prueba 4

### 10.5. Prueba 5: Determinación de luminosidad de imágenes de la Base de datos

Para la quinta prueba se necesitó realizar diferentes luminosidades de la zona interior donde se realiza la base de datos local mediante la cámara de profundidad, por lo que mediante 5 aplicaciones se determinó que la luminosidad promedio en interiores es de 112.8 lx y se optó por variar la exposición de la cámara de profundidad, que determina la cantidad de luz que recibe al obtener imágenes. Se realizó un rango de 0 a -12 de exposición a la base de datos y se hizo un análisis de reconocimiento facial a 5 imágenes como se puede observar en la Figura 49.



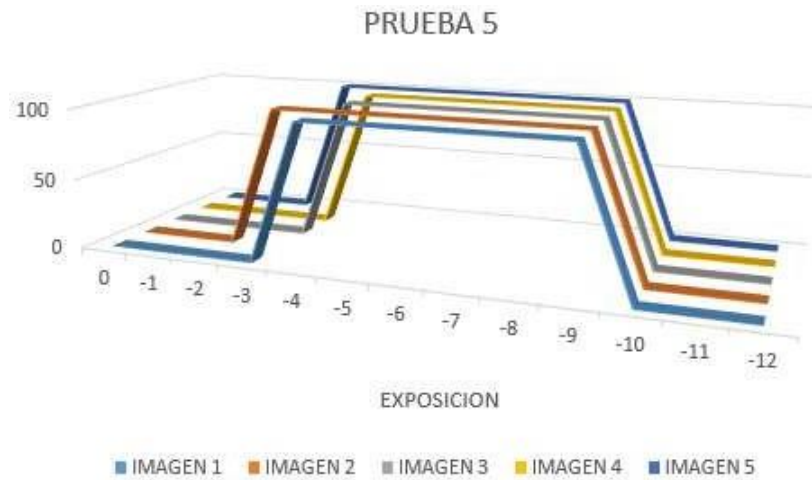
**Figura 49.** Rango de exposición de las imágenes base de datos Prueba 5

El resultado fue que con una luminosidad promedio anteriormente mencionada, la exposición ideal es en el rango de -4 a -9 como se puede observar en la Tabla 16 y su representación gráfica en la Figura 50. Se tomó la decisión de ocupar una exposición de -6 dando un resultado aceptable como se puede ver en la Figura 51.

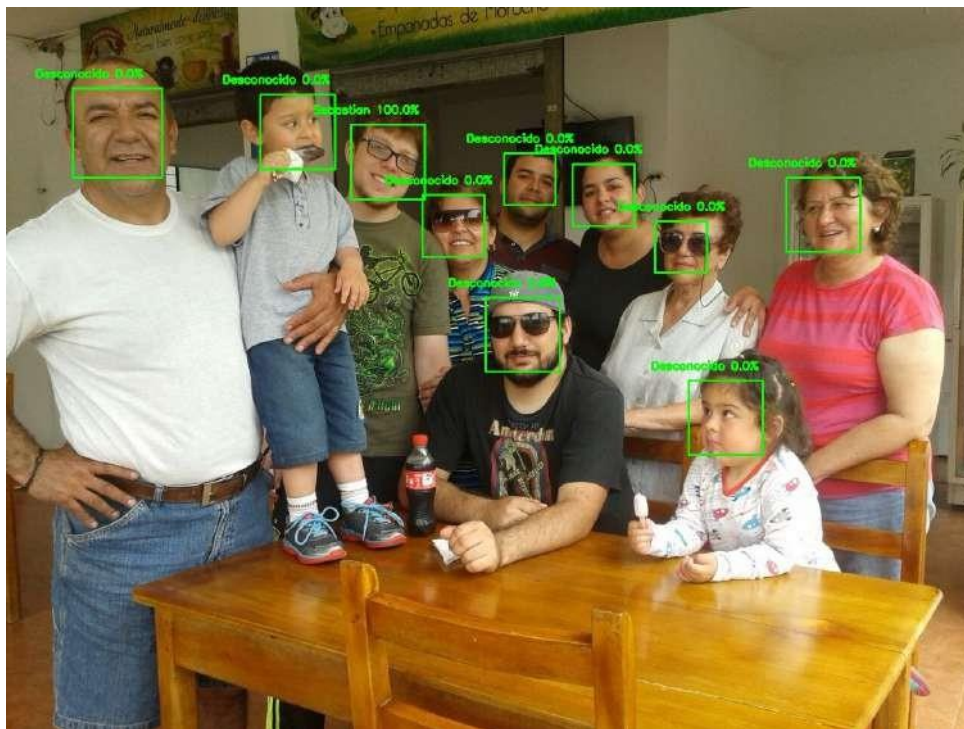
**Tabla 16.** Prueba 5 Luminosidad en la Base de Datos

Número de imagen	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
Imagen 1	0	0	0	0	100	100	100	100	100	100	0	0	0
Imagen 2	0	0	0	100	100	100	100	100	100	100	0	0	0
Imagen 3	0	0	0	0	100	100	100	100	100	100	0	0	0
Imagen 4	0	0	0	0	100	100	100	100	100	100	0	0	0
Imagen 5	0	0	0	100	100	100	100	100	100	100	0	0	0





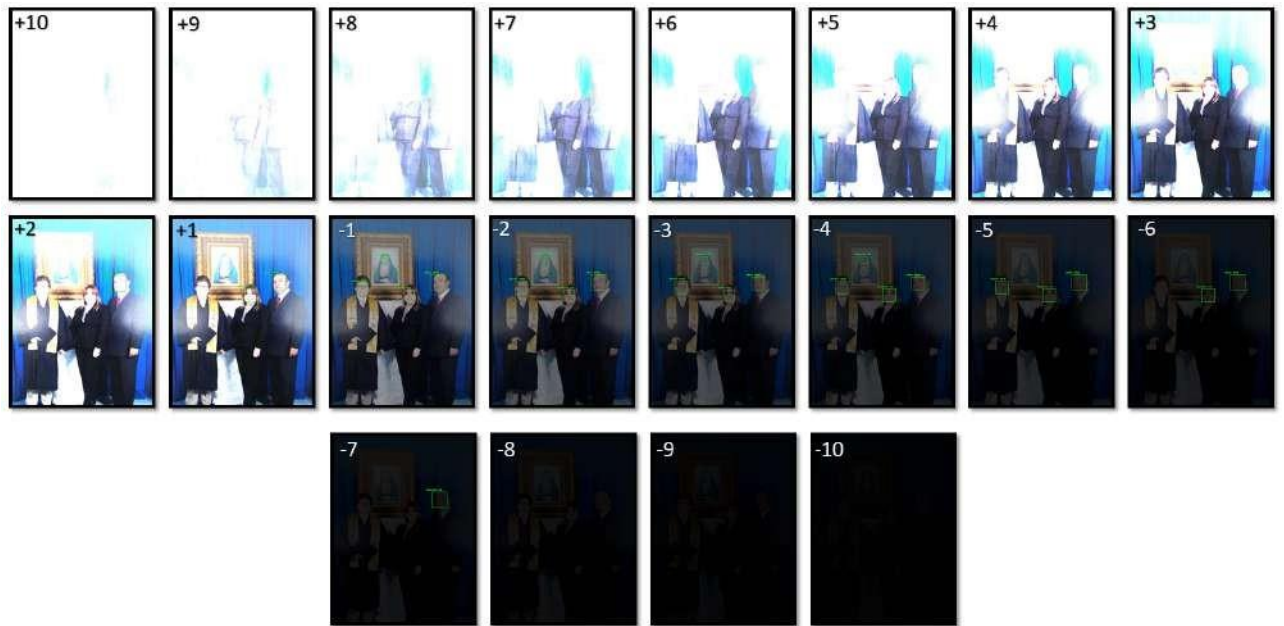
**Figura 50.** Representación Gráfica Prueba 5



**Figura 51.** Imagen resultado almacenado Prueba 5

### 10.6. Prueba 6: Determinación de luminosidad para detección facial

Para la sexta prueba se necesitó realizar diferentes luminosidades de la imagen a ser analizada, por lo que de la misma forma que en la anterior prueba se optó por variar la exposición de las imágenes mediante el programa Adobe® Photoshop CS6 en un rango de +10 a -10 de exposición a cada imagen como se puede observar en la Figura 52 y se procedió con su respectivo análisis tomando como base de datos una familia.



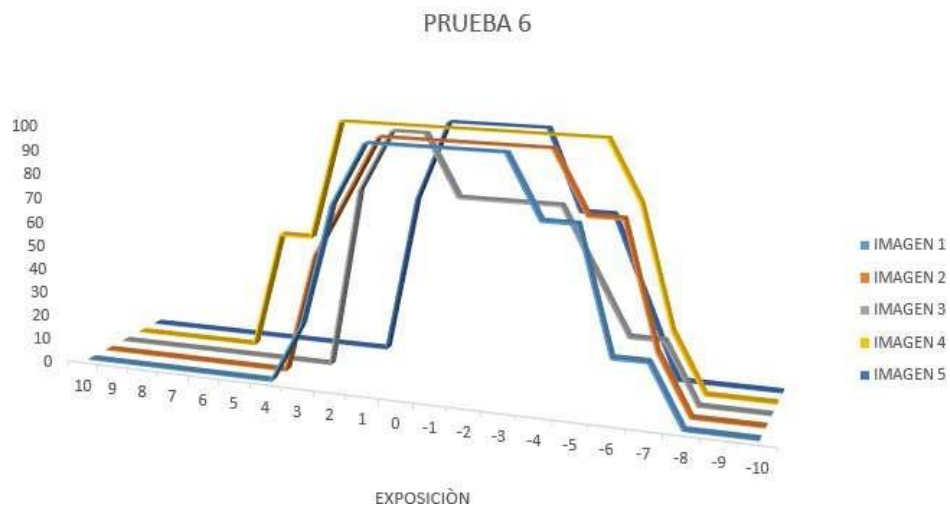
**Figura 52.** Rango de exposición de una imagen Prueba 6

El resultado fue que con una luminosidad promedio se obtienen resultados de detección óptimo, es decir con una exposición 0 pero en el rango de +1 a -3 es aceptable como se puede observar en la Tabla 17 y su representación gráfica en la Figura 53.

**Tabla 17.** Prueba 6 Luminosidad externa

Exposición	Imagen 1	Imagen 2	Imagen 3	Imagen 4	Imagen 5
+10	0	0	0	0	0
+9	0	0	0	0	0
+8	0	0	0	0	0
+7	0	0	0	0	0
+6	0	0	0	0	0
+5	0	0	0	50	0
+4	0	0	0	50	0
+3	25	50	0	100	0
+2	75	75	75	100	0
+1	100	100	100	100	66.667
0	100	100	100	100	100
-1	100	100	75	100	100
-2	100	100	75	100	100

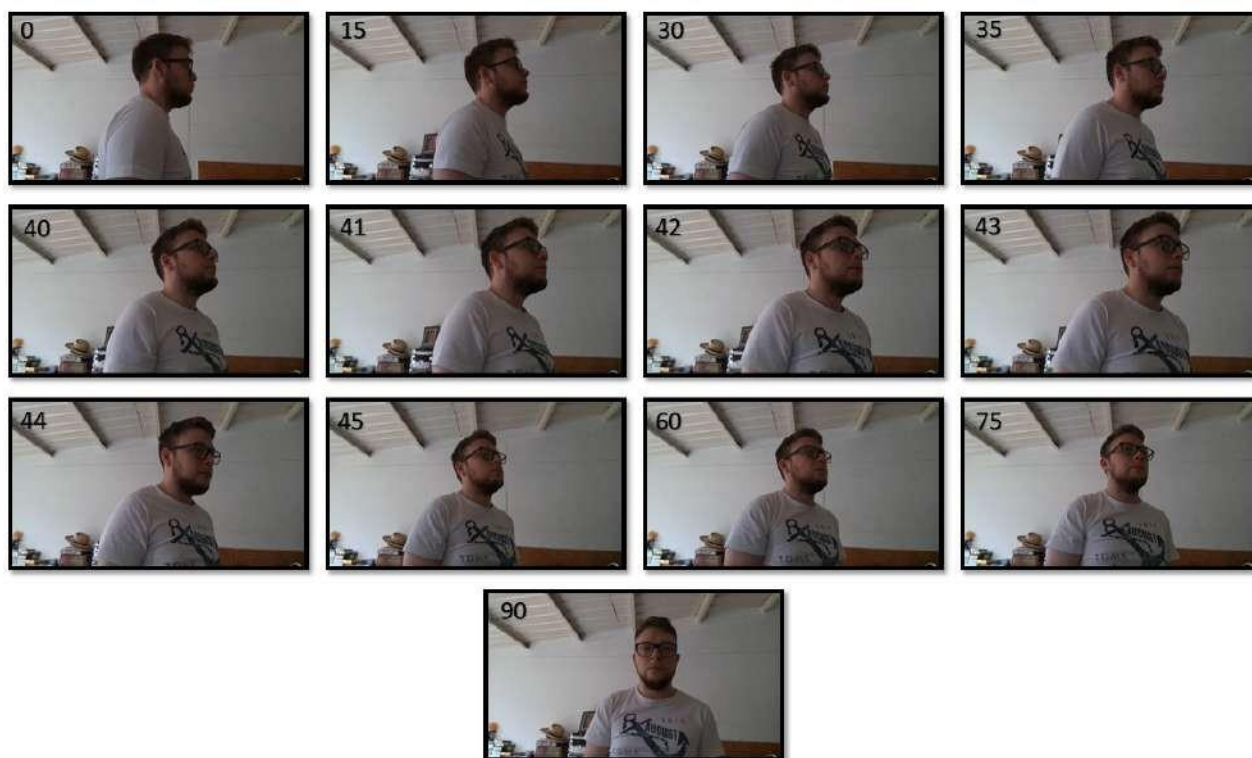
Exposición	Imagen 1	Imagen 2	Imagen 3	Imagen 4	Imagen 5
-3	100	100	75	100	100
-4	75	100	75	100	66.667
-5	75	75	50	100	66.667
-6	25	75	25	75	33.333
-7	25	25	25	25	0
-8	0	0	0	0	0
-9	0	0	0	0	0
-10	0	0	0	0	0



**Figura 53.** Representación Gráfica Prueba 6

### 10.7. Prueba 7: Determinación de ángulo mínimo para detección de rostro

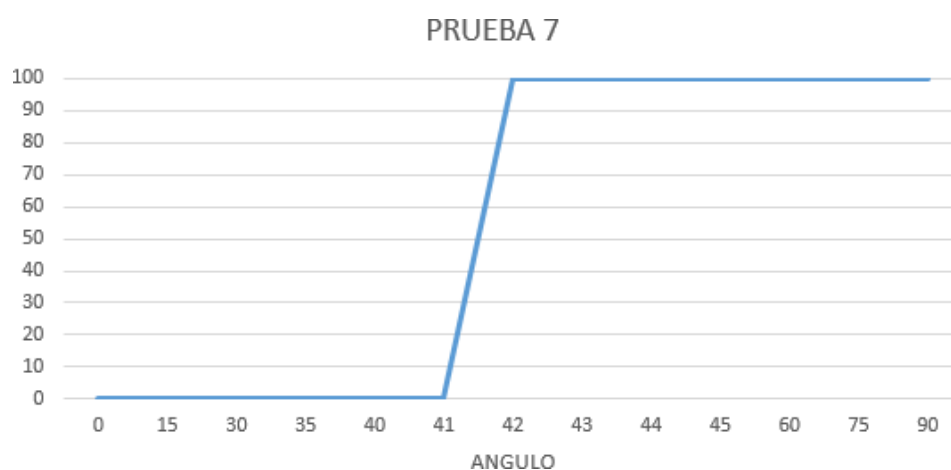
La séptima prueba consiste en el análisis del porcentaje de acierto de un rostro a diferentes ángulos. Se tomaron 13 fotografías en el rango de 0 a 90 grados como se puede ver en la Figura 54, dando como resultado que el sistema es capaz de detectar el rostro a partir de los 42 grados hasta los 132 grados como se puede observar en la Tabla 18 y su representación gráfica en la Figura 55.



**Figura 54.** Fotografías en el rango de 0 a 90 grados Prueba 7

**Tabla 18.** Prueba 7 ángulo mínimo para la detección de rostro

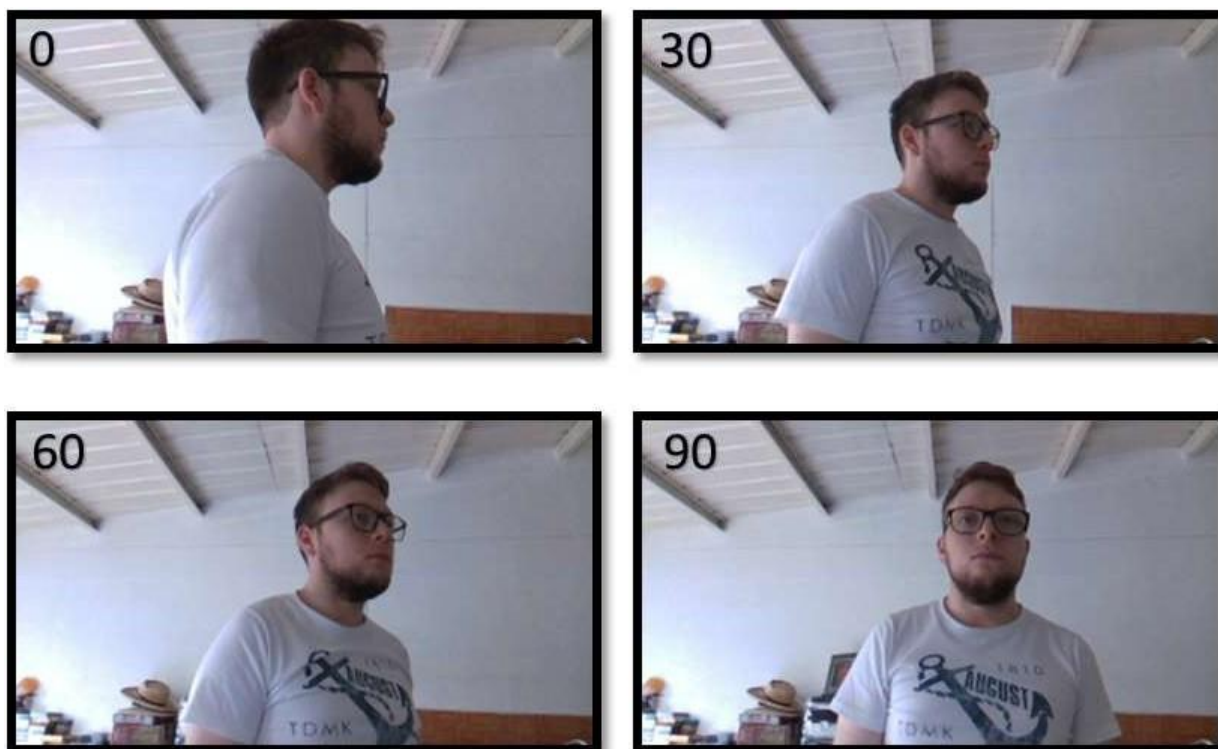
Grado	0	15	30	35	40	41	42	43	44	45	60	75	90
Detección	0	0	0	0	0	0	100	100	100	100	100	100	100



**Figura 55.** Representación Gráfica Prueba 7

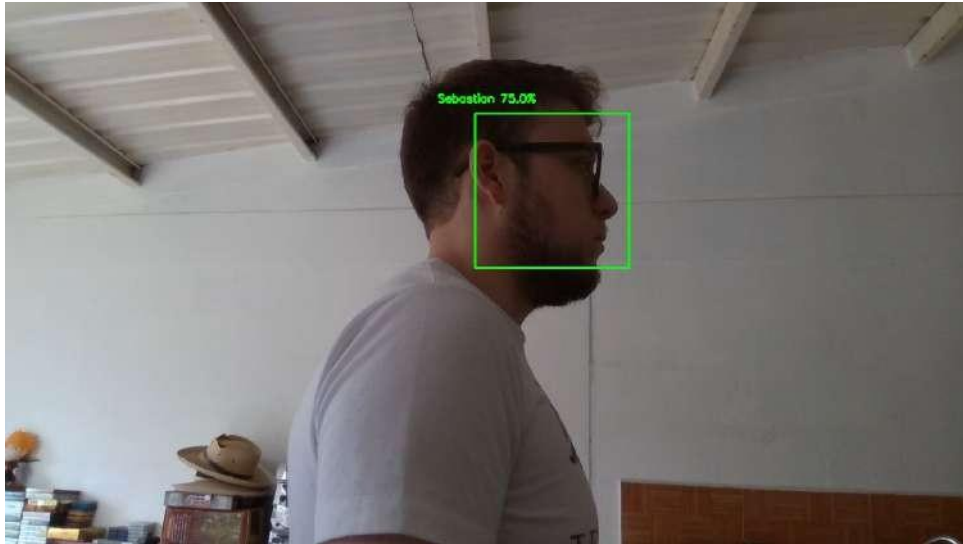
### 10.8. Prueba 8: Determinación de factibilidad de diferentes ángulos dentro de la base de datos

La octava prueba consiste en el análisis del porcentaje de acierto de un rostro a diferentes ángulos, en cinco imágenes anteriormente analizadas tomando en consideración la segunda prueba que dio como resultado una base de datos de 40 por lo que se implementa una base de datos formada por 10 imágenes de 0, 30, 60 y 90 grados como se puede ver en la Figura 56.



**Figura 56.** Fotografías en el rango de 0 a 90 grados Prueba 8

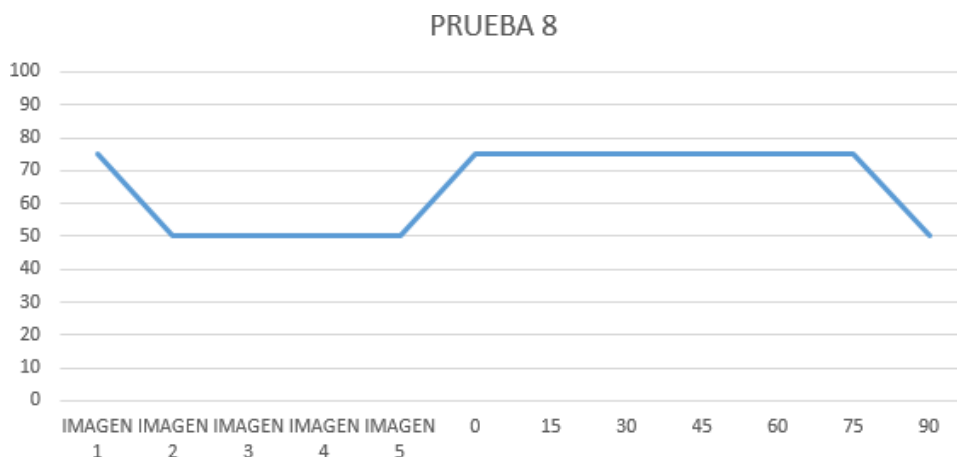
Se tomó la decisión de no aplicar una base de datos conformada por diferentes ángulos porque el sistema es capaz de detectar el rostro en un mayor rango de 0 a 90 grados pero no en un porcentaje aceptable como se puede observar en la Figura 57. La detección facial se encuentra con valores inferiores o iguales a 75 % detallados en la Tabla 19 y su representación gráfica en la Figura 58.



**Figura 57.** Imagen resultado almacenado Prueba 8

**Tabla 19.** Prueba 8 Ángulo de la Base de Datos

Imagen	Detección
Imagen 1	75
Imagen 2	50
Imagen 3	50
Imagen 4	50
Imagen 5	50
Imagen rostro 0 grados	75
Imagen rostro 15 grados	75
Imagen rostro 30 grados	75
Imagen rostro 45 grados	75
Imagen rostro 60 grados	75
Imagen rostro 75 grados	75
Imagen rostro 90 grados	50



**Figura 58.** Representación Gráfica Prueba 8

### 10.9. Prueba 9: Determinación de distancia máxima

La novena prueba consiste en el análisis del porcentaje de acierto de un rostro a diferentes distancias para la determinación de la distancia máxima de detección como se puede ver en la Figura 59.



**Figura 59.** Fotografías a diferentes distancias en metros Prueba 9

El resultado fue que la distancia máxima de reconocimiento es 4.3 metros como se puede observar en la Tabla 20 y su representación gráfica en la Figura 60.

**Tabla 20.** Prueba 9 Distancia máxima de reconocimiento

Distancia [m]	Detección
1	100

Distancia [m]	Detección
2	100
3	100
4	100
4.1	100
4.2	100
4.3	100
4.4	0
4.5	0
5	0



**Figura 60.** Representación Gráfica Prueba 9

#### 10.10. Prueba 10: Determinación de calidad de imágenes

La décima prueba consiste en el análisis del porcentaje de acierto de un rostro en una imagen a diferentes resoluciones, específicamente de: 144p (256 x 144 píxeles), 240p (426 x 240 píxeles), 360p (640 x 360 píxeles), 480p (854 x 480 píxeles), 720p (1280 x 720 píxeles) y 1080p (1920 x 1080 píxeles) para determinar la calidad mínima de la imagen, para el adecuado funcionamiento del sistema como se puede observar en la Figura 61.





**Figura 61.** Imagen a diferentes resoluciones Prueba 10

El resultado fue que la resolución mínima es de 240p (426 x 240 píxeles) como se puede observar en la Tabla 21 y su representación gráfica en la Figura 62. Cabe destacar que la cámara de profundidad satisface esta necesidad ya que tiene una resolución HD es decir 720p (1280 x 720 píxeles).

**Tabla 21.** Prueba 10 Calidad de imágenes

Resolución	Detección
144p (256 x 144 píxeles)	0
240p (426 x 240 píxeles)	100
360p (640 x 360 píxeles)	100
480p (854 x 480 píxeles)	100
720p (1280 x 720 píxeles)	100
1080p (1920 x 1080 píxeles)	100



**Figura 62.** Representación Gráfica Prueba 10

### 10.11. Prueba 11: Determinación de calidad de imágenes de la base de datos

La décima primera prueba consiste en el análisis del porcentaje de acierto de un rostro con una base de datos con diferentes resoluciones (iguales a las de la prueba 10) para determinar la calidad mínima de imagen dentro de la base de datos para el adecuado funcionamiento del sistema como se puede observar en la Figura 63.



**Figura 63.** Imágenes de la base de datos a diferentes resoluciones Prueba 11

El resultado fue que el reconocimiento fue exitoso en todas las resoluciones pero se recomienda el uso de imágenes de igual o mayor calidad a la cámara a ser ocupada, en este caso se decide realizar la base de datos a 720p (1280 x 720 píxeles) por las características de resolución de la cámara de profundidad como se puede observar en la Tabla 22 y su representación gráfica en la Figura 64.

**Tabla 22.** Prueba 11 Calidad de imágenes dentro de la base de datos

Revolución	Detección
144p (256 x 144 píxeles)	100
240p (426 x 240 píxeles)	100
360p (640 x 360 píxeles)	100
480p (854 x 480 píxeles)	100
720p (1280 x 720 píxeles)	100
1080p (1920 x 1080 píxeles)	100

**Figura 64.** Representación Gráfica Prueba 11

### 10.12. Prueba 12: Determinación de calidad de vídeo

La décima segunda prueba consiste en el análisis del porcentaje de acierto de rostros dentro de un vídeo donde se puede observar diferentes personajes de la serie Glee con diferentes resoluciones (iguales a las de la prueba 10 y 11). El vídeo tiene una duración de 15 minutos 17 segundos y se lo analizó con el método de observación. Tres fragmentos del vídeo con su respectivo análisis se lo puede ver en las Figuras 65, 66 y 67.

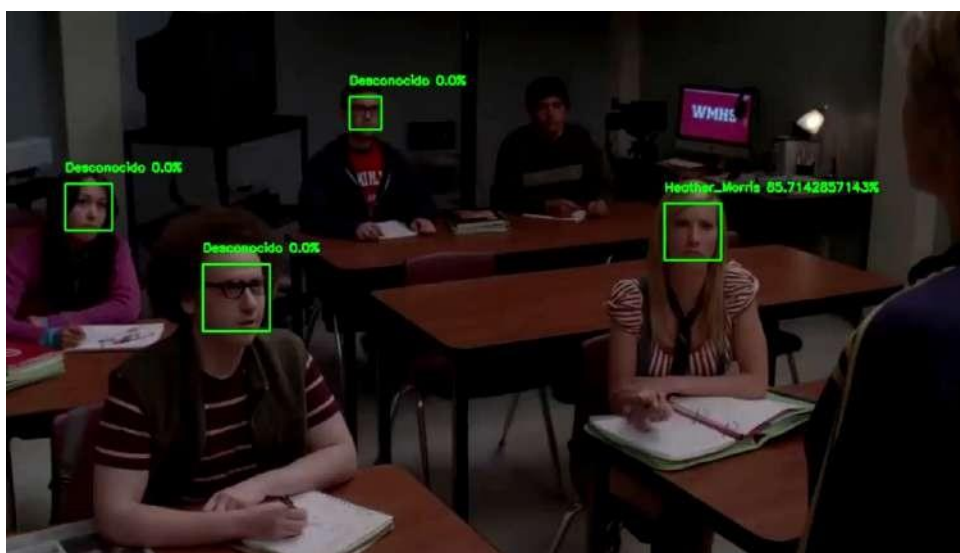


Figura 65. Fragmento 1 Vídeo serie Glee a 720p Prueba 12

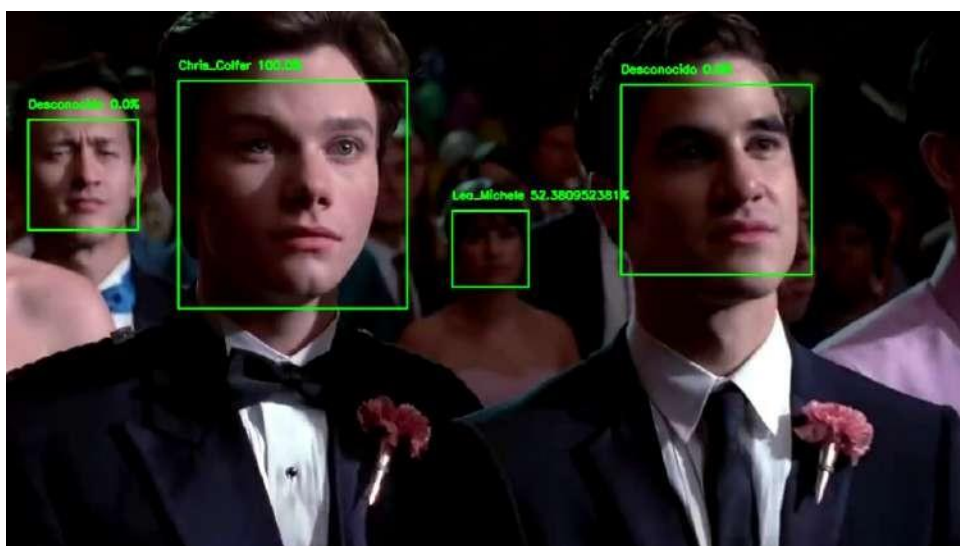


Figura 66. Fragmento 2 Vídeo serie Glee a 720p Prueba 12

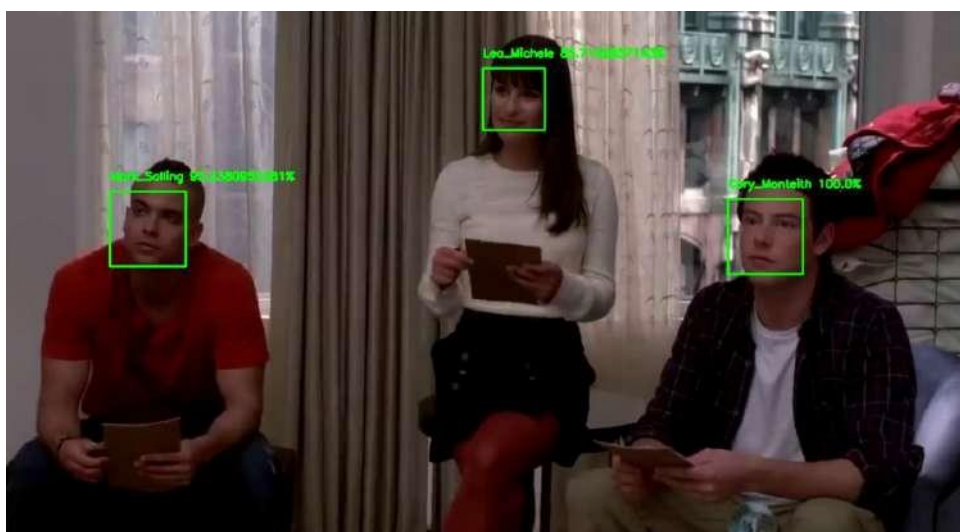


Figura 67. Fragmento 3 Vídeo serie Glee a 720p Prueba 12

El resultado obtenido mediante observación indica que mientras mayor calidad sea el vídeo se obtiene una mejor identificación facial pero con una resolución de 720p se obtuvo un porcentaje de reconocimiento aceptable y se tiene una menor dificultad al momento de detectar rostros, mientras que con una resolución menor es decir de 480p a 144p existe una mayor dificultad de reconocimiento del rostro y el porcentaje no es aceptable.

### 10.13. Prueba 13: Comparación de cámara de profundidad con cámara normal

La décima tercera prueba consiste en la comparación de una cámara normal de 20 Megapíxeles con la cámara de profundidad con características mencionadas anteriormente. Para la comparación se realizó una base de datos y un análisis facial en imágenes a diferentes distancias y ángulos obtenidas con la cámara normal para diferenciarlas con pruebas anteriores. En la Figura 68 se puede observar la diferencia de las imágenes obtenidas con las cámaras.



**Figura 68.** Imágenes de las bases de datos de cada cámara

Como se puede visualizar la cámara de profundidad da una distancia de profundidad de 0.3 metros y brinda una mayor vista a los alrededores.

Al realizar la detección de rostro dentro de las imágenes a diferentes distancias se pudo obtener exactamente el mismo resultado que con la cámara de profundidad, es decir que la distancia máxima de detección es de 4.3 metros con ambas cámaras, el análisis de las imágenes a diferentes ángulos mostró que con una cámara normal se detecta hasta un máximo de 45 grados mientras que la cámara de profundidad es capaz de detectar hasta 42 grados como se puede observar en las Tablas 23 y 24 con sus representaciones gráficas en las Figuras 69 y 70

**Tabla 23.** Prueba 13 Distancia máxima de reconocimiento con cámara normal

Distancia [m]	Detección
1	100
2	100
3	100
4	100
4.1	100
4.2	100
4.3	100
4.4	0
4.5	0
5	100

**Figura 69.** Representación Gráfica distancias Prueba 13**Tabla 24.** Prueba 13 Ángulo mínimo para la detección de rostro con cámara normal

Ángulo	Detección
0	0
15	0
30	0
41	0

Ángulo	Detección
42	0
43	0
44	0
45	100
60	100
75	100
90	100



**Figura 70.** Representación Gráfica ángulos Prueba 13

#### 10.14. Prueba 14: Comparación del sistema con cámaras de seguridad en el mercado actual

Como última prueba se realizó una comparativa del prototipo final con cámaras de seguridad con reconocimiento facial dentro del mercado.

En el mercado los sistemas de detección facial se encuentran de tres maneras: como software (licencia), como un dvr es decir un grabador de vídeo digital (servidor) y como una cámara con el sistema integrado.


En el distrito metropolitano de Quito se han instalado cámaras Hikvision, las cuales logran una identificación facial mediante el software Hik control logrando hasta 96 % de efectividad. Esto se efectivizó como plan de videovigilancia de la ciudad en dos fases: la piloto conformada por 78 dispositivos colocados en diferentes puntos estratégicos del Centro His-


tórico y una segunda fase donde se implementará 290 dispositivos en los sectores del sur obteniendo en su totalidad 2000 cámaras aproximadamente a lo largo del distrito, siendo 368 capaces de realizar reconocimiento facial. [42] [43]

La empresa Hikvision ofrece diferentes productos como las cámaras AcuSense que son basadas en algoritmos de deep learning con la capacidad de detectar objetos, personas y vehículos pero no tiene la capacidad de reconocer rostros, para esto ofrece las tres opciones anteriormente mencionadas, los software que tiene son: Hik-Connect, Hik-ProConnect, HikCentral e iVMS-5200. En lo que respecta a servidores dvr existen 6 modelos: IDS-7204HQHI-M1/FA, IDS-7204HUHI-M1(2)/FA, IDS-7208HQHI-M1/FA, IDS-7208HUHI-M1(2)/FA e IDS-7216HQHI-M1/FA, lo que les diferencia uno de otro es la cantidad de canales que admiten y los megapíxeles que soportan. Y por último ofrece una cámara de reconocimiento facial IDS-2CD8426G0/F-I. En la Tabla 25 se puede observar la comparación de las características de los productos.



**Tabla 25.** Características Sistemas Hikvision

Sistema	Características
<p data-bbox="261 640 389 674">Software</p> <p data-bbox="261 703 411 736">HikCentral</p> 	<p data-bbox="676 349 1318 383">Depende de las características de la cámara.</p> <p data-bbox="676 412 1437 506">Es una plataforma para los sistemas de gestión de vídeo.</p> <p data-bbox="676 539 999 573">Reconocimiento facial.</p> <p data-bbox="676 607 983 640">Visualización en vivo.</p> <p data-bbox="676 674 959 707">Imágenes térmicas.</p> <p data-bbox="676 741 1015 775">Reproducción de vídeo.</p> <p data-bbox="676 808 943 842">Control de acceso.</p> <p data-bbox="676 875 1078 909">Administración de vehículos.</p> <p data-bbox="676 943 1086 976">Almacenamiento expandible.</p> <p data-bbox="676 1010 1398 1043">Asistencia automatizada por reconocimiento facial.</p> <p data-bbox="676 1077 1062 1111">Administración de alarmas.</p> <p data-bbox="676 1144 975 1178">Control de visitantes.</p> <p data-bbox="676 1211 1007 1245">Control de ascensores.</p> <p data-bbox="676 1279 959 1312">Monitoreo continuo.</p> <p data-bbox="676 1346 1334 1379">Smart Wall (visualización agradable a la vista).</p> <p data-bbox="676 1413 1150 1447">Costo licencia una cámara \$ 385.</p>

Sistema	Características
<p data-bbox="261 539 331 568">DVR</p> <p data-bbox="261 602 587 631">IDS-7204HQHI-M1/FA</p> 	<p data-bbox="676 271 1321 300">Depende de las características de la cámara.</p> <p data-bbox="676 333 1437 427">Algoritmo de deep learning para la comparación de imágenes de rostros.</p> <p data-bbox="676 461 1437 618">Un canal para reconocimiento facial o 4 canales para disminución de falsa alarma (detección de humanos, objetos y vehículos).</p> <p data-bbox="676 651 1437 745">Biblioteca de imágenes faciales de hasta 80 MB de capacidad, hasta 16 bibliotecas de imágenes faciales.</p> <p data-bbox="676 779 1437 936">Admite cámaras de 5 megapíxeles a 12 cuadros por segundo, en el caso del canal 1 soporta hasta 8 megapíxeles a 8 cuadros por segundo.</p> <p data-bbox="676 969 1166 999">Tiene un máximo de 8 cámaras IP.</p> <p data-bbox="676 1032 1054 1061">Tiene un costo de \$ 1.739.</p>

Sistema	Características
<p data-bbox="264 658 376 689">Cámara</p> <p data-bbox="264 719 555 750">IDS-2CD8426G0/F-I</p> 	<p data-bbox="676 271 1437 365">Algoritmo de deep learning para la comparación de imágenes de rostros.</p> <p data-bbox="676 398 1294 430">Dual Lente 2mm (2.8mm/4mm opcionales).</p> <p data-bbox="676 463 1437 557">Resolución 1920 x 1080 píxeles a 30 cuadros por segundo.</p> <p data-bbox="676 591 890 622">2 megapíxeles.</p> <p data-bbox="676 656 1374 687">Almacenamiento SD/SHHC/ SDXC hasta 128Gb.</p> <p data-bbox="676 721 1043 752">DUAL Chip CMOS 1/2.8".</p> <p data-bbox="676 786 1249 817">Tensión 36V (Fuente 220 Vac a 36 Vdc).</p> <p data-bbox="676 851 1126 882">Iluminación IR hasta 10 metros.</p> <p data-bbox="676 916 1027 947">Estabilizador de imagen.</p> <p data-bbox="676 981 1075 1012">30 rostros simultáneamente.</p> <p data-bbox="676 1046 887 1077">Material Metal.</p> <p data-bbox="676 1111 1066 1142">Entrada de Alarma y Audio.</p> <p data-bbox="676 1176 1342 1207">Altura recomendada de instalación 1,8 metros.</p> <p data-bbox="676 1240 1437 1335">Distancia de detección 1,9 metros, ancho hasta 2,5 metros.</p> <p data-bbox="676 1368 1222 1400">Dimensiones 180.4 x 147 x 117.9 mm.</p> <p data-bbox="676 1433 858 1464">Peso 1,5 Kg.</p> <p data-bbox="676 1498 1066 1529">Tiene un precio de \$ 1.827.</p>

Entre los sistemas que ofrece la empresa Hikvision si se toma como referencia la precisión del reconocimiento visual y el diseño para el uso en interiores, la mejor es la cámara IDS-2CD8426G0/F-I ya que como se observo en pruebas anteriores, mientras sea mejor la calidad de la imagen se obtendrá un margen menor de error. Además los otros productos dependen de una inversión extra para cámaras de seguridad y apuntan a un uso más industrial por lo que tienen un elevado costo.

Comparando con la cámara de profundidad Intel®RealSense™ modelo D415 con la cámara Hikvision en lo que respecta a características ópticas es inferior, esto se evidencia en

la resolución ya que esta es 720p pero la supera en los cuadros por segundo que admite y lo más importante en la distancia de profundidad que ofrece. Al comparar la estructura de la cámara Hikvision con el prototipo se puede observar que las dimensiones de la Hikvision son más pequeñas y tiene una estructura más sólida, pero carece de movimiento de 360 grados. En conclusión el sistema puede competir tomando como referencia a la compañía Hikvision ya que el prototipo tiene tanto un costo menor, utiliza software libre, tiene un porcentaje de reconocimiento facial semejante, ofrece tecnología de profundidad y realiza un movimiento global de 360 grados.

## 11. Conclusiones

El método de Deep Learning con las redes neuronales convolucionales satisfacen la necesidad de clasificar una gran cantidad de datos sin saturar el sistema con demasiadas restricciones puesto que consta con una secuencia de convoluciones y reducciones (pooling).

La estructura de la cámara de seguridad es de tipo domo con mecanismos PTZ ya que es para uso de interiores, tiene una sujeción directa a la superficie, consta de protección a los elementos internos y genera un movimiento de 360 grados.

La cámara de profundidad ocupada para el proyecto es la cámara Intel® RealSense™ modelo D415 ya que tiene un costo accesible, dispone de una distancia de profundidad representativa y tiene una estructura externa que permite la sujeción adecuada en el prototipo.

El tipo de motor seleccionado es el servomotor de corriente continua por su costo, sus dimensiones pequeñas y porque cumple con un torque medio para la rotación de la cámara.

El material para la estructura del prototipo es de ABS ya que satisface todas las características mecánicas para el uso de una cámara de seguridad de uso de interiores.

La fuerza de apoyo de una persona promedio a 30 grados es igual al 20 % de su peso.

El cálculo en viga en voladizo con cargas puntuales y carga distribuida permite la obtención de un espesor referencia por la forma cilíndrica no uniforme del prototipo. La decisión de la forma cilíndrica se la toma mediante simulación y fue de 5mm.

El Torque mínimo de los motores requerido para el prototipo es de 1 Nm.

El dimensionamiento del eje al estar construido de ABS, que es un material frágil, se de-

be realizar mediante un diseño estático y comprobarlo por medio de un ensayo destructivo de torsión para determinar el número de ciclos que soporta el eje.

El diseño mecánico analítico fue comprobado mediante simulación con errores menores al 12 % en el software de diseño SOLIDWORKS® Student Edition 2019-2020.

La simulación del torque de ajuste es necesario para garantizar que el prototipo no falle al momento de ser instalado. Se garantiza la seguridad seleccionando un perno que falle antes que el prototipo.

El software que se ocupa para el proyecto es Python™ en la versión 2.7 el cual es implementado en proyectos de investigación a nivel mundial porque consta con la mayoría de librerías a diferencia de sus versiones superiores. La librería OpenCV puede trabajar sin inconvenientes en esta versión.

El proyecto da tres opciones para la base de datos, la primera es una pre-hecha, la segunda a partir de la descarga de imágenes de Google y la tercera es la obtención en tiempo real mediante una cámara.

Para la descarga de imágenes de la base de datos de Google se necesita obligatoriamente una API key que es un identificador que sirve como el medio de autenticación de un usuario para el uso de los servicios proporcionados por google, esta se obtiene desde su pagina oficial como una prueba gratuita de 30 días con un consumo limitado, en el caso que se requiera un mayor uso se debe pagar \$7.

En el momento de realizar la descarga de imágenes para la base de datos es muy importante considerar los posibles errores HTTP para su adecuado funcionamiento.

La creación de la red neuronal convolucional es mediante una serie de procesos, primero se realiza una convolución y un pooling que permite la obtención de las características faciales, luego se realiza una identificación facial y una codificación con una red neuronal ya entrenada con 3 millones de imágenes de rostros con una precisión de 99.38 % generando un vector de 128d categorizando cada persona dentro de la base de datos.

Para la detección de rostros es bastante importante el tratamiento de imagen como por ejemplo transformar a RGB para que las características faciales destaquen y sean más fáciles de encontrar.

El programa de reconocimiento facial realiza una comparación de la imagen, vídeo pre-

grabado o en tiempo real con la codificación de la red neuronal convolucional.

El porcentaje de coincidencia se calcula con el número de coincidencias dividido para el tamaño de la base de datos, multiplicado por 100.

Los rectángulos, nombre y porcentaje de coincidencia son de color verde ya que por estadística el verde es el color menos ocupado, por lo que se podrá visualizar sin inconvenientes salvo circunstancias especiales.

La tarjeta de vídeo es de suma importancia para el desarrollo de Deep Learning ya que el tiempo en que se entrena la red neuronal convolucional es mucho menor. En una computadora de escritorio con una tarjeta gráfica NVIDIA ® GeForce GTX 1650 para una imagen en la base de datos se demora 8.67 segundos a diferencia de la computadora portátil que toma un tiempo de 290.83 segundos. Al momento de realizar una red con una base de datos de un gran tamaño es decir con un número significativo de imágenes existe la necesidad de tener una tarjeta gráfica.

El actuador seleccionado es el servomotor MG995 de rotación continua, es decir que gira 360 grados, ya que sus dimensiones son acoplables al prototipo, no consume tanta corriente (100 mA), gira a una velocidad semejante a las cámaras de seguridad en el mercado y tiene un torque mayor a 1 Nm.

El microcontrolador elegido es el Arduino UNO ATmega328P porque sus dimensiones son acoplables a la base del prototipo, tiene un conector hembra para su alimentación, ofrece los pines necesarios para la conexión de los actuadores y brinda una fácil conectividad, no requiere una fuente externa y es de fácil adquisición.

Para las diferentes pruebas se realizaron tres bases de datos: La primera se realizó mediante la descarga de imágenes de Google de la serie Glee, la segunda base de datos es realizada mediante la cámara de profundidad seleccionada y contiene imágenes de personas conocidas y la tercera base de datos se realizó con una cámara normal de 20 Megapíxeles.

El resultado de la primera prueba fue que para que exista un porcentaje de acierto mayor al 85 % se debe tener 21 imágenes por cada personaje de la base de datos obtenida de Google.

La segunda prueba dio un resultado de 40 imágenes dentro de la base de datos local

obtenidas por una cámara de profundidad para que exista un porcentaje de acierto mayor al 85 %.

Al momento de comparar en la tercera prueba el uso del clasificador Cascade con el mismo programa sin ocupar este método para la creación de la base de datos y el uso de accesorio dio como resultado que no es necesario realizar un pre-procesamiento de imágenes mediante el clasificador Cascade y que los accesorios no afectan al reconocimiento facial.

En la cuarta prueba se determinó que el programa de detección facial no tiene ninguna dificultad o confusión en el caso que se encuentren familiares en la imagen, vídeo pre-grabado o en tiempo real analizado.

Con una luminosidad promedio de 112.8 lx, la exposición ideal, la cual se modifica dentro del código para las imágenes de la base de datos, determinada en la quinta prueba es en el rango de -4 a -9. Se tomó la decisión de ocupar una exposición de -6.

Mediante el programa Adobe® Photoshop CS6 se modificó la exposición de imágenes en la sexta prueba para determinar la luminosidad ideal para la detección facial y fue de un rango +1 a -3 es decir que con una luminosidad promedio de 112.8 lx el reconocimiento es óptimo.

El ángulo mínimo para detección de rostros obtenido en la prueba siete es de 42 grados, dando así un rango de hasta 132 grados.

En búsqueda de un mayor ángulo de detección en la prueba ocho se realizó una base de datos conformada por imágenes de un rostro a diferentes ángulos y dio como resultado que se detecta el rostro en un mayor rango, pero no es factible ya que se encontró con porcentaje de detección inferiores o iguales al 75 %.

La distancia máxima determinada en la prueba nueve es de 4.3 metros.

La calidad de la imagen de los rostros que se desean detectar en la prueba diez dio como resultado que la resolución mínima es de 240p (426 x 240 píxeles).

La prueba once determina que en la calidad de imágenes que conforman la base de datos no importa su resolución.

El análisis por observación en la prueba doce dio como resultado que mientras mayor sea la calidad del vídeo, se tendrá una mejor identificación facial con una menor dificultad.

En la prueba trece se realizó una base de datos y se tomó diferentes imágenes a diferentes distancias y ángulos con una cámara normal de 20 Megapíxeles, para compararla con la cámara de profundidad la cual se ocupó para las pruebas anteriores. La diferencia es que la imagen obtenida por la cámara de profundidad brinda una mayor vista a los alrededores y mediante el análisis de detección facial tiene un mayor rango de detección, específicamente hasta 42 grados gracias a su distancia de profundidad de 0.3 m, mientras que la cámara normal tiene límite hasta 45 grados.

Comparando diferentes sistemas de la empresa Hikvision con el prototipo en la prueba catorce dio que el prototipo compite e incluso supera en varios ámbitos a las cámaras de seguridad con reconocimiento facial en el mercado, puesto que aunque la cámara Hikvision tiene mejores características ópticas y una estructura sólida con dimensiones menores, el prototipo admite mayores cuadros por segundo específicamente 90, ofrece una distancia de profundidad de 0.3 m, tiene un movimiento de 360 grados, tiene un costo menor, funciona en software libre y tiene un porcentaje de coincidencia semejante.

El sistema es capaz de crear bases de datos de manera local o mediante descarga de la base de datos de google, entrena una red neuronal convolucional, puede realizar el reconocimiento facial dentro de imágenes, vídeos pre-grabados o en tiempo real con una precisión mayor al 85 %.

## **12. Recomendaciones**

Para el correcto funcionamiento de los programas se debe tomar en consideración la ubicación de estos que deben estar en la misma carpeta.

Al momento de la ejecución de los diferentes programas, es de suma importancia la actualización de la información dentro de estos como por ejemplo directorios o nombres según se requiera.

En cada programa se debe respetar el formato de los archivos en el caso que existan, especialmente en el caso de la red neuronal convolucional que no se debe modificar su terminación (.pickle).

Para una tarjeta gráfica de la marca NVIDIA ® se debe verificar que tengan compatibilidad para aplicaciones CUDA el cual permite acelerar procesos. En la página web de



NVIDIA developer se puede encontrar el listado.

Si se tiene accesorios de uso permanente se recomienda que se ocupe al momento de la creación de la base de datos local.

Las imágenes que conforman la base de datos, las imágenes y vídeos pre-grabados que se deseen analizar, se recomienda tener la misma o mejor calidad a la de la mínima de la cámara, en el caso del proyecto la cámara de profundidad tiene una resolución de 720p (1280 x 720 píxeles).

## BIBLIOGRAFÍA

- [1] Las claves de los sistemas de reconocimiento facial: ¿cuál es su verdadero nivel de seguridad? [En línea]. Disponible: <https://www.xataka.com/seguridad/las-claves-de-los-sistemas-de-reconocimiento-facial-cual-es-su-verdadero-nivel-de-seguridad> [Fecha de consulta: Enero 2020]
- [2] M. G. Galván Robles, “Neuronas artificiales y neuromarketing, lo nuevo en la investigación de mercados,” *Congreso Internacional de Contaduría, Administración e Informática*, vol. 1, no. 1, p. 9, Octubre 2018.
- [3] Nabucco center » redes neuronales. [En línea]. Disponible: <http://nabucco.es/redes-neuronales/> [Fecha de consulta: Enero 2020]
- [4] Deep learning, python, c y azure. [En línea]. Disponible: <https://www.slideshare.net/ManuelRodrigoCabello/deep-learning-python-c-y-azure> [Fecha de consulta: Enero 2020]
- [5] Red neuronal convolucional cnn - diego calvo. [En línea]. Disponible: <http://www.diegocalvo.es/red-neuronal-convolucional/> [Fecha de consulta: Enero 2020]
- [6] Building a convolutional neural network for image classification with tensorflow. [En línea]. Disponible: <https://towardsdatascience.com/building-a-convolutional-neural-network-for-image-classification-with-tensorflow-f1f2f56bd83b> [Fecha de consulta: Enero 2020]
- [7] Tipos o clases de cámaras de seguridad según su forma. [En línea]. Disponible: <http://www.neotec.com.co/tienda/4-videovigilancia-cctv> [Fecha de consulta: Abril 2020]
- [8] Products intel realsense. [En línea]. Disponible: <https://store.intelrealsense.com/products.html> [Fecha de consulta: Abril 2020]
- [9] Servomotores. [En línea]. Disponible: <https://www.areatecnologia.com/electricidad/servomotor.html> [Fecha de consulta: Abril 2020]

- [10] Tipos de servomotores que existen actualmente. [En línea]. Disponible: <https://www.fabricantes-maquinaria-industrial.es/tipos-de-servomotores/> [Fecha de consulta: Abril 2020]
- [11] Onu habitat - violencia e inseguridad en las ciudades. [En línea]. Disponible: <https://onuhabitat.org.mx/index.php/violencia-en-inseguridad-en-las-ciudades> [Fecha de consulta: Noviembre 2018]
- [12] El comercio - la estadística señala menos delitos, pero la inseguridad no cede. [En línea]. Disponible: <https://www.elcomercio.com/actualidad/estadistica-delitos-inseguridad-delincuencia-policia.html> [Fecha de consulta: Noviembre 2020]
- [13] Oacnudh - violencia e inseguridad. [En línea]. Disponible: <http://www.oacnudh.org/areas-de-trabajo/seguridad-ciudadana/> [Fecha de consulta: Noviembre 2018]
- [14]J. Tapia Pérez, “La inseguridad pública: causas y consecuencias,” *El Cotidiano [en línea]*, vol. 1, no. 180, pp. 103–112, julio 2013.
- [15]V. Lio, “Ciudades, cÁmaras de seguridad y video-vigilancia: Estado del arte y perspectivas de investigaciÓn,” *Astrolabio*, vol. 1, no. 15, pp. 274–276, julio 2015.
- [16]M. V. Rull, “Reconocimiento de objetos usando deep learning,” Proy. pregrado, Escuela Técnica Superior de Ingeniería Universidad de Sevilla, Sevilla, España, 2016.
- [17]M. F. M. Alba, “Fundamentos de deep learning y una aplicación para predecir la demanda de energía eléctrica en México,” Proy. pregrado, Instituto Tecnológico Autónomo de México, Ciudad de México, México, 2019.
- [18]A. Rodriguez, “Desarrollo de una aplicación de reconocimiento en imágenes utilizando deep learning con opencv,” Proy. pregrado, Universidad Politécnica de Valencia, Valencia, España, 2018.
- [19]J. S. S. Dominguez, *Reconocimiento facial robusto a oclusiones por medio de seccionamiento de imágenes*, 2010.

- [20]Reconocimiento facial - kimaldi. [En línea]. Disponible: [https://www.kimaldi.com/blog/biometria/reconocimiento\\_facial/](https://www.kimaldi.com/blog/biometria/reconocimiento_facial/) [Fecha de consulta: Junio 2019]
- [21]H. M. Dominguez y F. Sáez Vacas, *Domótica: Un enfoque sociotécnico*, 1st ed. Fundación Rogelio Segovia - España, 2006.
- [22]N. J. Nilsson, "Principles of Artificial Intelligence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 1, pp. 7–18, 1981.
- [23]Machine learning: Tres cosas que es necesario saber. [En línea]. Disponible: <https://la.mathworks.com/discovery/machine-learning.html> [Fecha de consulta: Junio 2019]
- [24]Introducción al machine learning con bigml. [En línea]. Disponible: <https://www.slideshare.net/AldoRamiro/introduccion-al-machine-learning-con-bigml> [Fecha de consulta: Junio 2019]
- [25]C. A. Gutiérrez, "Rdeep learning aplicado al diagnóstico de soldadura mediante espectroscopía Óptica de plasmas," Proy. pregrado, Universidad de Cantabria, Santander, España, 2017.
- [26]¿qué es el deep learning y para qué sirve? [En línea]. Disponible: <https://www.indracompany.com/es/blogneo/deep-learning-sirve> [Fecha de consulta: Noviembre 2016]
- [27]Ibm knowledge center. [En línea]. Disponible: [https://www.ibm.com/support/knowledgecenter/es/SS3RA7\\_sub/modeler\\_mainhelp\\_client\\_ddita/components/neuralnet/neuralnet\\_model.html](https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html) [Fecha de consulta: Junio 2019]
- [28]Prototipo de una red neuronal artificial con arquitectura dinámica basada en el proceso de neurogénesis de redes neuronales maduras. [En línea]. Disponible: <https://roanboc.wordpress.com/2016/11/20/prototipo-de-una-red-neuronal-artificial-con-arquitectura-dinamica-basada-en-el-proceso-de-neurogenesis-de-redes-neuronales-maduras/> [Fecha de consulta: Junio 2019]

- [29]A. Rosebrock, *Deep Learning for Computer Vision with Python: Starter Bundle*. PyImageSearch, 2017.
- [30]Redes neuronales convolucionales. [En línea]. Disponible: <https://la.mathworks.com/solutions/deep-learning/convolutional-neural-network.html> [Fecha de consulta: Junio 2019]
- [31]M. Arsenovic, S. Sladojevic, A. Anderla, y D. Stefanovic, “FaceTime - Deep learning based face recognition attendance system,” *SISY 2017 - IEEE 15th International Symposium on Intelligent Systems and Informatics, Proceedings*, pp. 53–57, 2017.
- [32]X. Qu, T. Wei, C. Peng, y P. Du, “A Fast Face Recognition System Based on Deep Learning,” *Proceedings - 2018 11th International Symposium on Computational Intelligence and Design, ISCID 2018*, vol. 1, pp. 289–292, 2018.
- [33]Materiales de impresión 3d (ii): Abs (acrilonitrilo butadieno estireno). [En línea]. Disponible: <http://hxx.es/2015/03/23/materiales-de-impresion-3d-ii-abs-acrilonitrilo-butadieno-estireno/> [Fecha de consulta: Abril 2020]
- [34]El policarbonato. [En línea]. Disponible: <http://www.acrilico-y-policarbonato.com/policarbonato.html> [Fecha de consulta: Abril 2020]
- [35]Material properties of abs - acrylonitrile-butadlene-styrene. [En línea]. Disponible: <https://dielectricmfg.com/knowledge-base/abs/> [Fecha de consulta: Abril 2020]
- [36]J. V. Molino Osejos, “Caracterización de Materiales Termoplásticos de ABS y PLA semi-rígido impresos en 3D con cinco mallados internos diferentes,” pp. 38–111, 2016.
- [37]R. Bott, *Diseño en Ingeniería Mecánica de Shigley*, 2014.
- [38] *Mujeres y hombres en México 2012*, 2012.
- [39]Cámara domo con movimiento y zoom interior. [En línea]. Disponible: <http://www.superinventos.com/S134400.htm> [Fecha de consulta: Abril 2020]
- [40]Servo motor tower pro mg995 12kgf.cm engrenagem de metal 180 graus importado. [En línea]. Disponible: <https://www.moduloeletronica.com.br/SERVO-MOTOR-TOWER-PRO-MG995-IMPORTADO> [Fecha de consulta: Abril 2020]

- [41]Tabla de torques. [En línea]. Disponible: <http://www.redica.com.co/tabla-de-torques.html> [Fecha de consulta: Abril 2020]
- [42]El comercio - detección facial se suma a videovigilancia. [En línea]. Disponible: <https://www.elcomercio.com/actualidad/quito-camaras-videovigilancia-deteccion-facial.html> [Fecha de consulta: Julio 2019]
- [43]El comercio - 78 cámaras de reconocimiento facial fueron instaladas en el centro histórico de Quito; 13 de ellas solo en la marín. [En línea]. Disponible: <https://www.elcomercio.com/actualidad/camaras-reconocimiento-facial-quito-marin.html> [Fecha de consulta: Diciembre 2019]
- [44]S.-i. K. Renjie Wu, “A JOINTLY LOCAL STRUCTURED SPARSE DEEP LEARNING NETWORK FOR FACE Graduate School of Information , Production and Systems , Waseda University , Japan.”
- [45]S. Contreras y F. De la Rosa, “Aplicación de deep learning en robótica móvil para exploración y reconocimiento de objetos basados en imágenes,” pp. 1–8, 2016.
- [46]J. L. Galárraga, “Clasificador de hojas mediante Deep Learning,” 2017.
- [47]A. Şengür, Z. Akhtar, Y. Akbulut, S. Ekici, y Ü. Budak, “Deep Feature Extraction for Face Liveness Detection,” *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*, pp. 5–8, 2019.
- [48]A. Almadhor, “Deep Learning Based Face Detection Algorithm for Mobile Applications,” *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2018-October, no. October, pp. 1158–1162, 2019.
- [49]Y. Akbulut, A. Şengür, Ü. Budak, y S. Ekici, “Deep learning based face liveness detection in videos,” *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, pp. 6–9, 2017.
- [50]J. Zeng, J. Zeng, y X. Qiu, “Deep learning based forensic face verification in videos,” *Proceedings of 2017 International Conference on Progress in Informatics and Computing, PIC 2017*, pp. 77–80, 2017.

- [51]X. Ye, X. Chen, H. Chen, Y. Gu, y Q. Lv, "Deep learning network for face detection," *International Conference on Communication Technology Proceedings, ICCT*, vol. 2016-February, pp. 504–509, 2016.
- [52]S. M. Islam, H. Mahmood, A. A. Al-Jumaily, y S. Claxton, "Deep learning of facial depth maps for obstructive sleep apnea prediction," *Proceedings - International Conference on Machine Learning and Data Engineering, iCMLDE 2018*, pp. 154–157, 2019.
- [53]K. Shailaja y B. Anuradha, "Effective face recognition using deep learning based linear discriminant classification," *2016 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2016*, 2017.
- [54]P. Méndez y J. Ibarra, "Implementación de una red neuronal de convolución para el reconocimiento de poses en imágenes de rostros," *ACI Avances en Ciencias e Ingenierías*, vol. 6, no. 2, pp. 1–6, 2014.
- [55]O. A. Aghdam y H. K. Ekenel, "Robust deep learning features for face recognition under mismatched conditions," *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018*, pp. 1–4, 2018.
- [56]D. Kim, B. Comandur, H. Medeiros, N. M. Elfiky, y A. C. Kak, "Multi-view face recognition from single RGBD models of the faces," *Computer Vision and Image Understanding*, vol. 160, pp. 114–132, 2017. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2017.04.008>
- [57]A. Sharma, M. A. Haj, J. Choi, L. S. Davis, y D. W. Jacobs, "Robust pose invariant face recognition using coupled latent space discriminant analysis," *Computer Vision and Image Understanding*, vol. 116, no. 11, pp. 1095–1110, 2012. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2012.08.001>
- [58]E. G. Ortiz y B. C. Becker, "Face recognition for web-scale datasets," *Computer Vision and Image Understanding*, vol. 118, pp. 153–170, 2014. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2013.09.004>

- [59]V. K. Pothos, C. Theoharatos, y G. Economou, "A local spectral distribution approach to face recognition," *Computer Vision and Image Understanding*, vol. 116, no. 6, pp. 663–675, 2012. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2012.01.006>
- [60]J. Shi, A. Samal, y D. Marx, "How effective are landmarks and their geometry for face recognition?" *Computer Vision and Image Understanding*, vol. 102, no. 2, pp. 117–133, 2006.
- [61]R. Gopalan y D. Jacobs, "Comparing and combining lighting insensitive approaches for face recognition," *Computer Vision and Image Understanding*, vol. 114, no. 1, pp. 135–145, 2010. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2009.07.005>
- [62]T. C. Faltemier, K. W. Bowyer, y P. J. Flynn, "Using multi-instance enrollment to improve performance of 3D face recognition," *Computer Vision and Image Understanding*, vol. 112, no. 2, pp. 114–125, 2008. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2008.01.004>
- [63]J. R. Beveridge, G. H. Givens, P. J. Phillips, y B. A. Draper, "Factors that influence algorithm performance in the Face Recognition Grand Challenge," *Computer Vision and Image Understanding*, vol. 113, no. 6, pp. 750–762, 2009. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2008.12.007>
- [64]C. Peng, N. Wang, J. Li, y X. Gao, "DLFace: Deep local descriptor for cross-modality face recognition," *Pattern Recognition*, vol. 90, pp. 161–171, 2019. [En línea]. Disponible: <https://doi.org/10.1016/j.patcog.2019.01.041>
- [65]A. Sokhandan y A. Monadjemi, "Visual tracking in video sequences based on biologically inspired mechanisms," *Computer Vision and Image Understanding*, 2018. [En línea]. Disponible: <https://doi.org/10.1016/j.cviu.2018.10.002>
- [66]X. Luo, Y. Xu, y J. Yang, "Multi-resolution dictionary learning for face recognition," *Pattern Recognition*, vol. 93, pp. 283–292, 2019. [En línea]. Disponible: <https://doi.org/10.1016/j.patcog.2019.04.027>



- [67]H. Hu, "ICA-based neighborhood preserving analysis for face recognition," *Computer Vision and Image Understanding*, vol. 112, no. 3, pp. 286–295, 2008. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2008.05.003>
- [68]N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali, y M. Zareapoor, "Hybrid deep neural networks for face emotion recognition," *Pattern Recognition Letters*, vol. 115, pp. 101–106, 2018. [En línea]. Disponible: <https://doi.org/10.1016/j.patrec.2018.04.010>
- [69]D. Smeets, J. Keustermans, D. Vandermeulen, y P. Suetens, "MeshSIFT: Local surface features for 3D face recognition under expression variations and partial data," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 158–169, 2013. [En línea]. Disponible: <http://dx.doi.org/10.1016/j.cviu.2012.10.002>
- [70]S. Bhattacharya, G. S. Nainala, S. Rooj, y A. Routray, "Local Force Pattern (LFP): Descriptor for Heterogeneous Face Recognition," *Pattern Recognition Letters*, vol. 125, pp. 63–70, 2019. [En línea]. Disponible: <https://doi.org/10.1016/j.patrec.2019.03.028>

## Anexo A: Trabajos Relacionados

**Tabla 1.** Análisis de Sistemas Existentes

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2018	2018 11th International Symposium on Computational Intelligence and Design (ISCID)	A Fast Face Recognition System Based On Deep Learning (Xiujie Qu, Tianbo Wei, Cheng Peng, Peng Du, 2018)	FPGA	Basado en el principio de la red neuronal de convolución (CNN), se propuso un método de reconocimiento facial en tiempo real en FPGA, que mejora la velocidad y la precisión del reconocimiento facial. El método se divide en dos partes. Primero, el terminal de PC se utiliza para completar la formación y obtener los parámetros de la red. El segundo es el sistema de reconocimiento facial es construido en el FPGA.	Los resultados de la prueba mostraron que la velocidad de reconocimiento del sistema ha alcanzado los 400 fps, superando con creces los resultados existentes. La tasa de reconocimiento es del 99,25 %, superior a la del ojo humano. Además, posee una buena robustez para la luz del ambiente. [32]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2016	2016 IEEE International Conference on Image Processing (ICIP)	A Jointly Local Structured Sparse Deep Learning Network for Face Recognition (Renjie Wu, Sei-Ichiro Kamata, 2016)	SDLN	En este documento se propuso un modelo optimizado de Red de aprendizaje profundo (SDLN) para el reconocimiento facial. El SDLN propuesto está afinado conjuntamente para optimizar la tarea del reconocimiento facial. El SDLN logra un alto rendimiento en la base de datos ORL y FERET.	El método SDLN extrae y reconoce las características visuales locales mejorando el rendimiento del reconocimiento facial comparado con los métodos NN, SVM, SRC, AGL, ESRC, ESRC-KSVD y SVDL. Sin embargo, la muestra del entrenamiento no es suficiente y causa que el diccionario creado no esté completo. [44]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2016	2016 IEEE 11th Colombian Computing Conference (CCC)	Aplicación de Deep Learning en Robótica Móvil para Exploración y Reconocimiento de Objetos basados en Imágenes (Stevenson Contreras, Fernando De la Rosa, 2016)	Red neuronal genérica parametrizable	En este trabajo se busca mejorar la exploración de un robot móvil en espacios interiores con la aplicación de Deep Learning implementando un sistema de red neuronal genérica parametrizable, para poder crear varias instancias de red adaptables al tipo de información e integrarlos a un robot móvil.	Se obtuvo como resultados diferentes maneras y datos a la definición de dos redes neuronales: una de reconocimiento de imágenes con un alto nivel de confianza 98 % y otra de reconocimiento de situaciones de proximidad a obstáculos con un nivel de confianza aceptable de 89 %. [45]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2017	Escuela Técnica Superior de Ingenieros Informáticos UPM	Clasificador de hojas mediante Deep Learning (Jose Luis Galarraga, 2017)	CNN Alex-Net	En este trabajo se describe la generación de un clasificador de enfermedades en las plantas basado en imágenes, que utiliza un método basado en aprendizaje profundo como algoritmo de clasificación en una red neuronal convolucional	Utilizando CNN se obtiene un clasificador capaz de clasificar correctamente 2041 imágenes de un total de 2053 ejemplos pre-clasificados por un experto. [46]
2018	2018 International Conference on Artificial Intelligence and Data Processing (IDAP)	Deep Feature Extraction for Face Liveness Detection (Abdulkadir Şengür , Zahid Akhtar, Yaman Akbulut , Sami Ekici , Ümit Budak,2018)	CNN Alex-Net VGG16	Este estudio explora diferentes características profundas y las compara en un terreno común para la detección facial en los vídeos utilizando dos bases de datos NUAA y CASIA-FASD.	El mejor resultado obtenido es dado en la base de datos NUAA con una capa fc6 AlexNet concatenada con una precisión del 88.09 %. [47]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2018	TENCON 2018 - 2018 IEEE Region 10 Conferen- ce	Deep Lear- ning Based Face Detec- tion Algorithm for Mobile Applications (Ahmad Al- madhor, 2018)	LBP Haar Cas- cade SSD	Este artículo pro- pone un algorit- mo de detección de rostros basa- do en el aprendi- zaje profundo para aplicaciones móvi- les con Fddb da- taset.	Los resultados obtenidos son que LBP con clasifica- dor Haar Cascade fue capaz de detectar caras frontales; sin em- bargo, fracasó en detectar múltiples caras, diferentes orientaciones, y caras ocluidas. Correspondien- temente, LBP entrenado en el patrón SSD fue capaz de detectar rostros parciales, pero fallaron en detectar múltiples caras. Mientras SSD funcionó bien, pero dio fal- sos positivos [48]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2017	2017 International Artificial Intelligence and Data Processing Symposium (IDAP)	Deep Learning based Face Liveness Detection in Videos (Yaman Akbulut, Abdulkadir Sengur, Umit Budak, Sami Ekici, 2017)	LRF-ELM CNN	En este documento se propone la detección de rostros mediante aprendizaje profundo con dos modelos diferentes CNN y LRF-ELM con las bases de datos NUAA y CASIA. LRF-ELM es un modelo desarrollado recientemente que contiene una convolución y una capa de agrupación antes de una capa completamente conectada que hace que el modelo sea rápido. CNN, sin embargo, contiene una serie de capas de convolución y agrupación.	El modelo LRF-ELM produce valores de mayor precisión para ambas bases de datos. Para la base de datos NUAA, el modelo LRF-ELM obtiene una tasa de clasificación correcta de 84.04 %, el modelo CNN alcanza el 76.31 %. En otras palabras, el modelo LRF-ELM produce casi un 8 % de resultados más precisos. Para la base de datos CASIO, el modelo LRF-ELM produce casi un 6 % de resultados más precisos que para el modelo CNN. [49]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2017	2017 International Conference on Progress in Informatics and Computing (PIC)	Deep learning based forensic face verification in videos (Jinhua Zeng, Jinfeng Zeng, Xiulian Qiu, 2017)	Winner-take-all	En este documento se estudia la representación de rostros basado en el aprendizaje profundo para la verificación forense de imágenes humanas. Se discutieron sus estrategias de aplicación y limitaciones técnicas. Se propuso la estrategia de "winner-take-all" en el caso de la identificación forense de imágenes humanas en videos.	Los resultados de la estrategia propuesta fueron 0.427 y 0.472 en las secuencias de vídeo indicando la cara conocida y la cara cuestionada demostrando la viabilidad y utilidad del sistema. [50]



Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2015	2015 IEEE 16th International Conference on Communication Technology (ICCT)	Deep Learning Network for Face Detection (Xueyi Ye, Xueting Chen, Huahua Chen, Yafeng Gu, Qiuyun Lv, 2015)	multi-layer nonlinear mapping semantic feature extraction	Por multi-layer nonlinear mapping y semantic feature extraction del aprendizaje profundo se propone una red de aprendizaje profundo para la detección facial y superar el reto de detectar rostros de manera precisa y rápida en un caso no ideal.	Como resultado se obtuvo que el método propuesto tiene una alta precisión con rostros con una pequeña rotación 94.8 % pero igualmente para rostros con una gran rotación 85.4 %. Así que podemos concluir que el método de este artículo tiene una fuerte robustez frente a la rotación. [51]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2018	2018 International Conference on Machine Learning and Data Engineering (iCMLDE)	Deep Learning of Facial Depth Maps for Obstructive Sleep Apnea Prediction (Syed MS Islam, Hassan Mahmood, Adel Ali, Al-Jumaily, Scott Claxton, 2018)	CNN VGG Face fc7 PAMs-Alex fc7 PAMs-VGG19	En este documento, se investigó la aplicación de técnicas de aprendizaje profundo para diagnosticar la enfermedad a través del mapa de profundidad de las exploraciones faciales humanas. El mapa de profundidad proporcionará más información sobre la morfología facial en comparación con la imagen en color liso en 2-D.	Como resultado se obtuvo que la mejor precisión fue con VGG Face fc7 con una precisión de 68.75 % y 67.42 %. PAMs-VGG19 es más profundo que VGG Face pero no funciona bien. La ventaja de VGGface es que está pre-entrenado sobre una gran conjunto de datos [52]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2016	2016 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)	Effective Face Recognition using Deep Learning based Linear Discriminant Classification (K. Shailaja, Dr. B. Anuradha, 2016)	LDRC	En este documento, se introduce el método de Aprendizaje Profundo como parte de la estrategia basada en el aprendizaje para proporcionar un análisis completo sobre las muestras de rostros presentes en el sistema. Se propuso realizar un seguimiento de la información histórica sobre los rostros de entrada con una clasificación de regresión discriminante lineal con la base de datos YALE y ORL.	Los resultados se obtuvieron de manera gráfica del método LDRC en cada base de datos y se obtuvo que tiene una alta precisión en ambas si se aplica con un aprendizaje profundo. [53]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2017	2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)	FaceTime – Deep Learning Based Face Recognition Attendance System (Marko Arsenovic, Srdjan Sladojevic, Andras Anderla, Darko Stefanovic, 2017)	CNN CNN casca- de	En este paper se propone un Sistema de reconocimiento facial mediante dos técnicas de aprendizaje profundo CNN cascade y CNN para su empleo práctico.	En este artículo se obtuvo el resultado de 95.02 % es decir una precisión alta mediante el método propuesto con un sistema de asistencia y la combinación de métodos de aprendizaje profundo. [31]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2014	ACI Avances en Ciencias e Ingenierías	Implementación de una red neuronal de convolución para el reconocimiento de poses en imágenes de rostros (Paul Méndez, Julio Ibarra, 2014)	RNPC Ros- tros Propios	El Paper explora la utilización de las redes neuronales de convolución en el reconocimiento de imágenes de poses horizontales fuera del plano de rostros. Se propone una implementación basada en las bibliotecas de código abierto OpenCV para la clasificación de imágenes de rostros humanos dentro de 7 poses pre-determinadas y se presenta en detalle la arquitectura de la red y la estrategia de aprendizaje.	La implementación entrenada con conjuntos de 2600 imágenes, alcanza una tasa de aciertos promedio del 85 % superior a la obtenida con el algoritmo de Rostros Propios cercana al 78 %, con un tiempo de ejecución similar. [54]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2016	Dep. Ingeniería de Sistemas y Automática Escuela Técnica Superior de Ingeniería Universidad de Sevilla	Reconocimiento de Objetos usando Deep Learning (Marina Vázquez Rull, 2016)	NMS	En este proyecto se trata el tema de aprendizaje de máquina mediante la visión artificial. Concretamente, se plantea el problema de localización y reconocimiento de objetos de distinta clase en una imagen, con el propósito de que sirvan como información para un sistema autónomo. Para conseguir resolver dicho problema, se hace uso del aprendizaje profundo, y en particular de las redes neuronales convolucionales.	En este trabajo se ha presentado un método concreto de resolución del problema de detección y clasificación de objetos en imágenes, el cual pertenece a la rama de estudio Aprendizaje Profundo y que se denominan Redes Neuronales Convolucionales. Con una aplicación concreta de ellas se ha dado resultado a problemas concretos de detección en casos con un único objeto a clasificar y en otros con imágenes en las que resulta interesante distinguir varios objetos diferentes y sus distintas localizaciones. [16]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2018	2018 26th Signal Processing and Communications Applications Conference (SIU)	Robust Deep Learning Features for Face Recognition under Mismatched Conditions (Omid Abdollahi Aghdam, Hazım Kemal Ekenel, 2018)	CNN VGG Face2	En este documento, se aborda el problema del reconocimiento facial en condiciones no coincidentes. Se propone el modelo VGG Face2 para la representación de rostros el cual es un modelo de redes neuronales convolutivas pre-entrenadas para extraer 2048 vectores de características dimensionales de imágenes de caras del ICB-RW 2016.	El método propuesto supera significativamente a las tasas de identificación de Rango 1, Rango 5, y al Área bajo la curva de puntaje acumulativo del mejor enfoque en el desafío ICB-RW 2016, que fue de 69.8 %, 85.3 % y 0.954, respectivamente [55]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2017	Magazine Computer Vision and Image Un- derstanding Volume 160	Multi-view fa- ce recognition from single RGBD models of the faces	Deep convo- lutional neural net- works.	En este documen- to se supone que cada individuo en una población pueda ser mode- lado por una sola imagen de rostro RGBD frontal, es posible llevar a cabo el reconoci- miento de rostro para tal población utilizando múl- tiples imágenes 2D capturadas de puntos de vista arbitrarios me- diante enfoques jerárquicos.	El método pro- puesto que se denotó como VGG-NET no supera los enfo- ques PSS y CVS comparados con otros tres dataset cuando el número de imágenes de consulta es mayor a 7. [56]



Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2012	Magazine Computer Vision and Image Un- derstanding Volume 116 Issue 11	Robust pose invariant face recognition using coupled latent space discriminant analysis	Discri- minant Mul- tiple Coupled Latent Sub- pace frame- work	En este documen- to se abarca el problema del reco- nocimiento facial mediante técnicas de pose invarian- te. Se propone un nuevo enfoque lla- mado subespacio latente acoplado múltiple discrimi- nante. Encuentra los conjuntos de direcciones de proyección para diferentes poses de tal manera que las imágenes proyectadas del mismo sujeto en diferentes poses se correlacionan al máximo en el espacio latente.	El documento tuvo como resultado que usar más de dos poses simul- táneamente con CCA resulta en un mejor rendimiento. También se obtuvo resultados de vanguardia para el reconocimiento facial invariable de pose en PIE CMU y FERET y resultados compa- rables en MultiPIE cuando se usan solo cuatro puntos fiduciales para las características de alineación e intensidad. [57]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2014	Magazine Computer Vision and Image Un- derstanding Volume 118	Face recogni- tion for web- scale datasets	LASRC	En este docu- mento se abarca el problema del reconocimiento facial en las pá- ginas web. Se propone un nuevo algoritmo de Cla- sificación basada en representación dispersa lineal- mente aproximada (LASRC).	El algoritmo pro- puesto logro un rendimiento com- parable al SRC con una velocidad de 100 a 250 veces y exhibe un recuerdo similar a los SVM con un entrenamien- to mucho más rápido. [58]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2012	Magazine Computer Vision and Image Un- derstanding Volume 116 Issue 6	A local spectral distribution approach to face recognition	Face's local spectral distribution	En este documento se abarca el problema del reconocimiento facial mediante el uso total de la distribución espectral de la estructura facial. Su funcionamiento se basa en la construcción del árbol de expansión mínima, que es una herramienta eficaz para preservar y utilizar la estructura múltiple del conjunto de datos. La nueva representación es a la vez holística, considerando la distribución de las características como un todo, mientras que al mismo tiempo utiliza la extracción de información local.	Los resultados experimentales en cuatro bases de datos faciales de referencia demuestran las propiedades favorables de la metodología propuesta sobre los enfoques tradicionales, particularmente en el caso de imagen única". [59]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2006	Magazine Computer Vision and Image Un- derstanding Volume 102 Issue 2	How effective are landmarks and their geo- metry for face recognition?	Landmark- based Pro- crustes Geo- metry	En este docu- mento se abarca el problema del reconocimiento facial mediante puntos de referen- cia biológicamente significativos y su geometría extraída de las imágenes faciales.	.Los resultados experimentales muestran que nuestra distancia EWC supera las normas I1 y I2 e incluso la distancia de Mahalanobis. Nuestro estudio también encuentra que los puntos de referencia y su enfoque basado en la geometría pueden explicar muy bien las va- riaciones de la expresión facial y el envejeci- miento. Por lo tanto, se pueden usar en modo independiente o en combina- ción con otros enfoques para reducir el espacio de búsqueda a priori. [60]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2010	Magazine Computer Vision and Image Un- derstanding Volume 114 Issue 1	Comparing and combi- ning lighting insensitive approaches for face recog- nition	Classifie algo- rithms	En este documen- to se abarca el problema del reco- nocimiento facial bajo condiciones de iluminación cambiantes. Se analiza las for- talezas relativas de diferentes re- presentaciones insensibles a la iluminación, y proponemos esquemas de combinación de clasificadores eficientes que resultan en me- jores tasas de reconocimiento.	Los resultados obtenidos fueron los siguientes:  1. Gradient di- rection funciona muy bien tanto en galerías ho- mogéneas como en configuracio- nes de galería heterogéneas.  2. Self-quotient funciona muy bien en galerías ho- mogéneas. 3. No todos los métodos que usan datos de entrenamiento funcionan mejor que los simples.  4. Different re- presentations funcionan bien en diferentes partes de la cara. 5. Es posible mejorar el rendimiento con combinacio- nes. [61]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2008	Magazine Computer Vision and Image Un- derstanding Volume 112 Issue 2	Using multi- instance enrollment to improve performance of 3D face recognition	3D sha- pe ICP	En este documen- to se abarca el uso de la inscrip- ción en varias ins- tancias como un medio para mejo- rar el rendimiento del reconocimien- to facial.	Los resultados de este docu- mento indican que el enfoque de inscripción de múltiples ins- tancias supera a un enfoque de reconocimiento basado en compo- nentes de última generación, en el que la cara a reconocer se considera como un conjunto in- dependiente de regiones. [62]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2009	Magazine Computer Vision and Image Un- derstanding Volume 113 Issue 6	Factors that influence algorithm performance in the Face Recognition Grand Cha- llenge	The cova- riates and statis- tical model GLMM	En este documen- to se presenta un estudio estadístico que cuantifica los efectos de las co- variables como el género, la edad, la expresión, la reso- lución de la ima- gen y se centra en tres algoritmos de reconocimien- to facial. Específi- camente, se utili- za un modelo de efecto mixto lineal generalizado para relacionar la pro- babilidad de verifi- cación con las co- variables de sujeto e imagen.	Los resultados de este docu- mento sugieren posibles nuevas direcciones de investigación. Una dirección es la fusión de algo- ritmos basada en covariables. Otra dirección podría abordar la causalidad. El estudio actual informa cambios observados debi- do a covariables; sin embargo, el análisis no intenta explicar la causa del efecto. Res- ponder la causa subyacente de los efectos ayudará a diseñar algorit- mos robustos de reconocimiento facial. [63]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2019	Magazine Pattern Recognition Volume 90	DLFace: Deep local descriptor for cross-modality face recognition	Deep local descriptor learning framework	<p>En este documento se propone un deep local descriptor learning framework para el reconocimiento facial de modalidad cruzada, que tiene como objetivo aprender información local discriminante y compacta directamente de parches faciales. Se propone una nueva pérdida de enumeración de modalidad cruzada, que luego se integra en redes neuronales convolucionales para la extracción profunda de descriptores locales.</p>	<p>En este documento se tuvo como resultado un descriptor local profundo para el reconocimiento de rostros entre modalidades. También una red EnumerateNet integrando una función de pérdida de enumeración para generar local profundo descriptor de parches de imágenes locales. Finalmente el método propuesto del Deep local descriptor es fácil de usar en el reconocimiento facial tradicional y se proporciona el rendimiento en el algoritmo Fisherface superando métodos modernos. [64]</p>



Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2018	Magazine Computer Vision and Image Un- derstanding	Visual trac- king in video sequences based on biologically inspired me- chanisms	BIT model	En este documen- to se abarca el problema del se- guimiento visual. En este documen- to, teniendo en cuenta el rendi- miento superior de la visión biológica en el seguimiento visual, se introdu- ce un algoritmo de seguimiento visual inspirado biológicamente.	Los resultados de los experimentos realizados en el algoritmo, así como la compa- ración con otros algoritmos de seguimiento co- nocidos, indicaron que el algorit- mo propuesto se ha enfren- tado bien a los desafíos que en- frenta el problema de seguimiento en términos de precisión y robus- tez y tiempo de ejecución. [65]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2019	Magazine Pattern Recognition Volume 93	Multi-resolution dictionary learning for face recognition	multi-resolution dictionary learning	En este documento se abarca el problema del reconocimiento facial mediante el estudio del aprendizaje de diccionarios. Se propone un método de aprendizaje de diccionarios de resolución múltiple que proporciona múltiples diccionarios cada uno asociado con una resolución.	Los resultados del documento indican que el método del diccionario no son muy adecuadas para el caso real donde las muestras tienen diferentes resoluciones. En comparación con otros algoritmos de aprendizaje de diccionario el método proporciona diccionarios asociados y agrega una restricción relativamente fuerte para mantener la similitud de las representaciones obtenidas usando diferentes diccionarios en la fase de entrenamiento y no ser tan sensible al ruido. [66]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2008	Magazine Computer Vision and Image Un- derstanding Volume 112 Issue 3	ICA-based neighborhood preserving analysis for fa- ce recognition	ICA NPA IC-NPA	En este documen- to se presenta un Nuevo méto- do de reducción de dimensio- nalidad para el reconocimiento facial, que se llama análisis de preservación de vecindad basado en componentes independientes (IC-NPA). IC-NPA muestra más po- der discriminante que los métodos tradicionales del subespacio cuan- do se trata de las variaciones resultantes de los cambios en la iluminación, la expresión facial y la pose.	Los resultados del método pro- puesto IC-NPA muestra un mejor rendimiento que el método popular, como el méto- do Eigenface, el método ICA, el método basado en LDA y el método Laplacianface. [67]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2018	Magazine Pattern Recognition Letters Volume 115	Hybrid deep neural net- works for face emotion recognition	CNN Hybrid Convolut Recurrent Neural Net- work	En este docu- mento se propone una red neuronal recurrente de con- volución híbrida para el recono- cimiento facial. La arquitectura de red propuesta consiste en capas de convolución seguidas por la re- d neuronal recu- rrente (RNN) que el modelo com- binado extrae las relaciones dentro de las imágenes faciales y al usar la red recurrente las dependen- cias temporales que existen en las imágenes pueden conside- rarse durante la clasificación.	Los resultados de este método híbrido propuesto evaluado en base a dos conjuntos de datos públicos son prometedores en comparación con los métodos más modernos. [68]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2013	Magazine Computer Vision and Image Un- derstanding Volume 117 Issue 2	meshSIFT: Local surface features for 3D face recog- nition under expression variations and partial data	meshSIFT	En este documen- to se abarca el reconocimiento fa- cial con el algorit- mo meshSIFT. Es- te algoritmo s ro- busto para varia- ciones de expre- sión, datos faltan- tes y valores atípi- cos.	Los resultados del documento son: 1. El número de características meshSIFT coin- cidentes es una medida confiable para el reconoci- miento de rostros invariantes de expresión 2. La simetría de los descriptores de características permite comparar dos superficies faciales 3D con superposición limitada o nu- la. 3. El uso de meshSIFT ayuda también para la normalización de pose y estima- ción del plano de simetría. [69]

Año	Publicación	Nombre de Investigación	Método	Explicación	Resultados
2019	Magazine Pattern Recognition Letters Volume 125	Local Force Pattern (LFP): Descriptor for Heteroge- neous Face Recognition	LFP	Este documen- to abarca el problema del reconocimiento facial mediante el método Local Force Pattern LFP. Las características de LFP codifican la información direccional de las texturas de las imágenes faciales dadas de forma compacta, produciendo un código más dis- criminatorio que los métodos de vanguardia.	Los resultados de este documento muestran que el descriptor pro- puesto funciona de manera con- sistente sobre los métodos más modernos para di- ferentes conjuntos de datos de caras heterogéneos como CUSF, IIIT-D y CASIA. [70]

## Anexo B: Análisis de Alternativas

Para el análisis de alternativas se va a ocupar la matriz de decisión de Pugh, tomando en consideración diferentes criterios de evaluación según cada temática pero en cada uno de estos se referenciara a un peso específico el cual se presenta en la Tabla 2.

**Tabla 2.** Asignación de Valor de los Pesos

0	No cumple
1	Podría utilizarse
2	Cumple lo mínimo
3	Es necesario
4	Cumple a cabalidad
5	Cumple satisfactoriamente

### B.1 Estructura de Cámaras de Seguridad

Los criterios de evaluación que se utilizan son:

- **Instalación:** La estructura debe ser de fácil instalación.
- **Costo:** El costo debe ser accesible o competitivo de \$100.
- **Localización:** El prototipo es de uso en interiores.
- **Tamaño:** La cámara de seguridad no debe afectar en la privacidad por lo que debe ser de un tamaño aceptable aproximadamente de 30 cm.
- **Diseño:** Debido a que es de uso en interiores debe ser agradable a la vista.
- **Movimiento:** La estructura debe ser capaz de moverse en 360 grados.
- **Mantenimiento:** El sistema debe permitir un mantenimiento fácil e incluso rápido.

En la Tabla 3 se presenta la matriz de Pugh para seleccionar la estructura de la Cámara de Seguridad.

**Tabla 3.** Evaluación de la estructura de la cámara de seguridad

	Peso	Domo	Cubo	Robóticas	Camuflada	Bala	PTZ
Instalación	2	2	2	0	0	2	0
Costo	4	4	4	-4	0	-4	-4
Localización	5	5	5	5	5	-5	5
Tamaño	2	0	-2	0	0	2	2
Diseño	4	4	0	4	-4	0	4
Movimiento	5	0	-5	5	-5	0	5
Mantenimiento	2	-2	2	-2	-2	2	-2

Suma Positivos	15	13	14	5	6	16
Suma Negativos	2	7	6	11	9	6
Suma General	13	6	8	-6	-3	10

## B.2 Cámara de Profundidad

Los criterios de evaluación que se utilizan son:

- **Dimensiones:** La cámara debe tener un tamaño competitivo respecto a cámaras de seguridad. La cámara de profundidad debe estar dentro de una cámara de tipo domo por lo que debe ser de un tamaño menor a 15 cm.
- **Costo:** El costo debe ser accesible o competitivo alrededor de los \$150.
- **Resolución:** La calidad de imagen debe ser mínimo 720p.
- **Estructura:** Tiene que tener un diseño agradable a la vista, de fácil manipulación y con protección.
- **Conexión:** Conector a computadora USB.
- **Localización:** El prototipo es de uso en interiores.
- **Procesador:** La cámara debe permitir su control sin programas externos.

En la Tabla 4 se presenta la matriz de Pugh para seleccionar la Cámara de Profundidad del catálogo de Intel®.

**Tabla 4.** Evaluación de la cámara de profundidad

	Peso	T265	D435i	D435	D415
Dimensiones	5	5	5	5	5
Costo	4	4	-4	-4	4
Resolución	4	-4	4	4	4
Estructura	4	-4	4	4	4
Conexión	2	2	2	2	2
Localización	4	4	4	4	4
Procesador	3	3	0	0	3

Suma Positivos	18	19	19	26
Suma Negativos	8	4	4	0
Suma General	10	15	15	26



### B.3 Tipo de Motor

Los criterios de evaluación que se utilizan son:

- **Dimensiones:** El tamaño del motor debe ser adecuado para el prototipo. Debe haber disponibilidad de un tamaño aproximado de 5x2x5 cm.
- **Costo:** El costo debe ser accesible o competitivo cercano a \$20.
- **Accesibilidad:** El producto no debe ser industrial.
- **Mantenimiento:** El motor debe permitir un mantenimiento fácil e incluso rápido.
- **Torque:** La característica mecánica del torque debe satisfacer la necesidad del prototipo. Torques medios.
- **Control:** EL motor tiene que ser de un manejo fácil (para todo público) y un control adecuado según la experiencia que se tenga.

En la Tabla 5 se presenta la matriz de Pugh para seleccionar el tipo de motor a utilizar en el prototipo.

**Tabla 5.** Evaluación del tipo de motor

	Peso	CC	AC	Imanes Permanentes	Paso a Paso
Dimensiones	5	5	0	-5	5
Costo	4	4	0	-4	4
Accesibilidad	3	3	3	0	3
Mantenimiento	3	3	0	0	0
Torque	4	4	4	4	4
Control	2	2	0	-2	2

Suma Positivos	21	7	4	18
Suma Negativos	0	0	11	0
Suma General	21	7	-7	18

### B.4 Materiales de Cámaras de Seguridad

Los criterios de evaluación que se utilizan son:

- **Costo:** El costo debe ser accesible o competitivo, cercano a los \$50 tomando en cuenta la materia prima y los medios que necesita el material para la construcción del prototipo (impresión 3D o termoformado).

- **Módulo de Young:** La característica mecánica de la elasticidad debe satisfacer la necesidad del prototipo, considerando las cámaras de seguridad en el mercado para el uso interno, aproximadamente de 2 GPa.
- **Resistencia a la Flexión:** La característica mecánica de la flexión debe satisfacer la necesidad del prototipo, considerando las cámaras de seguridad en el mercado para el uso interno, aproximadamente de 80 MPa.
- **Resistencia al Impacto:** El material debe ser capaz de soportar golpes o caídas desde una distancia mínima de 2 metros.
- **Resistencia la Tracción:** La característica mecánica de la tracción debe satisfacer la necesidad del prototipo, considerando las cámaras de seguridad en el mercado para el uso interno, aproximadamente de 30 MPa.
- **Temperatura deflexión de calor:** La característica térmica de deflexión debe satisfacer la necesidad del prototipo, considerando las cámaras de seguridad en el mercado para el uso interno debe soportar una temperatura mínima de 27 grados Celsius ya que es la temperatura máxima recomendada por el Reglamento de Instalaciones Térmicas en Edificios (RIT) para un lugar de trabajo.
- **Diseño:** EL material debe ser adecuado para una cámara de seguridad para el uso en interiores y agradable a la vista.
- **Mantenimiento:** El material debe permitir un manejo y un mantenimiento fácil.

En la 65 se presenta la matriz de Pugh para seleccionar el material de la estructura del prototipo.

**Tabla 6.** Evaluación del material de la estructura de la cámara de seguridad

	<b>Peso</b>	<b>ABS</b>	<b>Policarbonato</b>
Costo	4	4	4
Módulo de Young	5	5	5
Resistencia a la Flexión	5	5	5
Resistencia al Impacto	4	4	4
Resistencia a la Tracción	5	5	5
Temperatura de Flexión	3	3	3
Diseño	3	3	0
Mantenimiento	3	3	-3

Suma Positivos	32	26
Suma Negativos	0	3
Suma General	32	23

## B.5 Servomotores

Los criterios de evaluación que se utilizan son:

- **Dimensiones:** El tamaño del motor debe ser adecuado para el prototipo. Debe tener un tamaño aproximado de 5x2x5 cm.
- **Costo:** El costo debe ser accesible o competitivo, cercano a \$20.
- **Consumo:** El motor debe funcionar con una alimentación de 5V y un máximo de 250 mA de corriente.
- **Materiales:** El mecanismo debe ser realizado mediante engranajes metálicos.
- **Torque:** La característica mecánica del torque debe satisfacer la necesidad del prototipo diseñado, debe ser mayor a 1 Nm.
- **Velocidad:** El motor debe tener una velocidad de giro aceptable, es decir aproximadamente de 0.20 s / 60 grados.

En la Tabla 7 se presenta la matriz de Pugh para seleccionar el servomotor que se encargara del movimiento 360 grados del sistema.

**Tabla 7.** Evaluación Servomotores

	Peso	MG995	43R SM-4315R	DS04-NFC	Futaba S3003
Dimensiones	5	5	5	5	5
Costo	3	3	3	0	3
Consumo	4	4	4	4	4
Materiales	3	3	3	3	3
Torque	5	5	-5	5	-5
Velocidad	2	2	0	0	2

Suma Positivos	22	15	14	17
Suma Negativos	0	0	8	5
Suma General	22	15	6	12

## B.6 Microcontrolador

Los criterios de evaluación que se utilizan son:

- **Dimensiones:** El tamaño del controlador debe ser adecuado para el prototipo. Debe tener un tamaño menor a 7x5x5 cm.
- **Costo:** El costo debe ser accesible o competitivo, máximo de \$20.
- **Conexiones:** La placa debe tener un conector hembra para la alimentación y disponibilidad para la conexión de los motores.
- **Consumo:** El prototipo debe funcionar solo con la alimentación de la computadora (USB 5V 500mA).
- **Pines:** El microcontrolador dispone del número de pines necesarios para la conexión de los motores.
- **Software:** La programación del controlador es con software libre y se encuentra dentro de la experiencia.

En la Tabla 8 se presenta la matriz de Pugh para seleccionar el microcontrolador responsable del control de la cámara de seguridad.

**Tabla 8.** Evaluación Microcontroladores

	Peso	Arduino UNO	Arduino Nano	chipKIT Uno32	NodeMCU ESP8266
Dimensiones	3	3	3	-3	3
Costo	3	3	3	0	3
Conexiones	5	5	0	5	0
Consumo	5	5	5	5	5
Pines	4	4	4	4	4
Software	3	3	3	3	3

Suma Positivos	23	18	17	18
Suma Negativos	0	0	3	0
Suma General	23	18	14	18

## Anexo C: Algoritmo de Control

### C.1 Creación de Base de Datos mediante descarga de imágenes de Google

```
#PAQUETE PARA CONDICIONES HTTP
    from requests import exceptions
#LIBRERÍA REQUESTS HTTP
    import requests
#LIBRERÍA OPEN CV
    import cv2
#LIBRERÍA PARA CREACIÓN DE CARPETAS
    import os
#LIBRERÍA TIEMPO
    import time
#INICIAR UN CRONOMETRO
    start_time = time.time()
#Microsoft Cognitive Services API KEY PARA BUSCAR EN LA BASE DE DATOS DE
GOOGLE
    API_KEY = "b83152ad574842a1acab3ab6f11ba549"
#NUMERO DE BÚSQUEDAS
    MAX_RESULTS = 10
#NUMERO DE GRUPO DE RESULTADOS
    GROUP_SIZE = 5
#API URL
    URL = "https://api.cognitive.microsoft.com/bing/v7.0/images/search"
#EXCEPCIONES DE POSIBLES ERRORES HTTP FILTRAR IMÁGENES
#FileNotFoundError ES PARA VERSIONES PYTHON 3 EN ADELANTE SE DEBE
PONER CASO MANUAL
    try:
        FileNotFoundError
```

```

except NameError:
    FileNotFoundError = IOError

EXCEPTIONS = set([IOError, FileNotFoundError, exceptions.RequestException,
exceptions.HTTPError, exceptions.ConnectionError, exceptions.Timeout])

#PATH BASE DE DATOS
directorio = "C:\Users\Sebastian Perez\Desktop\PRUEBA DEEP LEARNING\dataset"
#CREACIÓN DIRECTORIO / CARPETA CON SU RESPECTIVA VERIFICACIÓN
try:
    os.mkdir(directorio)
except OSError:
    #print("La creación del directorio %s falló" % directorio)
else:
    #print("Se ha creado el directorio: %s " % directorio)

PATH SUBDIRECTORIO BASE DE DATOS
directorio2 = "C:\Users\Sebastian Perez\Desktop\PRUEBA DEEP LEARNING\dataset\
INSERTE NOMBRE DE BUSQUEDA"
#CREACIÓN DIRECTORIO / CARPETA CON SU RESPECTIVA VERIFICACIÓN
try:
    os.mkdir(directorio2)
except OSError
    #print("La creación del directorio %s falló" % directorio)
else:
    #print("Se ha creado el directorio: %s " % directorio)

#TERMINO A SER BUSCADO
term = "INSERTE NOMBRE DE BUSQUEDA"
#PATH SUBDIRECTORIO
output = "dataset\INSERTE NOMBRE DE BUSQUEDA"
#PARÁMETROS DE BÚSQUEDA
headers = {"Ocp-Apim-Subscription-Key": API_KEY}
params = {"q": term, "offset": 0, "count": GROUP_SIZE}

```

```
#VERIFICACIÓN
```

```
print("BUSCANDO '{}'".format(term))
```

```
#BÚSQUEDA
```

```
search = requests.get(URL, headers=headers, params=params)
```

```
search.raise_for_status()
```

```
results = search.json()
```

```
#RESULTADOS DE BÚSQUEDA
```

```
estNumResults = min(results["totalEstimatedMatches"], MAX_RESULTS)
```

```
#VERIFICACIÓN
```

```
print("RESULTADOS OBTENIDOS '{}'".format(estNumResults, term))
```

```
#CONTADOR
```

```
total = 0
```

```
#BUCLE PARA OBTENER LAS IMÁGENES
```

```
for offset in range(0, estNumResults, GROUP_SIZE):
```

```
    #VERIFICACIÓN
```

```
        print("HACIENDO SOLICITUD DE GRUPO {}-{} DE {}..."
              .format(offset, offset +
                      GROUP_SIZE, estNumResults))
```

```
    #PARÁMETROS
```

```
        params["offset"] = offset
```

```
    #ACTUALIZACIÓN DE BÚSQUEDA
```

```
        search = requests.get(URL, headers=headers, params=params)
```

```
    #BÚSQUEDA
```

```
        search.raise_for_status()
```

```
    #RESULTADO DE BÚSQUEDA
```

```
        results = search.json()
```

```
    #VERIFICACIÓN
```

```
        print("GRABANDO IMAGEN DE GRUPO {}-{} DE {}..."
              .format(offset, offset +
                      GROUP_SIZE, estNumResults))
```

```
    #BUCLE DE RESULTADOS PARA DESCARGAR IMAGEN
```

```
        for v in results["value"]:
```

```

try:
    #VERIFICACIÓN
    print("SOLICITANDO: {}".format(v["contentUrl"]))
    #SOLICITUD DE DESCARGA DE IMAGEN
    r = requests.get(v["contentUrl"], timeout=30)
    #PATH DE IMAGEN
    ext = v["contentUrl"][v["contentUrl"].rfind("."): ]
    p = os.path.sep.join([output, "{}{}".format(str(total).zfill(8), ext)])
    #GRABADO DE IMAGEN EN DISCO
    f = open(p, "wb")
    f.write(r.content)
    f.close()
#EXCEPCIÓN ERROR AL DESCARGAR IMAGEN
except Exception as e:
    if type(e) in EXCEPTIONS:
        print("SALTAR: {}".format(v["contentUrl"]))
        continue
#LEER IMAGEN EN EL DISCO
image = cv2.imread(p)
#CONDICIÓN ERROR EN IMAGEN DESCARGADA
if image is None:
    print("ELIMINANDO: {}".format(p))
#ELIMINACION
os.remove(p)
continue
#ACTUALIZACION DE CONTADOR
total += 1
#SACAR EN PANTALLA EL TIEMPO TOTAL
print(" %s segundos" % (time.time() - starttime))

```

## C.2 Creación de Base de Datos Local con clasificador " Cascade"



```

#LIBRERÍA OPEN CV
import cv2

#LIBRERÍA PARA CREACIÓN DE CARPETAS
import os

#LIBRERÍA TIEMPO
import time

#INICIAR UN CRONOMETRO
start_time = time.time()

#ABRIR CAMARA
web_cam = cv2.VideoCapture(1)

#PATH CASCADE CARPETA
cascPath = "Cascades\haarcascade_frontalface_default.xml"

#PRIMER CLASIFICADOR CASCADE
faceCascade = cv2.CascadeClassifier(cascPath)

#CONTADOR
count = 0

#PATH BASE DE DATOS
directorio = "C:\Users\Sebastian Perez\Desktop\PRUEBA DEEP LEARNING\dataset"

#CREACIÓN DIRECTORIO / CARPETA CON SU RESPECTIVA VERIFICACIÓN
try:
    os.mkdir(directorio)
except OSError:
    #print("La creación del directorio %s falló" % directorio)
else:
    #print("Se ha creado el directorio: %s " % directorio)

PATH SUBDIRECTORIO BASE DE DATOS
directorio2 = "C:\Users\Sebastian Perez\Desktop\PRUEBA DEEP LEARNING\dataset\
INSERTE NOMBRE"

#CREACIÓN DIRECTORIO / CARPETA CON SU RESPECTIVA VERIFICACIÓN
try:

```

```

        os.mkdir(directorio2)
except OSError
    #print("La creación del directorio %s falló" % directorio)
else:
    #print("Se ha creado el directorio: %s " % directorio)
while(True):
    #LEER CAMARA
    _, imagen_marco = web_cam.read()
    #TRANSFORMACION IMAGEN A RGB
    rgb = cv2.cvtColor(imagen_marco, cv2.COLOR_BGR2RGB)
    #DETECCIÓN ROSTRO CASCADE
    rostro = faceCascade.detectMultiScale(rgb, 1.5, 5)
    for(x,y,w,h) in rostro:
        #REALIZACIÓN DE RECTÁNGULO
        cv2.rectangle(imagen_marco, (x,y), (x+w, y+h), (255,0,0), 4)
        #AUMENTO CONTADOR
        count += 1
        #GRABADO DEL ROSTRO
        cv2.imwrite("dataset/Sebastian/INSERTE NOMBRE_-str(count)+".jpg", rgb[y:y+h,
x:x+w])
        #MOSTRAR FRAME
        cv2.imshow("Creando dataset", imagen_marco)
    #CONDICION EMERGENTE DE CAMARA
    if cv2.waitKey(1) && 0xFF == ord('q'):
        break
    #CONDICIÓN CONTADOR
    elif count >= 100:
        break
#CONDICIÓN CIERRE CÁMARA
web_cam.release()

```

```
#SACAR EN PANTALLA EL TIEMPO TOTAL
print(" %s segundos" % (time.time() – start_time))
cv2.destroyAllWindows()
```

### **C.3 Creación de Base de Datos Local sin clasificador**

```
#LIBRERÍA OPEN CV
import cv2

#LIBRERÍA PARA CREACIÓN DE CARPETAS
import os

#LIBRERÍA TIEMPO
import time

#INICIAR UN CRONOMETRO
start_time = time.time()

#ABRIR CÁMARA
web_cam = cv2.VideoCapture(1)

#PARÁMETROS CÁMARA ANCHO
web_cam.set(3,1280)

#PARÁMETROS CÁMARA LARGO
web_cam.set(4,720)

#PARÁMETROS CÁMARA CUADROS POR SEGUNDO
web_cam.set(5,30)

#DELAY
time.sleep(2)

#PARÁMETROS CÁMARA EXPOSICIÓN
web_cam.set(15, –7.0)

#CONTADOR
count = 40
i = 1

#PATH BASE DE DATOS
directorio = "C:\Users\Sebastian Perez\Desktop\PRUEBA DEEP LEARNING\dataset"
```

```
#CREACIÓN DIRECTORIO / CARPETA CON SU RESPECTIVA VERIFICACIÓN
```

```
try:
    os.mkdir(directorio)
except OSError:
    #print("La creación del directorio %s falló" % directorio)
else:
    #print("Se ha creado el directorio: %s " % directorio)
```

```
PATH SUBDIRECTORIO BASE DE DATOS
```

```
directorio2 = "C:\Users\Sebastian Perez\Desktop\PRUEBA DEEP LEARNING\dataset\
INSERTE NOMBRE"
```

```
#CREACIÓN DIRECTORIO / CARPETA CON SU RESPECTIVA VERIFICACIÓN
```

```
try:
    os.mkdir(directorio2)
except OSError:
    #print("La creación del directorio %s falló" % directorio)
else:
    #print("Se ha creado el directorio: %s " % directorio)
```

```
#LEER CAMARA
```

```
ret, imagen_marco = web_cam.read()
```

```
#MOSTRAR FRAME
```

```
#cv2.imshow("Creando dataset", imagen_marco)
```

```
while i <= count :
```

```
    #GRABADO DEL ROSTRO
```

```
    cv2.imwrite("dataset/INSERTE NOMBRE/Mariela_-str(i)+".jpg", imagen_marco)
```

```
    i += 1
```

```
#CONDICIÓN CIERRE CÁMARA
```

```
web_cam.release()
```

```
#SACAR EN PANTALLA EL TIEMPO TOTAL
```

```
print(" %s segundos" % (time.time() – start_time))
```

```
cv2.destroyAllWindows()
```

## C.4 Entrenamiento de Red Neuronal Convolutacional

```

#PAQUETE PARA OPERACIONES DE PROCESAMIENTO DE IMÁGENES
from imutils import paths

import imutils

#PARTE DE LA LIBRERIA DONDE SE ENCUENTRA LA RED NEURONAL ENTRENA-
DA
import face_recognition

#LIBRERÍA PARA EL OBJETO CLASIFICADOR
import pickle

#LIBRERÍA OPEN CV
import cv2

#LIBRERÍA PARA CREACIÓN DE CARPETAS
import os

#LIBRERÍA TIEMPO
import time

#INICIAR UN CRONOMETRO
start_time = time.time()

#MÉTODO A SER OCUPADO REDES NEURONALES CONVOLUCIONALES O
HISTOGRAMA DE GRADIENTES ORIENTADOS hog
metodo = "cnn"

#BASE DE DATOS
dataset = "dataset"

#RED NEURONAL CONVOLUCIONAL
red = "RedNeuronalConvolutacional.pickle"

#VERIFICACIÓN
print("OBTENIENDO IMAGENES...")

#PATH IMÁGENES DE LA BASE DE DATOS
imagePaths = list(paths.list_images(dataset))

#LISTA CODIFICACIONES
knownEncodings = []

```

```

#LISTA NOMBRES
    knownNames = []
#BUCLE NUMERO DE IMÁGENES
for (i, imagePath) in enumerate(imagePaths):
    #VERIFICACION
        print("PROCESANDO IMAGEN {}/{}".format(i + 1, len(imagePaths)))
    #EXTRACCION NOMBRE IMAGEN
        name = imagePath.split(os.path.sep)[-2]
    #LEER IMAGEN
        image = cv2.imread(imagePath)
    #TRANSFORMACION A RGB
        rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        rgb = imutils.resize(rgb, width=1280)
    #DETECCION COORDENADAS ROSTRO
        boxes = face_recognition.face_locations(rgb, model=metodo)
    #CODIFICAR CARACTERISTICAS ROSTRO
        encodings = face_recognition.face_encodings(rgb, boxes)
    #ALMACENAMIENTO EN LA LISTA NOMBRE Y CODIFICACION PARA LA
    CLASIFICACION
        for encoding in encodings:
            knownEncodings.append(encoding)
            knownNames.append(name)
#VERIFICACION
    print("REALIZANDO RED NEURONAL CONVOLUCIONAL...")
#SACAR EN PANTALLA EL TIEMPO TOTAL
    print(" %s segundos" % (time.time() - start_time))
#ALMACENAMIENTO EN DISCO DE LA RED EN .PICKLE
    data = "encodings": knownEncodings, "names": knownNames
    f = open(red, "wb")
    f.write(pickle.dumps(data))

```

```
#CIERRE
```

```
f.close()
```

## C.5 Reconocimiento Facial de imágenes

```
#LIBRERÍA PARA OPERACIONES DE PROCESAMIENTO DE IMÁGENES
```

```
import imutils
```

```
#PARTE DE LA LIBRERÍA DONDE SE ENCUENTRA LA RED NEURONAL ENTRENADA
```

```
import face_recognition
```

```
#LIBRERÍA PARA EL OBJETO CLASIFICADOR
```

```
import pickle
```

```
#LIBRERÍA OPEN CV
```

```
import cv2
```

```
#LIBRERÍA TIEMPO
```

```
import time
```

```
#INICIAR UN CRONOMETRO
```

```
start_time = time.time()
```

```
#MÉTODO A SER OCUPADO REDES NEURONALES CONVOLUCIONALES O  
HISTOGRAMA DE GRADIENTES ORIENTADOS hog
```

```
metodo = "cnn"
```

```
#RED NEURONAL CONVOLUCIÓN
```

```
red = "RedNeuronalConvolutcional.pickle"
```

```
#IMAGEN A SER ANALIZADA
```

```
imagen = "INSERTE NOMBRE IMAGEN.jpg"
```

```
#NUMERO DE IMAGENES DE LA BASE DE DATOS
```

```
dataset = 40
```

```
#VERIFICACIÓN
```

```
print("CARGA DE RED NEURONAL CONVOLUCIONAL...")
```

```
#LEER RED NEURONAL CONVOLUCIONAL
```

```
data = pickle.loads(open(red, "rb").read())
```

```

#APERTURA DE IMAGEN
    image = cv2.imread(imagen)
    image = imutils.resize(image, width = 1280)
#TRANSFORMACIÓN IMAGEN A RGB
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
#VERIFICACIÓN
    print("DETECTANDO ROSTROS...")
#COORDENADAS ROSTRO
    boxes = face_recognition.face_locations(rgb,model=metodo)
#CODIFICACIÓN
    encodings = face_recognition.face_encodings(rgb, boxes)
#LISTA NOMBRES
    names = []
#LISTA PORCENTAJES
    porcentajes = []
#BUCLE CLASIFICADOR
    for encoding in encodings:
        #COMPARACION IMAGEN CON BASE DE DATOS
        matches = face_recognition.compare_faces(data["encodings"],encoding)
        #CASO QUE NO RESCONOZCA ROSTRO
        name = "Desconocido"
        porcentaje = "0.0"
        #SI EXISTIO COINCIDENCIA CREACION DICCIONARIO
        if True in matches:
            #ENCONTRAR LOS INDICES DE TODOS LOS ROSTROS COINCIDENTES
            matchedIdxs = [i for (i, b) in enumerate(matches) if b]
            #CONTADOR
            counts = {}
            #BUCLE EN CADA INIDICE
            for i in matchedIdxs:

```



```

#OBTENCIÓN NOMBRE ROSTRO COINCIDENTE
    name = data["names"][i]
#CONTADOR
    counts[name] = counts.get(name, 0) + 1
print(counts)
#CALCULO DE PORCENTAJE DE COINCIDENCIA
    porcentaje = str((float(float(counts[name]) / float(dataset))*100)
#DETERMINACIÓN ROSTRO SEGÚN MAYOR NUMERO DE
COINCIDENCIAS CON BASE DE DATOS
    name = max(counts, key=counts.get)
#ACTUALIZACIÓN LISTA NOMBRES
    names.append(name)
#ACTUALIZACIÓN LISTA PORCENTAJES
    porcentajes.append(porcentaje)
#COORDENADAS Y NOMBRES DE ROSTRO EN LA IMAGEN
for ((top, right, bottom, left), name, porcentaje) in zip(boxes, names, porcentajes):
    #TRAZO DE RECTÁNGULO DEL ROSTRO
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)
    y = top - 15 if top - 15 >15 else top + 15
    cv2.putText(image, name + + porcentaje + " %", (left-50, y),
    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
#SACAR EN PANTALLA EL TIEMPO TOTAL
    print(" %s segundos" % (time.time() - start_time))
#SACAR EN PANTALLA IMAGEN CON ROSTROS ANALIZADOS
    cv2.imshow('Imagen', image)
    cv2.imwrite('imagen_analizada.jpg', image)
#CONDICIÓN OPEN CV
    cv2.waitKey(0)

```

## C.6 Reconocimiento Facial de video

```

#LIBRERÍA PARA OPERACIONES DE PROCESAMIENTO DE IMÁGENES
import imutils

#PARTE DE LA LIBRERÍA DONDE SE ENCUENTRA LA RED NEURONAL ENTRENA-
DA
import face_recognition

#LIBRERÍA PARA EL OBJETO CLASIFICADOR
import pickle

#LIBRERÍA OPEN CV
import cv2

#LIBRERÍA TIEMPO
import time

#INICIAR UN CRONOMETRO
start_time = time.time()

#MÉTODO A SER OCUPADO REDES NEURONALES CONVOLUCIONES O
HISTOGRAMA DE GRADIENTES ORIENTADOS hog
metodo = "cnn"

#RED NEURONAL CONVOLUCIONAL
red = "RedNeuronalConvolutcional.pickle"

#VÍDEO A SER ANALIZADO
video = "INSERTE NOMBRE.mp4"

#NUMERO DE IMÁGENES DE LA BASE DE DATOS
dataset = 21

#VÍDEO RESULTADO
salida = "video_analizado.avi"

#VERIFICACIÓN
print("CARGA DE RED NEURONAL CONVOLUCIONAL...")

#LEER RED NEURONAL CONVOLUCIONAL
data = pickle.loads(open(red, "rb").read())

#VERIFICACIÓN
print("PROCESANDO VIDEO...")

```

```
#OBTENCIÓN DE VÍDEO
```

```
stream = cv2.VideoCapture(video)
```

```
writer = None
```

```
#BUCLE HASTA QUE FINALICE EL VÍDEO
```

```
while True:
```

```
    #LEER VÍDEO
```

```
    (grabbed, frame) = stream.read()
```

```
    #CONDICIÓN NO LECTURA DE VÍDEO
```

```
    if not grabbed:
```

```
        break
```

```
    #TRANSFORMACIÓN A RGB
```

```
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

```
    #TRANSFORMACIÓN A 720P ANCHO 1280 PÍXELES
```

```
    rgb = imutils.resize(frame, width=1280)
```

```
    #OBTENCIÓN FORMA DE IMAGEN RGB
```

```
    r = frame.shape[1] / float(rgb.shape[1])
```

```
    #COORDENADAS ROSTRO
```

```
    boxes = face_recognition.face_locations(rgb,model=metodo)
```

```
    #CODIFICACIÓN
```

```
    encodings = face_recognition.face_encodings(rgb, boxes)
```

```
    #LISTA NOMBRES
```

```
    names = []
```

```
    #LISTA PORCENTAJES
```

```
    porcentajes = []
```

```
    #BUCLE CLASIFICADOR
```

```
    for encoding in encodings:
```

```
        #COMPARACIÓN VÍDEO CON BASE DE DATOS
```

```
        matches = face_recognition.compare_faces(data["encodings"],encoding)
```

```
        #CASO QUE NO RECONOZCA ROSTRO
```

```
        name = "Desconocido"
```

```

porcentaje = "0.0"
#SI EXISTIÓ COINCIDENCIA CREACIÓN DICCIONARIO
if True in matches:
    #ENCONTRAR LOS ÍNDICES DE TODOS LOS ROSTROS
    COINCIDENTES
    matchedIdxs = [i for (i, b) in enumerate(matches) if b]
    #CONTADOR
    counts =
    for i in matchedIdxs:
        #OBTENCIÓN NOMBRE ROSTRO COINCIDENTE
        name = data["names"][i]
        #CONTADOR
        counts[name] = counts.get(name, 0) + 1
    #DETERMINACION ROSTRO SEGUN MAYOR NUMERO DE
    COINCIDENCIAS CON BASE DE DATOS
    name = max(counts, key=counts.get)
    #CALCULO DE PORCENTAJE DE COINCIDENCIA
    porcentaje = str((float(float(counts[name]) / float(dataset)))*100)
#ACTUALIZACIÓN LISTA NOMBRES
names.append(name)
#ACTUALIZACIÓN LISTA PORCENTAJES
porcentajes.append(porcentaje)
#COORDENADAS Y NOMBRES DE ROSTRO EN EL VÍDEO DE SALIDA
for ((top, right, bottom, left), name, porcentaje) in zip(boxes, names, porcentajes):
    top = int(top * r)
    right = int(right * r)
    bottom = int(bottom * r)
    left = int(left * r)
#TRAZO DE RECTÁNGULO DEL ROSTRO
cv2.rectangle(frame, (left, top), (right, bottom),(0, 255, 0), 2)

```

```

y = top - 15 if top - 15 > 15 else top + 15
cv2.putText(frame, name + " " + porcentaje + "%", (left, y),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

#PARA GRABAR RESULTADO

if writer is None and salida is not None:
    fourcc = cv2.VideoWriter_fourcc(* "MJPG" )
    writer = cv2.VideoWriter(salida, fourcc, 24,(frame.shape[1], frame.shape[0]),
True)

if writer is not None:
    writer.write(frame)

#SACAR EN PANTALLA EL VÍDEO
#cv2.imshow("Frame", frame)
#CONDICIÓN VÍDEO FALLE
key = cv2.waitKey(1) & 0xFF
if key == ord("q"):
    break

SACAR EN PANTALLA EL TIEMPO TOTAL
print(" %s segundos" % (time.time() - start_time))
#CERRAR VÍDEO
stream.release()

```

## C.7 Reconocimiento Facial en tiempo real

```

#PAQUETE PARA CARGAR CÁMARA
from imutils.video import VideoStream

#LIBRERÍA PARA OPERACIONES DE PROCESAMIENTO DE IMÁGENES
import imutils

#PARTE DE LA LIBRERÍA DONDE SE ENCUENTRA LA RED NEURONAL ENTRENA-
DA
import face_recognition

#LIBRERÍA PARA EL OBJETO CLASIFICADOR

```

```

import pickle

#LIBRERÍA OPEN CV

import cv2

#LIBRERÍA TIEMPO

import time

#INICIAR UN CRONOMETRO

start_time = time.time()

#MÉTODO A SER OCUPADO REDES NEURONALES CONVOLUCIONALES O
HISTOGRAMA DE GRADIENTES ORIENTADOS hog

metodo = "cnn"

#RED NEURONAL CONVOLUCIONAL

red = "RedNeuronalConvolutcional.pickle"

#NUMERO DE IMÁGENES DE LA BASE DE DATOS

dataset = 25

#VERIFICACIÓN

print("CARGA DE RED NEURONAL CONVOLUCIONAL...")

#LEER RED NEURONAL CONVOLUCIONAL

data = pickle.loads(open(red, "rb").read())

#VERIFICACIÓN

print("INICIANDO CAMARA...")

#APERTURA CÁMARA SELECCIONADA EN ESTE CASO LA DE PROFUNDIDAD

vs = VideoStream(src=2).start()

#DELAY 2 SEG

time.sleep(2.0)

#BUCLE INFINITO TIEMPO REAL

while True:

    #LEER CÁMARA

    frame = vs.read()

    #TRANSFORMACION A RGB

    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

```

```

#TRANSFORMACION A 720P ANCHO 1280 PÍXELES
    rgb = imutils.resize(frame, width=1280)
#OBTENCION FORMA DE IMAGEN RGB
    r = frame.shape[1] / float(rgb.shape[1])
#COORDENADAS ROSTRO
    boxes = face_recognition.face_locations(rgb,model=metodo)
#CODIFICACION
    encodings = face_recognition.face_encodings(rgb, boxes)
#LISTA NOMBRES
    names = []
#LISTA PORCENTAJES
    porcentajes = []
#BUCLE CLASIFICADOR
    for encoding in encodings:
        #COMPARACIÓN VÍDEO TIEMPO REAL CON BASE DE DATOS
            matches = face_recognition.compare_faces(data["encodings"],encoding)
        #CASO QUE NO RECONOZCA ROSTRO
            name = "Desconocido"
            porcentaje = "0.0"
        #SI EXISTIÓ COINCIDENCIA CREACIÓN DICCIONARIO
            if True in matches:
                #ENCONTRAR LOS ÍNDICES DE TODOS LOS ROSTROS
                COINCIDENTES
                    matchedIdxs = [i for (i, b) in enumerate(matches) if b]
                #CONTADOR
                    counts =
                    for i in matchedIdxs:
                        #OBTENCIÓN NOMBRE ROSTRO COINCIDENTE
                            name = data["names"][i]
                        #CONTADOR

```

```

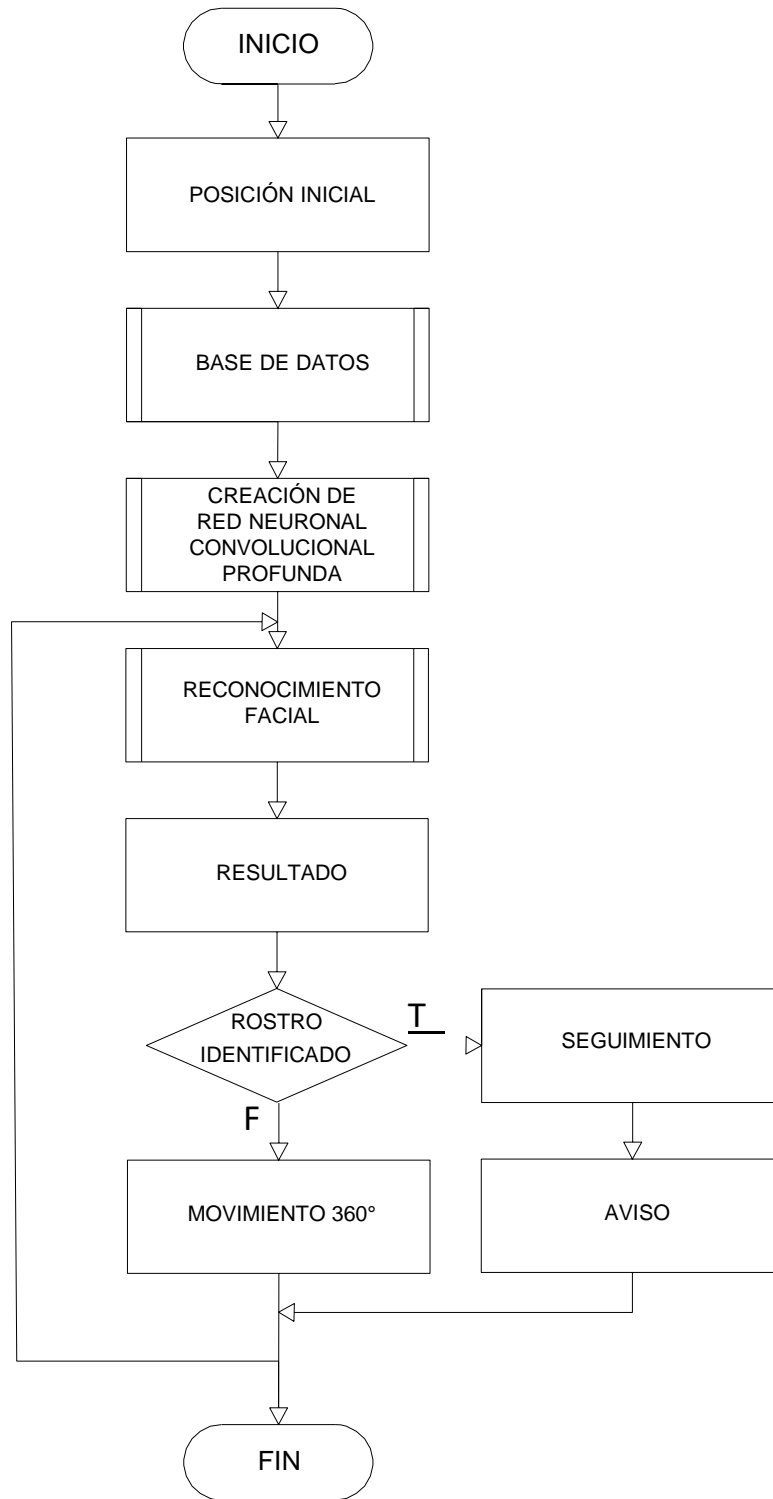
        counts[name] = counts.get(name, 0) + 1
#DETERMINACIÓN ROSTRO SEGÚN MAYOR NUMERO DE
COINCIDENCIAS CON BASE DE DATOS
    name = max(counts, key=counts.get)
#CALCULO DE PORCENTAJE DE COINCIDENCIA
    porcentaje = str((float(float(counts[name]) / float(dataset)))*100)
#ACTUALIZACIÓN LISTA NOMBRES
    names.append(name)
#ACTUALIZACIÓN LISTA PORCENTAJES
    porcentajes.append(porcentaje)
#COORDENADAS Y NOMBRES DE ROSTRO EN TIEMPO REAL
for ((top, right, bottom, left), name, porcentaje) in zip(boxes, names, porcentajes):
    top = int(top * r)
    right = int(right * r)
    bottom = int(bottom * r)
    left = int(left * r)
#TRAZO DE RECTÁNGULO DEL ROSTRO
    cv2.rectangle(frame, (left, top), (right, bottom),(0, 255, 0), 2)
    y = top - 15 if top - 15 >15 else top + 15
    cv2.putText(frame, name + + porcentaje + " %", (left, y),
        cv2.FONT_HERSHEY_SIMPLEX,0.5, (0, 255, 0), 2)
#SACAR EN PANTALLA LA CÁMARA
    cv2.imshow("Frame", frame)
#CONDICIÓN CÁMARA FALLE
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
SACAR EN PANTALLA EL TIEMPO TOTAL
    print(" %s segundos" % (time.time() - start_time))
#CERRAR VÍDEO




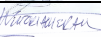
```

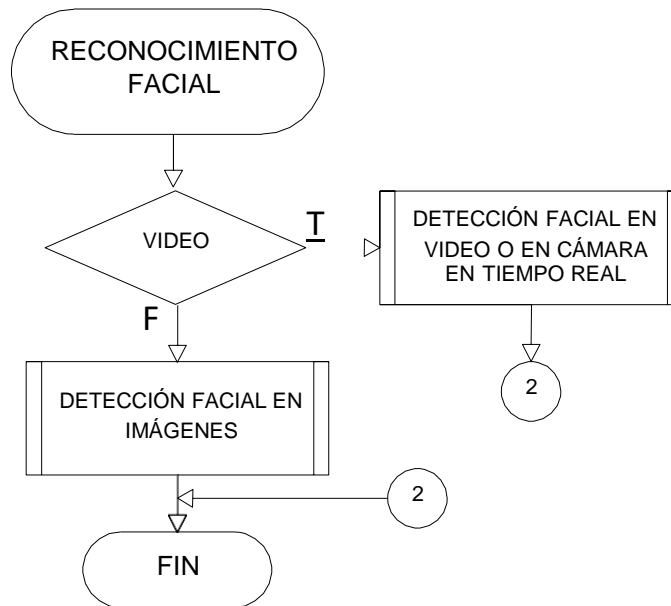
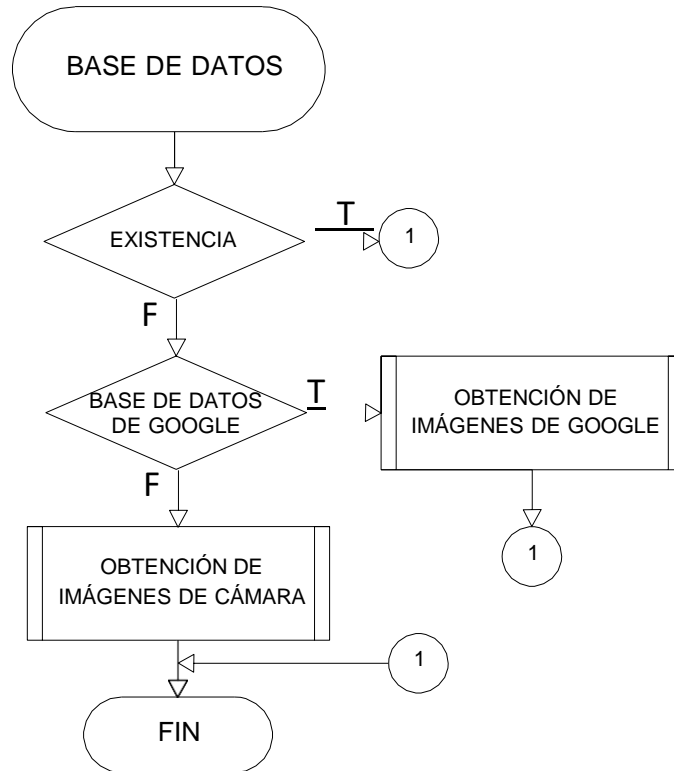






`stream.release()`

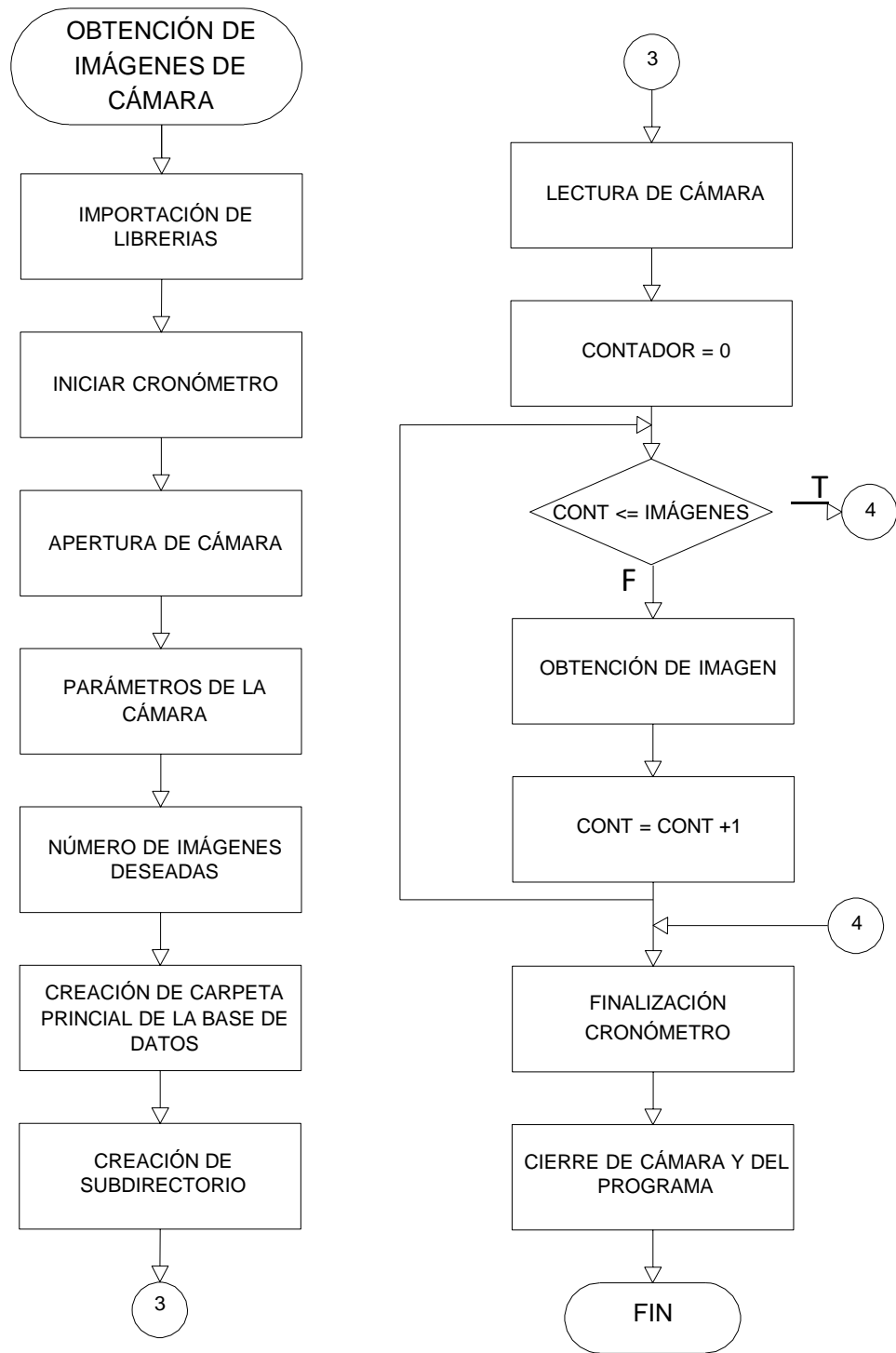
## **Anexo D: Planos de Construcción**



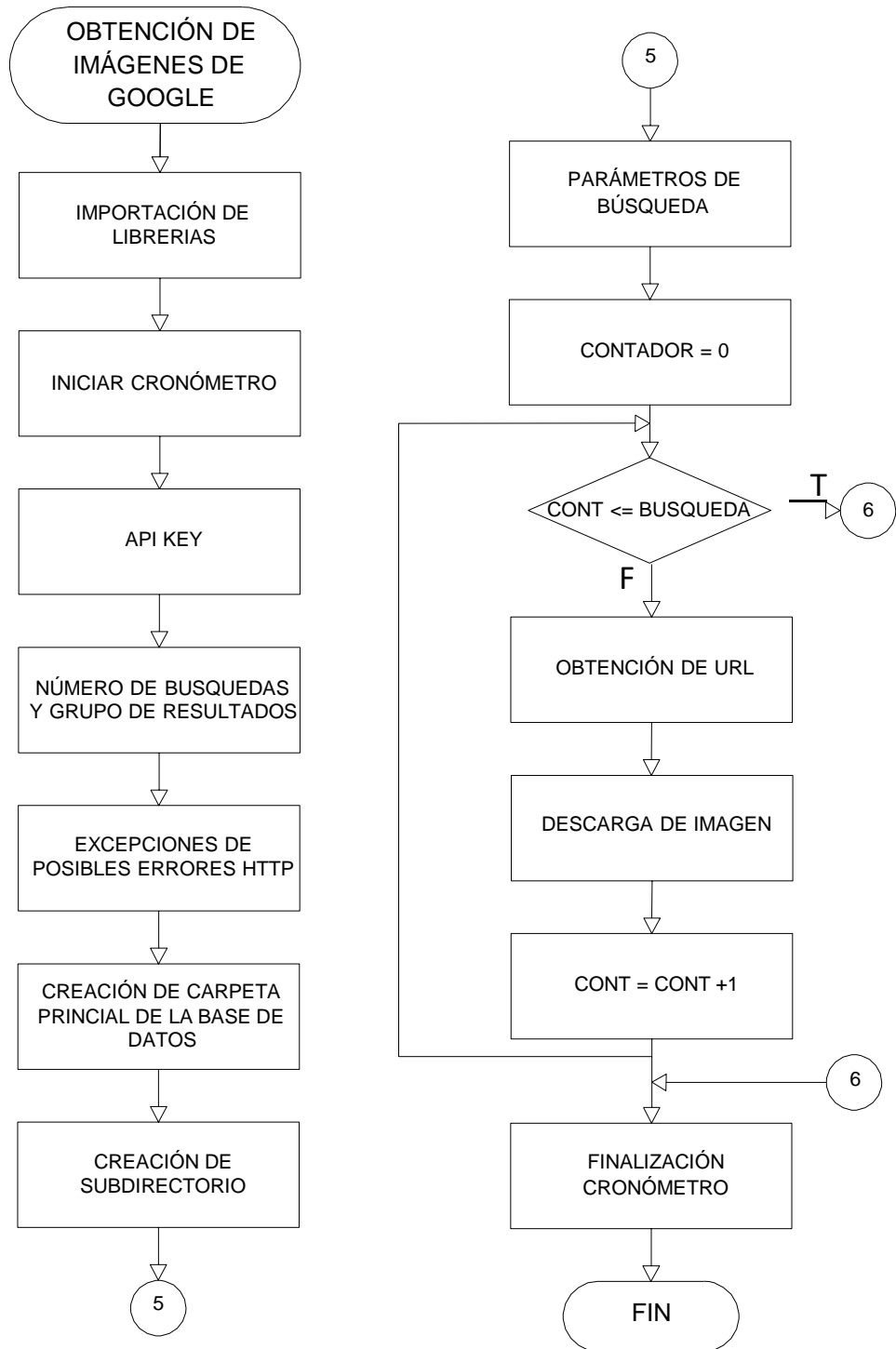
<b>UIDE</b>		<b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	20-10-2020	
			DIS.	PÉREZ S.	05-04-2020	
			REV.	MERA V.	17-12-2020	
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING - RUTINA PLANO INFORMÁTICO			<b>D01-001</b>		ESCALA N/A	
					N/A	







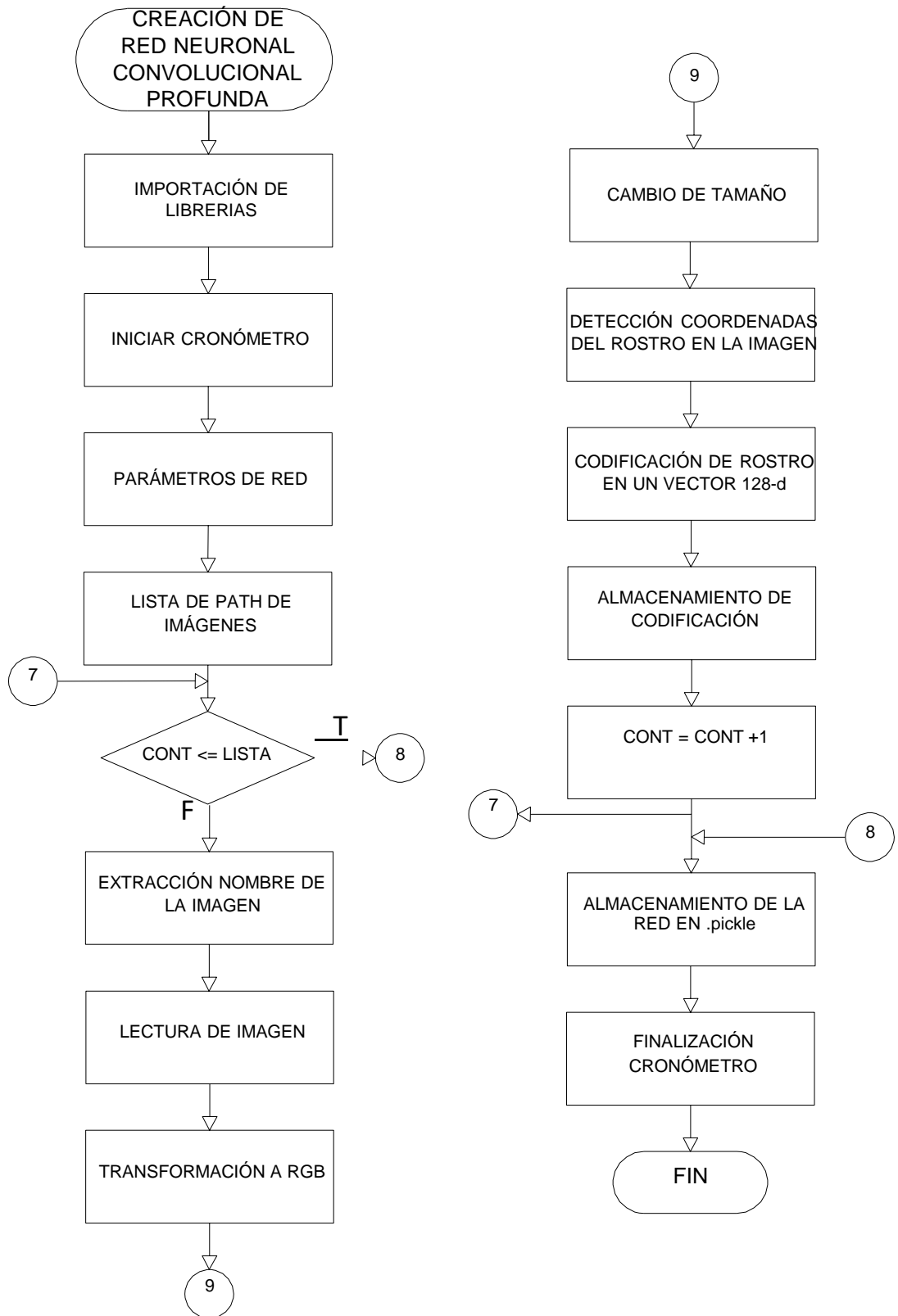
<b>UIDE</b>  <b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	20-10-2020	
	DIS.	PÉREZ S.	05-04-2020	
	REV.	MERA V.	17-12-2020	
<b>DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING - SUBROUTINAS PLANO INFORMÁTICO</b>		<b>D01-002</b>		ESCALA N/A
				N/A







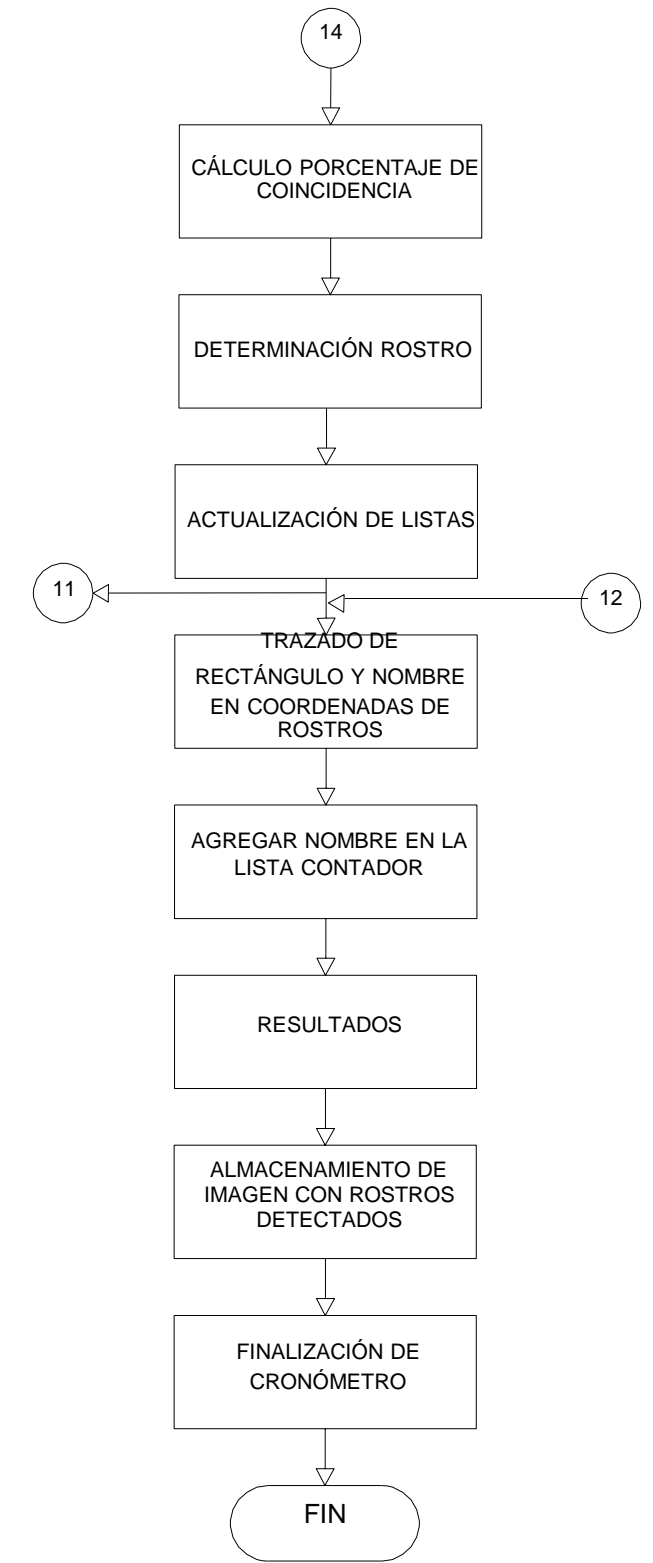
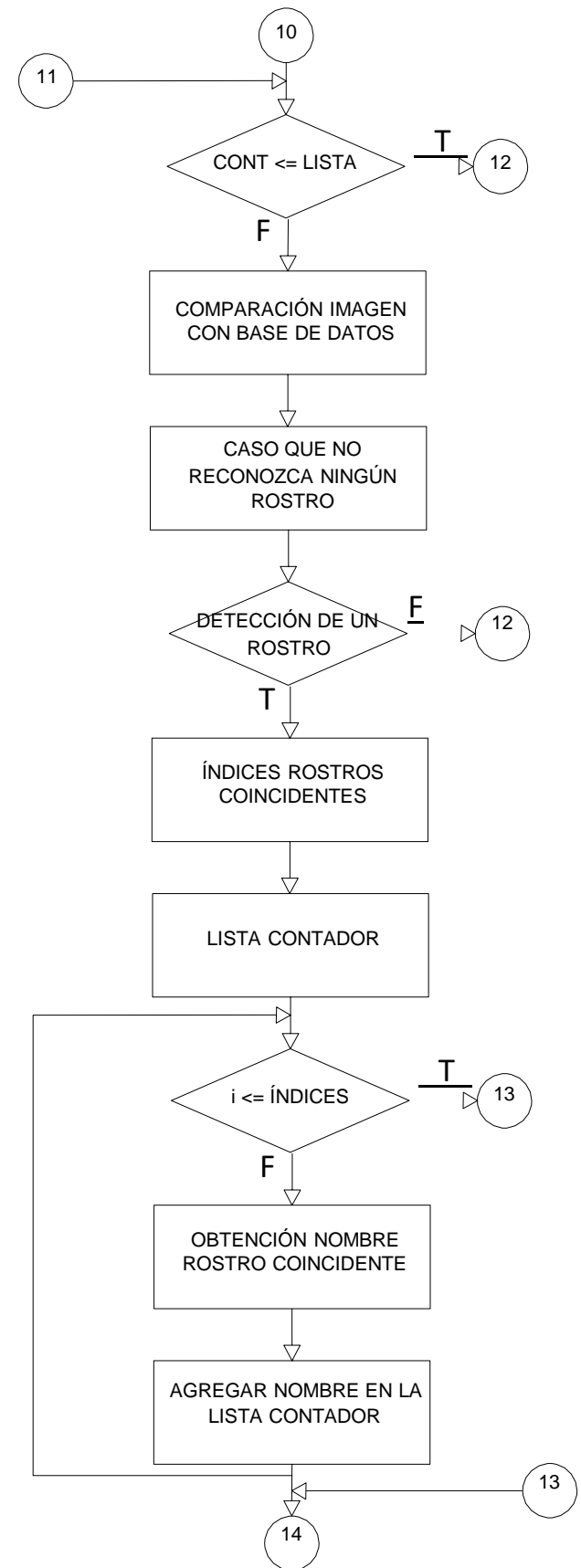
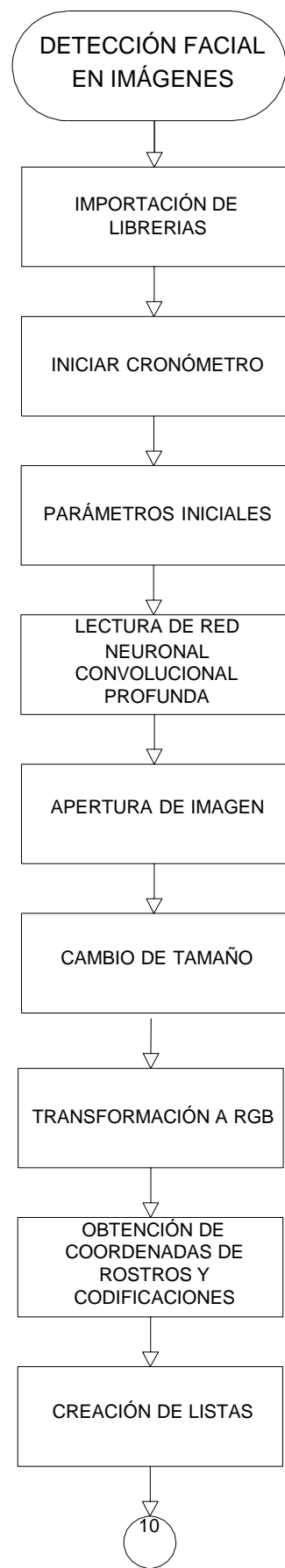
<b>UIDE</b>		<b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	20-10-2020	
			DIS.	PÉREZ S.	05-04-2020	
			REV.	MERA V.	17-12-2020	
<b>DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING - SUBRUTINAS PLANO INFORMÁTICO</b>			<b>D01-003</b>		ESCALA	N/A
					N/A	



UIDE	 <b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	20-10-2020	
		DIS.	PÉREZ S.	05-04-2020	
		REV.	MERA V.	17-12-2020	
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING - SUBROUTINAS PLANO INFORMÁTICO		<b>D01-004</b>			ESCALA N/A
					N/A

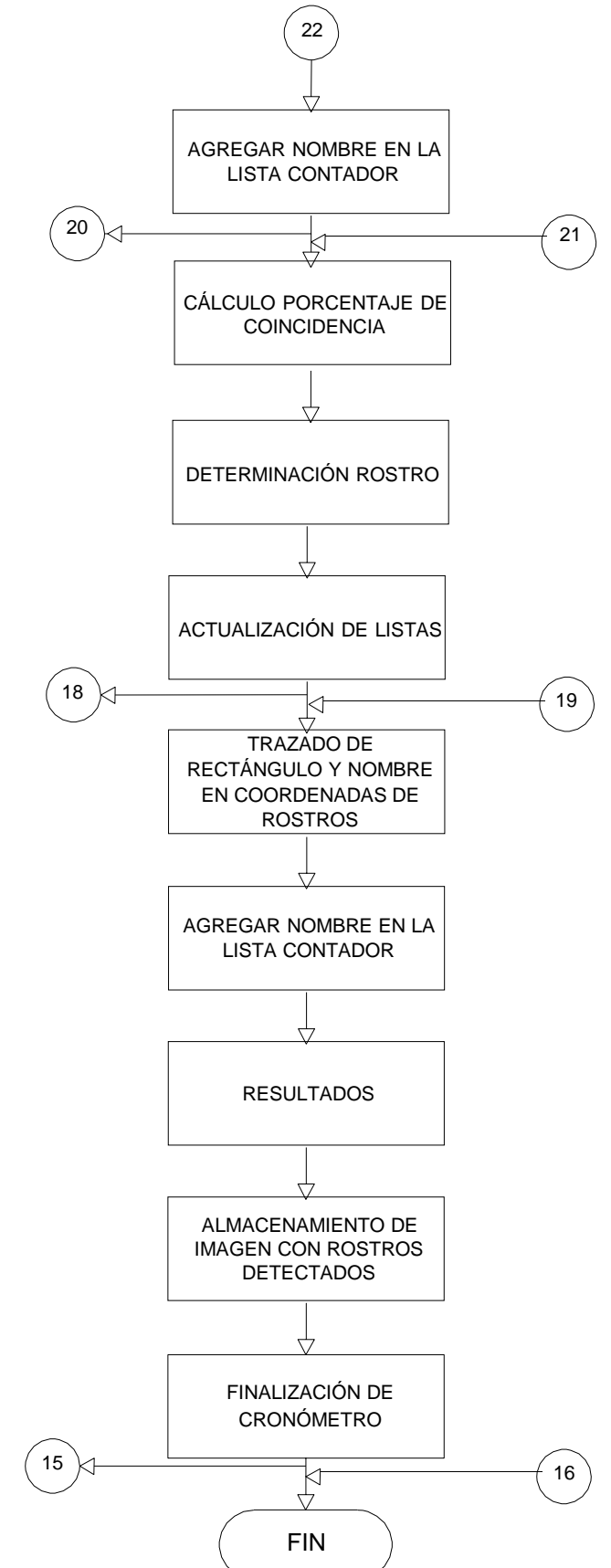
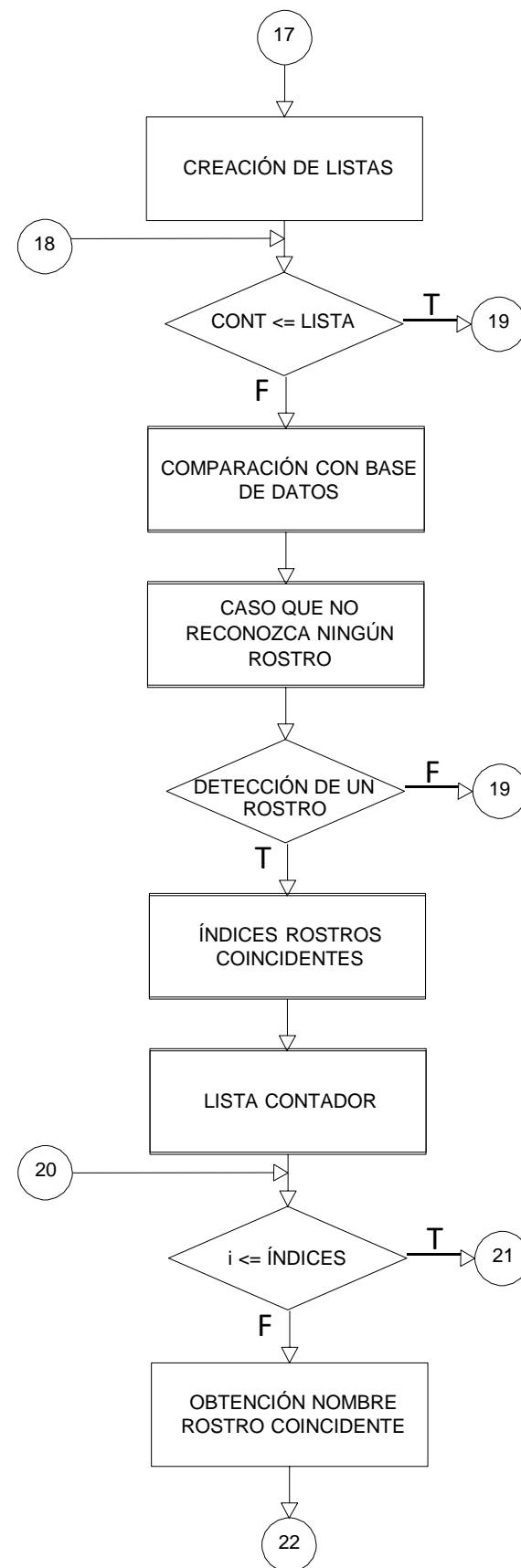
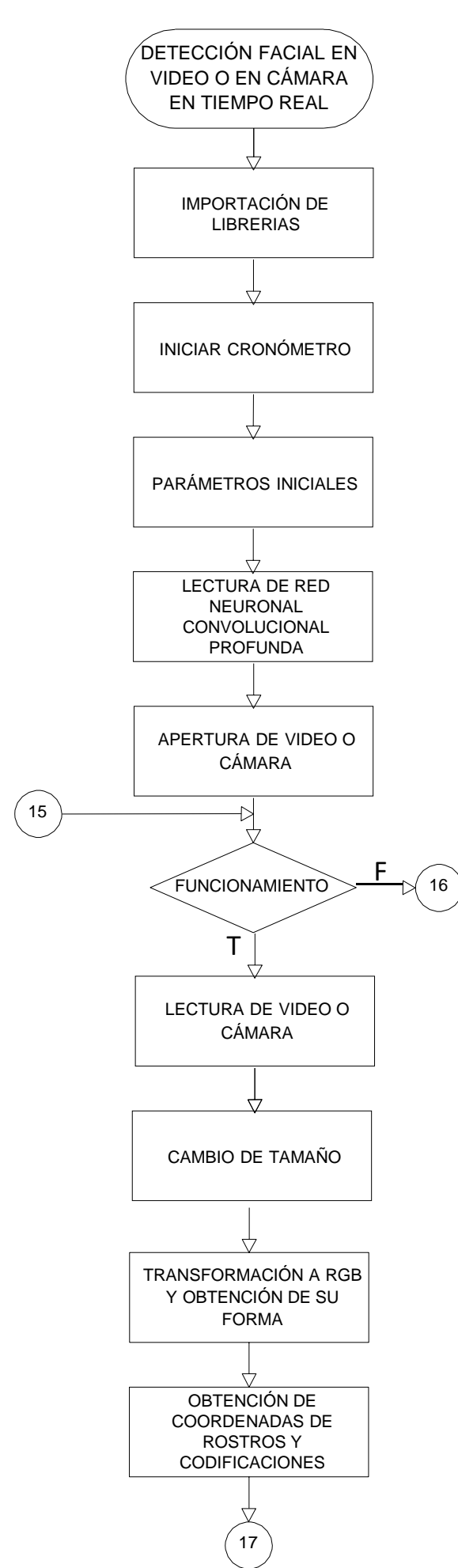




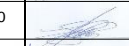
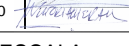
UIDE	 <b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	20-10-2020	
		DIS.	PÉREZ S.	05-04-2020	
		REV.	MERA V.	17-12-2020	
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING - SUBROUTINAS PLANO INFORMÁTICO		<b>D01-005</b>			ESCALA N/A
					N/A







<b>UIDE</b>		<b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	20-10-2020	
			DIS.	PÉREZ S.	05-04-2020	
			REV.	MERA V.	17-12-2020	
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING - SUBRUTINAS PLANO INFORMÁTICO			<b>D01-006</b>		ESCALA	N/A
					N/A	

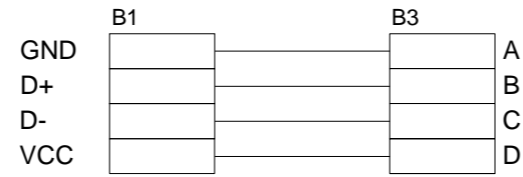




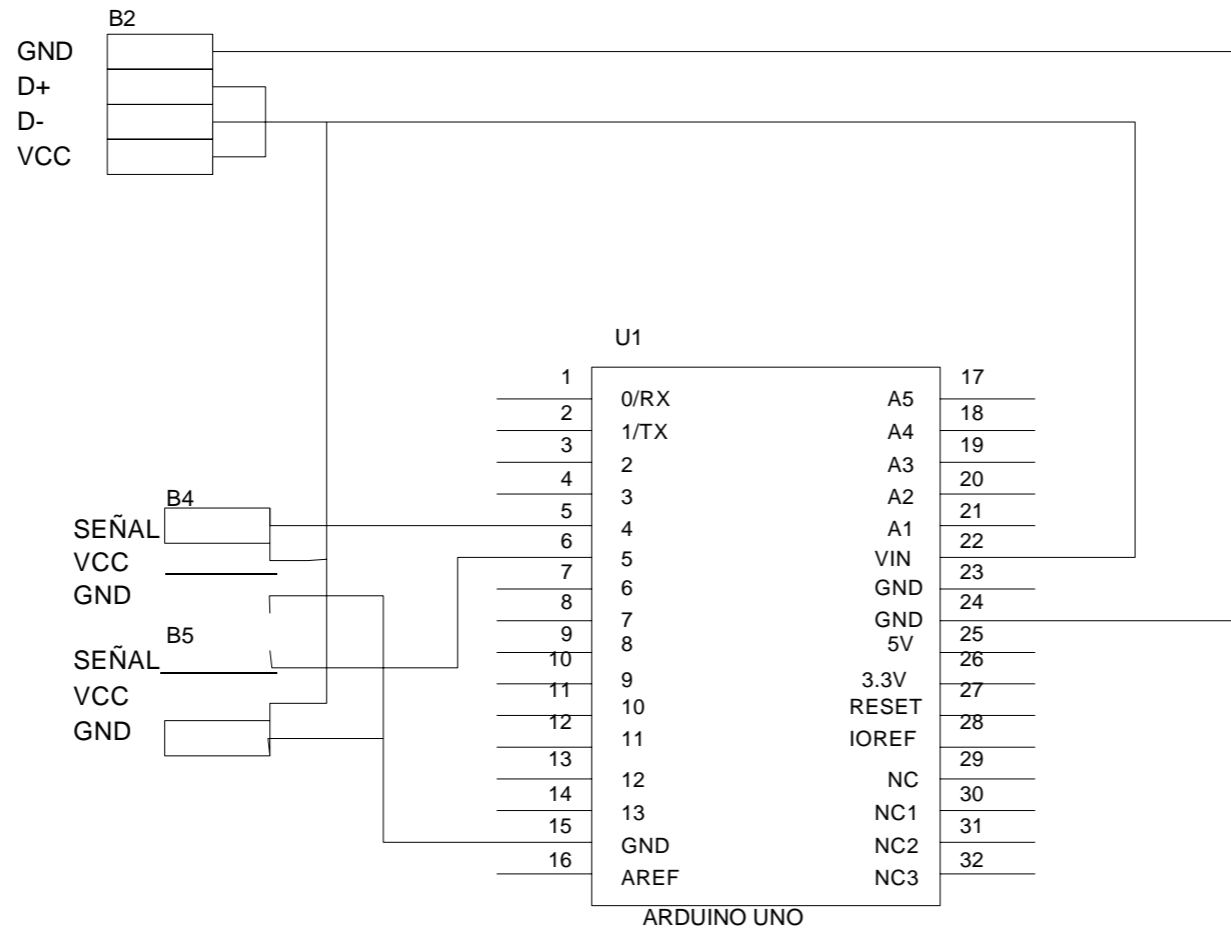
<b>UIDE</b>  <b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	20-10-2020	
	DIS.	PÉREZ S.	05-04-2020	
	REV.	MERA V.	17-12-2020	
<b>DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING - SUBRUTINAS</b> <b>PLANO INFORMÁTICO</b>		<b>D01-007</b>		ESCALA
				N/A
				N/A

POS.	DESCRIPCIÓN	CANT.	OBSERVACIONES
	BORNERAS		
B1	4 PINES USB	1	USB-C 3.1
B2	4 PINES USB	1	USB-B 2.0
B3	4 PINES CÁMARA DE PROFUNDIDAD	1	INTEL REALSENSE D415
B4 - B5	3 PINES SERVO MOTOR	2	MG995 100mA - 12 kgf cm
	MICROCONTROLADORES		
U1	ARDUINO UNO ATMEGA328P	1	

<b>UIDE</b>		<b>INGENIERÍA MECATRÓNICA</b>	DIB.	PÉREZ S.	21-05-2020	
			DIS.	PÉREZ S.	14-05-2020	
			REV.	MERA V.	17-12-2020	
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING PLANO ELECTRÓNICO			<b>D02-001</b>			ESCALA N/A
						N/A



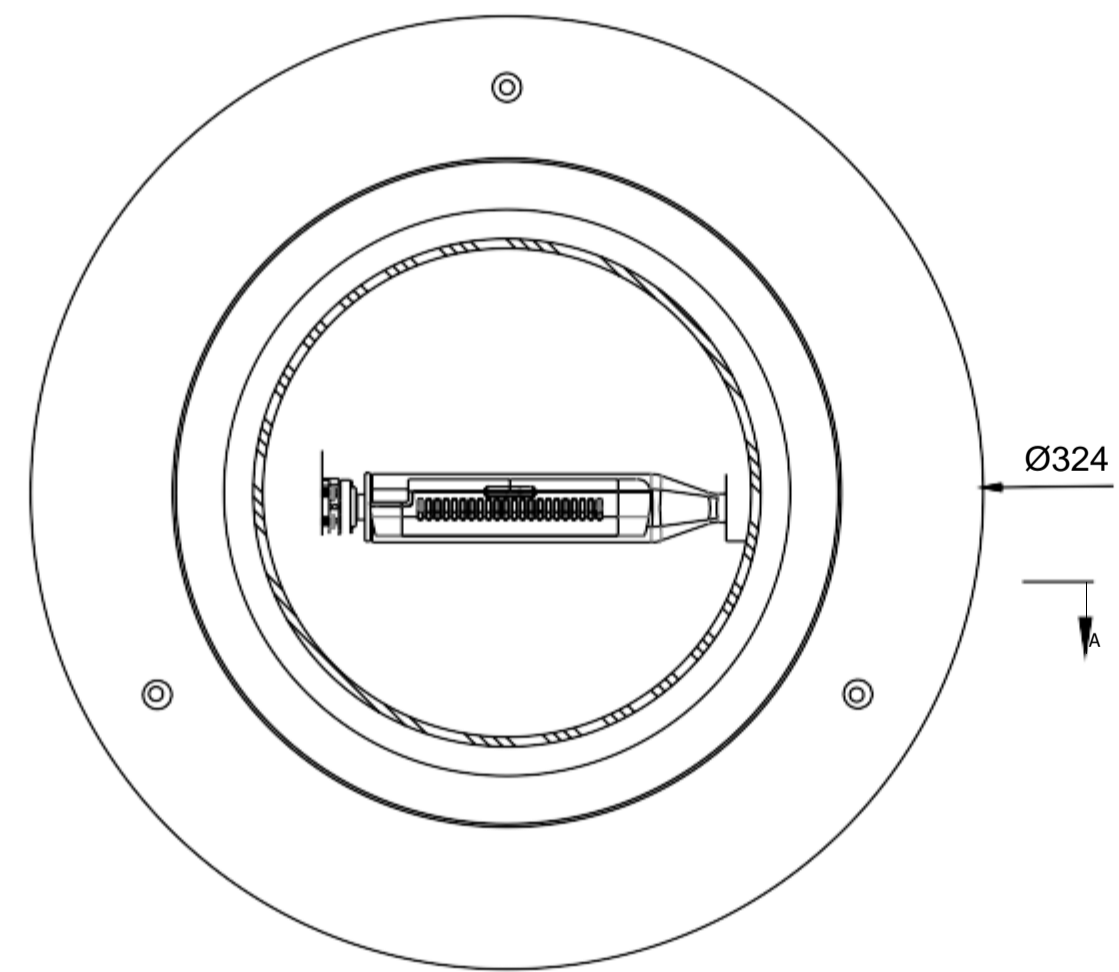
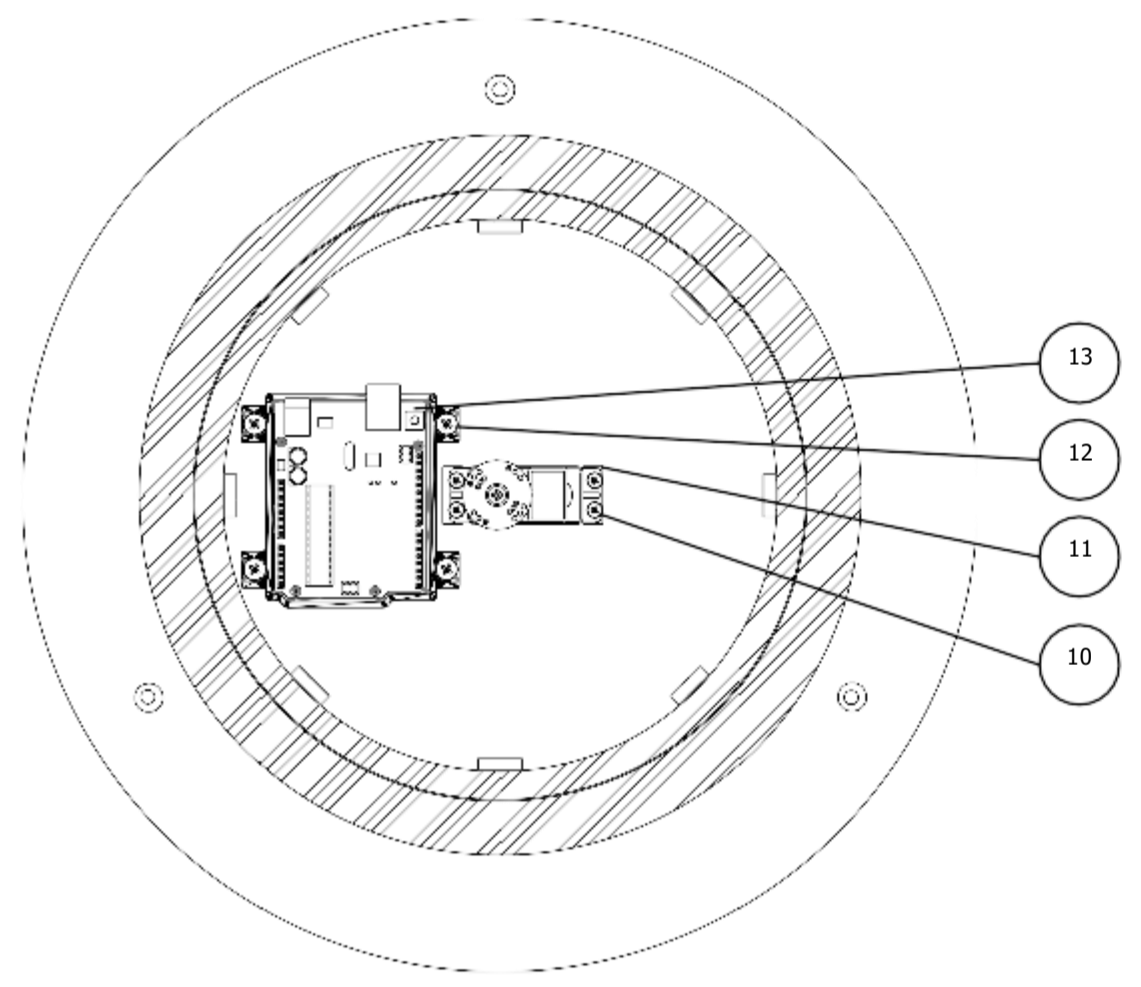
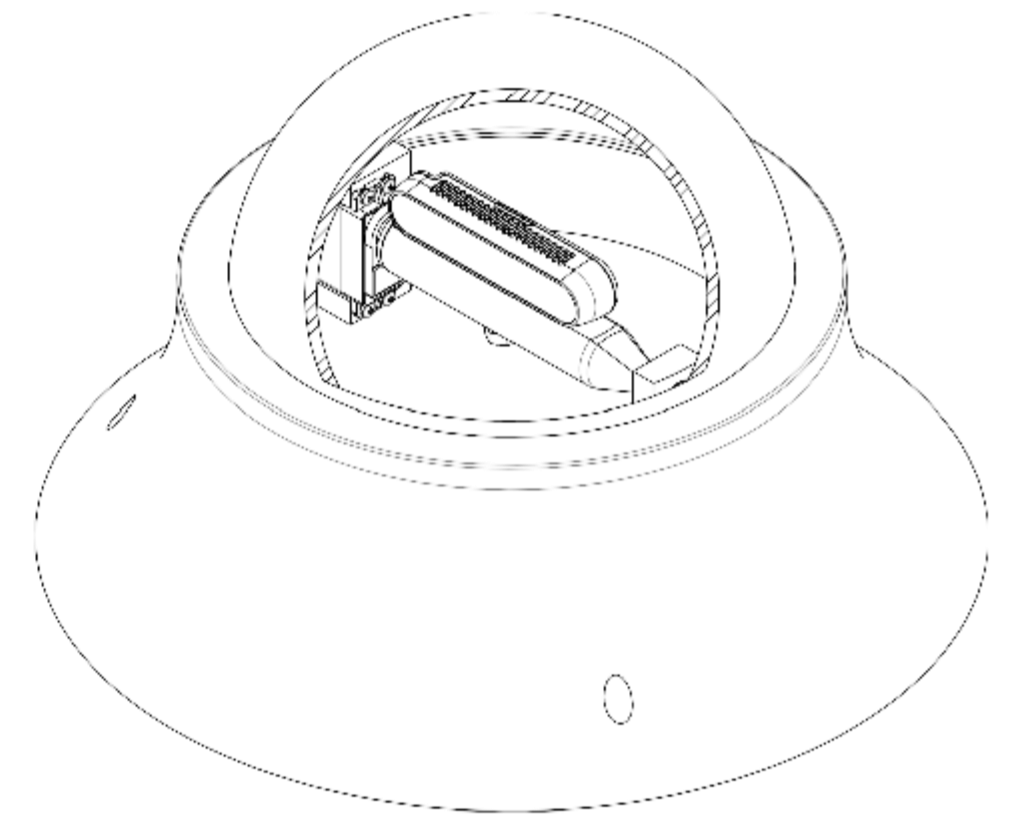
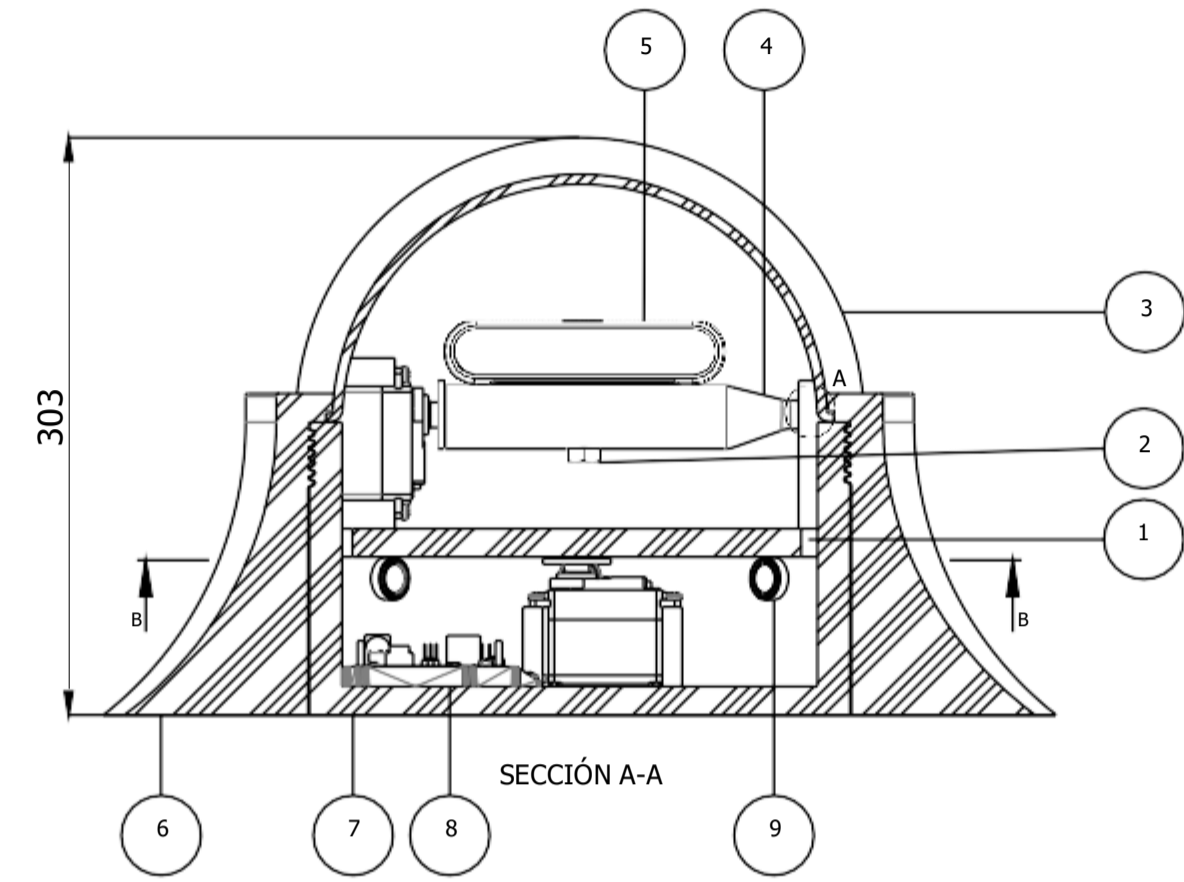
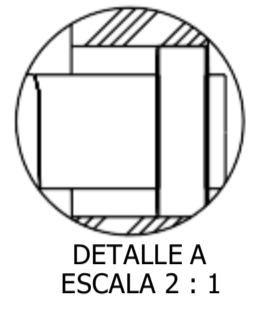
B1 - B2 - USB	
GND	Ground 0 [Vdc]
D+	Datos +
D-	Datos -
VCC	5 [Vdc]



B3 - CÁMARA DE PROFUNDIDAD	
A	Ground 0 [Vdc]
B	Datos +
C	Datos -
D	5 [Vdc]

B4 - B5 - SERVO MOTOR	
SEÑAL	0 - 5 [Vdc]
VCC	5 [Vdc]
GND	Ground 0 [Vdc]

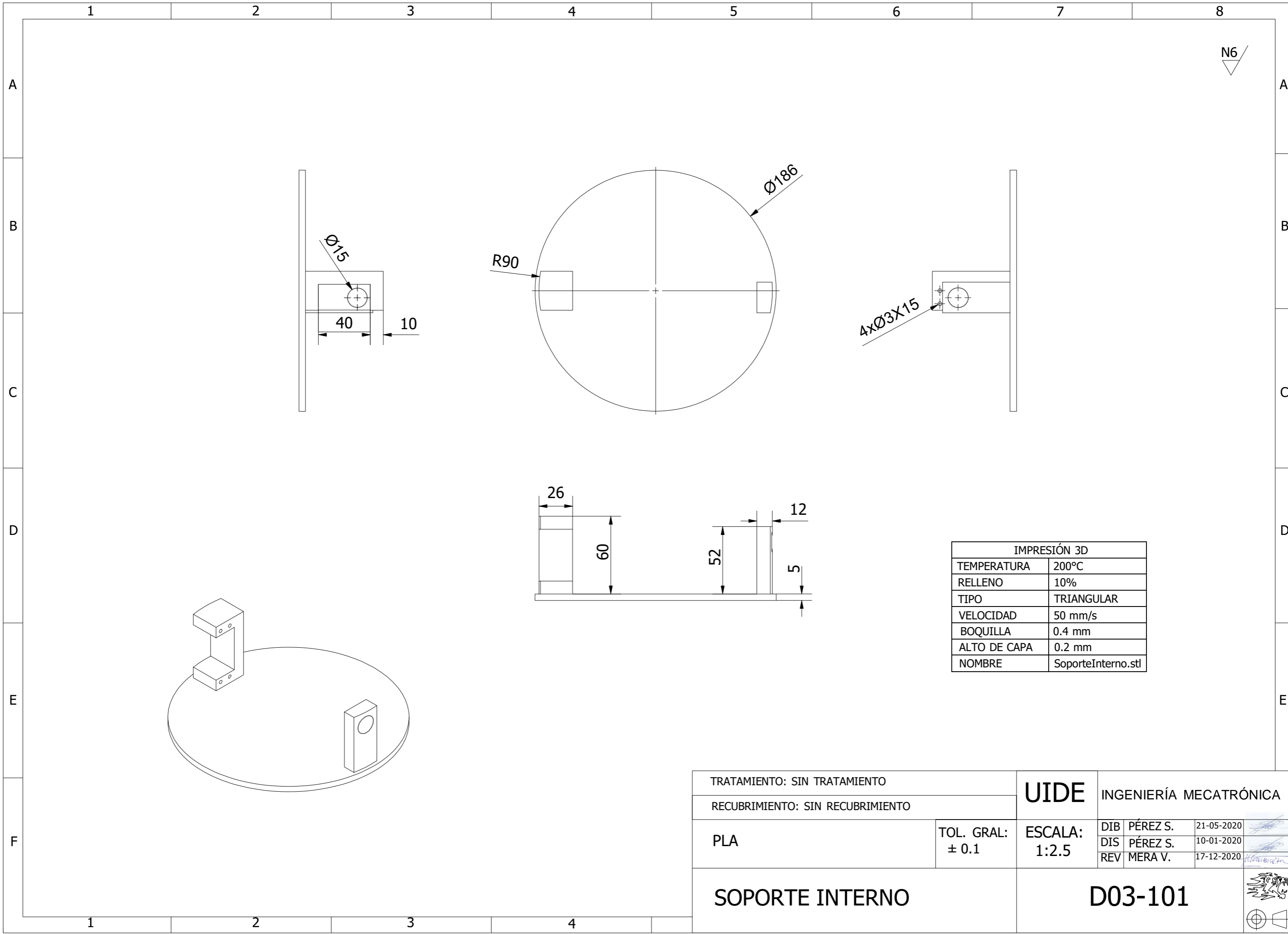
<b>UIDE</b>		DIB.	PÉREZ S.	21-05-2020	
		DIS.	PÉREZ S.	14-05-2020	
		REV.	MERA V.	17-12-2020	
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN DEEP LEARNING PLANO ELECTRÓNICO		<b>D02-001</b>		ESCALA N/A	
				N/A	



ESPECIFICACIONES TÉCNICAS	
ANCHO	324
ALTO	303
PROFUNDIDAD	324
ALIMENTACIÓN	+5V / 1A
PESO APROXIMADO	3.072 kg

13	1-D	MICROCONTROLADOR	1	VARIOS		ATMEGA328P
12	2-D	TORNILLO SOPORTE MICROCONTROLADOR	4	ACERO A2 INOX Z		GRADO 4.6 M4 X 5
11	2-D	SERVOMOTOR	2	VARIOS		MG995 ROTACIÓN CONTINUA
10	2-D	TORNILLO SERVOMOTOR	8	ACERO A2 INOX Z		GRADO 4.6 M3 X 16
9	4-B	RODAMIENTO	9	VARIOS		NSK 6700ZZ
8	4-B	SOPORTE MICROCONTROLADOR	1	PLA		<a href="https://www.thingiverse.com/thing:33327">https://www.thingiverse.com/thing:33327</a>
7	4-B	BASE INTERNA	1	PLA	D03-105	
6	3-B	BASE EXTERNA	1	PLA	D03-104	
5	4-B	CÁMARA DE PROFUNDIDAD	1	VARIOS		INTEL REALSENSE D415
4	4-B	EJE	1	PLA	D03-103	
3	4-A	DOMO	1	ACRÍLICO	D03-102	
2	4-B	PERNO CÁMARA	1	ACERO A2 INOX Z NEGRO		GRADO 12.9 1/4"-20 UNC x 5/8"
1	4-B	SOPORTE INTERNO	1	PLA	D03-101	
POS	ZONA	DENOMINACIÓN	CANT	MATERIAL	PLANO NORMA	OBSERVACIÓN

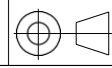
UIDE		INGENIERÍA MECATRÓNICA	DIB	PÉREZ S.	17-04-20
			DIS	PÉREZ S.	10-01-20
			REV	MIERA V.	17-12-20
PROTOTIPO CÁMARA DOMO TIPO PTZ			D03-001		ESCALA 1:2.5

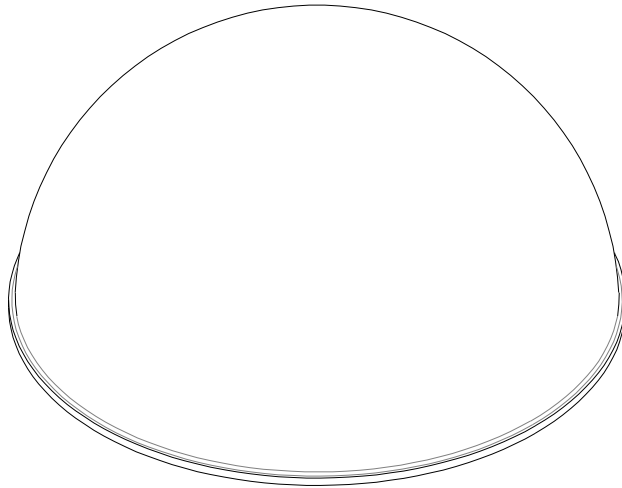
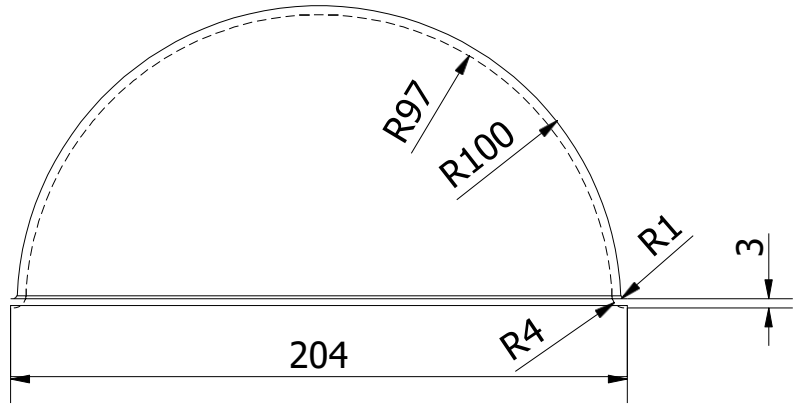


N6

IMPRESIÓN 3D	
TEMPERATURA	200°C
RELLENO	10%
TIPO	TRIANGULAR
VELOCIDAD	50 mm/s
BOQUILLA	0.4 mm
ALTO DE CAPA	0.2 mm
NOMBRE	SoporteInterno.stl

TRATAMIENTO: SIN TRATAMIENTO		<b>UIDE</b>	INGENIERÍA MECATRÓNICA		
RECUBRIMIENTO: SIN RECUBRIMIENTO			DIB	PÉREZ S.	21-05-2020
PLA	TOL. GRAL: ± 0.1	ESCALA: 1:2.5	DIS	PÉREZ S.	10-01-2020
			REV	MERA V.	17-12-2020
			<b>SOPORTE INTERNO</b>		<b>D03-101</b>

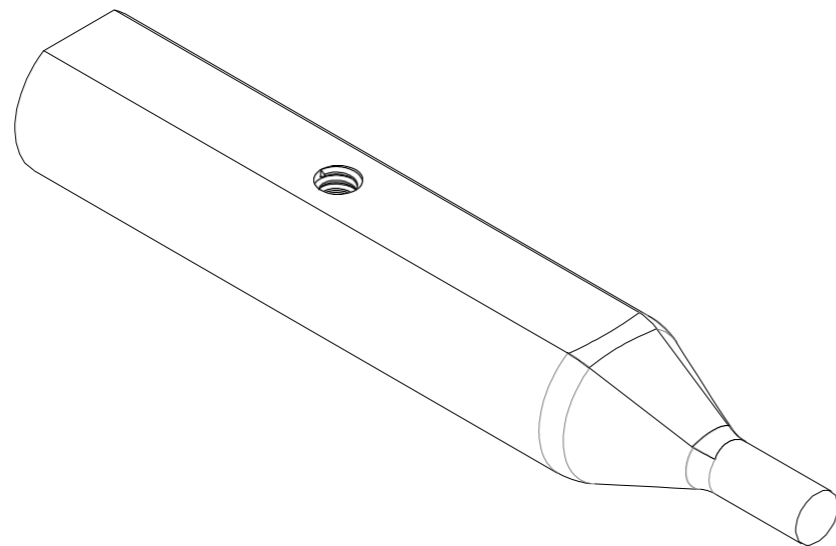
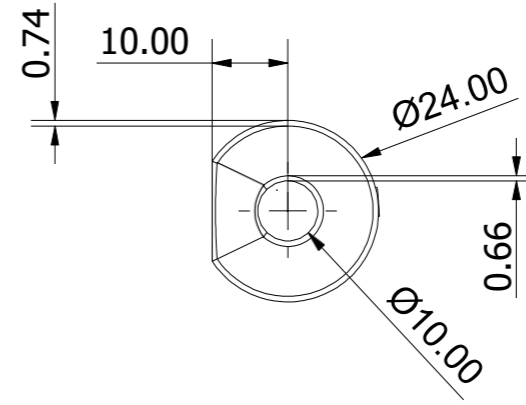
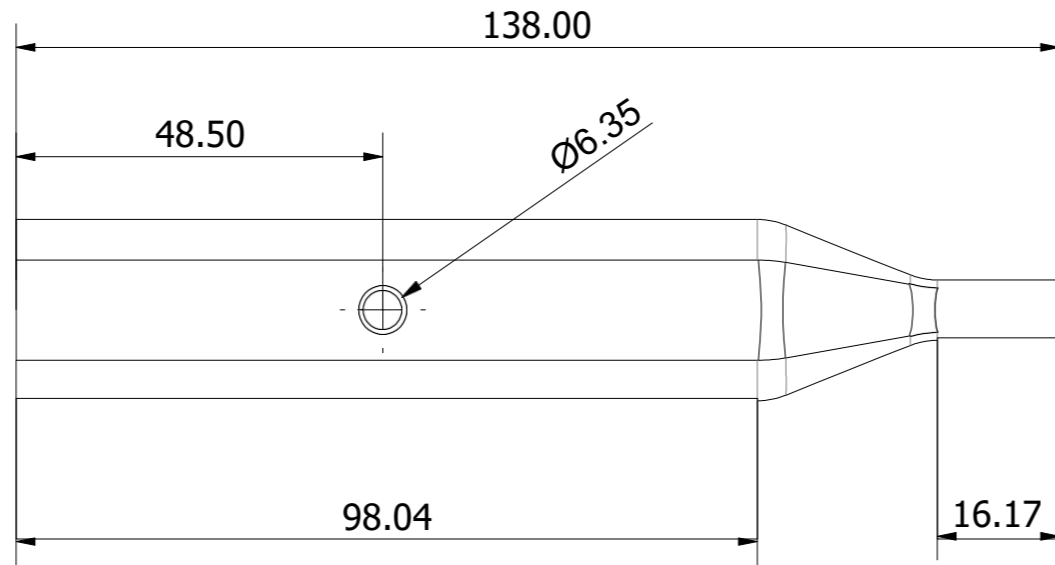




NOTA: TERMOFORMADO

TRATAMIENTO: SIN TRATAMIENTO		<b>UIDE</b>	INGENIERÍA MECATRÓNICA			
RECUBRIMIENTO: SIN RECUBRIMIENTO			DIB	PÉREZ S.	21-05-2020	
ACRÍLICO	TOL. GRAL: ± 1	ESCALA: 1:2.5	DIS	PÉREZ S.	10-01-2020	
			REV	MERA V.	17-12-2020	
DOMO		<b>D03-102</b>				

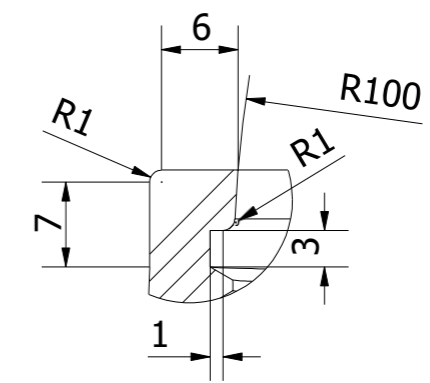
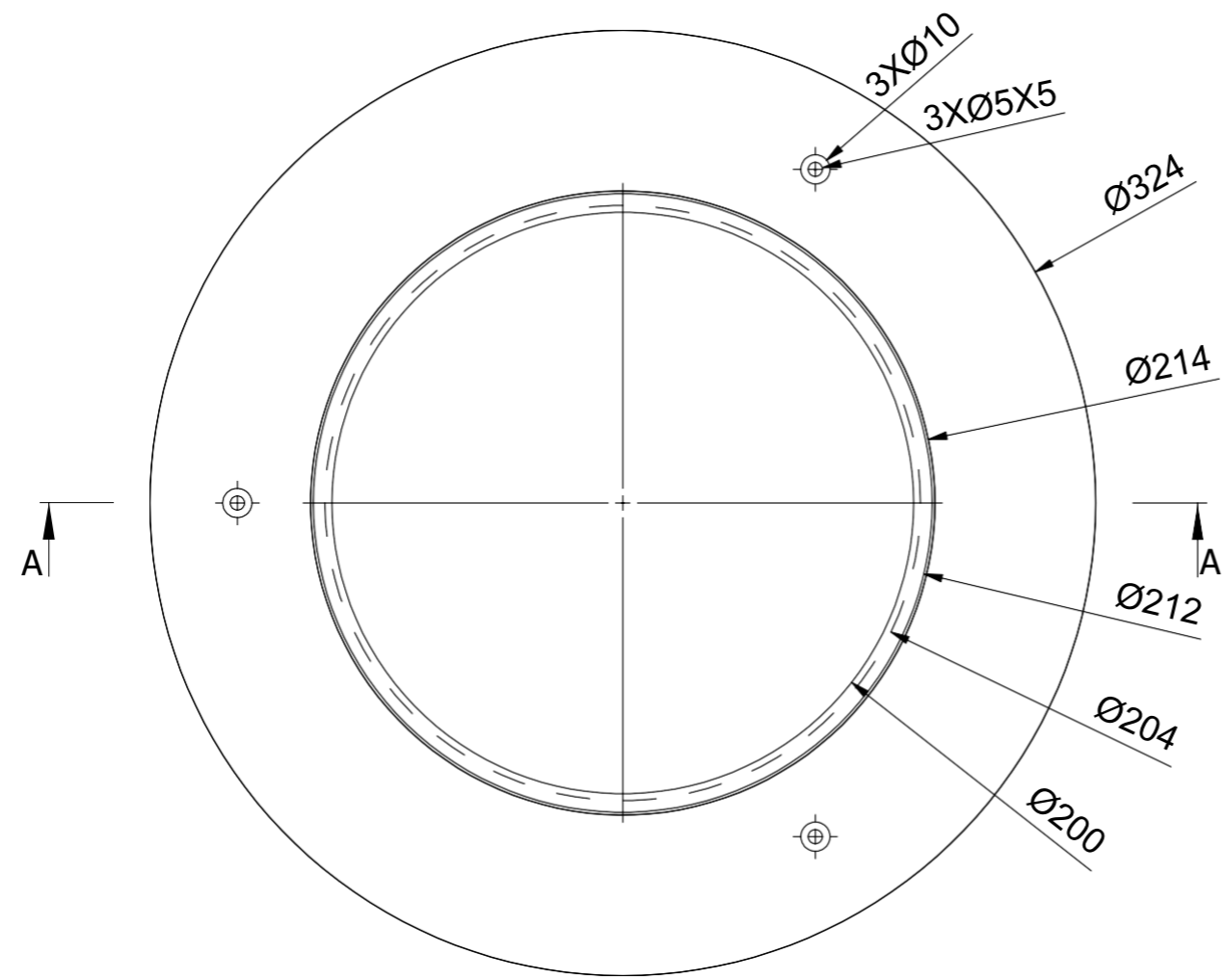
N6



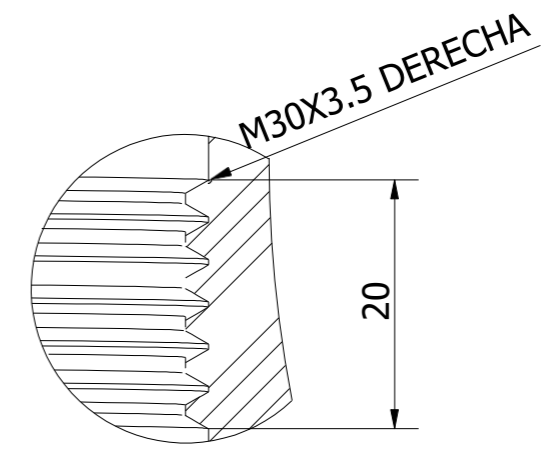
IMPRESIÓN 3D	
TEMPERATURA	200°C
RELLENO	10%
TIPO	TRIANGULAR
VELOCIDAD	50 mm/s
BOQUILLA	0.4 mm
ALTO DE CAPA	0.2 mm
NOMBRE	Eje.stl

TRATAMIENTO: SIN TRATAMIENTO		<b>UIDE</b>	INGENIERÍA MECATRÓNICA			
RECUBRIMIENTO: SIN RECUBRIMIENTO			DIB	PÉREZ S.	21-05-2020	
PLA	TOL. GRAL: ± 0.1	ESCALA: 1:1	DIS	PÉREZ S.	10-01-2020	
			REV	MERA V.	17-12-2020	
<b>EJE</b>		<b>D03-103</b>				

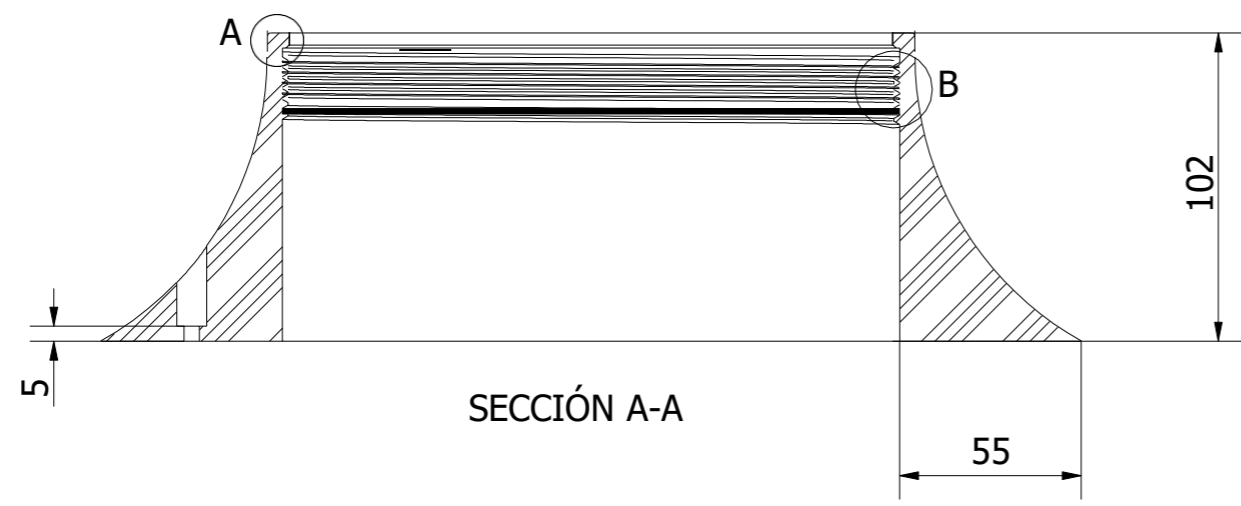




DETALLE A  
ESCALA 2 : 1



DETALLE B  
ESCALA 2 : 1

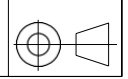


SECCIÓN A-A

IMPRESIÓN 3D	
TEMPERATURA	200°C
RELLENO	10%
TIPO	TRIANGULAR
VELOCIDAD	50 mm/s
BOQUILLA	0.4 mm
ALTO DE CAPA	0.2 mm
NOMBRE	BaseExterna.stl

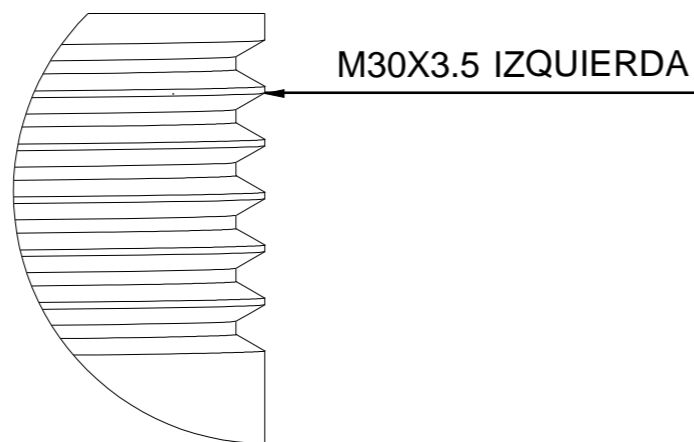
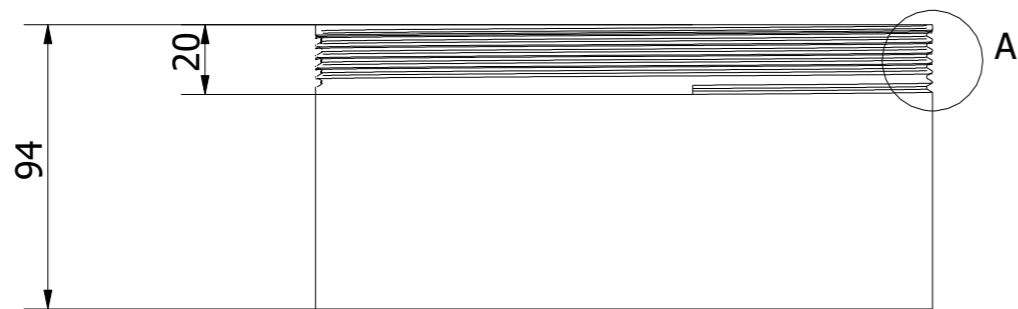
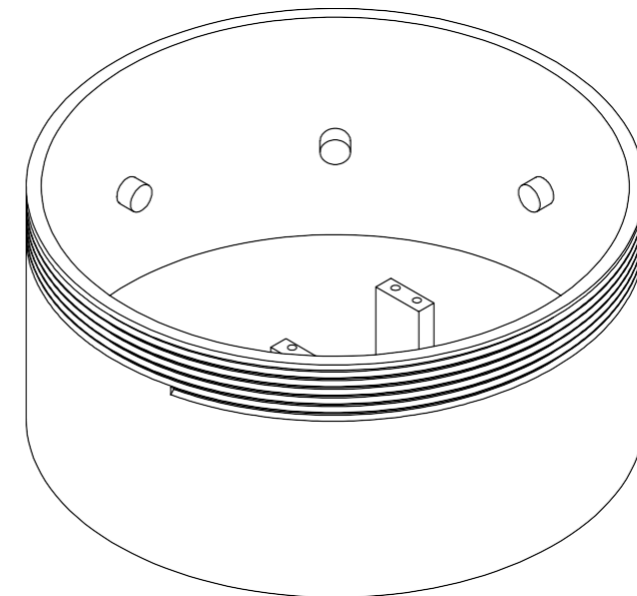
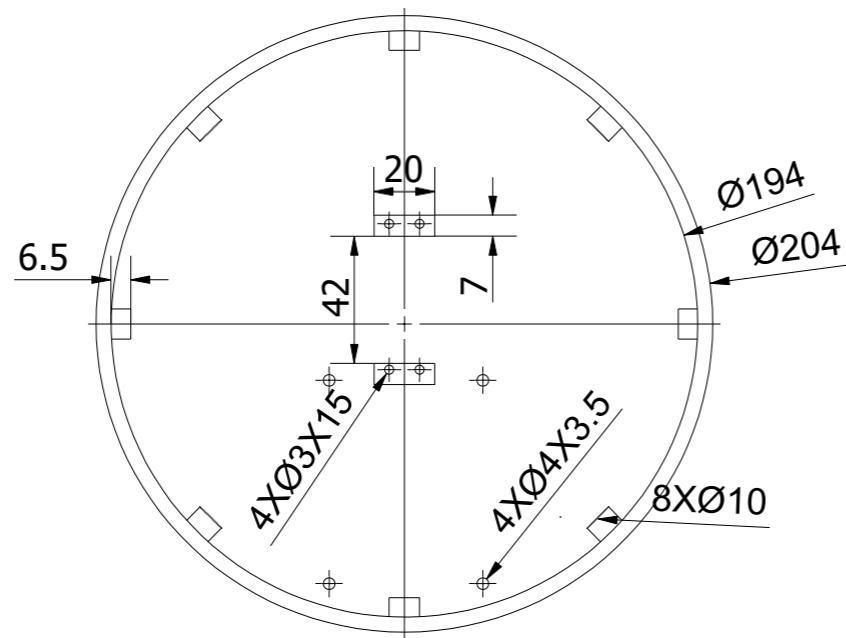
TRATAMIENTO: SIN TRATAMIENTO		<b>UIDE</b>	INGENIERÍA MECATRÓNICA		
RECUBRIMIENTO: SIN RECUBRIMIENTO			DIB	PÉREZ S.	21-05-2020
PLA	TOL. GRAL: ± 0.1	ESCALA: 1:2.5	DIS	PÉREZ S.	10-01-2020
			REV	MERA V.	17-12-2020
			<b>BASE EXTERNA</b>		

N6





N6



DETALLE A  
ESCALA 2 : 1

IMPRESIÓN 3D	
TEMPERATURA	200°C
RELLENO	10%
TIPO	TRIANGULAR
VELOCIDAD	50 mm/s
BOQUILLA	0.4 mm
ALTO DE CAPA	0.2 mm
NOMBRE	BaseInterna.stl

TRATAMIENTO: SIN TRATAMIENTO		<b>UIDE</b>	INGENIERÍA MECATRÓNICA		
RECUBRIMIENTO: SIN RECUBRIMIENTO			DIB	PÉREZ S.	21-05-2020
PLA	TOL. GRAL: ± 0.1	ESCALA: 1:2.5	DIS	PÉREZ S.	10-01-2020
			REV	MERA V.	17-12-2020
			<b>BASE INTERNA</b>		<b>D03-105</b>

