

Clasificador de la semilla Sacha inchi mediante aplicación de visión artificial

Autor

Isaac David Rangel Blanco

Universidad Autónoma de Bucaramanga
Facultad de Ingeniería Mecatrónica Bucaramanga
2022

Clasificador de la semilla Sacha inchi mediante aplicación de visión artificial

Autor:

Isaac David Rangel Blanco

Proyecto de grado presentado como requisito parcial para optar al título de:
Ingeniero Mecatrónico

Director:

M.Sc. Ing. Oscar Eduardo Rueda Sánchez

Codirector:

M.Sc. Ing. Hernando Gonzales Acevedo

Universidad Autónoma de Bucaramanga
Facultad de Ingeniería Mecatrónica Bucaramanga
2022

AGRADECIMIENTOS

Quiero agradecer principalmente a mis familiares que estuvieron dándome ánimos y apoyándome en todo este proceso, a quien más destacaría sería a mi madre que siempre estuvo cada día pendiente de mi progreso y de mi salud mientras desarrollaba el proyecto.

Además, debo reconocer la ayuda que me prestaron varios compañeros de la carrera y obviamente mi director de proyecto Oscar Rueda, al cual le doy mis más sinceros agradecimientos por todo el apoyo, gestión, seguimiento y confianza ofrecida en la elaboración de este proyecto.

Por último, quiero agradecer a la Universidad Autónoma de Bucaramanga por los conocimientos y experiencias brindadas, las cuales me aportaron para mi crecimiento profesional y para el desarrollo e implementación del proyecto.

TABLA DE CONTENIDOS

		Pág.
1	INTRODUCCIÓN	12
1.1	DESCRIPCIÓN DEL PROBLEMA.....	12
1.2	JUSTIFICACIÓN DEL PROBLEMA	13
2	OBJETIVOS	15
2.1	OBJETIVO GENERAL	15
2.2	OBJETIVOS ESPECÍFICOS	15
3	ESTADO DEL ARTE	16
4	MARCO TEORICO.....	19
4.1	GENERALIDADES DEL FRUTO SACHA INCHI	19
	4.1.1 CARACTERÍSTICAS DEL GÉNERO Y LA FAMILIA	19
	4.1.2 MORFOLOGÍA	19
	4.1.3 DISTRIBUCIÓN GEOGRÁFICA	19
	4.1.4 CARACTERÍSTICAS NUTRICIONALES.....	20
	4.1.5 BENEFICIOS PARA LA SALUD	20
	4.1.6 PRODUCTOS DERIVADOS DEL SACHA INCHI.....	21
	4.1.7 EL ACEITE DEL SACHA INCHI.	22
	4.1.8 PROCESO DE EXTRACCIÓN DEL ACEITE DEL SACHA INCHI. 22	22
4.2	VISIÓN ARTIFICIAL.....	23
4.3	TIPOS DE ILUMINACIÓN	24
	4.3.1 ILUMINACIÓN DIFUSA TIPO DOMO.....	24
	4.3.2 ILUMINACIÓN A CONTRALUZ O DE FONDO	25
	4.3.3 ILUMINACIÓN CUADRADA O DIRECTA.....	26
4.4	REDES NEURONALES CONVOLUCIONALES PARA DETECCIÓN DE OBJETOS	26
	4.4.1 LA NEURONA	26
	4.4.2 ESTRUCTURA DE LAS REDES NEURONALES.....	28
4.5	REDES NEURONALES CONVOLUCIONALES	30
	4.5.1 CONVOLUCIÓN	31
	4.5.2 POOLING	32

4.5.3	CAPA TOTALMENTE CONECTADA O FULLY CONNECTED	33
4.5.4	CAPA DE CLASIFICACIÓN.....	33
4.5.5	ARQUITECTURAS DE LAS CNN.....	33
4.5.6	PARÁMETROS E HIPERPARÁMETROS	34
5	METODOLOGIA	35
5.1.1	CONCEPTO DE OPERACIONES.	36
5.1.2	REQUISITOS DEL SISTEMA.	36
5.1.3	INGENIERÍA DE DETALLE.	36
5.1.4	DISEÑO DETALLADO.....	36
5.1.5	VERIFICACIÓN DE LOS COMPONENTES.	36
5.1.6	VERIFICACIÓN DEL SUBSISTEMA.	36
5.1.7	VERIFICACIÓN Y DESARROLLO DEL SISTEMA.....	36
5.1.8	VALIDACIÓN DEL SISTEMA.	36
5.1.9	OPERACIONES DE MANTENIMIENTO.....	36
6	INSTRUMENTACIÓN	37
6.1	DISPOSITIVO DE CAPTACIÓN DE IMÁGENES.....	37
6.2	DISPOSITIVO DE PROCESAMIENTO Y CONTROL	38
6.3	SENSOR DE PRESENCIA DE OBJETOS.....	39
6.4	SISTEMA DE EXPULSIÓN DE SEMILLAS.....	41
6.5	SISTEMA DE ILUMINACIÓN	42
7	MODIFICACIONES DEL PROTOTIPO	43
7.1	INSTALACIÓN DE LA RECÁMARA DE ILUMINACIÓN	44
7.2	REEMPLAZO DE CINTAS DE LA BANDA TRANSPORTADORA...	45
7.3	FIJACIÓN DE SENSORES Y ACTUADORES.....	46
7.4	CAJA O PANEL DE CONTROL	49
7.5	ESTADO FINAL DE LA MÁQUINA CON LAS MODIFICACIONES .	51
8	DISEÑO ELECTRONICO.....	52
8.1	FUENTE DE ENERGÍA.....	52
8.2	CONTROLADOR	52
8.3	CONEXIONES DE LOS MOTORES, VÁLVULAS Y CINTAS LED ..	53
8.4	CONEXIÓN DE LAS CÁMARAS WEB, SENSORES, PULSADORES Y PANTALLA LED	53

9	PROGRAMACIÓN	55
9.1	CONFIGURACIÓN DE LA RASPBERRYPI PI4.....	55
	9.1.1 LIBRERÍAS PRINCIPALES	55
9.2	CÓDIGO EN PYTHON.....	56
	9.2.1 ENTRENAMIENTO DEL CÓDIGO	56
	9.2.2 FUNCIONAMIENTO GENERAL	60
10	VALIDACIÓN	63
11	CONCLUSIONES Y RECOMENDACIONES	68
12	REFERENCIAS.....	69
13	ANEXOS	72

LISTA DE ILUSTRACIONES

	Pág.
Ilustración 1 Fruto de sachá inchi maduro de color marrón claro.....	13
Ilustración 2 Fruto de sachá inchi de color marrón oscuro.....	14
Ilustración 3 Fruto de sachá inchi 4,5,6 y 7 lóbulos	19
Ilustración 4 Aceite orgánico de Sachá inchi	22
Ilustración 5 Iluminación tipo domo.....	24
Ilustración 6 Aplicación de iluminación de domo a una tapa de plástico.....	25
Ilustración 7 Iluminación de fondo.....	25
Ilustración 8 Aplicación de iluminación de fondo a una regleta de pines conectores	25
Ilustración 9 Iluminación cuadrada o directa	26
Ilustración 10 Aplicación de iluminación cuadrada en un CI	26
Ilustración 11 Estructura de una neurona artificial	27
Ilustración 12 Funciones de activación	28
Ilustración 13 Representación gráfica de una red neuronal monocapa	29
Ilustración 14 Estructura de una red neuronal multicapa	29
Ilustración 15 Representación gráfica de una CNN	30
Ilustración 16 Representación gráfica de una RNN	30
Ilustración 17 Convolución de un Kernel sobre una imagen	31
Ilustración 18 Max pooling	32
Ilustración 19 Metodología basada en el modelo V	35
Ilustración 20 Válvula neumática 2V025-08-12V	41
Ilustración 21 Láminas de la recámara de iluminación	42
Ilustración 22 Recámara de iluminación ensamblada (sin tapa superior)	42
Ilustración 23 Prueba de funcionamiento de la cinta led en tapa lateral de la recámara.....	42
Ilustración 24 Prototipo de máquina clasificadora sin modificaciones.....	43
Ilustración 25 Motores DC	44
Ilustración 26 Recámaras de iluminación instaladas	45
Ilustración 27 Soportes de aluminio para las recámaras de iluminación.....	45
Ilustración 28 Cintas de las bandas transportadoras en mal estado	46
Ilustración 29 Nuevas cintas de las bandas transportadoras	46
Ilustración 30 Diseño 3D de los soportes de las válvulas neumáticas	47
Ilustración 31 Soportes de las válvulas neumáticas.....	47
Ilustración 32 Válvulas neumáticas acopladas a los soportes y fijadas	47
Ilustración 33 Unidad de mantenimiento fijada	48
Ilustración 34 Conector neumático en T	48
Ilustración 35 Diseño 3D del soporte para el sensor infrarrojo	49
Ilustración 36 Sensores infrarrojo acoplados a los soportes y fijados.....	49
Ilustración 37 Panel de control.....	50

Ilustración 38 Panel de control fijado al prototipo de maquina clasificadora .	50
Ilustración 39 Conexiones eléctricas internas de la caja de control	50
Ilustración 40 Prototipo de máquina clasificadora con las modificaciones	51
Ilustración 41 Fuente conmutada 12V 10A	52
Ilustración 42 Pines de conexión de la Raspberrypi 4	52
Ilustración 43 Conexiones de las cámaras, válvulas neumáticas y cintas led	53
Ilustración 44 Conexiones de los sensores, pulsadores, pantalla led y cámaras web	54
Ilustración 45 .Interfaz de Raspberry Pi Imager	55
Ilustración 46 Propiedades de la carpeta de imágenes para entrenamiento	56
Ilustración 47 Propiedades de las carpetas de semillas claras y oscuras de la carpeta de entrenamiento	57
Ilustración 48 Propiedades de la carpeta de imágenes para validación	57
Ilustración 49 Propiedades de las carpetas de semillas claras y oscuras de la carpeta de validación	57
Ilustración 50 Muestra de imagen de semilla "Clara"	57
Ilustración 51 Muestra de imagen de semilla "Oscura"	58
Ilustración 52 Parámetros requeridos para el entrenamiento	58
Ilustración 53 Proceso de entrenamiento visualización	59
Ilustración 54 Representación gráfica de los datos de optimización de la red neuronal	60
Ilustración 55 Archivos de modelo y pesos generados del entrenamiento ...	60
Ilustración 56 Diagrama de funcionamiento de la maquina clasificadora.....	61
Ilustración 57 Diagrama de flujo del código de funcionamiento	62
Ilustración 58 Pantalla LED con datos de ejemplo.....	62
Ilustración 59 Muestra #1 de semillas Claras	63
Ilustración 60 Muestra #1 de semilla oscura	64
Ilustración 61 Muestras #9 y #16 de semilla Oscura con respuesta errónea	66
Ilustración 62 Grafica de análisis estadístico de las pruebas de validación..	66

LISTA DE TABLAS

	Pág.
Tabla 1 Cámaras web investigadas	37
Tabla 2 Características de cámaras web investigadas	37
Tabla 3 Matriz QFD de las cámaras web	38
Tabla 4 Características de unidad de procesamiento	39
Tabla 5 Matriz QFD de la unidad de procesamiento	39
Tabla 6 Sensores de presencia de objetos investigados	40
Tabla 7 Características de sensores investigados	40
Tabla 8 Matriz QFD de los sensores investigados	40
Tabla 9 Mediciones de tensión y corriente	44
Tabla 10 Datos de optimización de la red neuronal	59
Tabla 11 Prueba con muestras de semillas Claras	63
Tabla 12 Pruebas con muestras de semillas Oscuras	64
Tabla 13 Pruebas con muestras de semillas mixtas	65

LISTA DE ANEXOS

	Pág.
Anexo 1 Código de entrenamiento de la CNN	72
Anexo 2 Código de control de la máquina de clasificación	74
Anexo 3 Manual de usuario de la máquina clasificadora de sachá inchi	81
Anexo 4 Características de la Raspberrypi 4 modelo B.....	82
Anexo 5 Especificaciones eléctricas de la Raspberrypi 4 modelo B	83
Anexo 6 Plano de dimensiones de recamara de iluminación.....	84
Anexo 7 Plano del soporte de la recamara de iluminación	85
Anexo 8 Plano del soporte del sensor infrarrojo	86
Anexo 9 Pata izquierda del soporte de la válvula	87
Anexo 10 Pata derecha del soporte de la válvula	88
Anexo 11 Tapa frontal del soporte de la válvula	89
Anexo 12 Tapa superior del soporte de la válvula	90
Anexo 13 Ensamble completo del soporte de la válvula.....	91
Anexo 14 Primeras 3 tablas de pruebas de validación de cada grupo de semillas.....	92
Anexo 15 Ultimas 3 tablas de pruebas de validación de cada grupo de semillas	93

1 INTRODUCCIÓN

El presente proyecto se enfoca en la mejora del método de visión artificial implementado en el proyecto “Construcción de un Prototipo de una Máquina Selectora de Semillas de Sacha Inchi” el cual concluyó con un prototipo de máquina clasificadora, que permite realizar un proceso dosificación y con la implementación de ciertos dispositivos adicionales permite el posterior procesamiento computacional para seleccionar la semillas dependiendo su color y forma. También es importante resaltar que fueron requeridas ciertas modificaciones físicas o estructurales en dicha máquina prototipo debido a las necesidades presentadas en el transcurso del proyecto actual, con el fin de lograr una mejoría del proceso de clasificación.

Las semillas ingresadas al prototipo de máquina clasificadora se plantean a las anteriores de la etapa de descascarado, las cuales están mezcladas entre capsulas marrones claros y semillas marrones oscuras que se contempla en el proceso general de obtención de aceite a partir de la semilla *Plukenetia* o más conocida como Sacha Inchi.

1.1 Descripción del problema

Debido al gran auge de la producción de aceite y demás derivados de los frutos de la planta de Sacha Inchi, en caso de Colombia la cual es producida principalmente en el Putumayo, como alternativa de la grave economía ilegal de la coca que se da en la región según informes de la UNODC (Oficina de las naciones unidas contra la Droga y el delito), y la cual se destina principalmente a la comercialización en fresco y a la extracción de aceite. También según un Boletín Técnico de abril del 2016 del MINISTERIO DE AGRICULTURA Y DESARROLLO RURAL de Perú, se encuentran plantaciones distribuidas por todo el trópico latinoamericano desde el Sur de México, Indias Occidentales, la Amazonía y el acre en Bolivia; además de distintos distritos del Perú como Madre de Dios, Huánuco, Pasco, San Martín, Ucayali y Loreto.

La especie no sólo es interesante desde una perspectiva nutricional, medicinal y cultural, si no también podría ser una interesante alternativa económica a los cultivos ilícitos. La especie sólo se conoce en hábitat bosque nuboso natural. Métodos de cosecha sostenibles pueden ser desarrollados, la especie podría convertirse en una nueva fuente de ingresos para las comunidades marginadas, y que por lo tanto crea un producto de conservación excelente. (Bussman, Tellez, & Glenn, 2009)

Actualmente el cultivo y producción de los derivados de esta planta es poco tecnificada y ante la gran demanda de los consumidores, las empresas se han visto obligadas a desarrollar máquinas automatizadas para aumentar la producción comercial debido a que hay varios países a los cuales se exporta el aceite de Sacha Inchi, pues se ha pasado de 244976,91 toneladas de

exportación concentradas en algunos países de destino en el año 2007 a 877002,33 toneladas dirigidas a 30 países de destino en el año 2013. Estados Unidos es el segundo destino de las exportaciones después de Alemania, que hace ya varios años viene siendo el principal destino de las exportaciones de aceite de sacha inchi (Grandez Muñoz & Chingo Coronado, 2015). En general el mercado europeo se resalta por ser el mercado más interesado en los productos del Sacha Inchi enfocándose en el aceite del mismo, debido a su alto contenido de omega 3, el cual mayormente es extraído del bacalao, pero posee propiedades energéticas y nutritivas inferiores a las del Sacha Inchi, además de ser un animal en peligro de extinción,

Se entiende que para obtener algún producto del fruto de sacha inchi, hay que realizar una previa selección de los frutos con la mejor calidad, ya que de esto depende el resultado final. Dicho proceso de clasificación usualmente se hace de manera manual. Lo anterior mencionado es el problema principal por tratar en el presente proyecto, ya que para poder realizar esta selección de forma eficiente y evitando la mayor cantidad de errores es necesario automatizar este proceso de selección y también implementar un algoritmo más elaborado con técnicas de visión artificial o métodos de selección más apropiado.

1.2 Justificación del problema

La automatización del proceso de selección es sumamente beneficiosa para las empresas, ya que se traduce en tiempos más cortos de procesamiento y por ende mayores ganancias comerciales. Dentro de las etapas para la extracción de aceite de Sacha Inchi se encuentran las siguientes: pesado, descascarado, selección de frutos, triturado, tamizado, tratamiento térmico, prensado y filtración.

Se describe a continuación dos fases en las cuales se puede presentar el fruto del Sacha Inchi después del proceso de descascarado:

- a) Fruto maduro en forma de capsula de color marrón claro (la cual contiene dentro la siguiente fase y sirve como cascara principal) ilustración 1.

Ilustración 1 Fruto de sacha inchi maduro de color marrón claro



Fuente: autor del proyecto

b) Fruto o semilla en forma de cápsula de color marrón oscuro (obtenida del descascarado) ilustración 2.

Ilustración 2 Fruto de sacha inchi de color marrón oscuro



Fuente: autor del proyecto

Además, después de este proceso de descascarado y selección según comentó Carlos Andrés Palacio Lopera, actual presidente del Ecosistema SumaSach'a en SumaSach'a, en el Tercer Workshop Proyecto Agrópolis - Mesa del Sistema de Innovación Agroindustrial de Santander (MACTOR): “Debido a que dichos frutos pueden ser muy similares en tamaños, se puede presentar en ocasiones problemas al seleccionar de manera manual los frutos después del proceso de descascarado el cual no siempre logra dar como resultado final el fruto marrón oscuro”.

Ahora los criterios de selección que se tendrán en cuenta para realizar la clasificación de las semillas serían dos (2), las cuales son el color y la forma del fruto que son las características más destacables para poder diferenciar ambas presentaciones del fruto.

El presente proyecto se enfocará en la etapa de selección de frutos marrones oscuros para continuar a la siguiente etapa, ya que actualmente es un proceso que se realiza de forma manual por operarios de las distintas empresas en las zonas rurales de cultivo y procesamiento, es posible realizar una automatización mecatrónica de la misma, para poder entregar dicho producto más rápidamente a los procesos subsiguientes y disminuir tiempos de trabajo generando un mayor valor comercial a todo el proceso. Para la implementación del algoritmo de selección se utilizará un prototipo de clasificadora ya elaborado, al cual se le harán las debidas modificaciones para ajustarse a los requerimientos y necesidades que se presenten acorde al proceso de selección que realice el algoritmo teniendo en cuenta el reemplazo de elementos como sensores y actuadores a implementar.

2 OBJETIVOS

2.1 Objetivo general

- Implementar en un sistema seleccionador de frutos de Sacha Inchi un algoritmo inteligente de visión artificial.

2.2 Objetivos específicos

- Diseñar un algoritmo clasificador basado en técnicas de inteligencia artificial y procesamiento de imágenes para establecer si el producto presenta la apariencia requerida para la etapa posterior al descascarado del fruto de Sacha Inchi.
- Adecuar la maquina prototipo del proyecto “Construcción de un Prototipo de una Máquina Selectora de Semillas de Sacha Inchi” para realizar la implementación del algoritmo de selección.
- Implementar en un sistema embebido los algoritmos de procesamiento y clasificación con el objetivo de automatizar el proceso de selección utilizando la maquina prototipo.
- Validar el funcionamiento del sistema seleccionador de frutos de Sacha Inchi en el prototipo de Máquina selectora.

3 ESTADO DEL ARTE

La calidad de los productos provenientes de la industria alimenticia depende de varios factores, entre ellos se encuentra el estado de estos vegetales, frutas o cualquier producto agrícola, pero también es importante el tiempo que se toma en seleccionarlos o clasificarlos, ya que el estado o fase en que se encuentre, influye directamente en la calidad y tipo de los productos derivados de estos. En la actualidad los procesos de automatización enfocados en la clasificación de productos de frutas y verduras se han ido desarrollando con nuevas tecnologías las cuales se ven acompañadas de sistemas de procesamiento basados en inteligencia artificial, estas tecnologías le permiten a cualquiera de estos procesos productivos ser cada vez más flexibles y sostenibles, lo que se traduce en grandes beneficios para las empresas agrícolas. Lo anterior se encuentra evidenciado en los siguientes proyectos:

En el trabajo de (Paguay Donoso & Valarezo Rodriguez, 2018) , se presenta el diseño e implementación de un prototipo de clasificador de huevos de gallina basado en las imperfecciones de la cascara aplicando visión artificial. Dicho prototipo tenía como centro de procesamiento una minicomputadora RaspberryPi3, que permitía mediante el uso de una cámara, adquirir imágenes de las diversas caras del huevo junto a la aplicación de técnicas de ovoscopia y un arreglo de espejos los cuales garantizaban la cobertura total del área en cuestión. Estas imágenes eran procesadas mediante la implementación de un algoritmo escrito en el lenguaje de programación Python, además de implementar métodos estadísticos y de visión artificial para detectar las anomalías presentes en la cascara del huevo. A partir de este procesamiento y con ayuda de una serie de mecanismos, fueron clasificados y separados los huevos que presentaron anomalías y dando paso solamente a los que no las presentaron, para posteriormente seguir con la etapa de clasificación habitual por pesos y tamaños, usando también otro algoritmo de comparación. Como resultados de este proyecto se obtuvo una reducción en los tiempos de clasificación de aproximadamente 60% con respecto al método de clasificación manual que suele implementarse; también se observó una mejora en la calidad del producto y mayor eficiencia del proceso.

En el artículo de (Weber, y otros, 2014), se presenta una metodología experimental que identifica con mayor precisión y rapidez distintos tipos comerciales de maíz, lo que permitiría orientar con eficiencia los destinos de industrialización del grano. El nuevo sistema todavía se encuentra en fase experimental, distingue en segundos el tipo comercial de maíz, lo que permite fijar de modo rápido y preciso la proporción de los componentes del grano.

En el trabajo de (Alimboyong, Hernandez, & Medina, 2018), se desarrolló un algoritmo capaz de segmentar y clasificar imágenes. La segmentación está

destinada a separar la planta de la imagen original, mientras que la clasificación identifica qué imágenes pertenecen a las dos clases definidas. Se aplica una red neuronal convolucional -CNN sigla en inglés- para discriminar las plantas de maíz de las malezas en tiempo real, en las etapas iniciales de desarrollo del cultivo.

En el artículo de (Ahila Priyadharshini, Arivazhagan, Arun, & Mirnalini, 2019), se detalla la utilización de una CNN y se presenta una arquitectura para la clasificación de la enfermedad de la hoja del maíz. Las CNN propuestas están capacitadas para identificar cuatro clases diferentes (tres enfermedades y una clase saludable).

En el artículo de (Mendoza & Martin, 2014), se presenta el procedimiento llevado a cabo para el reconocimiento de objetivos en imágenes de video en tiempo real. El trabajo se basa en el reconocimiento de patrones mediante el uso de tecnologías de visión por computador por medio de la librería de código abierto OpenCV (Open source Computer Vision) para el tratamiento de imágenes y entrenamiento de clasificadores de objetos.

En el trabajo de (Sevillano Bustos, 2015), se presenta la Implantación de un sistema de visión artificial que permita la clasificación de granos de trigo en función de su morfología y tonalidad. En el cual se desarrolló tanto el algoritmo de control que permitió realizar las inspecciones de los granos, como el código correspondiente a la interfaz de usuario que facilitará el manejo de la aplicación y también el programa del controlador Arduino que controla los distintos accionamientos mecánicos y los comunica con la parte de visión.

En el trabajo de (Moreno, 2019), se presenta un estudio enfocado en las redes neuronales convolucionales y la importancia que tienen en el análisis de imágenes. El objetivo fue de entrenar una red para conseguir un elevado porcentaje de precisión, sensibilidad y especificidad, se desarrolló una investigación de los lenguajes de programación más utilizados en el campo de la Inteligencia Artificial, bases de datos para explotar la información y el funcionamiento propio de las Redes Neuronales Convolucionales.

En el trabajo de (Valdes Avila & Baquero Vanegas, 2019), se realizó una aplicación de Deep Learning, específicamente detección de Viviendas Sin Servicio de energía (VSS) en el mapa satelital del caserío de Media Luna, ubicado en el municipio de Uribia al norte de La Guajira. A partir de un conjunto de datos que conforman imágenes satelitales de viviendas de diferentes zonas rurales de Colombia, obtenidas a través de Google Earth, se realizaron dos modelos de predicción diferentes, se hizo una comparación de estos modelos, con el objetivo de obtener una minimización del error de predicción.

En el trabajo de (Castellanos Castilla & Castañeda Dias, 2017) , se plantea la construcción de una máquina que permite la clasificación automatizada del fruto por medio de visión artificial con una metodología de reconocimiento de patrones de colores en el fruto, permitiendo así la inspección visual del mismo facilitando la clasificación.

En el trabajo de (Dorado Esther, 2019), que consistió en el diseño, implementación y evaluación de diferentes arquitecturas de redes neuronales convolucionales aplicadas al control de calidad de naranjas durante la fase de clasificación antes de su encajado. Estas redes neuronales debían detectar los pezones, los culos y los defectos sobre las imágenes de naranjas capturadas en una línea de producción.

4 MARCO TEORICO

4.1 Generalidades del fruto Sacha Inchi

Sacha inchi (*Plukenetia volubilis* Linneo) es una planta que pertenece a la familia de la Euphorbiaceae, originaria de la Amazonía, conocida como, sacha inchi, amui, sacha yuchi, sacha yuchiqui, sampannankii, suwaa, maní del monte, sacha maní, maní del inca, maní jibaro o inca peanuts (Alvarez & Rios, 2007).

4.1.1 Características del género y la familia

Este género pertenece a la familia de la Euphorbiaceae, se caracteriza porque presenta plantas con una importancia económica ya que se obtienen productos tales como el caucho, la tapioca, ceras y aceites. Este género comprende 17 especies, 12 en América, 3 en África, una en Madagascar y una en Asia (Jimenez, G.M, & Cruz, 2000).

4.1.2 Morfología

Es una planta trepadora, voluble, semileñosa con hábito de crecimiento indeterminado; en cuanto a sus hojas, son alternas de forma acorazonada de 10 a 12 cm de largo y de 8 a 10 cm de ancho, elípticos, aserrados y con pecíolos de 2 a 6 cm de largo. Las nervaduras nacen en la base de la nervadura central orientándose al ápice (Tasso et al., 2013). En cuanto al fruto, tiene forma de estrella, con número variable de lóbulos que pueden ir desde cuatro hasta ocho, predominando aquellos que tienen cuatro o cinco lóbulos como se puede observar en la ilustración 3. Estos frutos se dividen cuando está madura y se diferencia, endureciendo sus paredes. Cuando el fruto se encuentra maduro, dentro del fruto se hallan las semillas que son de color marrón oscuro, corrugadas y venadas, de forma lenticular y con 1.5 a 2 cm de diámetro (Gomez, 2004).

Ilustración 3 Fruto de sacha inchi 4,5,6 y 7 lóbulos



Fuente: <https://docplayer.es/82360873-El-sacha-inchi-plukenetia-volubilis-y-los-aceites-funcionales-omega.html>

4.1.3 Distribución geográfica

La planta de sacha inchi se ha encontrado desde América Central hasta Bolivia, en América del Sur se ha registrado en la Amazonía peruana, boliviana y en las Antillas. En el Perú están distribuidos principalmente en las zonas

selváticas de los departamentos de San Martín, Ucayali y Loreto (Gómez & Torres, 2007; Chirinos et al, 2009). En Colombia se encuentra en estado silvestre en diversos lugares de la Orino-Amazonia y en el pacífico y como cultivo establecido se ha reportado en el departamento del Chocó, en el Putumayo, Caquetá y en el Amazonas; se han registrado cultivos de sacha inchi en Fresno Tolima y Tibacuy Cundinamarca (Karisma , 2015).

Según Agronet (Agronet, 2015), los mayores productores de *P. volubilis* son Putumayo y Choco. Putumayo con un área de 33 hectáreas y Chocó con una hectárea. Cada departamento presento un rendimiento de 0,5 toneladas de semilla hectárea año.

4.1.4 Características nutricionales.

Los principales componentes del Sacha Inchi son proteínas, aminoácido, ácidos grasos esenciales (omegas 3, 6, y 9) y vitamina E (tocoferoles, tocotrienoles). El aceite del Sacha inchi es el más rico en ácidos grasos insaturados, entre las semillas oleaginosas.

La composición en ácidos grasos del aceite de sacha presenta en su composición un alto contenido de insaturaciones, un 9.39% por ácidos grasos monoinsaturados y un 83.3% por ácidos grasos poliinsaturados, el perfil de ácidos grasos en conjunto con las características fisicoquímicas y la viabilidad en la obtención de la materia prima, hacen del aceite de sacha inchi una alternativa con un amplio rango de aplicaciones a nivel industrial y alimenticio, como un producto nutracéutico (Sanchez S. L., 2012)

4.1.5 Beneficios para la salud

En general podemos afirmar que el consumo continuo puede mejorar o ayudar en las siguientes patologías:

- **Corazón:** previene las arritmias, reduce el riesgo de infartos y problemas vasculares. Equilibra el colesterol y los triglicéridos.
- **Cerebro:** mantiene e incrementa las capacidades cerebrales, mejora la irrigación cerebral, mejora la memoria, mejora el funcionamiento del sistema nervioso central (demencia, depresiones, stress). Ayuda a mejorar a los pacientes con Alzheimer.
- **Sistema digestivo:** facilita y acelera el metabolismo. Evita el estreñimiento.
- **Procesos inflamatorios:** nivela la presión arterial, contrarresta el envejecimiento e inflamación de los pulmones, ayuda a inhibir la formación de grasas en el hígado y limita la producción de sustancias pro inflamatoria, reduce la sintomatología de enfermedades

inflamatorias como: inflamación intestinal, artritis reumatoide, colitis ulcerosa, asma, neumonía viral y bacteriana.

- **Gestación, recién nacidos y niños:** disminuye el riesgo de desarrollar hipertensión al embarazo, esencial para el desarrollo neurológico del feto, esencial para el desarrollo del tejido nervioso del feto en el tercer trimestre, mejora el desarrollo psicomotor de los recién nacidos, efectos pasivos en el desarrollo mental, esencial para el desarrollo infantil.
- **Efectos anticancerígenos:** reduce el crecimiento de las células cancerígenas humanas.

4.1.6 Productos derivados del Sacha Inchi

La semilla de sachá inchi es el insumo que se utiliza para la producción industrial de aceite virgen, torta y harina proteica. Como usos podemos resumirlo como (Sanchez & Amiqueri, 2004):

1. Planta entera
 - Alimento para animales (henificación): vacunos, pavos, etc.
 - Hojas verdes: hortalizas, harina verde.
 - Abono seco.
 - Fuente de ceniza.
2. Fruto
 - a) Cápsula o cáscara
 - Cama para vacunos, ovinos, porcinos, aves de corral, etc.
 - Fertilizante: Molienda fina.
 - En paneles de construcción de paredes.
 - Fermentación industrial: obtención de alcohol y acetona.
 - Combustible.
 - b) Almendra o semilla
 - Cake o torta.
 - Harina: panificación en general, galletas, alimentos dietéticos, alimentos para niños y ancianos, base de diversos platos típicos (guisos, sopas, salsas, bebidas proteicas, etc.).
 - Alimento para animales: vacunos, aves, porcinos, ovinos, caprinos, cuyes, conejos, etc.
 - c) Aceite
 - Cake o torta.
 - Comestible: margarina, mayonesa, ensaladas, aceite de mesa, salsa de ají, en cocina general, quesos, pescado enlatado.
 - Industrial: pomadas, velas, cerosota, pasta para zapatos, pinturas, tinta, plásticos, adhesivos, linazas, brilladores de metal, barnices de madera, crema de afeitar, Shampoos, blanqueadores, aceites para lanificación, lubricantes y en iluminación.

- Medicinal: alimento funcional (omega 3 y 6), frotaciones antirreumáticas, bronquitis asmáticas, reconstituyente funcional.
- Cosméticos: base natural para cosméticos orgánicos naturales, rejuvenecedores de la piel.
- Almendra o semilla entera: alimentación en general, alimentos para desayunos, saltados, salditos, turrone, confitados, pudines, maní fresco, cereales de maní, guisos, granulados, polvos sazonadores, etc.
- Almendra entera tostada: leche, milk shake, quesos, pastas (mantequillas, caramelos, confitados, turrone, polvos de mantequilla, etc.).

4.1.7 El aceite del Sacha Inchi.

El proceso industrial de extracción del aceite (ilustración 4) se realiza en frío y se busca evitar las altas temperaturas. Este aceite es utilizado como aceite de mesa y de cocina, en alimentos enriquecidos con omega producidos industrialmente, en cosméticos y en medicina.

Ilustración 4 Aceite orgánico de Sacha inchi



Fuente: <https://sumasacha.com/wp-content/uploads/2022/07/sacha-inchi-oil-3.png>

4.1.8 Proceso de extracción del aceite del Sacha Inchi.

Para la extracción de aceites en semillas oleaginosas existen tres métodos comúnmente utilizados, los cuales son (Gomez V, 2011):

- **Extracción por solventes:** Esta es la acción de separar con un líquido una fracción específica de una muestra. Sus principales inconvenientes son los elevados costos de instalación inicial. Aunque este proceso presenta una alta eficiencia. Este tratamiento requiere un pre tratamiento: triturar, calentar con vapores a alta temperaturas.
- **Prensado discontinuo:** en estas se realizan la extracción del aceite a altas presiones. Se utiliza generalmente un pistón de acción intermitente.
- **Prensado continuo:** es básicamente una prensa de jaula, en la que, la presión se desarrolla por medio de un eje rotatorio o tornillo sinfín. En

esta jaula se desarrolla una presión extremadamente alta, del orden de 140 a 2800 Kg/cm².

El subproducto agroindustrial que se obtiene después del proceso de prensado le denominan torta. Estudios han demostrado que su alto contenido de aminoácidos esenciales junto con su alta digestibilidad, muestran que gracias a su composición nutricional esta puede ser aplicado como alimento o suplemento alimenticio para la preparación de mezclas nutritivas. Además, esta representa una alternativa de la torta de soya en la actividad avícola y pecuaria, gracias que por su alto valor y calidad proteínico (O., 2013).

4.2 Visión artificial

El aumento de la competitividad es un requerimiento primordial en la situación económica actual. Un factor clave para mejorar la competitividad es aumentar la productividad incorporando nuevos sistemas de inspección que permitan la automatización de diferentes procesos. Los sistemas de inspección visual automática, basados en visión por computador, han demostrado ser una herramienta fundamental para mejorar los procesos. Estos sistemas de visión permiten la inspección continua, evitando fatigas y distracciones, y facilitando la cuantificación de las variables de calidad en prácticamente el 100% de la producción. Esto se traduce, no solo en una mejora de la calidad final de los productos, sino también en un ahorro en términos económicos y medioambientales. (Dorado E. d., DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED NEURONAL CONVOLUCIONAL PARA LA DETECCIÓN DE PUNTOS PRINCIPALES EN NARANJAS, 2019-2020)

La visión artificial, o también denominada visión por computador, es considerada una rama de la inteligencia artificial y su principal propósito es que las máquinas sean capaces de interpretar el entorno que las rodea tal y como lo hacemos nosotros mediante el sentido de la vista. (Sevillano B., 2015)

Estas son algunas ventajas que presenta la visión humana frente a la visión artificial y viceversa:

- **Sistema humano:**
 - Mejor reconocimiento de objetos.
 - Mejor adaptación a situaciones imprevistas.
 - Utilización de conocimiento previo.
 - Mejor en tareas de alto nivel de proceso.
- **Sistema artificial:**
 - Mejor midiendo magnitudes físicas.
 - Mejor para la realización de tareas rutinarias.
 - Mejor en tareas de bajo nivel de proceso.

4.3 Tipos de iluminación

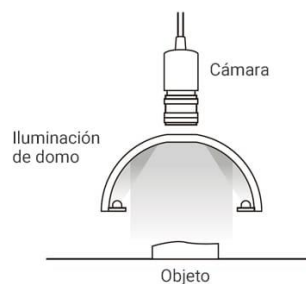
Uno de los elementos que conforman los sistemas de visión artificial es la fuente de iluminación. Su importancia es muy grande ya que de ello dependerá el que tengamos una buena imagen y no tengamos que mejorarla vía software. De esta forma el tiempo de procesado de la imagen es mucho menor y conseguiremos una mayor frecuencia de análisis.

Usualmente el color más apropiado para usar como fuente de iluminación es el blanco ya que como en su mayoría se va a trabajar con imágenes en color no conviene alterar la escena con luces de otros colores.

4.3.1 Iluminación difusa tipo domo

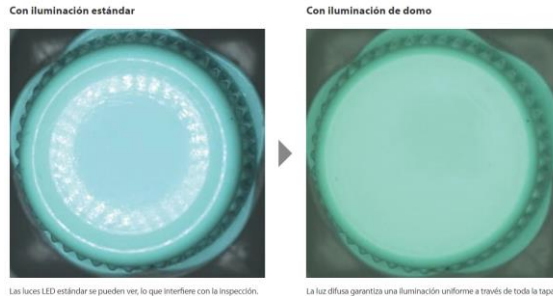
La iluminación de domo irradia luz indirecta en varios ángulos, a causa de los LEDs que están montados en la superficie y brillando hacia el interior del domo como se observa en la ilustración 5. Cuando se utiliza iluminación estándar, un reflejo directo de la pieza puede resultar en un punto resplandeciente (halo), causando problemas de inspección y fiabilidad. Como la iluminación de domo se difunde en todas las direcciones, un objeto con una forma complicada puede ser iluminado de manera uniforme con luz difusa desde todos los ángulos eliminando los puntos resplandecientes. Dado que el objeto está uniformemente iluminado, se puede establecer una inspección fiable y estable obteniendo como resultado una imagen como la que se observa en la ilustración 6.

Ilustración 5 Iluminación tipo domo



Fuente: https://www.keyence.com.mx/Images/peripheral_ca_dd_img_02_1863120.jpg

Ilustración 6 Aplicación de iluminación de domo a una tapa de plástico

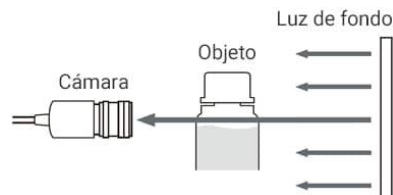


Fuente: https://www.keyence.com.mx/ss/products/vision/peripheral/ca-d/ca_dd.jsp

4.3.2 Iluminación a contraluz o de fondo

La iluminación frontal estándar puede producir un contraste inconsistente y pobre, de las características de un objeto debido a su forma, color y acabado superficial. El uso de una luz de fondo asegura la transmisión de la luz a través del objeto hacia la cámara como se observa en la ilustración 7. Si el objeto es opaco, se genera una silueta creando un fuerte contraste y contorno para una inspección estable. Incluso si la muestra es transparente puede presentar un rango de densidades (tales como materiales en capas o líquido en una botella). Aun así, se puede lograr un contraste a partir de los cambios de absorción y transmisión de la luz por el objeto como se observa en la ilustración 8.

Ilustración 7 Iluminación de fondo



Fuente: https://www.keyence.com.mx/Images/peripheral_ca_ds_img_07_1863191.jpg

Ilustración 8 Aplicación de iluminación de fondo a una regleta de pines conectores



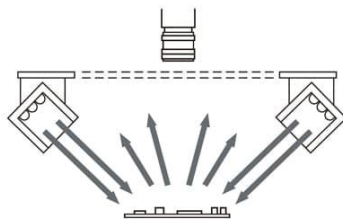
Fuente: https://www.keyence.com.mx/ss/products/vision/peripheral/ca-d/ca_ds.jsp

4.3.3 Iluminación cuadrada o directa

Las luces cuadradas se componen de cuatro barras de luz, que pueden ser controlados y posicionadas individualmente.

El uso de cuatro iluminaciones de barra en una disposición cuadrada permite un ajuste en cuatro sentidos independientes. De acuerdo con el perfil del objeto se puede configurar la iluminación óptima para resaltar y extraer características específicas como se observa en la ilustración 9.

Ilustración 9 Iluminación cuadrada o directa



Fuente: https://www.keyence.com.mx/Images/peripheral_ca_dq_img_03_1863142.jpg

Ilustración 10 Aplicación de iluminación cuadrada en un CI



Fuente: https://www.keyence.com.mx/ss/products/vision/peripheral/ca-d/ca_dq.jsp

Este último sería el sistema de iluminación implementado para el presente proyecto.

4.4 Redes neuronales convolucionales para detección de objetos

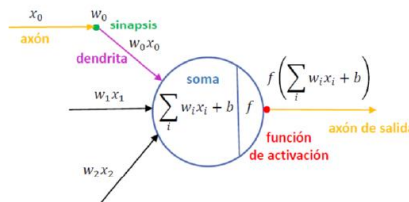
A continuación, se presentan los fundamentos teóricos en los que se basa este proyecto. Se detallará en que consiste la unidad básica de procesamiento de una red neuronal, conocida como neurona, los tipos de estructuras de las redes neuronales, así como las redes neuronales convolucionales, ampliamente utilizadas en el campo de la detección de objetos.

4.4.1 La neurona

Una red neuronal consta de unidades básicas de procesamiento, las neuronas y conexiones ponderadas entre esas neuronas (Kriesel, 2007)

Reciben el nombre de neuronas artificiales por su similitud a las neuronas biológicas. De forma similar a una neurona biológica, estas neuronas tienen conexiones de entrada a través de las que reciben estímulos externos, las variables de entrada. Con estos valores la neurona realizará un cálculo interno y generará un valor de salida. Internamente, la neurona utiliza todos los valores de entrada para realizar una suma ponderada de ellos. La ponderación de cada una de las entradas viene dada por el peso que se le asigna a cada una de las conexiones de entrada. Es decir, cada conexión que llega a la neurona tendrá asociado un valor que definirá con qué intensidad cada variable de entrada afecta a la salida de la neurona.

Ilustración 11 Estructura de una neurona artificial



Fuente: Autor de los apuntes (Dorado E. , 2020)

Como se observa en la ilustración 11, las neuronas constan de los siguientes componentes:

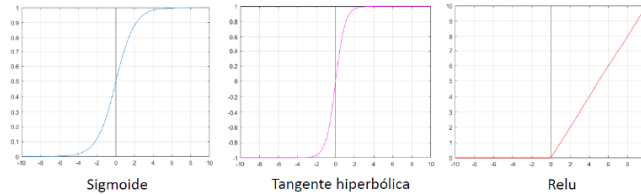
- Variables de entrada, X_i .
- Pesos, uno por cada variable de entrada, W_i . Estos pesos determinarán la importancia de cada conexión.
- Sesgo o bias en la literatura inglesa. Es el término independiente de la función. El sesgo indicará la facilidad que tiene la neurona de activarse. La neurona permite obtener valores de salida continuos, no binarios. Para que la salida sea binaria se comparará con un determinado umbral, siendo 0 si el valor de la salida es menor o igual que el del umbral y 1 si es mayor que él. El valor del sesgo será el opuesto al umbral.
- Función de propagación, consiste en la suma ponderada de las variables de entrada con sus respectivos pesos y el término de sesgo. En el caso más simple y general esta función de propagación es lineal y tiene la siguiente fórmula matemática:

$$y = \sum_{i=1}^N w_i x_i + b$$

- Función de activación, recibe la salida de la función de propagación y determinará si se activa la salida de la neurona, $z = f_{act}(y)$. Las más comunes son la sigmoide, la tangente hiperbólica y la función Relu

(Unidad Lineal Rectificada) representadas en la ilustración 12. Gracias a ella tiene sentido encadenar varias neuronas.

Ilustración 12 Funciones de activación



$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

$$f(x) = \tanh(x) = 2\sigma(2x) - 1$$

$$f(x) = \max(0, x)$$

Fuente: Autor de los apuntes (Dorado E. , 2020)

La función de activación Sigmoide hace que los valores muy grandes se saturan en 1 y los valores muy pequeños en 0. Se emplea en modelos en los que tenemos que predecir la probabilidad como una salida en el rango de 0 a 1.

La función de activación Tanh es como la función Sigmoide, pero en el rango de -1 a 1.

La función RELU se comporta como una función lineal cuando es positiva y constante a 0 cuando el valor de entrada es negativo. Esta función tiene algunas ventajas sobre el resto, es menos costosa y converge más rápidamente que Sigmoide o Tanh.

4.4.2 Estructura de las redes neuronales

Las redes neuronales están formadas por neuronas interconectadas, estas neuronas pueden agruparse siguiendo diferentes estructuras. Las más comunes son las redes recurrentes y las redes feedforward, estas últimas son las más utilizadas.

Redes feedforward

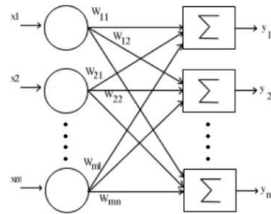
Esta estructura consiste en una capa de entrada, una serie de capas ocultas y una capa de salida donde la información se mueve en una única dirección, de la capa de entrada hacia la de salida. La capa de entrada en realidad no es una capa compuesta de neuronas, sino una forma de representar las entradas a la primera capa oculta, la cual es la primera capa que consta de neuronas.

Dentro de las redes feedforward dependiendo del número de capas, podemos distinguir la red neuronal monocapa o perceptrón simple y la red neuronal multicapa o perceptrón multicapa.

La red neuronal monocapa (ilustración 13) se corresponde con la red neuronal más simple, las entradas están conectadas directamente a las salidas donde

se realizan los diferentes cálculos. (Calvo, Clasificación de redes neuronales artificiales, 2017)

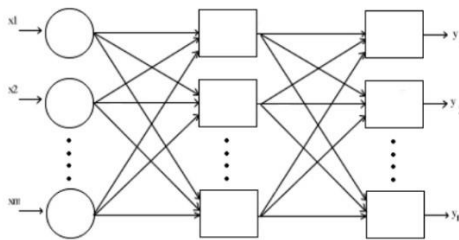
Ilustración 13 Representación gráfica de una red neuronal monocapa



Fuente: <https://sistemasinteligentesuss.blogspot.com/2016/07/redes-neuronales-las-redesneuronales.html>

La red neuronal multicapa (ilustración 14) es una generalización de la red neuronal monocapa, la diferencia reside en que mientras la red neuronal monocapa está compuesta por una capa de neuronas de entrada y una capa de neuronas de salida, esta dispone de un conjunto de capas intermedias, llamadas capas ocultas, entre la capa de entrada y la de salida (Calvo, Clasificación de redes neuronales artificiales, 2017)

Ilustración 14 Estructura de una red neuronal multicapa

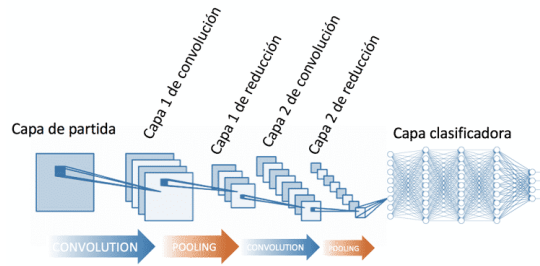


Fuente: <https://sistemasinteligentesuss.blogspot.com/2016/07/redes-neuronales-las-redesneuronales.html>

Red neuronal convolucional (CNN)

La principal diferencia de la red neuronal convolucional (ilustración 15) con el perceptrón multicapa viene en que cada neurona no se una con todas y cada una de las capas siguientes, sino que solo con un subgrupo de ellas, con esto se consigue reducir el número de neuronas necesarias y la complejidad computacional necesaria para su ejecución. (Calvo, Clasificación de redes neuronales artificiales, 2017)

Ilustración 15 Representación gráfica de una CNN

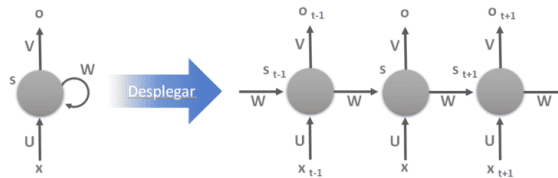


Fuente: <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>

Red neuronal recurrente (RNN)

Las redes neuronales recurrentes permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos como se observa en la ilustración 16, con esto se consigue crear la temporalidad permitiendo que la red tenga memoria (Calvo, Clasificación de redes neuronales artificiales, 2017).

Ilustración 16 Representación gráfica de una RNN



Fuente: <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>

4.5 Redes neuronales convolucionales

Las redes neuronales convolucionales (Convolutional Neural Networks o CNN) son un algoritmo de DeepLearning que está diseñado para trabajar con imágenes, tomando estas como entrada, asignándole pesos a ciertos elementos en la imagen para así poder diferenciar unos de otros. Las redes convolucionales contienen varias capas ocultas (capas que se encuentran entre la capa de entrada y la capa de salida), en las primeras capas se extraen características con más detalle como pueden ser los bordes o los colores, mientras que en las capas más profundas se extraen características cada vez más abstractas. Las tareas comunes de este tipo de redes son la detección o categorización de objetos, clasificación de escenas y clasificación de imágenes en general (Silva & Freire, 2019).

En este tipo de redes las neuronas de las capas que la componen no se unen con todas las neuronas de la capa siguiente, sino sólo con alguna de ellas. Esto simplifica el aprendizaje de la red y reduce los costes computacionales (Calvo, Clasificación de redes neuronales artificiales, 2017).

Los algoritmos de detección de objetos distinguen dos fases: la extracción de características de la imagen y la búsqueda de objetos basada en esas características para su clasificación. Las CNN realizan estas dos fases de forma paralela, facilitando el aprendizaje de la red.

Estas redes pueden ser entrenadas desde cero con un amplio conjunto de imágenes etiquetadas mediante el cual la red aprenderá las características de los datos etiquetados. Esto puede tener un alto coste computacional y el tiempo de entrenamiento puede llegar a ser muy elevado.

La estructura de las CNN, se compone de tres tipos de capas, capa de convolución, capa de reducción o pooling y capa totalmente conectada o fullyconnected. Las neuronas que componen una CNN están especializadas en dos operaciones, la operación de convolución y la operación de reducción o pooling.

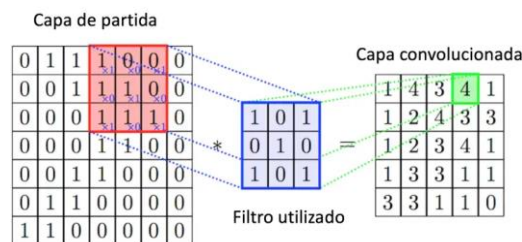
4.5.1 Convolución

La red toma como entrada una imagen y la trata como una matriz de dimensiones número de píxeles vertical x número de píxeles horizontal x número de canales de color. Si la imagen es en color el número de canales será 3, RGB.

Para realizar la convolución se necesita un filtro que se encargara de extraer las características de la imagen, este filtro se denomina Kernel. En el caso de que la imagen sea RGB se tendrán 3 Kernel del mismo tamaño que se sumaran para obtener una imagen de salida.

La convolución consiste en desplazar el Kernel por cada uno de los elementos de la imagen original y realizar el producto escalar entre la matriz y el filtro obteniendo así el mapa de características.

Ilustración 17 Convolución de un Kernel sobre una imagen



Fuente: <https://www.diegocalvo.es/red-neuronal-convolucional/>

Como se observa en la anterior ilustración 17 la salida no tiene el mismo tamaño que la entrada.

Existen varios parámetros que nos permiten controlar las dimensiones del mapa de características de salida:

- **El tamaño del Kernel**, así como el número de ellos, este parámetro indicará la profundidad de la salida. Por ejemplo, si tenemos 6 filtros, obtendremos 6 mapas de activación separados.
- **El parámetro stride**: Cuando la convolución del Kernel con la imagen no se realiza en todos los píxeles, sino que se hace cada cierto número de ellos. Este paso se denomina stride y cuanto mayor sea más pequeña será la salida.
- **Zero padding**: Consiste en agregar píxeles de valor cero alrededor de la imagen original, de esta forma, el mapa de características podrá tener el mismo tamaño que la imagen de entrada.

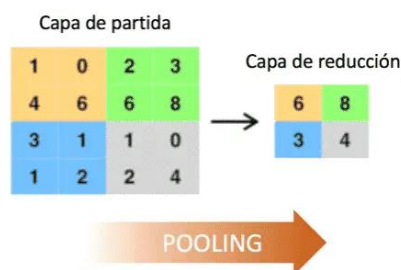
A medida que se entrena la red, los valores de los Kernels o lo que es lo mismo, los pesos de las neuronas se van actualizando para disminuir el error en la predicción.

Después de aplicar la convolución se les aplica a los mapas de características una función de activación (Calvo, Red Neuronal Convolutiva CNN, 2017).

4.5.2 Pooling

En las capas de reducción o pooling se disminuyen las dimensiones espaciales del mapa de características, esto no afecta a la dimensión de profundidad como se observa en la ilustración 18. El objetivo es hacer una simplificación de la información y crear una versión condensada de las características más importantes. La forma más común se conoce como Max pooling, conserva el valor máximo de una región fija del mapa de características (Calvo, Red Neuronal Convolutiva CNN, 2017).

Ilustración 18 Max pooling



En resumen, el pooling se utiliza para reducir la dimensionalidad espacial de los datos anteriores además de conseguir que el sistema sea invariante a pequeñas traslaciones de la entrada, es decir, ante pequeños cambios en la posición de la imagen.

4.5.3 Capa totalmente conectada o fully connected

Después de varias capas sucesivas de convolución y pooling se añaden una o más capas totalmente conectadas. Se trata de una capa en la que todos los resultados de la capa anterior están conectados a todos los nodos de la capa siguiente

4.5.4 Capa de clasificación

La última capa de esta red es una capa clasificadora que tendrá tantas neuronas como número de clases se quieran predecir. La salida de esta capa será un vector que indique la probabilidad de una subimagen de la imagen de entrada de pertenecer a una clase concreta.

4.5.5 Arquitecturas de las CNN

Existen algunas arquitecturas de redes neuronales convolucionales, las más utilizadas son las siguientes:

- **LeNet:** Fue la primera aplicación exitosa descrita por Yann LeCun. El sistema se desarrolló para su uso en un problema de reconocimiento de caracteres a mano.
- **AlexNet:** Fue un modelo creado por Alex Krizhevsky para el concurso ILSVRC que tiene por objetivo clasificar fotografías de objetos. Algunos de los métodos novedosos empleados en el modelo AlexNet fueron la utilización de funciones de activación ReLU después de cada capa convolucional, el uso de la función de activación softmax en la capa de salida que actualmente es un elemento básico para la clasificación de múltiples clases con redes neuronales y la aplicación de maxpooling en lugar de la agrupación media. Otra novedad fue la aplicación de capas convolucionales, anteriormente cada capa convolucional estaba seguida de una capa de pooling.
- **VGGNet:** Fue implementada por Karen Simonyan y Andrew Zisserman. La primera diferencia con las arquitecturas previamente definidas es el uso de un gran número de filtros pequeños.
- **Redes residuales ResNet:** Esta innovación fue propuesta por Kaiming He en 2016. Este modelo utiliza conexiones residuales que permiten

conexiones directas entre capas no consecutivas, se saltan las capas intermedias (sitiobigdata, 2019).

4.5.6 Parámetros e hiperparámetros

Las redes neuronales constan de parámetros e hiperparámetros, los primeros son las variables que se estiman durante el proceso de entrenamiento, es decir, los pesos de las neuronas. Los hiper- parámetros son variables que pueden ser configuradas por el usuario para ajustar los algoritmos de aprendizaje, algunos de ellos son los siguientes:

- **Número de capas de la red:** Cuantas más capas contenga la red mayor será el tiempo de cálculo y su capacidad de aprendizaje.
- Número de neuronas por capa.
- **Número de épocas:** Cada época representa el paso de todos los datos de entrenamiento por la red neuronal.
- **Tamaño del batch:** El conjunto de datos se divide en lotes más pequeños, batches, el tamaño del batch indica el número de ejemplos de entrenamiento que contiene cada lote.
- Tamaño de los filtros empleados.
- **Tasa de aprendizaje o Learning rate:** Este hiperparámetro regula la velocidad de aprendizaje, indica cuanto afecta el gradiente a la actualización de nuestros parámetros en cada iteración.
- **Tipo de optimización:** los algoritmos de optimización empleados pueden ser el descenso de gradiente, descenso de gradiente estocástico o descenso de gradiente con momentum entre otros.
- Función de activación
- Función de coste o perdidas

5 METODOLOGIA

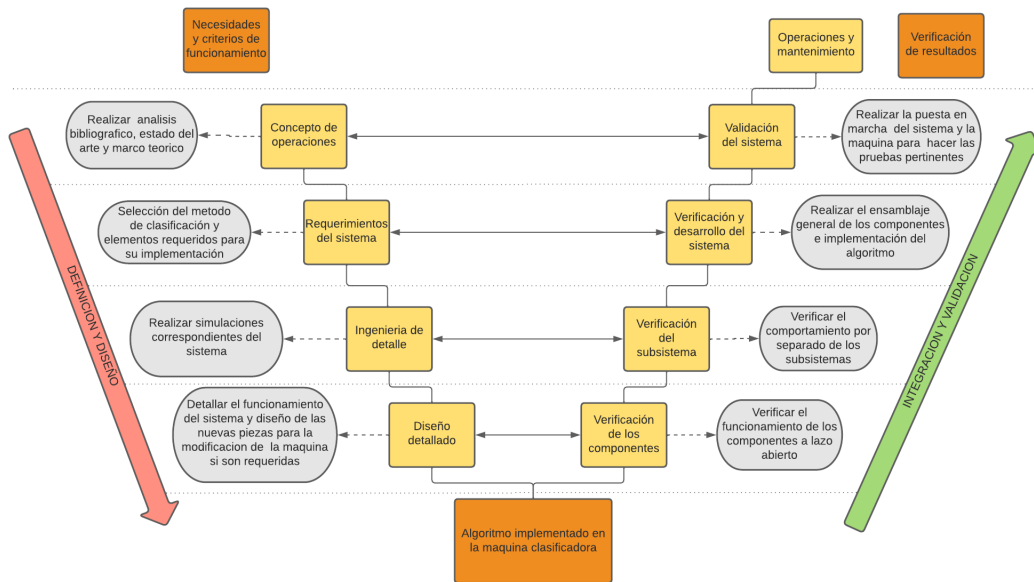
La metodología escogida es un modelo de diseño originalmente creado para la planeación y el desarrollo de software. El modelo V (ilustración 19) resume los pasos principales a ser llevados a cabo en conjunto con los entregables dentro del marco de la validación de los sistemas mecatrónicas.

Esta metodología se divide en diferentes niveles:

El primer nivel es el de funcionalidad y en este se realiza la investigación de proyectos similares y de materiales e instrumentos que pueden empleados para el desarrollo de nuestro proyecto y a partir de esto plantea una solución del problema. El segundo nivel es el de sistema y en este se realiza el diseño mecánico y electrónico a partir de la solución planteada.

El tercer nivel es el de subsistema, el cual se basa en el desarrollo del software que va a controlar la máquina. El cuarto nivel es el de componentes el cual se basa en el montaje del sistema, es decir, construcción mecánica, cableado, implementación de la instrumentación. Cada nivel tiene una parte de validación, la cual va de manera ascendente desde el nivel de componentes hasta el nivel de funcionalidad.

Ilustración 19 Metodología basada en el modelo V



Fuente: autor del proyecto

A continuación, se describe cada una de las fases del modelo en V:

5.1.1 Concepto de operaciones.

A partir de las necesidades y criterios de funcionamiento definidos, se plantean las directrices generales del proyecto. Se investigan sobre proyectos similares, para obtener información útil que sirva para poder desarrollar el proyecto.

5.1.2 Requisitos del sistema.

Se realizan pruebas con códigos basados en sistemas de visión artificial y clasificación de objetos para poder definir cuales se implementarán, también se determinan los elementos necesarios para su implementación.

5.1.3 Ingeniería de detalle.

En este paso se verifica el funcionamiento del método de clasificación seleccionado, probándolo en conjunto con las cámaras web y también se realizan pruebas de códigos que permitan controlar los actuadores y sensores a implementar, lo cual requiere la elaboración de conexiones eléctricas.

5.1.4 Diseño detallado.

A partir de las modificaciones requeridas se definen la posición de montaje de los actuadores y sensores a implementar, así como también las conexiones eléctricas y demás elementos.

5.1.5 Verificación de los componentes.

Una vez adquiridos los componentes se procede a la verificación de los componentes por separado, realizando una prueba de su funcionamiento a lazo abierto.

5.1.6 Verificación del subsistema.

Para este nivel se divide la máquina clasificadora en subsistemas y realizan pruebas de los diferentes subconjuntos y el comportamiento de cada elemento en su posición.

5.1.7 Verificación y desarrollo del sistema.

Se realiza el montaje de todos los elementos requeridos para el funcionamiento del sistema general.

5.1.8 Validación del sistema.

Se realiza la puesta en marcha de la máquina clasificadora con semillas de muestra y se verifica el funcionamiento de todas las etapas para corregir posibles inconvenientes y errores que se puedan presentar.

5.1.9 Operaciones de mantenimiento.

En caso de cualquier falla mecánica, eléctrica o del funcionamiento del algoritmo implementado, se realiza su respectiva revisión para dar con la solución pertinente.

6 INSTRUMENTACIÓN

6.1 Dispositivo de captación de imágenes

Para la etapa de captación de imágenes la cual sería de las más importantes para el proceso de selección de semillas, debido a que evidentemente se requiere obtener muestras de las semillas que se irán procesando y posteriormente utilizarlas para el procesamiento de imágenes y aplicación del algoritmo de visión artificial implementado.

En vistas que se requería una cámara, se realizó una investigación sobre distintas cámaras webs (ver Tabla 1) y sus características que están detalladas en la Tabla 2:

Tabla 1 Cámaras web investigadas



Tabla 2 Características de cámaras web investigadas

Características	Auto Focus VS18	OEM X12-2K	Logitech C525	HAVIT FHD 4K 8 MP HV-NM07P
Campo de visión	90°/360°	120°	69°/360°	69°/360°
Resolución	8 Mpx, enfoque automático	8Mpx	8 Mpx	8Mpx, enfoque automático
Resolución video	3264 x 2448 2k	2K 2560X1440	HD 720p	640x480 - 3264x2448
FPS	16	30	30	15
Precio	\$ 90.250	\$ 90.000	\$140.000	\$160.000

En esta investigación se enfocó en características tales como la resolución enfocando en que fuese de 8Mpx o más y el costo no fuese muy elevado. En medio de la investigación se encontró un aspecto que fue bastante interesante y es el sistema de Autoenfoque el cual mejora bastante la resolución que tiene

la imagen final, convirtiéndose en otra característica importante al momento de elegir la cámara a obtener.

Se decidió a partir de esta investigación y con la elaboración de un QFD (Tabla 3) basado en características como: campo de visión, resolución, Disponible Autoenfoco y costo, dando como mejor opción la Cámara Web Autofocus VS18, ya que entre las opciones que se plantearon es la que más se adecua para la realización del proyecto.

Tabla 3 Matriz QFD de las cámaras web

Customer importance rating	Función	Auto Focus	OEM	Logitech C525	HAVIT FHD 4K 8 MP HV-NM07P
5	Campo de visión	9	3	3	9
5	Resolución	9	9	9	9
3	Autoenfoco	9	1	1	9
3	Costo	9	9	3	3
	Technical importance score	144	90	72	126

El QFD realizado y los siguientes a este tiene como parámetros el “Customer importance rating” el cual valora cada una de las funciones o especificaciones técnicas de dichos elementos, esta se toma como 1 (menor importancia) hasta 5 (mayor importancia), también los valores internos de calificación con números de color azul de cada característica se valoran como 1 (poca o débil valoración), 3 (mediana o moderada valoración) y 9 (alta o gran valoración) los anteriores valores se multiplican por su respectivo rating y luego se suman para determinar el puntaje total de cada elemento.

6.2 Dispositivo de procesamiento y control

Para la elección del dispositivo de procesamiento y control se comparan las características principales (Ver Tabla 4), que en este caso son: memoria RAM, número de puertos, lenguaje de programación, alimentación y costo. Por lo que con la matriz QFD (Ver Tabla 5) realizada se concluye que la Raspberry pi 4 fue la que mejor se adaptaba a las condiciones del sistema, ya que para realizar el análisis del estado del fruto de Sacha Inchi se requiere un alto nivel de procesamiento.

Tabla 4 Características de unidad de procesamiento

Características	Raspberry Pi 2	Raspberry Pi 3	Raspberry Pi 4	Arduino Mega 2560	Arduino uno
Puertos	4 USB 2.0, Ethernet, HDMI, USB 3.0	4 USB 2.0, Ethernet, HDMI, USB 3.0	2 USB 2.0, Ethernet, 2 micro HDMI, 2 USB 3.0	USB 2.0, 4 UARTS	USB 2.0
RAM	1 GB	1 GB	4 GB	8 KB	8 KB
Lenguaje de programación	C, C++, PHP, Ruby	C, C++, PHP, Ruby	C, C++, PHP, Ruby	Flash, processing, MaxMSP	Flash, processing, MaxMSP
Alimentación	5 V DC	5 V DC	5 V DC	5 V DC	5 V DC
Precio	\$ 115.590	\$ 200.000	\$ 359.900	\$157.086	\$107.884

Tabla 5 Matriz QFD de la unidad de procesamiento

Customer importance rating	Función	Raspberry Pi 2	Raspberry Pi 3	Raspberry Pi 4	Arduino Mega	Arduino UNO
5	Memoria RAM	3	3	9	1	1
3	Numero de puertos	3	3	3	3	1
4	Lenguaje de programación	3	3	3	9	9
2	Alimentación	3	3	3	3	3
3	Costo	9	9	3	3	9
	Technical importance score	69	69	81	65	77

Además, se tuvieron en cuenta los más comúnmente usados en proyectos parecidos y se consultaron sus características tomando como referencia también las conclusiones finales de dichos proyectos referente a mejoras en esta etapa de procesamiento y control, dejando al final como mejor elección la Raspberry Pi4. Esta selección fue hecha teniendo en cuenta la cantidad de memoria disponible de las distintas opciones, la Raspberry Pi4 posee 4GB lo cual es mucho más de lo que tienen las otras.

6.3 Sensor de presencia de objetos

Para la etapa de detección de objetos, la cual permitiría saber de la presencia de una semilla en la banda transportadora, se pensaron en algunos de los sensores típicamente usados para este fin (Ver Tabla 6), los cuales se enlistan a continuación con sus características en la Tabla 7:

Tabla 6 Sensores de presencia de objetos investigados

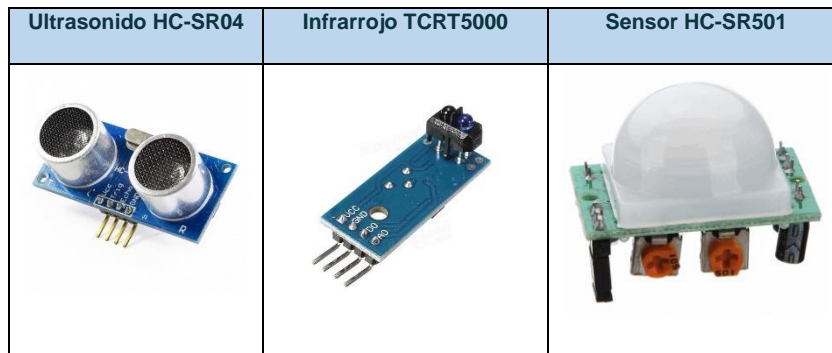


Tabla 7 Características de sensores investigados

Características	Ultrasonido HC-SR04	Modulo infrarrojo TCRT5000	Sensor HC-SR501
Voltaje de operación	5V DC	3,3 V a 5 V	5-20V
Número de conexiones requeridas	4	3	3
Rango de detección	2cm a 450cm	1 mm a 25 mm	3 m a 6 m
Dimensiones	45x20x15 mm	31 mm x 14 mm	32,2 x 24,3 x 25,4 mm
Área de la superficie de detección requerida (a partir de pruebas)	Amplia	Mediana	Amplia
Precio	\$ 8000	\$ 8000	\$ 8500

Tabla 8 Matriz QFD de los sensores investigados

Customer importance rating	Función	Ultrasonido HC-SR04	Modulo infrarrojo TCRT5000	Sensor HC-SR501
5	Voltaje de operación	3	9	3
3	Número de conexiones requeridas	3	9	9
3	Rango de detección	9	3	3
2	Dimensiones	3	9	3
3	Área de detección requerida	3	9	3
2	Costo	9	9	9
	Technical importance score	48	108	66

A partir de los resultados obtenidos de la matriz QFD (Tabla 8) y también teniendo en cuenta la complejidad que podría presentar la implementación, teniendo en cuenta sus voltajes de alimentación y la complejidad de las conexiones requeridas de cada uno de estos sensores, se determinó que el sensor infrarrojo TCRT5000 era el más adecuado.

Es importante resaltar que respecto a la alimentación se tuvo mucho en cuenta que fuesen de 3.3V ya que usualmente ese sería el voltaje de la salida digital, la cual es importante a tener en cuenta al conectarlas a la RaspberryPi, porque si este voltaje es mayor al mencionado se podría dañar el dispositivo y esto representaría un gran inconveniente para el proyecto.

6.4 Sistema de expulsión de semillas

De los sistemas de expulsión más implementados está el método de expulsión mediante chorro de aire, el cual desde un principio llamo mucho la atención el querer implementarlo por lo cual se consiguieron dos válvulas neumáticas (ilustración 20) de dimensiones reducidas y que permitiesen una fácil implementación:

Ilustración 20 Válvula neumática 2V025-08-12V



Fuente: Autor del proyecto

Algunas especificaciones técnicas de estas válvulas:

- Voltaje de funcionamiento: 12V
- Presión de trabajo: 0-10kgf/cm²
- Máxima presión soportada: 1.5Mpa
- Potencia de consumo: 3W DC
- Tipo: 2 vías, normalmente cerrada
- Apertura de la tasa de flujo: 2.5mm
- Tamaño del puerto: 1/4"
- Fluidos compatibles: aire, gas
- Protección: IP65
- Fabricada en aleación de aluminio y plástico

6.5 Sistema de iluminación

Este sistema de iluminación se basa en el método de iluminación cuadrada o directa. Para ello fue requerido diseñar un tipo de recámara la cual contuviese los focos de luz incidentes en cada una de las paredes laterales, para ello se utilizaron trozos de cintas led de color blanco y para la recámara utilizaron láminas de acrílico:

Ilustración 21 Láminas de la recámara de iluminación



Como se observa en la ilustración 21 y 22 estas piezas están compuestas por 4 tapas laterales en las cuales cada una tendrá fijada un trozo de cinta led (ilustración 23) y además también se tiene una tapa superior en la que se colocaría la cámara web.

Ilustración 22 Recámara de iluminación ensamblada (sin tapa superior)

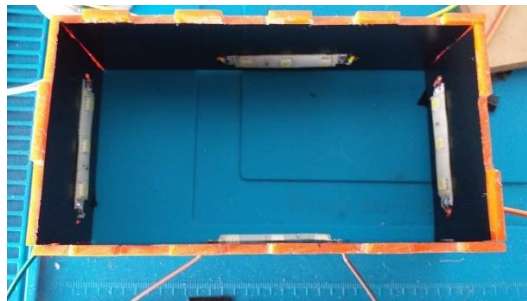


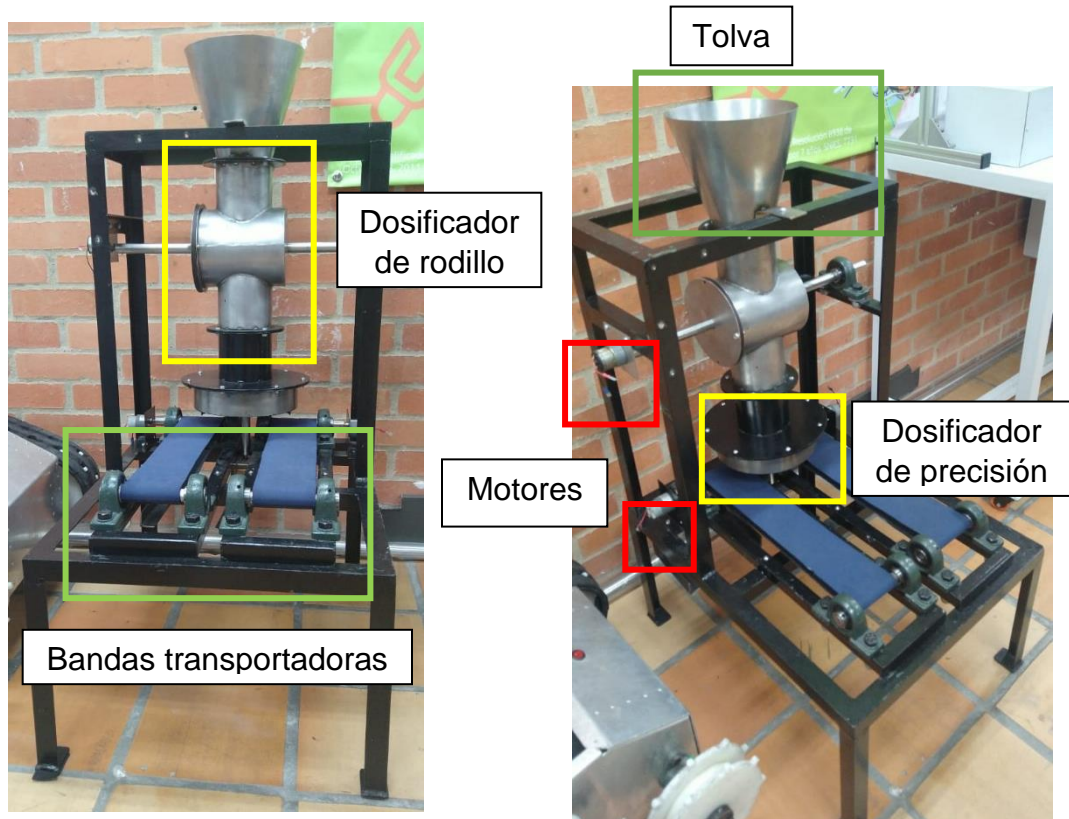
Ilustración 23 Prueba de funcionamiento de la cinta led en tapa lateral de la recámara



7 MODIFICACIONES DEL PROTOTIPO

El prototipo de maquina clasificadora resultante del proyecto “Construcción de un Prototipo de una Máquina Selectora de Semillas de Sacha Inchi” (Soto Archila, 2020) que me fue entregado en un inicio se encontraba en el siguiente estado:

Ilustración 24 Prototipo de maquina clasificadora sin modificaciones



Fuente: Autor del proyecto

Como se puede observar en la ilustración 24 el prototipo de máquina clasificadora no tenía ningún actuador, sensor o cualquier otro elemento adicional a los motores que mueven los dosificadores y las bandas transportadoras.

Las capacidades para las cuales fue diseñado el prototipo de maquina clasificadora son las siguientes:

- Capacidad máxima de semillas en la tolva de 200 gr
- Capacidad de procesamiento de un aproximado de 3 semillas por segundo
-

En dicho proyecto también se elaboraron pruebas tensión y corriente de los actuadores, lo cual permitió definir la potencia y el correcto funcionamiento de los actuadores

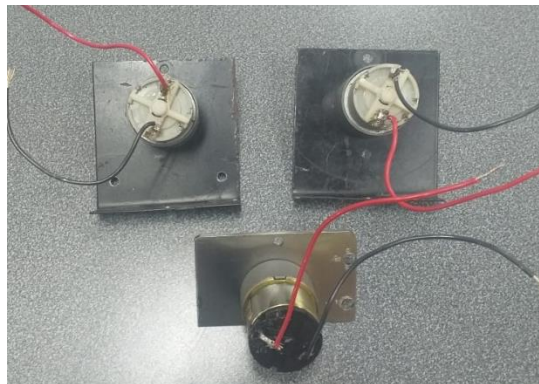
Tabla 9 Mediciones de tensión y corriente

Actuador	Corriente [A]	Tensión[V]	Potencia[W]
Motor dosificador de rodillo	0.275	11.95	3.2862
Motor dosificador de precisión	0.295	11.87	3.5017
Motor banda 1	0.335	11.75	3.9362
Motor banda 2	0.349	11.67	4.102

Fuente: autor del proyecto (Soto Archila, 2020)

Para el presente proyecto, principalmente se verifico el funcionamiento de los motores que son 4, los cuales 2 de ellos serán para las bandas transportadores y los otros dos para el funcionamiento de los dosificadores estos presentan las siguientes especificaciones:

Ilustración 25 Motores DC



- Voltaje: 12 V
- Reducción 131.25:1
- Velocidad sin carga 12V: 78 RPM
- Corriente Stall 12V: 5.5A
- Corriente free-run: 300 mA
- Velocidad sin carga 6V: 38 RPM
- Stall torque 12V: 28 kg·cm

7.1 Instalación de la recámara de iluminación

Principalmente se hizo el diseño y construcción de la recámara de iluminación teniendo en cuenta las dimensiones de la cámara web que se iba a utilizar y el espacio donde se fijaría en las bandas transportadoras, como resultado se obtuvo dos recámaras de iluminación adecuadas para soportar las cámaras y

permitiesen mantener una iluminación constante que no se viera afectada por luces ambientales externas.

Ilustración 26 Recámaras de iluminación instaladas



Como se puede observar en la ilustración 26, adicional a las recámaras también fue necesario crear unos soportes para poder mantener dichas recámaras a una altura adecuada de las cintas de las bandas transportadoras, estos soportes se crearon a partir de perfiles de aluminio, los cuales fueron colocados a ambos lados de la estructura como se observa en la ilustración 27 que soporta las chumaceras, intentando conservar la simetría de las posiciones. Fueron fijadas mediante el uso de L's y remaches los cuales permiten una correcta fijación y un acabado estético mejor que el que resultaría de usar otros métodos como atornillar o soldar, además que soldar aluminio requiere de herramientas las cuales no son fáciles de conseguir y es costoso.

Ilustración 27 Soportes de aluminio para las recamaras de iluminación



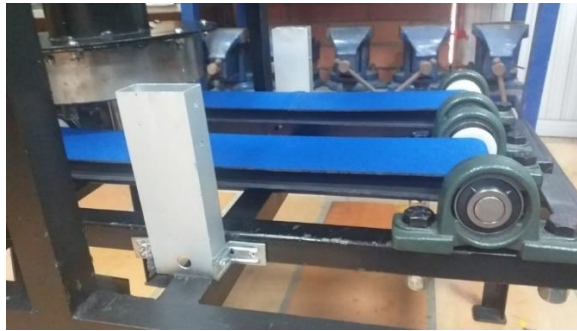
7.2 Reemplazo de cintas de la banda transportadora

Debido al mal estado en que se encontraban las cintas de las bandas transportadoras (ilustración 28), se procedió a reemplazarlas por unas nuevas, y de mejor calidad como se evidencia en la ilustración 29.

Ilustración 28 Cintas de las bandas transportadoras en mal estado



Ilustración 29 Nuevas cintas de las bandas transportadoras



7.3 Fijación de sensores y actuadores

Ya teniendo seleccionados los sensores y actuadores que se iban a implementar, era requerido fijarlos en puntos específicos para poder hacer uso adecuado de estos, con lo cual se diseñaron y fabricaron mediante impresión 3D, soportes o elementos que permitiesen fijar los sensores infrarrojos y las válvulas neumáticas, teniendo en cuenta sus dimensiones y funcionalidad. A continuación, en la ilustración 30 se puede observar el modelo 3D de las piezas y en la ilustración 31 la pieza resultante ya impresa y acopladas a las válvulas.

Ilustración 30 Diseño 3D de los soportes de las válvulas neumáticas

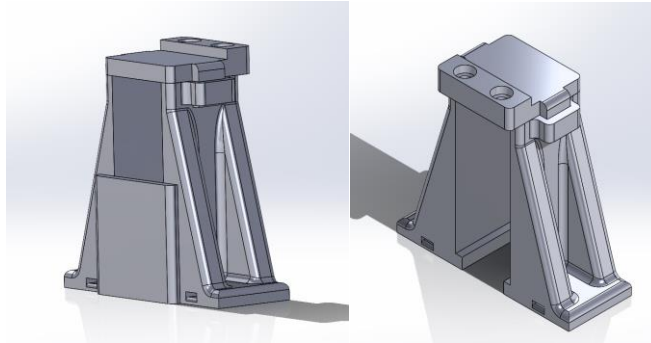
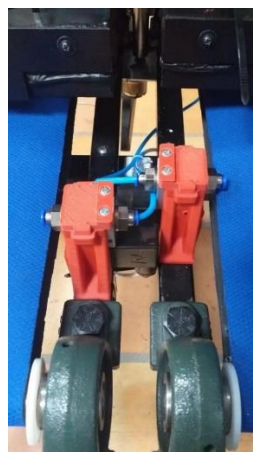


Ilustración 31 Soportes de las válvulas neumáticas



Después de comprobar que las válvulas neumáticas entraban adecuadamente en los soportes fabricados, se procedió a fijarlos en sus respectivas posiciones y realizar la conexión de las mangueras que suministrarían el flujo de aire como se observa en la ilustración 32.

Ilustración 32 Válvulas neumáticas acopladas a los soportes y fijadas



Adicional a estos soportes también como parte del acondicionamiento de las válvulas neumáticas, se instaló una unidad de mantenimiento de aire (ilustración 33), la cual se fijó en la parte trasera del prototipo de la máquina y tiene como principales funciones:

- Filtrar el aire para mantenerlo libre de impurezas, atrapando contaminantes como agua, polvo y aceite.
- Regular constantemente la presión del aire para asegurar que los equipos reciban el suministro necesario.
- Lubricar el aire comprimido para reducir la fricción entre las partes móviles de la maquinaria.

Ilustración 33 Unidad de mantenimiento fijada



Respecto a las conexiones neumáticas, las mangueras que suministran el aire son de 4mm, también fue necesario utilizar un conector neumático en T similar al de la ilustración 34 para poder dividir en dos el flujo de aire suministrado del compresor de aire que se utilizará.

Ilustración 34 Conector neumático en T



En el caso del sensor infrarrojo también se diseñó y fabricó mediante impresión 3D su respectivo soporte. A continuación, en la ilustración 35 se puede observar el modelo 3D de las piezas y en la ilustración 36 las piezas resultante ya impresas y fijadas a la estructura.

Ilustración 35 Diseño 3D del soporte para el sensor infrarrojo

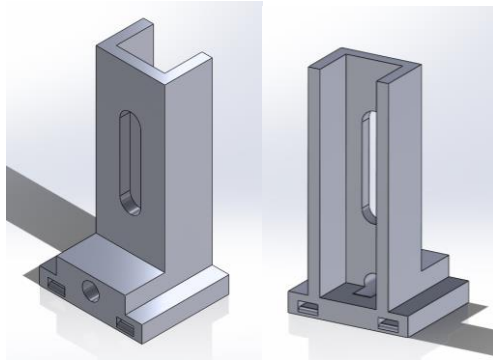
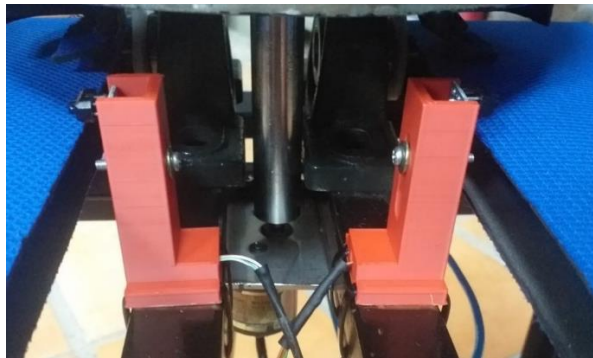


Ilustración 36 Sensores infrarrojo acoplados a los soportes y fijados



El diseño es bastante sencillo, pero cumple con su propósito de manera adecuada.

7.4 Caja o panel de control

Siendo evidente la necesidad de realizar conexiones eléctricas para permitir el control de los distintos actuadores y sensores que se están implementando, fue necesario crear un entorno de protección y seguridad para dichas conexiones, por ello se utilizó una caja de control metálica, además también de funcionar como panel de control como se mencionó esta tiene como fin generar conexiones seguras y aisladas a las diferentes señales para controlar y monitorear el sistema, también se le adicióno una pantalla led en la cual el operario puede visualizar la cantidad de semillas que va seleccionando, consta de un botón de inicio y una parada de emergencia como se observa en la ilustración 37.

Ilustración 37 Panel de control

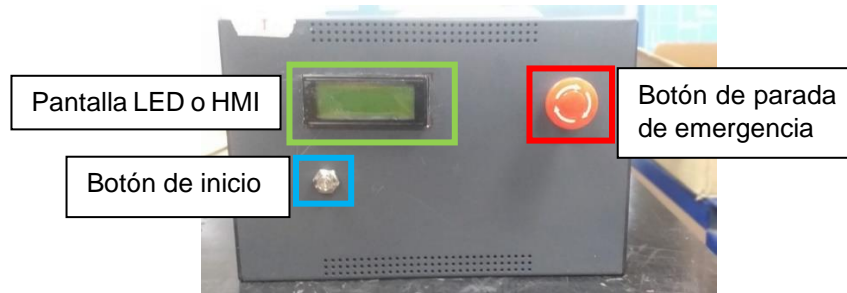
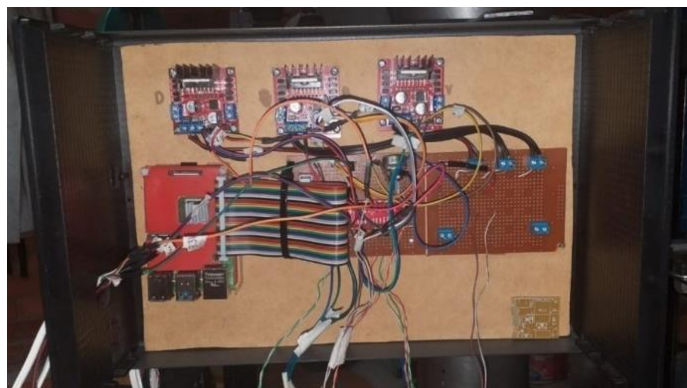


Ilustración 38 Panel de control fijado al prototipo de maquina clasificadora



Ilustración 39 Conexiones eléctricas internas de la caja de control



7.5 Estado final de la máquina con las modificaciones

Después de instalar los distintos elementos de la máquina mencionados anteriormente, se puede observar en la ilustración 40 el resultado obtenido de las modificaciones realizadas:

Ilustración 40 Prototipo de máquina clasificadora con las modificaciones



8 DISEÑO ELECTRONICO

El diseño electrónico consta de varias partes, como lo son las conexiones de los motores, de la Raspberrypi, de las cámaras, de los sensores infrarrojos, las válvulas neumáticas, de los módulos o drivers usados y la fuente de energía. Por lo que a continuación se explica cada una de las conexiones.

8.1 Fuente de energía

Como fuente de energía para los motores, válvulas neumáticas y de los sistemas de iluminación como las cintas leds y del botón de inicio, se utilizó una fuente conmutada de 12V a 10A similar al de la ilustración 41, ya que estos son los elementos que trabajan a dicho voltaje.

Ilustración 41 Fuente conmutada 12V 10A

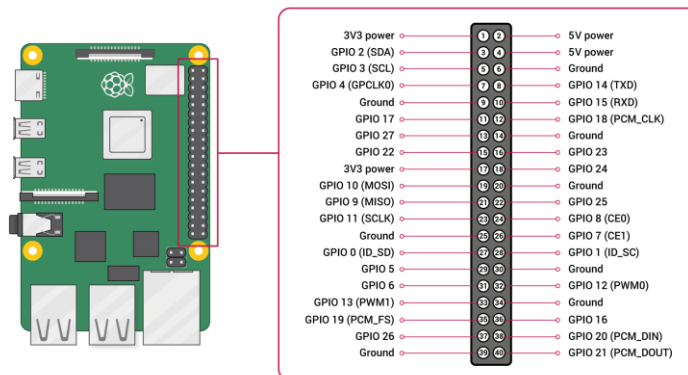


Fuente: autor

8.2 Controlador

Para el control de diversos elementos como lo son los motores DC, los sensores infrarrojos y las cámaras web, se emplea la Raspberrypi pi 4, a la cual se conectan los dispositivos, módulos y drivers necesarios para el correcto funcionamiento de la máquina. En la ilustración 42 se puede observar la Raspberrypi pi 4 con sus respectivos pines (Número y función).

Ilustración 42 Pines de conexión de la Raspberrypi 4



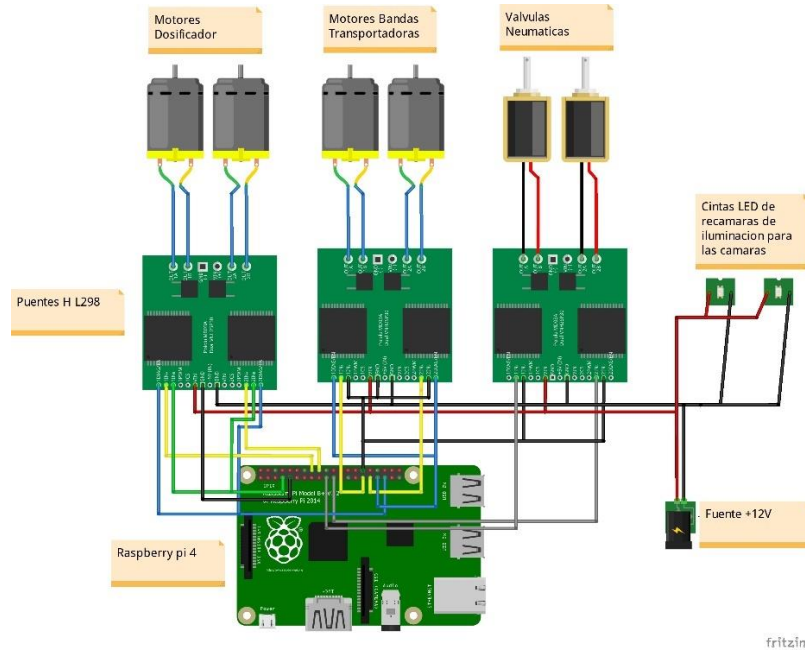
Fuente:
1200x689.png.webp

<https://www.hwlibre.com/wp-content/uploads/2020/03/gpio-raspberry-pi-4-3->

8.3 Conexiones de los motores, válvulas y cintas led

Para el control de los motores DC es necesario usar un driver, para nuestro caso es el módulo puente H L298N. Este driver se alimenta a 12V y nos permite controlar dos motores cada uno, además se pueden utilizar para controlar el accionamiento de las válvulas neumáticas. Estas conexiones van a los pines de la Raspberrypi como se muestra en la ilustración 43.

Ilustración 43 Conexiones de las cámaras, válvulas neumáticas y cintas led

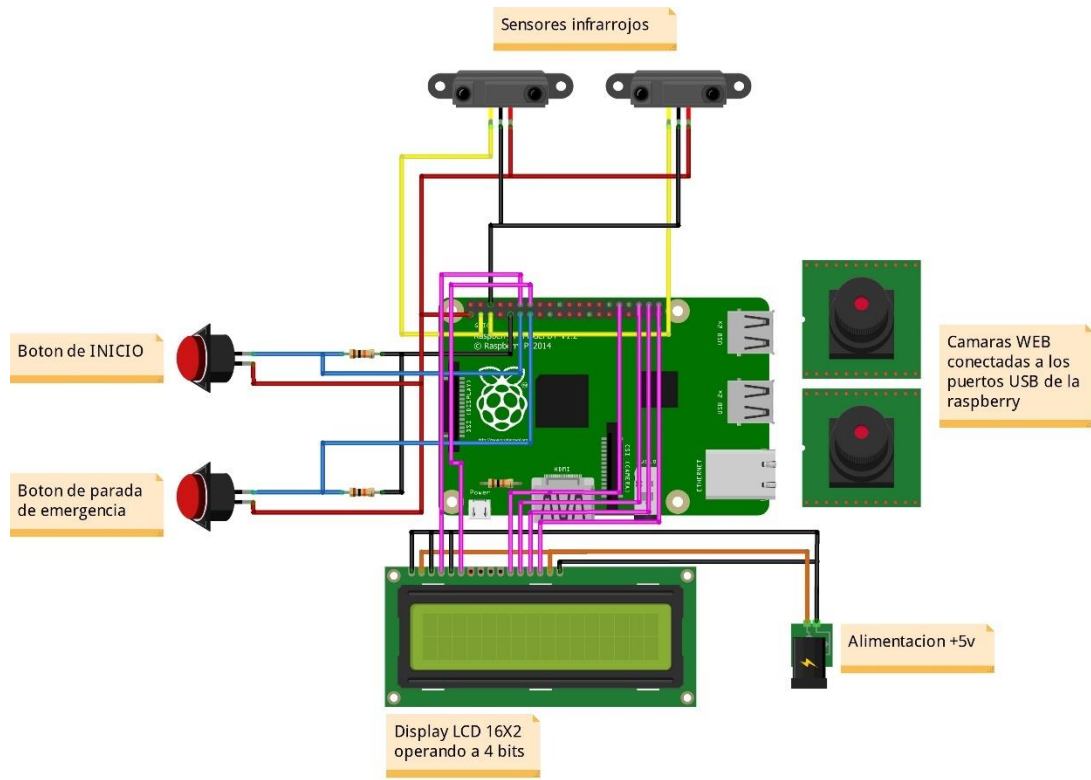


Por otro lado, las cintas leds tienen una conexión simple directa a la fuente de alimentación de +12V.

8.4 Conexión de las cámaras web, sensores, pulsadores y pantalla LED

En el caso de los sensores infrarrojos, estos funcionan a 3.3V que son suministrados por la misma Raspberrypi, además es el mismo voltaje con el cual funcionan las señales de los pulsadores. La pantalla LED funciona a 5V que podrían ser suministrados por la Raspberrypi, pero por no sobrecargarla y evitar problemas de corriente, se procedió a obtener dicha alimentación de los puentes H, los cuales tienen la opción o capacidad de suministrar 5V por uno de sus puertos. Por ultimo las cámaras web se a cualquiera de los 4 puertos USB de la Raspberrypi. Estas conexiones van a los pines de la Raspberrypi como se muestra en la ilustración 44.

Ilustración 44 Conexiones de los sensores, pulsadores, pantalla led y cámaras web



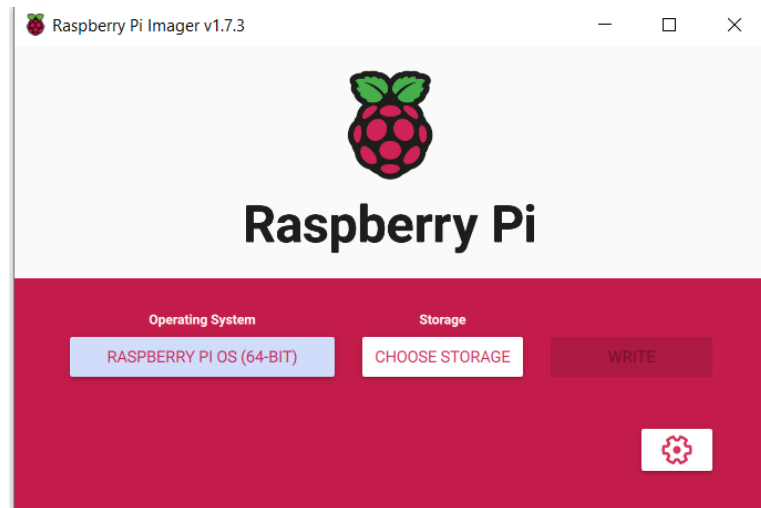
fritzing

9 PROGRAMACIÓN

9.1 Configuración de la Raspberrypi pi4

Para poder utilizar las Raspberry Pi4 o cualquier otro modelo de esta serie de dispositivos es necesario instalarle el sistema operativo, Raspberry Pi OS, para ello primero debemos tener una tarjeta microSD en la cual mediante el uso de Raspberry Pi Imager (ilustración 45), el cual es la forma rápida y fácil de instalar Raspberry Pi OS y otros sistemas operativos en una tarjeta microSD.

Ilustración 45 .Interfaz de Raspberry Pi Imager



En nuestro caso se instaló la versión de 64-bits de Raspberry PI OS, la cual es compatible con las librerías utilizadas

9.1.1 Librerías principales

Tensor Flow: es un sistema de Deep Learning que funciona a gran escala y en entornos heterogéneos. Con esta librería se construyen y entrenan redes neuronales para detectar correlaciones y descifrar patrones análogos al aprendizaje y razonamiento usados por los humanos.

Keras: es una API de alto nivel escrita en Python que corre sobre diferentes motores de Deep Learning como TensorFlow, CNTK o Theano y corre tanto en CPUs como en GPUs. Esta API ofrece modelos en los que las capas que se quieran agregar, puedan combinarse según los requisitos que se tengan. los modelos de Keras son totalmente compatibles con los tensores de TensorFlow, y como resultado, Keras hace un gran complemento de definición de modelo para TensorFlow.

OpenCV: Es una librería de código abierto que contiene implementaciones que abarcan más de 2500 algoritmos. Además, está especializada en el

sistema de visión artificial y machine Learning. Su utilización principal abarca la detección de objetos y rostros, especialmente en aspectos como la seguridad, el marketing o incluso la fotografía.

Es importante al momento de instalar todas las librerías, verificar que las versiones de estas sean compatibles.

Otros programas necesarios para continuar con la configuración de la Raspberry Pi4 son el VNC server. Mediante el VNC viewer accederemos a el dispositivo desde un computador para poder visualizar la interfaz gráfica o el escritorio de la Raspberry sin necesidad de conectarla a una pantalla, con mouse y teclado. Así podremos realizar todas las configuraciones posteriores que se requieran.

9.2 Código en Python

9.2.1 Entrenamiento del código

Para la visión artificial se empleó el tipo de red neuronal convolucional o CNN. La cual permite distintas clases de clasificación, en nuestro caso las entradas son dos clases (Oscuras y Claras). Para el entrenamiento de la red, cada clase se debe entrenar con dos tipos de imágenes para su creación, la primera es de tipo Entrenamiento, en nuestro caso se usaron cuatrocientas cincuenta (450) imágenes para el entrenamiento de cada uno. El otro tipo es de Validación, la cual debe ser mínimo tres cuartos (3/4) de la cantidad de imágenes con la que se entrenaron, por lo que en nuestro caso se emplearon ciento cincuenta (150), como se mencionó anteriormente, esta cantidad es para cada clase. En las siguientes ilustraciones 46, 47, 48 y 49 se observan las carpetas con las respectivas cantidades de imágenes de semillas mencionadas anteriormente:

Ilustración 46 Propiedades de la carpeta de imágenes para entrenamiento

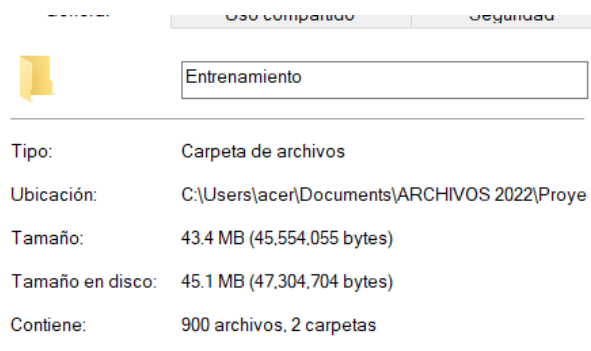


Ilustración 47 Propiedades de las carpetas de semillas claras y oscuras de la carpeta de entrenamiento

Oscuras		Claras	
Tipo:	Carpeta de archivos	Tipo:	Carpeta de archivos
Ubicación:	C:\Users\acer\Documents\ARCHIVOS 2022\Proye	Ubicación:	C:\Users\acer\Documents\ARCHIVOS 2022\Proye
Tamaño:	21.0 MB (22,076,791 bytes)	Tamaño:	22.3 MB (23,477,264 bytes)
Tamaño en disco:	21.8 MB (22,900,736 bytes)	Tamaño en disco:	23.2 MB (24,403,968 bytes)
Contiene:	450 archivos, 0 carpetas	Contiene:	450 archivos, 0 carpetas

Ilustración 48 Propiedades de la carpeta de imágenes para validación

Validacion	
Tipo:	Carpeta de archivos
Ubicación:	C:\Users\acer\Documents\ARCHIVOS 2022\Proye
Tamaño:	14.1 MB (14,797,911 bytes)
Tamaño en disco:	14.6 MB (15,396,864 bytes)
Contiene:	300 archivos, 2 carpetas

Ilustración 49 Propiedades de las carpetas de semillas claras y oscuras de la carpeta de validación

Claras		Oscuras	
Tipo:	Carpeta de archivos	Tipo:	Carpeta de archivos
Ubicación:	C:\Users\acer\Documents\ARCHIVOS 2022\Proye	Ubicación:	C:\Users\acer\Documents\ARCHIVOS 2022\Proye
Tamaño:	7.24 MB (7,601,357 bytes)	Tamaño:	6.86 MB (7,196,554 bytes)
Tamaño en disco:	7.53 MB (7,901,184 bytes)	Tamaño en disco:	7.14 MB (7,495,680 bytes)
Contiene:	150 archivos, 0 carpetas	Contiene:	150 archivos, 0 carpetas

Ilustración 50 Muestra de imagen de semilla "Clara"



Ilustración 51 Muestra de imagen de semilla "Oscura"



Luego de tener las imágenes como las muestras de la ilustración 50 y 51 para la creación de la red neuronal se procede a declarar los parámetros de entrenamiento. Lo primero es estipular la cantidad de épocas de entrenamiento, luego estipular el tamaño de las imágenes, el tamaño del lote o batch size, la cantidad de pasos para la parte de entrenamiento y para la validación, los filtros para las convoluciones, el tamaño de los filtros y la ratio de aprendizaje como se observa en la ilustración 52.

Ilustración 52 Parámetros requeridos para el entrenamiento

```
"""
Parametros
"""
epocas=20
longitud, altura = 150, 150
batch_size = 15
pasos = 12
validation_steps = 12
clases = 2
lr = 0.0004

filtrosConv1 = 32
filtrosConv2 = 64
tamano_filtro1 = (3, 3)
tamano_filtro2 = (2, 2)
tamano_pool = (2, 2)
```

Los anteriores parámetros fueron definidos y ajustados de acuerdo a la capacidad de cómputo del equipo que se utilizó para realizar el entrenamiento, las épocas son el número de veces que se iba a iterar sobre todo los datos de ingreso, que serían las imágenes de las semillas, en cada una de estas épocas se procesan imágenes un número de veces definidos por el número de pasos, la cantidad de imágenes procesadas en estos grupos depende del "batch size". Se aplicaron 2 tipos de filtros de convolución con distintas profundidades el primero de 32 para detectar detalles más evidentes de la imagen como colores y formas; y el otro de 64 para poder lograr detectar detalles más precisos como sombras, contrastes y bordes presentes en la imagen. Además de un filtro Max pooling realizar la reducción de los datos provenientes de cada uno los filtros de convolución, y por último el Learning rate (lr) siempre se busca utilizar un valor muy bajo valor que permita reduciendo el error conforme van pasando

las épocas hasta llegar a nuestro mínimo buscado este valor siempre usualmente suelen ser de la escala 1×10^{-1} , 1×10^{-2} , 1×10^{-3} o menores, entre más bajo sea el valor de este parámetro, menor será la cantidad de pasos que necesitaremos utilizar.

Cuando se comienza el entrenamiento se puede observar la época en la que va, el tiempo que falta para terminarse, el error que se genera en el aprendizaje (lo ideal es que este valor sea lo más pequeño posible, para garantizar que las predicciones van a ser correctas) y además, se tiene el porcentaje de aprendizaje (ilustración 53). Este valor también es muy importante, indica el porcentaje de aprendizaje que tiene nuestra red, por lo que lo ideal es que sea un valor muy cercano a uno (1).

Ilustración 53 Proceso de entrenamiento visualización

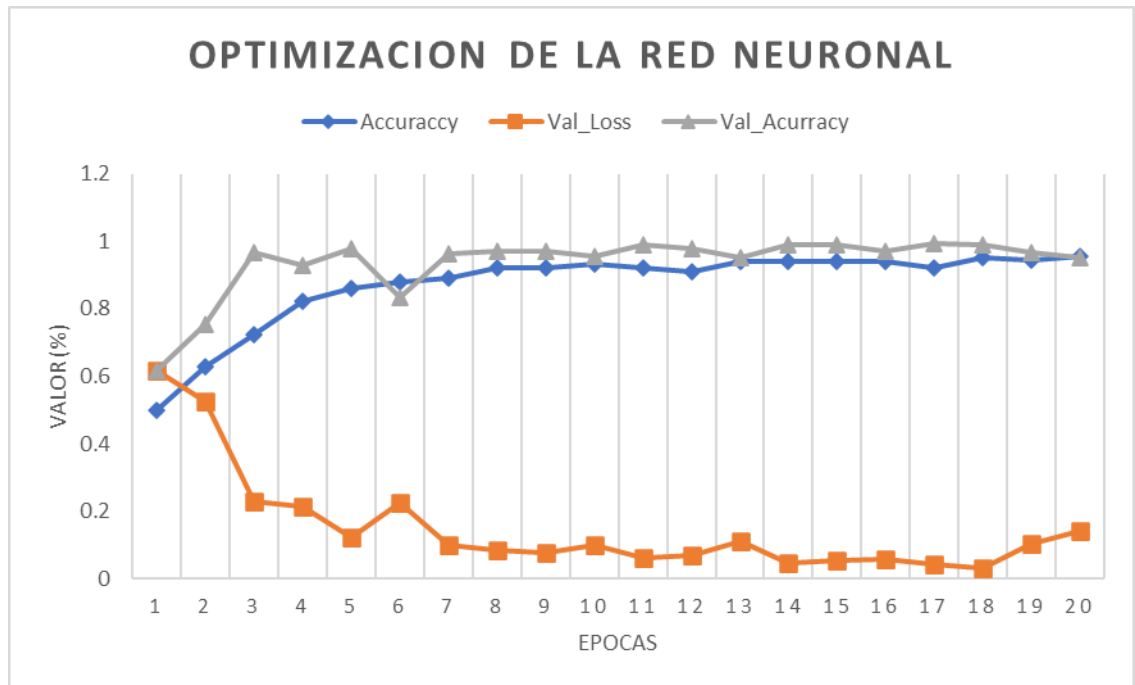
```
Epoch 1/20
12/12 [=====] - 2s 126ms/step - loss: 2.2537 - accuracy: 0.5056 - val_loss: 1.2554 - val_accuracy: 0.4667
Epoch 2/20
12/12 [=====] - 1s 115ms/step - loss: 1.0624 - accuracy: 0.5500 - val_loss: 0.6536 - val_accuracy: 0.5667
Epoch 3/20
12/12 [=====] - 2s 127ms/step - loss: 0.7380 - accuracy: 0.6278 - val_loss: 0.5275 - val_accuracy: 0.7611
Epoch 4/20
4/12 [=====>.....] - ETA: 0s - loss: 0.6769 - accuracy: 0.6000
```

Se realizó la tabla 10 con datos de cada una de las épocas y se procedió a graficarlo (ilustración 54) para poder observar como la red neuronal se iba optimizando disminuyendo las pérdidas “val_loss” y mejorando el porcentaje de “Accuracy” el cual indica que tan bien está aprendiendo después de realizar la validación “val_accuracy” en cada una de las épocas:

Tabla 10 Datos de optimización de la red neuronal

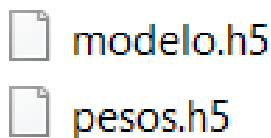
Epoca	Accuraccy	Val_Loss	Val_Accurracy
1	0.5	0.6169	0.6167
2	0.6278	0.5242	0.7556
3	0.7222	0.2298	0.9667
4	0.8222	0.213	0.9278
5	0.8611	0.1228	0.9778
6	0.8778	0.2252	0.833
7	0.8889	0.1018	0.9611
8	0.9222	0.0847	0.9722
9	0.9222	0.0788	0.9722
10	0.9333	0.1007	0.9556
11	0.9222	0.0635	0.9889
12	0.9111	0.0688	0.9778
13	0.9389	0.1131	0.95
14	0.9389	0.0489	0.9889
15	0.9389	0.0559	0.9889
16	0.9389	0.058	0.9722
17	0.9222	0.0443	0.9944
18	0.95	0.0323	0.9889
19	0.9444	0.1047	0.9667
20	0.9556	0.1405	0.95

Ilustración 54 Representación gráfica de los datos de optimización de la red neuronal



Del entrenamiento se obtienen dos archivos Modelo y Peso (ilustración 55). Estos archivos contienen toda la información del entrenamiento, por eso se guardan y son los archivos que luego son cargados cuando se quiera realizar una predicción.

Ilustración 55 Archivos de modelo y pesos generados del entrenamiento



9.2.2 Funcionamiento general

El funcionamiento general de la maquina clasificadora (ilustración 56) y el código implementado en la Raspberrypi funciona como se muestra en la ilustración 57, Comenzando por la activación del pulsador de inicio, accionando los motores de los dosificadores y la banda transportadora. Luego de esto el sensor infrarrojo de cada una de las bandas transportadoras informara si hay una semilla presente.

Cuando el sensor detecta la presencia de una semilla, se procede esperar que esta semilla se posicione debajo de la recama de iluminación donde se

permitirá la captura de la imagen correspondiente de la semilla actual, luego de esto Raspberry comenzara el procesamiento respectivo de la imagen.

Después del procesamiento y análisis de la imagen, se obtiene el resultado de comparación de la imagen actual con los datos obtenidos del entrenamiento, de la cual se tomará la decisión de expulsar la semilla si esta es definida como “Clara” o se permitirá continuar en la banda transportadora si esta es definida como “Oscura”

Ambas semillas después de ser procesadas son almacenadas en sus respectivos contenedores.

Es de aclarar que mientras no haya semillas para procesar los motores y las luces led de las recamaras de iluminación seguirán funcionando de manera indefinida hasta que el usuario deshabilite el pulsador de inicio del proceso o desconecte las fuentes de alimentación. A su vez los sensores, válvulas neumáticas y la misma Raspberypi estarán en espera.

Ilustración 56 Diagrama de funcionamiento de la maquina clasificadora

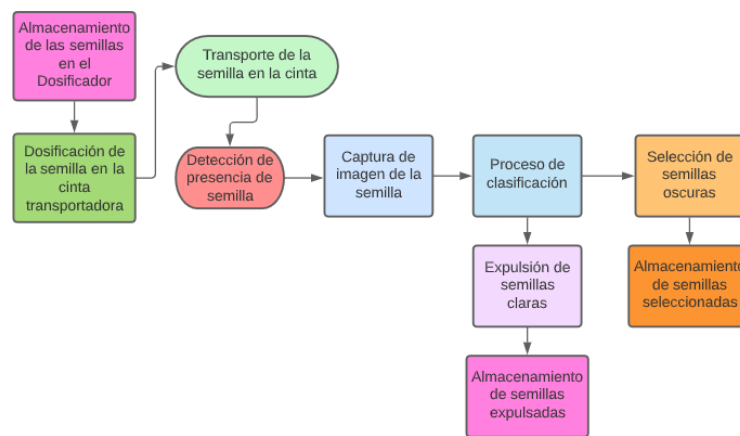
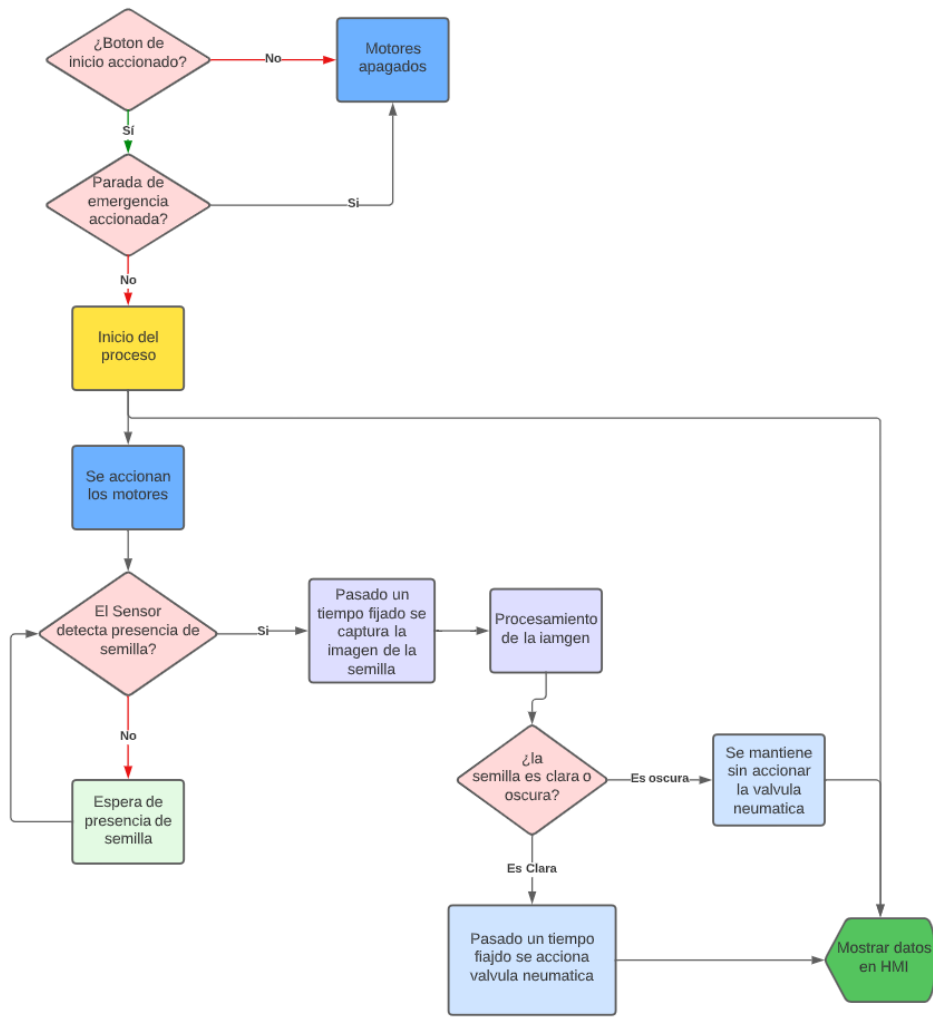


Ilustración 57 Diagrama de flujo del código de funcionamiento



Los datos a mostrar en la pantalla led, serían la cantidad de semillas detectadas total en ambas bandas (ST), la cantidad de semillas expulsadas (SE), cantidad de semillas procesadas por la banda #1 (SB1) y cantidad de semillas procesadas por la banda #2 (SB2) como se observa en la ilustración 58.

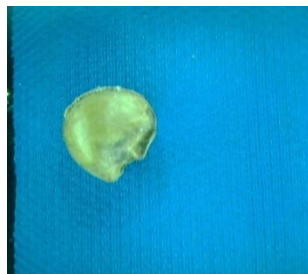
Ilustración 58 Pantalla LED con datos de ejemplo



10 VALIDACIÓN

Para la validación del funcionamiento del algoritmo de selección mediante visión artificial implementado en el prototipo de máquina clasificadora, se realizaron tres pruebas. La primera fue con muestras de imágenes capturadas de semillas “Claras” (Tabla 11), la segunda con muestras de semillas “Oscuras” (Tabla 12) y por último una mezcla de semillas “Oscuras” y “Claras” (Tabla 13) para verificar que la maquina era capaz de responder adecuadamente ante la variación del tipo de semilla procesada. Para cada prueba se utilizaron 20 muestras de semillas similares a las ilustraciones 59 y 60 . Los resultados obtenidos fueron los siguientes:

Ilustración 59 Muestra #1 de semillas Claras



```
>>> predict('./Claras Banda1/objeto_1.jpg
')
pred: Clara
```

Tabla 11 Prueba con muestras de semillas Claras

Muestra de semilla Clara	Valor real	Valor de predicción	Resultado de predicción
1	Clara	Clara	Correcta
2	Clara	Clara	Correcta
3	Clara	Clara	Correcta
4	Clara	Clara	Correcta
5	Clara	Clara	Correcta
6	Clara	Clara	Correcta
7	Clara	Clara	Correcta
8	Clara	Clara	Correcta
9	Clara	Clara	Correcta
10	Clara	Clara	Correcta
11	Clara	Clara	Correcta
12	Clara	Clara	Correcta
13	Clara	Clara	Correcta
14	Clara	Clara	Correcta
15	Clara	Clara	Correcta
16	Clara	Clara	Correcta
17	Clara	Clara	Correcta
18	Clara	Clara	Correcta

19	Clara	Clara	Correcta
20	Clara	Clara	Correcta
		% de acierto	100%

Ilustración 60 Muestra #1 de semilla oscura



```
...
>>> predict('./Oscuras Banda2/objeto_1.jpg')
pred: Oscura
```

Tabla 12 Pruebas con muestras de semillas Oscuras

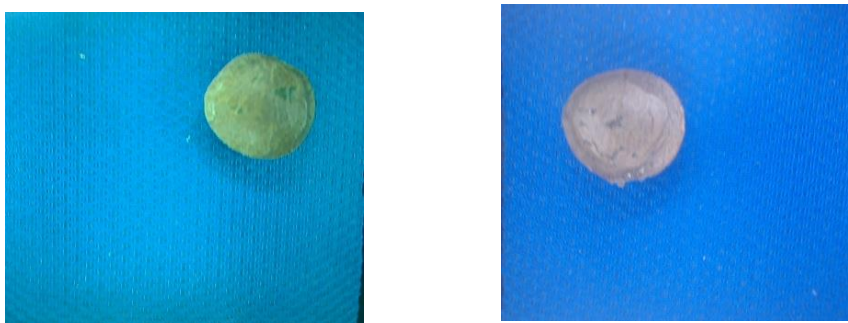
Muestra de semilla oscura	Valor real	Valor de predicción	Resultado de predicción
1	Oscura	Oscura	Correcta
2	Oscura	Oscura	Correcta
3	Oscura	Oscura	Correcta
4	Oscura	Oscura	Correcta
5	Oscura	Oscura	Correcta
6	Oscura	Oscura	Correcta
7	Oscura	Oscura	Correcta
8	Oscura	Oscura	Correcta
9	Oscura	Clara	Incorrecta
10	Oscura	Oscura	Correcta
11	Oscura	Oscura	Correcta
12	Oscura	Oscura	Correcta
13	Oscura	Oscura	Correcta
14	Oscura	Oscura	Correcta
15	Oscura	Oscura	Correcta
16	Oscura	Clara	Incorrecta
17	Oscura	Oscura	Correcta
18	Oscura	Oscura	Correcta
19	Oscura	Oscura	Correcta
20	Oscura	Oscura	Correcta
		% de acierto	90%

Tabla 13 Pruebas con muestras de semillas mixtas

Muestra de semilla Mixtas	Valor real	Valor de predicción	Resultado de predicción
1	Clara	Clara	Correcta
2	Clara	Clara	Correcta
3	Clara	Clara	Correcta
4	Clara	Clara	Correcta
5	Clara	Clara	Correcta
6	Oscura	Oscura	Correcta
7	Oscura	Oscura	Correcta
8	Oscura	Oscura	Correcta
9	Oscura	Oscura	Correcta
10	Oscura	Oscura	Correcta
11	Clara	Clara	Correcta
12	Clara	Clara	Correcta
13	Clara	Clara	Correcta
14	Clara	Clara	Correcta
15	Clara	Clara	Correcta
16	Oscura	Oscura	Correcta
17	Oscura	Oscura	Correcta
18	Oscura	Oscura	Correcta
19	Oscura	Oscura	Correcta
20	Oscura	Oscura	Correcta
		% de acierto	100%

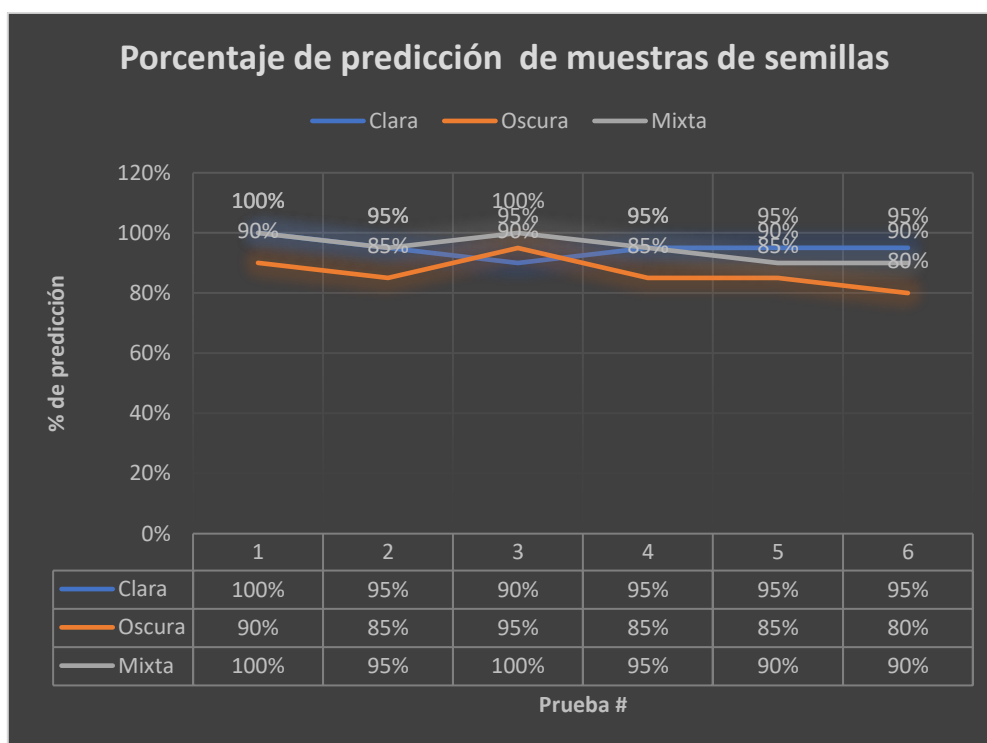
Como puede observarse para el caso de las pruebas realizadas con semillas Oscuras, se obtuvieron dos respuestas de “Clara” lo cual se considera un error en el porcentaje de predicción, las muestras en cuestión que fueron analizadas para esos dos casos presentaron un color más claro de lo usual, un tamaño mucho mayor o forma distinta a comparación de otras como se observa en la ilustración 61.

Ilustración 61 Muestras #9 y #16 de semilla Oscura con respuesta errónea



Para corroborar la información obtenida de la anterior prueba, se realizaron adicionalmente 6 (Ver anexo 14 y 15) pruebas con la misma configuración, a partir de estas se obtuvo un análisis estadístico el cual se puede observar en la gráfica de la ilustración 62.

Ilustración 62 Grafica de análisis estadístico de las pruebas de validación



Se pudo observar una tendencia de haber mayor porcentaje de respuestas incorrectas en las semillas “Oscuras” y mayor porcentaje de acierto o respuestas correctas en las semillas “Claras”.

11 CONCLUSIONES Y RECOMENDACIONES

- Es importante lograr que el dosificador de precisión sea realmente capaz de entregar una semilla a intervalos de tiempo casi constantes, esto aseguraría que la cantidad de semillas en las imágenes capturadas por las cámaras siempre sean una y no un grupo de semillas, lo cual es una situación que en ocasiones se puede llegar a presentar debido al mecanismo de dosificación.
- Debido a la similitud de las semillas en algunos casos, como se pudo observar en las pruebas de validación, el porcentaje de predicción de las semillas no logró ser del 100% siempre, pero se consiguió un buen porcentaje de acierto.
- Si se desea mejorar la velocidad de procesamiento se requiere incrementar la memoria RAM del controlador, por lo que se recomendaría una memoria RAM de 8 GB, debido a que las redes neuronales para la clasificación consumen bastante memoria de cómputo.
- Se debe tener en cuenta el voltaje con el que funcionan los GPIO que es de 3.3V de la Raspberry para evitar daños en el procesador, cuando se realiza trabajos con otras placas de desarrollo como Arduino que trabajan con 5V.
- Debido a problemas en la adquisición de dos cámaras del mismo modelo, se tuvo que utilizar una con características parecidas a la seleccionada inicialmente, lo que puede influir también en los resultados de predicción, por ello sería recomendable utilizar el mismo modelo de cámara para evitar cualquier pequeña variación en la calidad de imágenes capturadas.
- Es muy importante el tipo de iluminación que se utiliza para tener una correcta clasificación, así se pueden evitar sombras que afecten el procesamiento y resultado de predicción final de las imágenes capturadas.
- Se podría utilizar un mejor método de detección de presencia de las semillas, el cual permita detectar objetos más pequeños, ya que en ocasiones estas semillas pueden tener un área bastante reducida lo cual dependiendo el sensor este puede no tener la capacidad de detectar dicha semilla.

12 REFERENCIAS

- Karisma . (2015). *Una propuesta para cultivar y procesar Sacha Inchi en la Zona*. Obtenido de <https://karisma.org.co>
- Agronet. (10 de 3 de 2015). *Análisis - Estadísticas*. Obtenido de <http://www.agronet.gov.co>
- Ahila Priyadharshini, R., Arivazhagan, S., Arun, M., & Mirnalini, A. (2019). Maize leaf disease classification using deep convolutional neural networks. *Neural Comput & Applic*.
- Alimboyong, C., Hernandez, A. A., & Medina, R. P. (2018). Classification of Plant Seedling Images Using Deep Learning. Region 10 conference TENCON 2018.
- Alvarez, G. F., & Rios, T. R. (2007). Estudio de viabilidad económica del cultivo de Plukenetia volubilis Linneo "sacha inchi" del departamento de San martin. Iquitos, Perú: Programa de ordenamiento ambiental – POA evaluación económica opciones productivas amazonia peruana.
- Bussman, R., Tellez, C., & Glenn, A. (2009). Journal Of Botanic(Diario Internacional de Botánica y Micología). 17_21.
- Cachique, D., Ruiz, H., Hidalgo, L., Garcia, M., Guerrero, J., Solis, R., & Mesen, F. (2010). Propagación vegetativa del sachá inchi (Plukenetia volubilis L.) Mediante injerto, bajo condiciones controladas en San Martín, Perú. En *Congreso Peruano de Mejoramiento Genético y Biotecnología Agrícola* (págs. 74-75). Lima, Perú: UNAL/EPG.
- Calvo, D. (13 de Julio de 2017). *Clasificación de redes neuronales artificiales*. Obtenido de <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>
- Calvo, D. (20 de Julio de 2017). *Red Neuronal Convolutacional CNN*. Obtenido de <https://www.diegocalvo.es/red-neuronal-convolutacional/>
- Castellanos Castilla, J. d., & Castañeda Dias, J. S. (2017). *Diseño e implementación de un sistema automatizado para la clasificación del limón Tahiti de acuerdo con el tamaño y el grado de madurez*. Bucaramanga: Universidad Autónoma de Bucaramanga.
- Dorado, E. (2020). Apuntes del curso Deep Learning aplicado al análisis de señales e imágenes. Valencia: CFP Universidad Politécnica de Valencia.
- Dorado, E. d. (2019). *Diseño, implementación y evaluación de una red neuronal convolutacional para la detección de puntos principales en naranjas*. Valencia : Universidad politécnica de valencia.
- Dorado, E. d. (2019-2020). DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED NEURONAL CONVOLUCIONAL PARA LA DETECCIÓN DE PUNTOS PRINCIPALES EN NARANJAS. Valencia: Universidad Politecnica de valencia.

- Gomez V, V. Z. (2011). Diseño y construcción de una máquina tipo expeller para la extracción de aceite vegetal a partir de la semilla de la *Jatropha curcas* con capacidad de 200 Kilogramos/hora para la E.S.P.E-IASA II.
- Gomez, M. (2004). *Monografía y cultivo de sacha inchi, oleaginosas promisorias para la diversificación productiva en el trópico*. . Corporación Colombiana de Investigación agropecuaria CORPOICA.
- Grandez Muñoz, E. A., & Chingo Coronado, D. (2015). *ALICIA CONCYTEC*. Obtenido de https://alicia.concytec.gob.pe/vufind/Record/UNTR_1c52168c34fa888227796f2cd6b3f53e
- Hermoza Llanos, E. A. (2018). *Diseño de una máquina seleccionadora de semillas de sacha inchi peladas, por color con capacidad de 200kg/h*. Lima, Perú: Pontificia universidad católica del Perú.
- Jimenez, R., G.M, M., & Cruz, D. (2000). El género *Plukenetia* (Euphorbiaceae) en México. *Anales del Instituto de Biología Universidad Nacional Autónoma de México, Serie Botánica*, 71(1), 11-18.
- Kriesel, D. (2007). A Brief Introduction to NeuralNetworks.
- Mendoza, R. I., & Martin, M. d. (2014). *Reconocimiento de objetivos en imágenes de video utilizando Visión artificial*. Madrid, España: Universidad Politecnica de Madrid.
- Moreno, A. A. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python*. Sevilla: Escuela Técnica Superior de Ingeniería, Universidad de Sevilla.
- O., H. (2013). Análisis composicional de la torta y aceite de semillas de Sacha Inchi (*Plukenetia volubilis*) cultivada en Colombia. Palmira, COlombia.
- Paguay Donoso, D. G., & Valarezo Rodriguez, L. X. (2018). *Diseño e implementación de un prototipo clasificador de huevo de gallina basado en las imperfecciones de la cascara aplicando vision artificial*. Riobamba, Ecuador: Escuela Politecnica de chimborazo.
- Sanchez, R., & Amiqueri, B. (2004). Manual de cultivo de sacha inchi. Agroservicios LIMAG.
- Sanchez, S. L. (2012). Caracterización y cuantificación de los ácidos grasos omega 3 y omega 6 presentes en el aceite de sacha inchi (*Plukenetia volubilis* L.). Colombia, Medellín: Universidad Nacional de Colombia, Facultad de Ciencias - Escuela de Química.
- Sevillano B., F. J. (2015). Equipo de clasificación de semillas basado en visión artificial. Implantación sobre hardware libre. Valladolid: UNIVERSIDAD DE VALLADOLID ESCUELA DE INGENIERIAS INDUSTRIALES.
- Sevillano Bustos, F. J. (2015). *Equipo de clasificación de semillas basado en visión artificial. Implantación sobre hardware libre*. Valladolid: Universidad de Valladolid.
- Silva, S., & Freire, E. (23 de Noviembre de 2019). *Introducción a las redes neuronales convolucionales*. Obtenido de

- <https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8>
- sitiobigdata. (1 de mayo de 2019). *Innovaciones arquitectónicas en redes neuronales*. Obtenido de <https://sitiobigdata.com/2019/05/01/innovaciones-arquitectonicas-redes-neuronales-clasificacion-imagenes/#>
- Soto Archila, M. F. (2020). *Construcción de un Prototipo de una Máquina Selectora de Semillas de Sacha Inchi*. Bucaramanga,: Universidad Autonoma de Bucaramanga.
- Valdes Avila, L. S., & Baquero Vanegas, J. M. (2019). *Deep Learning aplicado a imagenes satelitales como herramienta de detección de viviendas sin servicio de energia en el caserío Media Luna-Uribia-Guajira*. Bogota, Colombia: Universidad Distrital Francisco Jose de Caldas.
- Weber, C., Dai Pra, A. L., Rabal, L. I., J., H., Trivi, M., & Poggio Agurre, G. J. (2014). Determination of maize hardness by biospeckle and fuzzy granularity. *Food Science Nutrition*, 2(5), 557-564.

13 ANEXOS

Anexo 1 Código de entrenamiento de la CNN

```
import sys
import os

import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dropout, Flatten, Dense,
Activation
from tensorflow.python.keras.layers import Convolution2D, MaxPooling2D
from tensorflow.python.keras import backend as K

K.clear_session()

data_entrenamiento = './Imagenes finales/Entrenamiento'
data_validacion = './Imagenes finales/Validacion'

"""
Parametros
"""
epocas=20
longitud, altura = 150, 150
batch_size = 15
pasos = 12
validation_steps = 12
clases = 2
lr = 0.0004

filtrosConv1 = 32
filtrosConv2 = 64
tamano_filtro1 = (3, 3)
tamano_filtro2 = (2, 2)
tamano_pool = (2, 2)

"""
cantidad de imagenes por etiqueta
```

```

cremas= 115
manchadas= 66
oscuras= 117

Preparacion de las imagenes
"""
entrenamiento_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1. / 255)

entrenamiento_generador = entrenamiento_datagen.flow_from_directory(
    data_entrenamiento,
    target_size=(altura, longitud),
    batch_size=batch_size,
    class_mode='categorical')

validacion_generador = test_datagen.flow_from_directory(
    data_validacion,
    target_size=(altura, longitud),
    batch_size=batch_size,
    class_mode='categorical')

"""
creacion de red
"""
cnn = Sequential()
cnn.add(Convolution2D(filtrosConv1, tamaño_filtro1, padding ="same",
input_shape=(longitud, altura, 3), activation='relu'))
cnn.add(MaxPooling2D(pool_size=tamaño_pool))

cnn.add(Convolution2D(filtrosConv2, tamaño_filtro2, padding ="same"))
cnn.add(MaxPooling2D(pool_size=tamaño_pool))

cnn.add(Flatten())
cnn.add(Dense(256, activation='relu'))
cnn.add(Dropout(0.5))
cnn.add(Dense(clases, activation='softmax'))

cnn.compile(loss='categorical_crossentropy',

```



```

        optimizer=Adam(learning_rate=lr),
        metrics=['accuracy'])

cnn.fit(entrenamiento_generador,
        steps_per_epoch=pasos,
        epochs=epocas,
        validation_data=validacion_generador,
        validation_steps=validation_steps)

target_dir = './Modelo final/'
if not os.path.exists(target_dir):
    os.mkdir(target_dir)

cnn.save('./modelo/modelo.h5')
cnn.save_weights('./modelo/pesos.h5')

```

Anexo 2 Código de control de la máquina de clasificación

```

# Librerias vision artificial y tratamiento de imagenes
import cv2
import numpy as gfg
import imutils
import os

from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.models import load_model

from PIL import Image, ImageDraw

# Libreria configuracion de raspberry
import RPi.GPIO as GPIO
import time

#libreria LCD
sys.path.append('/home/isaakrangel/lcd')
import lcd
lcd.lcd_init()

# libreria para usar threading o hilos
# import threading

```

```

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

#cargar modelos y pesos de entrenamiento
# modelo = './modelo/modelo.h5'
# pesos_modelo = './modelo/pesos.h5'
# cnn = load_model(modelo)
# cnn.load_weights(pesos_modelo)

#crear carpeta y permitir capturar con la camara
Datos = 'sacha inchi'
if not os.path.exists(Datos):
    print('Carpeta creada: ',Datos)
    os.makedirs(Datos)

# camaras de captura
cap1 = cv2.VideoCapture(0)
cap2 = cv2.VideoCapture(2)

# coordenadas de recorte
x1, y1 = 30, 0
x2, y2 = 420, 350
longitud, altura = 150, 150

# Señales de control (boton on/off, apagado de emergencia)
on = 17
emerg = 27

GPIO.setup(on, GPIO.IN)
GPIO.setup(emerg, GPIO.IN)

# Pines y parametros infrarrojo
sen1 = 2
sen2 = 3

GPIO.setup(sen1, GPIO.IN)
GPIO.setup(sen2, GPIO.IN)

# Pines y parametros motores y valvula
# motores dosificador
en1 = 19
in11 = 23
in13 = 24

```

```

in14 = 4
    # motores banda
en2 = 13
in21 = 5
in23 = 6
    # valvulas neumaticas
in31 = 10
in33 = 9

    # define salidas motores
#pines motores dosificador
GPIO.setup(en1, GPIO.OUT) # pines enable puente H de motores de
dosificadores
GPIO.setup(in11, GPIO.OUT)
GPIO.setup(in13, GPIO.OUT)
GPIO.setup(in14, GPIO.OUT)
#pines motores banda
GPIO.setup(en2, GPIO.OUT) # pines enable puente H de motores de bandas
GPIO.setup(in21, GPIO.OUT)
GPIO.setup(in23, GPIO.OUT)
#pines valvulas
GPIO.setup(in31, GPIO.OUT)
GPIO.setup(in33, GPIO.OUT)

#Define las salidas PWM de motores dosificador, banda y valvulas
pwm_1 = GPIO.PWM(en1,255)
pwm_2 = GPIO.PWM(en2,255)

# inicializa los pmw
pwm_1.start(0)
pwm_2.start(0)

# Funciones de encendido y apagado puentes H
def Encender_Motores_DosificadorGD():
    GPIO.output(in11,True)
    GPIO.output(in13,True)
    GPIO.output(in14,False)

def Encender_Motores_DosificadorGL():
    GPIO.output(in11,True)
    GPIO.output(in13,False)

```

```

        GPIO.output(in14,True)

def Encender_Motores_Banda():
    GPIO.output(in21,True)
    GPIO.output(in23,True)

def Apagar_Motores_Dosificador():
    GPIO.output(in11,False)
    GPIO.output(in13,False)
    GPIO.output(in14,False)

def Apagar_Motores_Banda():
    GPIO.output(in21,False)
    GPIO.output(in23,False)

def Encender_Valvulas(hilo):

    if hilo==1:
        GPIO.output(in31,True)
    if hilo==2:
        GPIO.output(in33,True)

def Apagar_Valvulas(hilo):

    if hilo==1:
        GPIO.output(in31,False)
    if hilo==2:
        GPIO.output(in33,False)
    else:
        GPIO.output(in33,False)
        GPIO.output(in31,False)

# Funcion de prediccion dependiendo el hilo

def predict(file,hilo):
    x = load_img(file, target_size=(longitud, altura))
    x = img_to_array(x)
    x = gfg.expand_dims(x, axis=0)
    array = cnn.predict(x)
    result = array[0]
    answer = gfg.argmax(result)

    if answer == 0:
        print("pred: Oscura")
        Seleccionadas = Seleccionadas + 1

```

```

elif answer == 1:
    print("pred: Clara")
    time.sleep(5.2) #Tiempo para llegar a valvula
    Encender_Valvulas(hilo)
    Expulsadas = Expulsadas + 1
    time.sleep(1) #Tiempo funcionando valvula
    Apagar_Valvulas(hilo)

return answer

def camara(hilo):

    # codigo prediccion
    # tomar foto y proceso de prediccion
    if hilo==1:
        image = frame1.copy()
    elif hilo==2:
        image = frame2.copy()

    objeto = image
    objeto = objeto[y1:y2,x1:x2]
    objeto = cv2.resize(objeto,(460,345),interpolation=cv2.INTER_CUBIC)
    cv2.imwrite(Datos+'/objeto_{}.jpg'.format(hilo),objeto)
    print('Imagen guardada:'+'{/objeto_{}.jpg'.format(hilo))

    # predict('./sacha ichi/objeto_{}.jpg'.format(hilo), hilo)

# contadores para visualizar en el LCD
countC1 = 0; # contador de imagenes tomadas por la camara1
countC2 = 0; # contador de imagenes tomadas por la camara2
semillas = 0; # total semillas detectadas
Expulsadas = 0; # total semillas expulsadas
Seleccionadas = 0; # total semillas selccionadas o claras
tiempoMotor = 0; #tiempo de funcionamiento Motor dosificador 2

# Codigo principal de control
while True:

    ret1, frame1 = cap1.read() #verificacion de camaras funcionando
    if ret1 == False: break

    ret2, frame2 = cap2.read()
    if ret2 == False: break

```

```

lcd.lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
lcd.lcd_string("Clasificador", 2)
lcd.lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
lcd.lcd_string("Inicializando", 2)

time.sleep(3)
#lectura de señales de entrada pulsadores y sensores infrared
inicio = GPIO.input(on)
apagado_emergencia = GPIO.input(emerg)
sensor1 = GPIO.input(sen1)
sensor2 = GPIO.input(sen2)

if inicio == True and apagado_emergencia == False:
    time.sleep(0.3)
    # encender motores
    Encender_Motores_Banda()
    Encender_Motores_DosificadorGD()
    lcd.lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
    lcd.lcd_string("SS:"+str(semillas)+ "SE:"+ str(Expulsadas), 1)
    lcd.lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
    lcd.lcd_string("SB1:"+str(countC1)+ "SB2:"+str(countC2), 1)

    if sensor1 == 0:
        time.sleep(3.2)#Tiempo para llegar a la camara 1
        semillas = semillas + 1
        countC1 = countC1 +1
        lcd.lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
        lcd.lcd_string("SS:"+str(semillas)+ "SE:"+ str(Expulsadas),
1)

        lcd.lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
        lcd.lcd_string("SB1:"+str(countC1)+ "SB2:"+str(countC2), 1)
        camara(1)

    if sensor2 == 0:
        time.sleep(3.2) #tiempo para llegar a la camara 2
        semillas = semillas + 1
        countC2 = countC2 + 1
        lcd.lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
        lcd.lcd_string("SS:"+str(semillas)+ "SE:"+ str(Expulsadas),
1)

        lcd.lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
        lcd.lcd_string("SB1:"+str(countC1)+ "SB2:"+str(countC2), 1)
        camara(2)

```

```

elif apagado_emergencia == True and inicio == True:
    time.sleep(0.3)
    lcd.lcd_byte(lcd.LCD_LINE_1, lcd.LCD_CMD)
    lcd.lcd_string("SS:"+str(semillas)+ "SE:"+ str(Expulsadas),
1)

    lcd.lcd_byte(lcd.LCD_LINE_2, lcd.LCD_CMD)
    lcd.lcd_string("SB1:"+str(countC1)+ "SB2:"+str(countC2), 1)
    Apagar_Motores_Banda()
    Apagar_Motores_Dosificador()
    Apagar_Valvulas()

else:
    pwm_1.stop()
    pwm_2.stop()
    Apagar_Valvulas()
    #finalizar pwm / motores apagados

cap1.release()
cap2.release()
cv2.destroyAllWindows()

```

**MANUAL DEL USUARIO DE
LA MÁQUINA
CLASIFICADORA DE
SEMILLAS DE SACHA
INCHI**

Descripción de las partes de la máquina clasificadora

1. Tolva
2. Dosificador de rodillo
3. Dosificador de precisión
4. Bandas transportadoras
5. Motores de dosificadores y bandas transportadoras
6. Recamaras de iluminación
7. Cámaras webs
8. Sensores infrarrojos
9. Válvulas neumáticas
10. Depósito de semillas seleccionadas
11. Depósito de semillas expulsadas
12. Caja y panel de control
13. Fuente de comutada alimentación 12V a 10A
14. Adaptador para Raspberry 5V a 3A
15. Unidad de mantenimiento de aire

Descripción de las partes del panel de control

1. Botón de inicio
2. Botón de parada de emergencia
3. Pantalla LED

**ETAPA 1
FLUACION DE LA MÁQUINA**

Siempre se recomienda posicionar la máquina en una superficie de tal manera que quede estable y no tenga movimiento. Esto otorga:

1. Facilidad de trabajo
2. Eficiencia
3. Rapidez
4. Precisión

**ETAPA 2
USO DE LA MÁQUINA**

1. Conectar la fuente comutada de alimentación de 12v a la red eléctrica de 110V o 220V.
2. Conectar el adaptador para Raspberry a la red eléctrica de 110V.
3. Conectar la manguera de aire al compresor.
4. Verificar que las luces de las recamaras de iluminación y de la pantalla LED estén encendidas.
5. Esperar 5 segundos para permitir inicialización de la Raspberry.
6. Presionar el botón de inicio.
7. Verificar que los 4 motores estén funcionando.
8. Agregar semillas en la tolva
9. Finalmente esperar que se dosifiquen las semillas a las bandas para continuar con el proceso de selección.

Recomendaciones

- No exponga la máquina y ninguna de sus partes al agua o la humedad.
- No exponga la máquina a fuentes de calor elevado; está diseñada para un funcionamiento fiable a temperatura ambiental normal.
- No colocar la máquina en una superficie conductora ya que la estructura es metálica y puede conducir corriente con facilidad.
- Coloque la máquina sobre una superficie estable y sin vibraciones, que le permita funcionar sin perturbaciones externas.
- Evite manipular las conexiones eléctricas o dispositivos eléctricos mientras la máquina este encendida.
- Utilice la máquina en un sitio bien ventilado, no lo cubra o coloque elementos encima durante su uso.

Errores frecuentes

- En ocasiones los motores de los dosificadores suelen frenarse por atascamiento de semillas en estas etapas.
- Puede ocurrir que semillas no sean debidamente detectadas o no se procesen correctamente, principalmente por tener un tamaño más pequeño de lo normal.



2 Features

2.1 Hardware

- Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
- 1, 2 and 4 Gigabyte LPDDR4 RAM options
- H.265 (HEVC) hardware decode (up to 4Kp60)
- H.264 hardware decode (up to 1080p60)
- VideoCore VI 3D Graphics
- Supports dual HDMI display output up to 4Kp60

2.2 Interfaces

- 802.11 b/g/n/ac Wireless LAN
- Bluetooth 5.0 with BLE
- 1x SD Card
- 2x micro-HDMI ports supporting dual displays up to 4Kp60 resolution
- 2x USB2 ports
- 2x USB3 ports
- 1x Gigabit Ethernet port (supports PoE with add-on PoE HAT)
- 1x Raspberry Pi camera port (2-lane MIPI CSI)
- 1x Raspberry Pi display port (2-lane MIPI DSI)
- 28x user GPIO supporting various interface options:
 - Up to 6x UART
 - Up to 6x I2C
 - Up to 5x SPI
 - 1x SDIO interface
 - 1x DPI (Parallel RGB Display)
 - 1x PCM
 - Up to 2x PWM channels
 - Up to 3x GPCLK outputs

2.3 Software

- ARMv8 Instruction Set
- Mature Linux software stack
- Actively developed and maintained
 - Recent Linux kernel support
 - Many drivers upstreamed
 - Stable and well supported userland
 - Availability of GPU functions using standard APIs



Symbol	Parameter	Minimum	Maximum	Unit
VIN	5V Input Voltage	-0.5	6.0	V

Table 2: Absolute Maximum Ratings

Please note that VDD_IO is the GPIO bank voltage which is tied to the on-board 3.3V supply rail.

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V _{IL}	Input low voltage ^a	VDD_IO = 3.3V	-	-	TBD	V
V _{IH}	Input high voltage ^a	VDD_IO = 3.3V	TBD	-	-	V
I _{IL}	Input leakage current	TA = +85°C	-	-	TBD	μA
C _{IN}	Input capacitance	-	-	TBD	-	pF
V _{OL}	Output low voltage ^b	VDD_IO = 3.3V, IOL = -2mA	-	-	TBD	V
V _{OH}	Output high voltage ^b	VDD_IO = 3.3V, IOH = 2mA	TBD	-	-	V
I _{OL}	Output low current ^c	VDD_IO = 3.3V, VO = 0.4V	TBD	-	-	mA
I _{OH}	Output high current ^c	VDD_IO = 3.3V, VO = 2.3V	TBD	-	-	mA
R _{PU}	Pullup resistor	-	TBD	-	TBD	kΩ
R _{PD}	Pulldown resistor	-	TBD	-	TBD	kΩ

^a Hysteresis enabled

^b Default drive strength (8mA)

^c Maximum drive strength (16mA)

Table 3: DC Characteristics

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t _{rise}	10-90% rise time ^a	-	TBD	-	ns
Digital outputs	t _{fall}	90-10% fall time ^a	-	TBD	-	ns

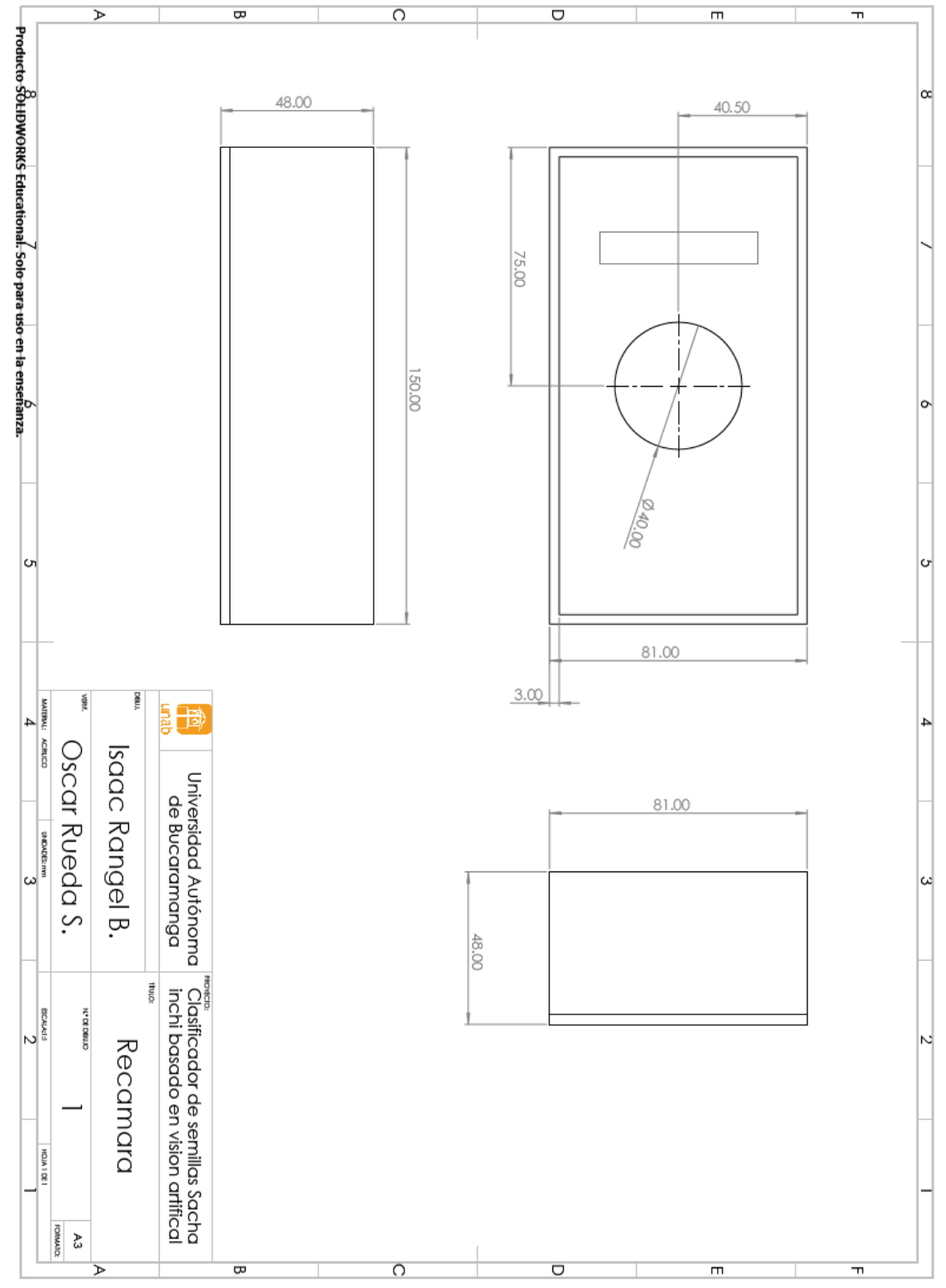
^a Default drive strength, CL = 5pF, VDD_IO = 3.3V

Table 4: Digital I/O Pin AC Characteristics

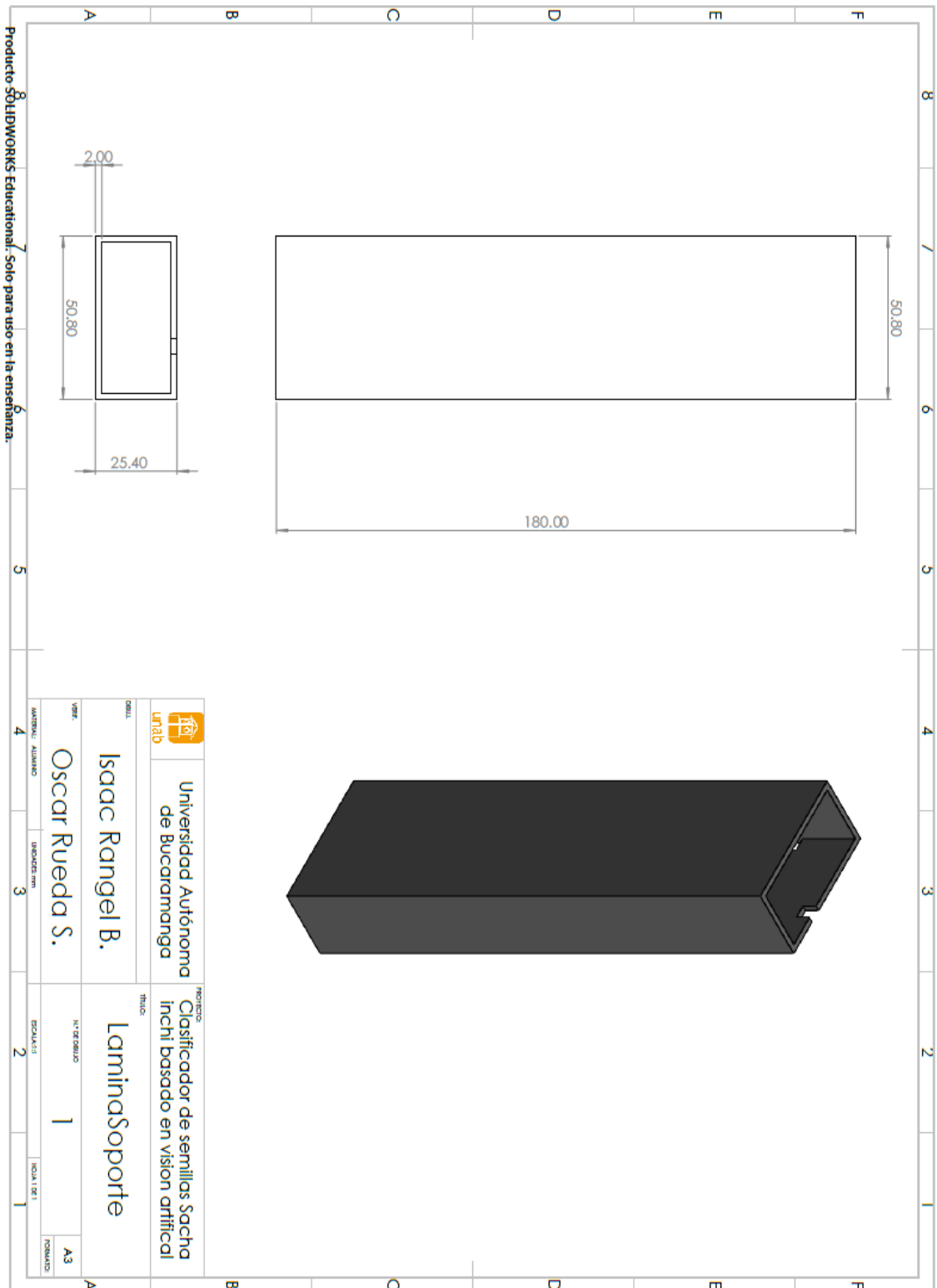


Figure 2: Digital IO Characteristics

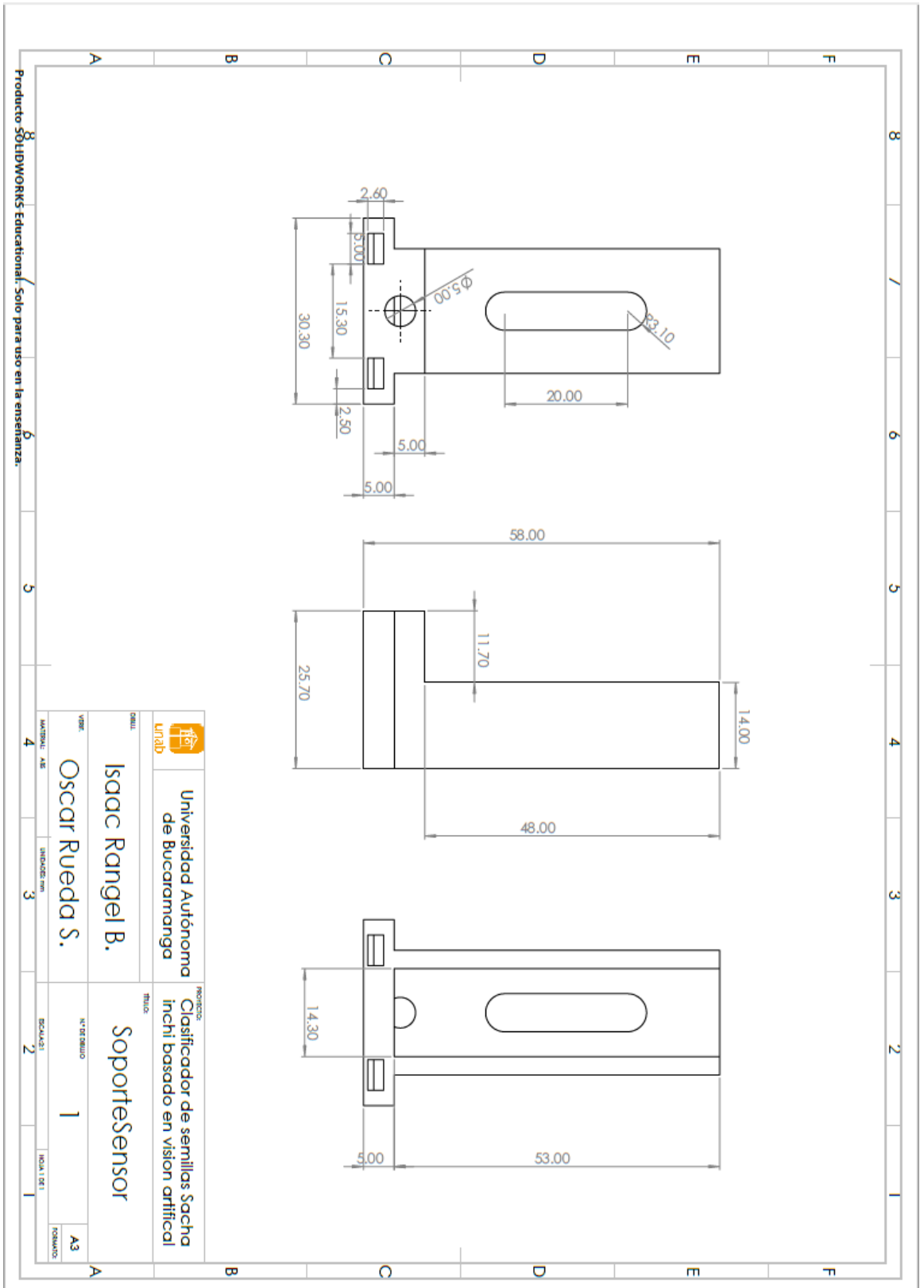
Anexo 6 Plano de dimensiones de recamara de iluminación



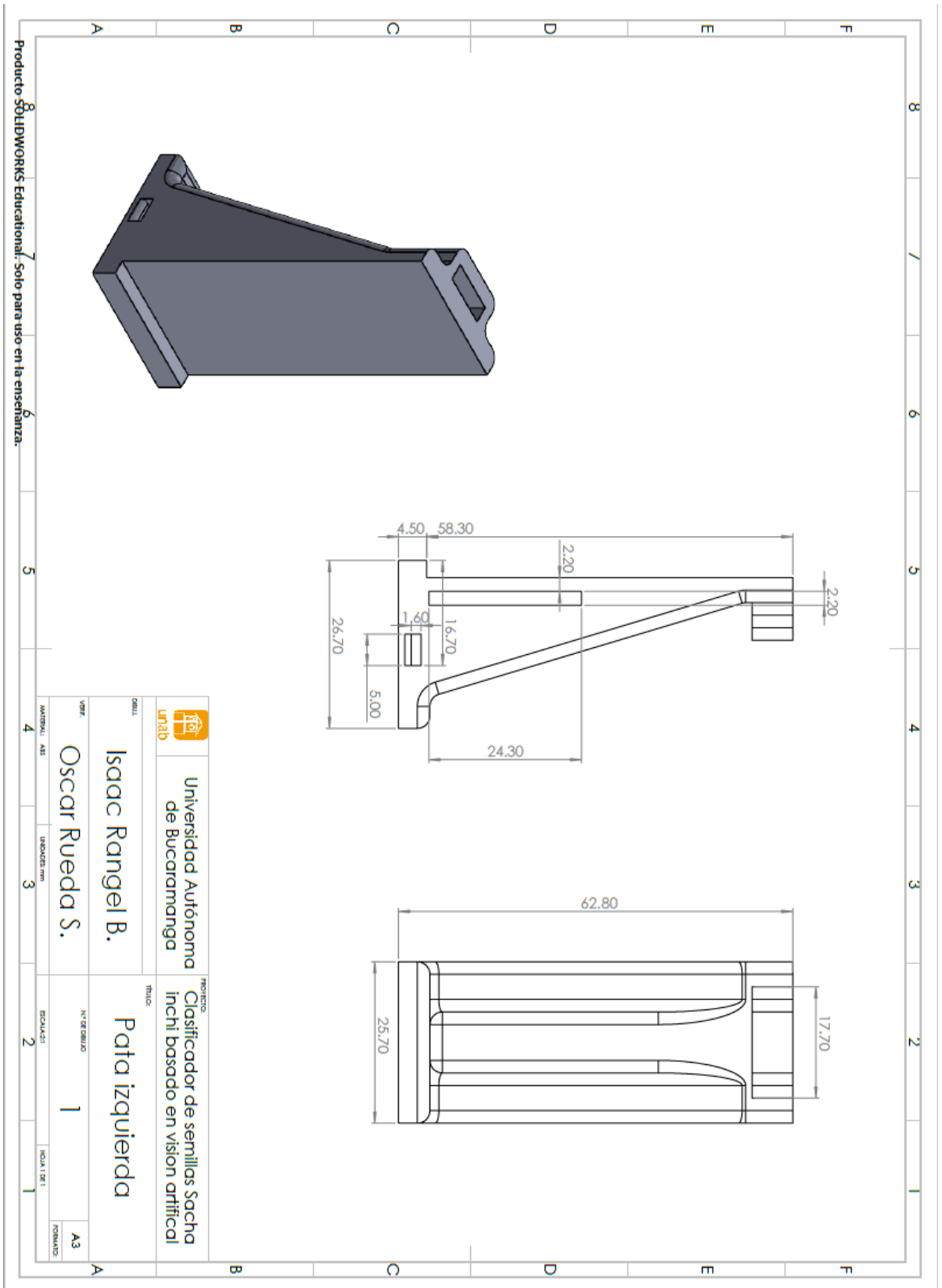
Anexo 7 Plano del soporte de la recamara de iluminación



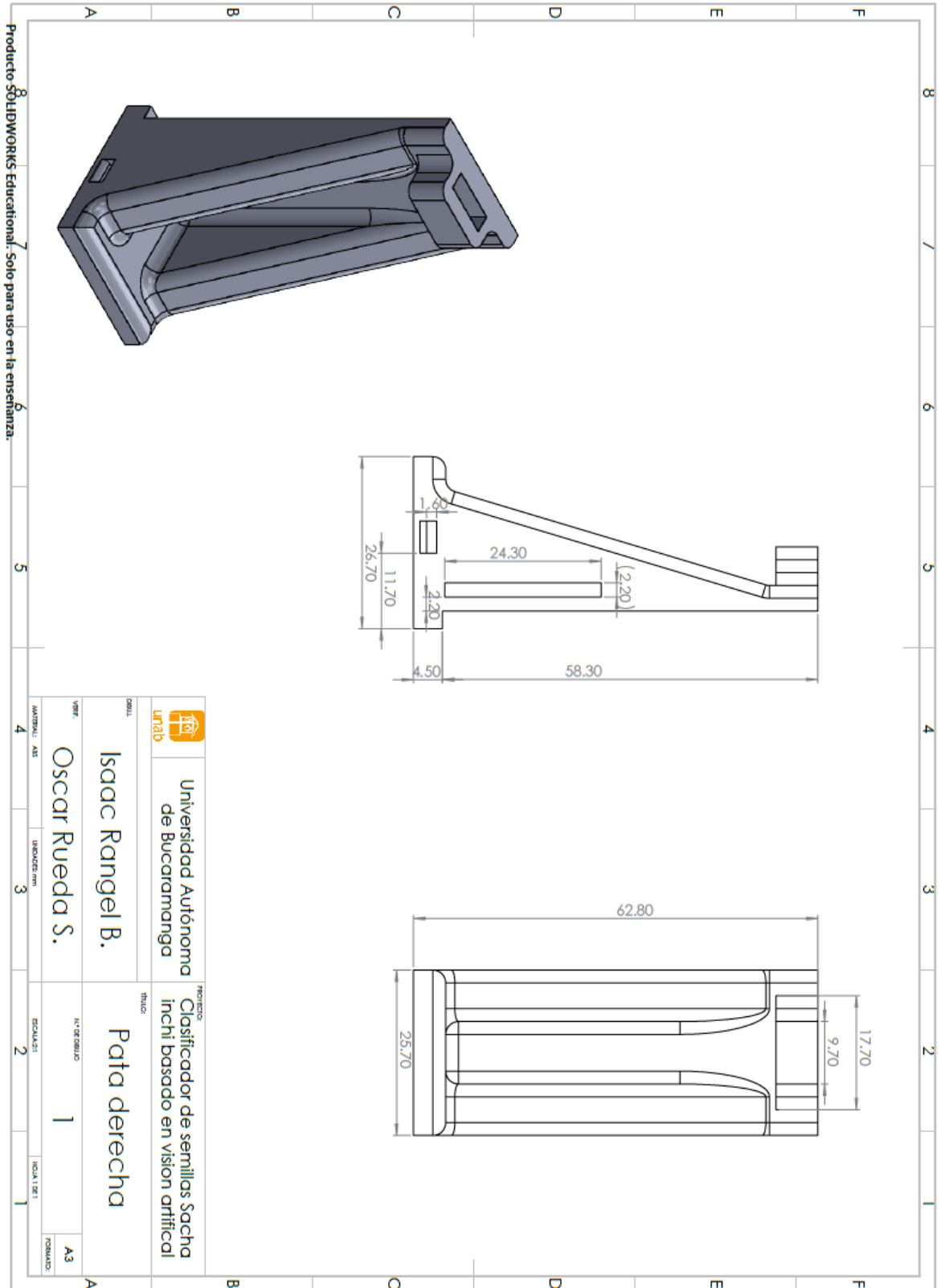
Anexo 8 Plano del soporte del sensor infrarrojo



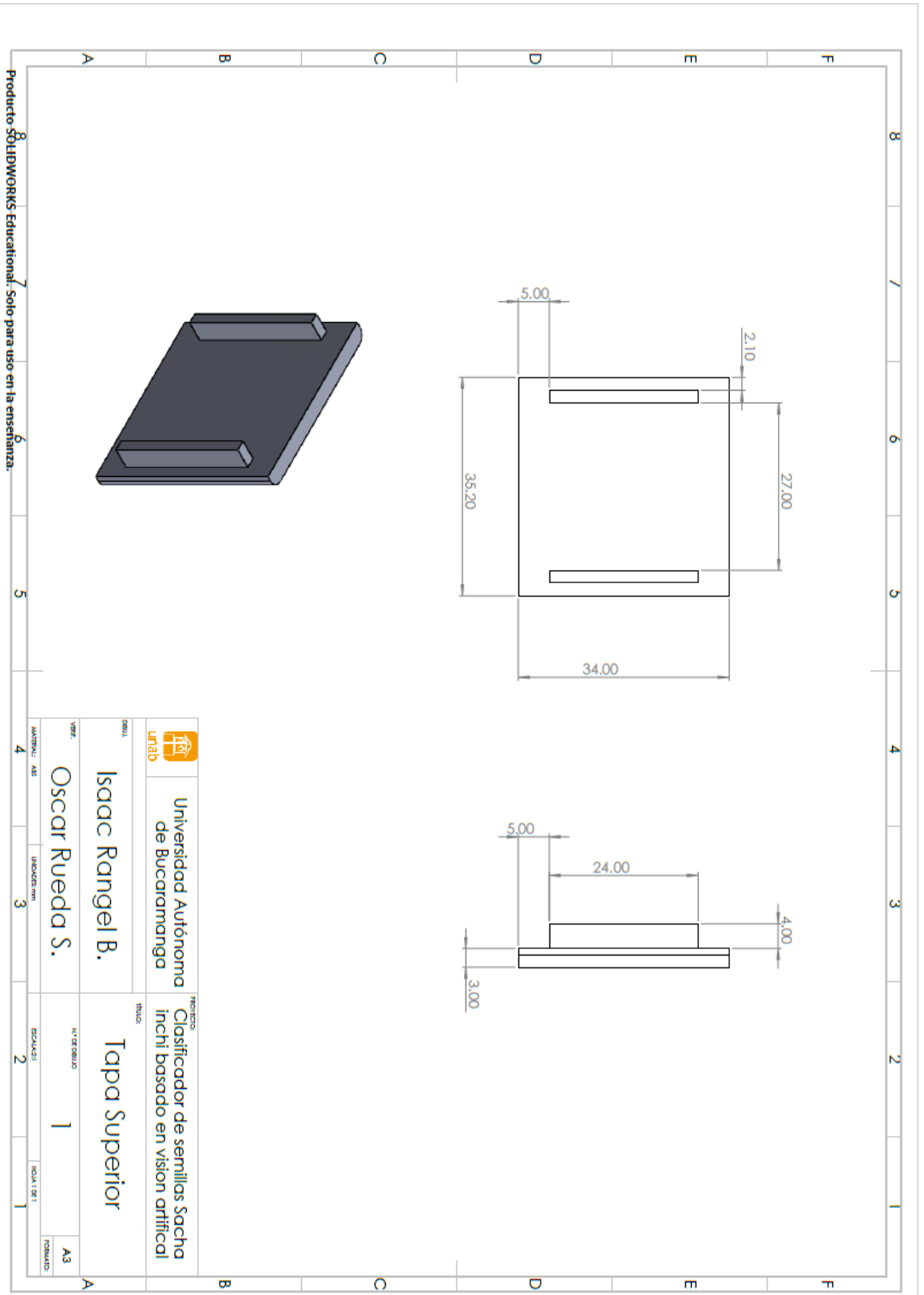
Anexo 9 Pata izquierda del soporte de la válvula



Anexo 10 Pata derecha del soporte de la válvula



Anexo 11 Tapa frontal del soporte de la válvula



Anexo 12 Tapa superior del soporte de la válvula

