



**UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR**

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA**

**SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE PRODUCTOS
BÁSICOS EN LA PERCHA DE UN SUPERMERCADO**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES: Bogran Mathew Aviles Segarra
John Cristhian González Rosado
TUTOR: Ing. Jonathan Salvador Paillacho Corredores M.Sc.

Guayaquil - Ecuador
2025

A handwritten signature in blue ink, located in the bottom right corner of the page. The signature is stylized and appears to be the name of the author or tutor.

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Bogran Mathew Aviles Segarra** con documento de identificación N° **0957030018** y **John Cristhian González Rosado** con documento de identificación N° **0955877253**; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 27 de Febrero del año 2025

Atentamente,



Bogran Mathew Aviles Segarra
0957030018



John Cristhian González Rosado
0955877253

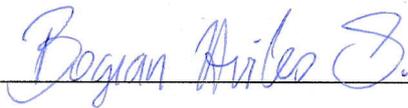
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA
UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Bogran Mathew Aviles Segarra** con documento de identificación N° **0957030018** y **John Cristhian González Rosado** con documento de identificación N° **0955877253**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE PRODUCTOS BÁSICOS EN LA PERCHA DE UN SUPERMERCADO**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

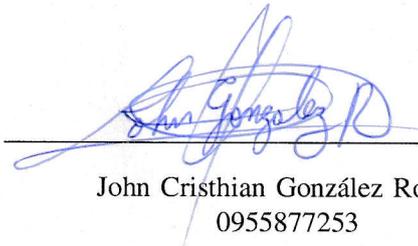
En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 27 de Febrero del año 2025

Atentamente,



Bogran Mathew Aviles Segarra
0957030018



John Cristhian González Rosado
0955877253

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Jonathan Salvador Paillacho Corredores**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE PRODUCTOS BÁSICOS EN LA PERCHA DE UN SUPERMERCADO**, realizado por **Bogran Mathew Aviles Segarra** con documento de identificación N° **0957030018** y **John Cristhian González Rosado** con documento de identificación N° **0955877253**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 27 de Febrero del año 2025

Atentamente,



Ing. Jonathan Salvador Paillacho Corredores
1718907874

DEDICATORIA

Este trabajo de titulación está dedicado a mi familia, mi padre Bogran Jorge Aviles Abad, sin ti este título nunca pudo haberse hecho realidad, mi madre Elvira Hortencia Segarra Ubilla, gracias por siempre creer en mí, por guiarme con tus palabras y consejos, mi hermano Michael Andreus Aviles Segarra, por tu aliento y confianza en mí, y mi hermana Dara Caroline Aviles Segarra, mi compañera en casi todas mis vivencias. Gracias a Dios por permitirme ser parte de esta familia, orgulloso de formar parte. Finalmente me dedico este trabajo a mí mismo, por todas las dificultades personales que pasé. Estoy muy orgulloso de no haberme rendido nunca y haber seguido siempre con la cabeza en alto.

Bogran Mathew Aviles Segarra

Este trabajo de titulación está dedicado a mi padre John Manuel González Velastegui, quien es el pilar de mi familia gracias a su gran esfuerzo, apoyo constante y sacrificio lograré terminar mis estudios universitarios. A mi madre María Elena Rosado Cerezo quien me alentó a seguir estudiando esta carrera, me enseñó a trabajar arduamente para conseguir y lograr todos mis objetivos, ambos me enseñaron lo importante que es la perseverancia y lo duro que puede ser la vida. A mis hermanos Steven y José González Rosado quienes además de mis padres son las personas que me apoyan en mis triunfos y caídas, han estado en todo momento en cada paso que he dado como persona. A mi abuela Zoila Velastegui Andrade, por sus valores y amor incondicional. Su legado vive en cada página de esta tesis. Agradezco a Dios por permitirme ser parte de esta familia maravillosa. Finalmente me dedico este trabajo de titulación a mí mismo, ya que el mejor regalo que me pueden dejar mis padres son mis estudios.

John Crithian González Rosado

AGRADECIMIENTO

Agradezco a Dios, a mi familia, por siempre estar presentes, por preocuparse y enseñarme lo valioso que es la confianza en una persona, por todas sus enseñanzas inculcadas en mí, por educarme hasta el punto de poder culminar esta etapa en mi vida.

Agradezco a la Universidad Politécnica Salesiana, a los docentes, a mis compañeros que fueron parte importante en este camino.

Agradezco al Ing. Jonathan Salvador Paillacho Corredor, por el tiempo dispuesto, por sus sugerencias y por ser guía innato en este proceso.

Bogran Mathew Aviles Segarra

Agradezco a Dios, a mi familia por estar en los momentos más importa de mi vida, por su amor incondicional, apoyo constante y sacrificio incansable, esta tesis es tan suya como mía.

Agradezco a la Universidad Politécnica Salesiana, los docentes de la carrera de ingeniería en mecánica por sus enseñanzas y consejos, me han brindaron las herramientas para un futuro exitoso para ser excelente ingeniero mecánico.

Agradezco a mis compañeros de estudios, a mis amigos porque me han demostrado ser grandes seres humanos que siempre me han brindado de su apoyo con quienes pasamos momentos divertidos, tristes, estudiar para los exámenes y las enseñanzas que les brinde como tutor.

Agradezco al Ing. Jonathan Salvador Paillacho Corredor, por ser nuestro tutor de tesis y ser un guía en esta última etapa de culminación para convertirme en un profesional.

John Cristhian González Rosado

RESUMEN

El presente proyecto tiene como objetivo principal crear un algoritmo asistido por visión artificial, para realizar reconocimiento de productos básicos en una percha de supermercado, buscando optimizar tiempo, eficiencia y reducir el porcentaje de errores al momento de colocar los productos en las perchas.

Para la visualización del algoritmo se contó con una interfaz de visión artificial que permite clasificar productos de consumo tales como: frutas, bebidas embotelladas, vegetales, enlatados, entre otros.

Para el entrenamiento de la red neuronal se utilizó YOLO como herramienta principal; YOLO destaca en campos de visión artificial, ya que ayuda a optimizar parámetros principales para el reconocimiento de objetos, detección de objetos y conteo de objetos.

Como componentes principales se utilizaron una Raspberry Pi4 con conexión a internet, y una cámara web HD. Para ejecutar el funcionamiento del código se inicializa la terminal de la RPI, utilizando comandos específicos para abrir una interfaz creada para el proyecto. Una vez iniciado el algoritmo de visión artificial, el proyecto comenzará con la detección de productos de consumo básico. La ejecución de la detección de objetos se realizará por medio de la cámara web con detección en tiempo real, funcionando con una resolución de 640 x 480 píxeles; y al momento de capturar los productos la misma dará una mejora en la imagen y cambiará a 1280 x 720 píxeles, comparando los productos con el inventariado del mismo.

Para finalizar, la interfaz presenta una tabla mostrando los resultados del conteo de productos que se muestran en la percha.

Palabras claves: Visión artificial, Raspberry, Python, OpenCv, detección de objetos, conteo de objetos, procesamiento de imágenes, automatización, aplicaciones tecnológicas.

ABSTRACT

The main objective of this project is to create an algorithm assisted by artificial vision to recognize basic products on a supermarket shelf, aiming to optimize time, improve efficiency, and reduce errors when placing products on the shelves.

For the visualization of the algorithm, an artificial vision interface was developed to classify consumer products such as fruits, bottled beverages, vegetables, canned goods, among others.

YOLO was used as the main tool for training the neural network. YOLO excels in artificial vision applications as it helps optimize key parameters for object recognition, object detection, and object counting.

The main components used in this project include a Raspberry Pi 4 with an internet connection and an HD webcam. To execute the code, the RPI terminal is initialized using specific commands to open an interface designed for the project.

Once the artificial vision algorithm is started, the system begins detecting basic consumer products. Object detection is performed in real-time through the webcam, operating at a resolution of 640 x 480 pixels. When capturing products, the resolution enhances to 1280 x 720 pixels, allowing for image improvement and comparison with the inventory database.

Finally, the interface displays a table showing the results of the product count on the shelf. **Keywords:** Computer vision, Raspberry Pi, Python, OpenCV, object detection, object counting, image processing, automation, technological applications.

ÍNDICE

I.	PROBLEMÁTICA	13
II.	JUSTIFICACIÓN	14
III.	OBJETIVOS	15
III-A.	Objetivo general	15
III-B.	Objetivos específicos	15
IV.	FUNDAMENTOS TEÓRICOS	16
IV-A.	Computer Vision (CV)	16
IV-B.	El Aprendizaje Automático y la Relevancia dentro de una IA	16
IV-C.	Tipos de visión por computadora	16
IV-D.	Detección de Objetos	17
IV-E.	Conteo de Objetos	17
IV-F.	Procesamiento a las Imágenes	17
IV-G.	Python	19
IV-H.	Bibliotecas recomendadas para el reconocimiento de objetos	19
IV-I.	Deep Learning	19
IV-J.	Raspberry Pi	20
IV-K.	Facecam 1000x V2	20
V.	MARCO METODOLÓGICO	21
V-A.	Dataset	21
V-B.	Base de datos de productos	21
V-C.	Instalación de paquetes	22
V-D.	Modulo del proyecto	23
V-E.	Elaboración del algoritmo	23
V-F.	Diseño de Percha	24
V-G.	Análisis estático	24
V-H.	Diseño de Base para Cámara Webcam	30
V-I.	Interfaz	30
V-J.	Set up cámara	33
V-K.	Etiquetado de imágenes	33
V-L.	Lista de productos a usar	34
V-M.	Análisis final del Entrenamiento	35
V-N.	Métricas de Rendimiento	36
VI.	RESULTADOS	40
VI-A.	Resultados para entrenamiento	40
VI-B.	Resultados de la interfaz	42
VI-C.	Resultados con el proyecto terminado	42
VI-D.	Verificación de resultados finales	45
VII.	CRONOGRAMA	50
VIII.	PRESUPUESTO	51
IX.	CONCLUSIONES	52
X.	RECOMENDACIONES	53

Apéndice A: Planos	55
Apéndice A: Equipo de Manufactura	72
Apéndice B: Código del Web Scraper	73
Apéndice C: Código del modelo de aprendizaje automático	74

ÍNDICE DE FIGURAS

1.	Captura de imagen digital. Fuente [6].	16
2.	Detección de objetos. Fuente: por B. Aviles y J. González.	17
3.	Funcionamiento de la CNN. Fuente [7].	17
4.	Esquema del conteo de objetos. Fuente [9]	18
5.	Procesamiento de imágenes. Fuente: por B. Aviles y J. González.	18
6.	Raspberry Pi4. Fuente: por B. Aviles y J. González	20
7.	Cámara Facecam 1000x V2. Fuente: por B. Aviles y J. González	20
8.	Dataset compilado de un supermercado Fuente: por B. Aviles y J. González.	21
9.	Base de datos de productos generada compilado de un supermercado Fuente: por B. Aviles y J. González.	21
10.	Instalación para las librerías de python Fuente: por B. Aviles y J. González	22
11.	OpenCv/Linux. Fuente: por B. Aviles y J. González	22
12.	Esquema del proyecto. Fuente: por B. Aviles y J. González.	23
13.	Comparación de modelos YOLO. Fuente: Ultralytics	23
14.	Percha diseñada en 3D. Fuente: por B. Aviles y J. González	24
15.	Lugar donde se aplicara la fuerza para su respectivo análisis. Fuente: por B. Aviles y J. González	25
16.	Análisis en la base para su respectivo estudio. Fuente: por B. Aviles y J. González	25
17.	Tensión ejercida en estante superior. Fuente: por B. Aviles y J. González	26
18.	Desplazamiento ejercido en estante superior. Fuente: por B. Aviles y J. González	26
19.	Deformación de percha por causa del desplazamiento. Fuente: por B. Aviles y J. González	26
20.	Tensión ejercida con una masa mayor para observar resultados. Fuente: por B. Aviles y J. González	27
21.	Desplazamiento del material. Fuente: por B. Aviles y J. González	27
22.	Deformación del estante. Fuente: por B. Aviles y J. González	27
23.	Tensión ejercida en el estante de base. Fuente: por B. Aviles y J. González	28
24.	Desplazamiento ejercido en la base. Fuente: por B. Aviles y J. González	28
25.	Deformación de percha por causa del desplazamiento. Fuente: por B. Aviles y J. González	28
26.	Tensión ejercida en el estante de base. Fuente: por B. Aviles y J. González	29
27.	Desplazamiento ejercido en la base. Fuente: por B. Aviles y J. González	29
28.	Deformación de percha por causa del desplazamiento. Fuente: por B. Aviles y J. González	29
29.	Base de cámara diseñada 3D. Fuente: por B. Aviles y J. González	30
30.	Interfaz del navegador web. Fuente: por B. Aviles y J. González	30
31.	Presentación interfaz. Fuente: por B. Aviles y J. González	31
32.	Interacción con interfaz. Fuente: por B. Aviles y J. González	31
33.	Dataset del Catálogo. Fuente: por B. Aviles y J. González	32
34.	Datos del Backend. Fuente: por B. Aviles y J. González	32
35.	Set up cámara. Fuente: por B. Aviles y J. González	33
36.	Etiquetado manual a los productos. Fuente: por B. Aviles y J. González	34
37.	Dataset propio. Fuente: por B. Aviles y J. González	35
38.	Métricas de rendimiento. Fuente: por B. Aviles y J. González	36
39.	Imágenes de resultados de entrenamiento . Fuente: por B. Aviles y J. González	37
40.	Resultados del entrenamiento. Fuente: por B. Aviles y J. González	38
41.	Diagrama de flujo. Fuente: por B. Aviles y J. González	39
42.	Matriz de confusión. Fuente: por B. Aviles y J. González	40
43.	Labels. Fuente: por B. Aviles y J. González	41
44.	Curvas de entrenamiento. Fuente: por B. Aviles y J. González	41
45.	Interfaz. Fuente: por B. Aviles y J. González	42
46.	Iniciar algoritmo. Fuente: por B. Aviles y J. González	42
47.	Algoritmo en funcionamiento. Fuente: por B. Aviles y J. González	43
48.	Gráfico del barra. Fuente: por B. Aviles y J. González	44
49.	Baja Iluminación. Fuente: por B. Aviles y J. González	45

50.	Captura del algoritmo con baja iluminación. Fuente: por B. Aviles y J. González	45
51.	Iluminación media. Fuente: por B. Aviles y J. González	46
52.	Captura del algoritmo con iluminación media. Fuente: por B. Aviles y J. González	46
53.	Alta Iluminación. Fuente: por B. Aviles y J. González	47
54.	Captura del algoritmo con alta iluminación. Fuente: por B. Aviles y J. González	47
55.	Luz natural. Fuente: por B. Aviles y J. González	48
56.	Captura del algoritmo con luz natural. Fuente: por B. Aviles y J. González	48
57.	Iluminación Regular. Fuente: por B. Aviles y J. González	49
58.	Base Percha, por B. Aviles y J. González, Plano de pieza	55
59.	Soporte delantero Base, por B. Aviles y J. González, Plano de pieza	56
60.	Soporte delantero Lateral, por B. Aviles y J. González, Plano de pieza	57
61.	Pared Base, por B. Aviles y J. González, Plano de pieza	58
62.	Soporte Pie Amigo, por B. Aviles y J. González, Plano de pieza	59
63.	Base Superior, por B. Aviles y J. González, Plano de pieza	60
64.	Base Posterior, por B. Aviles y J. González, Plano de pieza	61
65.	Soporte posterior Lateral, por B. Aviles y J. González, Plano de pieza	62
66.	Soporte posterior Base, por B. Aviles y J. González, Plano de pieza	63
67.	Soporte Pie Amigo Base, por B. Aviles y J. González, Plano de pieza	64
68.	Base Cámara, por B. Aviles y J. González, Plano de pieza	65
69.	Eslabón Base, por B. Aviles y J. González, Plano de pieza	66
70.	Base Cámara, por B. Aviles y J. González, Plano de pieza	67
71.	Soporte Eslabón, por B. Aviles y J. González, Plano de pieza	68
72.	Soporte Cámara, por B. Aviles y J. González, Plano de pieza	69
73.	Lámina del contorno en madera mdf, por B. Aviles y J. González, Diseño en modelos 3D	70
74.	Lámina del prototipado de las piezas en plástico PETG, por B. Aviles y J. González, Diseño en modelos 3D	71
75.	Máquina de Corte láser para las piezas en madera mdf, por B. Aviles y J. González	72
76.	Impresora 3D para enviar a imprimir las piezas diseñadas, por B. Aviles y J. González	72

ÍNDICE DE TABLAS

I.	Distribución de Clases en <i>train2017</i> y <i>val2017</i>	35
II.	Métricas de Validación	36
III.	Cronograma. Fuente: por B. Aviles y J. González	50
IV.	Tabla de Presupuesto. Fuente: Autor	51

I. PROBLEMÁTICA

Hace más de diez mil años, comenzaron las transacciones comerciales con el propósito principal de intercambiar productos o practicar el trueque. Con el tiempo, este proceso evolucionó hacia una modalidad de compra y venta, en la cual cuando una persona o una empresa posee algunos bienes o servicios a su favor se pueden vender a cambio de una compensación económica, marcando así un hito en la evolución del comercio. Con el pasar de los años se ha buscado la manera de facilitar el sistema de inventariado[1].

En los supermercados, es crucial una administración eficaz del inventario y una adecuada presentación de los productos para prevenir pérdidas financieras causadas por la falta o exceso de inventario. No obstante, los procedimientos manuales convencionales empleados para supervisar y documentar el stock son susceptibles a fallos humanos, son caros en cuanto a tiempo y recursos, y no facilitan una actualización en tiempo real.

Además, la falta de un sistema automatizado que ayude en la facilidad de identificar y contar los productos en las perchas provoca problemas para asegurar su disponibilidad, impactando de manera negativa en la experiencia del cliente y la eficacia en las operaciones. Por lo tanto, se presenta la exigencia de crear una solución tecnológica fundamentada en la utilización de cámaras y métodos de aprendizaje profundo que automatiza el proceso de identificación, recuento y identificación de residuos.

De acuerdo a estudios recientes, se sugiere que un sistema de inventario es un mecanismo diseñado para gestionar las existencias y decidir cuánto y cuándo pedir de cada ítem[2].

Además, los inventarios reflejan la presencia de productos almacenados para diversos propósitos como compra, alquiler, venta, uso o transformación. Estos constituyen uno de los recursos fundamentales para cualquier empresa comercial o industrial. La gestión de productos es crucial, ya que afecta directamente su éxito comercial y su capacidad para satisfacer las necesidades de sus clientes. Un manejo adecuado de productos implica planificar, adquirir, almacenar, distribuir y vender mercancías de manera eficiente y rentable. Esto permite mantener un inventario óptimo, evitar la obsolescencia de productos, mejorar la rotación de existencias y optimizar los recursos disponibles [3].

Los sistemas que utilizan visión artificial no son adecuados o no funcionan adecuadamente debido a la falta de condiciones de operabilidad confiable, por ejemplo la falta de luz en la toma de las imágenes puede causar que los datos tomados no sean representaciones adecuadas del objeto analizado, si el sensor no logra capturar la información con precisión, puede producirse interferencia, lo que lleva a una interpretación errónea por parte del procesador de imágenes o falta de algoritmos de procesamiento de imágenes adecuados.

II. JUSTIFICACIÓN

El manejo adecuado del inventario es fundamental para la gestión efectiva de los negocios, ya que influye significativamente en las necesidades que demandan los clientes, los gastos operativos, el cronograma de la producción y la administración de la cadena en el suministro. Asimismo, un control eficiente en el inventario puede contribuir a la disminución de los costos relacionados con el almacenamiento, mitigar el riesgo de obsolescencia y optimizar la gestión de los recursos empresariales.

Este proyecto se enfoca en la gestión de exhibición del producto en un supermercado, para conocer el número de productos y gestionar de manera eficiente el inventario de los mismos. Se plantea emplear una cámara, instalada frente a la percha, para capturar todos los productos. Se va a utilizar técnicas de aprendizaje profundo (Deep Learning) para detectar, contar y reconocer cada producto. Una vez detectado todos los productos se compara en el sistema para verificar el stock existente. Los resultados de la detección y reconocimiento de los productos serán informados al personal responsable de llevar el inventario de los productos [4].

La visión artificial se tornó en una herramienta solicitada para una optimización en diferentes ámbitos, su eficacia resalta en el funcionamiento, tales como iluminación necesaria, exactitud en la recolección de datos y sus algoritmos de procesamiento. La pérdida de una de estas condiciones puede dejar en riesgo la fiabilidad del sistema, obteniendo información incorrecta y disminuyendo su utilidad en escenarios reales. Lo que hace destacar la importancia de crear soluciones que reduzcan estas restricciones, incorporando tecnologías y métodos sofisticados que habiliten a los sistemas de visión artificial para ajustarse a situaciones difíciles, optimizando su rendimiento y ampliando su espectro de usos prácticos.

Con este sistema, el reabastecimiento de productos puede ser más fácil para el personal y los clientes pueden encontrar fácilmente los artículos que buscan. Se pretende lograr avances en la recopilación de datos de imágenes almacenadas, en los etiquetados y en el reconocimiento de productos, utilizando el algoritmo para validar información, más la interfaz que me permita gestionar los productos en inventario.

El proyecto tendrá como estructura una percha con iluminación, una cámara para el reconocimiento de productos básicos específicos. La cámara tendrá como capturadora o almacenamiento de datos una Raspberry Pi la cual mandará información al algoritmo de procesamiento para después usar la interfaz y así obtener datos en tiempo real. Un claro ejemplo de que el control del inventario mediante visión artificial tiene un impacto positivo se desarrolló en México. Una cadena de supermercados de dicho país innovó en utilizar en una de sus tiendas cámaras con inteligencia artificial para la venta de productos de manera automática optimizando tiempo y costos en contratación de personal que esté veinticuatro horas en el local [5].

III. OBJETIVOS

III-A. Objetivo general

Desarrollar un algoritmo de visión artificial que realice la identificación de productos básicos en la estantería de un supermercado para la reposición de producto de acuerdo al consumo registrado.

III-B. Objetivos específicos

- Establecer una base de datos de los productos para la estandarización del control del inventario de productos.
- Implementar el sistema de visión por computadora para la identificación de los productos en percha.
- Crear una interfaz de usuario que permita el control de la frecuencia en la reposición de cada producto según las diferentes condiciones de ventas.

IV. FUNDAMENTOS TEÓRICOS

IV-A. Computer Vision (CV)

La Visión por Computador (VC) representa un área de gran actividad dentro de la Inteligencia Artificial (IA), dedicada a dotar a los sistemas informáticos de la habilidad para comprender y analizar información visual de forma análoga tal como lo interprete el ojo humano. Al emplear extensos conjuntos de datos junto con algoritmos especializados, estos sistemas son capaces de examinar tanto imágenes como videos para extraer y procesar información útil. Como resultado, las máquinas pueden “observar” e interpretar datos visuales, lo que las faculta para tomar decisiones basadas en dichas observaciones. Se diferencia del mero procesamiento de imágenes, cuyo objetivo principal suele ser modificar o ajustar archivos visuales, ya que la Visión por Computador busca comprender lo que realmente sucede en las escenas, replicando de cierto modo como las personas interpretan lo que ven.

IV-B. El Aprendizaje Automático y la Relevancia dentro de una IA

En los ámbitos de las tecnologías avanzadas, una Visión por Computadora se ha transformado en un pilar fundamental para numerosos proyectos de IA y aprendizaje automático (ML). La adopción de técnicas de aprendizaje alternos, particularmente redes neuronales convolucionales (CNN), han catapultado el desempeño de la Visión por una Computadora, logrando tasas de acierto cada vez más sobresalientes. Para los profesionales del aprendizaje automático, la VC proporciona estrategias sólidas para automatizar y agilizar procesos complejos, por ejemplo, la anotación de datos y el entrenamiento de modelos, impulsando la evolución y eficiencia en múltiples campos de aplicación.

IV-C. Tipos de visión por computadora

La visión artificial se clasifica en cuatro secciones como se puede observar en la figura 1. Las cuales son: Clasificación, Localización, Detección de objetos y Segmentación semántica.

Cuando se trata de solo un objeto se procede con la parte de clasificación única o para subir un grado de dificultad se suma la localización. Para detectar varios objetos se opta por la detección de objetos o para un trabajo más exacto, con segmentación semántica.

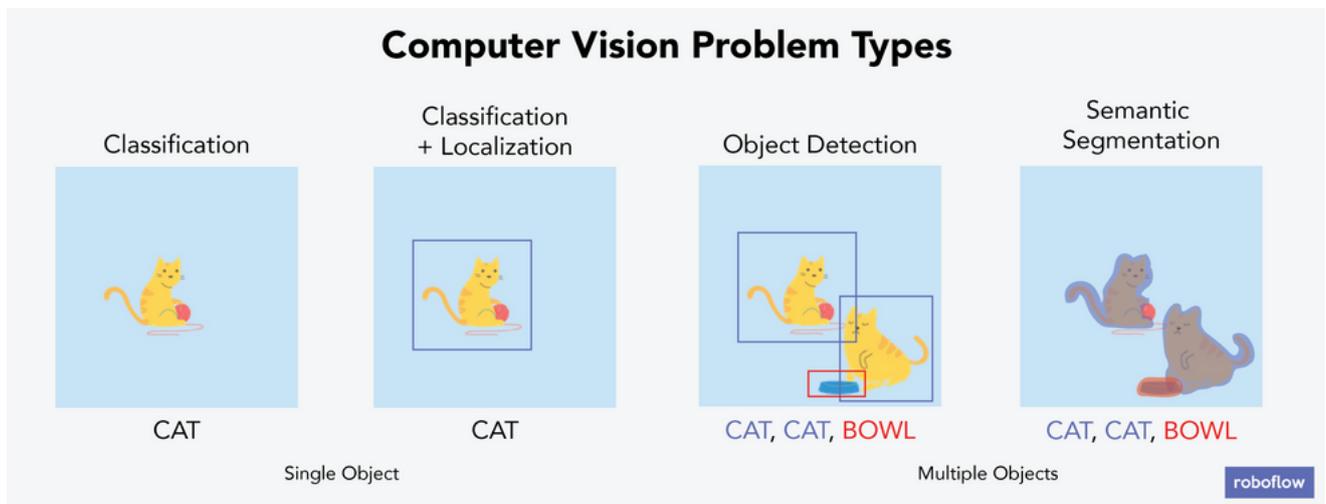


Figura 1. Captura de imagen digital. Fuente [6].

IV-D. Detección de Objetos

La localización o detección en los objetos es una parte fundamental para la visión asistida por una computadora, ya que permite a las máquinas localizar e identificar elementos específicos como las imágenes o los videos. A diferencia de la clasificación de imágenes, que solo determina la presencia de un objeto, la detección va más allá al marcar su ubicación con cuadros delimitadores, facilitando así su identificación dentro del entorno visual.

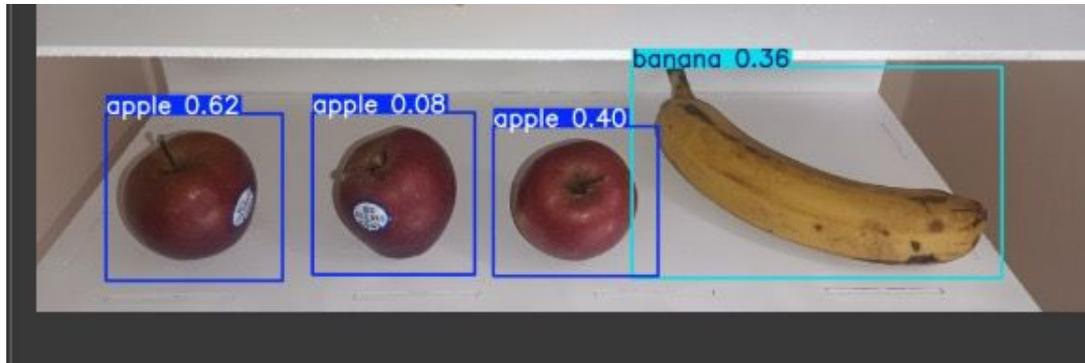


Figura 2. Detección de objetos. Fuente: por B. Aviles y J. González.

Para ello se suelen utilizar algoritmos sofisticados, a menudo basados en Redes Neuronales Convolucionales (CNN), que aprenden a reconocer patrones y rasgos que caracterizan a los distintos objetos. La precisión en modelos para la detección en los objetos suele evaluarse utilizando métricas como la Intersección sobre la Unión (IoU) y la Precisión Media (mAP).

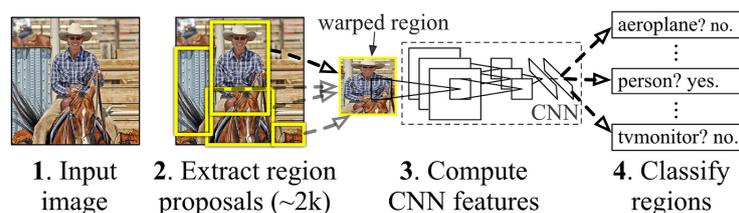


Figura 3. Funcionamiento de la CNN. Fuente [7].

IV-E. Conteo de Objetos

El conteo de objetos implica determinar la cantidad de veces que un objeto específico aparece en una sola imagen o en varias serie de imágenes. Esta técnica resulta valiosa para registrar, supervisar y gestionar distintos procesos técnicos. Un paso fundamental en la visión computacional para reconocer objetos es dividir la imagen en partes significativas, lo que se conoce como segmentación. Una de las estrategias para segmentar imágenes consiste en detectar sus bordes. Este proceso permite condensar la información de bajo nivel, reduciendo millones de píxeles a solo unas cuantas regiones, lo que agiliza la identificación de objetos. Para segmentar o delimitar estas áreas, se emplean características como la intensidad, la textura, el color y el gradiente. En la visión computacional de nivel intermedio, se parte de la idea de que los píxeles de un mismo objeto presentan propiedades similares. [8].

IV-F. Procesamiento a las Imágenes

En el procesamiento de las imágenes se implica una serie de técnicas destinadas a mejorar su calidad, reducir el ruido y resaltar sus características más importantes. Estas herramientas son esenciales para garantizar una detección y un conteo precisos. [10].

Conteo por inteligencia artificial

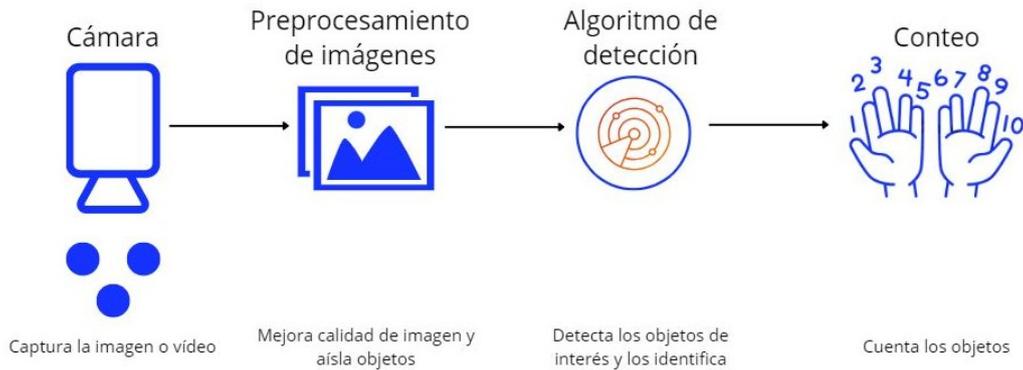


Figura 4. Esquema del conteo de objetos. Fuente [9]

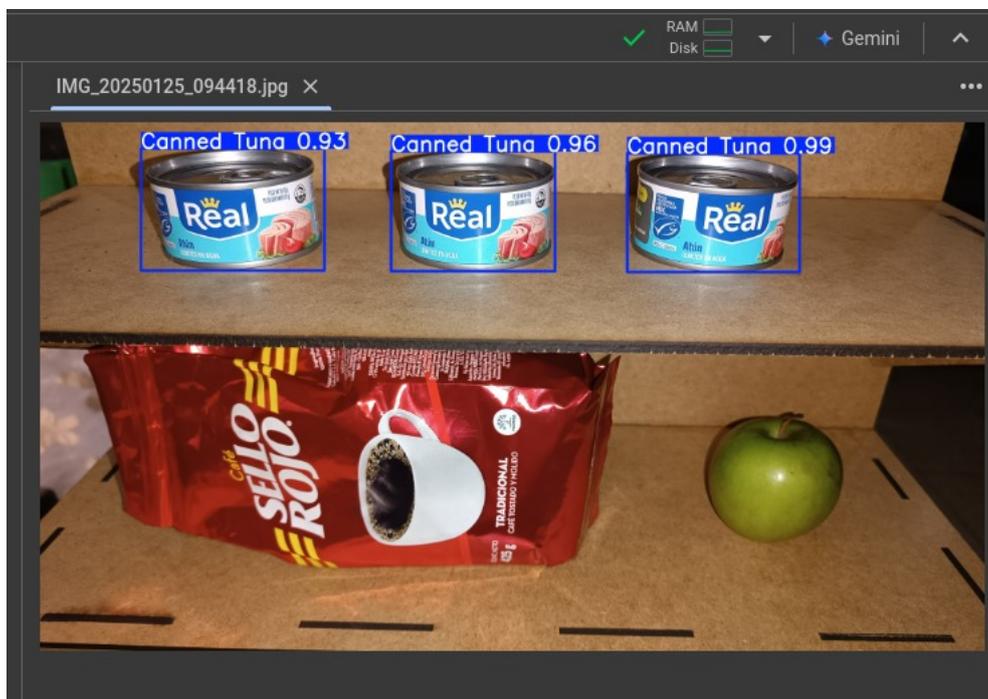


Figura 5. Procesamiento de imágenes. Fuente: por B. Aviles y J. González.

Para realizar el cálculo de superpíxeles se basa a las agrupaciones de píxeles con características similares, preservando los bordes de la imagen. Algunos enfoques para generar superpíxeles se consideran técnicas de segmentación y pueden aplicarse mediante métodos jerárquicos. [11].

Una vez que el modelo de reconocimiento de imágenes analiza el video de monitoreo, resulta sencillo obtener las coordenadas de los objetos en cada fotograma.[12].

IV-G. Python

Python es un lenguaje de programación versátil, interpretado y de alto nivel, creado por Guido van Rossum y lanzado en 1991. Su sintaxis sencilla y fácil de leer lo hace ideal para diversas aplicaciones, como el desarrollo de software, la ciencia de datos y la inteligencia artificial. [13].

Python se emplea extensamente en diversos ámbitos gracias a su versatilidad y a la multigama de bibliotecas y frameworks que provee. Estas herramientas adicionales ofrecen funcionalidades específicas para diferentes tareas, como el procesamiento de imágenes con bibliotecas como OpenCV.

IV-H. Bibliotecas recomendadas para el reconocimiento de objetos

- **imutils:** Biblioteca utilizada para recuperar imágenes de la cámara para crear un flujo de vídeo [14].
- **OpenCV:** Fue lanzado como un proyecto de investigación dentro de una corporación, con el propósito de impulsar tecnologías en aplicaciones que requieren un uso intensivo de la CPU. Los objetivos de este proyecto fueron los siguientes:
 - * Impulsar la investigación en visión al ofrecer no solo código abierto, sino también optimizado para la infraestructura básica de visión.
 - * Facilitar la difusión del conocimiento en visión al proporcionar una infraestructura común sobre la cual los desarrolladores puedan construir, simplificando así la legibilidad y la transferencia de código.
 - * Promover el avance de las aplicaciones comerciales basadas en visión al ofrecer código portátil y optimizado para el rendimiento de forma gratuita, con una licencia que no exija que las aplicaciones sean abiertas o gratuitas [15].
- **NumPy:** NumPy posibilita la creación de una estructura de datos versátil, lo que conlleva a una mejora en el análisis de datos al permitir el intercambio de datos entre diversos algoritmos. [16].
- **PyTorch:** Un marco de aprendizaje profundo de código abierto conocido por su flexibilidad y facilidad de uso. La librería PyTorch permite gráficos de cálculo dinámicos, lo que lo hace popular entre investigadores y desarrolladores.[17]
- **TensorFlow:** Desarrollado por Google, TensorFlow es otro marco ampliamente utilizado que soporta tanto entornos de investigación como de producción. Ofrece un completo ecosistema de herramientas, bibliotecas y recursos comunitarios.
- **Keras:** Una biblioteca de redes neuronales fácil de usar que puede ejecutarse sobre TensorFlow o PyTorch. Keras simplifica el proceso de construcción y entrenamiento de modelos de aprendizaje profundo.
- **Ultralytics YOLO:** La primera vez que se utiliza "YOLO", Ultralytics YOLO los modelos son modelos de detección de objetos de última generación que aprovechan las arquitecturas CNN para lograr una gran precisión y velocidad. Estos modelos están disponibles a través del HUB Ultralytics , que proporciona herramientas para entrenar, desplegar y gestionar modelos de forma eficiente. [18]

IV-I. Deep Learning

El "Deep Learning" constituye una vertiente del aprendizaje automático que se apoya en redes neuronales artificiales (RNA) para representar abstracciones de alto nivel de los datos de entrada. Esto se logra mediante una estructura gráfica que involucra múltiples capas de procesamiento. Por ejemplo, en sistemas avanzados de asistencia al conductor (ADAS) basados en visión por computadora, los algoritmos de aprendizaje profundo mejoran la detección y el reconocimiento de diversos objetos, respaldan la clasificación de los mismos y

permiten la identificación y la predicción de acciones [19].

IV-J. Raspberry Pi

Es un microcontrolador de bajo precio, también se lo puede utilizar para programar una serie de computadoras de placa única (SBC). Fue desarrollada por la Raspberry Pi Foundation en el Reino Unido. Las mini computadoras o mini CPU's tienen como función ser accesibles, económicas y versátiles, lo que las hace una herramienta popular para la educación en general, la experimentación y la creación de proyectos de electrónica y computación entre otras más [20].



Figura 6. Raspberry Pi4. Fuente: por B. Aviles y J. González

IV-K. Facecam 1000x V2

La FaceCam 1000X V2 es una opción confiable para quienes buscan una cámara web de alta definición con funciones prácticas y un diseño adaptable a diferentes dispositivos. [21]



Figura 7. Cámara Facecam 1000x V2. Fuente: por B. Aviles y J. González

V. MARCO METODOLÓGICO

Se trabaja sobre un sistema operativo Linux, en el cual se desarrollara todo el código de programación que sera embebido en la tarjeta de programación del raspberry pi 4.

V-A. Dataset

La generación y etiquetado de imágenes para su entrenamiento correspondiente en los modelos de redes neuronales es la parte más dificultosa para el desarrollo en los algoritmos de visión asistida por computadora, ya que la precisión de las etiquetas y calidad de las imágenes generará mejores resultados en el modelo entrenado para reconocer objetos.

Para el funcionamiento del proyecto se ha utilizado un dataset de productos de estantes que generalmente se encuentran en un supermercado, por ello, se realizó una compilación de productos de diferentes marcas, categorías y tipos para así obtener un dataset variado y que el modelo pueda reconocer diferentes productos.

En la siguiente figura 8 se mostrará la información del Dataset. El código del desarrollo puede ser encontrado en el anexo *Código del Web Scraper*

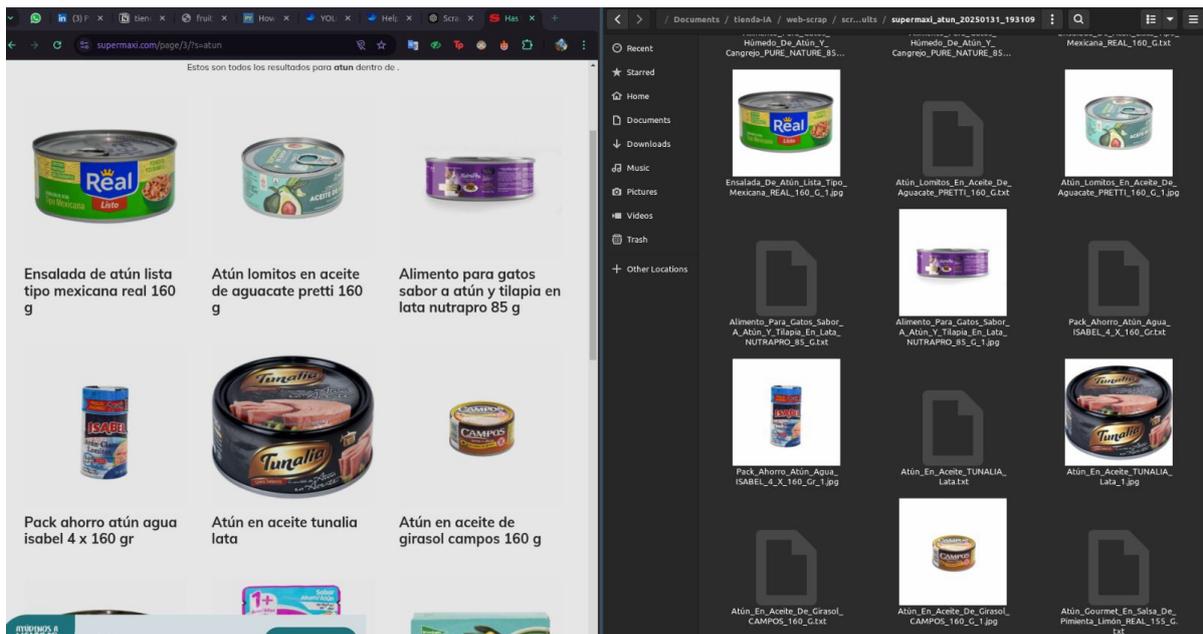


Figura 8. Dataset compilado de un supermercado Fuente: por B. Aviles y J. González.

V-B. Base de datos de productos

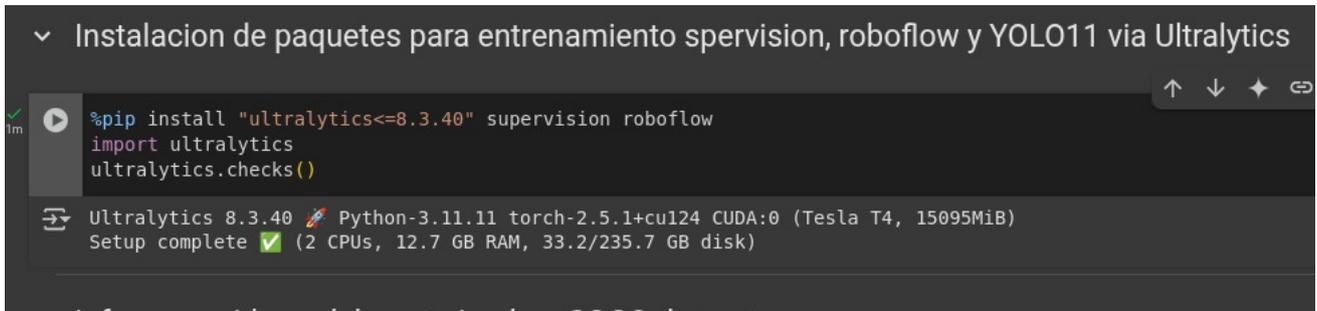
Se etiquetaron las imágenes para el dataset, desde una base de datos de los productos con diferente información como precio, tipo, marca y peso. A continuación, se mostrará un listado de los productos que serán etiquetado en la figura 9.

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	
a.Nombre	a.Marca	a.Categoria	a.Subcategoria	a.ID_Producto	a.SKUCODE	a.Codigo_Barras	a.Peso	a.Precio_Afiliado	a.Precio_Normal	
Atun Lomos En Aceite ILE 170 G	ILE	ALACENA	Atunes en Aceite	383527	807334	7861021250019	170 G	1.44	1.51	Delici
Ensalada De Atun Lista Tipo Mexicana REAL 160 G	REAL	ALACENA	Listos y Mixtos	383081	892639	7862100600619	160 G	1.21	1.27	Atún
Atun Lomitos En Aceite De Aguacate PRETTI 160 G	PRETTI	ALACENA	Atunes en Aceite	382106	1625432	7868001116532	160 G	1.94	2.04	Una ti
Pack Ahorro Atun Agua ISABEL 4 X 160 Gr	ISABEL	ALACENA	Atunes en Agua	381361	539607	8410111004521	160 G	5.6	5.88	Alime
Atun En Aceite TUNALIA Lata	TUNALIA	ALACENA	Atunes en Aceite	374061	1001610	7862101031504	N/A	1.49	1.56	Prepa
Atun En Aceite De Girasol CAMPOS 160 G	CAMPOS	ALACENA	Atunes en Aceite	365961	1385272	7861167301118	160 G	1.6	1.68	El Atú
Atun Gourmet En Salsa De Pimienta Limon REAL 155 G	REAL	ALACENA	Atunes en Aceite	364583	1559921	7862100605294	155 G	1.98	1.67	La nu
Atun Blanco En Trozos En Aceite De Oliva EL CORTE INGLES 111 G	EL CORTE INGLES	ALACENA	Atunes en Aceite	219543	1455070	8433329070690	111 G	N/A	N/A	Bonib
Atun Sanduchero ISABEL 4 X 80 G	ISABEL	ALACENA	Atunes en Aceite	209838	1638570	8410111011123	80 G	3	3.15	Disfrú
Atun Claro Lomitos En Aceite De Oliva ISABEL 3 X 140 g	ISABEL	ALACENA	Atunes en Aceite	209701	1642300	8410111011482	140 g	5.45	5.72	Alto c

Figura 9. Base de datos de productos generada compilado de un supermercado Fuente: por B. Aviles y J. González.

V-C. Instalación de paquetes

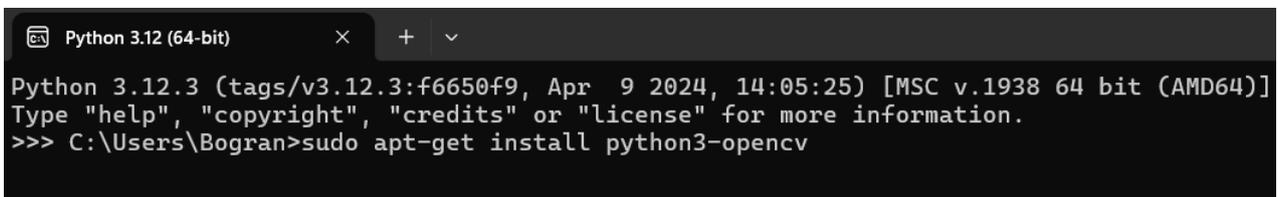
Para el entrenamiento y uso del modelo es necesario instalar en el sistema operativo Linux, una serie de librerías de python, el paquete de instalación para la librería se mostrado en la figura 10 para su posterior entrenamientos de datos.



```
▼ Instalacion de paquetes para entrenamiento spervision, roboflow y YOLO11 via Ultralytics
%pip install "ultralytics<=8.3.40" supervision roboflow
import ultralytics
ultralytics.checks()
Ultralytics 8.3.40 Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (Tesla T4, 15095MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 33.2/235.7 GB disk)
```

Figura 10. Instalación para las librerías de python Fuente: por B. Aviles y J. González

A continuación se mostrará el código para la instalación de OpenCV/Linux en la figura 11:



```
Python 3.12 (64-bit)
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> C:\Users\Bogran>sudo apt-get install python3-opencv
```

Figura 11. OpenCv/Linux. Fuente: por B. Aviles y J. González

V-D. Módulo del proyecto

El esquema del proyecto será mostrado en la figura 12 con sus respectivas conexiones entre el módulo de Raspberry Pi4, la cámara webcam y los resultados mostrados en pantalla.

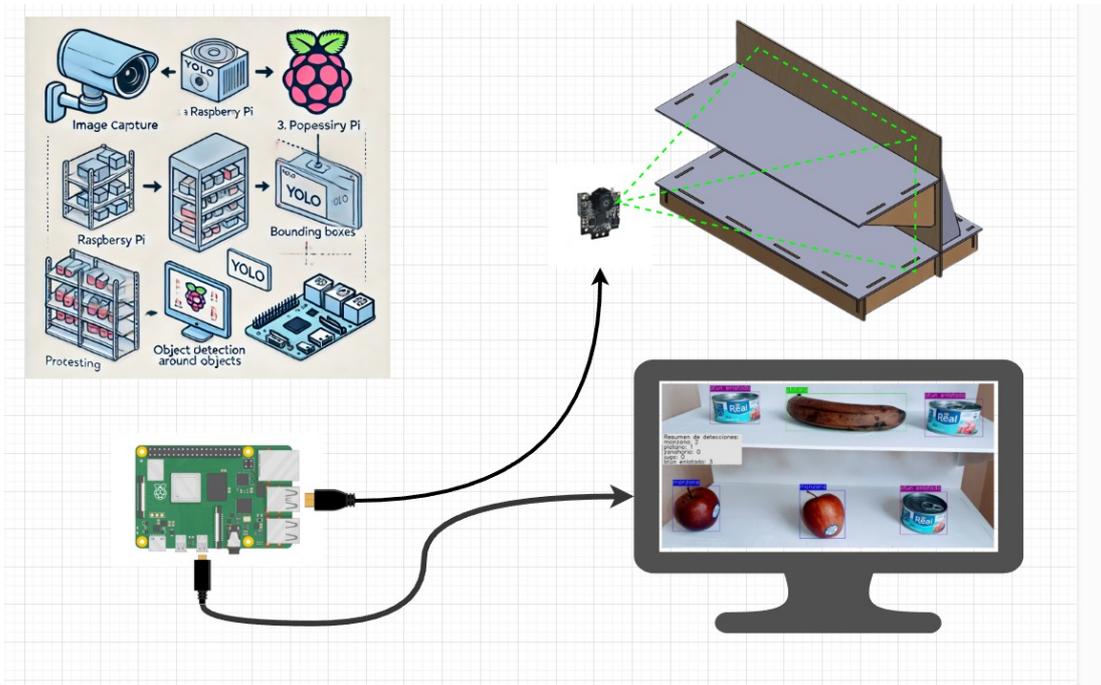


Figura 12. Esquema del proyecto. Fuente: por B. Aviles y J. González.

V-E. Elaboración del algoritmo

Para la visión artificial existen varios modelos de programación, para este proyecto se optó por YOLO, (You Only Look Once) debido a la eficiencia en tiempo real. Con esto se espera optimizar el modelo de visión artificial para limitar procesos largos de pruebas. Se mostrara una gráfica comparativa de modelos de YOLO en la siguiente figura 37.

El código puede ser encontrado en los anexos *Script de fine-tuning de modelo de aprendizaje*

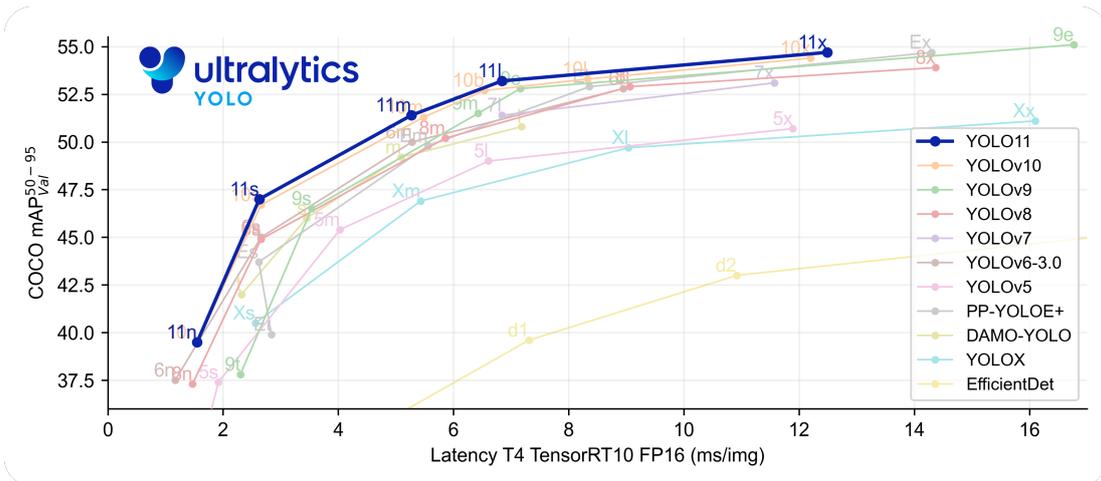


Figura 13. Comparación de modelos YOLO. Fuente: Ultralytics

V-F. Diseño de Percha

Por medio de un programa de modelado 3D, se procedió con el diseño de la percha a usarse en este proyecto. Se realizó mediante un mockup basado en el modelo similar a las que se utilizan en los supermercados del país.

En la figura 14 se observa el modelado 3D.

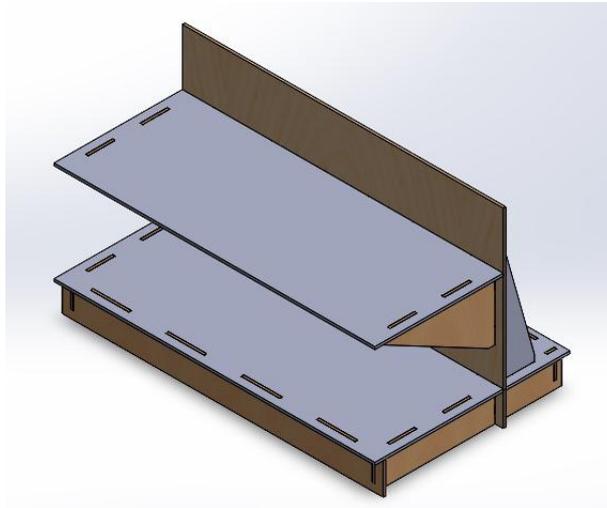


Figura 14. Percha diseñada en 3D. Fuente: por B. Aviles y J. González

V-G. Análisis estático

Para la percha se realizó un estudio de análisis estático para evaluar el diseño, capacidad y eficiencia que deba cumplir su función principal de exhibir productos en un supermercado. El análisis se centra en varios puntos para ser evaluados tales como:

- Material: qué tipo de material que se va usar, en este caso se optó por madera MDF para el armado de la misma, ya que es un material de bajo precio y resistente.
- Dimensiones: la percha cuenta con una altura de 30cm, una anchura de 50cm y la profundidad de 30.4cm.
- Diseño: para este punto se considera la capacidad de carga, el espacio que ocupara cada producto en la percha y los números de estantes, la percha cuenta con 2 estantes con la capacidad de una carga de 1kg en la parte superior y 2kg aproximadamente para la parte inferior.
- Productos: se exhibirán los productos a evaluarse como: manzana, banana, zanahoria, enlatado de atún y botella de refresco.

Para el estudio de análisis de cargas estáticas se usó un programa de análisis de elementos finitos, que permite configurar la fuerza que se quiere aplicar al producto. Como la percha se encuentra dividida en 2 estantes lo que implica que habrá dos tipos de cargas. Para encontrar la unidad de fuerza ejercida, se usa la siguiente fórmula:

$$F = m * g \quad (1)$$

Donde la F es la fuerza a encontrar, (m) la masa que se aplicara y (g) la gravedad, su unidad de medida estaría representada en Newtons (N).

En las siguientes figuras 15 y 16 se aplicara las fuerzas de tensión para su posterior estudio de análisis estático.

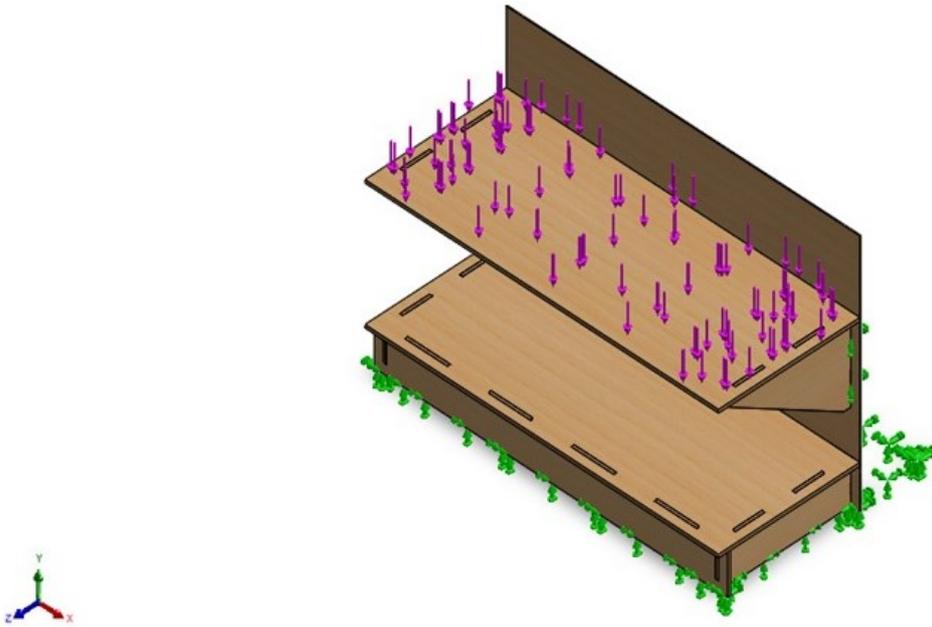


Figura 15. Lugar donde se aplicara la fuerza para su respectivo análisis. Fuente: por B. Aviles y J. González

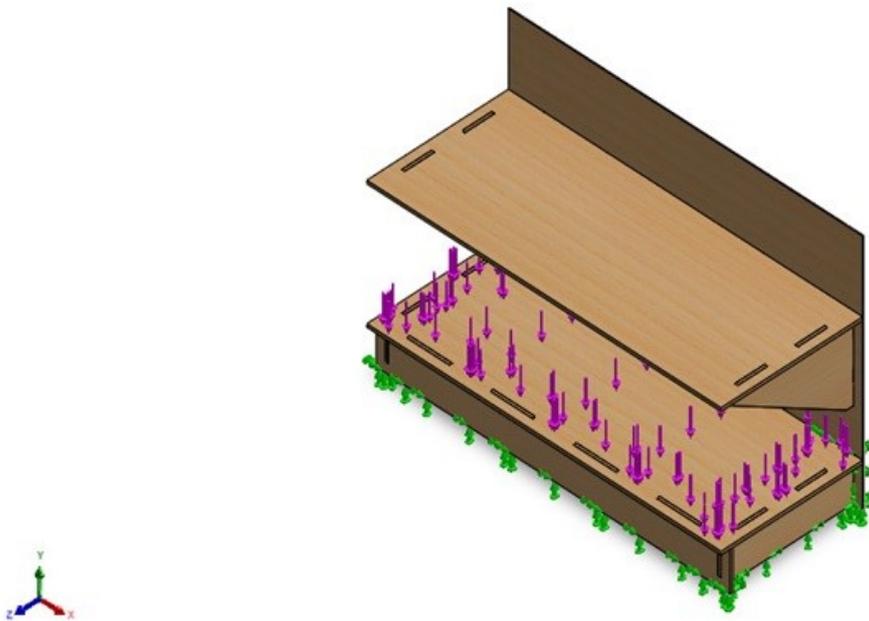


Figura 16. Análisis en la base para su respectivo estudio. Fuente: por B. Aviles y J. González

El primer análisis se realizó en el estante superior, los resultados arrojaron una fuerza ejercida de 9800 Newtons que equivale a una carga de 1kg aproximadamente, para observar el comportamiento de la percha con un peso en específico, se muestran en las siguientes figuras 17, 18 y 19.

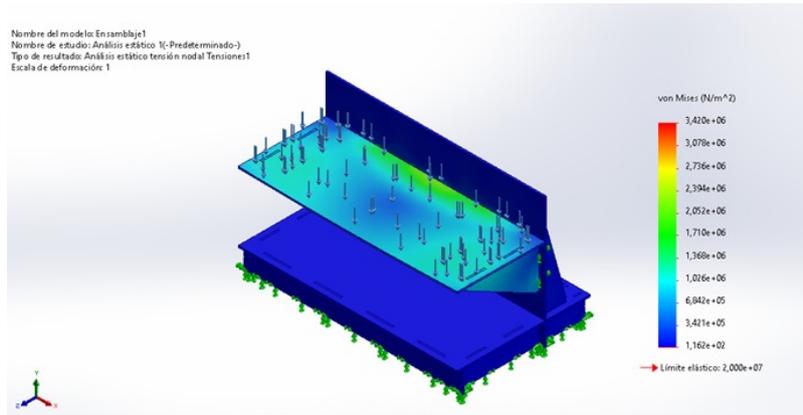


Figura 17. Tensión ejercida en estante superior. Fuente: por B. Aviles y J. González

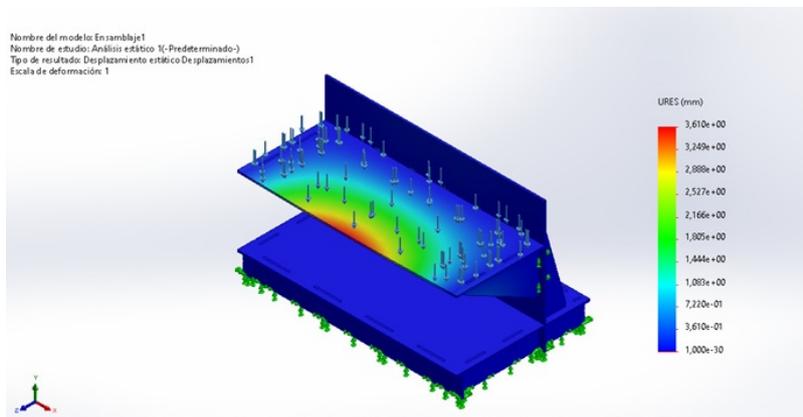


Figura 18. Desplazamiento ejercido en estante superior. Fuente: por B. Aviles y J. González

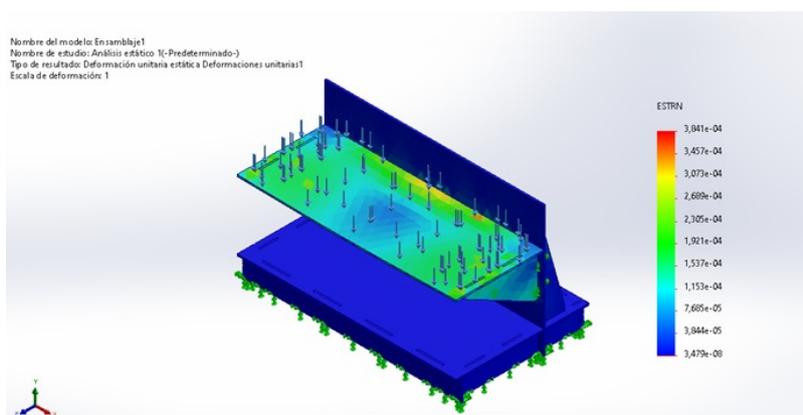


Figura 19. Deformación de percha por causa del desplazamiento. Fuente: por B. Aviles y J. González

El segundo análisis es una carga de 3kg, con una fuerza de 29400 Newtons, se puede observar el comportamiento de las fuerzas aplicadas en ese punto la cual mostró una deformación en la percha. Se puede observar la deformación de la percha en las siguientes figuras 20, 21 y 22.

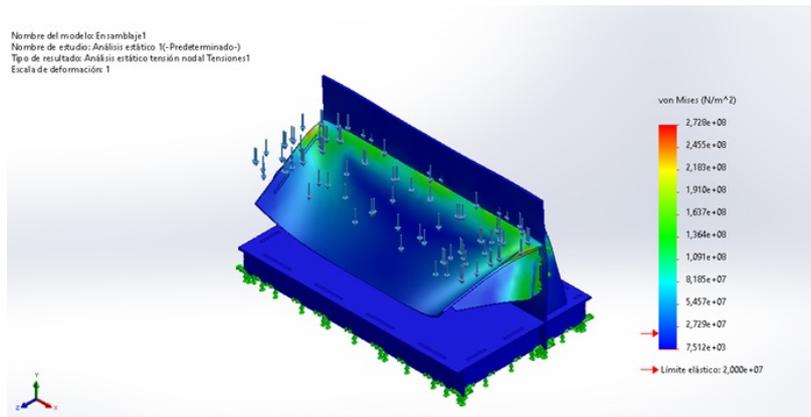


Figura 20. Tensión ejercida con una masa mayor para observar resultados. Fuente: por B. Aviles y J. González

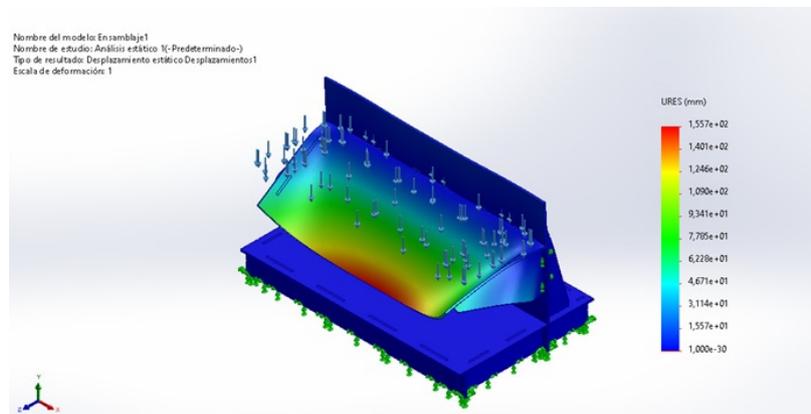


Figura 21. Desplazamiento del material. Fuente: por B. Aviles y J. González

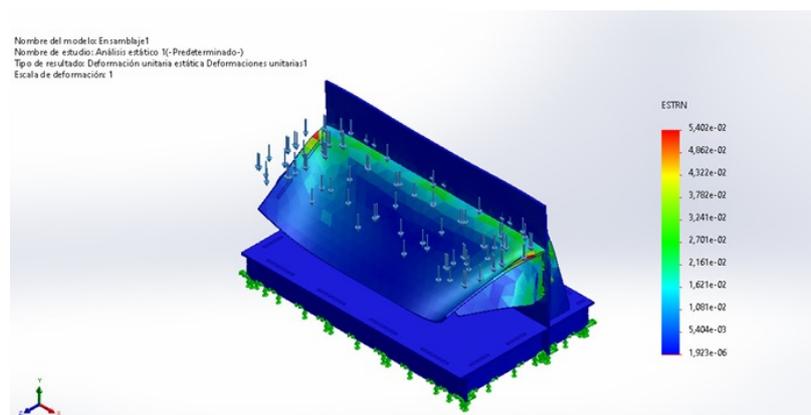


Figura 22. Deformación del estante. Fuente: por B. Aviles y J. González

El tercer estudio, es en el estante inferior, se aplicó un mayor peso en la masa la cual es de 2kg con una fuerza de 19600 Newtons. En las siguientes figuras 23, 24 y 25 se observa el estudio de análisis estático que se muestra en ese punto.

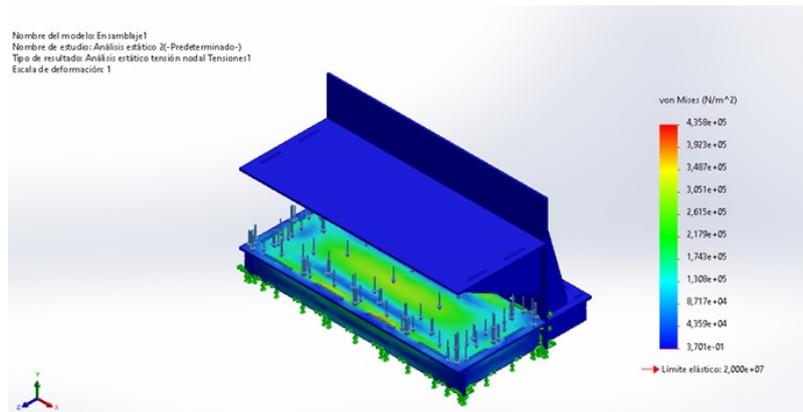


Figura 23. Tensión ejercida en el estante de base. Fuente: por B. Aviles y J. González

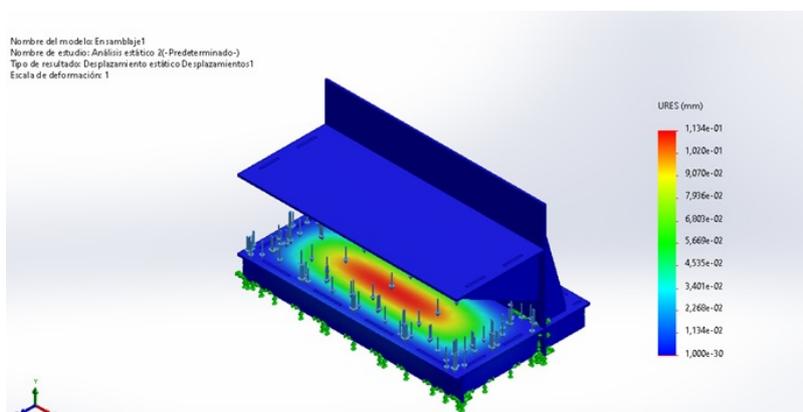


Figura 24. Desplazamiento ejercido en la base. Fuente: por B. Aviles y J. González

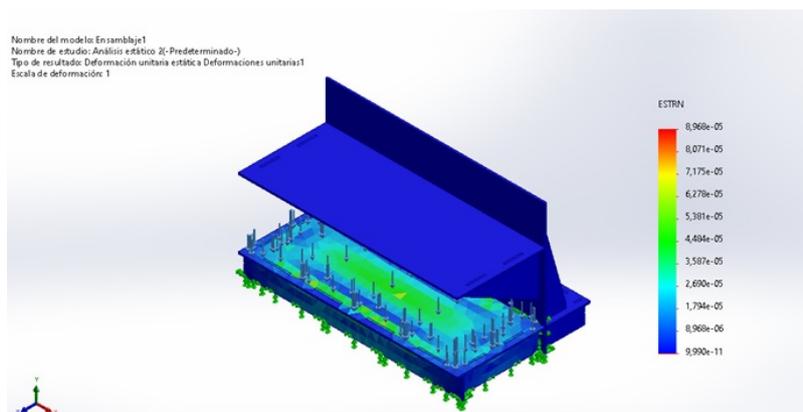


Figura 25. Deformación de percha por causa del desplazamiento. Fuente: por B. Aviles y J. González

Para el último análisis se aplicó una carga de 5kg con una fuerza ejercida de 49000 Newtons en la base, como se puede mostrar en las figuras el análisis la cual también demostró una deformación en la percha. En las figuras 26, 27 y 28 se observa el estudio aplicado.

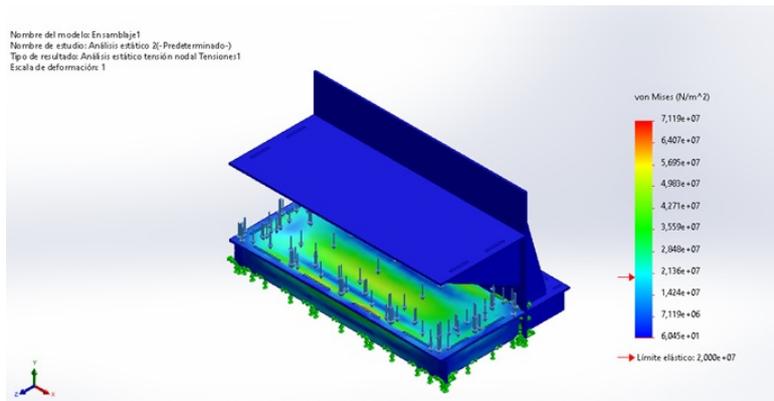


Figura 26. Tensión ejercida en el estante de base. Fuente: por B. Aviles y J. González

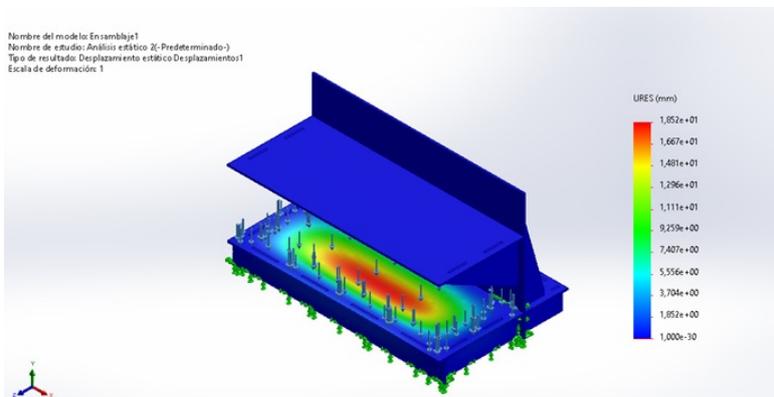


Figura 27. Desplazamiento ejercido en la base. Fuente: por B. Aviles y J. González

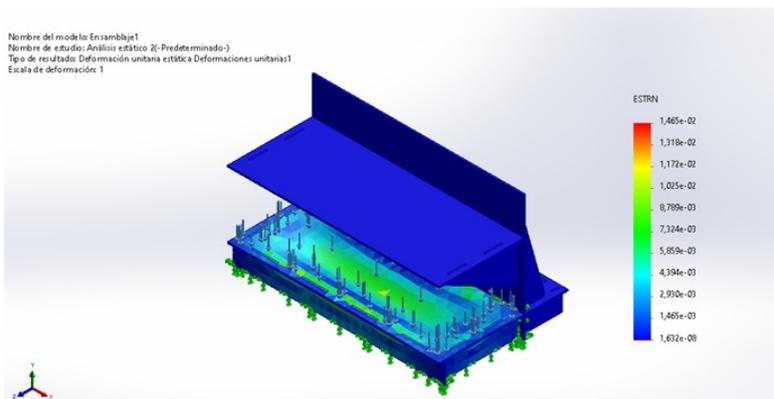


Figura 28. Deformación de percha por causa del desplazamiento. Fuente: por B. Aviles y J. González

Como conclusión este estudio de análisis estático mostró el comportamiento de las fuerzas ejercidas en la percha para tomar en cuenta el factor de seguridad y no sobrepasar el peso permitido en cada estante.

V-H. Diseño de Base para Cámara Webcam

Por medio de un programa de modelado 3D se procedió con el diseño de la base para sostener la cámara webcam para capturar las imágenes de la percha.

En la figura 29 se observa el modelo a utilizar para este proyecto.

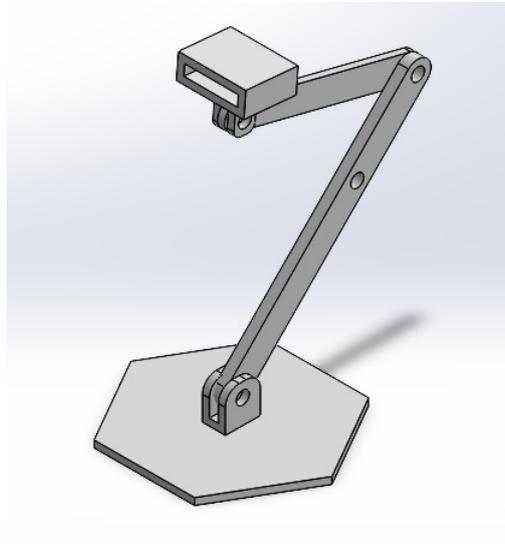


Figura 29. Base de cámara diseñada 3D. Fuente: por B. Aviles y J. González

V-I. Interfaz

Se procede a crear una interfaz capaz de cargar y funcionar con el algoritmo implementado. Por medio de una página web, se prueba el algoritmo en la visión artificial para que cumpla con las indicaciones de clasificar los objetos, se podrá evidenciar en la figura 30 la tabla de los productos registrados en la base de datos.

Productos Detectados			
ID	Nombre	Cantidad	Precio
1	Atun enlatado	3	\$2.00
2	Banana	1	\$0.75
3	Jugo	0	\$1.50
4	Manzana	2	\$1.00
5	Zanahoria	0	\$0.50

Detección de Objetos

Vídeo Procesado en Tiempo Real

Resumen de detecciones:
manzana: 2
plátano: 1
zanahoria: 0
jugo: 0
atun enlatado: 3

Ejecutar Detección

Figura 30. Interfaz del navegador web. Fuente: por B. Aviles y J. González

El sistema se basa en una arquitectura del servidor de clientes, en la que el cliente (navegador web) interactúa con el servidor de backend para recuperar datos y aplicaciones de procesos. Debe interactuar con el usuario para observar imágenes en vivo, catálogos de productos y resultados de percepción reales. Se comunica con PostgreSQL para acceder a los datos del producto y ejecutar scripts de identificación de productos con imágenes del algoritmo empleado en el proyecto (Yolo).

En el navegador web mostrara la pantalla de inicio en la cual detallara la información del proyecto, a continuación en la siguiente figura 31 se observa el inicio a la pantalla de carga.



Figura 31. Presentación interfaz. Fuente: por B. Aviles y J. González

Una vez que selecciona un producto del inventario o catálogo, puede hacer clic en Ejecutar Detección. Se actualiza el número de productos en el cuadro de reconocimiento de productos e inventario, sera evidenciada en la figura 32.



Figura 32. Interacción con interfaz. Fuente: por B. Aviles y J. González

- **HTML:** Proporciona la estructura básica de un sitio web. Se conoce HTML como herramienta principal para el uso de páginas web.
HTML: Un lenguaje de marcado el cual emplea su uso para etiquetar la información en páginas web, dando su función de etiquetado para definir distintos elementos tales como imágenes, enlaces y texto.

Se puede visualizar la base de datos de backend y el catálogo. A continuación, se puede observar en la figura 33 obtiene los resultados y actualiza la interfaz dando sus Funciones de backend.

Se procede a comunicarse con la base de datos PostgreSQL para obtener productos disponibles y mostrarlos en el borde, para también recibir consultas SQL y productos de base de datos.

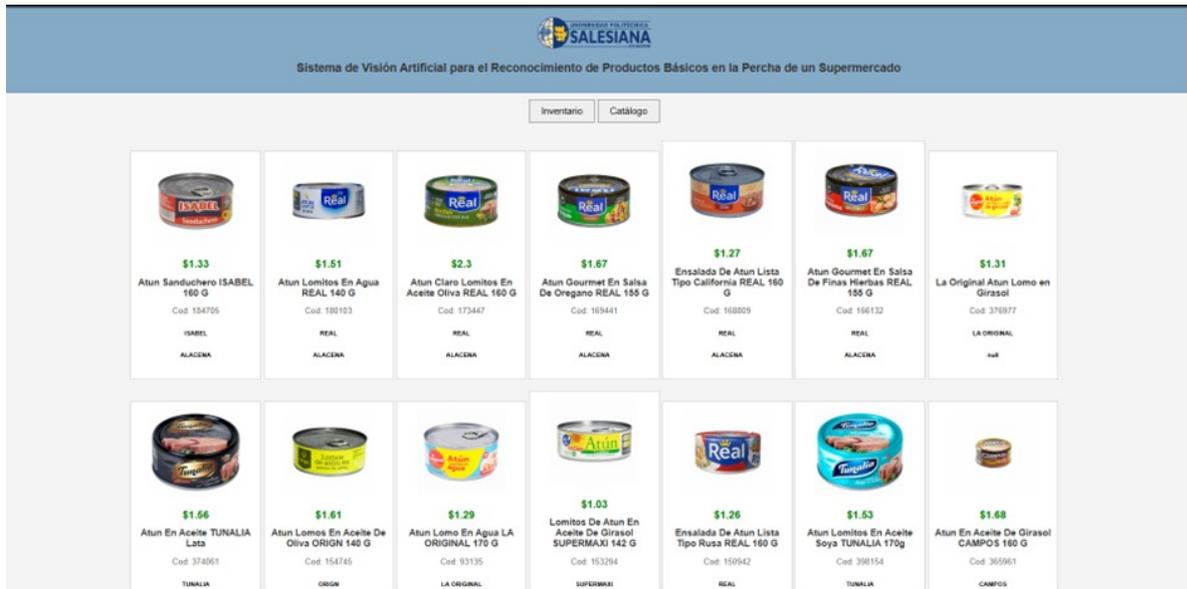


Figura 33. Dataset del Catálogo. Fuente: por B. Aviles y J. González

Como puntos finales de la interfaz tenemos que el backend explica la detección, Python procesa el video y genera imágenes con limitaciones de la frontera desde los archivos almacenados, el backend se mostrar de la figura 34.

	nombre	marca	categoria	subcategoria	id_producto	skucode	codigo_barras	peso	precio_afil
	text	text	text	text	[PK] integer	text	text	text	numeric
1	Lomitos De Atun En Aceite De Oliva SUPERMAXI 240 G	SUPERMAXI	ALACENA	Atunes en Aceite	194114	1413456	7861042564515	240 G	
2	3Pack Atunes En Aceite De Oliva + 1 Atun Gratis ISABEL 3 X 160 g	ISABEL	ALACENA	Atunes en Aceite	193556	1595725	8410111011512	160 g	
3	Atun Sandwichero ISABEL 160 G	ISABEL	ALACENA	Atunes en Aceite	184705	1634912	8410111009175	160 G	
4	Atun Desmechado. En Girasol LA ORIGINAL Lata	LA ORIGINAL	ALACENA	Pastas Cortas	200466	1515082	7861042571889	[null]	
5	Atun Lomitos Con Champinones VAN CAMPS 150 G	VAN CAMP'S	ALACENA	Listos y Mixtos	184254	210390	722008000461	150 G	
6	Atun Lomitos En Agua REAL 140 G	REAL	ALACENA	Atunes en Agua	180103	1445894	7862100604129	140 G	
7	Atun En Aceite y Jugo De Limon VAN CAMPS 240 G	VAN CAMP'S	ALACENA	Atunes en Aceite	177465	210393	722008000508	240 G	
8	Atun Claro Lomitos En Aceite Oliva REAL 160 G	REAL	ALACENA	Atunes en Aceite	173447	1259127	7862100603443	160 G	
9	Atun Gourmet En Salsa De Oregano REAL 155 G	REAL	ALACENA	Dulces de Leche	169441	1559925	7862100605287	155 G	
10	Ensalada De Atun Lista Tipo California REAL 160 G	REAL	ALACENA	Panela	168809	892640	7862100600602	160 G	
11	Atun Lomitos En Aceite De Girasol Con Abre Facil VAN CAMPS 8 X 184 G	VAN CAMP'S	ALACENA	Instantaneos	168011	1399156	722008001420	184 G	
12	Atun Lomitos Con Vegetales VAN CAMPS 150 G	VAN CAMP'S	ALACENA	Granolas	167120	210388	722008000416	150 G	
13	Encobollado De Atun Listo REAL 400G	REAL	ALACENA	Pastas Largas	166402	1195175	7862100603351	400G	
14	Atun Gourmet En Salsa De Finas Hierbas REAL 155 G	REAL	ALACENA	Granolas	166132	1561871	7862100605300	155 G	
15	Atun En Aceite Oliva VAN CAMPS 480 G	[null]	ALACENA	Maiconas	165637	210392	722008000492	480 G	
16	Pack De Atun ISABEL 640 G	ISABEL	ALACENA	Pastas Largas	165016	541587	8426920010882	640 G	
17	Atun Lomitos En Agua Light Con Abre Facil VAN CAMPS 184 G	VAN CAMP'S	ALACENA	Atunes en Agua	111192	210377	722008000102	184 G	
18	Atun Lomitos De Atun En Agua Precio Especial VAN CAMPS 6 X 80 G	VAN CAMP'S	ALACENA	Anvejas	161057	1399163	722008001413	80 G	
19	La Original Atun Lomo en Girasol	LA ORIGINAL	[null]	[null]	376977	None	7861042560630	[null]	
20	Atun En Aceite TUNALIA Lata	TUNALIA	ALACENA	Atunes en Aceite	374061	1001610	7862101031504	[null]	
21	Atun Lomos En Aceite De Oliva ORIGN 140 G	ORIGIN	ALACENA	Atunes en Aceite	154745	1647702	7861042583738	140 G	

Figura 34. Datos del Backend. Fuente: por B. Aviles y J. González

V-J. Set up cámara

Una vez obtenido los datos del algoritmo e interfaz para la Visión Artificial, se sugiere configurar la webcam para la ejecución del algoritmo para el reconocimiento de los productos especificados. La cámara abarcará toda el área de la percha obteniendo visión de todos los productos. Para que la cámara funcione correctamente, el área de trabajo debe estar con una iluminación adecuada para así evitar ruido de imagen, errores en la detección, y poder obtener resultados positivos. En la siguiente figura 35 se mostrara una captura de imagen de como el algoritmo reconoce los productos.

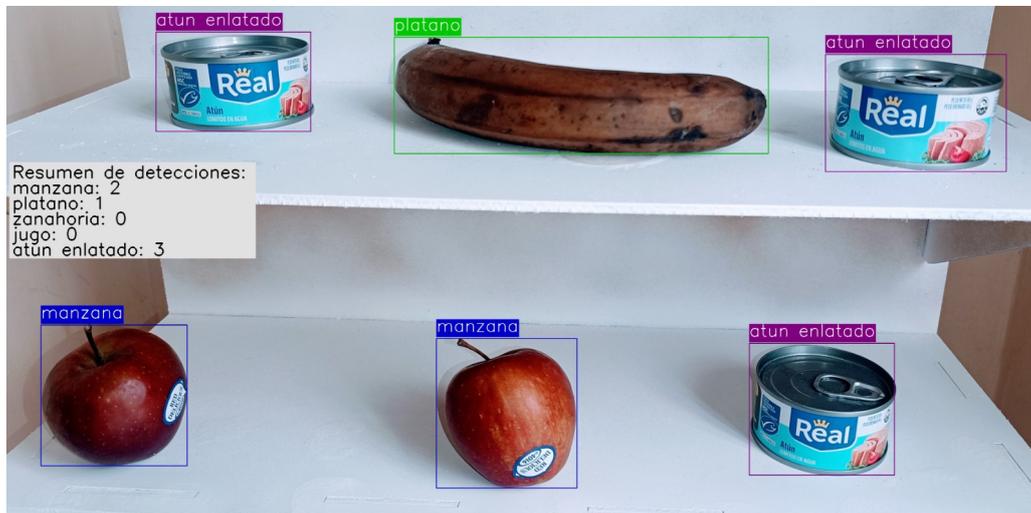


Figura 35. Set up cámara. Fuente: por B. Aviles y J. González

V-K. Etiquetado de imágenes

Se está recopilando y anotando imágenes cuidadosamente, teniendo en cuenta múltiples consideraciones/problemas como:

- Ruido: Variaciones irrelevantes o aleatorias de los datos.
- Incoherencia: Variaciones en los tamaños, formatos y calidad de las imágenes.
- Desequilibrio: Distribución desigual de clases o categorías en el conjunto de datos.

Entonces, se conoce la importancia del procesamiento de datos es muy alta, ya que consiste en convertir los datos en un formato adecuado para el entrenamiento que reduzca la carga computacional y ayude a mejorar el rendimiento del modelo, al evidenciarse en la figura 36.



Figura 36. Etiquetado manual a los productos. Fuente: por B. Aviles y J. González

Las imágenes obtenidas de fuentes de datos abiertos en la web fueron reorganizadas y clasificadas para realizar un entrenamiento adecuado. En la siguiente figura 37 se observa el esquema del dataset entrenado.

V-L. Lista de productos a usar

Para el proyecto se pensó qué productos básicos se podrían usar, se sabe que son productos básicos de un supermercado, por lo cual para una buena demostración se optó por dos frutas (manzana y banana), un vegetal (zanahoria), conservante (atún en lata) y una bebida (jugo artificial en botella).

309 Total Images

Train 279 Valid 17 Test 13

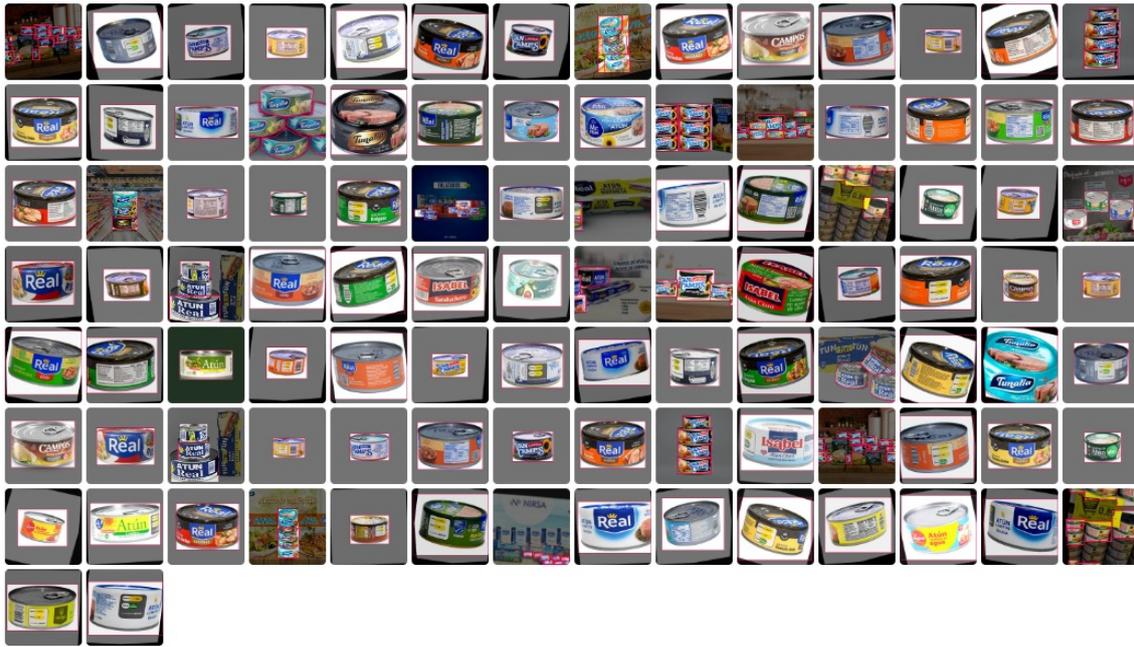


Figura 37. Dataset propio. Fuente: por B. Aviles y J. González

V-M. Análisis final del Entrenamiento

El presente proyecto se centró en el entrenamiento de un modelo para detectar objetos basados en la estructura de YOLO. Para ello, se emplearon subconjuntos del conjunto de datos COCO reducidos a cuatro clases principales (bottle, banana, apple, carrot, y adicionalmente canned tuna), prescindiendo explícitamente de escenas de fondo (background). A continuación, se exponen los elementos más sobresalientes del proceso, integrando tanto las métricas de rendimiento como la distribución de agrupaciones de datos en los entrenamientos y en las validaciones.

Tabla I
DISTRIBUCIÓN DE CLASES EN *train2017* Y *val2017*

Clase	train2017	val2017
bottle (0)	1422	1013
banana (1)	545	370
apple (2)	382	236
carrot (3)	540	365
canned tuna (4)	653	38

La clase *bottle* presenta la mayor cantidad de ejemplos en ambas particiones. El subconjunto *val2017* incluye un número limitado de instancias en la clase *canned tuna*, lo cual reduce la representatividad de dicha categoría durante la validación.

V-N. Métricas de Rendimiento

El modelo se entrena con la configuración estándar de YOLO y se evalúa mediante las métricas de Recall (R), Precisión (P), y en la *mean Average Precision* (mAP) al 50% de superposición (*mAP50*) y mAP en el rango 0.5–0.95 (*mAP50-95*). El conjunto de validación arroja los datos de resultados establecidos en la siguiente Tabla II. En la figura 38 estara detallado las métricas para el rendimiento del correcto funcionamiento y no tener futuros errores.

```

Entrenamiento completado en 0.263 horas

Validating /content/drive/MyDrive/dataset_pro/runs/yolov11s_train/weights/best.pt...
Ultralytics 8.3.40 Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (NVIDIA A100-SXM4-40GB, RAM 84GB)
YOLO11s summary (fused): 238 layers, 9,414,735 parameters, 0 gradients, 21.3 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	595	2022	0.43	0.331	0.319	0.241
bottle	379	1013	0.581	0.403	0.418	0.264
banana	103	370	0.354	0.23	0.186	0.0984
apple	76	236	0.285	0.114	0.0978	0.0614
carrot	81	365	0.346	0.222	0.213	0.135
canned tuna	17	38	0.582	0.684	0.682	0.647

```

Speed: 0.1ms preprocess, 0.6ms inference, 0.0ms loss, 1.4ms postprocess per image

```

Figura 38. Métricas de rendimiento. Fuente: por B. Aviles y J. González

Tabla II
MÉTRICAS DE VALIDACIÓN

Clase	P	mAP50	mAP50-95
all	0.43	0.319	0.241
bottle	0.58	0.418	0.264
banana	0.35	0.186	0.0984
apple	0.28	0.0978	0.0614
carrot	0.35	0.213	0.135
canned tuna	0.58	0.682	0.647

La clase *canned tuna* presenta un alto valor de mAP, incluso con pocas instancias en la fase de validación. La clase *bottle* alcanza valores moderados, mientras que las clases *banana* y *apple* registran mAP bajos.

- **Precisión (P = 0.892) y Recall (R = 0.996).** El modelo logra detectar casi todos los objetos que hay en las imágenes (elevado Recall), lo que implica muy pocos falsos negativos. Sin embargo, la Precisión ligeramente inferior (alrededor del 89%) indica que todavía existen algunos falsos positivos (detecciones que no corresponden a objetos reales).
- **mAP@0.5 = 0.98.** Esta métrica, calculada un umbral del IoU de un 0.5, muestra que el modelo delimita correctamente los objetos en la mayor parte de los casos (con al menos un 50% de superposición entre la predicción y la etiqueta real).
- **Tamaño del conjunto de validación.** Solo se cuenta con 20.000 imágenes. Aunque los resultados son notables, se corre el riesgo de sobre ajuste si la muestra no es suficientemente representativa. Se recomienda validar el modelo con un conjunto mayor o con casos de la vida real.

En síntesis, el modelo exhibe excelentes resultados en la localización de objetos y detecciones de los mismos, si bien es aconsejable comprobar su rendimiento en un entorno más amplio para confirmar la generalización de sus capacidades.



Figura 39. Imágenes de resultados de entrenamiento . Fuente: por B. Aviles y J. González



Figura 40. Resultados del entrenamiento. Fuente: por B. Aviles y J. González

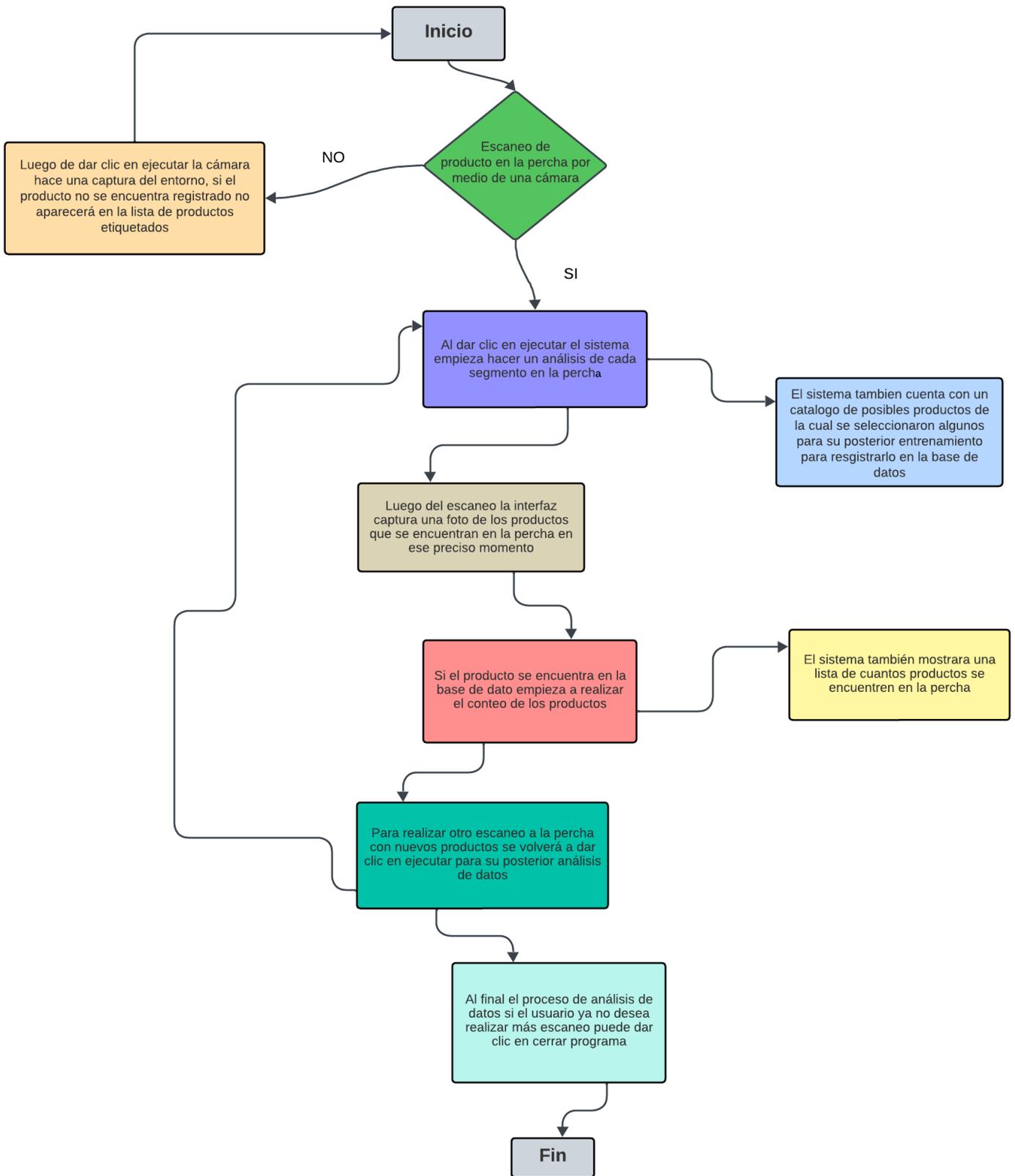


Figura 41. Diagrama de flujo. Fuente: por B. Aviles y J. González

VI. RESULTADOS

VI-A. Resultados para entrenamiento

A continuación se mostrarán los resultados del entrenamiento después del análisis final. Las visualizaciones del proceso de entrenamiento incluyen:

- Matriz de confusión normalizada: muestra menor coincidencia en categorías con menos instancias, como *banana* y *apple*.

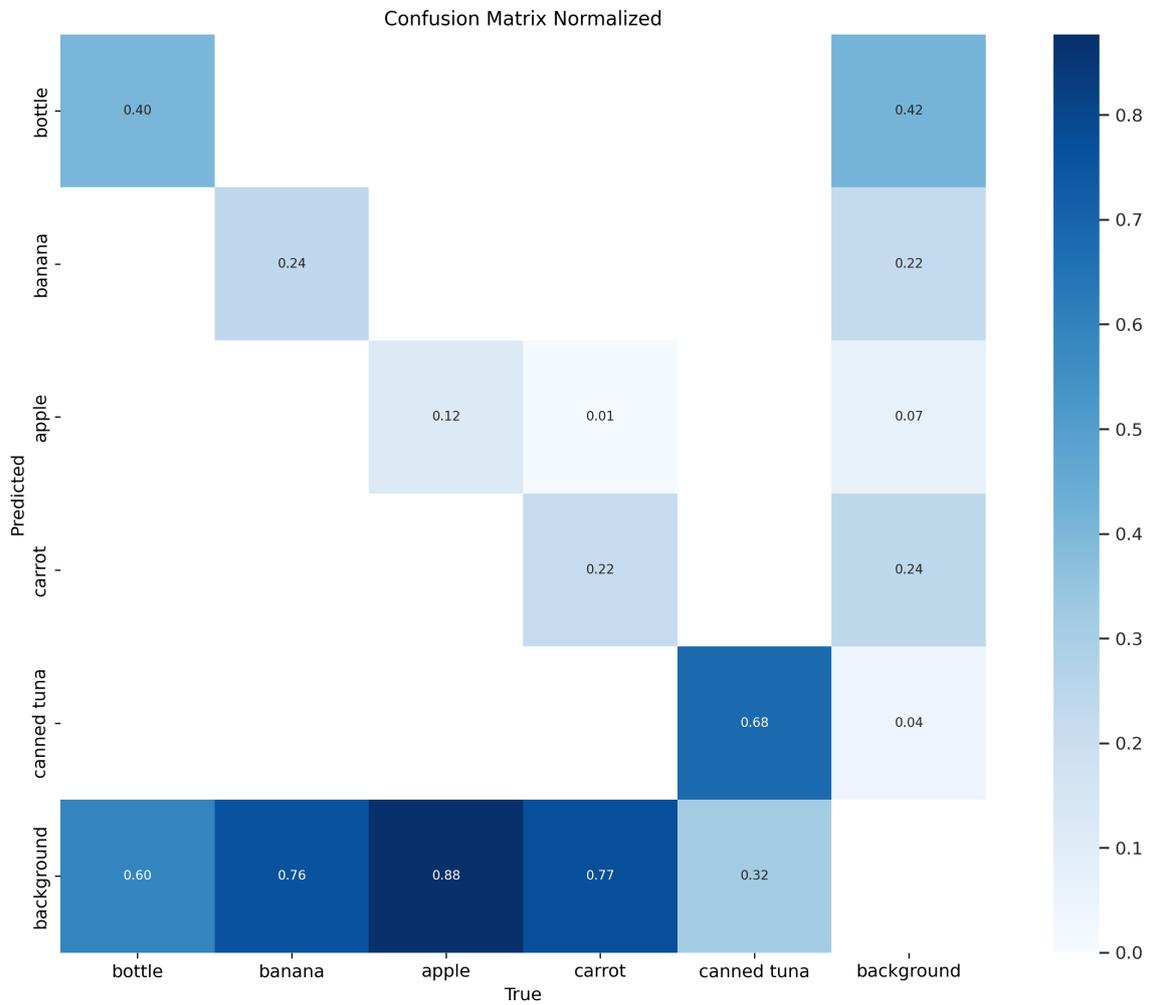


Figura 42. Matriz de confusión. Fuente: por B. Aviles y J. González

- Histogramas y dispersión de dimensiones de *bounding boxes*: revelan el predominio de *bottle* y el limitado número de muestras en otras clases.

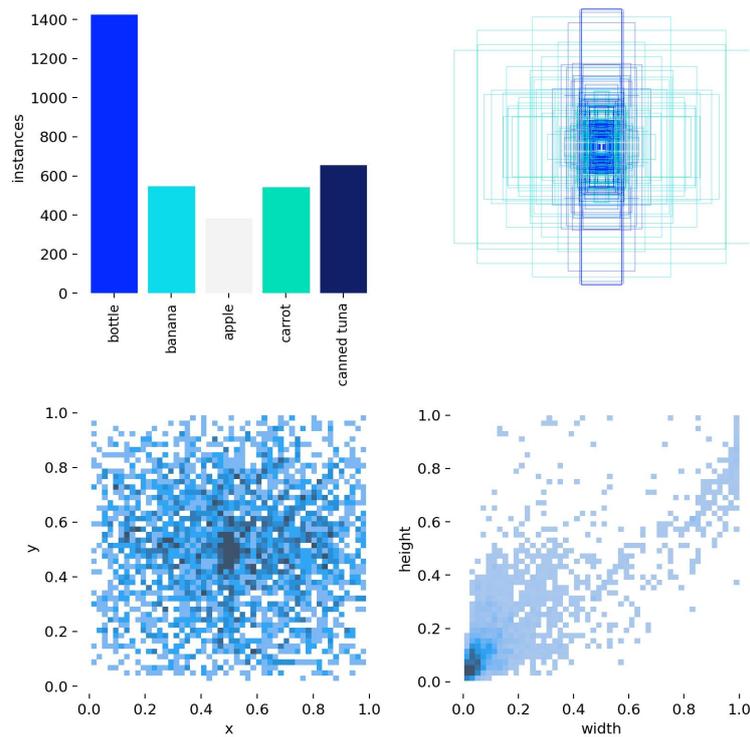


Figura 43. Labels. Fuente: por B. Aviles y J. González

- Curvas de entrenamiento: la función de pérdida converge y en las métricas *precision* y *recall* estabilizan los datos para obtener valores coherentes con la tabla de resultados.

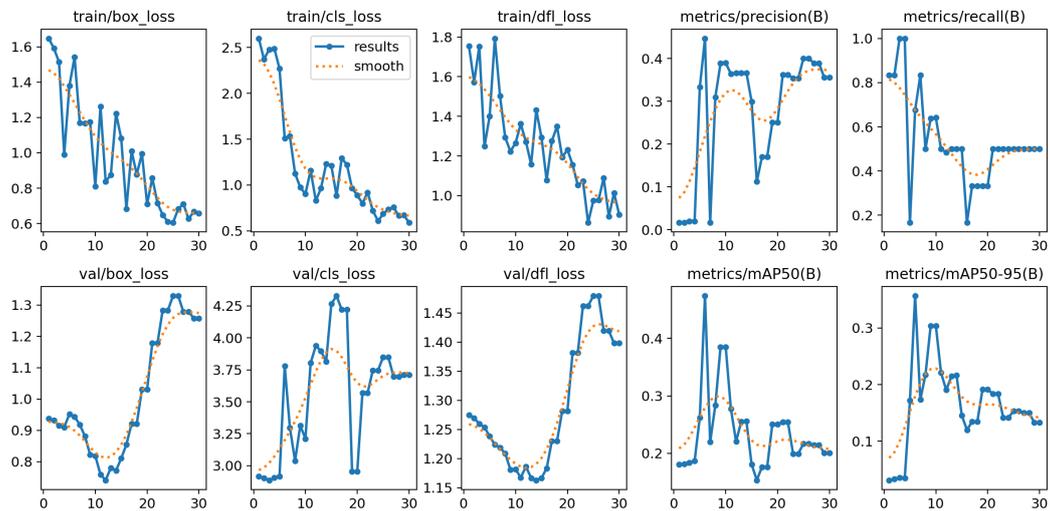


Figura 44. Curvas de entrenamiento. Fuente: por B. Aviles y J. González

VI-B. Resultados de la interfaz

Como resultado final tenemos la imagen 45, mostrando la interfaz en pleno funcionamiento a las especificaciones requeridas.

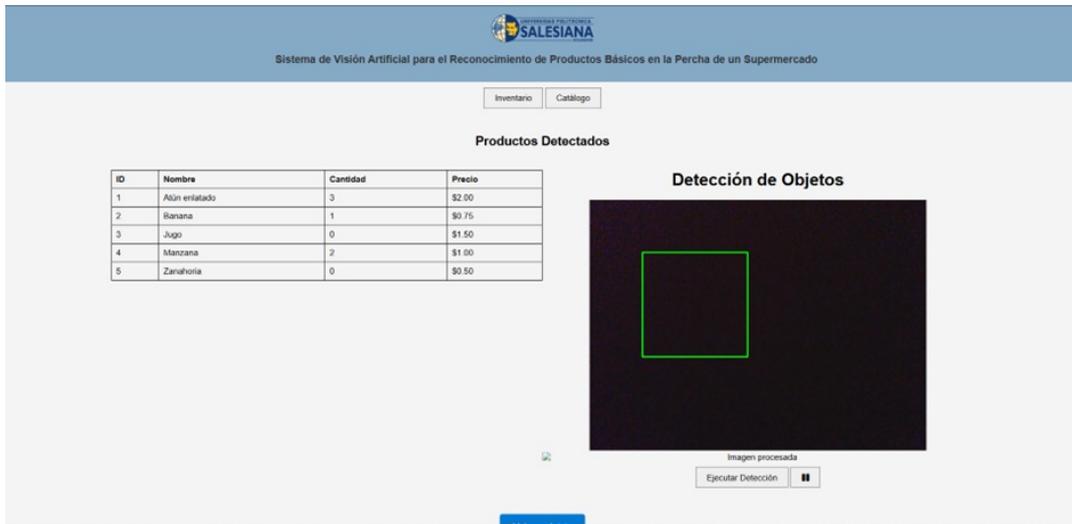


Figura 45. Interfaz. Fuente: por B. Aviles y J. González

VI-C. Resultados con el proyecto terminado

Después de realizar las pruebas mencionadas, se procedió a interactuar con el algoritmo de visión artificial ya implementado en la RPI.

Para iniciar el algoritmo se procede a abrir el terminal de la RPI para ingresar los códigos para poner en marcha el algoritmo.

Estos resultados se validan en la figura 46

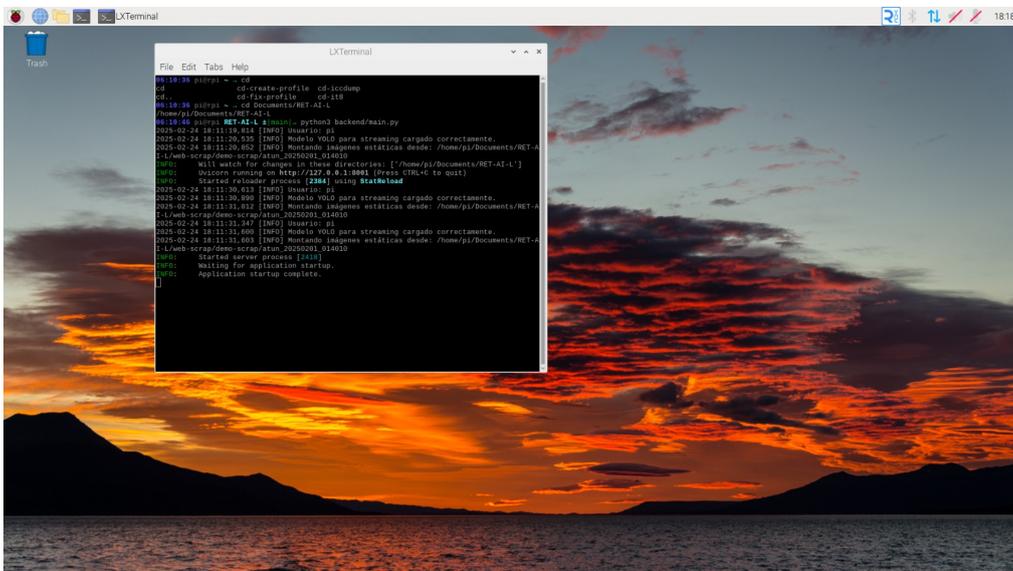


Figura 46. Iniciar algoritmo. Fuente: por B. Aviles y J. González

Con el algoritmo iniciado se procedió con las respectivas pruebas y se obtuvo lo siguiente:

Se colocó la cámara con un enfoque centrado en la percha; los resultados se detallan en la figura 47

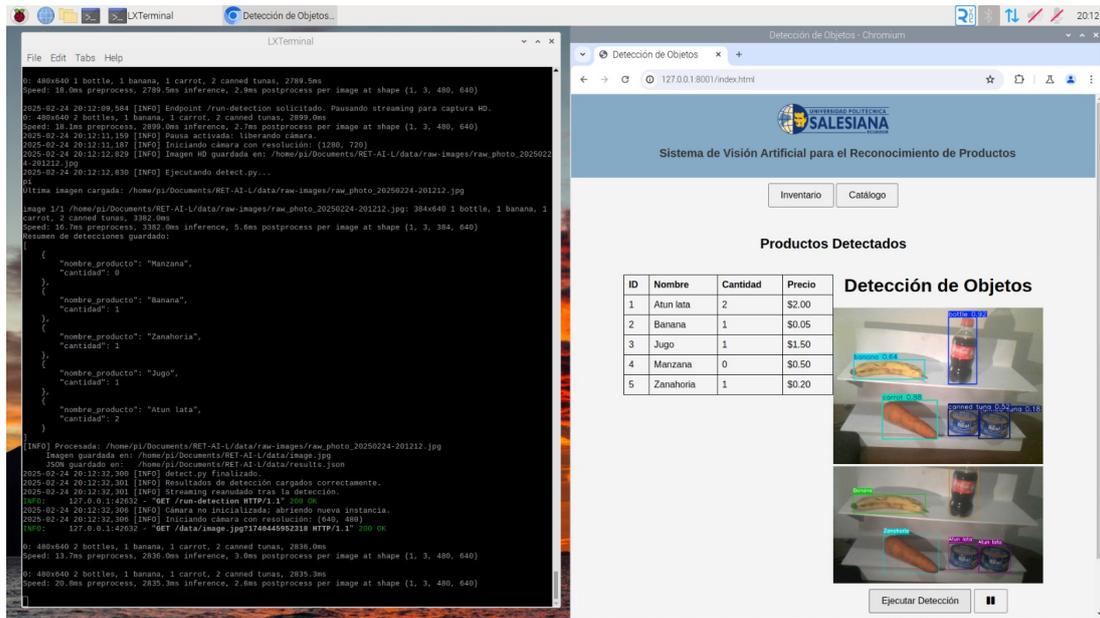


Figura 47. Algoritmo en funcionamiento. Fuente: por B. Aviles y J. González

En la figura 47 se observa cómo detecta los productos, los clasifica y al final muestra una tabla con los datos de la lectura de imagen.

Con los resultados adquiridos se procedió a realizar pruebas para validar el porcentaje de errores con respecto a la iluminación. Mediante un gráfico de barras se obtuvo los siguiente datos, mostrado en la siguiente figura 48 Como se puede observar, se logró una tasa de error de hasta el 40 %, teniendo parámetros tales como:

- **Baja Iluminación:** Con las pruebas realizadas, se obtuvo un porcentaje del 35 % ya que sin la iluminación adecuada, la cámara detecta productos con errores, a veces confundía un producto con otro y no hacía el conteo adecuado.
- **Iluminación media:** Mejoró la detección de objetos pero siguió dando falsos positivos en productos que no son con respecto al conteo de objetos.
- **Alta Iluminación:** Se sabe que el algoritmo detecta los objetos si estos pasan de un 50 %, con la iluminación adecuada se pudo detectar varios productos con un porcentaje de error mínimo. Mejoró bastante el ruido que había.
- **Luz Natural:** Se consideró que la luz natural ayudó en su totalidad a que la cámara reconozca de una manera precisa los productos mostrados en percha. La luz natural ayudó a eliminar sombras, eliminar ruidos y disminuyó falsos positivos.
- **Iluminación Regular:** Se procedió a realizar varias pruebas en este campo ya que como iluminación irregular se tuvo que cambiar y combinar de varias formas la iluminación que acompaña a la detección de objetos. A veces de cinco productos diferentes, detectaba solo dos, a veces uno, otras detectaba un producto pero con una etiqueta que no es.

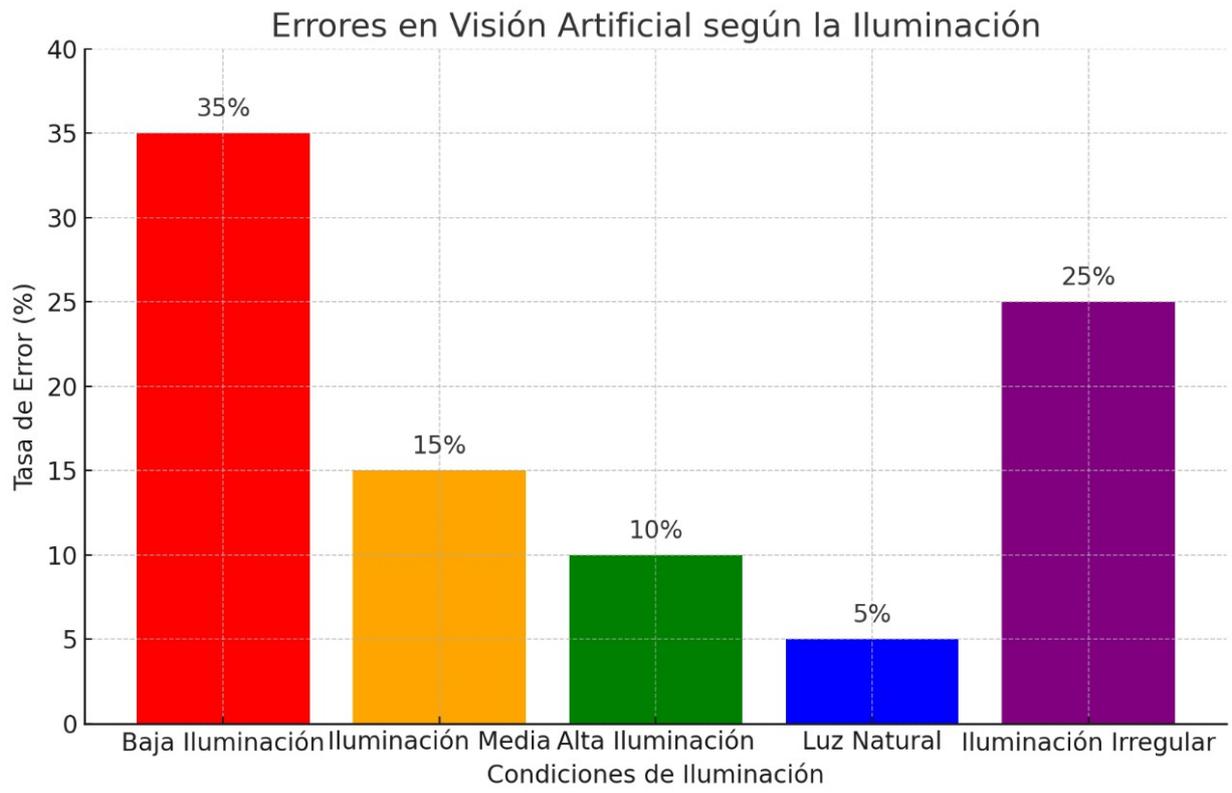


Figura 48. Gráfico del barra. Fuente: por B. Aviles y J. González

VI-D. Verificación de resultados finales

En la figura 49 se observa el funcionamiento del proyecto con una iluminación baja.

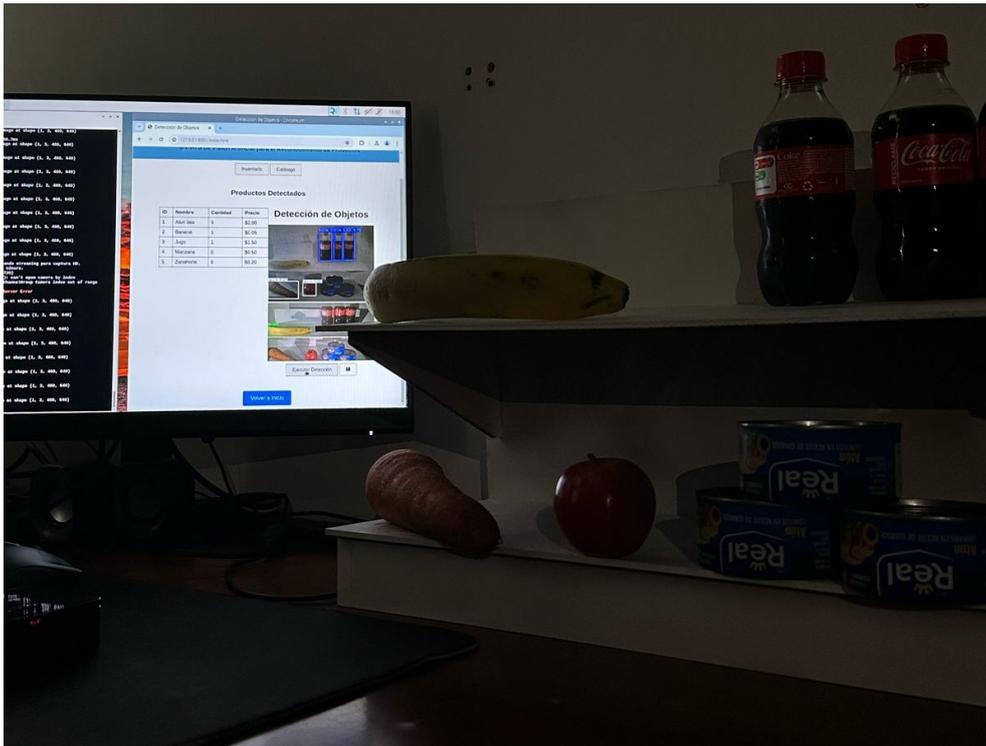


Figura 49. Baja Iluminación. Fuente: por B. Aviles y J. González

La figura 50 muestra una captura de pantalla desde la RPI donde se puede validar el funcionamiento en tiempo real con una iluminación baja.

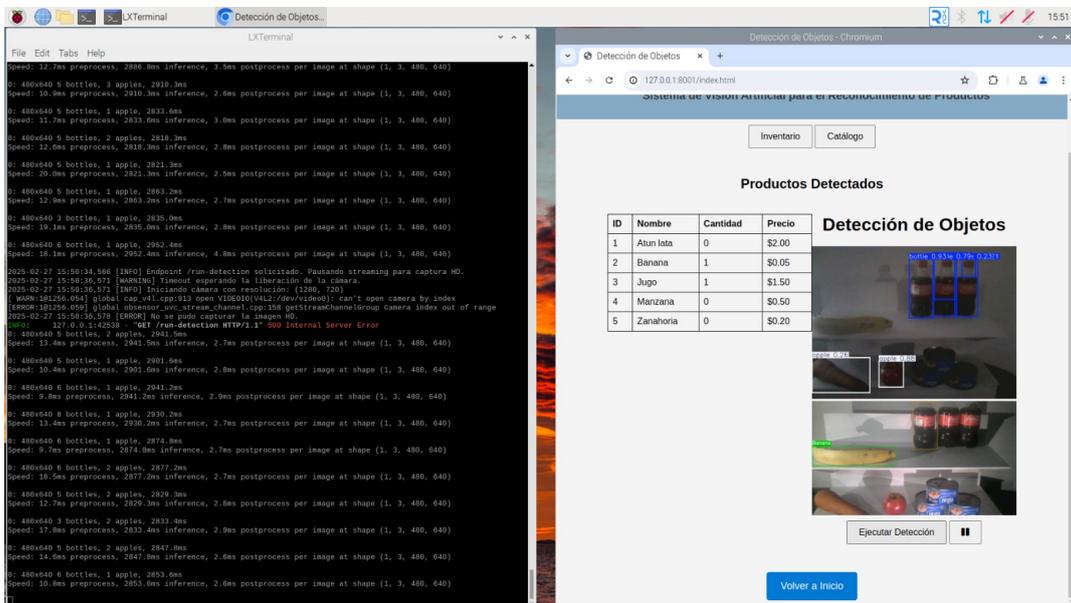


Figura 50. Captura del algoritmo con baja iluminación. Fuente: por B. Aviles y J. González

En la figura 51 se observa el funcionamiento del proyecto con una iluminación media.



Figura 51. Iluminación media. Fuente: por B. Aviles y J. González

La figura 52 muestra una captura de pantalla desde la RPI donde se puede validar el funcionamiento en tiempo real con una iluminación media.

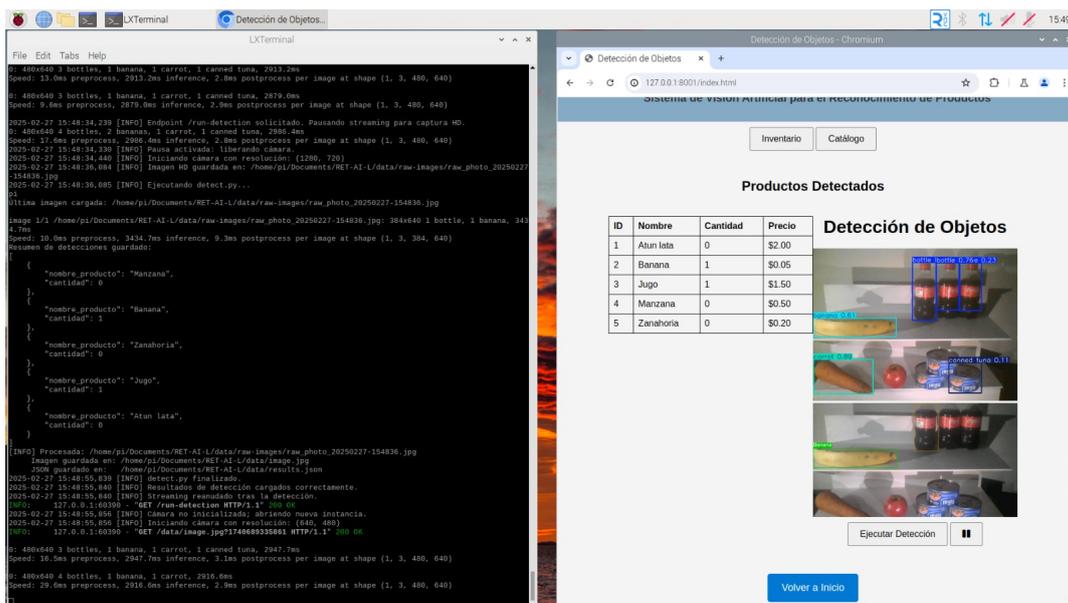


Figura 52. Captura del algoritmo con iluminación media. Fuente: por B. Aviles y J. González

En la figura 53 se observa el funcionamiento del proyecto con una iluminación alta.

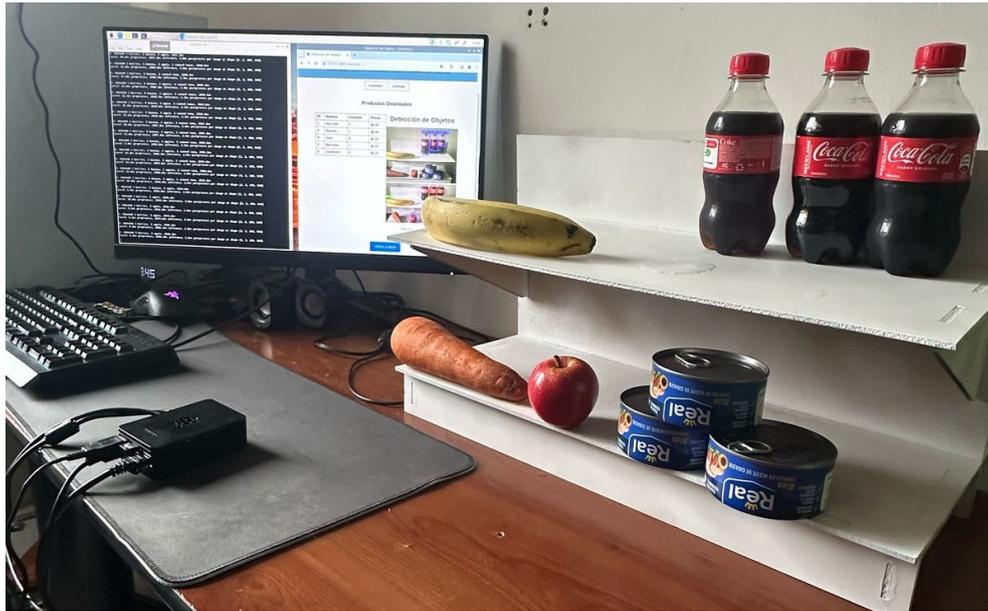


Figura 53. Alta Iluminación. Fuente: por B. Aviles y J. González

La figura 54 muestra una captura de pantalla desde la RPI donde se puede validar el funcionamiento en tiempo real con una iluminación alta.

ID	Nombre	Cantidad	Precio
1	Atun lata	2	\$2.00
2	Banana	1	\$0.05
3	Jugo	3	\$1.50
4	Manzana	1	\$0.50
5	Zanahoria	0	\$0.20

Figura 54. Captura del algoritmo con alta iluminación. Fuente: por B. Aviles y J. González

En la figura 55 se observa el funcionamiento del proyecto con luz natural.

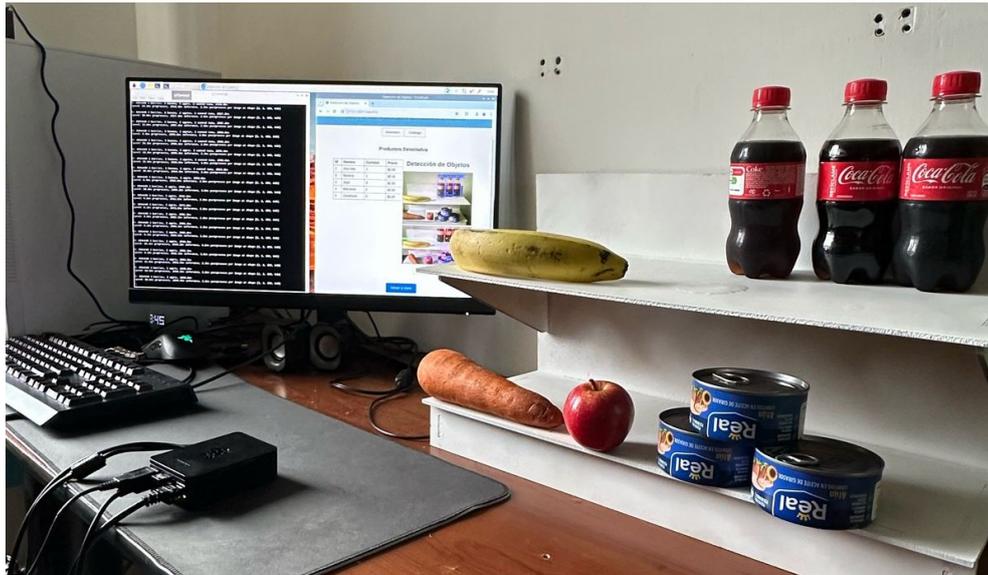


Figura 55. Luz natural. Fuente: por B. Aviles y J. González

La figura 56 muestra una captura de pantalla desde la RPI donde se puede validar el funcionamiento en tiempo real con luz natural.

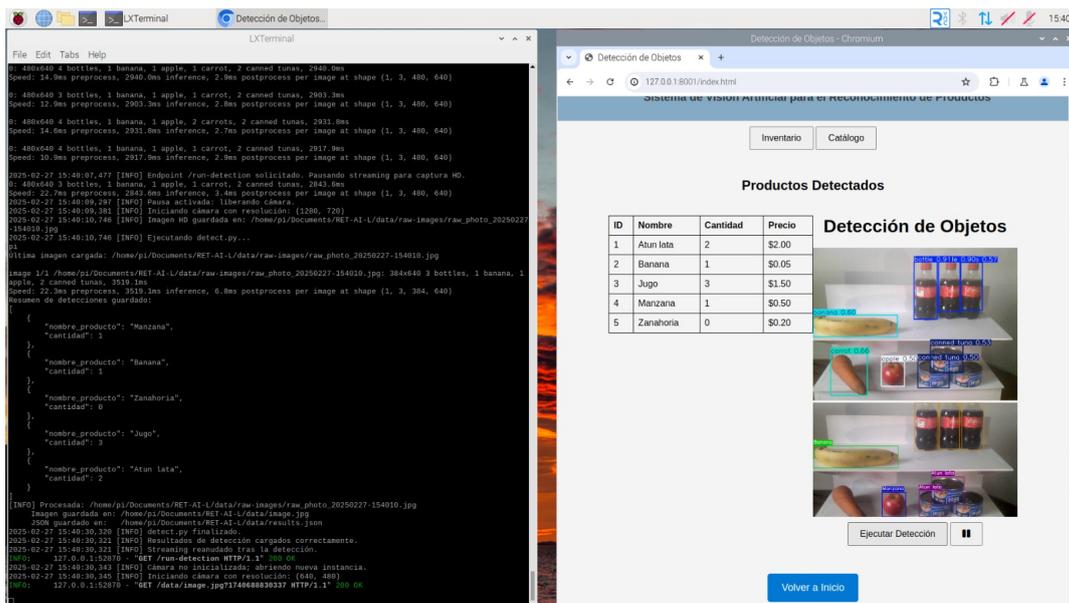


Figura 56. Captura del algoritmo con luz natural. Fuente: por B. Aviles y J. González

En la figura 57 se observa el funcionamiento del proyecto con iluminación regular.



Figura 57. Iluminación Regular. Fuente: por B. Aviles y J. González

VII. CRONOGRAMA

El cronograma de actividades del trabajo sera detallado en la tabla III

Tabla III
CRONOGRAMA. FUENTE: POR B. AVILES Y J. GONZÁLEZ

VII. CRONOGRAMA Y ACTIVIDADES A DESARROLLAR Tabla I. Cronograma de Actividades. Referencia: Autor		Noviembre							Diciembre				Enero				Febrero					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
No.	Tareas por realizar durante el proceso del Proyecto																					
1	Recolección de datos																					
2	Búsqueda y validación de base de datos de productos																					
3	Pruebas para búsqueda de algoritmo																					
4	Validación de algoritmo																					
5	Prueba de algoritmo con cinco productos propuestos																					
6	Montaje de componentes en prototipo de PCBs																					
7	Elaboración de la interfaz																					
8	Pruebas iniciales																					
9	Validación de proyecto y ajustes finales																					
10	Documentación final																					

VIII. PRESUPUESTO

Tabla IV
TABLA DE PRESUPUESTO. FUENTE: AUTOR

Nombre del elemento	Descripción	Cantidad	Valor total
Raspberry Pi	Controlador	1	120\$
Genius Facecam 1000x v2	Toma de imágenes y proceso de datos	1	26\$
Iluminación	Mejorar la luminosidad del entorno	1	20\$
Portátil 1	Instrumento de ayuda para el proyecto	1	830\$
Portátil 2	Instrumento de ayuda para el proyecto	1	510\$
Cables (HDMI)	Material para conexiones del proyecto	1	2\$
Percha	Objeto que se utiliza para exhibir productos	1	24 \$
Base de cámara	soporte para sostener la cámara	1	17 \$
SUBTOTAL MAQUETA			1549\$
Horas de trabajo		480 horas	4800\$
COSTOS SUBTOTAL DE LOGÍSTICOS			4800\$
TOTAL(COSTOS LOGÍSTICOS + MAQUETA)			6349\$

IX. CONCLUSIONES

Se evidenció que la iluminación juega un papel fundamental para precisión en un sistema asistido por visión artificial. Mientras la luz natural ayuda a reducir sombras y minimizar falsos positivos, la iluminación baja o irregular genera errores significativos en la detección y clasificación de productos.

La integración de un sistema de visión artificial basado en YOLO resultó ser una solución efectiva para el reconocimiento automatizado de productos en las perchas de los supermercados. A través del procesamiento en tiempo real, se mejoró notablemente la precisión en la identificación y categorización de los artículos, disminuyendo los errores en su colocación y distribución. Esta tecnología permitió un monitoreo dinámico del inventario, facilitando la automatización de tareas y optimizando la gestión logística dentro del establecimiento.

El diseño de una interfaz de usuario intuitiva y accesible hizo posible el seguimiento de la reposición de productos según el comportamiento de ventas. Esto permitió analizar patrones de demanda y tendencias de consumo, optimizando el reabastecimiento de manera más eficiente y adaptada a las necesidades del mercado. Como consecuencia, se mejoró la disponibilidad de los productos en las perchas, minimizando la escasez de stock y mejorando la experiencia de compra de los clientes.

X. RECOMENDACIONES

Es fundamental calibrar la cámara antes de trabajar con visión artificial para minimizar errores en el procesamiento de imágenes. La iluminación del entorno influye considerablemente en la calidad de la captura, por lo que es recomendable asegurarse de que haya una luz adecuada. De esta manera, se mejora el rendimiento del sistema y se reducen posibles lecturas incorrectas o falsos positivos.

Es esencial mantener la cámara estable, ya que cualquier movimiento puede afectar el procesamiento de datos y generar inestabilidad en el sistema.

Para antes del funcionamiento del algoritmo se debe primero medir el ángulo entre la cámara y la percha que muestra los productos, ya que con eso se evita la pérdida de tiempo y se obtienen resultados con mayor facilidad.

Para optimizar el rendimiento del sistema, se puede optar por mejorar la capacidad de procesamiento y memoria de la Raspberry Pi, así como utilizar una cámara de mayor calidad.

REFERENCIAS

- [1] J. D. M. G. J. G. V. García, *Diseño y simulación de estanterías inteligentes mediante el uso de inteligencia artificial*, Accedido: 06-2024, 2020. dirección: <https://www.redalyc.org/journal/5045/504551272009/html/>.
- [2] M. García, *Sistema para control de inventario por visión artificial*, Accedido: 11-2024, 2023. dirección: <https://rinacional.tecnm.mx/jspui/handle/TecNM/7140>.
- [3] M. L. Luis Asencio Edwin Gonzáles, *El inventario como determinante en la rentabilidad de las distribuidoras farmacéuticas*, Accedido: 06-2024, 2017. dirección: <https://www.redalyc.org/journal/5045/504551272009/html/>.
- [4] Majdi, «Product Stock Management Using Computer Vision,» *IEEE*, 2020.
- [5] E. Hernández, *Sin cajas y con inteligencia artificial, Oxxo abre su primera tienda Grab Go*, Accedido: 06-2024, 2023. dirección: <https://www.forbes.com.mx/sin-cajas-y-con-inteligencia-artificial-oxxo-abre-su-primera-tienda-grab-go/>.
- [6] Brems y Matt, *What is Computer Vision?* Accedido: 02-2025, 2025. dirección: <https://blog.roboflow.com/intro-to-computer-vision/>.
- [7] R. Gonzalez, *Digital Image Processing*, Accedido: 01-2025, 2001. dirección: <https://dl.icdst.org/pdfs/files4/01c56e081202b62bd7d3b4f8545775fb.pdf>.
- [8] A. R. Manzanares, *Reporte de Búsqueda y detección de conteo de objetos*, Accedido: 05-2024. dirección: https://www.researchgate.net/profile/Alonso-Ramirez-Manzanares/publication/228549378_Reporte_de_Busqueda_deteccion_y_conteo_de_objetos/links/541a13c40cf2218008bfa68d/Reporte-de-Busqueda-deteccion-y-conteo-de-objetos.pdf.
- [9] Urrutia y David, *Conteo automático por visión artificial*, Accedido: 02-2025, 2024. dirección: <https://atriainnovation.com/blog/conteo-automatico-por-vision-artificial/>.
- [10] J. J. E. Elizondo, *Fundamentos de Procesamientos de Imágenes*, Accedido: 05-2024, 2002. dirección: <https://t.ly/fNFYb>.
- [11] L. Tonioni Alessio; Di Stefano, *Image Analysis and Processing - ICIAP 2017*. Springer International Publishing, 2017.
- [12] M.-T. Lin Jia-Ping; Sun, «A YOLO-based traffic counting system,» *IEEE*, 2018.
- [13] A. Fernandez, *Python 3 al descubierto - 2a ed.* México: Alfaomega Grupo Editor, 2013.
- [14] Xukyo, *Reconocimiento de objetos con Python*, Accedido: 06-2024, 2023. dirección: <https://www.aranacorp.com/es/reconocimiento-de-objetos-con-python/>.
- [15] S. Brahmabhatt, *Practical OpenCV*. New York, EEUU: APRESS, 2013.
- [16] M. López, *Librerías de Python, ¿qué son y cuáles son las mejores?* Accedido: 06-2024, 2022. dirección: <https://immune.institute/blog/librerias-python-que-son/>.
- [17] PyTorch, *PyTorch*, Accedido: 02-2025, 2025. dirección: <https://pytorch.org>.
- [18] G. Jocher, A. Chaurasia y J. Qiu, *Ultralytics YOLOv8*, ver. 8.0.0, 2023. dirección: <https://github.com/ultralytics/ultralytics>.
- [19] F. Falcini, «Deep learning in automotive software,» *IEEE software*, 2017.
- [20] J. G. Pérez, *Introducción a ROS en Raspberry Pi*, Accedido: 06-2024, 2017. dirección: <https://openaccess.uoc.edu/bitstream/10609/63705/3/jgutierrezperTFM0617memoria.pdf>.
- [21] N. Digital, *CAMARA WEB GENIUS FACECAM 1000X V2 720P*, Accedido: 02-2025, 2025. dirección: <https://ntdigital.ar/producto/camara-web-genius-facecam-1000x-v2-720p?>.

APÉNDICE A
PLANOS

Planos de las piezas acotadas en Modelados de piezas en 3D:

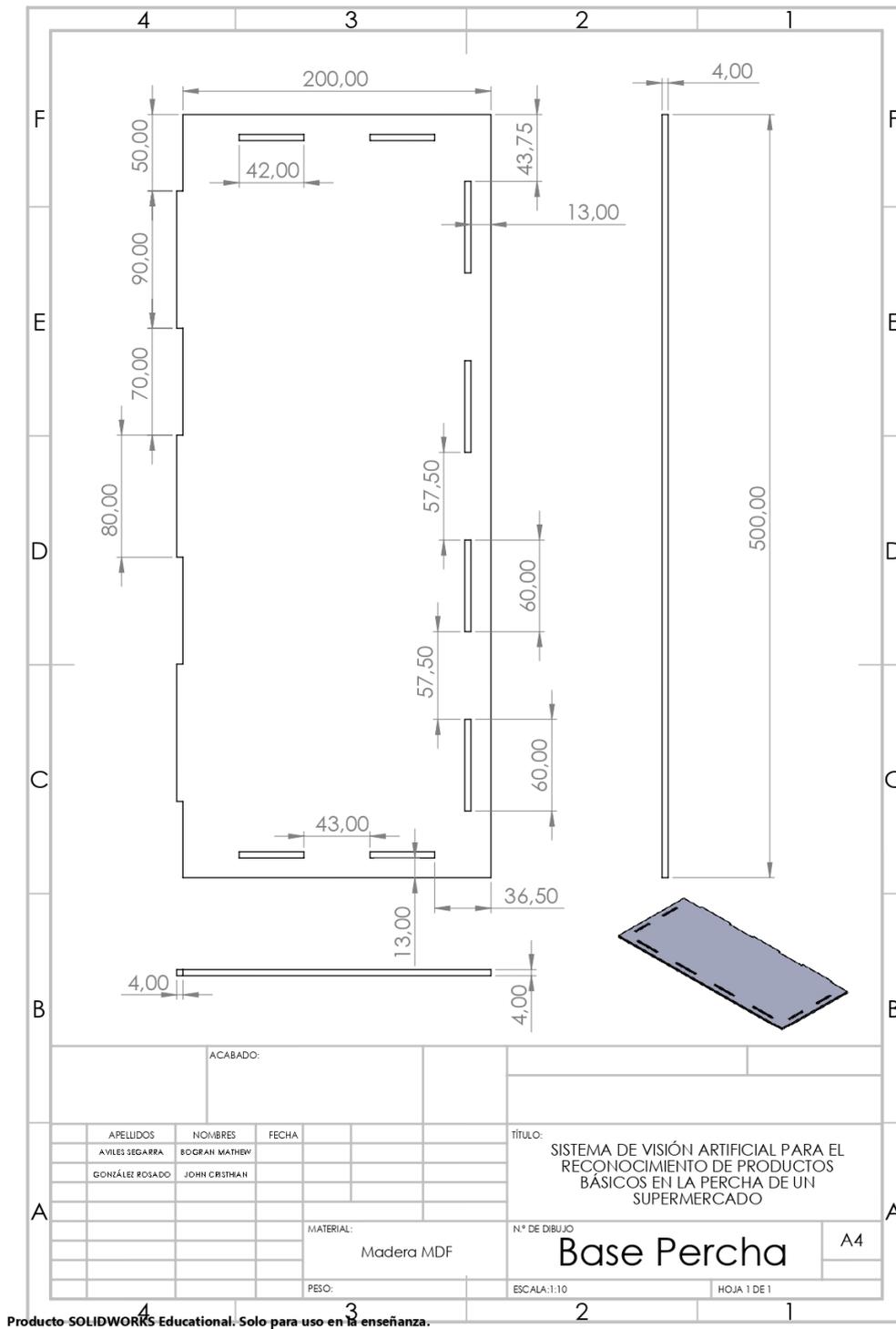
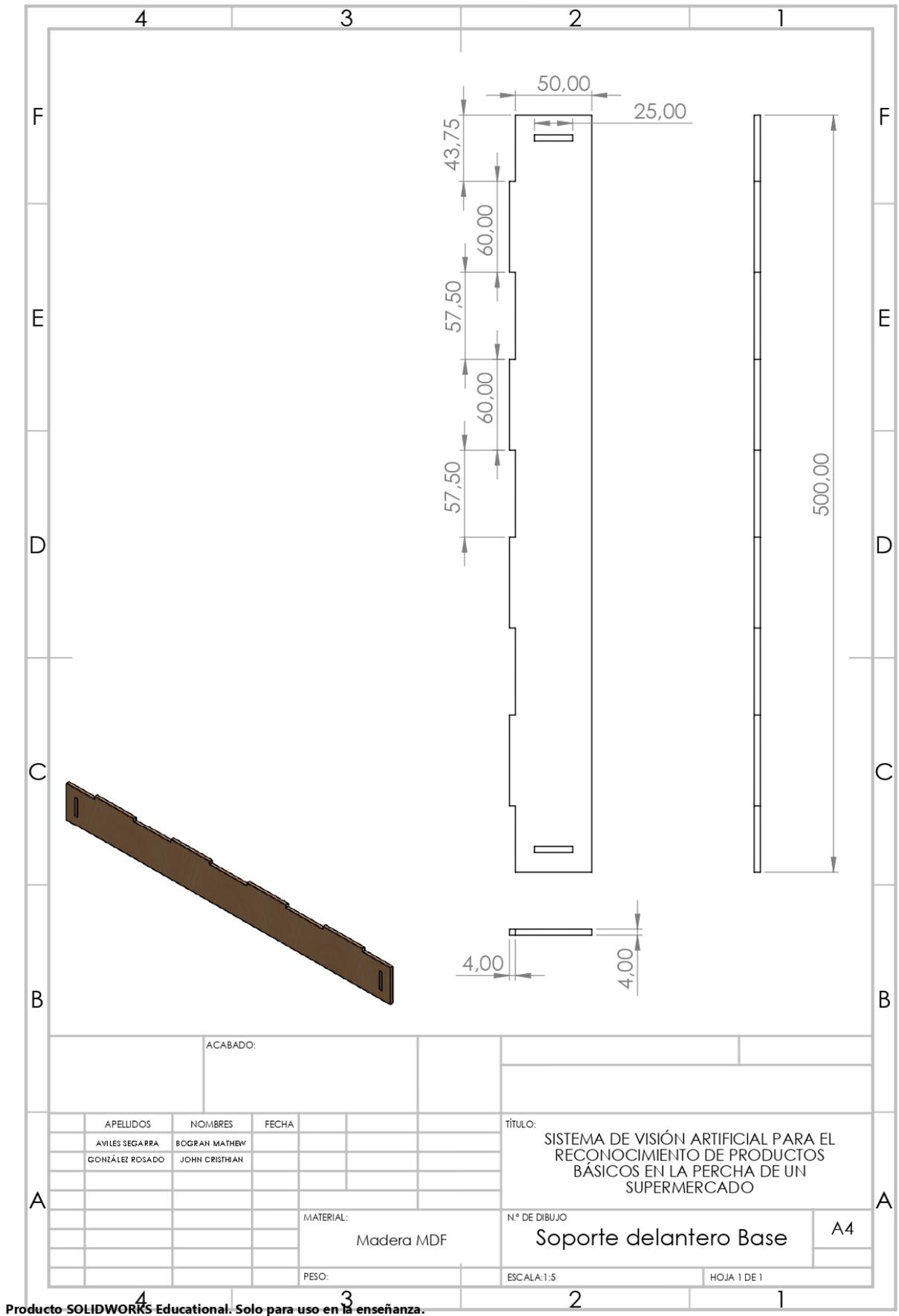
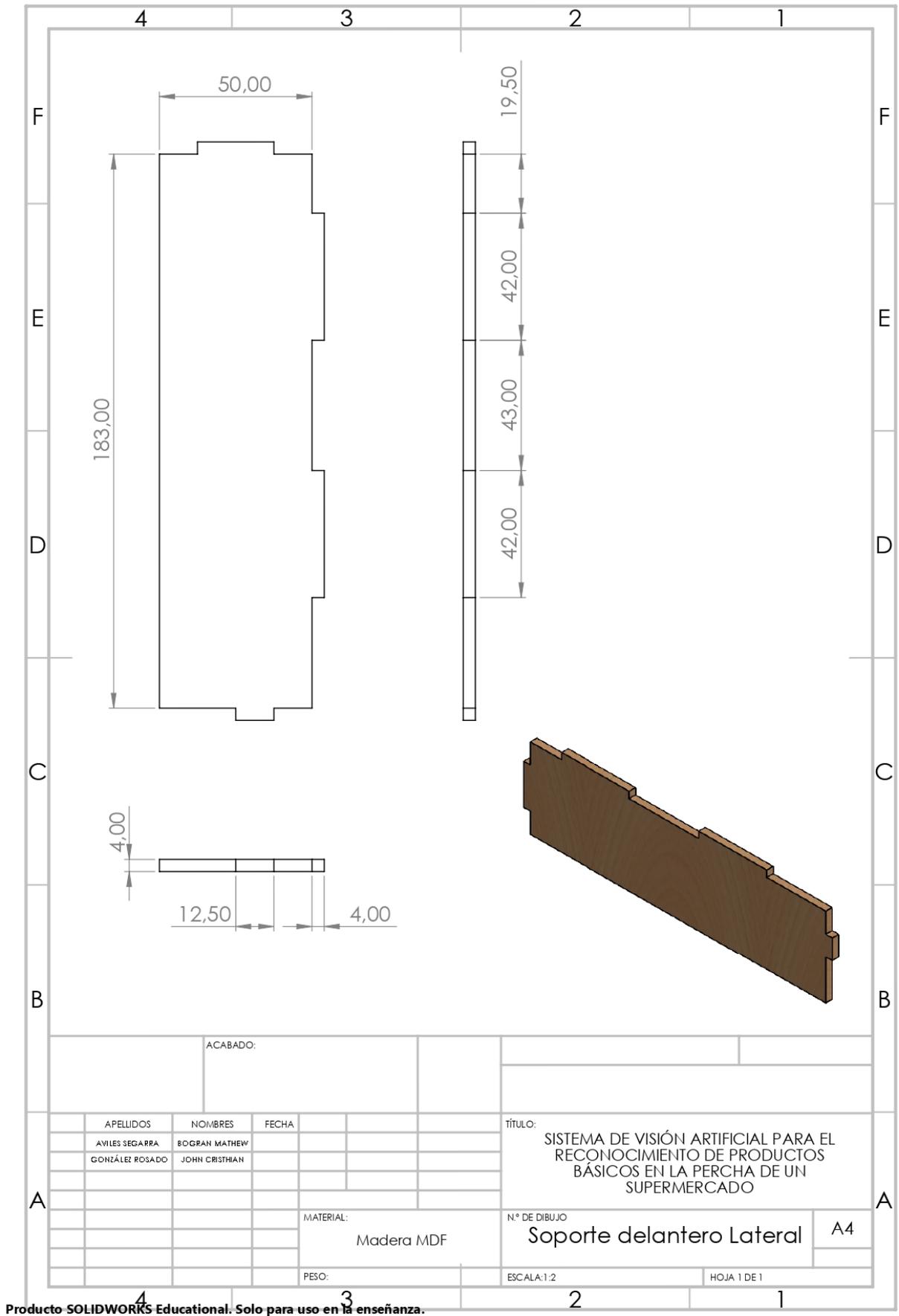


Figura 58. Base Percha, por B. Aviles y J. González, Plano de pieza



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 59. Soporte delantero Base, por B. Aviles y J. González, Plano de pieza



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 60. Soporte delantero Lateral, por B. Aviles y J. González, Plano de pieza

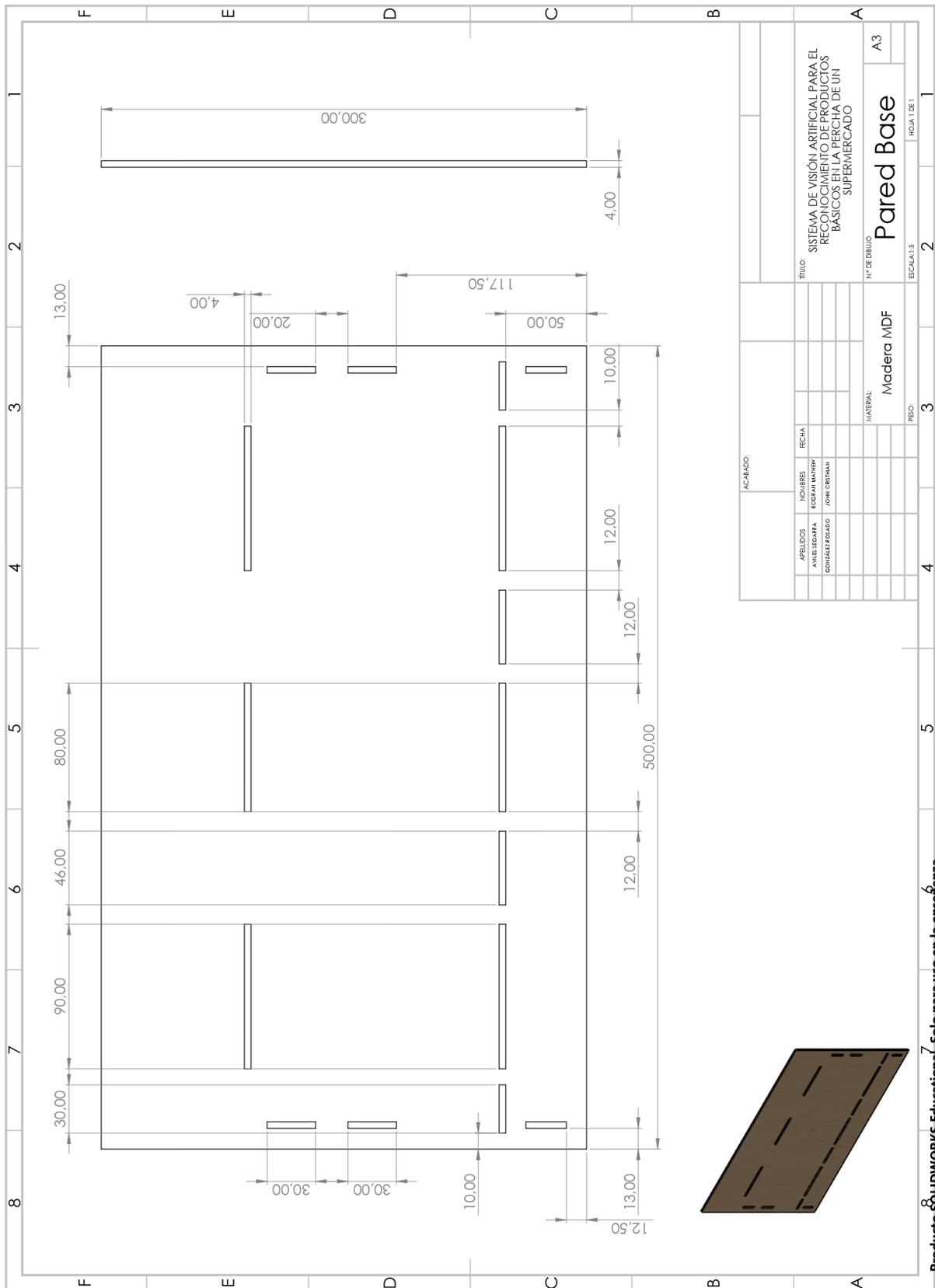
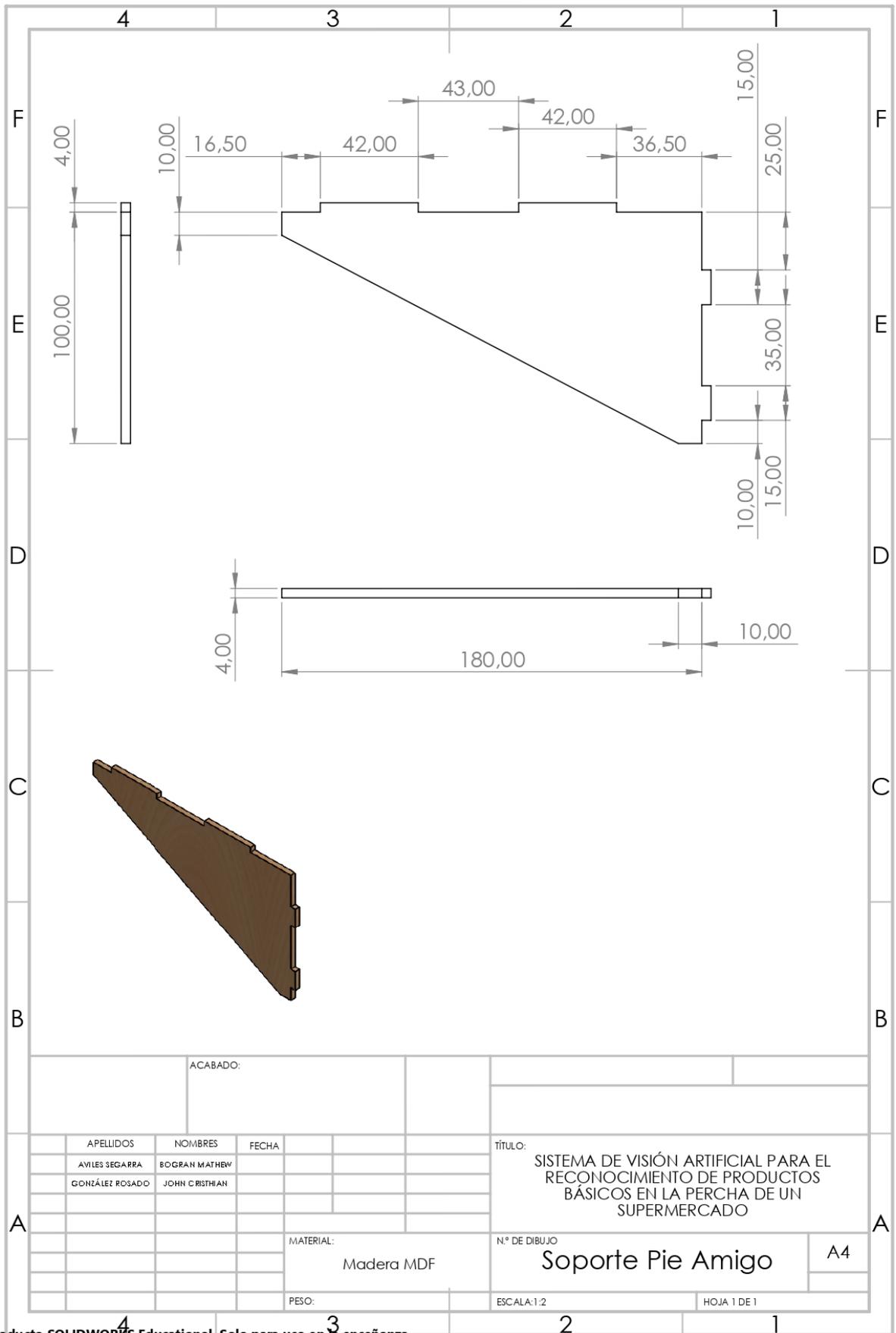


Figura 61. Pared Base, por B. Aviles y J. González, Plano de pieza



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 62. Soporte Pie Amigo, por B. Aviles y J. González, Plano de pieza

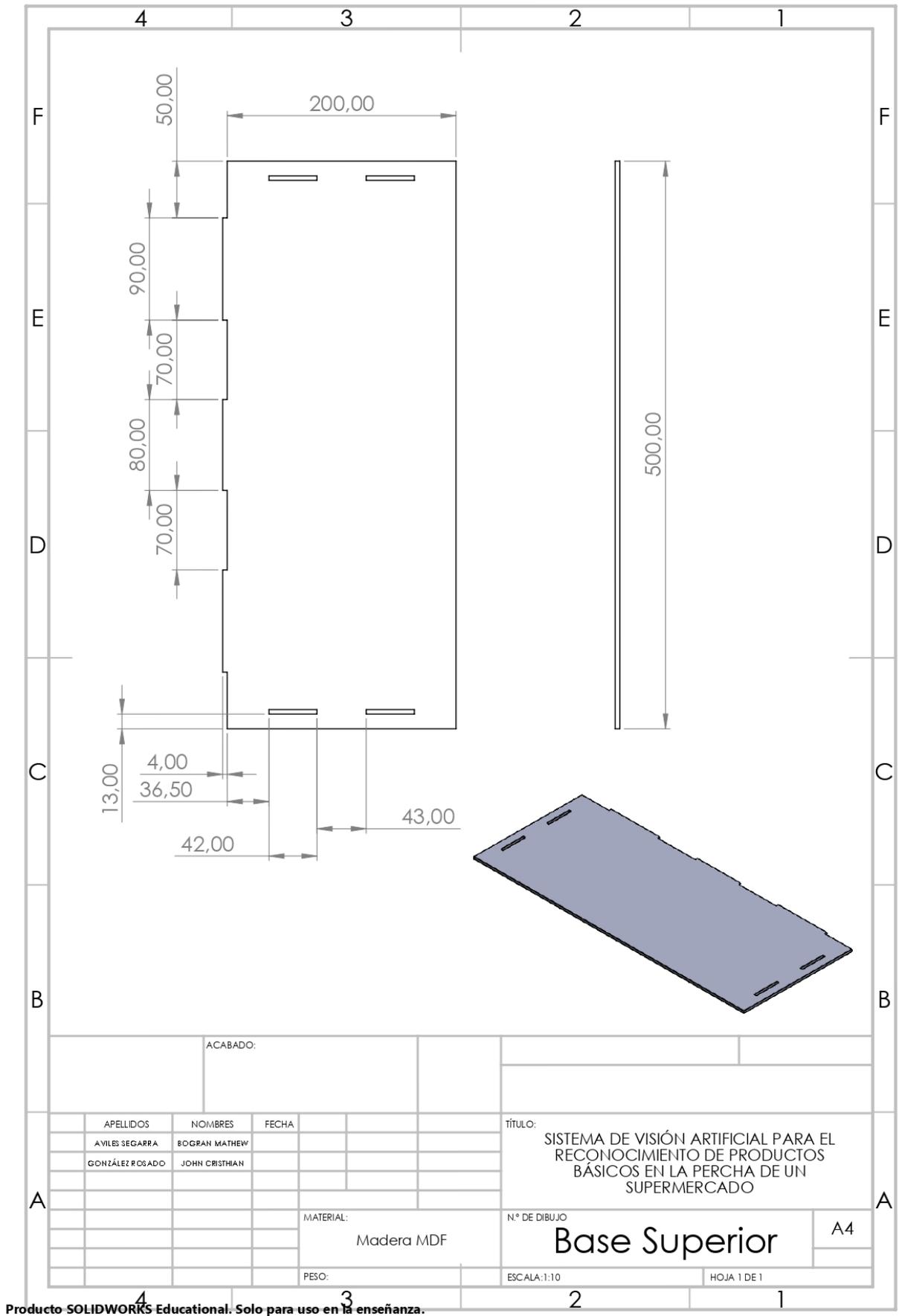


Figura 63. Base Superior, por B. Aviles y J. González, Plano de pieza

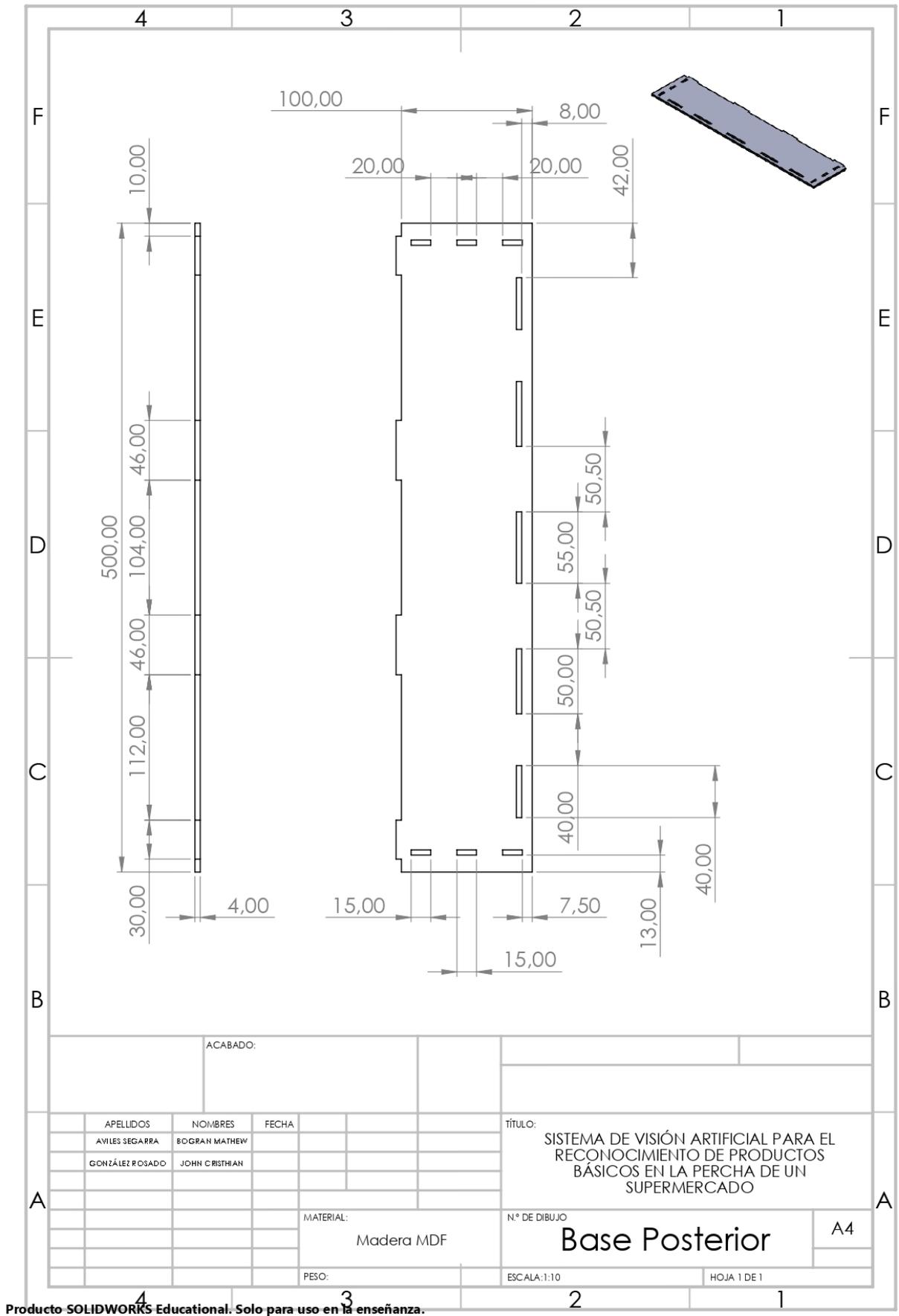
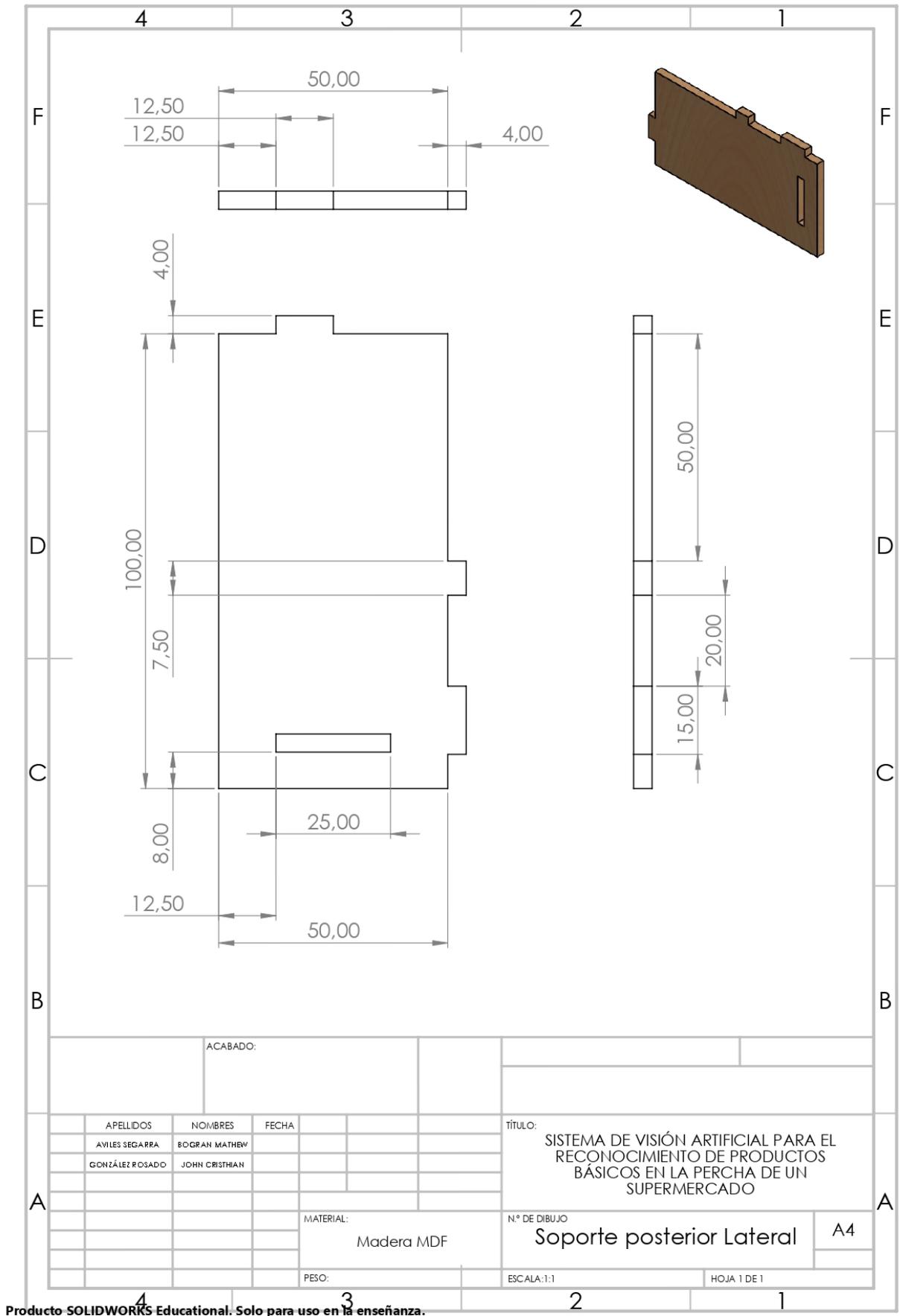


Figura 64. Base Posterior, por B. Aviles y J. González, Plano de pieza



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 65. Soporte posterior Lateral, por B. Aviles y J. González, Plano de pieza

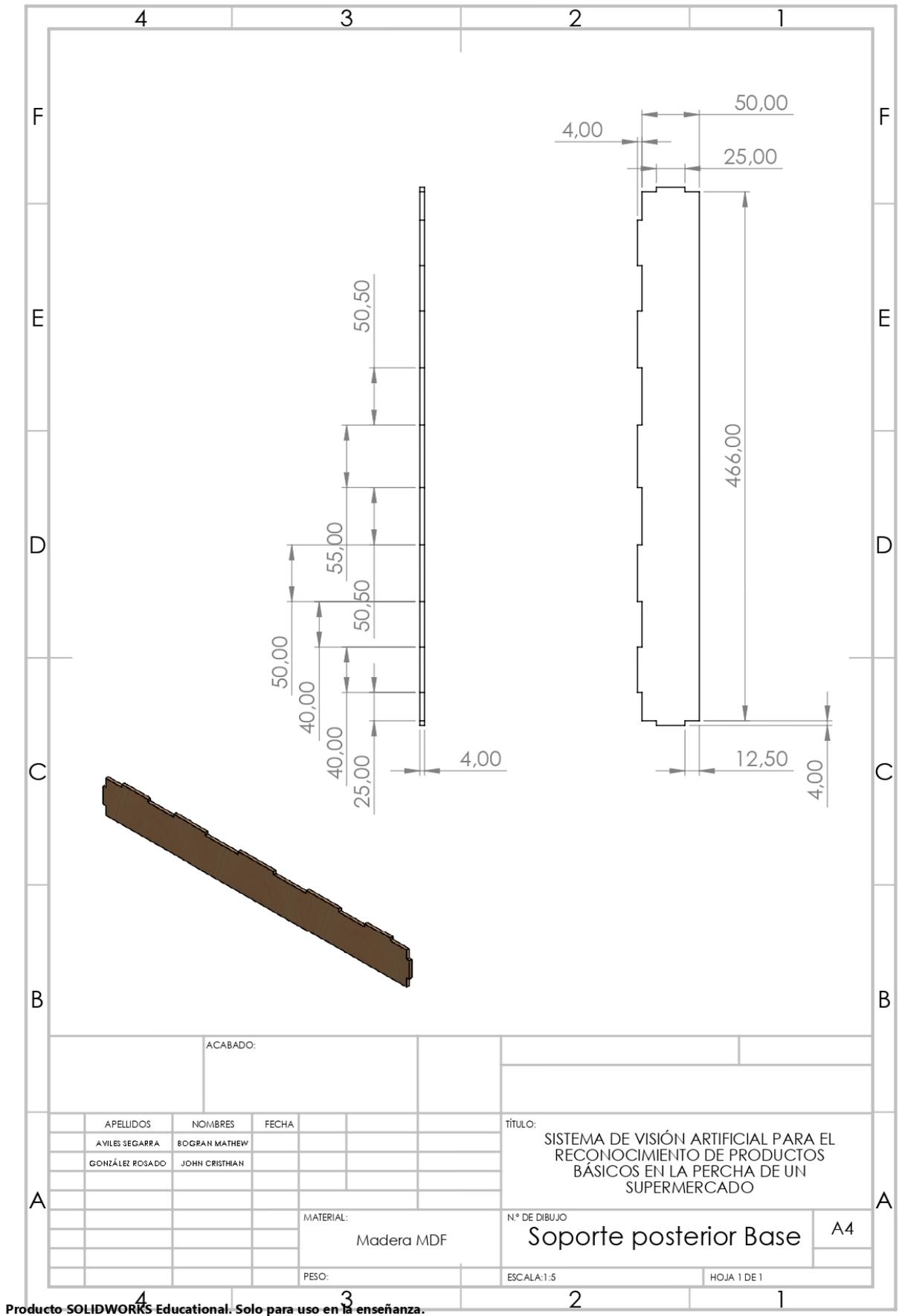


Figura 66. Soporte posterior Base, por B. Aviles y J. González, Plano de pieza

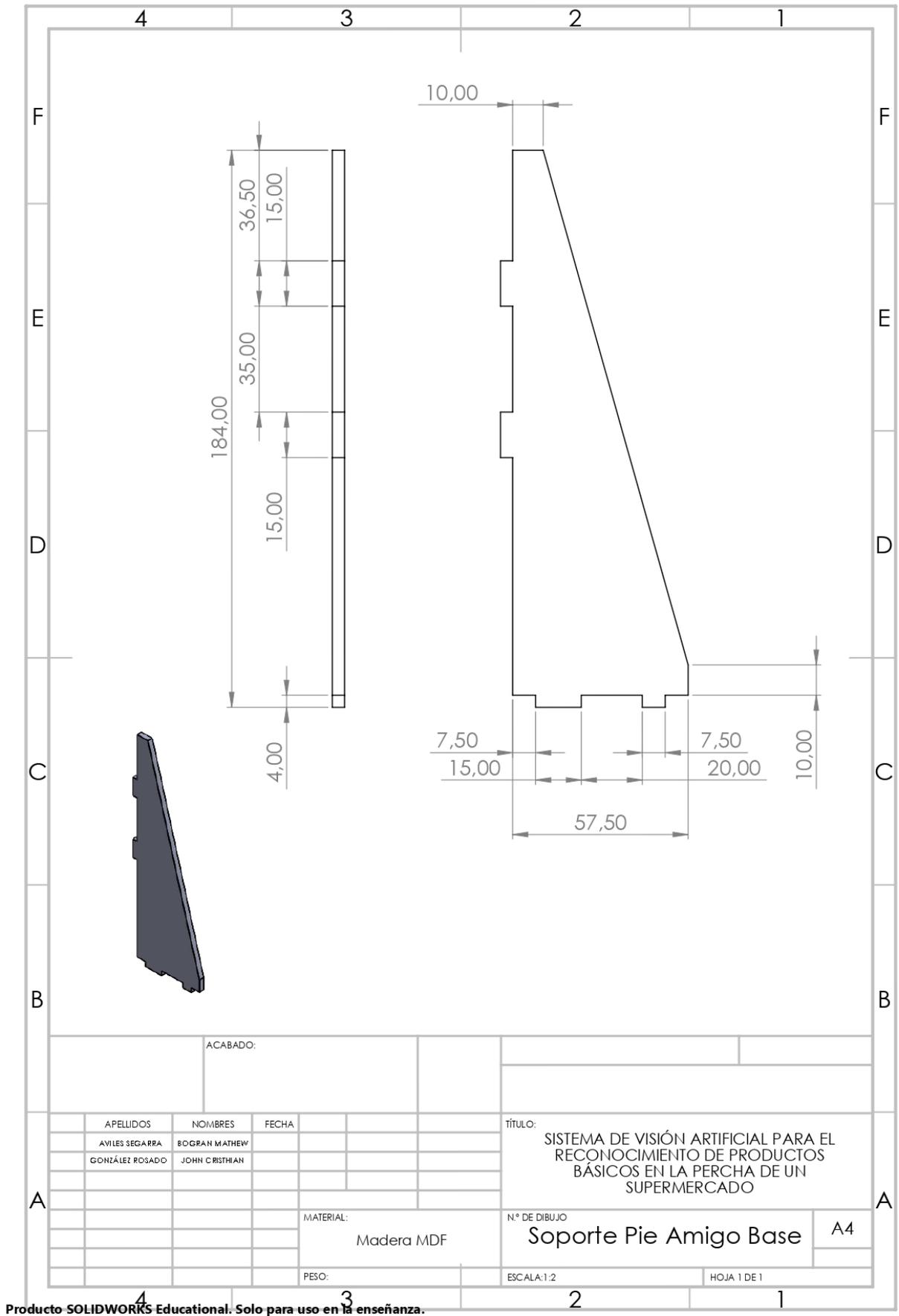
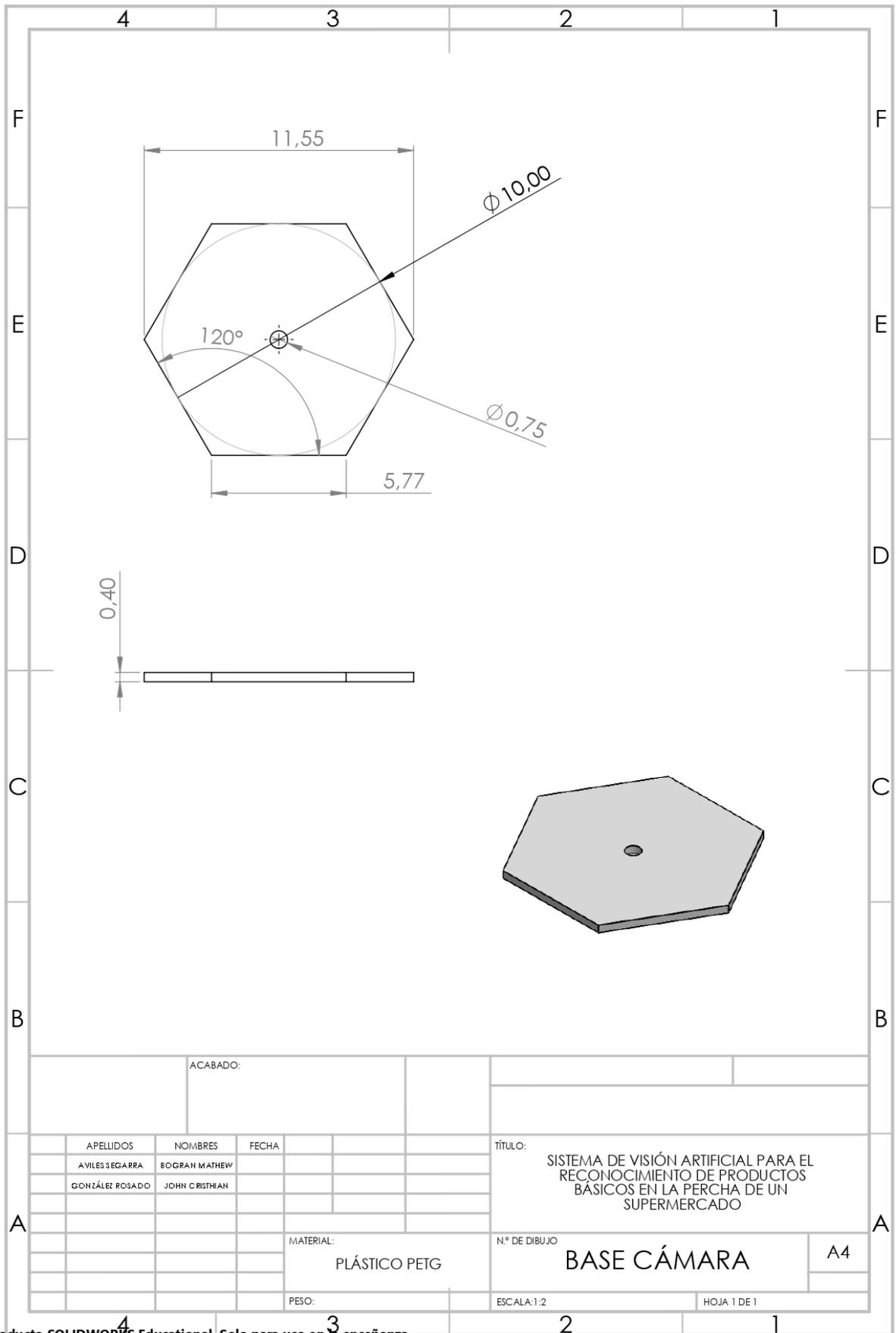


Figura 67. Soporte Pie Amigo Base, por B. Aviles y J. González, Plano de pieza



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 68. Base Cámara, por B. Aviles y J. González, Plano de pieza

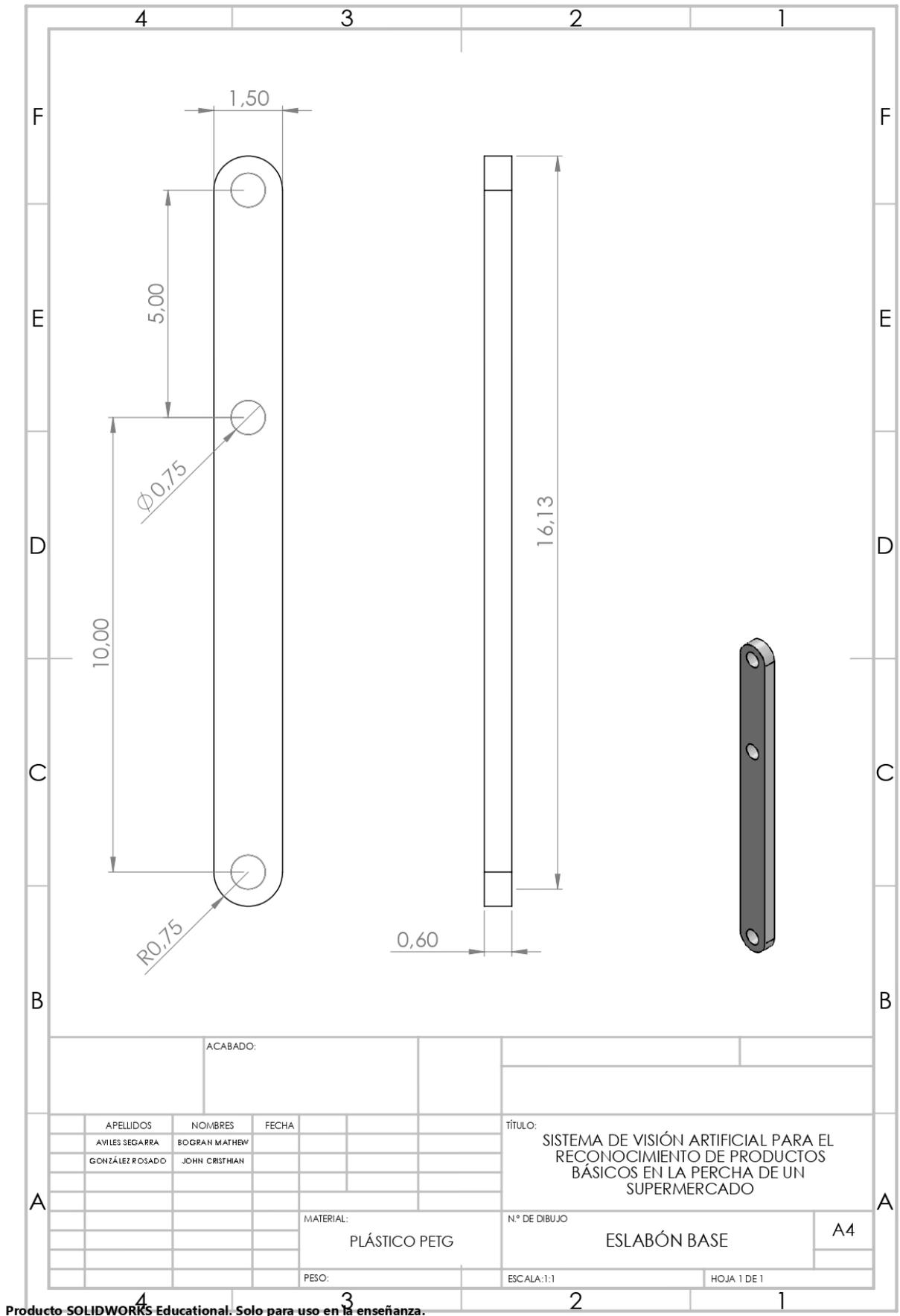
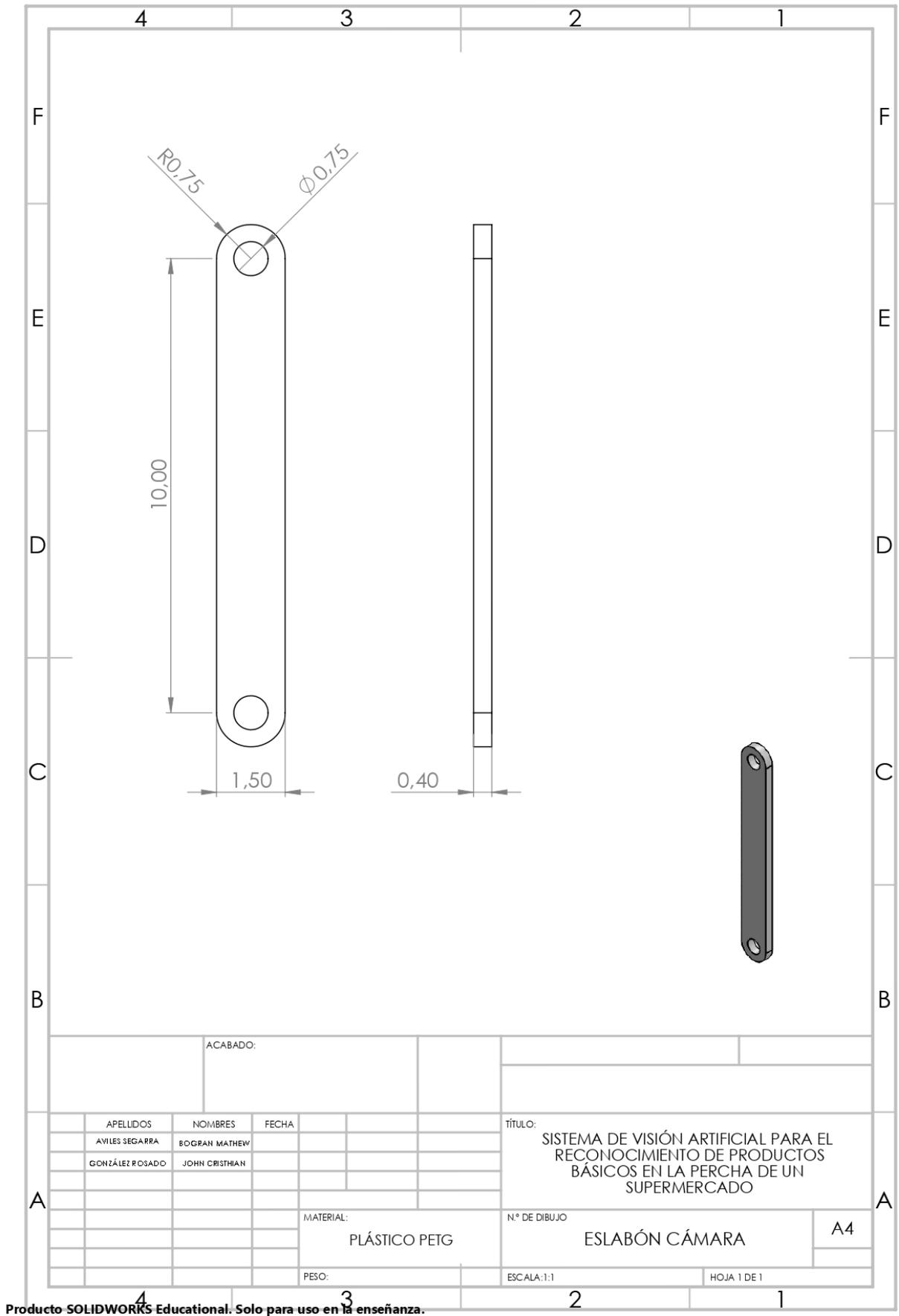
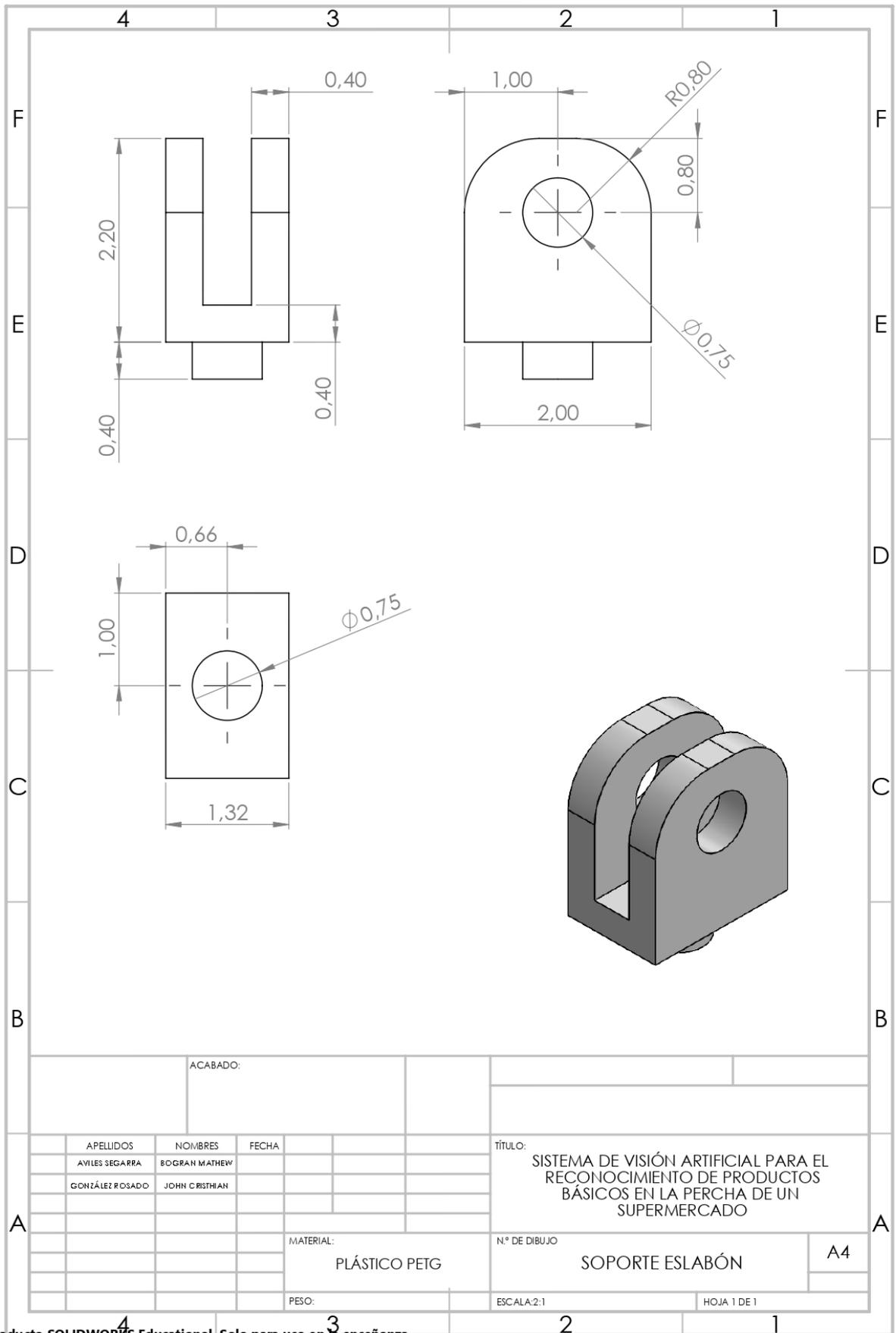


Figura 69. Eslabón Base, por B. Aviles y J. González, Plano de pieza



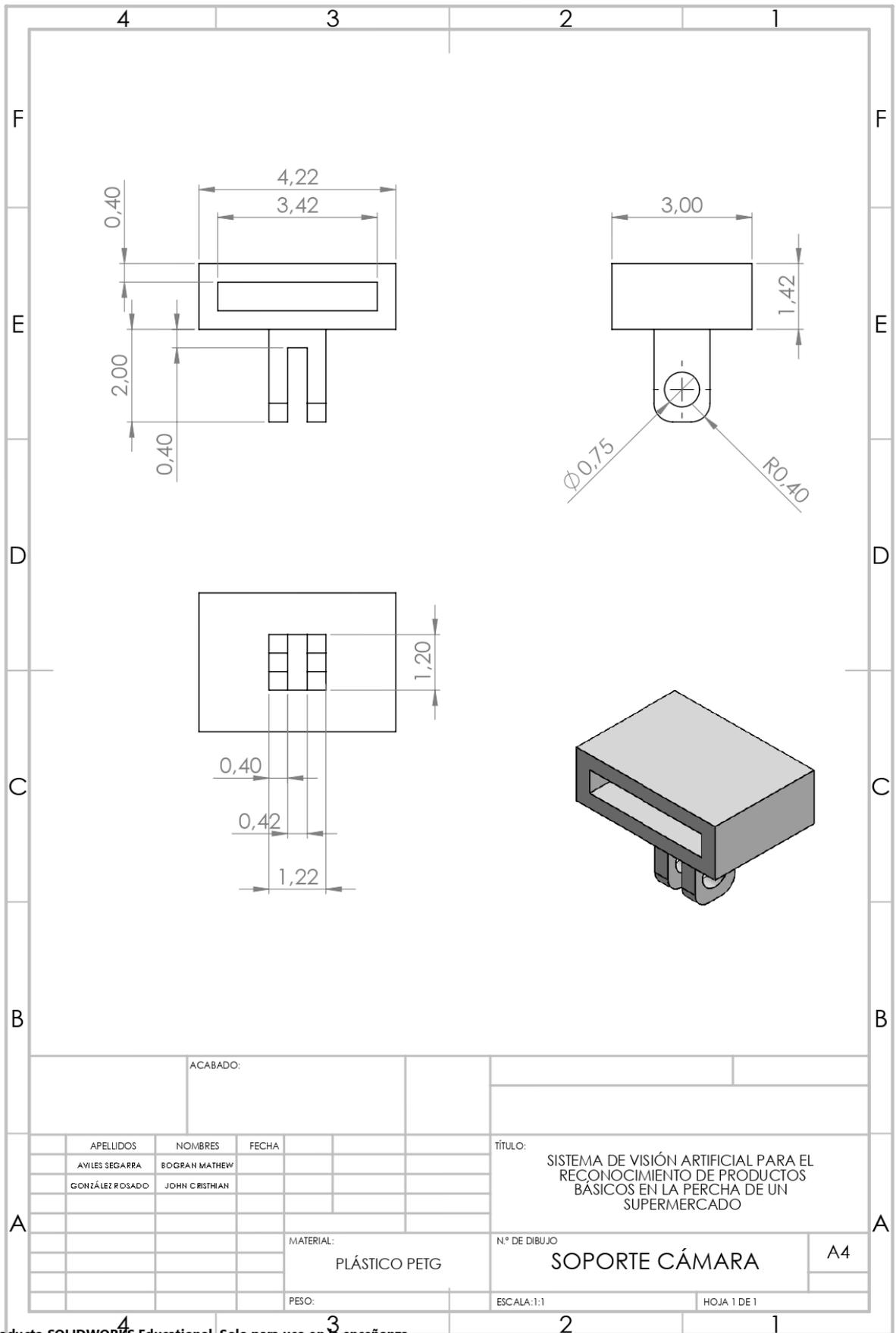
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 70. Base Cámara, por B. Aviles y J. González, Plano de pieza



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 71. Soporte Eslabón, por B. Aviles y J. González, Plano de pieza



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 72. Soporte Cámara, por B. Aviles y J. González, Plano de pieza

Plano de las pieza en 2d para el corte láser en madera mdf.

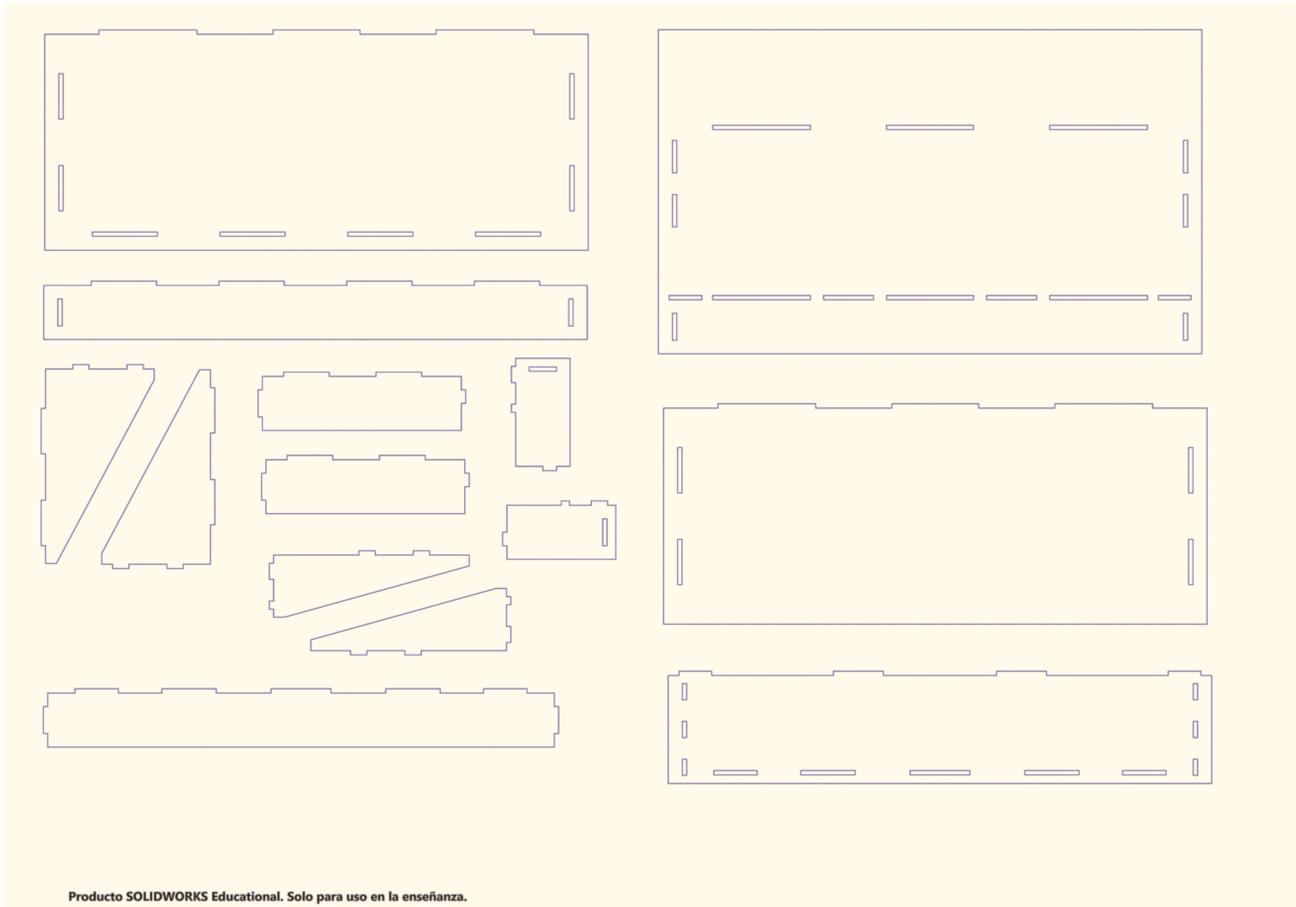


Figura 73. Lámina del contorno en madera mdf, por B. Aviles y J. González, Diseño en modelos 3D

Plano en 2D para la impresión de piezas en la impresora 3D.

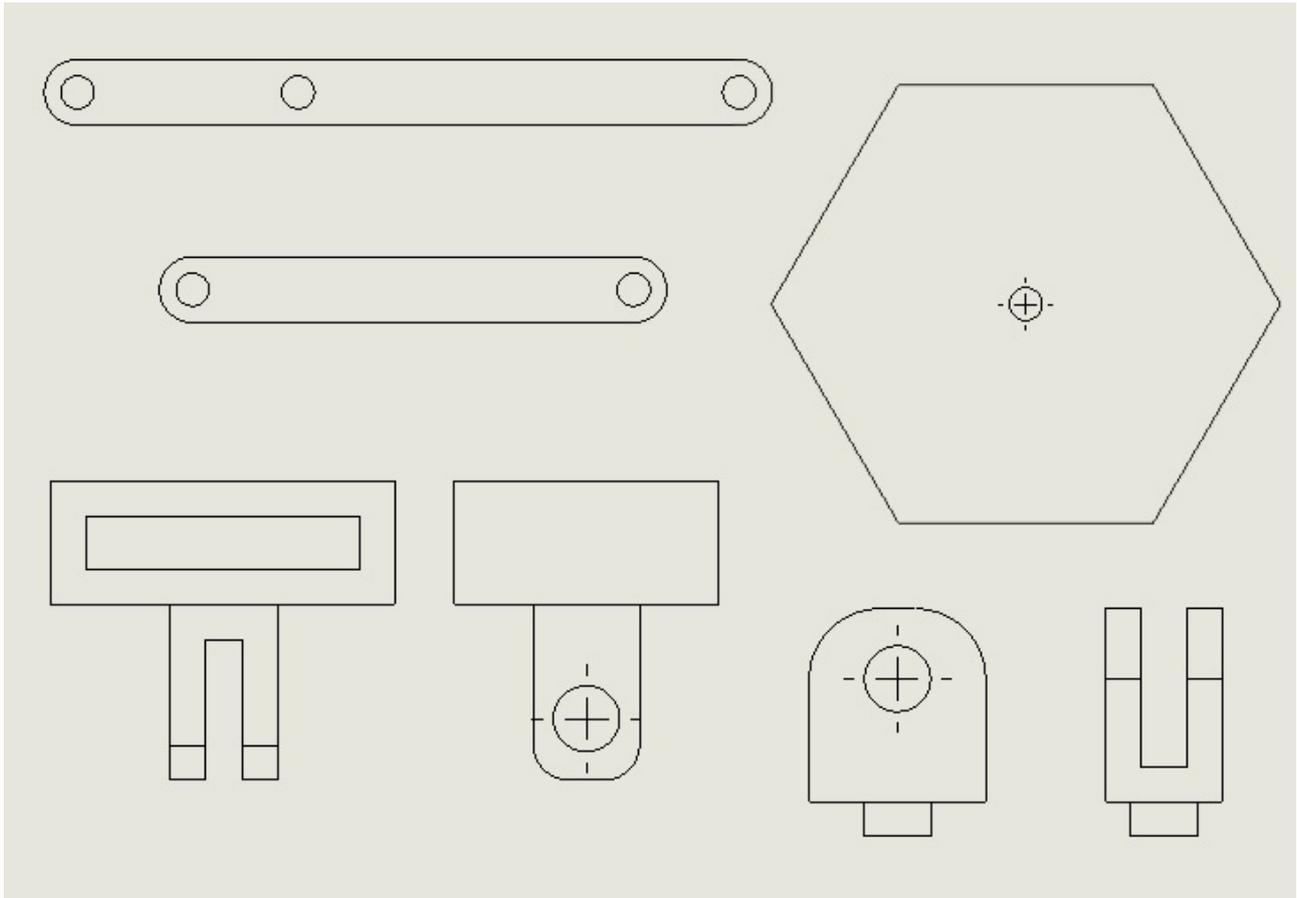


Figura 74. Lámina del prototipado de las piezas en plástico PETG, por B. Aviles y J. González, Diseño en modelos 3D

APÉNDICE A
EQUIPO DE MANUFACTURA



Figura 75. Máquina de Corte láser para las piezas en madera mdf, por B. Aviles y J. González



Figura 76. Impresora 3D para enviar a imprimir las piezas diseñadas, por B. Aviles y J. González

APÉNDICE B CÓDIGO DEL WEB SCRAPER

El siguiente código en Python implementa un web scraper para extraer información de productos desde Supermaxi.

```
import requests
from bs4 import BeautifulSoup
import os, re, json, csv, logging
from datetime import datetime

# Configuración inicial
SEARCH_TERM = "atun"
START_PAGE, END_PAGE = 1, 5
OUTPUT_FOLDER = "scraping_results"
timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
output_folder = os.path.join(OUTPUT_FOLDER, f"supermercado_{SEARCH_TERM}_{timestamp}")
os.makedirs(output_folder, exist_ok=True)

csv_file = os.path.join(output_folder, "productos.csv")
log_file = os.path.join(output_folder, f"scraping_{timestamp}.log")

# Configuración de Logging
logging.basicConfig(filename=log_file, level=logging.INFO, format="%(asctime)s [%(levelname)s] %(message)s")

# Función para obtener la página
def get_page_soup(page):
    url = f"https://www.supermercado.com/page/{page}/?s={SEARCH_TERM}"
    try:
        response = requests.get(url, timeout=10)
        response.raise_for_status()
        return BeautifulSoup(response.text, "html.parser")
    except:
        logging.warning(f"Error al acceder a la página {page}")
        return None

# Función principal
def main():
    for page in range(START_PAGE, END_PAGE + 1):
        soup = get_page_soup(page)
        if not soup:
            continue
        products = soup.find_all("article", class_="product")
        for p in products:
            title = p.find("h4", class_="entry-title")
            if title:
                logging.info(f"Producto encontrado: {title.text.strip()}")

if __name__ == "__main__":
    main()
```

Listing 1. Script de scraping de Supermaxi

APÉNDICE C CÓDIGO DEL MODELO DE APRENDIZAJE AUTOMÁTICO

El siguiente código en Python implementa un entrenamiento de un modelo pre-entrenado basado en YOLOv8 para reconocimientos de frutas-verduras y productos de abarrotes

```
# -*- coding: utf-8 -*-
"""
RET-AI-L
"""

!pip3 install --upgrade torch torchvision torchaudio --extra-index-url https://download.
    pytorch.org/whl/cu117
!pip install roboflow

from google.colab import drive
drive.mount('/content/drive')

import torch
torch.cuda.is_available()

!pip install --upgrade ultralytics

# from kaggle_secrets import UserSecretsClient
# user_secrets = UserSecretsClient()
# ROBOFLOW_API_KEY = user_secrets.get_secret("ROBOFLOW_API")

from roboflow import Roboflow
rf = Roboflow(api_key="")
project = rf.workspace("dev").project("atunes")
version = project.version(1)
dataset = version.download("yolov8")

from roboflow import Roboflow
rf = Roboflow(api_key="")
project = rf.workspace("dev").project("salsas")
version = project.version(1)
dataset = version.download("yolov8")

from roboflow import Roboflow
rf = Roboflow(api_key="")
project = rf.workspace("dev").project("frutas-vegetales")
version = project.version(3)
dataset = version.download("yolov8")

#
!yolo task=detect mode=train epochs=50 data=/content/My-First-Project-1/data.yaml model=
    yolov8m.pt patience=30
!yolo task=detect mode=train epochs=20 data=/content/Fruits-and-Vegetables-3/data.yaml
    model=/content/fruits-best.pt patience=30

from ultralytics import YOLO
from PIL import Image

image_path = "/content/IMG_20250130_080802.jpg"
image = Image.open(image_path)

model = YOLO("/content/runs/detect/train/weights/best.pt")
```

```
model = YOLO("/content/fruits-best.pt")

results = model.predict(source=image, conf=0.1, save=True) # save plotted images

from IPython.display import display, Image
display(Image(filename='/kaggle/working/runs/detect/predict/image0.jpg'))
```

Listing 2. Script de fine-tuning de modelo de aprendizaje