

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Mecánica y Ciencias de la**  
**Producción**

Diseño y simulación de estanterías inteligentes mediante el uso  
de inteligencia artificial

**PROYECTO INTEGRADOR**

Previo la obtención del Título de:  
**INGENIERO EN MECATRÓNICA**

Presentado por:

José Daniel Moreno González

Juan Guillermo Vera García

GUAYAQUIL - ECUADOR

Año: 2020

## DEDICATORIA

A mis padres Xavier Moreno y Patricia González, quienes han sido mi ejemplo a seguir y me inculcaron los valores que me han hecho la persona que soy el día de hoy. A mi hermano, por alentarme durante estos años. A mis abuelos, quienes creyeron en mí y siempre estuvieron presentes brindándome su cariño y apoyo.

**José Daniel Moreno González**

## DEDICATORIA

El presente trabajo se lo dedico de manera especial a mis padres por el apoyo, consejos, cariño y comprensión brindada en cada etapa de mi vida, quienes siempre han depositado la confianza en sus hijos y han sido claro ejemplo de esfuerzo y superación.

**Juan Guillermo Vera García**

## **AGRADECIMIENTOS**

Mi eterna gratitud a Dios y a mi familia, en especial a mis padres, quienes me han sabido guiar y han edificado mi vida a base de esfuerzo y sacrificios. A mi compañero Juan Vera por su gran aporte en el desarrollo de este proyecto. Al Ph.D Marcelo Fajardo por habernos brindado de su tiempo, experiencia y conocimientos a lo largo de la realización de este trabajo.

**José Daniel Moreno González**

## **AGRADECIMIENTOS**

Agradezco a Dios y mi familia, mis padres Juan Vera y Nelly García, a mis tías Dennys García, Cecilia Vera y Rebeca Vera, por todo el apoyo brindado y acompañarme en cada uno de los procesos que conlleva lograr este objetivo. También quiero agradecer a mi compañero José Moreno por el gran desempeño demostrado durante el proyecto; y al tutor Ph.D Marcelo Fajardo por su ayuda otorgada en el desarrollo de este trabajo.

**Juan Guillermo Vera García**

## DECLARACIÓN EXPRESA

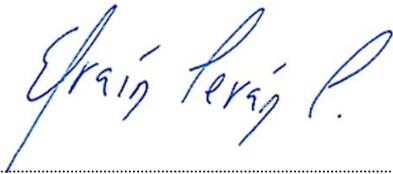
"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; José Daniel Moreno González y Juan Guillermo Vera García damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

A handwritten signature in blue ink, appearing to read 'José Daniel Moreno González', with a stylized, cursive script.

José Daniel Moreno González

Juan Guillermo Vera García

## EVALUADORES



---

Efraín Terán, M.Sc.  
PROFESOR DE LA MATERIA



Firmado electrónicamente por:  
MARCELO RODOLFO  
FAJARDO PRUNA

---

Marcelo Fajardo, Ph.D.  
PROFESOR TUTOR

## RESUMEN

El sector de comercio minorista está sujeto a enfrentar continuamente desafíos sobre la calidad de servicio que brindan al cliente ante un entorno comercial competitivo, por lo cual, para un óptimo funcionamiento de este modelo de negocio, los procesos de reposición y organización de la mercadería exhibida en las estanterías son aspectos clave a considerar. En la actualidad, se acostumbra a contratar personal para recorrer las estanterías una a una en busca de eventualidades a solventar; sin embargo, esto resulta poco eficiente, además conlleva una capacidad de reacción lenta, y está sujeto a error humano. Es por ello que el presente trabajo propone un sistema de monitoreo automático de abastecimiento y orden de productos, por medio de una cámara montada sobre una plataforma móvil para su desplazamiento a lo largo de toda una estantería. El diseño consistió en un software que utiliza inteligencia artificial para el reconocimiento de objetos, entrenado con una base de datos de más de 2000 imágenes con productos de supermercado. Posteriormente se diseñó la plataforma móvil en conjunto con el sistema embebido encargado de darle movilidad. Por último, se elaboró una base de datos junto con un aplicativo móvil a modo de planograma, sirviendo de interfaz de usuario. Mediante numerosas pruebas se comprobó el funcionamiento del software de detección, alcanzando una precisión de 95%, y evidenciando sus resultados en el aplicativo móvil en tiempo real. A su vez se hizo un análisis de elementos finitos para determinar la resistencia de la estructura móvil del sistema.

**Palabras Clave:** supermercado, planograma, inteligencia artificial, deep learning, monitoreo automático.

## **ABSTRACT**

*The retail industry is continually dealing with challenges on the quality of service provided to the customer in a competitive commercial environment, therefore, for an optimal functioning of this business model, the processes of replacement and organization of the merchandise exhibited on shelves are key aspects to consider. Currently, it is usual to hire personnel to go through the shelves one by one searching for eventualities to solve; However, this is not very efficient, additionally, it entails a slow reaction capacity, and it is susceptible to human error. That is why the present work proposes an automatic monitoring system of the supply and order of products, using a camera mounted on a mobile platform that allows it to move along an entire shelf. The design consisted of a software that uses artificial intelligence for object recognition, trained with a dataset of more than 2000 images with supermarket products. Later, a mobile platform was designed in addition with the embedded system responsible for giving its mobility. Finally, a database was developed together with a mobile application that works as a planogram, serving as a user interface. Through numerous tests, the operation of the detection software was verified, reaching a precision of 95%, and showing its results in the mobile application in real time. Additionally, a finite element analysis was carried out to determine the resistance of the mobile structure of the system.*

**Keywords:** *supermarket, planogram, artificial intelligence, deep learning, automatic monitoring.*

# ÍNDICE GENERAL

EVALUADORES.....	3
RESUMEN.....	I
ABSTRACT .....	II
ÍNDICE GENERAL .....	III
ABREVIATURAS.....	VI
SIMBOLOGÍA .....	VII
ÍNDICE DE FIGURAS .....	VIII
ÍNDICE DE TABLAS.....	X
ÍNDICE DE PLANOS.....	XI
CAPÍTULO 1.....	1
1.    Introducción .....	1
1.1    Descripción del problema .....	4
1.2    Justificación del problema .....	6
1.3    Objetivos .....	10
1.3.1    Objetivo general.....	10
1.3.2    Objetivos específicos .....	10
1.4    Marco teórico.....	11
1.4.1    Planograma.....	11
1.4.2    Redes neuronales.....	12
1.4.3    Reconocimiento de objetos.....	14
CAPÍTULO 2.....	16
2.    Metodología .....	16
2.1    Alternativas de solución.....	16
2.2    Proceso de diseño.....	20

2.3	Requerimientos de diseño .....	21
2.4	Diseño conceptual .....	21
2.5	Diseño del sistema móvil .....	22
2.6	Diseño del sistema embebido .....	24
2.7	Selección de actuadores .....	25
2.8	Diseño de las conexiones eléctricas.....	26
2.9	Diseño de software de monitoreo de la estantería .....	26
2.10	Selección de cámara de video .....	28
2.11	Selección de la unidad de cómputo y plataforma de programación .....	28
2.11.1	Unidad de cómputo.....	29
2.11.2	Plataforma de programación.....	29
2.12	Microcontrolador.....	30
2.13	Diseño de aplicación móvil .....	31
2.14	Diseño de base de datos .....	32
CAPÍTULO 3.....		34
3.	Resultados y análisis .....	34
3.1	Sistema móvil .....	34
3.2	Análisis de la estructura del sistema móvil .....	35
3.3	Análisis de la estructura de la estantería.....	37
3.4	Sistema embebido.....	38
3.5	Entrenamiento del modelo.....	38
3.6	Monitoreo de estanterías .....	41
3.7	Monitoreo en aplicativo móvil .....	43
3.8	Monitoreo en base de datos .....	45
3.9	Análisis de costos.....	47
CAPÍTULO 4.....		50

4.	Conclusiones y recomendaciones .....	50
4.1.	Conclusiones.....	50
4.2.	Recomendaciones.....	51
	BIBLIOGRAFÍA.....	52
	Apéndices.....	55

## ABREVIATURAS

IDE	Integrated Development Environment
DC	Direct Current
MP	Mega Pixels
RAM	Random Access Memory
GB	Gigabyte
APP	Application
G	Green
R	Red
PWM	Pulse Width Modulation
SSE	Streaming SIMD Extensions
MMX	Multimedia extension
RFID	Radio Frecuency Identifiacion
USD	United States Dollars
KB	Kilobyte
USART	Universal Synchronous/Asynchronous Receiver Transmitter
ADC	Analog-to-Digital Converter
VCC	Voltage COmmon Collector
VSS	Voltage Surce Supply

## SIMBOLOGÍA

mm	Milímetro
gr	Gramo
kg	Kilogramo
kgf	Kilogramo fuerza
kg·cm	Kilogramo centímetro
m	Metro
s	Segundo
h	Hora
N	Newton
cm	Centímetro
A	Amperio
mA	Miliamperio
V	Voltio
rad	Radián
Nm	Newton metro
RPM	Revoluciones por minuto
RPS	Revoluciones por segundo
mH	Milihenrio
$\Omega$	Ohmios
°C	Grados centígrados
kV	Kilovoltio
N	Newton
M Pa	Megapascal
m/s	Metro por segundo
m/s <sup>2</sup>	Metro por segundo al cuadrado

## ÍNDICE DE FIGURAS

Figura 1.1 Afluencia de usuarios en tienda comercial. ....	1
Figura 1.2 Carro inteligente de supermercado de Amazon. ....	3
Figura 1.3 Principales causas de desabastecimiento en supermercados. ....	5
Figura 1.4 Causas de insatisfacción del cliente en supermercados.....	7
Figura 1.5 Porcentaje de pérdidas monetarias por desabastecimiento en supermercados .....	8
Figura 1.6 Pérdidas monetarias en billones de USD por desabastecimiento en Norte América en supermercados.....	8
Figura 1.7 Planograma de una estantería. ....	12
Figura 1.8 Semejanza entre neurona y red neuronal .....	13
Figura 1.9 Estructura básica de una red neuronal.....	14
Figura 2.1 Celda de carga .....	17
Figura 2.2 Monitoreo de estantería por cámara.....	17
Figura 2.3 Lectura de un RFID .....	18
Figura 2.4 Flujograma de la metodología del proceso de diseño .....	20
Figura 2.5 Diseño conceptual del sistema de monitoreo. ....	22
Figura 2.6 Cámara móvil, montada sobre la caja metálica.....	23
Figura 2.7 Diseño del sistema móvil para la cámara de video.....	23
Figura 2.8 Diagrama esquemático del sistema de móvil. ....	25
Figura 2.9 Estantería con todos los productos ordenados adecuadamente. ....	27
Figura 2.10 Fotografías de cada producto tomada individualmente. ....	27
Figura 2.11 Videocámara Dragón Touch 4K .....	28
Figura 2.12 Tarjeta gráfica NVIDIA GTX 1080 .....	29
Figura 2.13 Plataformas de programación utilizadas .....	30
Figura 2.14 Arduino Mega .....	31
Figura 2.15 Aplicativo móvil de estado de estantería. ....	32
Figura 2.16 Estructura de la base de datos de la estantería. ....	33
Figura 3.1 Ubicación de los puntos de control en la plataforma. ....	35
Figura 3.2 Análisis de elementos finitos de la plataforma móvil. ....	36
Figura 3.3 Análisis de elementos finitos de la plataforma del riel móvil. ....	36
Figura 3.4 Análisis de elementos finitos de estanterías.....	37

Figura 3.5 Placa PCB del circuito electrónico.....	38
Figura 3.6 Resultado de entrenamiento del modelo .....	39
Figura 3.7 Frame de video de estantería completa .....	42
Figura 3.8 Frame video estantería con error .....	43
Figura 3.9 Estado de estanterías en aplicativo móvil con productos correctamente abastecidos y ubicados. ....	44
Figura 3.10 Estado de estanterías en aplicativo móvil con productos incorrectamente abastecidos y ubicados. ....	45
Figura 3.11 Estado de base de datos con productos correctamente abastecidos y ubicados. ....	46
Figura 3.12 Estado de base de datos con productos incorrectamente abastecidos y ubicados. ....	47

## ÍNDICE DE TABLAS

Tabla 1.1. Información de principales líneas de supermercados del Ecuador.....	2
Tabla 2.1 Matriz de decisión para el sistema de monitoreo.....	19
Tabla 3.1 Puntos de control en el recorrido del sistemaa móvil.....	35
Tabla 3.2 Costos profesionales. ....	48
Tabla 3.3 Costos mecánicos. ....	48
Tabla 3.4 Costos electrónicos. ....	49

## ÍNDICE DE PLANOS

PLANO 1	Diseño de conexiones de la placa PCB
PLANO 2	Diagrama esquemático del circuito electrónico
PLANO 3	Diseño de la estantería
PLANO 4	Diseño de la plataforma del riel.
PLANO 5	Diseño de la plataforma base.
PLANO 6	Diseño de la caja.

# CAPÍTULO 1

## 1. INTRODUCCIÓN

Las transacciones comerciales se remontan hace más de diez mil años, siendo en un principio un intercambio de productos conocido como trueque. Este modelo de negocio evolucionó con los años transformándose en una forma de comercio de compraventa, en la cual una persona o empresa ofrece un producto o servicio a cambio de una remuneración económica [1].

La compra y venta de productos de forma personal es una de las transacciones más empleadas a nivel mundial. En los países desarrollados este negocio se lleva a cabo de diversas maneras, ya sea personal, a distancia o multinivel, siendo cualquiera de estos un sistema ágil y seguro [2].

En países menos desarrollados, las ventas a distancia siguen en progreso, es decir, se van incorporando de a poco, resultando así que la gran parte de estas transacciones se realizan de forma personal, lo que se puede traducir como un mayor tránsito de personas en los locales comerciales, esto se puede observar en la Figura 1.1.



**Figura 1.1 Afluencia de usuarios en tienda comercial [3].**

Ecuador cuenta con grandes líneas de supermercados, encargadas de abastecer a la población ecuatoriana con productos de primera, segunda y tercera necesidad. Estas empresas tienen varios locales comerciales distribuidos a lo largo del país, destacando la Corporación Favorita C.A, El Rosado S.A, Tía S.A, Santamaría S.A y Gerardo Ortiz, teniendo en el 2016 la cantidad de locales mostrada en la Tabla 1.1, obteniendo un total de 386 establecimientos [4].

**Tabla 1.1. Información de principales líneas de supermercados del Ecuador [4].**

<b>Supermercados</b>	<b>Establecimientos</b>	<b>No. de locales</b>
Corporación Favorita C.A.	Supermaxi, Megamaxi, Akí, Gran Akí y Super Akí	113
Corporación El Rosado S.A.	Mi Comisariato, Ferrisariato, Mi Juguetería, Río Store y Supercines	64
TIA S.A.	TIA, Súper TIA, TIA Express y Multiahorro	179
Mega Santamaría	Santa María y Mega Santa María	21
Gerardo Ortiz e Hijos Cía. Ltda.	Coral Hipermercados	9

Las empresas de los locales comerciales tienen a su disposición una gran cantidad de trabajadores encargados de ayudar al cliente en su compra, a su vez cuentan con otros operadores que tienen como responsabilidad el reabastecimiento de las estanterías del local o en algunos casos es el vendedor el encargado de ambas tareas. Sin embargo, que el vendedor esté supervisando las estanterías a cada momento genera desperdicio de tiempo que puede ser aprovechado para la atención al cliente [5].

Por otro lado, frecuentemente algunos productos de las estanterías se encuentran desabastecidos provocando pérdidas económicas a la empresa [6]. Esto debido a que el cliente no encuentra el producto deseado, impidiendo finalizar la venta o a su vez prolongándola, causando insatisfacción en su compra [5].

El desabastecimiento de las estanterías es un inconveniente que sufren todos los locales comerciales y para ello se debe buscar una solución vanguardista. Por este motivo se propone diseñar un sistema de monitoreo automático utilizando distintas herramientas

de programación y otros mecanismos que permitan agilizar el proceso de abastecimiento optimizando el servicio, como se muestra en la Figura 1.2, el cual utiliza carros inteligentes para mejorar el proceso de pago.



**Figura 1.2 Carro inteligente de supermercado de Amazon [7].**

El presente trabajo se encuentra dividido en cuatro capítulos. El capítulo 1 presenta el marco teórico, donde se tratan los temas necesarios para entender la propuesta de solución al problema. En el capítulo 2 se desarrolla la metodología, en la que se detalla los procedimientos a seguir de la propuesta, indicando paso a paso las acciones a tomar. El capítulo 3 muestra los resultados y análisis respectivos, comentando las diferentes respuestas para distintos escenarios. Finalmente, se concluye en base a las salidas del sistema y a su vez se mencionan las respectivas recomendaciones.

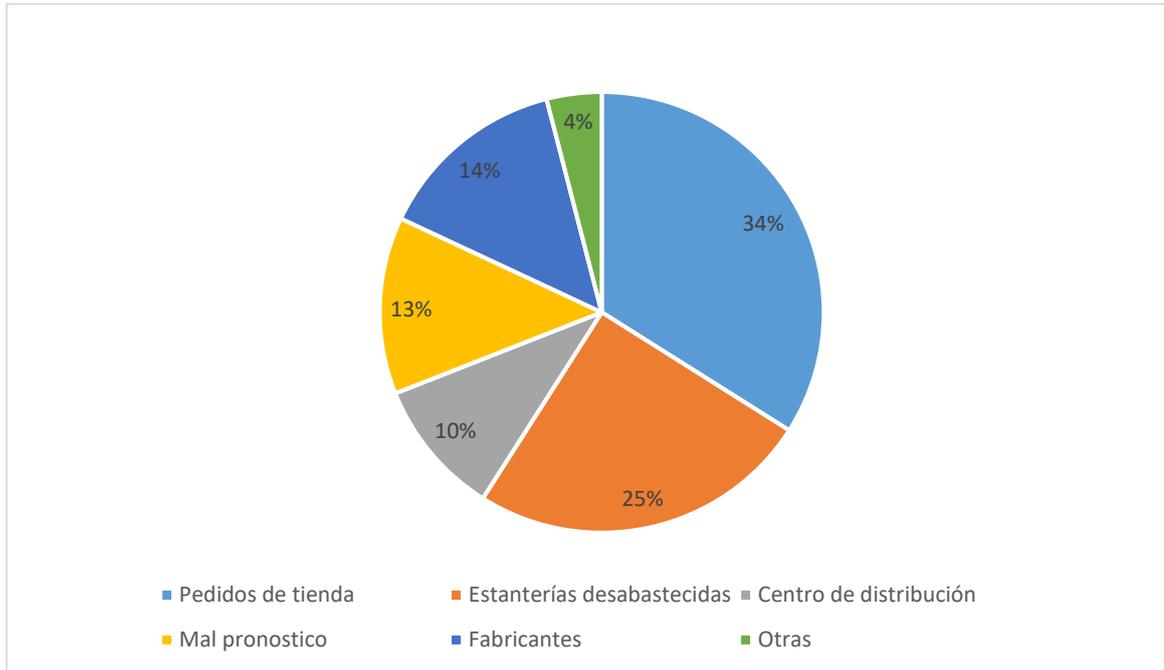
## **1.1 Descripción del problema**

Los locales de venta de productos de primera necesidad como los supermercados tienen que lidiar constantemente con eventualidades, como aquellas en las que un producto se agota de las estanterías o muchas veces se encuentra fuera de su sitio.

Este desorden generalmente es provocado por los mismos clientes, ya sea voluntariamente o por equivocación. Esto causa confusión a la hora de hacer compras, y al mismo tiempo interfiere con las estrategias de marketing empleadas por una empresa de este estilo, ya que la ubicación de los productos se realiza de forma estratégica para aumentar los beneficios de la misma.

En cuanto al desabastecimiento, según un estudio realizado por la Universidad Emory de Atlanta, en conjunto con la universidad de St. Gallen en Suiza y la Universidad de Colorado, se encontró que alrededor de un cuarto de los casos en los que se reporta una estantería vacía, el producto si se encuentre disponible en bodega (Figura 1.3) [6].

Este inconveniente suele ser provocado por un aumento repentino de la demanda de uno o varios productos. Dicha tendencia de compra ocasiona que la velocidad de adquisición de los productos en stock aumente de forma inesperada [8]. Como resultado de esta acción se tienen estanterías vacías durante el tiempo que transcurre hasta que los encargados noten este desabastecimiento, para luego notificarlo y finalmente reponer el producto o los productos faltantes.



**Figura 1.3 Principales causas de desabastecimiento en supermercados [6].**

Lo que se hace en la actualidad para lidiar con estos inconvenientes, es emplear personal para recorrer las estanterías, una a una, fijándose en la disponibilidad y orden adecuado de los productos. Esta es una tarea que resulta extensa al considerar un local con una gran cantidad de estanterías como en el caso de grandes supermercados. Adicionalmente, esta tarea debe ser realizada varias veces a lo largo del día, lo que implica que inevitablemente se efectúe mientras el cliente realiza sus compras, entorpeciendo así el tránsito en los pasillos del local [9].

A este aspecto se le suma el error humano al que está sujeto la detección de objetos faltantes y mal ubicados, lo cual puede costar tiempo y dinero. Es por esto que en la actualidad se están buscando tecnologías que permitan cubrir estas falencias y asistir en la optimización del servicio brindado.

## **1.2 Justificación del problema**

La monitorización automática del abastecimiento y orden de productos en stock en empresas de venta minorista, tales como supermercados, farmacias, jugueterías, entre otras, resulta muy oportuna y necesaria. Esto se traduce en beneficios económicos y de optimización en el manejo de los recursos de la empresa, más específicamente tiempo y personal, a la vez que aumenta la eficiencia del servicio brindado a los clientes.

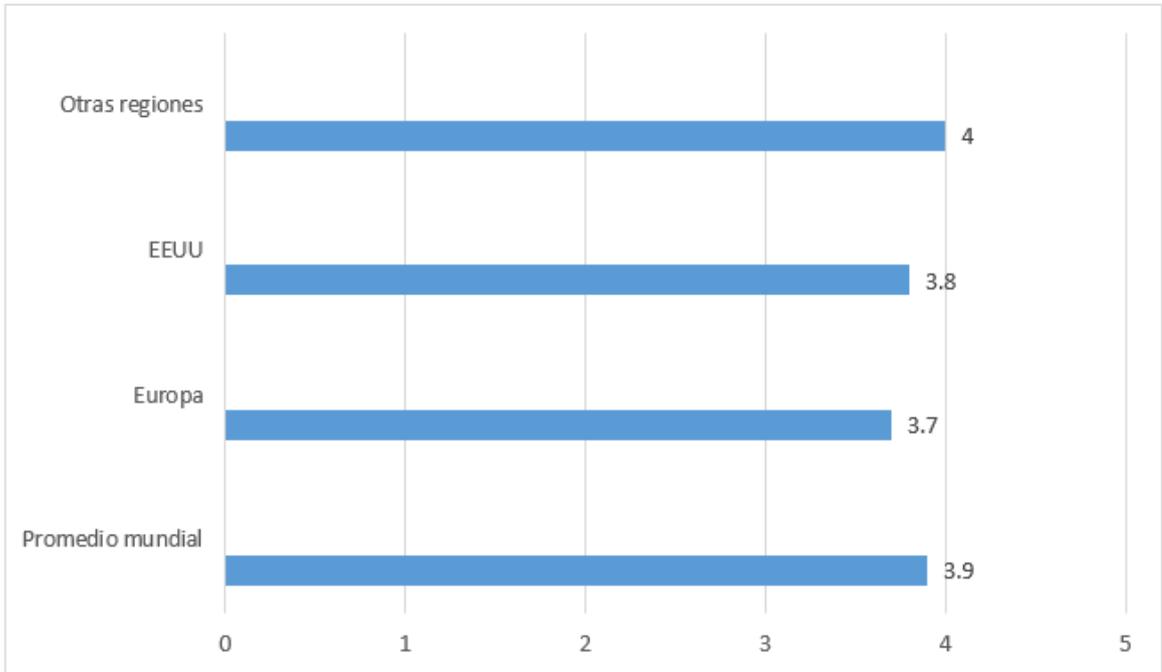
Para los consumidores, es muy importante poder realizar una compra rápida y encontrar los artículos que buscan. Es por esto que, con frecuencia, experimentan cierta frustración al buscar un determinado producto y encontrar la estantería vacía [6].

De acuerdo a una encuesta realizada en 2018 por IHL Group, se determinó que los usuarios se encuentran con un producto desabastecido 1 de cada 5 veces que acuden a un local de venta de alimentos o medicinas. Según este mismo estudio el 32% de las veces que un cliente sale insatisfecho de un local de venta de productos, se debe a una estantería vacía, y un 16% de las veces debido a que no encontraron personal que le ayude a encontrar un determinado producto (Figura 1.4). Esto ocasiona pérdida de lealtad de los clientes a dichos locales y marcas [10].



**Figura 1.4 Causas de insatisfacción del cliente en supermercados [10]**

Estos inconvenientes a su vez afectan negativamente las ganancias de la empresa. La Figura 1.5 muestra los porcentajes de pérdidas correspondientes a diversas regiones del mundo. Solo en Norteamérica se estiman pérdidas que alcanzan los 144.9 billones de dólares en la industria de supermercados, de los cuales, \$47.4 billones se deben a estanterías vacías [6], como indica la Figura 1.6.



**Figura 1.5 Porcentaje de pérdidas monetarias por desabastecimiento en supermercados [6]**



**Figura 1.6 Pérdidas monetarias en billones de USD por desabastecimiento en Norte América en supermercados [10]**

La instalación de este sistema de monitoreo automático de estanterías de productos no solo resultará en un beneficio monetario para los propietarios de las empresas que adquieran esta tecnología. Hacer uso de un sistema automático también creará un ambiente más amigable para los clientes a la hora de realizar sus compras, aumentando así su satisfacción, debido a que el personal de la empresa estará a la disponibilidad del cliente para ayudar o asesorar en su compra.

## **1.3 Objetivos**

### **1.3.1 Objetivo general**

Diseñar un sistema de monitoreo automático de abastecimiento y organización de productos que agilice y optimice el proceso de reposición de mercadería en las estanterías de la empresa.

### **1.3.2 Objetivos específicos**

- Establecer el diseño y los mecanismos de monitoreo automático de las estanterías para el control del desplazamiento del sistema a lo largo de los pasillos.
- Simular el funcionamiento del sistema embebido y del software para la comprobación de su aplicación en el desabastecimiento de estanterías.
- Diseñar un mecanismo capaz de desplazar el sistema embebido en el espacio de trabajo.
- Desarrollar el modelo 3D del sistema de monitoreo automático.
- Implementar un software de detección de objetos mediante una cámara que permita identificar los productos existentes en un supermercado, para así determinar su correcta posición y abastecimiento.
- Diseñar conexiones eléctricas y electrónicas del sistema.

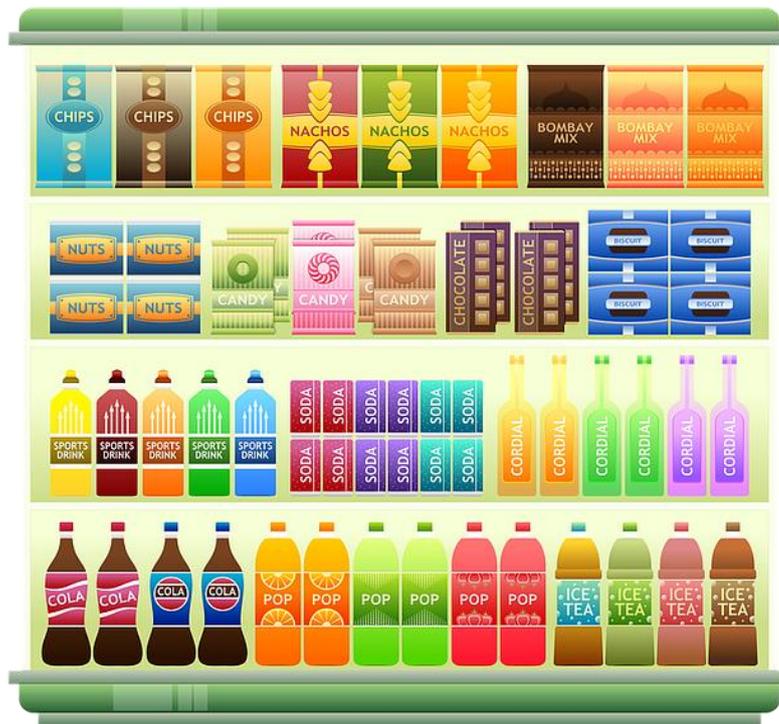
## **1.4 Marco teórico**

A fin de tener un óptimo rendimiento del negocio, los supermercados cuentan con una serie de procesos clave. Entre ellos está el correcto abastecimiento de los productos en las estanterías de supermercados, a cargo de empleados denominados reponedores. Estos cuentan con varias responsabilidades, siendo la principal rellenar las estanterías con nuevos suministros. Esto incluye la labor de revisar las estanterías, en busca de productos faltantes o mal ubicados [11]. Para lo cual, estos empleados se basan en planogramas establecidos por la empresa de forma estratégica.

### **1.4.1 Planograma**

Un planograma (Figura 1.7) es un método visual utilizado para categorizar y ubicar productos en una estantería ejecutando un análisis de mercado, donde se da a conocer la forma en la que estos productos deben ser presentados en base a sus características, con el fin de obtener un mejor alcance de ventas [12].

Es importante para una empresa de venta de productos, ya que ayuda a tener un control óptimo del inventario, como también cuidar que los productos estén correctamente ubicados, facilitando la salida de la mercadería y logrando una satisfacción del cliente. Por otra parte, facilita el reabastecimiento de los artículos, lo que implica una optimización de recursos y tiempo de los trabajadores [13].



**Figura 1.7 Planograma de una estantería [14].**

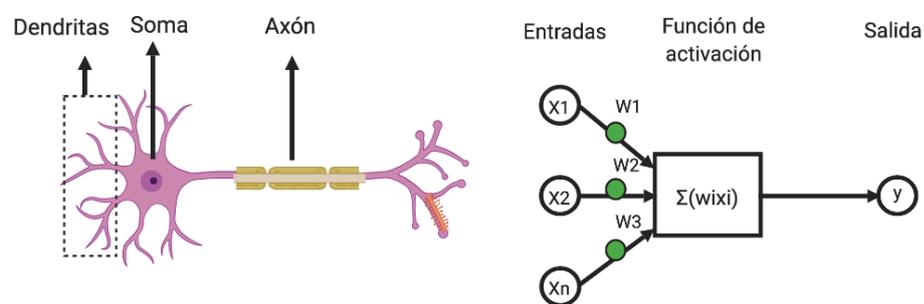
Al tener estos planogramas establecidos, se puede almacenar su estructura en una base de datos, y de esta manera poder usarlo como referencia para así automatizar el proceso de monitoreo de estanterías, prescindiendo de personal humano. Para ello, sería necesario contar con un programa con la capacidad de reconocer los productos de la estantería por medio de una cámara de video, para así comparar lo observado con el planograma almacenado.

Existen una serie de herramientas que permiten hacer esto posible, entre las principales tenemos redes neuronales y los algoritmos de reconocimiento de objetos.

#### **1.4.2 Redes neuronales**

Las redes neuronales se las puede definir como una forma de digitalización de las capacidades de aprender y memorizar del cerebro humano, haciendo que el hombre sea capaz de adquirir experiencia de distintos eventos. Llevando a nombrar este tipo de procesamiento de información computacional como red

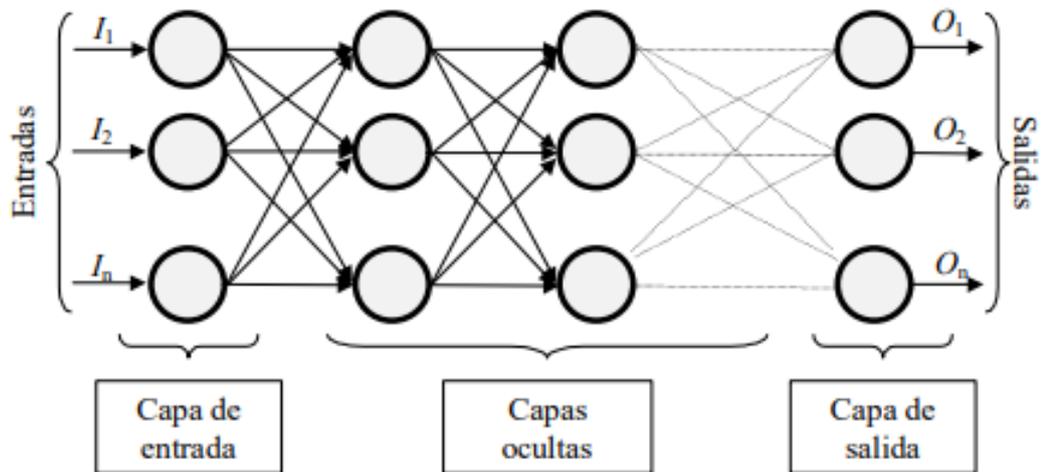
neuronal en referencia de la neurona humana, debido al funcionamiento similar que poseen (Figura 1.8 Semejenaza entre neurona y red neuronal [15]). Este comportamiento semejante se lo puede observar en la función de una neurona humana, cuando recibe una señal del cerebro (entrada), teniendo luego de esto un tratamiento de esta información recibida para posteriormente ser enviada y ejecutar la acción deseada (salida) [15].



**Figura 1.8 Semejenaza entre neurona y red neuronal [15].**

Las redes neuronales son capaces de ejecutar acciones sin necesidad de tener un algoritmo, gracias al aprendizaje que obtienen; si la información recibida se encuentra incompleta, poco clara o no antes vista, esta entra en un proceso de generalización de la entrada concluyendo así con un resultado [16].

Una red neuronal cuenta con tres capas principales: entrada, oculta y salida (Figura 1.9), cada una cumpliendo funciones específicas, conectadas todas entre sí. La capa de entrada, como el nombre lo indica, son las entradas de señal, la cual se hace un solo vector para luego ser procesada; en las capas ocultas se tiene la función de activación, encargada de hacer el procesamiento de la información de la capa entrada; por último, en la capa de salida se obtiene la conclusión de la red neuronal, teniendo una salida con un rango o también de forma binaria, dependiendo el caso [15].



**Figura 1.9 Estructura básica de una red neuronal [15].**

Se tiene dos formas principales de aprendizaje: *machine learning* y *deep learning*. La primera se caracteriza por tener un sistema de aprendizaje haciendo uso de un maestro, el cual indica la salida que debe tener la red en base a la entrada, en el caso que esta sea incorrecta, este maestro modifica los pesos de la función de activación, buscando obtener la salida deseada. El deep learning no hace uso de un maestro que cambie los pesos, más bien la misma red neuronal busca características, relaciones, categorías, etc, que sean necesarios como valores de entrada, quien luego establecerá las distintas probabilidades de salidas [16].

### 1.4.3 Reconocimiento de objetos

El reconocimiento de objetos es un método de visión artificial para identificar objetos en imágenes o vídeos. Para ello se emplean técnicas de deep learning tales como las redes neuronales convolucionales.

Se emplean dos formas principales: se puede generar un modelo desde cero o, para ahorrar tiempo, usar un modelo ya entrenado con un gran número de imágenes [17].

Existen librerías de código abierto que facilitan herramientas para desarrollar software de reconocimiento de imágenes para diversas aplicaciones. Entre las principales encontramos TensorFlow y OpenCV.

#### **1.4.4.1. TensorFlow**

TensorFlow es una biblioteca de software de código abierto para computación numérica, que usa gráficos de flujo de datos. Los nodos en las gráficas representan operaciones matemáticas, mientras que los bordes de las gráficas representan las matrices de datos multidimensionales (tensores) comunicadas entre ellos. Es una plataforma ideal para trabajar con redes neuronales [18].

Fue creado inicialmente por investigadores e ingenieros de Google Brain Team, con el fin de investigar en el campo del aprendizaje automático y las redes neuronales [18]. Sin embargo, gracias a su generalidad, permite abarcar una gran variedad de aplicaciones, tales como mejorar fotografías en *smartphone*, ayudar en diagnóstico médico y procesamiento de imágenes [18].

#### **1.4.4.2. OpenCV**

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión artificial de código abierto. Se creó para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de la máquina en los productos comerciales [19].

OpenCV se inclina principalmente hacia aplicaciones de visión en tiempo real y aprovecha las instrucciones MMX y SSE cuando están disponibles [19]. La biblioteca tiene más de 2500 algoritmos optimizados que se pueden utilizar para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, etc. [19].

# CAPÍTULO 2

## 2. METODOLOGÍA

La implementación de este proyecto se llevó a cabo haciendo uso de una metodología que satisfaga las necesidades del sistema de monitoreo, contemplando los objetivos antes mencionados en el presente documento (sección 1.3). Por ello se establecieron varias alternativas de solución, escogiendo la más relevante, a su vez se determinaron los requerimientos del proyecto, seguido de todos los elementos esenciales del desarrollo del diseño de la solución.

### 2.1 Alternativas de solución

Para realizar un monitoreo óptimo de la estantería se consideró tres alternativas, según su tipo de obtención de datos, lo cual repercute en la eficiencia del sistema, siendo este una de las principales características a tomar en cuenta a la hora de diseñar una estantería inteligente. Por otro lado, un factor importante a considerar son los costos que conlleva implementar y mantener este sistema.

Las tres alternativas a considerar se las detalla a continuación:

- **Balanza en la plataforma:** mide el peso sobre una plataforma mediante el uso de celda de cargas, este sensor devuelve un valor analógico en base a la fuerza aplicada sobre la superficie en la que fue instalada [20] como se puede visualizar en la Figura 2.1. De esta manera se puede determinar la cantidad de mercadería que se encuentra en la estantería.

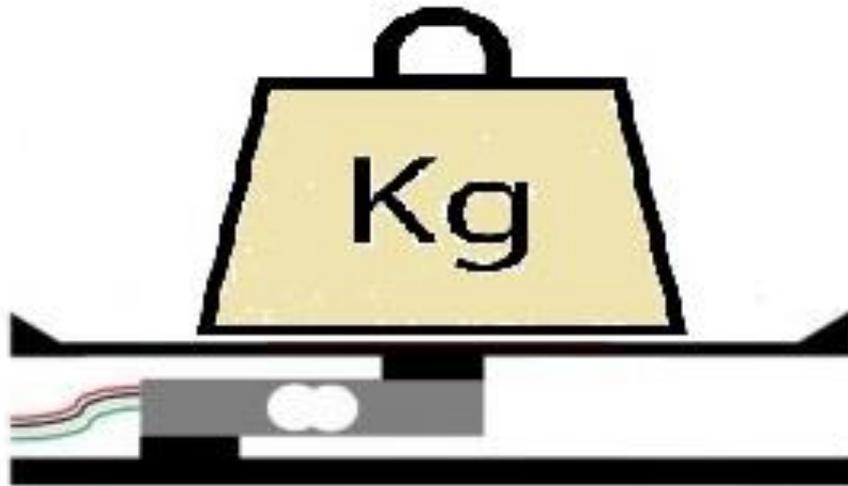


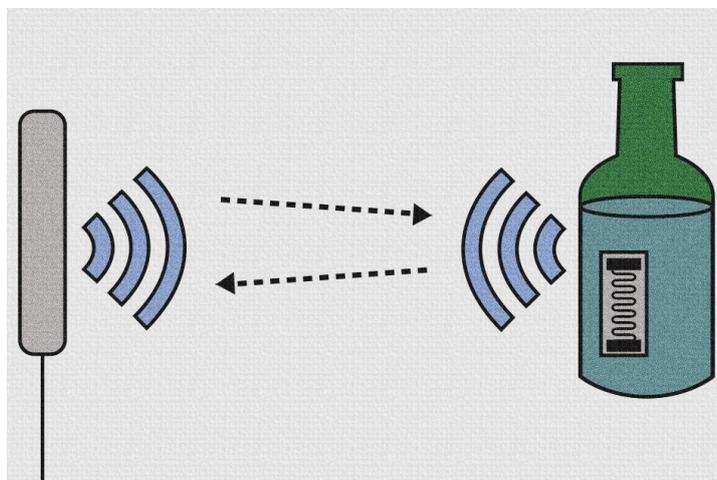
Figura 2.1 Celda de carga [21]

- **Monitoreo por cámaras:** se realiza un monitoreo en tiempo real a través de una cámara de alta resolución (Figura 2.2), este dispositivo analizará la estantería a través de software, dando como resultado la notificación de la ausencia de mercadería o de un producto mal ubicado.



Figura 2.2 Monitoreo de estantería por cámara [22]

- **Monitoreo por sensores:** haciendo uso de sensores RFID (identificador por radio frecuencia) los cuales al estar cerca del receptor establecen comunicación entre ellos [23], como se encuentra ejemplificado en la Figura 2.3, teniendo así en este caso la facultad de determinar la cantidad de mercadería existente en las estanterías y llevar un mejor control del producto en exhibición, ya que, este identificador de radio frecuencia guarda información específica.



**Figura 2.3 Lectura de un RFID [24]**

En base al proyecto realizado se elaboró una matriz de decisión (Tabla 2.1), con cinco factores más relevantes a tomar en consideración para realizar una selección correcta de la alternativa a seguir, estos criterios se los detalla a continuación:

- Costo inicial: costo que requiere la implementación y funcionamiento del sistema.
- Desempeño: la precisión y el número de datos que se puede obtener.
- Facilidad de instalación: cambios que se requieren hacer en las estanterías existentes para la implementación del sistema.
- Interacción con operadores: recursos de datos que recibe el operador respecto al monitoreo de las estanterías.
- Costo de mantenimiento: costos de revisión y arreglo de sensores y mecanismos.

**Tabla 2.1 Matriz de decisión para el sistema de monitoreo.**

	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5	
Descripción	Costo inicial	Desempeño	Facilidad de instalación	Interacción con operadores	Costo de mantenimiento	
	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5	Total
Peso	3	4	2	3	2	14
	21%	29%	14%	21%	14%	100%
OPTIONS	Calificación	Calificación	Calificación	Calificación	Calificación	
Balanza en la plataforma	5	7	4	6	4	5.5
Monitoreo por cámara	7	8	7	8	7	7.5
Monitoreo por sensores	4	9	4	8	6	6.6

En base al resultado obtenido en la matriz de decisión, se determinó que el mejor sistema de monitoreo es aquel que utiliza una cámara, debido a que el costo inicial de instalación es el más bajo, dado que se necesita únicamente una cámara y un mecanismo que mueva la misma. Por otro lado, con la balanza en la plataforma se requiere modificar completamente las estanterías, y la alternativa del sistema de sensores conlleva colocar un RFID en cada producto, lo cual representa un costo inicial y de mantenimiento más elevado, como también una instalación compleja. Por otro lado, tanto el desempeño como la interacción con operadores se determinó que se encontraban muy similares, predominando por muy poco puntaje el sensor RFID, sin embargo, los costos de instalación de estos no representan un gran impacto en comparación al monitoreo por cámara, concluyendo que la segunda alternativa es la más eficaz.

## 2.2 Proceso de diseño

El proceso de diseño que se siguió se puede dividir en cuatro fases, cada una de las cuales comprende varios pasos, estructurados en base a los requerimientos de diseño. Las fases son las siguientes:

- Elaboración del software de reconocimiento de objetos
- Diseño mecánico del sistema móvil
- Diseño y programación del sistema embebido a cargo de la movilidad del sistema
- Diseño de interfaz para usuarios

Los detalles de cada fase se pueden ver a detalle en la Figura 2.4. En color verde se pueden ver los pasos de la fase de software, en rosado la fase de diseño mecánico, en azul, la de diseño del sistema embebido, y por último en amarillo la fase de diseño de interfaz de usuarios.

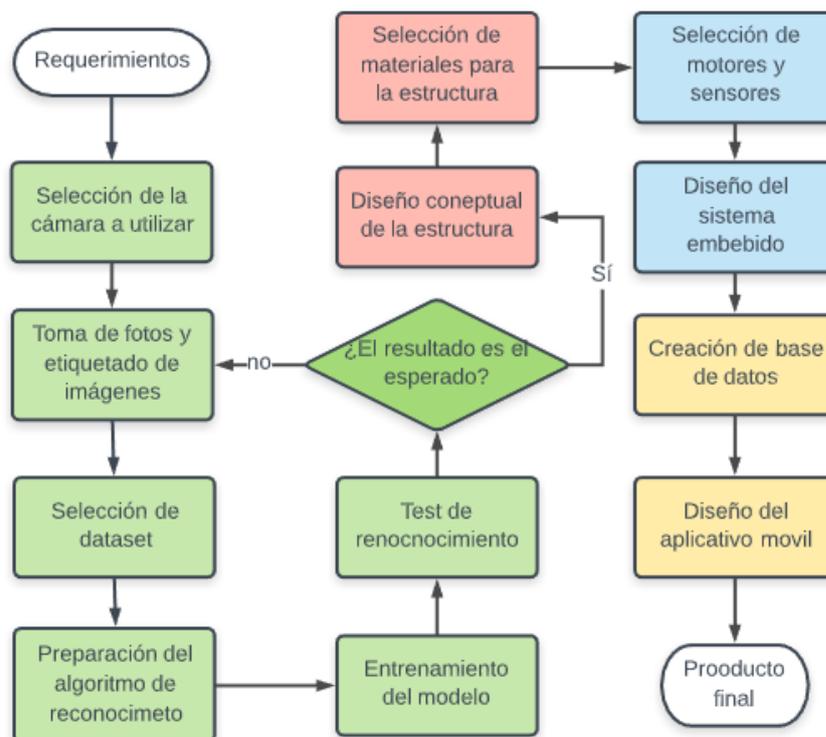


Figura 2.4 Flujograma de la metodología del proceso de diseño

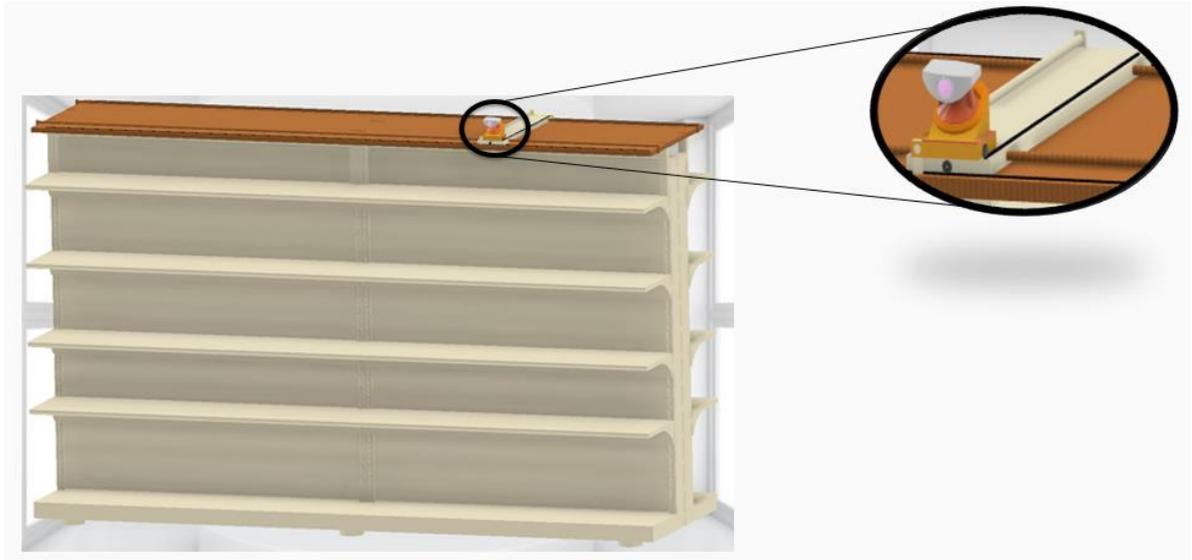
### **2.3 Requerimientos de diseño**

En base a una entrevista realizada a un cajero de una de las principales cadenas de supermercados del Ecuador [5], se recopilaron una serie de requerimientos que surgieron a partir de la propuesta del sistema de estanterías inteligentes. Estos se resumen a continuación:

- Flexibilidad de instalación, es decir, que tan adaptable es el sistema a diferentes tipos de estanterías con dimensiones distintas.
- Confiabilidad de los resultados arrojados, haciendo referencia al nivel de precisión.
- Velocidad de procesamiento de datos
- Frecuencia de mantenimiento que el sistema requiere
- Resistencia al polvo
- Comunicación: por cable
- Posibilidad de monitoreo en tiempo real

### **2.4 Diseño conceptual**

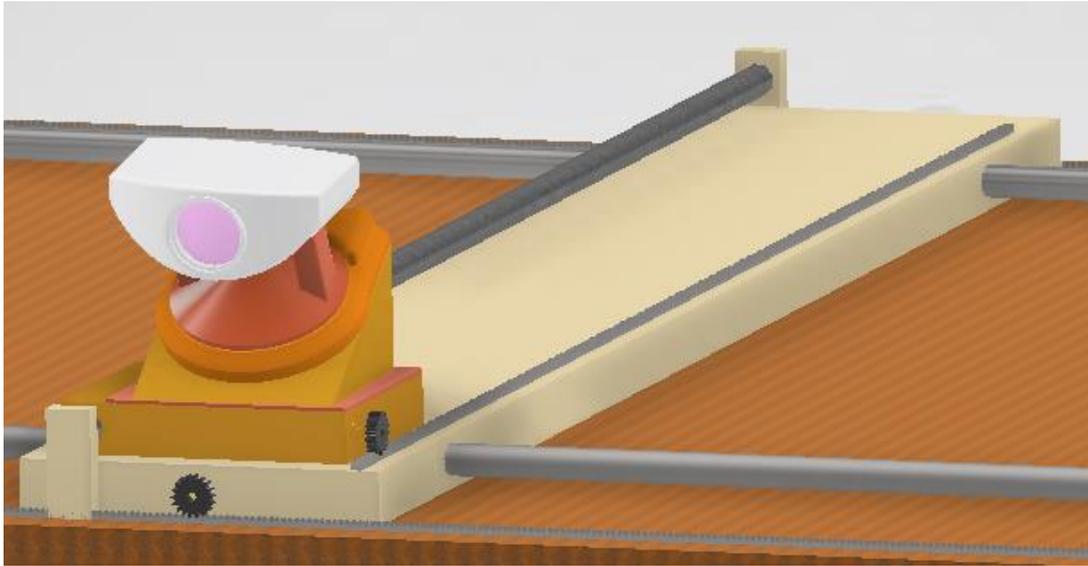
El diseño del sistema está dividido en las cuatro partes previamente mencionadas. El dispositivo consistió en una estructura móvil equipada con una cámara de video. Este mecanismo móvil se diseñó para ir montado por encima de cada par de estanterías contiguas opuestas, como se puede evidenciar en la Figura 2.5. La movilidad del sistema es controlada por un sistema embebido en conjunto con sensores y actuadores. De esta forma la cámara cuenta con la posibilidad de monitorear las estanterías en cuatro distintas direcciones, y mediante el software de reconocimiento de datos, detecta eventualidades concernientes a los productos y las reporta al personal correspondiente mediante un aplicativo móvil.



**Figura 2.5 Diseño conceptual del sistema de monitoreo.**

## **2.5 Diseño del sistema móvil**

El sistema mecánico móvil consistió en una caja metálica (Apéndice D) sobre la cual está montada la cámara (Figura 2.6 Cámara móvil, montada sobre la caja metálica). Dentro de esta caja se encuentra uno de los motores DC, que le da movilidad mediante un sistema piñón cremallera. El sistema se mueve a lo largo de una plataforma con un riel para lograr el desplazamiento de la cámara, el cual tiene una medida de un metro (aproximadamente) con un tubo guía de acero paralelo al riel, que atraviesa la caja lateralmente. El riel tiene una cremallera a lo largo de su extensión, que sirve para acoplar el piñón y permitir el desplazamiento de la caja. Para soportar el peso de la caja y la cámara, el sistema cuenta con ruedas ubicadas en la parte inferior.



**Figura 2.6 Cámara móvil, montada sobre la caja metálica**

A su vez, este riel se mueve a lo largo y por encima de otros dos rieles perpendiculares al mismo, de 4 metros cada uno, dotados también de cremalleras, ruedas y varillas guía. Se tienen dos motores DC, uno a cada lado del riel vertical, que le darán movilidad a este a través de las cremalleras de los rieles horizontales. Estos se ubican uno a cada extremo del riel vertical, de tal manera que el sistema actúa como una mesa XY como se muestra en la Figura 2.7 Diseño del sistema móvil para la cámara de video. De esta manera la caja puede realizar un recorrido rectangular por encima de la estantería, que le permite a la cámara tener cobertura total desde las cuatro caras de esta. Para ello la cámara viene acoplada sobre un motor de paso que cambiará la dirección en la que apunta la cámara cada vez que su recorrido cambia de dirección.



**Figura 2.7 Diseño del sistema móvil para la cámara de video**

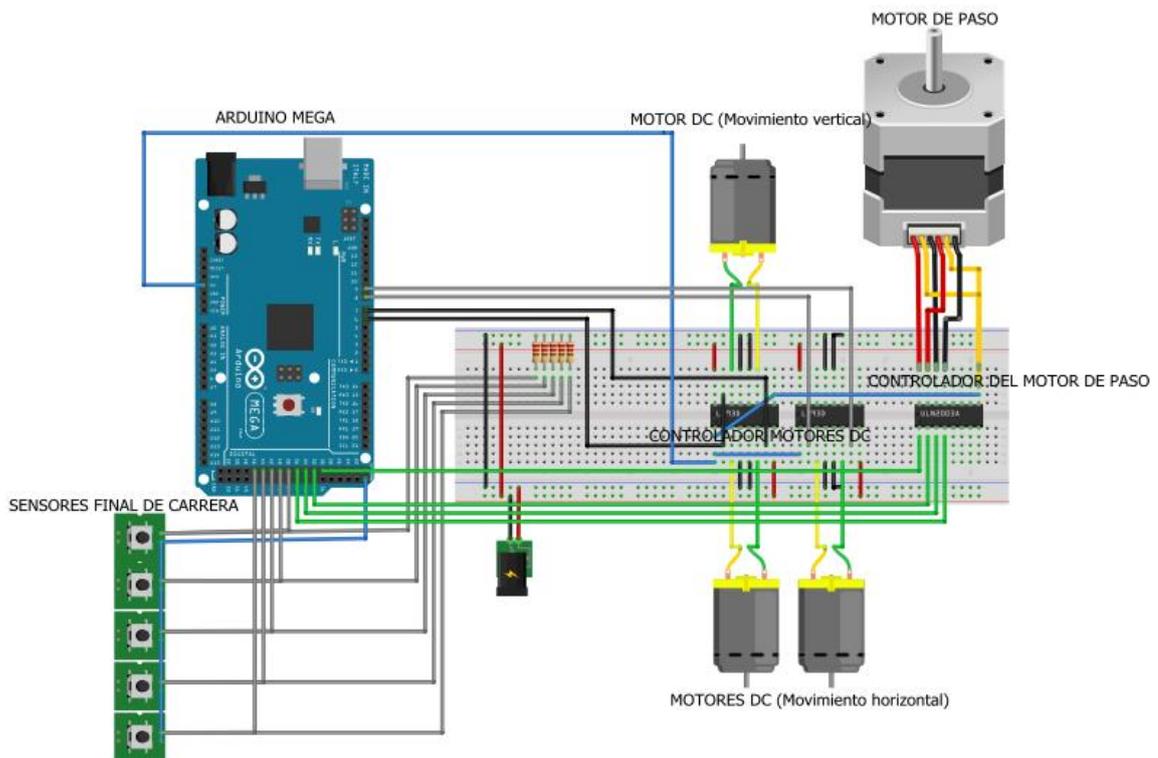
## **2.6 Diseño del sistema embebido**

Tanto los motores de los rieles como el motor de paso de la cámara son controlados por un microcontrolador Atmega2560. Los motores DC funcionan cada uno en conjunto con un integrado L293D, el cual sirve para controlar el sentido de giro de los motores. El movimiento del motor de paso se controla mediante un integrado ULN2003A.

Para este prototipo se tiene sensores finales de carrera de tipo microswitch (ver figura A.3 del Apéndice A, ubicados en posiciones pertinentes que en adelante se denominarán puntos de control. Se ubicó un punto de control en cada extremo del riel vertical y del riel horizontal inferior. Además, se colocó otro adicional en la mitad del riel horizontal inferior. Se tuvieron entonces 5 puntos de control, cada uno con su respectivo sensor final de carrera, que le permiten al sistema saber cuándo cambiar de dirección y cuando rotar la cámara, para así controlar su recorrido.

La programación del Arduino se realizó en la plataforma Arduino IDE y se utilizó una programación por hilos usando la librería FreeRTOS, la cual permite ejecutar múltiples tareas a modo de hilos. De acuerdo a esto, se programaron dos tareas: la que controla el desplazamiento de la caja, y la que controla el giro de la cámara.

El esquema del prototipo inicial, que incluye un Arduino Mega, motores DC, motor de paso y sensores se muestra en la Figura 2.8 Diagrama esquemático del sistema de móvil.



**Figura 2.8 Diagrama esquemático del sistema de móvil.**

## 2.7 Selección de actuadores

Se escogió un motor ET-PGM16-061600 de 6 voltios para el eje vertical que mueve la caja con la cámara y el sistema embebido, con un torque nominal de 0.00237 Nm y velocidad nominal de 1300 rpm. Para el movimiento horizontal que desplaza la plataforma móvil superior se seleccionaron dos motores ET-PGM16-06355 de 6 voltios con torque nominal de 0.01221 Nm y velocidad de 275 rpm. Los tres motores seleccionados provienen del catálogo del fabricante Etonm Motor [25].

En cuanto al motor de paso, encargado del giro de la cámara, se seleccionó un motor de paso unipolar NEMA 11 modelo 20-02D-AMT112S debido a su control relativamente sencillo y preciso, lo que permite que la cámara pueda girar sobre su mismo eje. A demás se usó esta opción frente a motores de paso bipolares ya que no se requiere altos torques de operación. Este modelo opera con un voltaje de 5VDC, con un ángulo por paso de  $1.8^\circ$ , y un Torque máximo de 0.12Nm [26].

## **2.8 Diseño de las conexiones eléctricas**

Para las conexiones internas se diseñó un sistema electrónico, en la cual irán montados los circuitos integrados, el microcontrolador AtMega2560, y las resistencias correspondientes a cada uno de los 5 sensores finales de carrera. En este sistema se incluyen conectores de dos vías para los motores DC, un conector de 5 vías para la conexión de los sensores, y un conector de 6 vías para los cables correspondientes al motor de paso. El esquema completo se muestra en la Figura D.2 en la sección de Apéndice D.

## **2.9 Diseño de software de monitoreo de la estantería**

Se elaboró un dataset con 2266 imágenes de los 8 tipos productos a detectar (cereal, galletas, yogurt, granola, menestra, lavavajillas, leche y café), cada producto fotografiado individualmente (Figura 2.10), dentro de este dataset también se tienen fotos con todos los productos juntos y ordenados adecuadamente (Figura 2.9). Haciendo uso de inteligencia artificial, específicamente Deep learning, en la cual utilizando la librería pytorch, se realizó un entrenamiento de 200 épocas con una cantidad de 2039 fotos y 227 imágenes para su validación, para lograr un resultado exitoso se hizo uso de un modelo pre entrenado, el cual fue yolov3.weights [27].

Este sistema de monitoreo fue programado para detectar los productos faltantes, enviando esta información a una base de datos (2.14) para posteriormente dar una alerta a los operadores de dicho evento. Por otro lado, se estableció un reconocimiento de localización, determinando de esta manera si el producto se encuentra correctamente ubicado, en caso de que esta mercadería no esté en el lugar adecuado, enviará un aviso a los operadores de la misma manera que cuando surge un desabastecimiento.



Figura 2.9 Estantería con todos los productos ordenados adecuadamente.



Figura 2.10 Fotografías de cada producto tomada individualmente.

## 2.10 Selección de cámara de video

La resolución de la imagen es una de las prioridades en este sistema de reconocimiento de la estantería, es por ello que hacer una elección de cual dispositivo visual utilizar es una tarea imprescindible. Las fotografías fueron capturadas con un teléfono que cuenta con 12 megapíxeles de resolución, dada estas características se ha seleccionado una videocámara llamada en la tienda Amazon “Dragón Touch 4K Action Camera Touch Screen 16MP Vision 3 Pro PC Web Camera 100 feet Waterproof” [28], la cual cuenta como se detalla en el nombre con una resolución de 16 megapíxeles y un de video 4K. De esta manera se logra tener un monitoreo de alta calidad, haciendo posible la detección de todos los objetos de la estantería.



Figura 2.11 Videocámara Dragón Touch 4K [28]

## 2.11 Selección de la unidad de cómputo y plataforma de programación

Para el desarrollo del software de detección fue importante seleccionar las herramientas necesarias para ejecutar dichas actividades de manera eficiente, ya que, el uso de deep learning exige un computador con capacidades óptimas de memorias y tarjeta gráfica, como a su vez de programas pertinentes para ejecutar dichas tareas.

### 2.11.1 Unidad de cómputo

La unidad de cómputo utilizada para este sistema es una computadora portátil con sistema operativo Windows 10, una memoria RAM de 16gb y tarjeta gráfica NVIDIA GTX 1080. Siendo estas características necesarias para lograr un correcto funcionamiento del sistema de monitoreo dada las capacidades de ejecutar el entrenamiento y detección de manera óptima.



Figura 2.12 Tarjeta gráfica NVIDIA GTX 1080 [29].

### 2.11.2 Plataforma de programación

Para la elaboración de este sistema se hizo uso de tres plataformas de programación: Conda, Android Studio y Arduino IDE (Figura 2.13). El primer entorno de programación fue utilizado para realizar el entrenamiento, validación y monitoreo de estanterías; la segunda plataforma se utilizó para diseñar el aplicativo móvil, encargado de interactuar con el operador; por último, Arduino IDE fue necesario para realizar la programación del sistema embebido, y así lograr el movimiento de la plataforma móvil.

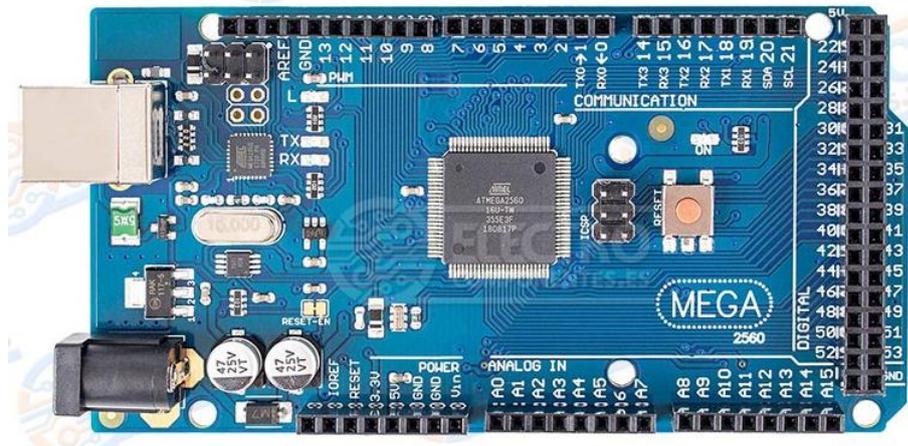


**Figura 2.13 Plataformas de programación utilizadas [30-32].**

## **2.12 Microcontrolador**

Para este prototipo se seleccionó un Arduino mega, el cual incorpora un microcontrolador Atmega 2560 (Figura 2.14). Este resulta óptimo para los propósitos de este trabajo ya que cuenta con la cantidad suficiente de pines digitales y analógicos utilizados para conectar los sensores y actuadores involucrados en el sistema embebido, como lo son los motores DC, el motor de pasos, y los finales de carrera.

A demás cuenta con cuatro puertos seriales que permiten la comunicación del sistema embebido con una computadora central.



**Figura 2.14** Arduino Mega[33]

### 2.13 Diseño de aplicación móvil

Para el diseño de la interfaz de usuario se utilizó el software Android Studio. En este se realizó una sencilla aplicación móvil en la cual se mostrará un planograma de forma gráfica en la que el usuario puede visualizar cada producto de una sección de estantería. Para este prototipo se mostrará una sección con dos filas, y 4 imágenes de productos en cada una de ellas, es decir 8 productos en total, como se puede observar en Figura 2.15. La aplicación está constantemente leyendo la información de la base de datos para saber el estado de los productos. Cada producto está rodeado por un marco de determinado color. Este color cambiará de acuerdo a si el producto se encuentra o no en su sitio correspondiente: se tornará rojo cuando exista una eventualidad a notificar y verde en caso contrario.

Además, la aplicación enviará una notificación cuando un producto pase en rojo por un tiempo superior a un minuto. De esta manera se evita enviar notificaciones por un falso positivo.



**Figura 2.15** Aplicativo móvil de estado de estantería.

## **2.14** Diseño de base de datos

La base de datos utilizada para este proyecto es una de tipo no relacional, elaborada en una plataforma de Google llamada Firebase (Figura 2.16), la cual otorga un monitoreo de la información en tiempo real, haciendo posible que, al hacer uso de este servicio, los operadores puedan monitorear la estantería de manera rápida y sencilla, a través de un aplicativo móvil, actualizando la información mostrada constantemente.

Para este prototipo, la información está estructurada de tal manera que se tienen dos filas y para cada una se tienen cuatro productos distintos, cuyos valores serán G o R, haciendo referencia a los dos estados en los que puede estar un producto: "Green" y "Red". Un producto con valor G indica que está en su lugar correcto, caso contrario su valor será R.

**planograma-8aa40-default-rtdb**



**Figura 2.16 Estructura de la base de datos de la estantería.**

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

En esta sección de resultados se detalló en primer lugar el diseño del sistema móvil, seguido con el análisis de elementos finitos del mismo, luego se realizó este mismo estudio para la estantería con todos los mecanismos. Seguido a estos resultados se dio a conocer el sistema embebido diseñado, el cual irá ubicado sobre las plataformas antes estudiadas. Habiendo logrado realizar el mecanismo móvil y su movimiento para colocar la cámara en los distintos puntos de monitoreo, se describe los valores obtenidos del entrenamiento de detección. Los siguientes tres temas tratan respecto a los resultados del monitoreo, teniendo al final del capítulo el análisis de costos.

### 3.1 Sistema móvil

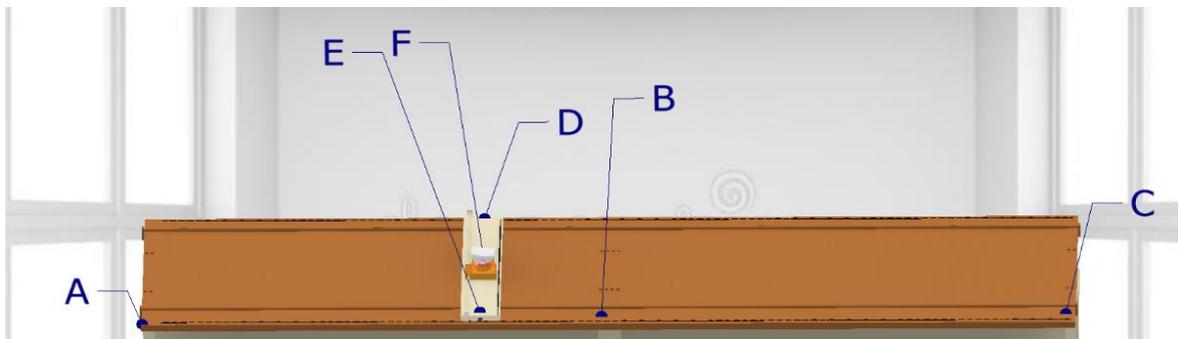
El sistema describe una trayectoria rectangular periódica con una detención de 5 segundos a la mitad de cada uno de los dos desplazamientos horizontales, con un desplazamiento total de 10 metros, 8 de movimiento horizontal y 2 de movimiento vertical, completando un recorrido completo en aproximadamente 1 minuto.

Durante el recorrido, los sensores finales de carrera son accionados a medida que el sistema se desplaza. El posicionamiento del sistema por sobre los puntos de control ubicados a lo largo del riel vertical y de uno de los rieles horizontales, permite tomar la correspondiente acción subsecuente.

La Tabla 3.1 describe las secuencias del recorrido y las acciones realizadas en cada punto de control. La ubicación de cada punto de control se muestra en la Figura 3.1.

**Tabla 3.1 Puntos de control en el recorrido del sistema móvil.**

Punto de Control	Acción
<b>E</b>	Accionamiento de motores hacia la derecha / Apagado de motor hacia abajo / Giro de la cámara.
<b>B</b>	Apagado de motor de desplazamiento horizontal durante 5 segundos.
<b>C</b>	Accionamiento de motor hacia arriba / Apagado de motores hacia la derecha / Giro de la cámara.
<b>D</b>	Accionamiento de motores hacia la izquierda / Apagado de motor hacia arriba / Giro de la cámara.
<b>A</b>	Accionamiento de motor hacia abajo / Apagado de motor hacia la izquierda / Giro de la cámara.

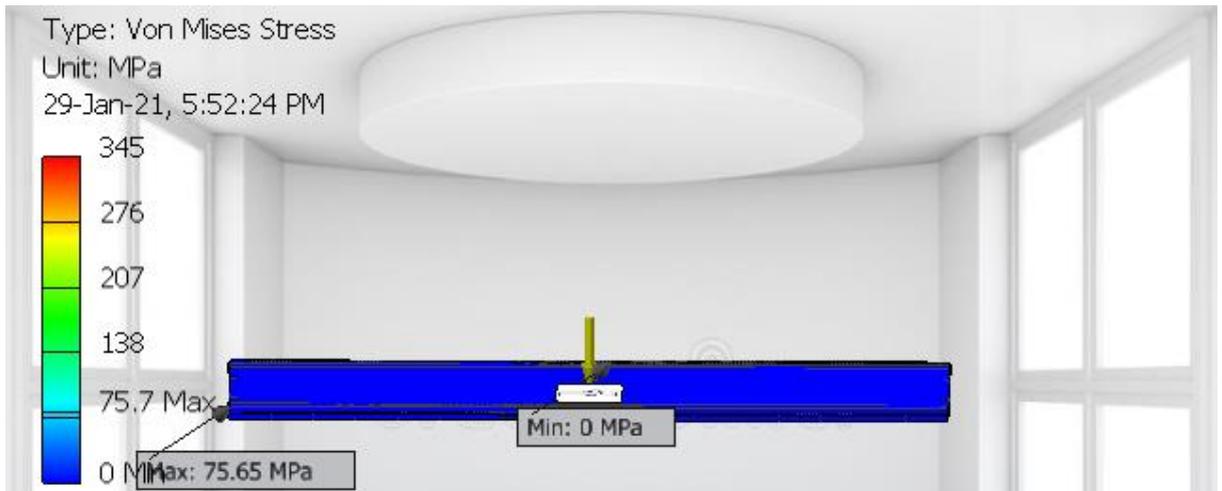


**Figura 3.1 Ubicación de los puntos de control en la plataforma.**

### 3.2 Análisis de la estructura del sistema móvil

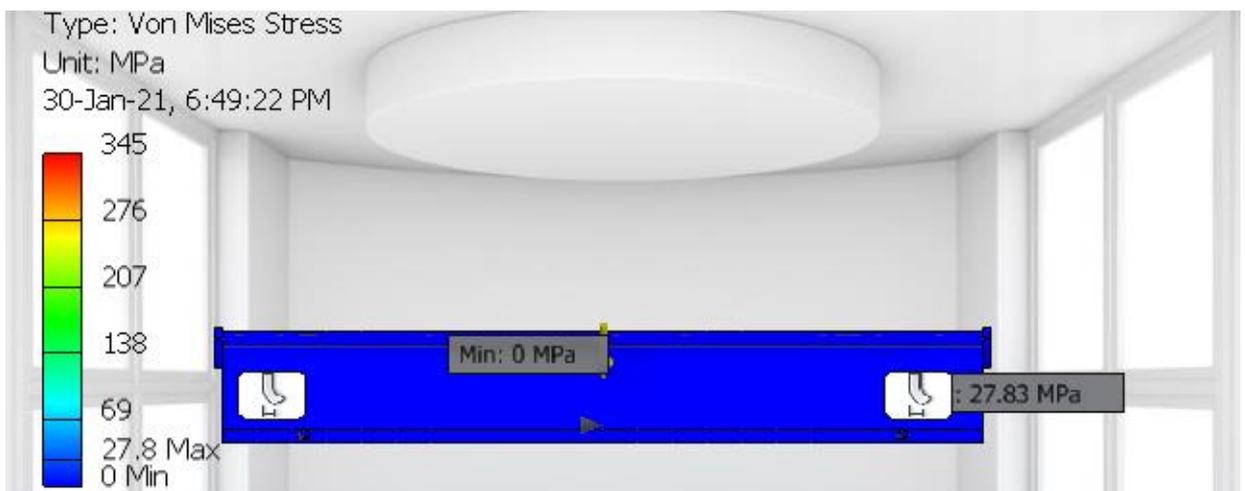
Para realizar el análisis estructural del sistema se utilizó la metodología de elementos finitos mediante el software Autodesk Inventor Professional 2020 con licencia académica No. 900-68909192/797L1. En esta se involucraron los pesos de la cámara, caja y las cargas ejercidas por la masa del riel. El material de diseño fue el acero inoxidable 430 con un límite de fluencia de 345 MPa [34].

Este análisis se realizó en el punto más crítico con las cargas involucradas, dando como resultado un factor de seguridad de 3 (Apéndice C), siendo esta plataforma un diseño seguro, como se observa en la Figura 3.2, detallando que el color azul es seguro y rojo estado crítico.



**Figura 3.2 Análisis de elementos finitos de la plataforma móvil.**

Para el análisis de la plataforma del riel móvil, el encargado de sostener la caja de la cámara y realizar el desplazamiento horizontal. Se involucraron las cargas de la cámara y de la caja que contiene los elementos electrónicos (Apéndice C), dando como resultado en base al análisis de Von Mises que el sistema soporta dichas fuerzas ejercidas sobre el mismo (Figura 3.3).



**Figura 3.3 Análisis de elementos finitos de la plataforma del riel móvil.**

### 3.3 Análisis de la estructura de la estantería

Dado que las estanterías en los supermercados ya se encuentran establecidas, se realiza un análisis de elementos finitos de estas, con el fin de conocer si las perchas soportan el mecanismo de monitoreo. Para realizar esta simulación se colocaron las cargas que ejercen dichos componentes, como también las fuerzas que podrían involucrar los productos de ventas (Apéndice C).

Según el análisis de elementos finitos (Figura 3.4) se concluyó que las estanterías cuentan con la capacidad de soportar dichas cargas de los elementos que involucran los elementos mecánicos del sistema móvil con un coeficiente de seguridad de 15 (Apéndice C).

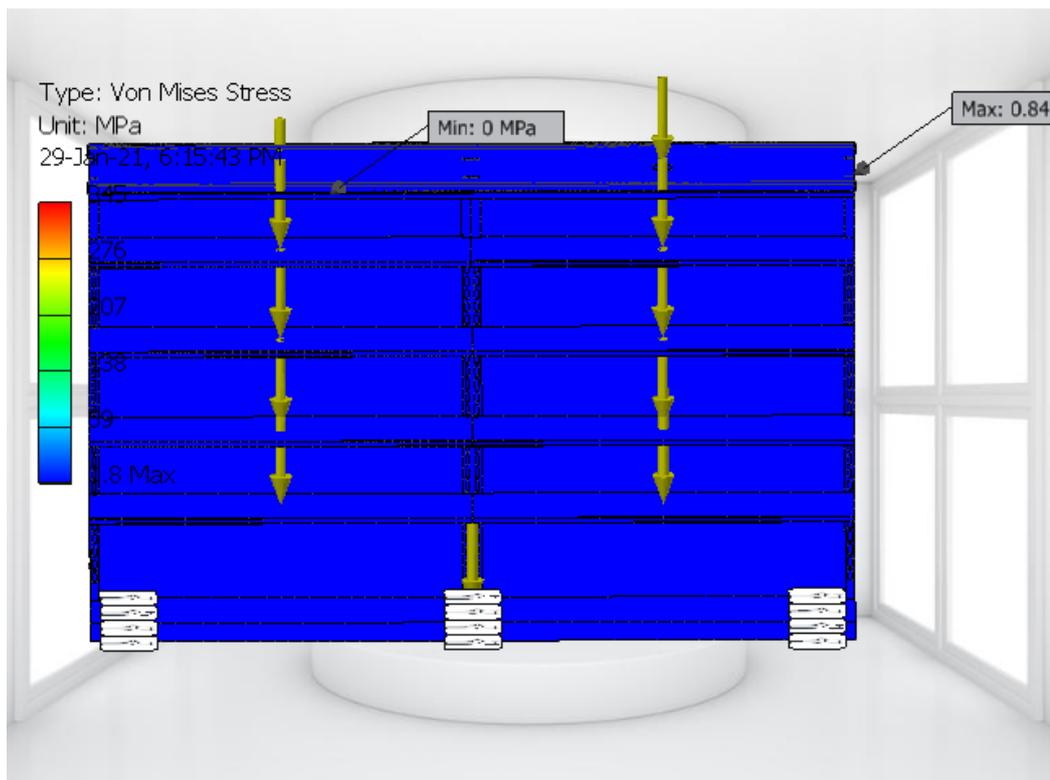


Figura 3.4 Análisis de elementos finitos de estanterías

### 3.4 Sistema embebido

A partir del esquema diseñado en el software Proteus, se modeló la apariencia final de la placa PCB (Figura 3.5), la cual contará con los componentes descritos en la sección 2.6. A demás contará con borneras para conectar los periféricos correspondientes. Tres de ellas, de dos entradas, para los motores DC y una de seis entradas para el motor de paso. Para la alimentación de 5 voltios para el microcontrolador y los componentes electrónicos, se tendrá una bornera de dos entradas. De igual manera se tendrá una bornera de dos entradas para la alimentación de 6V de los controladores de los motores DC.

Por último, se tiene una bornera de 5 pines para conectar cada uno de los cinco sensores finales de carrera.

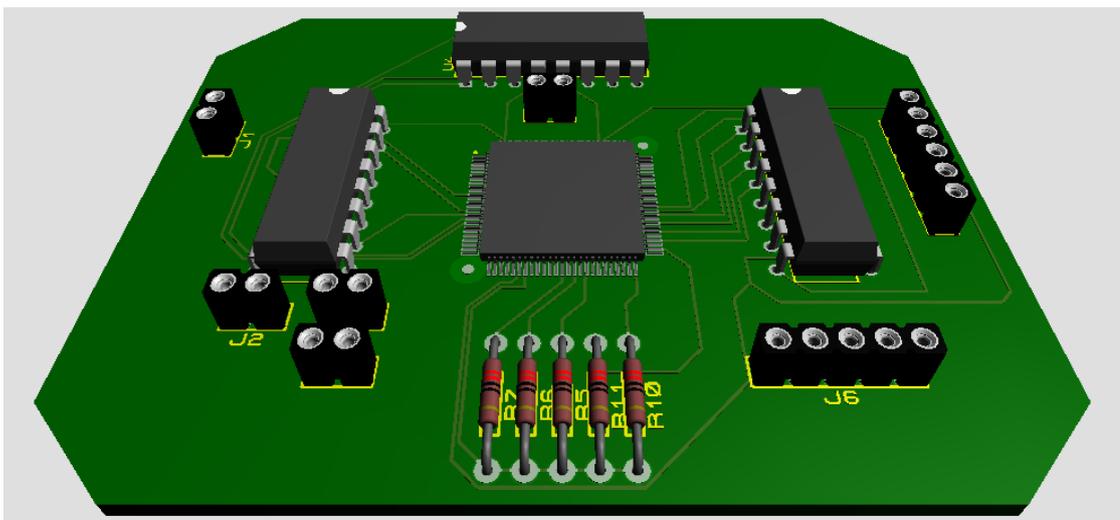


Figura 3.5 Placa PCB del circuito electrónico.

### 3.5 Entrenamiento del modelo

En la Figura 3.6 se detalla las métricas con sus valores correspondientes, mostrando tres capas de dicha red neuronal. En las que se puede destacar dos métricas más importantes que son: “loss” (pérdida) y “precisión” (precisión).

Metrics	YOLO Layer 0	YOLO Layer 1	YOLO Layer 2
grid_size	10	20	40
loss	0.019000	0.030639	0.047757
x	0.004885	0.014560	0.017147
y	0.001936	0.005113	0.014081
w	0.003546	0.003217	0.003772
h	0.003601	0.003720	0.003154
conf	0.005001	0.003991	0.009583
cls	0.000032	0.000038	0.000019
cls_acc	100.00%	100.00%	100.00%
recall50	1.000000	1.000000	1.000000
recall75	1.000000	1.000000	1.000000
precision	1.000000	0.961538	0.925926
conf_obj	0.995720	0.999165	0.998898
conf_noobj	0.000005	0.000031	0.000064

Total loss 0.0973951667547226  
 ---- ETA 0:00:00

**Figura 3.6 Resultado de entrenamiento del modelo**

La función de pérdida utilizada para medir la eficacia del modelo de detección de objetos está dada por la ecuación 3.1 donde:

- X, Y: ubicación del centroide de la caja de ancla.
- W, h: a la altura y ancho de la caja de ancla respectivamente.
- C: confiabilidad de la muestra para determinar si existe un objeto o no.
- P(c): función de pérdida de la clasificación.

$$\begin{aligned}
 loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
 & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2
 \end{aligned} \tag{3.1}$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

Cada uno de los términos de la función se encuentra determinada por el error cuadrático medio entre los datos reales y los predichos.

$\sum_{j=0}^B$  Cálculo de la función de pérdida para cada una de las cajas de ancla que se predicen por cada muestra 5.

$\sum_{i=0}^{S^2}$  Mapeo para cada una de las celdas de la imagen 13x13.

$\mathbb{1}_{ij}^{obj}$  Es 1 cuando hay un objeto en la celda especificada caso contrario toma el valor de cero.

$\mathbb{1}_{ij}^{noobj}$  Es 1 cuando no hay un objeto en la celda especificada caso contrario toma el valor de cero.

$\mathbb{1}_i^{obj}$  Es 1 cuando la clasificación predicha es correcta, sino es cero.

$\lambda$ : Hace referencia a constantes que dan prioridad a la ubicación de las coordenadas de las celdas.

$W$  y  $h$  se encuentran dentro de raíces cuadradas con la finalidad de penalizar a las cajas de anclas más pequeñas para así poder ajustarlas mejor.

En resumen:

- El primer término penaliza la mala localización del centroide de las cajas.
- El segundo término penaliza el error de predicción del ancho y largo de la caja.
- El tercer término reduce el valor del error si la puntuación de confianza es más cercana a 1 cuando hay un objeto en la celda.
- El cuarto término trata de reducir el valor de la puntuación de confianza cuando no hay un objeto en la celda.
- El quinto término es una función de pérdida del modelo de clasificación.

La función precisión del modelo de detección de objetos está dada por la ecuación 2 donde:

- TP: Hace referencia a los verdaderos positivos, es decir, verdaderos resultados.
- FP: Valores considerados como falsos positivos, es decir, predicciones que dieron como resultado error y resultó correcta dicha predicción.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.2)$$

Es así como el valor de precisión es la relación entre verdaderos positivos (TP) y el total de resultados correctos (TP+FP).

Concluyendo que el sistema tiene una precisión del 95%, resultando un reconocimiento de objetos confiable.

### **3.6 Monitoreo de estanterías**

Haciendo uso del modelo entrenado (Apéndice E), se analizó un video el cual filmaba la estantería en dos eventos distintos. El primero se tenía a los ocho productos colocados correctamente mientras que el otro contaba con un tipo de mercadería menos y a su vez algunos de estos se encontraban ubicados en lugares incorrectos.

En el primer video se puede observar el siguiente frame (Figura 3.7), lo que evidencia al analizar los contornos color verde alrededor de los productos que este fue reconocido exitosamente y se encuentra dentro de la ubicación correcta, colocando en la parte superior izquierda el nombre de dicha mercadería, mientras que en la derecha la probabilidad de que sea la clase correcta.



Figura 3.7 Frame de video de estantería completa

Al analizar el segundo video en base al modelo entrenado con las clases establecidas (2.9) y haciendo cumplir las condicionales antes mencionadas se puede evidenciar en la Figura 3.8 que encontramos tres productos que no se encuentran cumpliendo dichos requerimientos. Estos estados de fallo son encajonados en un recuadro color rojo, mostrando de esta manera al usuario el error en la estantería y así solucionar dicho error.



Figura 3.8 Frame video estantería con error

### 3.7 Monitoreo en aplicativo móvil

La aplicación móvil muestra mediante dos colores el estado de los productos. Cada producto está identificado con un icono mostrando la imagen del artículo en cuestión. La App está constantemente leyendo la base de datos, y funciona de tal manera que al detectar un cambio en el valor de un producto de 'G' a 'R', el fondo de la imagen de dicho producto se vuelve color rojo. En caso de que el producto conserve en valor 'G', el fondo permanecerá de color verde (Figura 3.9).

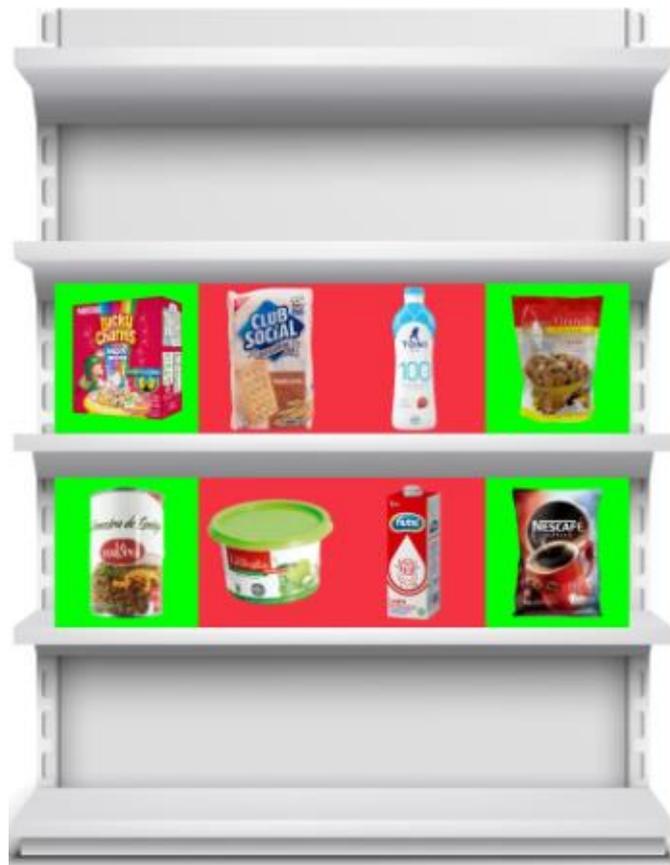


**Figura 3.9 Estado de estanterías en aplicativo móvil con productos correctamente abastecidos y ubicados.**

Se realizaron pruebas tanto con todos los objetos ubicados en su sitio correspondiente como con elementos faltantes. Estas pruebas se hicieron a partir de lo realizado previamente en la sección de monitoreo de estanterías del presente capítulo.

Para la prueba con elementos faltantes se obtuvo, como era esperado, cuatro ítems en color rojo, correspondientes a los productos faltantes (lavavajillas) y mal ubicados (leche, yogurt y galletas).

## Planograma



**Figura 3.10** Estado de estanterías en aplicativo móvil con productos incorrectamente abastecidos y ubicados.

### **3.8 Monitoreo en base de datos**

Siguiendo las pruebas de la sección 3.6, se parte del escenario en el que todos los productos se encuentran en su sitio. Como se muestra en la Figura 3.11.

### planograma-8aa40-default-rtdb



**Figura 3.11 Estado de base de datos con productos correctamente abastecidos y ubicados.**

Al detectar la ausencia o colocación equivocada de un producto el software se encarga de actualizar el valor del producto en cuestión a 'R' (Figura 3.12).

Complementando el ejemplo de la sección 3.6, se muestra los cambios resultantes en la base de datos, de los cuales se derivan las respuestas observadas en el aplicativo móvil en la sección 3.7.

### planograma-8aa40-default-rtdb



**Figura 3.12 Estado de base de datos con productos incorrectamente abastecidos y ubicados.**

### 3.9 Análisis de costos

El análisis de costos del proyecto se realizó en base a la elaboración de un solo diseño de prototipo. En este estudio se tomó en cuenta los honorarios de los ingenieros encargados en el desarrollo del trabajo, desglosando cada valor según el tipo de tarea a realizar. Por ello se decidió elaborar el análisis de costos en tres partes: profesionales, mecánicos y electrónicos.

**Tabla 3.2 Costos profesionales.**

<b>COSTOS PROFESIONALES</b>			
<b>Concepto</b>	<b>Horas</b>	<b>Costo/Hora</b>	<b>Total</b>
Instalación del sistema	3	\$ 30.00	\$ 90.00
Diseño del sistema embebido	8	\$ 40.00	\$ 320.00
Entrenamiento del modelo AI	20	\$ 5.00	\$ 100.00
Programación	20	\$ 20.00	\$ 400.00
Diseño del sistema móvil	10	\$ 40.00	\$ 400.00
App móvil y base de datos	6	\$ 30.00	\$ 180.00
Ensamble del sistema	5	\$ 20.00	\$ 100.00
<b>Total</b>			<b>\$ 1,590.00</b>

**Tabla 3.3 Costos mecánicos.**

<b>COSTOS MECÁNICOS</b>			
<b>Componentes</b>	<b>Cantidad</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
Base metálica	1	96.1	96.1
Caja metálica	1	13.99	13.99
Lamina de acero	15.54	6	93.24
Piñón - cremallera (1m)	10	3.7	37
Tubo acero	3	15.87	47.61
<b>Total</b>			<b>287.94</b>

**Tabla 3.4 Costos electrónicos.**

<b>COSTOS ELECTRÓNICOS</b>			
<b>Componentes</b>	<b>Cantidad</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
Arduino Mega	1	\$ 18.99	\$ 18.99
Motor paso a paso unipolar nema 11	1	\$ 17.99	\$ 17.99
Motor DC ET-PGM16	3	\$ 10.99	\$ 32.97
Cámara	1	\$ 54.99	\$ 54.99
Cables	1	\$ 2.49	\$ 2.49
integrado L293D	2	\$ 1.59	\$ 3.18
Sensor final de carrera	5	\$ 1.00	\$ 5.00
Integrado ULN2003A	1	\$ 1.00	\$ 1.00
<b>Total</b>			<b>\$ 136.61</b>

Obteniendo un total de costos del proyecto de USD 2014,55.

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1. Conclusiones

- El sistema de monitoreo de las estanterías realizado cuenta con un 95% de precisión, teniendo como efecto una detección de productos confiable, de tal manera que el uso de esta herramienta conllevó a grandes beneficios en el monitoreo constante de las estanterías en los supermercados.
- El mecanismo móvil diseñado permite realizar el recorrido de la cámara sobre la estantería, llegando a diferentes puntos estratégicos de monitoreo localizados, para supervisar el estado de los productos de las perchas a su alrededor.
- La implementación del software de detección de objetos Yolov3 fue útil para detectar los productos de una estantería que cuenta con un dataset de 2266 imágenes, lo que permitió al programar ejecutar el código con el modelo entrenado para hacer un eficiente reconocimiento de objetos.
- El estudio de elementos finitos determinó que los materiales involucrados en el sistema de monitoreo son resistente a los esfuerzos sometidos, y a su vez la estantería soporta dichas cargas que aplica el mecanismo móvil y sus dispositivos electrónicos.
- El diseño del aplicativo móvil permite mostrar de forma clara el estado de los productos en las estanterías en tiempo real, de tal forma que representaría una herramienta útil y fácil de usar en manos del personal encargado de las perchas.
- El diseño modular del prototipo hace posible una instalación sencilla del sistema, permitiendo así que este proyecto pueda ser aplicado a gran escala, de una manera rápida y segura.

## 4.2. Recomendaciones

- Crear un dataset con un número de imágenes superior a 1000 por cada producto, siendo estas fotografías tomadas en diferentes ángulos y posiciones (frontal, posterior, laterales), con el fin de garantizar una mejor precisión en el reconocimiento de los objetos.
- Utilizar una computadora que cuente con una capacidad alta de memoria RAM, tarjeta gráfica y almacenamiento para poder llevar a cabo todas las operaciones de software eficientemente.
- Utilizar un solo motor y un solo sistema piñón cremallera para el movimiento horizontal de la plataforma móvil, ubicado en el centro de la plataforma, destacando que se debería volver a dimensionar los elementos encargados de este movimiento.
- Al momento de implementar el sistema de monitoreo sobre las estanterías, verificar cada punto estratégico de control del estado de los productos en las estanterías.
- Diseñar una página web para monitorear las estanterías desde las computadoras con una interfaz más amigable.
- El sistema de monitoreo desarrollado puede ser adaptado para tener más funcionalidades tales como la detección de etiquetas faltantes y productos
- El sistema de monitoreo puede ser adaptado para complementarse con futuros avances tecnológicos, como un sistema de reposición automática de productos.

# BIBLIOGRAFÍA

- [1] Bolsalea, “Los orígenes del comercio - Bolsalea Blog,” 2020. [Online]. Available: <https://www.bolsalea.com/blog/2013/05/los-origenes-del-comercio/>. [Accessed: 09-Nov-2020].
- [2] X. Nuria, “Diferentes sistemas de ventas,” 2013. [Online]. Available: <https://es.slideshare.net/nuria229/diferentes-sistemas-de-ventas>. [Accessed: 09-Nov-2020].
- [3] B. Chadda, “Supermarket sweep | lots of words,” 2012. [Online]. Available: <https://biancajchadda.wordpress.com/2012/11/29/supermarket-sweep/>. [Accessed: 09-Nov-2020].
- [4] B. S. Avila Mendoza, “ANÁLISIS DE LA CADENA DE ABASTECIMIENTO DE SUPERMERCADOS EN ECUADOR,” Quito, 2017.
- [5] J. Vera, “Entrevista a empleado de supermercado,” 2020.
- [6] T. W. Gruen and S. Bharadwaj, “Retail Out-of-Stocks: A Worldwide Examination of Extent, Causes and Consumer Responses.”
- [7] R. Redman, “First Amazon Fresh store opens to public,” 2020. [Online]. Available: <https://www.supermarketnews.com/retail-financial/first-amazon-fresh-store-opens-public>. [Accessed: 09-Nov-2020].
- [8] “Compra de ciertos víveres sube hasta 137 % durante cuarentena | Economía | Noticias | El Universo,” Guayaquil, 20-Apr-2020.
- [9] “Reposición de productos - Pricer,” 2020. [Online]. Available: <https://www.pricer.com/es/casos-de-uso/operaciones-de-la-tienda/reposicion/>. [Accessed: 11-Nov-2020].
- [10] G. Buzek, “Out of Stocks, Out of Luck,” 2018.
- [11] “Reponedor - educaweb.com,” 2020. [Online]. Available: <https://www.educaweb.com/profesion/reponedor-235/>. [Accessed: 11-Nov-2020].
- [12] A. Fernández, G. En, and E. En, “Sistema automático de verificación de cumplimiento de planogramas usando inteligencia artificial aplicada a imágenes Treball de Fi de Grau,” 2018.
- [13] C. C. NAPOLEÓN GUALE and M. T. AURORA ENORIO, “INFLUENCIA DE LA IMAGEN CORPORATIVA EN LAS TÉCNICAS DE MERCHANDISING

PROPUESTA,” 2013.

- [14] “Estante Del Supermercado Productos - Imagen gratis en Pixabay.” [Online]. Available: <https://pixabay.com/es/illustrations/estante-del-supermercado-productos-1094815/>. [Accessed: 01-Feb-2021].
- [15] D. J. Matich, “Redes Neuronales: Conceptos Básicos y Aplicaciones.,” *Historia Santiago.*, p. 55, 2001.
- [16] F. Izaurieta and C. Saavedra, “Redes Neuronales Artificiales,” 2000.
- [17] “Reconocimiento de objetos - MATLAB & Simulink.” [Online]. Available: <https://es.mathworks.com/solutions/image-video-processing/object-recognition.html>. [Accessed: 10-Nov-2020].
- [18] “Todo lo que necesitas saber sobre TensorFlow, la plataforma para Inteligencia Artificial de Google – Puentes Digitales.” [Online]. Available: <https://puentesdigitales.com/2018/02/14/todo-lo-que-necesitas-saber-sobre-tensorflow-la-plataforma-para-inteligencia-artificial-de-google/>. [Accessed: 10-Nov-2020].
- [19] “About.” [Online]. Available: <https://opencv.org/about/>. [Accessed: 10-Nov-2020].
- [20] “¿Qué es una célula de carga y cómo funciona? Tipos de celda.” [Online]. Available: <https://es.omega.com/prodinfo/celulas-de-carga.html>. [Accessed: 09-Dec-2020].
- [21] “MSRobotics - Librería HX711, Balanza digital con Arduino.” [Online]. Available: <http://msrobotics.net/index.php/laboratorio-ard/234-libreria-hx711-balanza-digital-con-arduino>. [Accessed: 09-Dec-2020].
- [22] “3D Rendering Of Surveillance Camera In Supermarket. Stock Photo, Picture And Royalty Free Image. Image 57167601.” [Online]. Available: [https://www.123rf.com/photo\\_57167601\\_3d-rendering-of-surveillance-camera-in-supermarket-.html](https://www.123rf.com/photo_57167601_3d-rendering-of-surveillance-camera-in-supermarket-.html). [Accessed: 09-Dec-2020].
- [23] “Rfid: qué es y cómo funciona | VIU.” [Online]. Available: <https://www.universidadviu.com/co/actualidad/nuestros-expertos/rfid-que-es-y-como-funciona>. [Accessed: 09-Dec-2020].
- [24] “Simple, scalable wireless system uses the RFID tags on billions of products to sense contamination.” [Online]. Available: <https://techxplore.com/news/2018-11-simple-scalable-wireless-rfid-tags.html>. [Accessed: 09-Dec-2020].
- [25] “planetary gear motor,gear motor,dc gear motor,6v 18rpm.” [Online]. Available:

- [http://www.etonm.com/products\\_detail/productId=55.html](http://www.etonm.com/products_detail/productId=55.html). [Accessed: 01-Feb-2021].
- [26] “(No Title).” [Online]. Available: <https://mouser.com/pdfdocs/nema11-amt112s.pdf>. [Accessed: 01-Feb-2021].
- [27] “YOLO: Real-Time Object Detection.” [Online]. Available: <https://pjreddie.com/darknet/yolo/>. [Accessed: 10-Dec-2020].
- [28] “Amazon.com : Dragon Touch 4K Action Camera Touch Screen 16MP Vision 3 Pro PC Web Camera 100 feet Waterproof Camera Adjustable View Angle WiFi Sports Camera with Remote Control and Helmet Accessories Kit: Camera & Photo.” [Online]. Available: [https://www.amazon.com/Dragon-Touch-Vision-Pro-Accessories/dp/B07S38ZBFM/ref=sr\\_1\\_7?dchild=1&keywords=camara+web+12mp&qid=1607697230&sr=8-7](https://www.amazon.com/Dragon-Touch-Vision-Pro-Accessories/dp/B07S38ZBFM/ref=sr_1_7?dchild=1&keywords=camara+web+12mp&qid=1607697230&sr=8-7). [Accessed: 15-Dec-2020].
- [29] “Review: Tarjeta Gráfica NVIDIA Geforce GTX 1080 Founders Edition – Rompiendo los esquemas de Eficiencia y Rendimiento | OZEROS.” [Online]. Available: <https://www.ozeros.com/2016/05/review-tarjeta-grafica-nvidia-geforce-gtx-1080-founders-edition-rompiendo-los-esquemas-de-eficiencia-y-rendimiento/>. [Accessed: 15-Dec-2020].
- [30] “Conda Essentials Notes | Imad Dabbura.” [Online]. Available: <https://imaddabbura.github.io/post/conda-essentials/>. [Accessed: 10-Dec-2020].
- [31] “Username Reminder Request.” [Online]. Available: <http://www.engysbol.net/index.php/username-reminder>. [Accessed: 10-Dec-2020].
- [32] “Uno Rev3 - Ardubitronics.” [Online]. Available: <https://www.ardubitronics.com/compatible/38-uno-rev3.html>. [Accessed: 10-Dec-2020].
- [33] “Arduino Mega 2560 Atmega Original Italiano+cable Electronics | Mercado Libre.” [Online]. Available: [https://articulo.mercadolibre.com.ec/MEC-428575012-arduino-mega-2560-atmega-original-italianocable-electronics-\\_JM#position=3&type=item&tracking\\_id=1fdd3f84-835f-465f-a964-2d7247817006](https://articulo.mercadolibre.com.ec/MEC-428575012-arduino-mega-2560-atmega-original-italianocable-electronics-_JM#position=3&type=item&tracking_id=1fdd3f84-835f-465f-a964-2d7247817006). [Accessed: 07-Jan-2021].
- [34] “430 Stainless Steel annealed sheet.” [Online]. Available: <http://www.matweb.com/search/DataSheet.aspx?MatGUID=1306bb472ffc40dea7b5acab1ee2f5a0>. [Accessed: 01-Feb-2021].

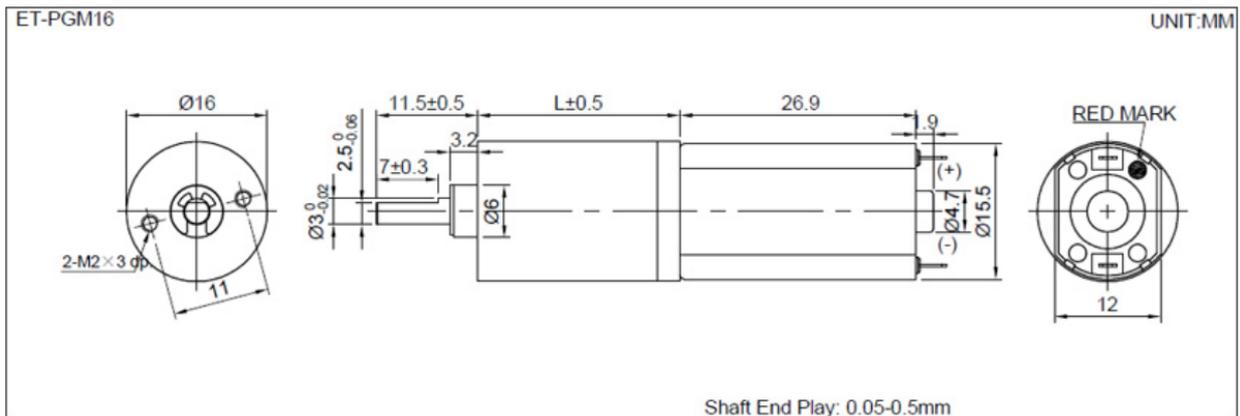
# Apéndice

# APÉNDICE A

## ESPECIFICACIONES TÉCNICAS DE LOS COMPONENTES SELECCIONADOS

**Tabla A.1 Especificaciones de los motores DC utilizados**

Modelo		ET-PGM16-061600	ET-PGM16-06355
Voltaje	Rango de operación	4.5-7.2V	4.5-7.2V
	Nominal	6V	6V
Sin carga	Velocidad (rpm)	1600	355
	Corriente (mA)	≤120	≤120
	Velocidad (rpm)	1300	275
Carga nominal	Corriente (mA)	≤220	≤220
	Torque (kg.cm)	0.03	0.13

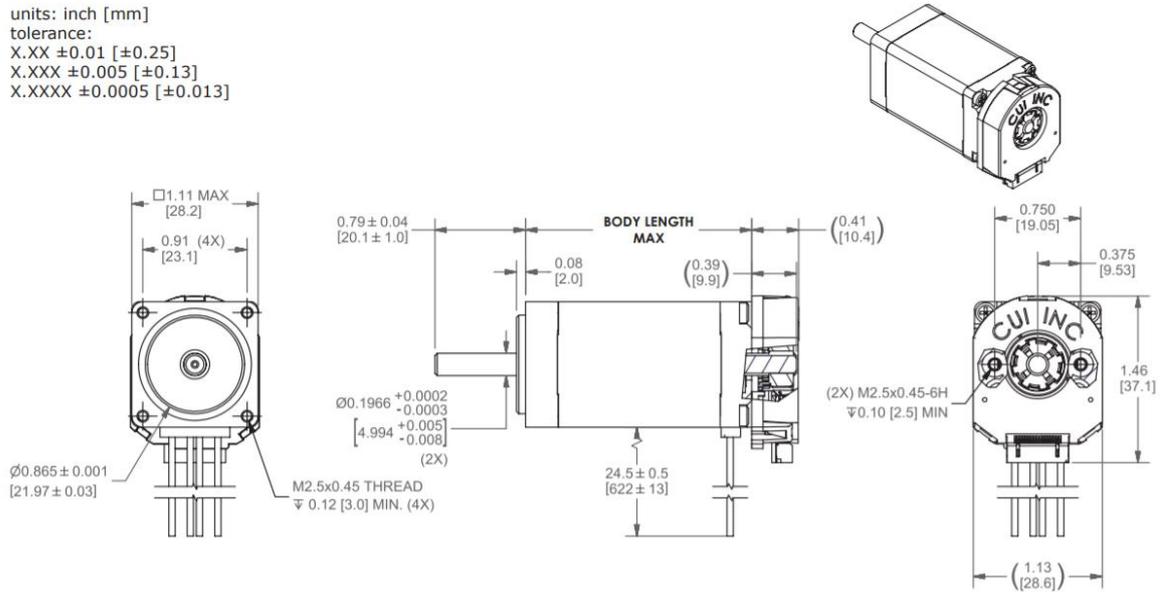


**Figura A.1 Dimensiones de los motores DC.**

**Tabla A.2 Especificaciones del motor de paso.**

Modelo	Nema 11 20-02D-AMT112S
Angulo de paso (°)	1.8
Corriente / fase (A)	1.3
Resistencia / fase (Ω)	1.9
Inductancia / fase (mH)	1.46
Torque máximo (Nm)	0.117
Velocidad máxima (rpm)	1320
Longitud (cm)	5.11

units: inch [mm]  
 tolerance:  
 X.XX ±0.01 [±0.25]  
 X.XXX ±0.005 [±0.13]  
 X.XXXX ±0.0005 [±0.013]



MODEL NO.	BODY LENGTH (inch)	WEIGHT (lb)
NEMA11-13-01D-AMT112S	1.26	0.24
NEMA11-18-01D-AMT112S	1.77	0.35
NEMA11-18-02D-AMT112S	1.77	0.35
NEMA11-20-01D-AMT112S	2.01	0.45
NEMA11-20-02D-AMT112S	2.01	0.45

Figura A.2 Dimensiones del motor de paso.

Tabla A.3 Datos técnicos del microcontrolador Atmega2560.

Flash	256KB
EEPROM	4KB
RAM	8KB
Pines I/O	86
Canales PWM de 16 bits de resolución	12
USARTs Seriales	4
Canales ADC	16
Alimentación	4.5 - 5.5V

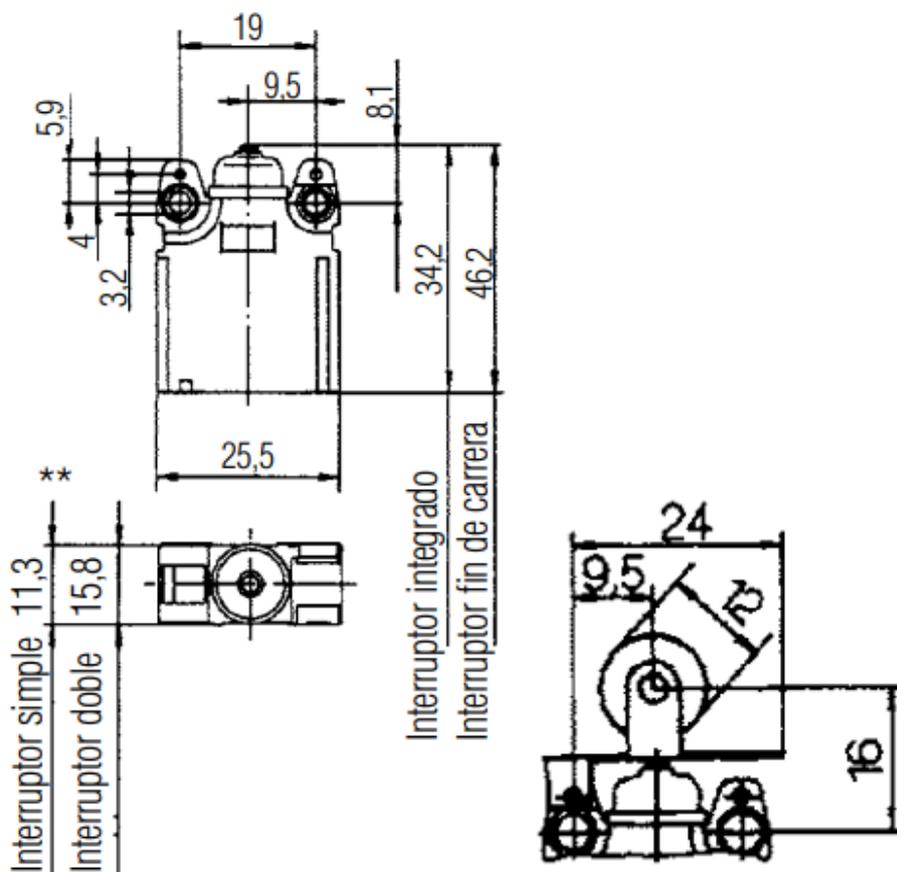


Figura A.3 Dimensiones de los sensores finales de carrera.

Tabla A.4 Especificaciones del integrado ULN2003A.

Parámetro	Valor	Unidad
Voltaje de salida	50	V
Voltaje de entrada	30	V
Corriente continua de colector	500	mA
Corriente continua de base	25	mA
Voltaje continuo de diodo de sujeción	350	mA
Voltaje reverso de diodo de sujeción	50	V
Temperatura ambiente de operación	-40 a 85	°C
Temperatura de almacenamiento	-55 a 150	°C
Temperatura de unión	150	°C
Tasa de descarga electrostática	2	kV

**Tabla A.5 Especificaciones del integrado L293D**

		Mínimo	Máximo	Unidad
Voltaje de alimentación	VCC1	4.5	7	V
	VCC2	VCC1	36	
Voltaje nivel alto	$VCC1 \leq 7V$	2.3	VCC1	V
	$VCC1 \geq 7V$	2.3	7	V
Voltaje nivel bajo		-0.3	1.5	V
Temperatura de operación		0	70	°C

## APÉNDICE B

### CÁLCULOS DE DISEÑO MECÁNICO

#### Piñón superior

Coeficiente de seguridad para un tiempo de trabajo elevado y una carga moderada.

$$K = 2 \quad (\text{B.1})$$

Aceleración requerida:

$$a = 0.134 \frac{m}{s^2} \quad (\text{B.2})$$

Velocidad requerida:

$$V = 1 \frac{m}{s} \quad (\text{B.3})$$

Masa de la carga:

$$P = 1.673 \text{ kg} \quad (\text{B.4})$$

Diámetro primitivo:

$$dp = 18 \text{ cm} \quad (\text{B.5})$$

*Rendimiento: 85%*

Fuerza tangencial para cremalleras horizontales:

$$F_{ht} = P * a = 1.673 * 0.134 = 0.224 \text{ N} \quad (\text{B.6})$$

Fuerza tangencial corregida:

$$F_{hc} = F_{ht} + K = 0.224 + 2 = 2.224 \text{ N} \quad (\text{B.7})$$

Módulo dentado recto:

$$m = 1 \quad (\text{B.8})$$

Número de dientes:

$$z = \frac{d_{max}}{m} = \frac{d_{max}}{m} \quad (\text{B.9})$$

$$z = \frac{dp}{m} = \frac{18}{1} = 18 \text{ dientes} \quad (\text{B.10})$$

$$dp = \frac{M * 2000 * \eta}{F_{ht}} \quad (\text{B.11})$$

Par de giro:

$$M = \frac{dp * F_{vt}}{2000 * \eta} = \frac{18 * 0.224}{2000 * 0.85} = 0.00237 \text{ Nm} = 0.02417 \text{ kg.cm} \quad (\text{B.12})$$

Velocidad angular del eje:

$$n = \frac{V * 60000}{dp * \pi} = \frac{1 * 60000}{18 * \pi} = 1061 \text{ rpm} \quad (\text{B.13})$$

### Piñones inferiores

Coefficiente de seguridad:

$$K = 2 \quad (\text{B.14})$$

Aceleración requerida:

$$a = 0.134 \frac{m}{s^2} \quad (\text{B.15})$$

Velocidad requerida:

$$V = 0.25 \frac{m}{s} \quad (\text{B.16})$$

Masa de la carga:

$$P = \frac{15.540 + 1.673}{2} = 8.6065 \text{ kg} \quad (\text{B.17})$$

Diámetro primitivo:

$$dp = 18 \text{ cm} \quad (\text{B.18})$$

Rendimiento: 85%

Fuerza tangencial para cremalleras horizontales:

$$F_{ht} = P * a = 8.6065 * 0.134 = 1.153 \text{ N} \quad (\text{B.19})$$

Fuerza tangencial corregida:

$$F_{hc} = F_{ht} + K = 1.343 + 2 = 3.153 \text{ N} \quad (\text{B.20})$$

Módulo dentado recto:

$$m = 1 \quad (\text{B.21})$$

Diámetro primitivo:

$$z = \frac{d_{max}}{m} = \frac{d_{max}}{m} \quad (\text{B.22})$$

$$z = \frac{dp}{m} = \frac{18}{1} = 18 \text{ dientes} \quad (\text{B.23})$$

$$dp = \frac{M * 2000 * \eta}{F_{ht}} \quad (\text{B.24})$$

Par de giro:

$$M = \frac{dp * F_{vt}}{2000 * \eta} = \frac{18 * 1.153}{2000 * 0.85} = 0.01221 \text{ Nm} = 0.1245 \text{ kg.cm} \quad (\text{B.25})$$

$$n = \frac{V * 60000}{dp * \pi} = \frac{0.25 * 60000}{18 * \pi} = 265.26 \text{ rpm} \quad (\text{B.26})$$

# APÉNDICE C

## Análisis de elementos finitos de estantería

Las estructuras analizadas serán divididas en tres partes mencionadas como: estantería, plataforma base y plataforma de riel.

- Estantería: diseño de la estantería con los mecanismos del sistema móvil.
- Plataforma base: diseño de la plataforma que soporta el mecanismo móvil.
- Plataforma del riel: diseño del mecanismo móvil, la superficie donde se encuentra apoyada la caja que soporta la cámara y encargada de hacer el movimiento horizontal del sistema.

### Análisis de estantería

Los elementos involucrados en el diseño de la estantería son el acero inoxidable 430. Con propiedades que se pueden observar en Tabla C.

**Tabla C.1 Propiedades del acero y elementos implementados.**

Nombre	Acero inoxidable 430	
Propiedad mecánica	Límite de fluencia	345 MPa
Parte Nombre(s)	base.ipt	
	estanteria.ipt	
	base.ipt	
	soporte.ipt	
	plataforma.ipt	

En la Figura C.1 se hace detalle de los elementos que involucran el diseño total de las estanterías, los cuales fueron llevados a cabo a partir del acero antes mencionado.

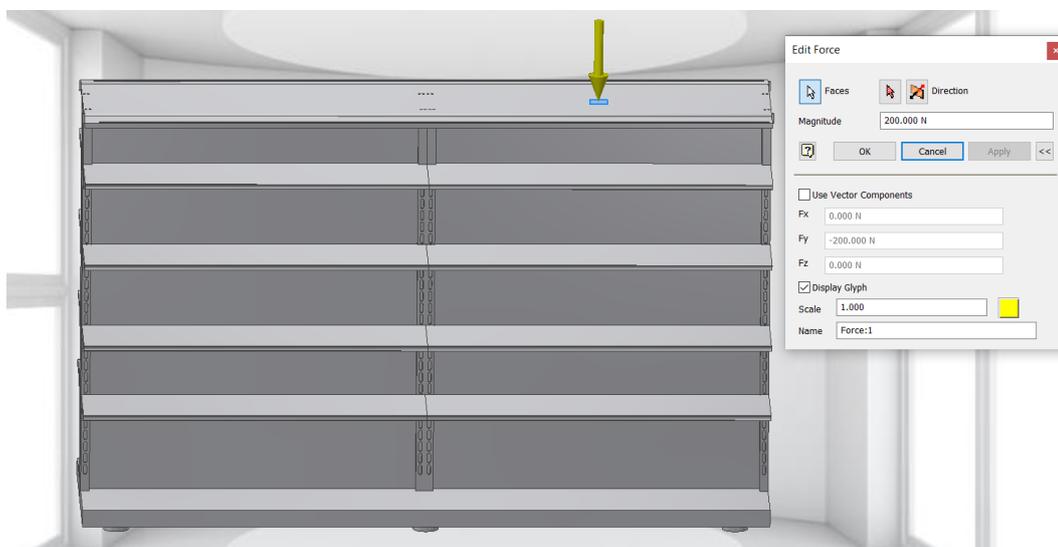
Assign Materials

Component	Original Material	Override Material	Safety Factor
Estanteria Base.iam			
shelff:1			
shelff:2			
base:1	Stainless Steel	(As Defined)	Yield Strength
base:2	Stainless Steel	(As Defined)	Yield Strength
base:4	Stainless Steel	(As Defined)	Yield Strength
estanteria:1	Stainless Steel	(As Defined)	Yield Strength
base:5	Stainless Steel	(As Defined)	Yield Strength
base_MIR1:1	Stainless Steel	(As Defined)	Yield Strength
base_MIR1:2	Stainless Steel	(As Defined)	Yield Strength
base_MIR1:3	Stainless Steel	(As Defined)	Yield Strength
base_MIR1:4	Stainless Steel	(As Defined)	Yield Strength
soporte:1	Stainless Steel	(As Defined)	Yield Strength
soporte:2	Stainless Steel	(As Defined)	Yield Strength
soporte:3	Stainless Steel	(As Defined)	Yield Strength
soporte:4	Stainless Steel	(As Defined)	Yield Strength
PlataformaSuperio			
plataforma:1	Stainless Steel	(As Defined)	Yield Strength

Materials... OK Cancel

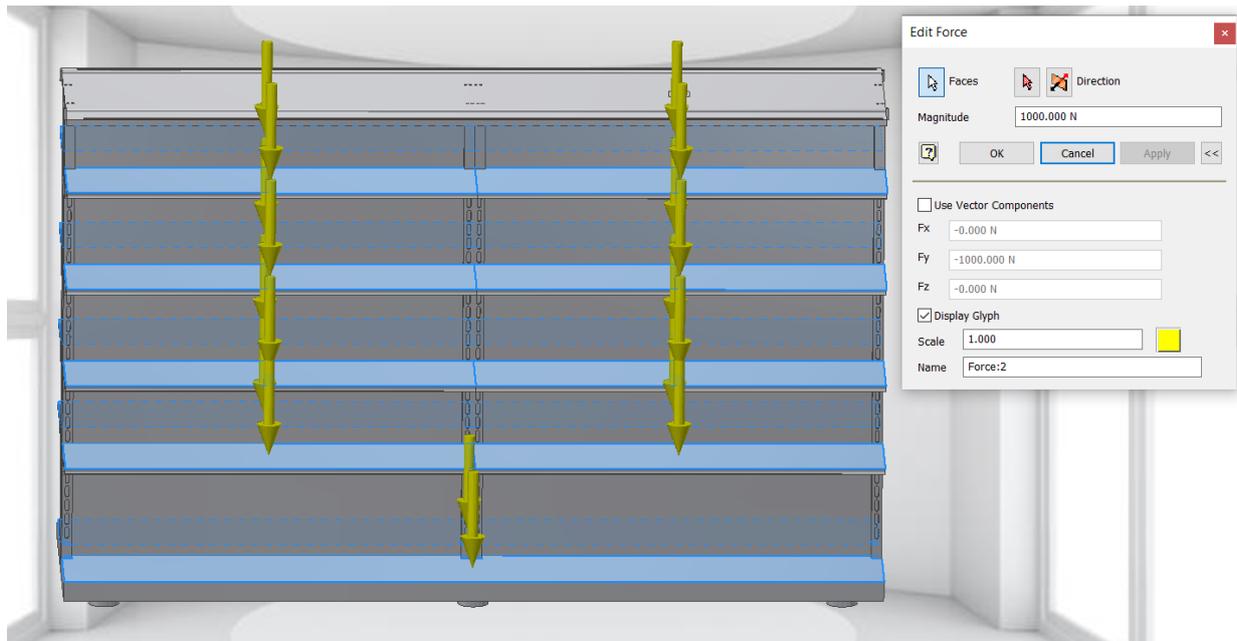
**Figura C.1 Descripción de los elementos en software Inventor.**

En la Figura C.2 se muestra la carga aplicada de 200 N (pesos de la caja y la plataforma del riel móvil), la cual se encuentra ubicada en el punto más crítico de la estantería, con el fin de evidenciar que el diseño es capaz de soportar las fuerzas en dicho lugar, para así concluir que, si el sistema es resistente en dicha ubicación, lo será en toda la estantería.



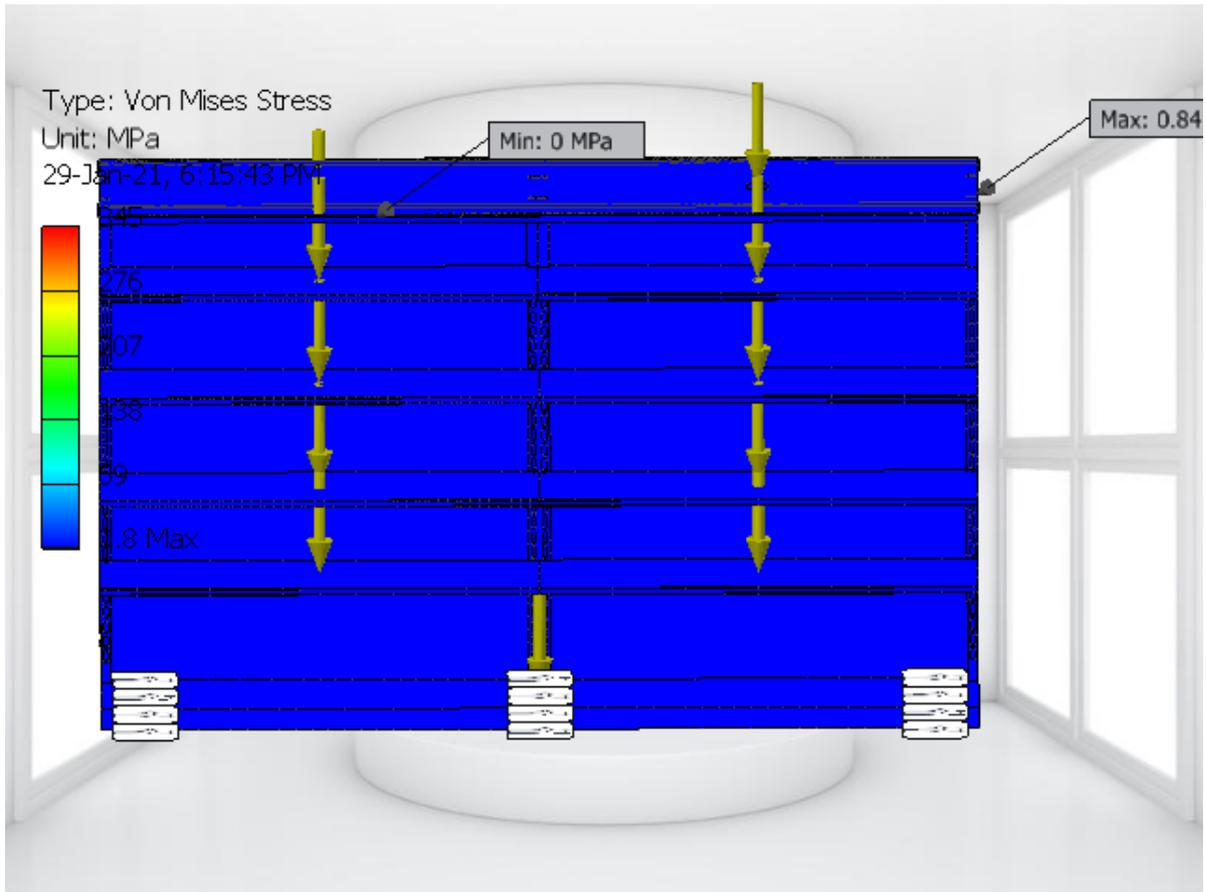
**Figura C.2 Aplicación de carga sobre la estantería.**

En la Figura C.3 se evidencia la fuerza aplicada de 1000 N, ejemplificando los pesos ejercidos de los productos, ubicándolo en la mitad de cada una, el punto más crítico de cada estantería.



**Figura C.3 Aplicación de carga sobre la estantería.**

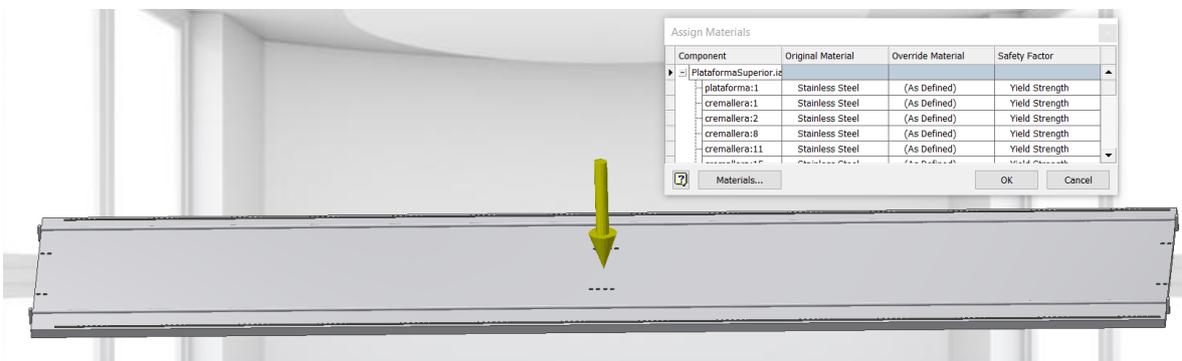
El resultado de la simulación, fue exitoso, los cálculos del software mostraron en el Von Mises (Figura C.4) que la estantería cuenta con la capacidad de tolerar dichas, dando un valor máximo de 0.84 MPa, teniendo como consecuencia un factor de seguridad de 15, concluyendo así que el sistema se encuentra correctamente diseñado para las estanterías.



**Figura C.4 Resultado de elementos finitos de la estantería.**

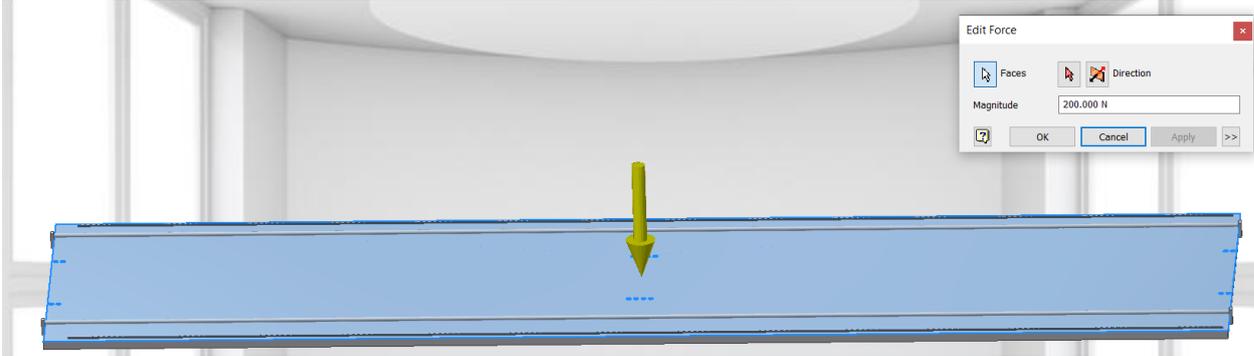
### **Análisis de la plataforma**

Para realizar el análisis de la plataforma se estableció el acero inoxidable 430, con las características detalladas en la Tabla C.1 y cremalleras del mismo material.



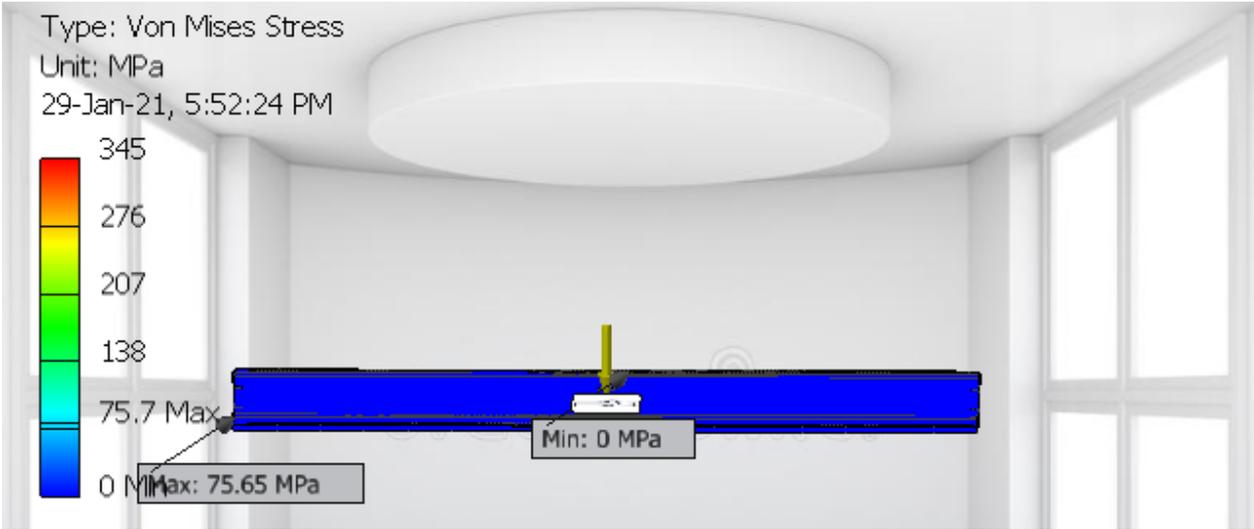
**Figura C.5 Descripción de los elementos en la plataforma.**

La carga aplicada de 200 N es en base al peso que ejerce la caja y el riel sobre la plataforma, colocado en la parte más crítica del diseño, siendo esta el centro como se observa en la Figura C.6.



**Figura C.6 Aplicación de carga sobre la plataforma.**

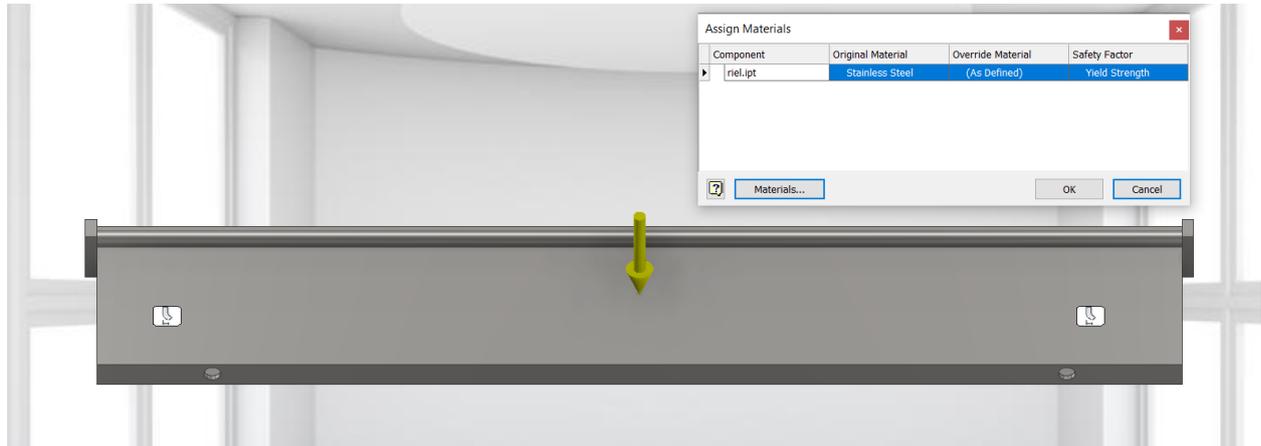
El resultado de la simulación determinó que el valor máximo de Von Mises fue de 75,65 MPa (Figura C.7). El diseño fue elaborado con un factor de seguridad de 3, demostrando así que el mecanismo se encuentra apto para soportar la plataforma del riel y los dispositivos sobre él.



**Figura C.7 Resultado de elementos finitos de la plataforma.**

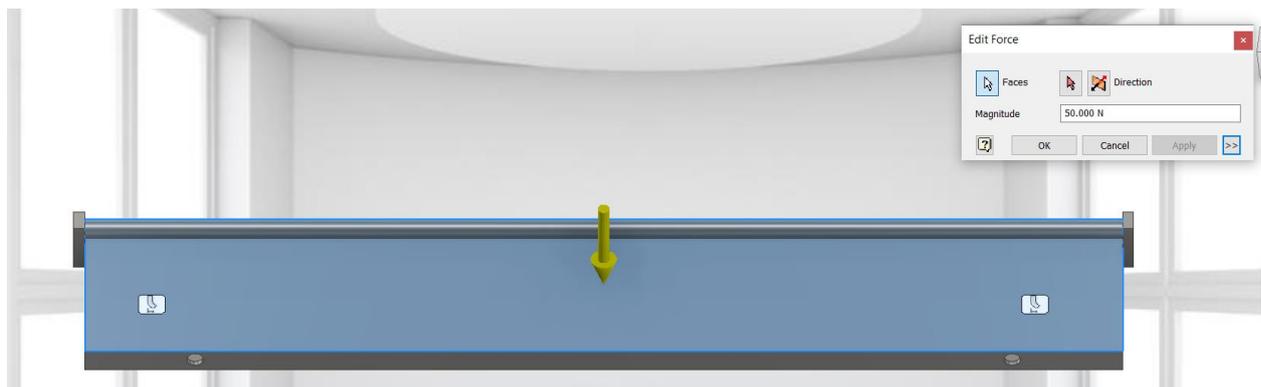
## Análisis del riel

El diseño de la plataforma del riel fue diseñado con acero inoxidable 430 (Figura C.8), con características que se pueden visualizar en la Tabla C.1.



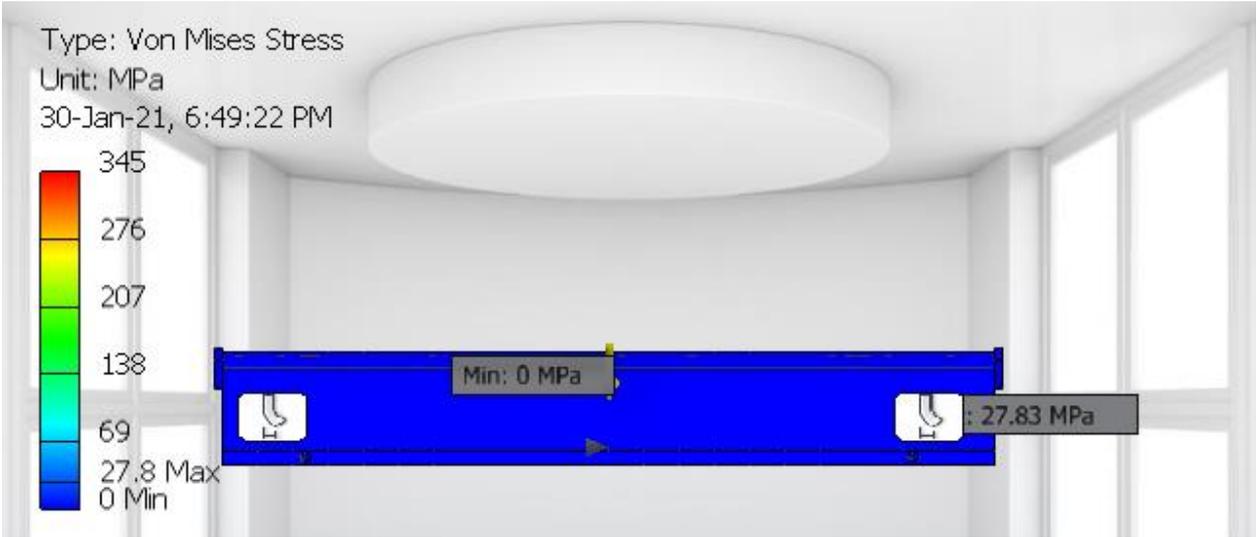
**Figura C.8** Descripción de los elementos de la plataforma del riel.

La carga aplicada de 50 N es en base al peso que ejerce la caja sobre la plataforma del riel, colocado en la parte más crítica del diseño, siendo esta el centro como se observa en la Figura C.9.



**Figura C.9** Aplicación de carga sobre el riel.

El resultado de la simulación determinó que el valor máximo de Von Mises fue de 27,83 MPa (Figura C.10). El diseño fue elaborado con un factor de seguridad de 8, demostrando así que el mecanismo se encuentra apto para soportar la plataforma del riel y los dispositivos sobre él.



**Figura C.10 Resultado de elementos finitos del riel.**

# APÉNDICE D

## Planos mecánicos y eléctricos

Las conexiones realizadas en esta placa se hicieron en base a pistas conductoras a doble capa mostradas con más detalle en la Figura D.1. En color rojo se muestra las pistas de la capa superior, y en color azul las pistas de la placa inferior.

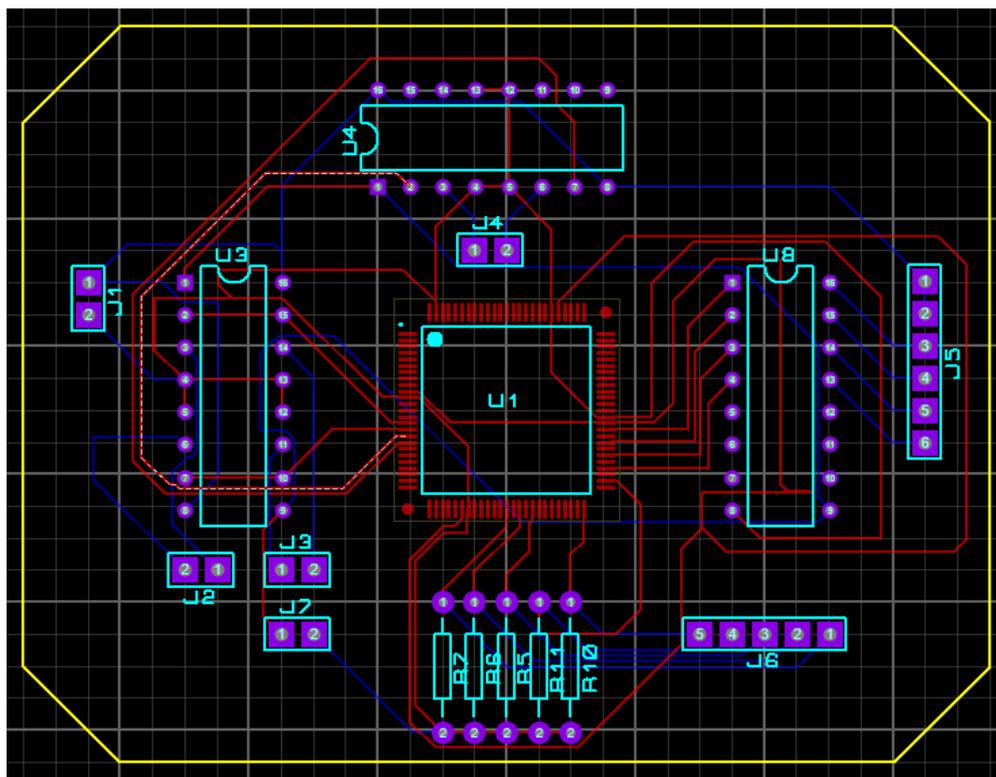
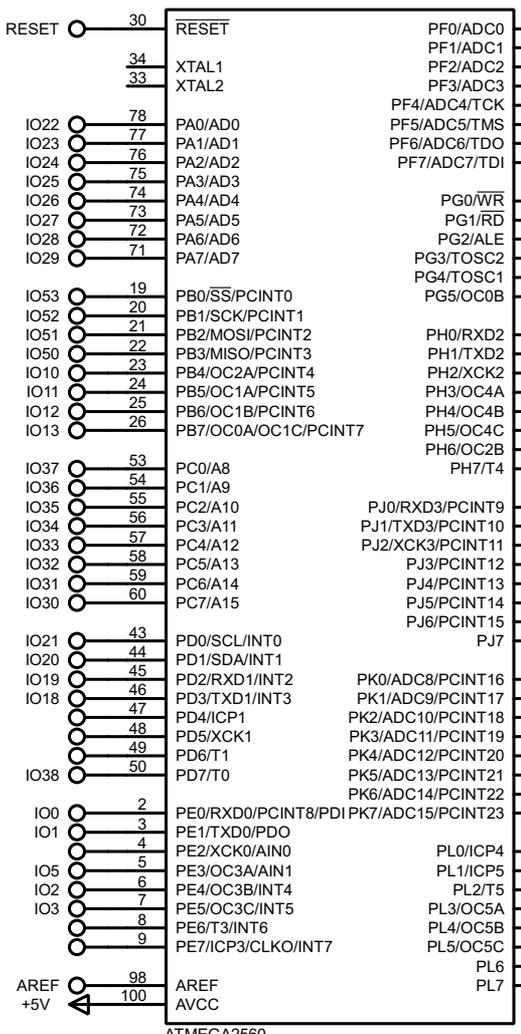
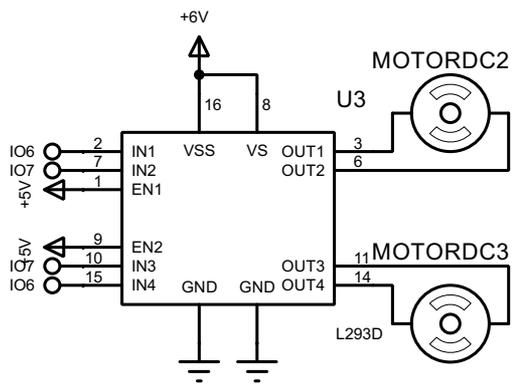
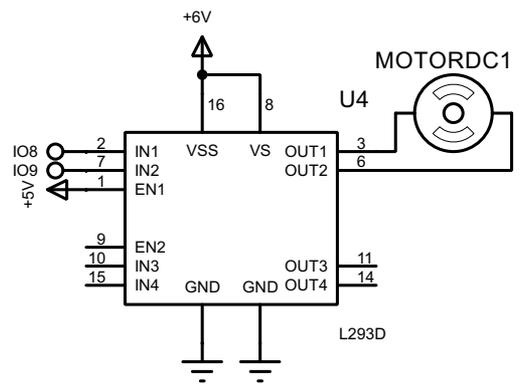
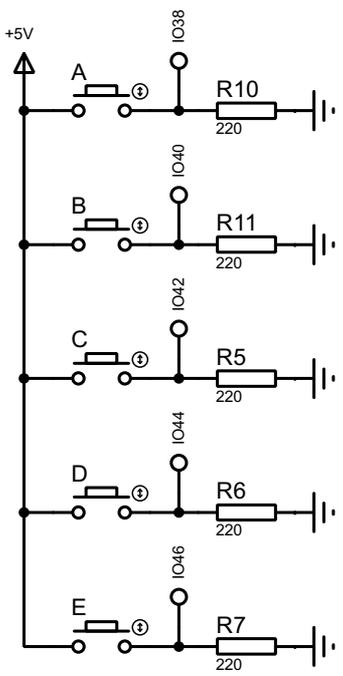
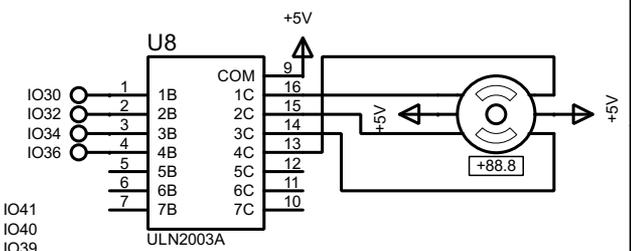
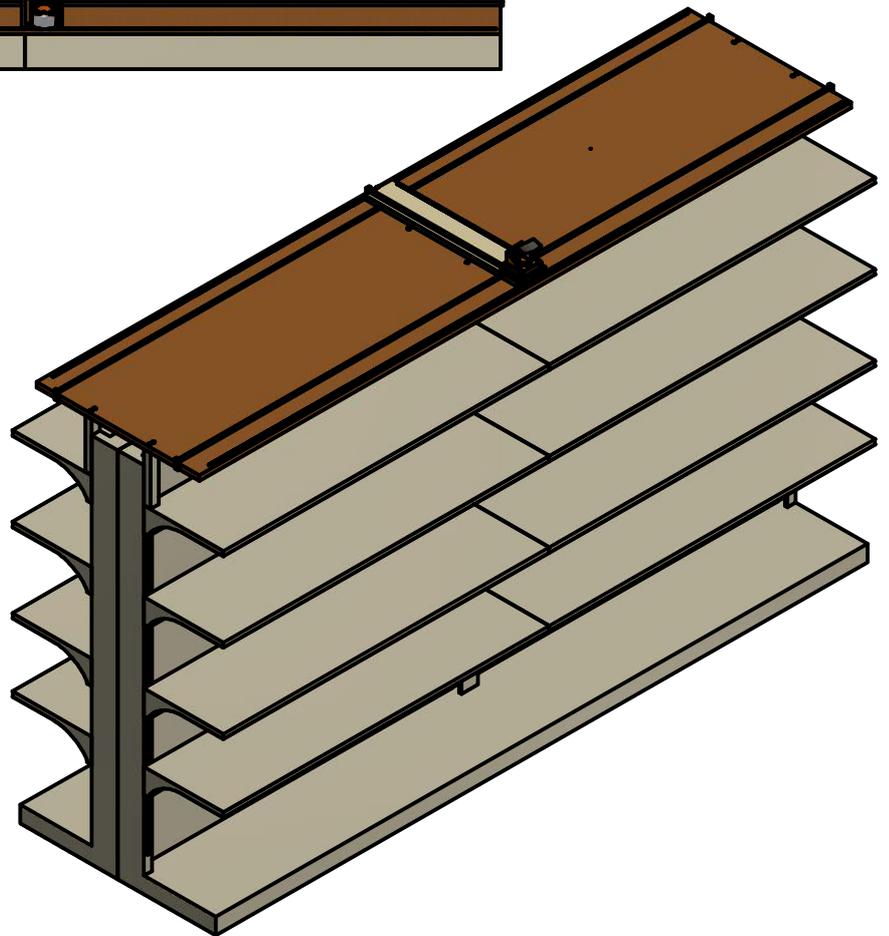
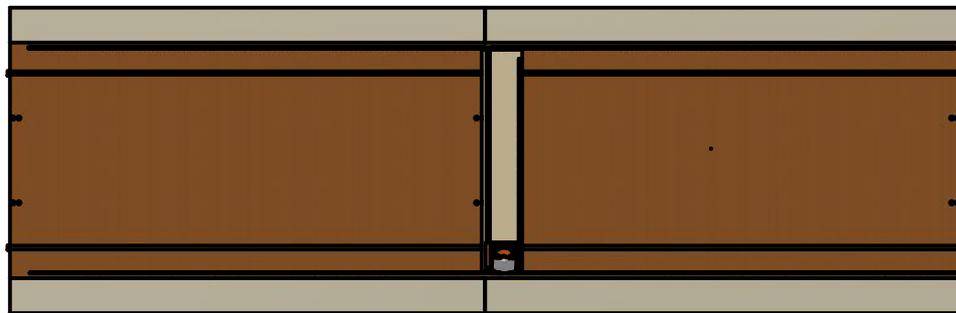
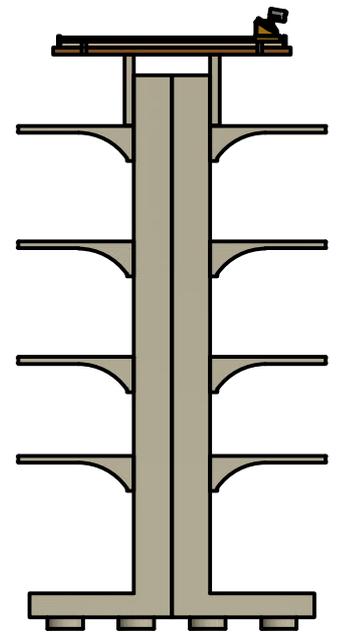
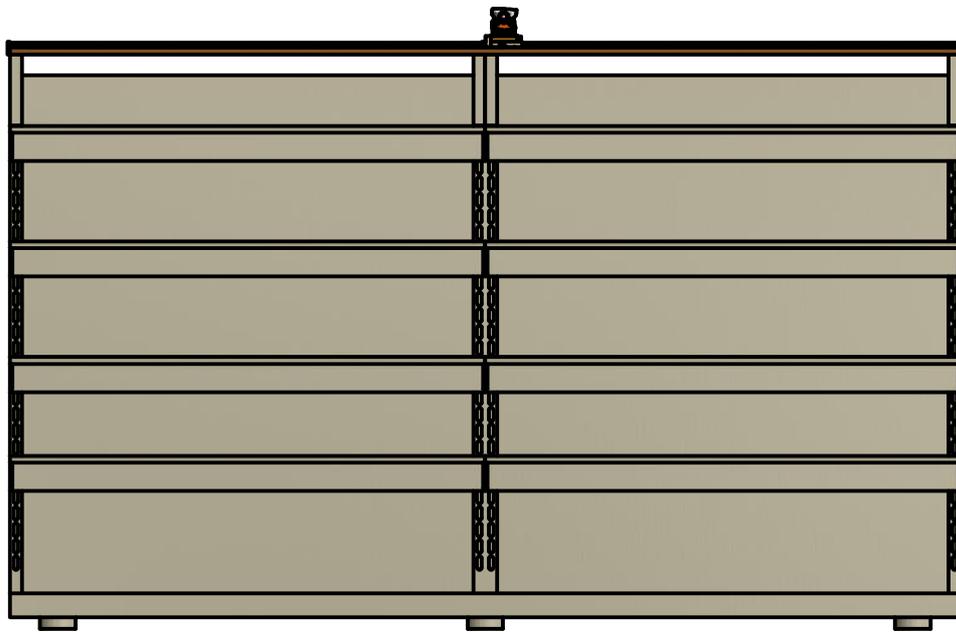


Figura D.1 Diseño de conexiones de la placa PCB.

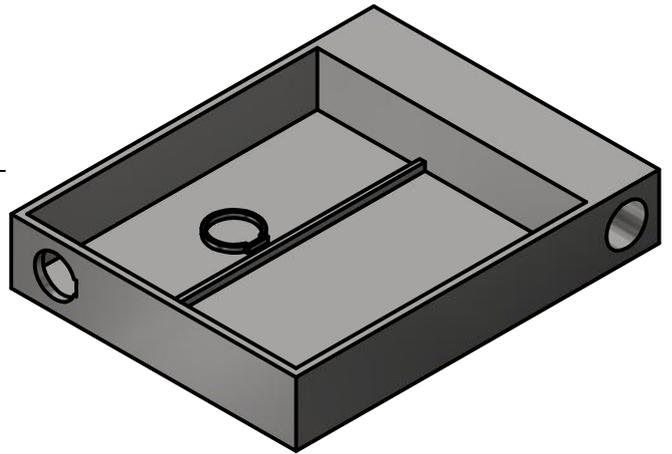
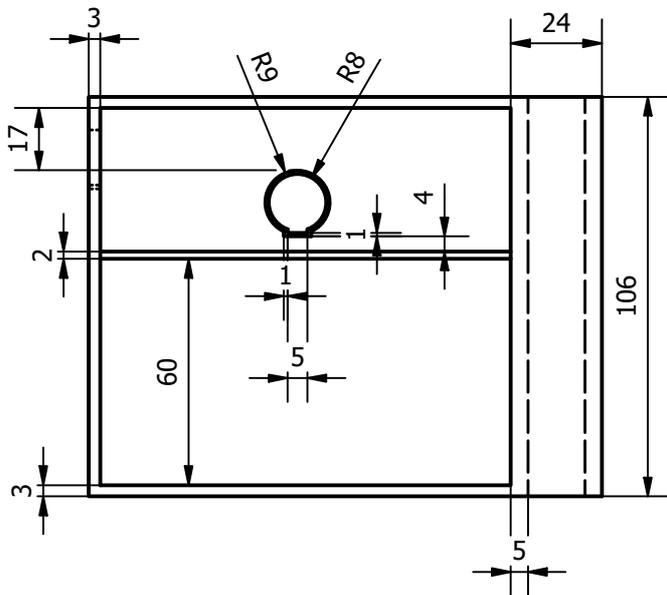
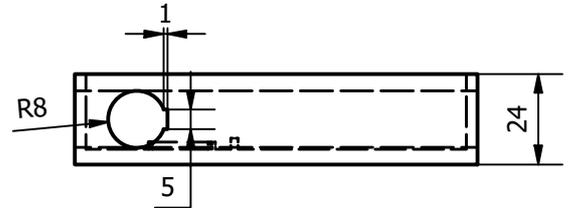
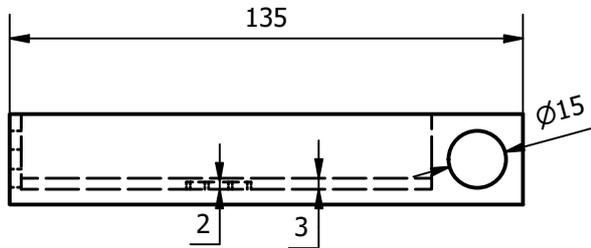


ATMEGA2560

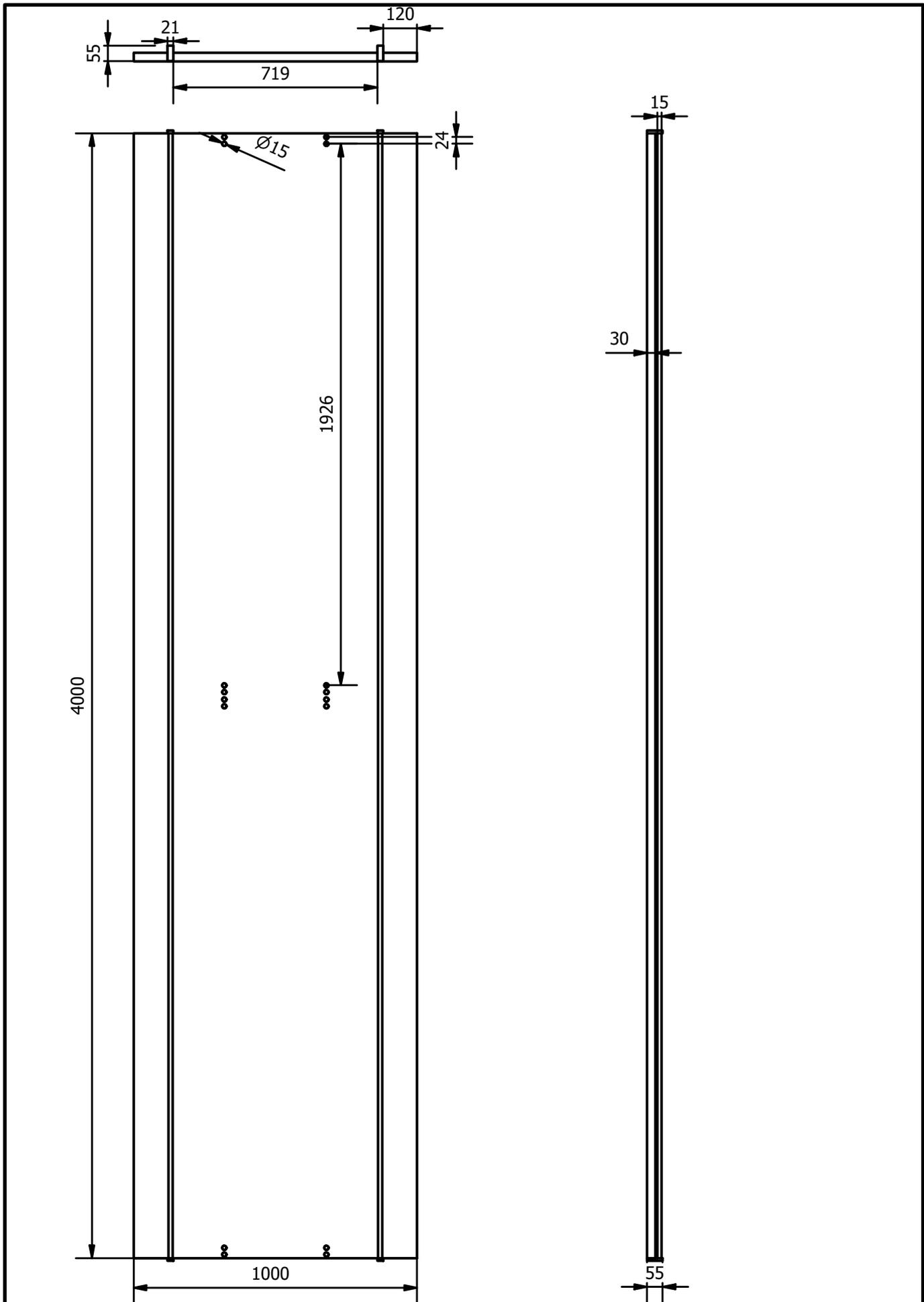




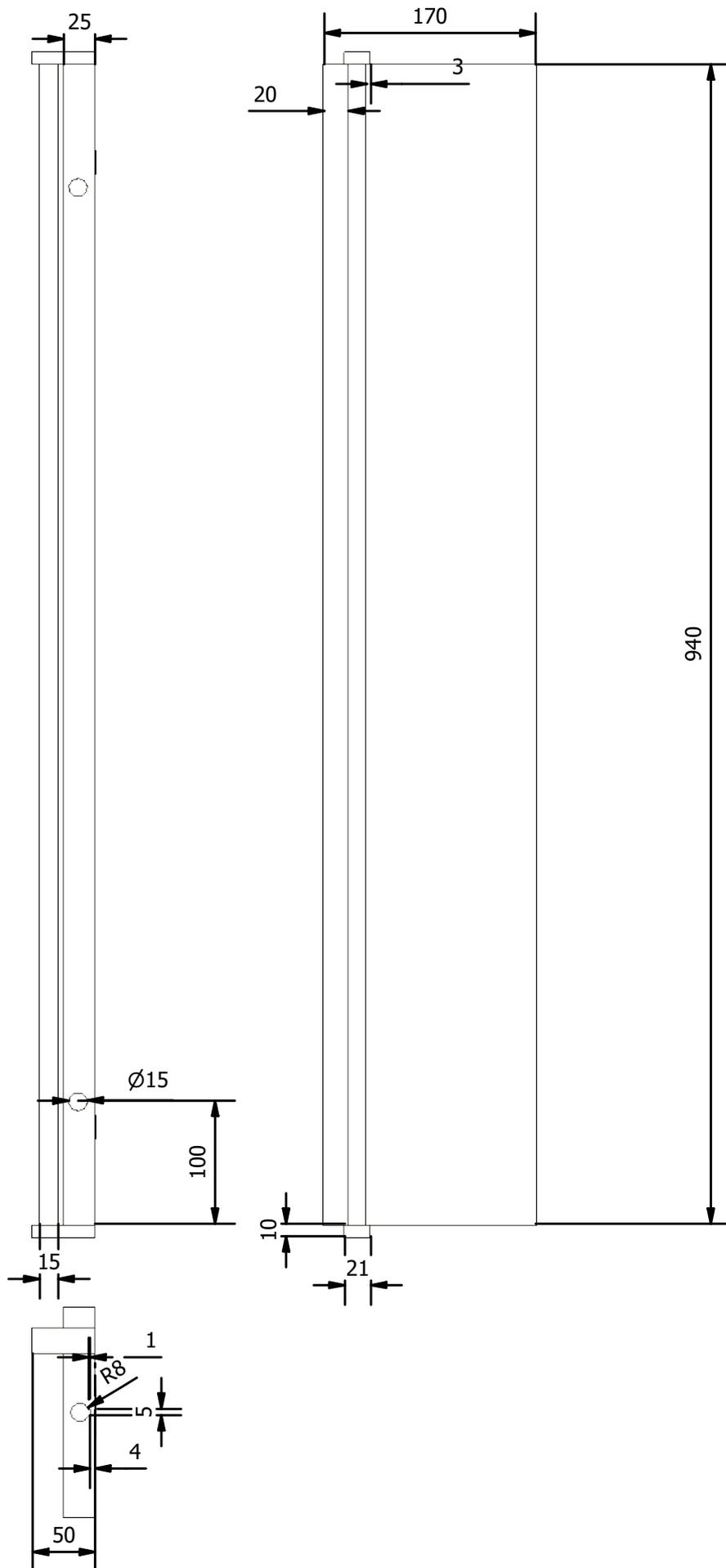
	Facultad de Ingeniería en Mecánica y Ciencias de la Producción	Dib.	José Moreno, Juan Vera	Escala 1 : 32
		Rev.	Marcelo Fajardo-Pruna, Ph.D.	
		Aprob.	Marcelo Fajardo-Pruna, Ph.D.	
Estantería	Diseño de sistema móvil	Material	AISI 430	Fecha
		Unidades	mm	01/01/2021



	Facultad de Ingeniería en Mecánica y Ciencias de la Producción	Dib.	José Moreno, Juan Vera		Escala  1 : 2
		Rev.	Marcelo Fajardo-Pruna, Ph.D.		
		Aprob.	Marcelo Fajardo-Pruna, Ph.D.		
Caja		Diseño del sistema móvil	Material	AISI 430	Fecha 01/01/2021
			Unidades	mm	



<b>espol</b>	Facultad de Ingeniería en Mecánica y Ciencias de la Producción	Dib.	José Moreno, Juan Vera	Escala  1 : 18
		Rev.	Marcelo Fajardo-Pruna, Ph.D.	
		Aprob.	Marcelo Fajardo-Pruna, Ph.D.	
Plataforma base	Diseño del sistema móvil	Material	AISI 430	Fecha
		Unidades	mm	01/01/2021



	Facultad de Ingeniería en Mecánica y Ciencias de la Producción	Dib.	José Moreno, Juan Vera	Escala 1 : 5
		Rev.	Marcelo Fajardo-Pruna, Ph.D.	
		Aprob.	Marcelo Fajardo-Pruna, Ph.D.	
Plataforma del riel	Diseño del sistema móvil	Material	AISI 430	Fecha 01/01/2021
		Unidades	mm	

# APÉNDICE E

## Proceso del software de detección de objetos

Al ejecutar el programa, este sigue un número de pasos (Figura E.1), inicialmente el software de detección ingresa a la ruta del video como a su vez a la del modelo entrenado, es así como este hace uso de la red neuronal convolucional obtenida del aprendizaje realizado en el entrenamiento (sección 3.5) lo que hace que así se pueda dividir los frames en regiones y así identificar la probabilidad de detección de un objeto en base a lo que le algoritmo ha aprendido. Una vez realizado este reconocimiento el software, esté valida dos condiciones importantes: la primera es que el objeto se encuentre dentro del rango establecido y la segunda condición a cumplir deberá ser que el producto detectado sea el correcto para dicha ubicación. En el caso de tener las dos condicionales correctas el recuadro del objeto será de color verde, sino este contorno se podrá observa de color rojo.

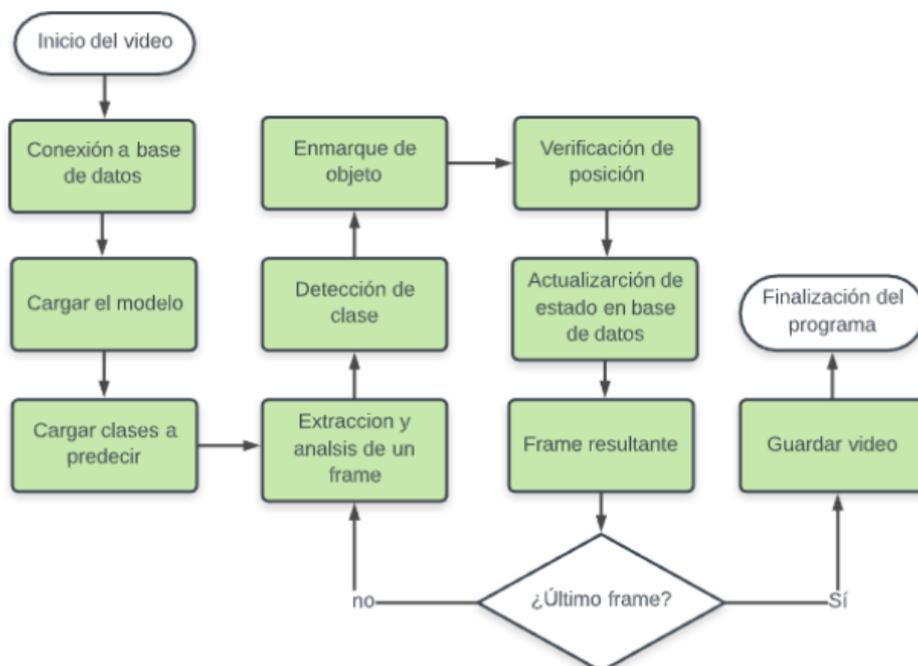


Figura E.1 Flujograma del software de monitoreo.

## Código del procesamiento del video

```
from __future__ import division
from models import *
from utils.utils import *
from utils.datasets import *
import os
import sys
import argparse
import cv2
from PIL import Image
import torch
from torch.autograd import Variable

def Convertir_RGB(img):
    # Convertir Blue, green, red a Red, green, blue
    b = img[:, :, 0].copy()
    g = img[:, :, 1].copy()
    r = img[:, :, 2].copy()
    img[:, :, 0] = r
    img[:, :, 1] = g
    img[:, :, 2] = b
    return img

def Convertir_BGR(img):
    # Convertir red, blue, green a Blue, green, red
    r = img[:, :, 0].copy()
    g = img[:, :, 1].copy()
    b = img[:, :, 2].copy()
    img[:, :, 0] = b
    img[:, :, 1] = g
    img[:, :, 2] = r
    return img

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("--image_folder", type=str, default="data/samples", help="path to dataset")
    parser.add_argument("--model_def", type=str, default="config/yolov3.cfg", help="path to model definition file")
    parser.add_argument("--weights_path", type=str, default="weights/yolov3.weights", help="path to weights file")
    parser.add_argument("--class_path", type=str, default="data/coco.names", help="path to class label file")
    parser.add_argument("--conf_thres", type=float, default=0.8, help="object confidence threshold")
    parser.add_argument("--webcam", type=int, default=1, help="Is the video processed video? 1 = Yes, 0 == no" )
```

```

parser.add_argument("--nms_thres", type=float, default=0.4, help="iou threshold for non-maximum suppression")
parser.add_argument("--batch_size", type=int, default=1, help="size of the batches")
parser.add_argument("--n_cpu", type=int, default=0, help="number of cpu threads to use during batch generation")
parser.add_argument("--img_size", type=int, default=416, help="size of each image dimension")
parser.add_argument("--directorio_video", type=str, help="Directorio al video")
parser.add_argument("--checkpoint_model", type=str, help="path to checkpoint model")
opt = parser.parse_args()
print(opt)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print("cuda" if torch.cuda.is_available() else "cpu")
model = Darknet(opt.model_def, img_size=opt.img_size).to(device)

if opt.weights_path.endswith(".weights"):
    model.load_darknet_weights(opt.weights_path)
else:
    model.load_state_dict(torch.load(opt.weights_path))

model.eval()
classes = load_classes(opt.class_path)
Tensor = torch.cuda.FloatTensor if torch.cuda.is_available() else torch.FloatTensor
if opt.webcam==1:
    cap = cv2.VideoCapture(0)
    out = cv2.VideoWriter('output.mp4',cv2.VideoWriter_fourcc('M','J','P','G'), 10, (1280,960))
else:
    cap = cv2.VideoCapture(opt.directorio_video)
    # frame_width = int(cap.get(3))
    # frame_height = int(cap.get(4))
    out = cv2.VideoWriter('outp.mp4',cv2.VideoWriter_fourcc('M','J','P','G'), 10, (1280,960))
colors = np.random.randint(0, 255, size=(len(classes), 3), dtype="uint8")
a=[]
while cap:
    ret, frame = cap.read()
    if ret is False:
        break
    frame = cv2.resize(frame, (1280, 960), interpolation=cv2.INTER_CUBIC)
    #LA imagen viene en Blue, Green, Red y la convertimos a RGB que es la entrada que requiere el modelo
    RGBimg=Convertir_RGB(frame)
    imgTensor = transforms.ToTensor()(RGBimg)
    imgTensor, _ = pad_to_square(imgTensor, 0)
    imgTensor = resize(imgTensor, 416)
    imgTensor = imgTensor.unsqueeze(0)
    imgTensor = Variable(imgTensor.type(Tensor))

```

```

with torch.no_grad():
    detections = model(imgTensor)
    detections = non_max_suppression(detections, opt.conf_thres, opt.nms_thres)

for detection in detections:
    if detection is not None:
        detection = rescale_boxes(detection, opt.img_size, RGBimg.shape[:2])
        for x1, y1, x2, y2, conf, cls_conf, cls_pred in detection:
            box_w = x2 - x1
            box_h = y2 - y1

            color = [int(c) for c in colors[int(cls_pred)]]

            print("Se detectà,£à, " {} en X1: {}, Y1: {}, X2: {}, Y2: {}".format(classes[int(cls_pred)], x1, y1, x2, y2))
            print(color)
            frame = cv2.rectangle(frame, (x1, y1 + box_h), (x2, y1), color, 5)

            cv2.putText(frame, classes[int(cls_pred)], (x1, y1), cv2.FONT_HERSHEY_SIMPLEX, 1, color, 5)# Nombre
de la clase detectada
            cv2.putText(frame, str("%.2f" % float(conf)), (x2, y2 - box_h), cv2.FONT_HERSHEY_SIMPLEX, 0.5,color,
5) # Certeza de prediccion de la clase
            #
            #Convertimos de vuelta a BGR para que cv2 pueda desplegarlo en los colores correctos

if opt.webcam==1:
    #cv2.imshow('frame', Convertir_BGR(RGBimg))
    out.write(RGBimg)
else:
    out.write(Convertir_BGR(RGBimg))
    #cv2.imshow('frame', RGBimg)
#cv2.waitKey(0)

if cv2.waitKey(25) & 0xFF == ord('q'):
    break
out.release()
cap.release()
cv2.destroyAllWindows()

```