

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIERO ELECTRÓNICO**

**TEMA:
AUTOMATIZACIÓN DEL PROCESO DE POSICIONAMIENTO DE LINER
EN TAPAS PLÁSTICAS PARA MEJORAR LA PRODUCTIVIDAD EN LA
EMPRESA IEPESA EN QUITO**

**AUTOR:
RICARDO GERMAN BRICEÑO BRACERO**

**TUTOR:
HAMILTON LEONARDO NÚÑEZ VERDEZOTO**

Quito, marzo 2019

CESIÓN DE DERECHOS DE AUTOR

Yo, Ricardo German Briceño Bracero, con documento de identificación N° 172197279-0, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: “AUTOMATIZACIÓN DEL PROCESO DE POSICIONAMIENTO DE LINER EN TAPAS PLÁSTICAS PARA MEJORAR LA PRODUCTIVIDAD EN LA EMPRESA IEPESA EN QUITO”, mismo que ha sido desarrollado para optar por el título de Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
Ricardo German Briceño Bracero
C.I.: 1721972790

Quito, marzo del 2019.

DECLARACIÓN DE COUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Trabajo de Titulación: “AUTOMATIZACIÓN DEL PROCESO DE POSICIONAMIENTO DE LINER EN TAPAS PLÁSTICAS PARA MEJORAR LA PRODUCTIVIDAD EN LA EMPRESA IEPESA EN QUITO”, realizado por Ricardo German Briceño Bracero, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerados como trabajo final de titulación.

Quito, marzo 2019



.....
Hamilton Leonardo Núñez Verdezoto
C.I.: 1712981487

DEDICATORIA

Dedico este Trabajo de Titulación a mis padres, hermanos y familiares por todo el apoyo y confianza depositada en mi persona para poder sacar adelante mi carrera profesional.

Dedico a mis seres queridos que ya no están en este mundo pero sé que desde el cielo me han sabido guiar a lo largo de todo el trayecto de mi vida.

AGRADECIMIENTOS

Agradezco a la Universidad Politécnica Salesiana por haberme inculcado todos los conocimientos profesionales y haber sembrado la semilla del saber y del conocimiento.

Agradezco al Ingeniero Hamilton Núñez, MSc. por haberme sabido guiar con paciencia y sabiduría a lo largo del presente Trabajo de Titulación.

ÍNDICE DE CONTENIDO

CESIÓN DE DERECHOS DE AUTOR.....	i
DECLARACIÓN DE COUTORÍA DEL DOCENTE TUTOR.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTOS	iv
ÍNDICE DE CONTENIDO.....	v
RESUMEN	xi
ABSTRACT	xii
INTRODUCCIÓN	xiii
CAPÍTULO 1 ANTECEDENTES.....	1
1.1. Planteamiento del Problema	1
1.2. Justificación	1
1.3. Objetivos.....	2
1.3.1. Objetivo General	2
1.3.2. Objetivo Específico	2
1.4. Metodología.....	2
1.5. Beneficiario.....	3
CAPÍTULO 2 FUNDAMENTACIÓN TEÓRICA	4
2.1. Material de Fabricación.....	4
2.2. Polietileno	4
2.3. Polipropileno.....	4
2.4. Poliestireno	5
2.5. Liner o sellos de seguridad	5
2.5.1. Sellos de Inducción.....	5
2.5.2. Sellos de barrera y temperatura	6
2.6. Inyección de plásticos	6

2.7. Ciclos de Moldeo	6
2.8. Automatización	7
2.9. Controladores Embebidos.....	8
2.10. Visión Artificial	9
2.11. Aplicaciones de Visión Artificial.....	9
2.12. Productividad	10
2.13. Interfaz Hombre Máquina.....	10
2.14. Proceso manual de colocación de <i>liner</i>	11
CAPÍTULO 3 DISEÑO E IMPLMETACIÓN.....	12
3.1. Características de la máquina de posicionador de <i>liner</i>	12
3.2. Diagrama de bloques del proceso colocador de <i>liner</i>	13
3.3. La banda transportadora	14
3.4. Cámara.....	15
3.5. Soporte del sistema de visión artificial.....	16
3.6. Tolva.....	17
3.7. Tablero electro neumático y electrónico	17
3.8. Diseño Electrónico	18
3.8.1. Protección de la máquina colocadora de <i>liner</i>	18
3.9. Algoritmo de control de la Raspberry	20
3.10. Software y Firmware necesarios	22
3.10.1. Software para Windows	22
3.10.2. Firmware y software para Raspberry.....	22
3.11. Diseño HMI	22
3.12. Instrumentación del proceso	24
3.13. Diseño Electro-Neumático.....	25
3.14. Tablero Interno Electrónico de control.....	28
3.15. Visión Artificial	31

3.16. Interpretación de las imágenes luego del procedimiento.....	37
3.17. Hardware de la maquina colocadora de liner.....	39
CAPÍTULO 4 PRUEBAS Y RESULTADOS	41
4.1. Procedimiento	41
4.2. Prueba de ingreso al sistema.....	42
4.3. Pruebas de la etapa de transporte	42
4.4. Pruebas de visión artificial.....	43
4.5. Prueba de tapas en posición incorrecta.....	44
4.6. Pruebas de la etapa de detección de tapas en posición correcta	44
4.7. Pruebas de desecho de tapas por falla de inyección.....	45
4.8. Pruebas de colocadora de <i>liner</i>	46
4.9. Pruebas del HMI	47
4.10. Materia Prima.....	48
4.11. Funcionamiento.....	48
4.12. Producción manual	50
4.13. Producción automática	51
4.14. Contraste de producción automática Vs manual.....	51
CONCLUSIONES	53
RECOMENDACIONES	55
REFERENCIAS	56
ANEXOS	

ÍNDICE DE FIGURAS

Figura 2.1 Liner y Selladores	5
Figura 2.2 Máquina Inyectora	7
Figura 2.3 Proceso manual de colocación de <i>liner</i>	11
Figura 3.1 Partes físicas de la máquina colocadora de liner	12
Figura 3.2 Diagrama de bloques colocador de liner	14
Figura 3.3 Banda transportadora horizontal y vertical.....	15
Figura 3.4 Soporte y Cámara.....	15
Figura 3.5 Soporte de visión artificial.....	16
Figura 3.6 Tolva	17
Figura 3.7 Tablero electro neumático y electrónico	18
Figura 3.8 Diagrama de flujo de programación en la Raspberry	21
Figura 3.9 Componentes del HMI	23
Figura 3.10 Diagrama de flujo del HMI	24
Figura 3.11 Diagrama de Instrumentación.....	25
Figura 3.12 Diseño neumático.....	26
Figura 3.13 Diseño Eléctrico.....	27
Figura 3.14 Diseño del interior del tablero de control.....	28
Figura 3.15 Esquema electrónico de entradas y salidas digitales de la Raspberry.....	29
Figura 3.16 Diagrama de Flujo Visión Artificial	32
Figura 3.17 Continuación de diagrama de flujo de visión artificial	33
Figura 3.18 Frame Gris (A) y filtro Gaussiano (B)	34
Figura 3.19 Resta (A) y Umbral (B)	35
Figura 3.20 Contorno, Erosión, Dilatación	36
Figura 3.21 Área de Tapas	37
Figura 3.22 Etapa de transporte, Etapa de visión artificial y HMI	39
Figura 3.23 Etapa de Rechazo	39
Figura 3.24 Etapa de Reciclaje	40
Figura 3.25 Etapa de colocación liner y Tablero electro-neumático y electrónico	40
Figura 4.1 Prueba de Ingreso al sistema	41
Figura 4.2 Etapa de transporte.....	42
Figura 4.3 Pruebas Visión Artificial	43

Figura 4.4 Pruebas de la etapa de tapas en posición incorrecta.....	44
Figura 4.5 Pruebas de la etapa de detección de tapas en posición correcta	44
Figura 4.6 Pruebas de tapa por falla de inyección.....	45
Figura 4.7 Prueba etapa de colocación de liner	46
Figura 4.8 Pruebas de funcionamiento HMI	47
Figura 4.9 Materia Prima Liner (B) y Tapa(A)	48
Figura 4.10 Comparación del tiempo de procesamiento en función de la resolución en las diferentes plataformas	50

ÍNDICE DE TABLAS

Tabla 2.1 Características de controladores embebidos	9
Tabla 3.1 Elementos y características físicas de la máquina.....	13
Tabla 3.2 Características de la Cámara	16
Tabla 3.3 Características de equipos eléctricos y electrónicos.....	19
Tabla 3.4 Items de la Pantalla HMI	23
Tabla 3.5 Simbología diagrama de instrumentación	25
Tabla 3.6 Simbología del diseño neumático	26
Tabla 3.7 Elementos del tablero electrónico-neumático	27
Tabla 3.8 Simbología de las Salidas digital de la Raspberry	30
Tabla 3.9 Elementos del tablero electrónico	30
Tabla 3.10 Áreas de tapas en posición correcta e incorrecta	38
Tabla 4.1 Resolución de Pantalla Vs FPS	49
Tabla 4.2 Colocación manual de Liner en Industrias IEPESA	50
Tabla 4.3 Producción automática de la máquina colocadora de <i>liner</i>	51
Tabla 4.4 Contraste de producción manual VS automático	51

RESUMEN

La automatización del proceso de calidad y colocación de *liner* en tapas se realiza mediante un controlador embebido y distintos equipos de control en el área de la línea alimenticia para obtener productos terminados con asepsia, además la mejora de la productividad se verá mejorada en calidad y número de tapas por hora. Uno de los principales controladores embebidos es la Raspberry PI que ofrece los distintos requerimientos técnicos para la adecuada y eficiente automatización del proceso, entre unas de sus características se tiene soporte técnico, bajo coste, software libre, cámara para digitalización de imágenes entre otros, en comparación de controladores de alta gama. La Raspberry PI se encargará del procesamiento digital de imágenes para una adecuada verificación de la tapa boca arriba y boca abajo, además el control de las tapas por deformación geométrica y de color para de esta manera garantizar que en la etapa de la colocación del *liner* la tapa cumpla con los requerimientos. Se realizará un monitoreo mediante un HMI para visualizar el conteo de tapas, el escaneo de las tapas e indicadores de los elementos electromecánicos en cada etapa del proceso, de esta manera llevar un historial de la producción en tiempo real.

ABSTRACT

The automation of the process of quality and placement of *liner* in lids will be carried out by means of an embedded controller and different final control equipment in the area of the food line to obtain finished products under the appropriate asepsis, in addition the improvement of the productivity will be improved in quality and number of tapas per hour. One of the main embedded controllers is the raspberry pi 3b that offers the different technical requirements for the adequate and efficient automation of the process, among some of its features is technical support, low cost, free software, camera for digitization of images among others, in comparison of high-end controllers. The Raspberry PI that will be responsible for the digital processing of images for an adequate verification of the cover face up and face down, in addition the control of the covers by geometric deformation and color to thus ensure that in the stage of the placement of the liner the lid meets the requirements of the customer request. A monitoring will be carried out by means of an HMI to visualize the count of caps, the scanning of the caps and indicators of the electromechanical elements in each stage of the process, in this way to keep a history of the production.

INTRODUCCIÓN

En la industria de inyección de plásticos orientada a la producción de piezas de línea alimenticia se necesita que el producto final cumpla con el control de calidad requerido por el cliente, en este caso los parámetros geométricos y deformaciones de la tapa plástica para colocar el *liner*. La tapa plástica es fabricada en base a la materia prima de polímeros cuyas propiedades físicas permiten plasticidad y dureza cumpliendo con los distintos requerimientos del cliente.

El control de calidad en la producción de piezas plásticas de línea alimenticia se lo realiza de forma sistemática. El presente estudio se basa en implementar un control de inspección visual para lograr un control de calidad y mejora tiempos de producción. La percepción del ojo humano no es suficiente para realizar el control de calidad de la pieza plástica debido a que después de un cierto tiempo el operador sufre de cansancio, por esta razón es indispensable el procesamiento digital de imágenes para obtener un mayor grado de eficiencia para mejora de la calidad de la pieza plástica.

En el capítulo 1 se detalla el planteamiento del problema y los distintos objetivos a realizar en el presente proyecto de acuerdo a los requerimientos del beneficiario.

En el capítulo 2 se realizará una investigación teórica de distintos factores que intervienen en la construcción, sellado y el colocado del *liner* de la tapa.

En el capítulo 3 se procederá a diseñar e implementar el sistema eléctrico, electro neumático y de visión artificial de forma integral con la finalidad de poder realizar a continuación las distintas pruebas de funcionamiento.

En el capítulo 4 se desarrolla las pruebas en cada etapa de la máquina evidenciándolas con fotos y videos para de esta manera detallar los resultados obtenidos.

En el capítulo 5 se redactará las distintas conclusiones y recomendaciones en base de las distintas pruebas y los resultados que se hayan hecho, durante y después de la construcción de la máquina.

CAPÍTULO 1

ANTECEDENTES

1.1. Planteamiento del Problema

Industria IEPESA, en la actualidad, no cuenta con una máquina automatizada para la colocación de *liner* en la cavidad de las tapas, por lo cual se crean cuellos de botella a pesar de que existen dos operarios para el proceso.

Al existir un turno de trabajo completo con pocas pausas activas hacia los operarios, el nivel de exigencia física se incrementa con el transcurso de las horas que requiere el proceso, Además las condiciones ergonómicas del operador no son ideales, ya que para cumplir la jornada laboral él mismo permanece de pie las ocho horas por turno.

También se considera que uno de los aspectos importantes que es la asepsia en el proceso de producción, ya que el operador está en contacto directo con el producto de línea alimenticia, a pesar del uso de equipos de protección personal.

Existen varios factores externos en el proceso de manufactura, como la descarga en la tolva de materia prima sobre la inyectora, por lo que puede producirse una pausa y descalibración del proceso de inyección provocando que las tapas plásticas salgan incompletas o deformes.

La inspección visual del ojo humano en el reconocimiento de las tapas salidas de la inyectora no ha resultado suficiente para controlar la calidad del producto. Además la velocidad de inspección para la colocación del *liner* en la actualidad, no permite reducir los tiempos de ciclo y por ende no permite incrementar la producción por turno, convirtiendo esta secuencia en un cuello de botella.

1.2. Justificación

Para mejorar la productividad del proceso se requiere automatizar el proceso de colocación de *liner* dentro de la cavidad de las tapas plásticas, donde resulta imperante eliminar cuellos de botella, de esta forma además poder mejorar la condición de

asepsia evitando la manipulación de las tapas, ya que se trata de un proceso de grado alimenticio.

Cabe recalcar que los turnos de jornada laboral del operario son de ocho horas diarias, por lo que un sistema automatizado ayudará a mejorar las condiciones de ergonomía del mismo.

1.3. Objetivos

1.3.1. Objetivo General

Automatizar el proceso de colocación de *liner* en tapas plásticas para mejorar la asepsia y reducir los cuellos de botella en la Empresa IEPESA Quito.

1.3.2. Objetivo Específico

Caracterizar el proceso de colocación del *liner* en la cavidad de la tapa plástica para determinar las variables de control involucradas en el proceso.

Desarrollar un sistema de control automático para la colocación del *liner* en la cavidad de la tapa plástica.

Desarrollar un sistema de visión artificial para identificar que las tapas se encuentren completas y en la posición requerida.

Realizar un análisis comparativo entre el proceso automático propuesto y el proceso manual actual para contrastar la producción de las tapas terminadas.

1.4. Metodología

Investigar cómo mejorar la productividad en la elaboración de tapas plásticas desarrollando un sistema automatizado para el proceso, para luego comparar sus resultados contrastando la producción actual del proceso manual versus el proceso automático que se propone desarrollar.

1.5. Beneficiario

La máquina colocadora de *liner* es posesión de la Empresa IEPESA, en especial el área de producción de tapas plásticas ayudando a la mejorar de productividad maximizando el rendimiento de la máquina verificando, identificando y visualizando defectos antes de añadir el valor al producto final cumpliendo con las expectativas del cliente.

CAPÍTULO 2

FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se estudiará la visión artificial, como inspección total de la productividad para mejorar la eficiencia de calidad en la tapa de tal forma incrementar la producción de tapas de acuerdo a los parámetros requeridos por el cliente. Además, se desarrollará un sistema automatizado que permita realizar el control de tapas que se encuentren en la posición correcta para colocar *liner*, verificación de tapas con o sin defectos de inyección y un mecanismo que permita la colocación del *liner* en la tapa.

2.1. Material de Fabricación

El proceso de colocación de *liner*, inicialmente, empezará en una máquina de inyección de plásticos, donde se tendrá que considerar el tipo de polímero que será el idóneo para la fabricación de las tapas.

Existe una variedad de materia prima en la industria de inyección de plástico, todos llamados con el término termoplásticos pues una vez que han sido sometidos a un calentamiento y se produce la fusión y posterior enfriamiento del material, que lo endurece, entre los más importantes se considera: (Sanchez, 2014)

2.2. Polietileno

Es el polímero más simple de obtener químicamente inerte y se obtiene de la polimerización del etileno. Sus propiedades físicas, químicas y mecánicas en forma de gránulos, blanquecina y translúcida lo hacen el plástico más consumido en el mundo. (Roca, 2005)

2.3. Polipropileno

Es un polímero termoplástico que se obtiene de polimerización del propileno. Es un termoplástico que se le conoce con las siglas PP, es plástico duro y resistente a altas temperaturas que se suaviza a 150 °C. (Corso, 2016)

2.4. Poliestireno

Polímero termoplástico que se obtiene de la polimerización del estireno. Existen cuatro tipos principales: el poliestireno cristal que es transparente, rígido y quebradizo; el poliestireno de alto impacto, de gran resistencia y opacidad; el poliestireno expandido, muy ligero y el poliestireno extruido, similar al expandido pero más denso e impermeable. (Ecoplas, 2011)

2.5. Liner o sellos de seguridad

Estos sellos tienen la especial característica de ser plásticos espumados, permiten la impermeabilidad del recipiente, es decir que no se podrá derramar lo envasado ajustando la tapa a los errores que se encuentran en los cuellos de las botella. (PILONI, 2017)

En la Figura 2.1 se observa distintos Liner y selladores

Figura 2.1 Liner y Selladores



Fuente: (PILONI, 2017)

2.5.1. Sellos de Inducción

Estos sellos que se activan por una frecuencia que calienta una lámina de aluminio y a su vez un polímero que se sella ante el envase. Sirven para evitar filtraciones durante el almacenamiento y transporte que puede soportar una considerable presión, fácil para el retiro del usuario. (Enercon, 2013)

2.5.2. Sellos de barrera y temperatura

Sin afectar la imagen del producto, es posible añadir a los sellos elementos de alta barrera, como finos de alta protección contra oxígeno y humedad. Se usa para aplicaciones en caliente o pasteurización y esterilización, existen sellos diseñados para resistir altas temperaturas (PILONI, 2017)

2.6. Inyección de plásticos

La fabricación de piezas plásticas por inyección se da cuando la materia prima es insertada en la tolva de la inyectora siendo fundida a altas temperaturas, luego es inyectado en un molde a alta presión donde el material es enfriado, solidificado y luego liberado al abrirse las dos mitades del molde dando como resultado las piezas plásticas. La inyección de plásticos es utilizada para la producción de piezas plásticas, en especial, para la industria alimenticia y farmacéutica donde se tiene que tomar muy en cuenta la asepsia y la calidad del producto final.

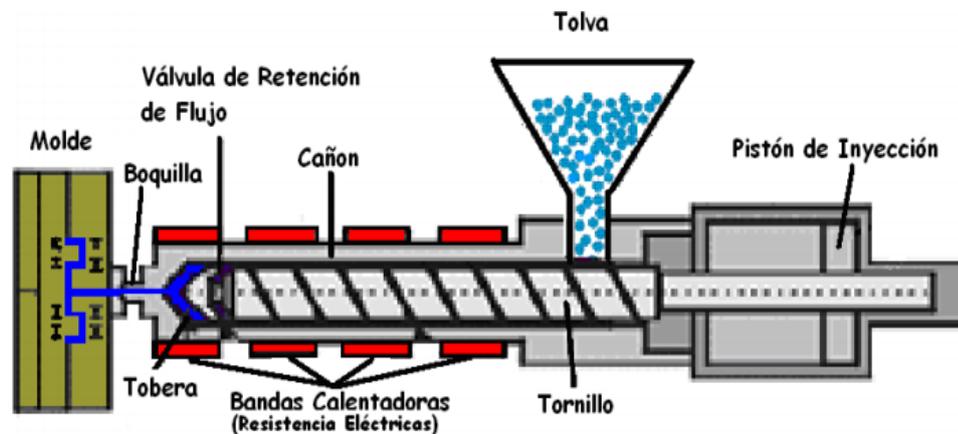
“El moldeo por inyección es una técnica muy conocida sobre todo en la fabricación con materiales poliméricos. Brinda un modo relativamente simple de producir componentes con geometrías de alta complejidad, con un coste e inversión de tiempo muy baja.” (Sanchez, 2014)

En la Figura 2.2 se observa la máquina inyectora de plásticos.

2.7. Ciclos de Moldeo

- a. Se cierra el molde.
- b. Se calienta para plastificar el material, manteniendo la temperatura en el cañón.
- c. Se empuja el material caliente hacia la cavidad del molde.
- d. El tornillo mantiene la presión hasta que se enfría el plástico.
- e. El tornillo retrocede para recoger material nuevo de la tolva y plastificar nuevamente.
- f. Se abre el molde y se extrae la pieza

Figura 2.2 Máquina Inyectora



Fuente: (Garavito, 2007)

2.8. Automatización

La automatización es una disciplina que abarca varias especialidades y que, por lo tanto, recurre a conocimientos y diversos procesos de ingeniería. Se define un autómatas como un sistema artificial que se comporta de determinadas maneras relacionando comandos de entrada con estados del sistema, con el fin de obtener las salidas necesarias para solucionar tareas. (F. Ebel, 2008)

En la actualidad, cuando se habla del concepto técnico de la automatización, se piensa en robots industriales y en sistemas de control mediante computadores. En un inicio la técnica de la automatización inicio mucho antes en los talleres artesanales y en las plantas industriales.

Para configurar procesos automáticos modernos se necesitan tres componentes:

- Sensores para captar los estados del sistema
- Actuadores para emitir los comandos de control
- Unidades de control para la ejecución del programa y para tomar decisiones

Una de las principales razones para el automatismo fue y sigue siendo la necesidad de generar costos bajos sin dejar de lado la eficacia en el proceso y poder ser competitivos en el campo industrial.

La técnica de la automatización contribuye a ese fin de varias maneras:

- Debido a los distintos requerimientos del cliente la manufactura precisa, de una automatización en busca de la mejora el control de calidad.
- Los tiempos muertos se ven reducidos significativa debido a que se producen menos interrupciones que en la línea de producción manual.
- El cambio de herramientas o reprogramación de las máquinas añade flexibilidad al proceso de manufactura. Todos estos ahorros se reflejan en mayores ganancias.
- La reducción de mano de obra que es reemplazada por robots de esta manera se produce un incremento de la productividad

2.9. Controladores Embebidos

La tecnología ha introducido al mercado industrial para mejoras de la productividad y por ende mejorar la competitividad. Estos sistemas embebidos son de bajo coste comparados con otro tipo de controladores ya que usa herramientas y metodología de diseño moderna basadas en software libre. (Barreño, 2012)

En la Tabla 2.1 se observa las características de varios controladores embebidos donde las principales diferencias se aprecian en el procesador (700Mhz de R-PI 1-B frente a 16Mhz de Arduino) y la memoria (512MB de R-PI 1-B frente a 0,002 MB de Arduino). Estas dos características permiten la adquisición y procesamiento de imagen, ya que requieren muchos recursos para obtener fluidez en el video. (Gonzales, 2016)

Tabla 2.1 Características de controladores embebidos

ELEMENTO	ARDUINO UNO	Raspberry PI 1-B	Raspberry PI 3-B
Velocidad de reloj	16 MHZ	700 MHZ	1,2 GHZ
# núcleos	1	1	4
RAM	0,002 MB	512MB	100MB
CPU	no	ARM1176JZF-S	ARM Cortex-A53
SOC	ATmega328P	Broadcom BCM2835	Broadcom BCM 2837
Flash	32KB	Tarjeta SD (2-16 Gb)	Tarjeta microSD (2-32 Gb)
SO	Ninguno	Distribuciones Linux	Distribuciones Linux
Cámara	NO	No	5Mp
Precio	22,88	34,19	51,29
Tamaño	7,6X1,9X6,4cm	8,6X5,4X1,7cm	8,6X5,4X1,7cm
Peso	40 gr	45 gr	45gr

Fuente: (Gonzales, 2016)

2.10. Visión Artificial

La visión artificial forma parte del estudio de la inteligencia artificial basado en el procesamiento digital de imágenes, que ha tenido un gran crecimiento en el campo industrial y comercial.

El objetivo principal de los sistemas de visión es combinar la captura de imágenes mediante una cámara con la velocidad de procesamiento de un ordenador para tomar decisiones de inspección y localización de objetos.

De esta manera las prestaciones que ofrece la visión artificial en la industria son diversas entre las más importantes cabe recalcar la calidad de los productos, mayor productividad, reducción de costes de producción, disminuir la participación humana en el proceso de fabricación en lugares peligrosos y además mejorar la ergonomía por las largas horas laborales de los trabajadores.

2.11. Aplicaciones de Visión Artificial

La visión artificial tiene diferentes aplicaciones en la industria debido a las prestaciones que se ofrece en la automatización de procesos y de esta manera manejar

patrones de control como es el de verificación, control de calidad, inspección entre otros.

2.12. Productividad

La productividad se puede definir como la utilización de los factores de producción en la generación de bienes y servicios para la sociedad, con la finalidad de mejorar la eficiencia con que son utilizados los recursos. Existen varios modelos de productividad: la productividad parcial (es la relación producida por uno solo de los elementos de producción, por ejemplo la mano de obra), la productividad total o de múltiples factores (es la relación entre la cantidad producida sobre la sumatoria de diversos factores de producción) y la productividad de valor agregado (la relación del valor agregado sobre la sumatoria de los factores de producción). (Soto, 2010)

En términos financieros, la productividad es todo el aumento de la producción que no se explica por incrementos en trabajo, capital o en cualquier otro insumo intermedio utilizado para producir. Esto se puede expresar matemáticamente como: (Galindo, 2015)

$$PIB = Productividad * f(capital, trabajo) \quad \text{Ec. (2.1)}$$

2.13. Interfaz Hombre Máquina

En los distintos procesos industriales es inherente la automatización en sus distintas fases de producción para lograr una calidad óptima en el resultado final.

Por aquello la importancia de que exista una buena comunicación entre el operario y la máquina como parte de un sistema de control de procesos automatizado, de tal manera que sólo así se podrán analizar las diferentes anomalías que puedan suceder en un intervalo de tiempo determinado; asimismo, también se podrán ajustar los diversos parámetros relacionados al proceso de control. (Edgar Ruiz Lizama, 2008)

La pantalla HMI representa el vínculo de interacción entre un usuario y un determinado hardware, para el caso de control de procesos la pantalla HMI, debe ser

mostrar al usuario datos básicos de todo sistema de control de procesos, tales como variable de proceso, variable de control y set point, todo esto presentado a tiempo real.

2.14. Proceso manual de colocación de *liner*

El proceso manual de tapas plásticas se desarrolla en turnos de ocho horas donde el operador entra a la cabina con su equipo de protección de pies, manos y boca para que la asepsia en el proceso.

Cada ciclo de la máquina proveerá de las tapas, donde los operadores están de pie esperando que la banda transportadora lleve el producto hacia su posición, donde en una mano estará el *liner* y la otra la usará para dar vuelta a la tapa, colocar el *liner* y extraer de la banda la colada o residuo de cada inyección.

La inspección de calidad de las tapas en las primeras horas de trabajo realizadas por los operadores resulta eficaz, pero al transcurrir el tiempo por falta de pausas activas, la ergonomía y cansancio visual donde la calidad del producto se ve reducido, por lo cual resulta imperante una máquina automatizada destinada para este propósito.

En la Figura 2.3 se puede observar la colocación de *liner* de forma manual

Figura 2.3 Proceso manual de colocación de *liner*



Elaborador por: Ricardo Briceño

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN

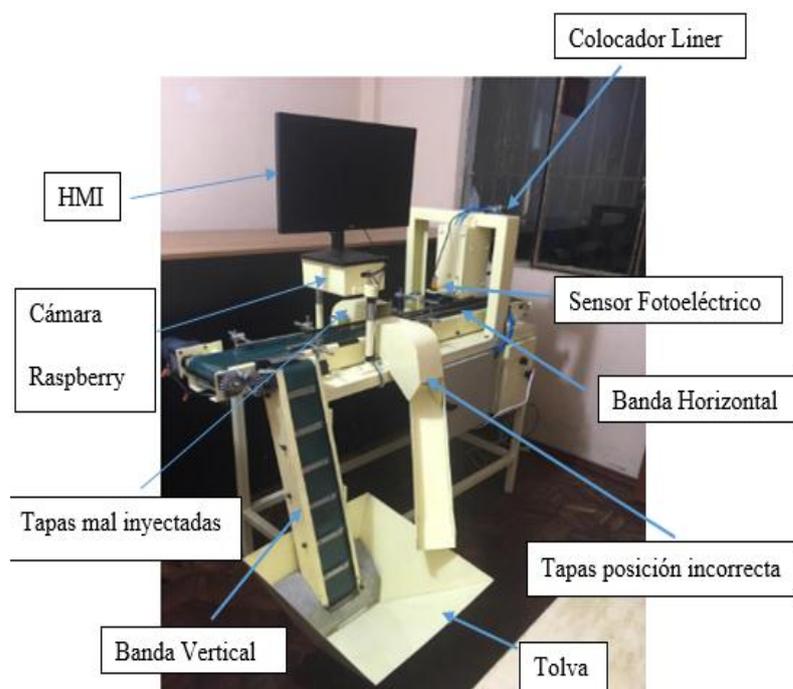
En el presente capítulo se procederá al diseño e implementación de todo el sistema de la máquina colocadora de *liner*, el diseño electrónico y el del HMI de dicha máquina. Para poder llevar a cabo el cometido será indispensable revisar de forma complementaria los aspectos mecánicos para poder acoplar eficientemente el sistema eléctrico y de control del equipo a automatizar.

3.1. Características de la máquina de posicionador de *liner*

La máquina de posicionamiento de *liner* se encargará de rechazar, retroalimentar y colocar la tapa en la posición deseada. Para este proceso se utilizará: una banda transportadora, una tolva, un soporte para el sistema de visión artificial, estructura de soporte, una caja eléctrica y caja de control.

En la Figura 3.1 se observa las partes físicas de la máquina colocadora de *liner*

Figura 3.1 Partes físicas de la máquina colocadora de *liner*



Elaborado por: Ricardo Briceño

Tabla 3.1 Elementos y características físicas de la máquina

Elemento	Características			
Tolva	Capacidad	0,02m ³		
	Flujo másico	16 tapas por minuto		
	Dimensiones	Superior	420x420 mm	
		Inferior	210x110 mm	
		Altura	270 mm	
Tipo de Material	Acero inoxidable	St 37		
Banda Transportadora	Material	PVC		
	Dimensiones	Largo	1200 mm	
		Ancho	120*3 mm	
Banda elevadora	Tipo de material	PVC		
	Dimensiones	Largo	600 mm	
		Ancho	110 mm	
		Espesor	3 mm	
Estructura De soporte	Tipo de material	Hierro		
	Dimensiones	Largo	1350 mm	
		Ancho	550mm	
		Altura	1200mm	
Soporte de visión artificial	Tipo de Material	Hierro		
	Dimensiones	Largo	183mm	
		Ancho	150*85 mm	

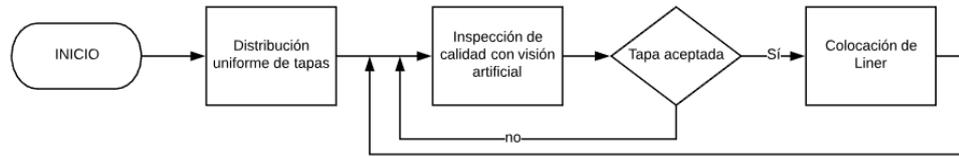
Fuente: Ricardo Briceño

En la Tabla 3.1 se detalla los distintos elementos y sus respectivas características con las cuales cuenta la máquina colocadora de *liner*.

3.2. Diagrama de bloques del proceso colocador de *liner*

Se explicara las distintas etapas del proceso mediante un diagrama de bloques que se podrá observar en la Figura 3.2

Figura 3.2 Diagrama de bloques colocador de liner



Elaborado por: Ricardo Briceño

3.3. La banda transportadora

El transporte de tapas en el proceso está formada por tres partes fundamentales: tambor motriz, banda transportadora de PVC y motor monofásico.

- **El Tambor motriz** es un dispositivo metálico encargado de funcionar como poleas que se ubican al final y al comienzo de la banda transportadora, para seleccionar se toma en cuenta factores como: potencia, velocidad, ancho de la cinta transportadora, etc.
- **Bandas PVC** se utilizan para el transporte interior de productos manufacturados en la industria alimenticia, ya que el tipo banda a seleccionar en un proceso dependerá gran parte velocidad, esfuerzo y tensión a la que sea sometida.
- **Motor monofásico** se alimenta con un voltaje de 110V, están provisto con un condensador de arranque que se fabrican como solución económica para aplicaciones en bandas transportadoras variando sus características según el proceso para el cual se ha requerido.

En la Figura 3.3 se puede observar que la etapa de transporte consta de dos bandas una vertical cuya función es receptora de las tapas que caen de la inyectora a la tolva y de las que se retroalimentan en el proceso, una horizontal que se encarga del transporte a cada una de las distintas etapas del proceso.

Figura 3.3 Banda transportadora horizontal y vertical



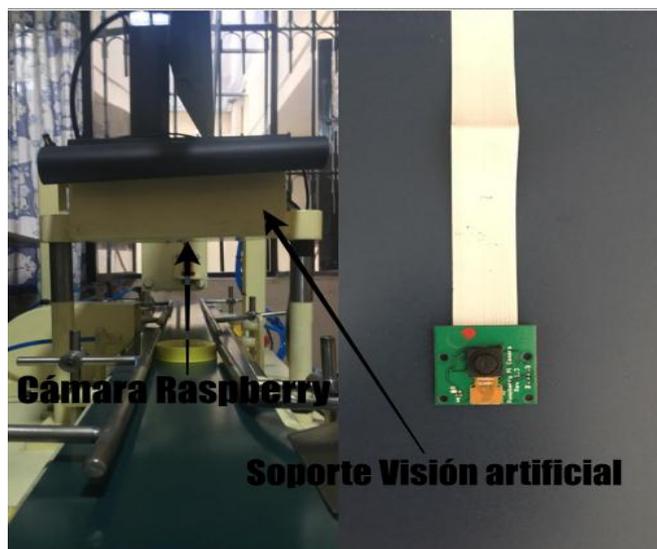
Elaborada por: Ricardo Briceño

3.4. Cámara

Es un dispositivo que funciona conjuntamente con las Raspberry para procesar imágenes, con una resolución de de 640x480 pixeles a 55 fps (fotografías por segundo) para una mayor velocidad de procesamiento. En la Tabla 3.2 se detallara las características de la cámara.

La cámara se halla ubicada en el soporte de visión artificial que se puede ver en la Figura 3.4

Figura 3.4 Soporte y Cámara



Elaborado por: Ricardo Briceño

Tabla 3.2 Características de la Cámara

Elemento	Módulo de cámara v1
Peso	3gr
Integración Linux	Driver V4L2 disponible
Sensor	OmniVision OV5647
Resolución del sensor	2592x1944 pixeles
Área de imagen del sensor	3,76*2.74 mm
Tamaño del pixel	1,4um*1,4um
Tamaño óptico	1/4"
Sensibilidad	680mV/lux-sec
Campo de visión horizontal	53,50+/-0,13 grados
Campo de visión vertical	41,41+/-0,11 grados
Longitud focal	3,60mm+/-0,01

Elaborado por: Ricardo Briceño

3.5. Soporte del sistema de visión artificial

En la Figura 3.5 el soporte de visión artificial es hecho de hierro donde su construcción se basa en dimensiones largo 180 mm, ancho 50mm y altura 85mm además se considera la altura desde el lente focal hasta la tapa, a una distancia de 125mm.

Figura 3.5 Soporte de visión artificial

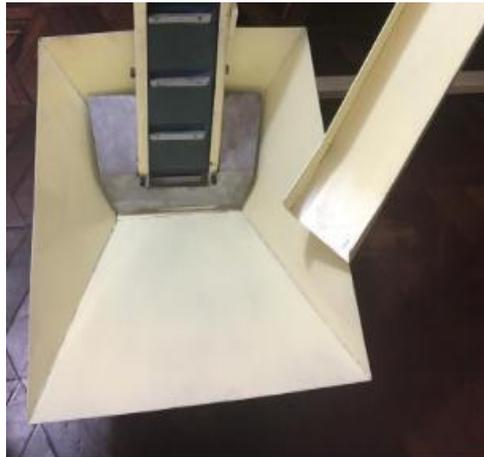


Elaborado por: Ricardo Briceño

3.6. Tolva

En la Figura 3.6 tolva es un dispositivo que tiene la capacidad de $0,02\text{m}^3$ para captar las tapas que expulsa la inyectora y de las que son retroalimentadas en el proceso, cuenta con un flujo masaco de 16 tapas por minuto con dimensiones largo 420x420mm, inferior 210x110mm y altura 270mm.

Figura 3.6 Tolva



Elaborado por: Ricardo Briceño

3.7. Tablero electro neumático y electrónico

En la Figura 3.7 el tablero electromecánico y electrónico se puede observar los distintos equipos electro neumático y electrónico etiquetados, del tal forma para poder realizar las distintas conexiones y cumplan con las distintas acciones en cada etapa del proceso.

Figura 3.7 Tablero electro neumático y electrónico



Elaborado por: Ricardo Briceño

3.8. Diseño Electrónico

En esta etapa del proceso se procede a desarrollar las conexiones en el tablero eléctrico para el correcto funcionamiento operativo del control electro neumático de los cilindros y de las diferentes salidas de aire y de protección para el motor de la banda transportadora.

3.8.1. Protección de la máquina colocadora de *liner*

En la Tabla 3.3 se especifica las características de los distintos equipos eléctricos y electrónicos de la máquina colocadora de *liner*, para lograr una protección efectiva del tablero eléctrico-neumático y electrónico además para un efectivo desempeño del sistema de control.

La selección del cableado de la máquina se dimensiona por medio de la información técnica de los equipos eléctricos y electrónicos utilizados en el proceso ya que el consumo de potencia es bajo en el sistema por lo cual se diseña con este tipo de protección. Para el cableado del sistema de control, se dispuso de cable 20 AWG multifilar ya que trabaja con corriente menores a 1A y voltaje de 24Vdc.

Tabla 3.3 Características de equipos eléctricos y electrónicos

Elemento	Características	
Motor	# Revoluciones	1550 rpm
	Tipo de conexión	Monofásico
	Potencia	1/8 Hp
	Corriente	0,77 A
	Dimensiones	D79,8*L187,5mm
Sensor	Tipo de sensor	Fotoeléctrico
	Dimensiones	D17*L45mm
	Cantidad	1
	Rango de detección	3cm a 80cm
Protección	Porta fusible	2
	Fusible	1A,2A
	Cantidad	2
	Cable	16AWG
Electroválvula 5/2 y 3/2	Configuración	Cinco vías dos posiciones 5/2 y tres vías dos posiciones 3/2
	Rango de voltaje	24Vcc
	Clase de protección	IP65
	Accionamiento	Por solenoide
	Presión de trabajo	21-114psi
	Cantidad	3 de 5/2 y 3 de 3/2
Cilindro doble efecto	Presión de trabajo	145psi
	Diámetro	16*50 y 20mm
	Patrón de movimiento	Doble efecto
	Medio de trabajo	aire
	Carrera	60mm
Cilindro rectangular	Diámetro	40*30mm
	Carrera	90mm
Relé	Voltaje in	5V
	Voltaje out	24V
	Corriente	15mA
	Dimensiones	19,1*15,1mm
Fuente DC	Voltaje in	12Vcc
	Voltaje out	24Vdc
	Corriente	10A
	Dimensiones	50*50*32mm
Raspberry	Voltaje	5Vdc
	Corriente	2.5 A
	Procesador	64bits Soc@1.4Ghz
	Memoria	1Gb SDRAM
	Dimensiones	85*56 mm

Elaborado por: Ricardo Briceño

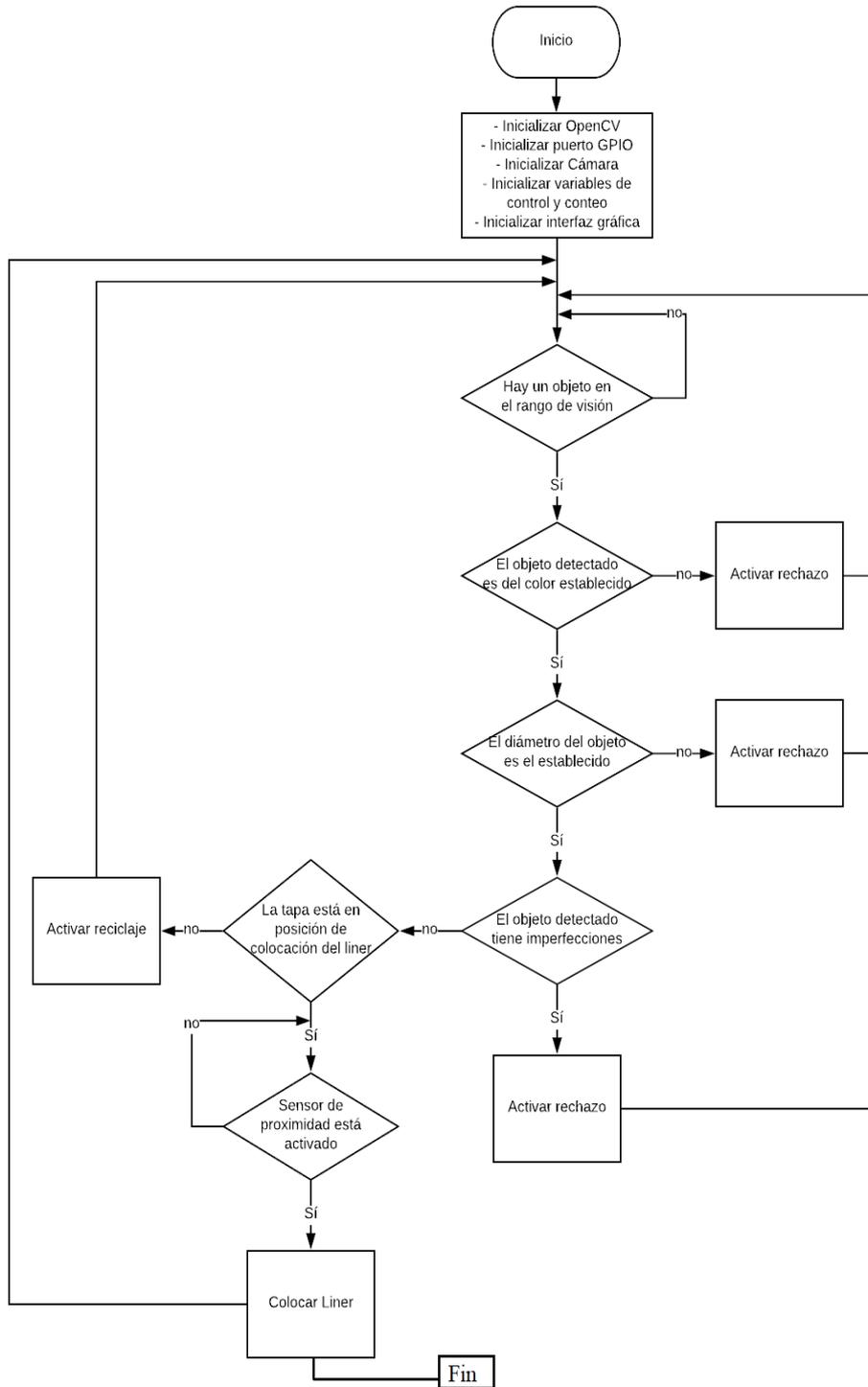
3.9. Algoritmo de control de la Raspberry

En la Figura 3.8 se halla el diagrama de flujo de la programación diseñada para la Raspberry, la misma que realiza las distintas acciones de forma autómatas y secuencial cumpliendo con los parámetros requeridos del usuario.

En el diagrama se visualiza el control de posicionamiento de la tapa en cada etapa del proceso, mientras la cámara detecte un objeto la Raspberry accionará las electroválvulas 3/2 (3 vías, 2 posiciones) que se encargan de separar las tapas por falla de color, geometría y las tapas que no están en la posición correcta para colocar el *liner*, retroalimentándolas para que mediante un proceso mecánico regresen a la línea de producción en la posición requerida.

De la misma forma cuando el sensor fotoeléctrico haya detectado la tapa, el cilindro doble efecto realiza un paro a la tapa que está lista para colocar el *liner*. En el Anexo 1 se podrá observar la programación de las Raspberry Pi.

Figura 3.8 Diagrama de flujo de programación en la Raspberry



Elaborado por: Ricardo Briceño

3.10. Software y Firmware necesarios

3.10.1. Software para Windows

- **Etcher:** graba el sistema operativo en la memoria Sd en formato de disco apto para Raspberry. (BalenaEtcher, s.f.)
- **Advanced Ip Scanner:** Software dedicado a mostrar las características de red de los dispositivos conectados a una misma red local. (Scanner, s.f.)

3.10.2. Firmware y software para Raspberry

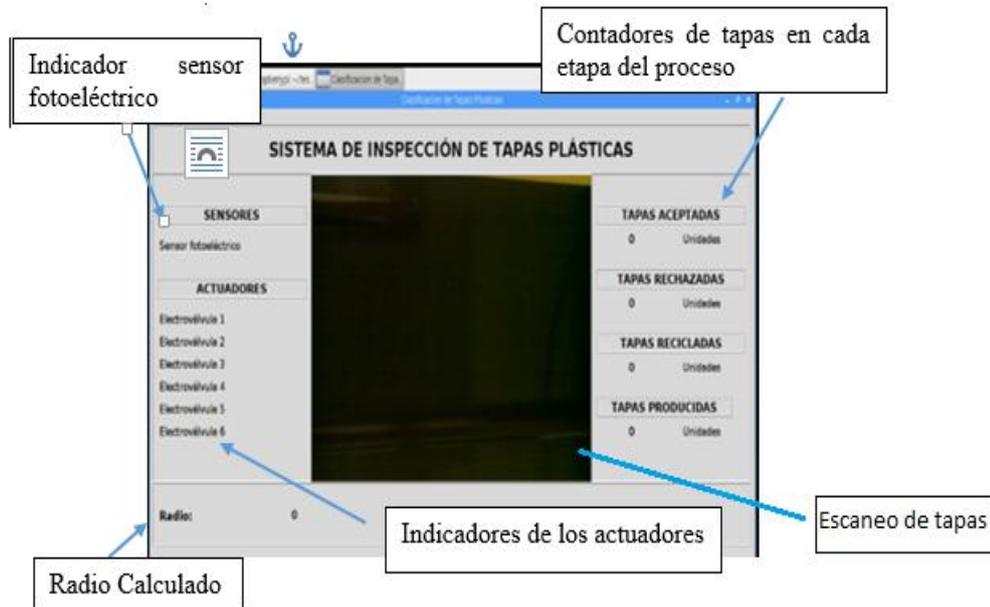
- **Raspbian:** Sistema operativo basado en Linux y optimizado para Raspberry. (Raspbian, 2018)
- **OpenCV:** Es una biblioteca de código abierto orientado al desarrollo de visión artificial, procesamiento de imágenes y aprendizaje automático. (OpenCv, s.f.)
- **Python:** Lenguaje de programación multiparadigma, orientado a objetos, programación imperativa y, en menor medida, programación funcional. (Python, s.f.)

3.11. Diseño HMI

El HMI se configura de acuerdo al controlador a utilizar, por temas de complejidad de programación y velocidad de procesamiento el uso de recursos aumenta en la Raspberry por este motivo se verá limitado a presentar más de una pantalla sin embargo es suficiente para cumplir con los requerimientos de visualización del usuario en el proceso. Para que logre existir comunicación entre dispositivos será mediante el cable HDMI.

En la Figura 3.9 se visualiza la pantalla del HMI donde se tiene los distintos elementos que intervienen en el proceso, en el centro de la pantalla se observa el escaneo de la tapa de acuerdo a los parámetros establecidos de calidad, en la izquierda los distintos elementos indicadores que se activaran cada vez que se realice una acción en cada etapa del proceso y por último el conteo de tapas en posición correcta (aceptadas), rechazadas, recicladas (retroalimentadas) y la producción total.

Figura 3.9 Componentes del HMI



Elaborado por: Ricardo Briceño

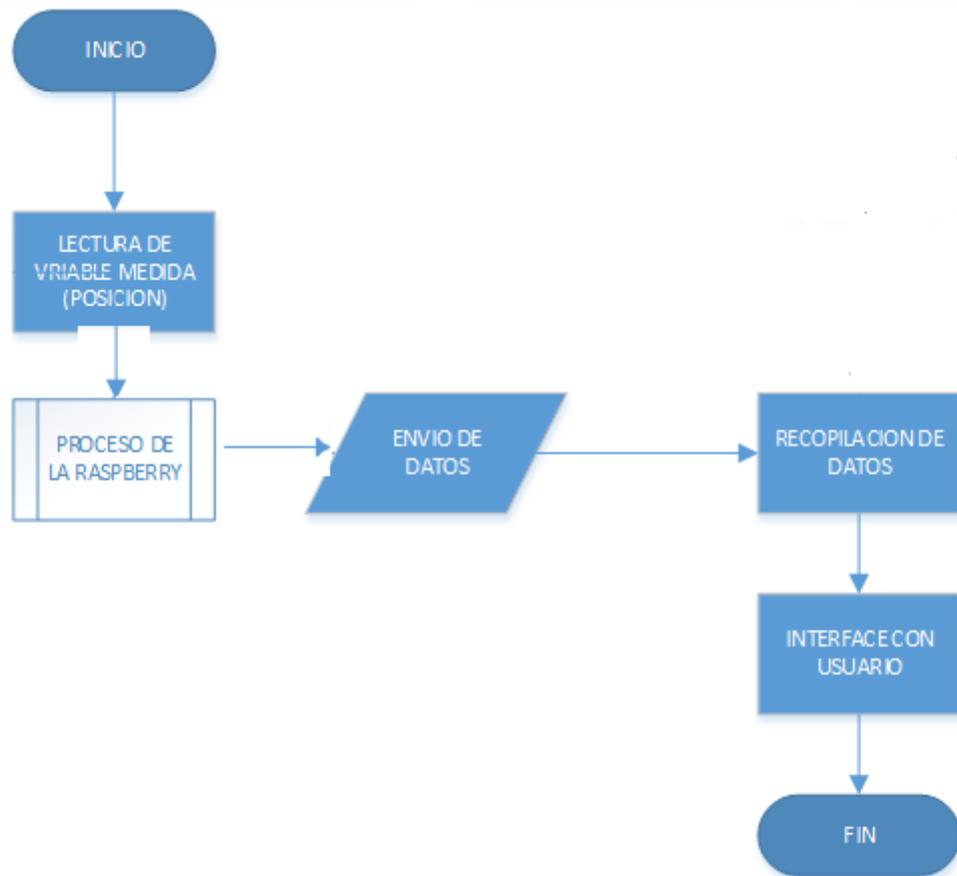
Tabla 3.4 Ítems de la Pantalla HMI

Ítem	Descripción
Sensor	Indicador de la tapas en correcta posición para colocar el <i>liner</i>
Actuador	Indicadores de las electroválvulas que se activarán en el proceso
Radio	Radio calculado de la tapa
Pantalla central	Se visualizará el escaneo de tapas
Contador	Se visualiza el número de tapas que son rechazadas, retroalimentadas, aceptadas

Elaborado por: Ricardo Briceño

En la Figura 3.10 se presenta la secuencia de funcionamiento del sistema donde la velocidad de procesamiento mediante la Raspberry será fundamental para tomar lectura de datos correctos y de esta manera poder visualizar a tiempo real los distintos indicadores mencionados en la Tabla 3.4

Figura 3.10 Diagrama de flujo del HMI



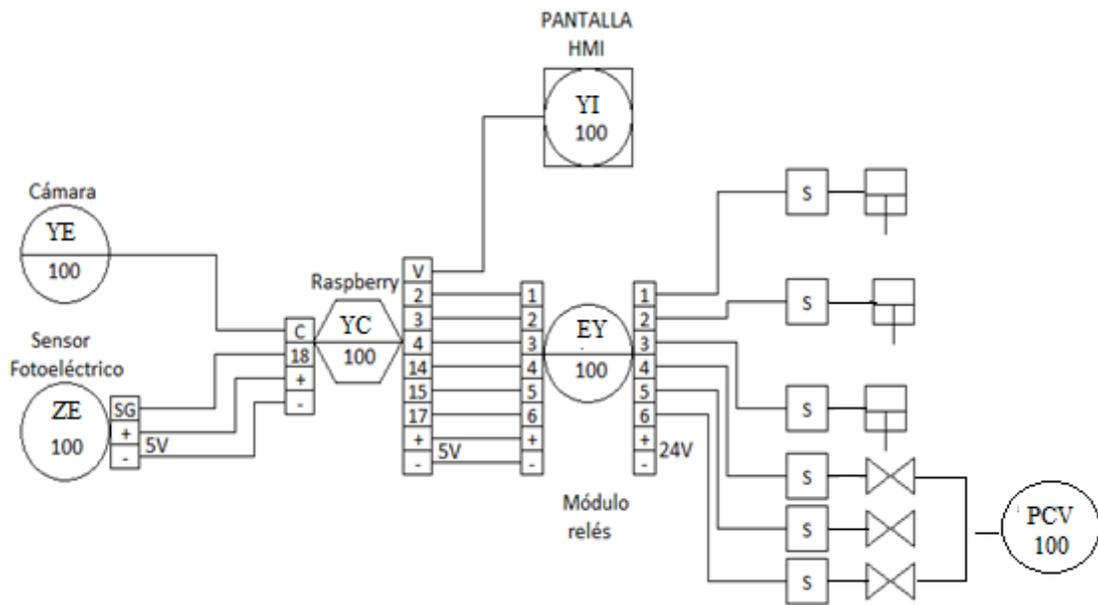
Elaborado por: Ricardo Briceño

3.12. Instrumentación del proceso

En la Figura 3.11 se puede observar el plano de instrumentación del proceso. Se tiene un lazo de control cerrado (100). Donde por medio de la cámara conectada al puerto de la Raspberry que está definido con la letra C, se realiza un escaneo de la tapa para que el sensor envíe una señal de 5V conectado al pin 18 del GPIO de la Raspberry que indica que está en la posición correcta para la colocación del *liner*.

Las salidas digitales de la Raspberry son conectadas a la entrada de 5V del módulo de relés con la finalidad de tener a su salida un voltaje de 24Vdc para poder tener el control sobre los solenoides y por ende activar y desactivar a las electroválvulas para que cada actuador realice la acción en cada etapa del proceso.

Figura 3.11 Diagrama de Instrumentación



Elaborado por: Ricardo Briceño

Tabla 3.5 Simbología diagrama de instrumentación

Símbolo	Descripción
YE	Sensor de Presencia
ZE	Sensor de posición
YC	Controlador de Presencia
YI	Indicador de Posición
EY	Relé de Voltaje
S	Solenoides
PCV	Válvula controladora de presión

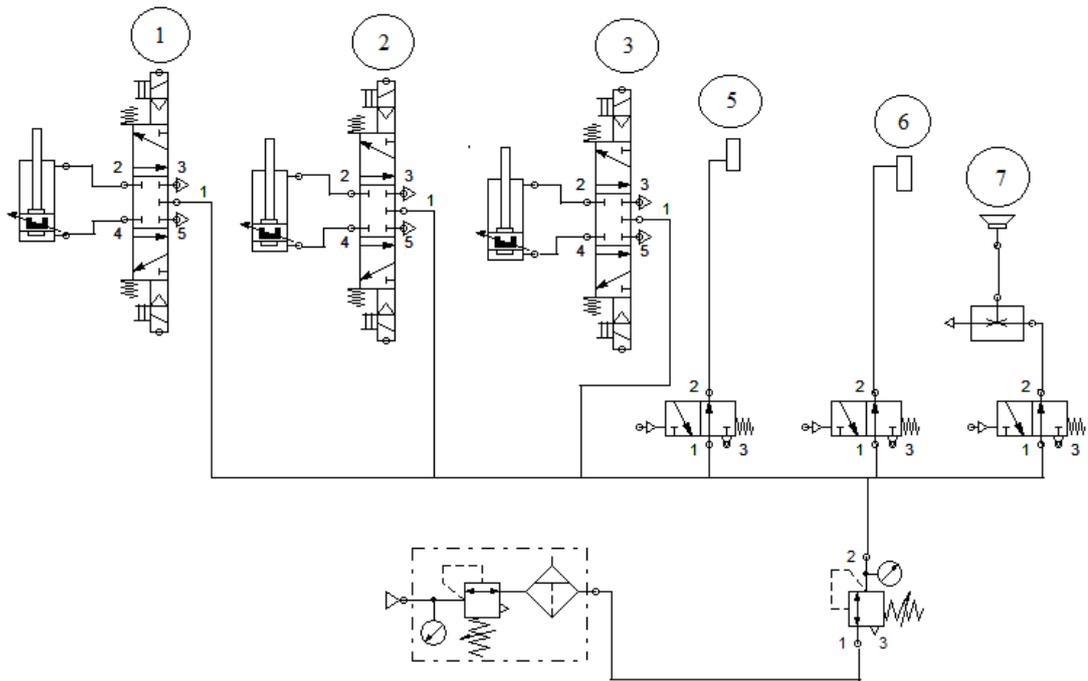
Elaborado por: Ricardo Briceño

En la Tabla 3.5, se muestra la simbología utilizada en el diagrama de instrumentación para la implementación del lazo de control del proceso.

3.13. Diseño Electro-Neumático

En la Figura 3.12 se observa el diseño neumático, donde los distintos equipos están distribuidos y etiquetados de acuerdo a las distintas conexiones del proceso para garantizar su óptimo funcionamiento.

Figura 3.12 Diseño neumático



Elaborado por: Ricardo Briceño

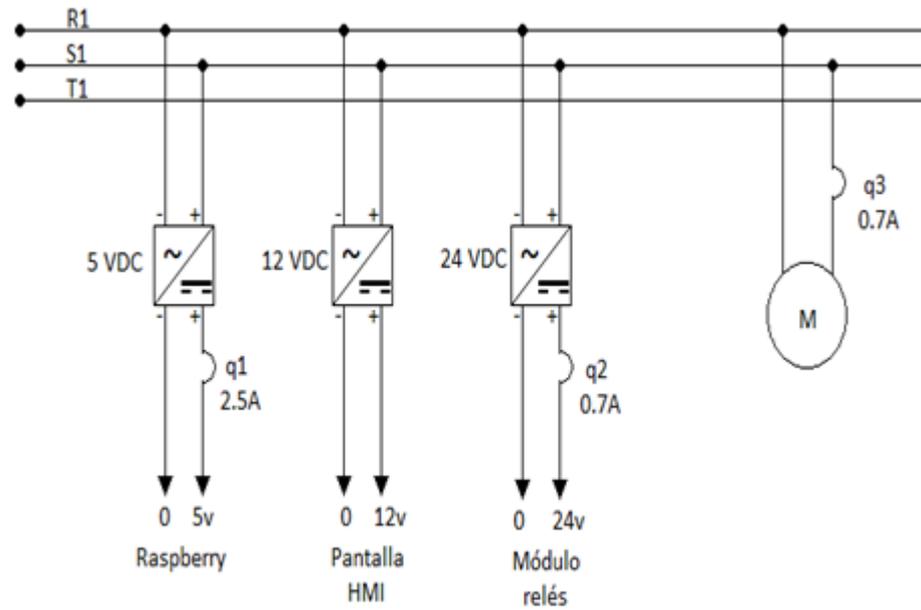
Tabla 3.6 Simbología del diseño neumático

Ítem	Equipo	Característica		
		40mm	30mm	90mm
1	Pistón rectangular	40mm	30mm	90mm
2	Pistón cilíndrico	$\phi 16$	50mm	
3	Pistón cilíndrico	$\phi 20$	60mm	
5	Salida 1	$\phi 6$ mm		
6	Salida 2	$\phi 6$ mm		
7	Ventosa	$\phi 50$		

Elaborado por: Ricardo Briceño

En la Figura 3.13 se observa el diseño eléctrico donde las tomas eléctricas son de 110Vcc conectadas a unos conversores de corriente con el objetivo de alimentar los equipos de control ya que estos trabajan con 5Vdc, 12Vdc y 24Vdc respectivamente. Contienen tres fusibles para protección de la Raspberry (2,5A), módulo de relés (0,7A) y el motor (0,7A).

Figura 3.13 Diseño Eléctrico



Elaborado por: Ricardo Briceño

Tabla 3.7 Elementos del tablero electrónico-neumático

Ítem	Elemento	Descripción
	Fusible	Protección motor
	Porta fusible	Contactos de fusible
1-2	Válvula 5-2	Accionamiento de pistón vertical rectangular giratorio
		Accionamiento para pistón que sujeta a la ventosa
3	Válvula 5-2	Accionamiento para el pistón de parada de tapas
5-6	Válvula 3-2	Accionamiento de la salida para rechazo
	Válvula 3-2	Accionamiento de la salida para reciclaje
7	Válvula 3-2	Accionamiento de la ventosa

Elaborado por: Ricardo Briceño

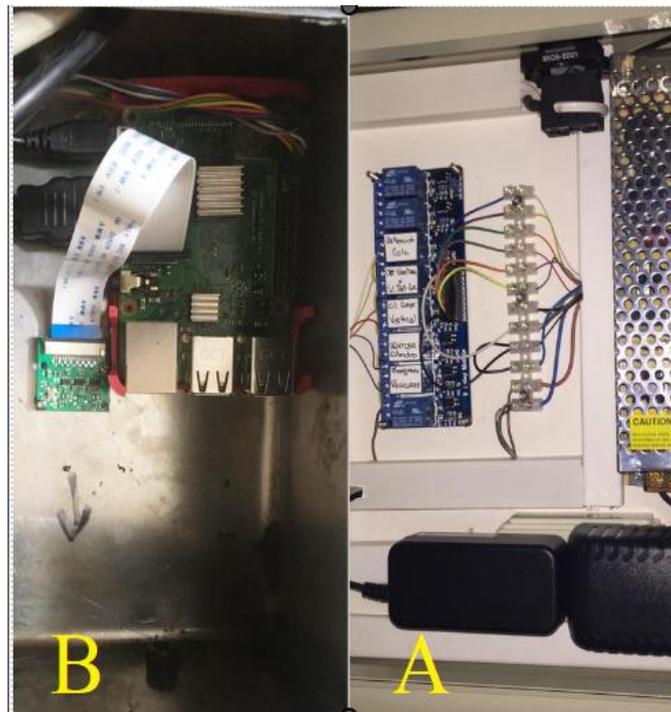
En la Tabla 3.7 se observan los distintos equipos eléctricos para protección del motor y neumáticos donde las electroválvulas activan los pistones, ventosa y salidas en cada etapa del proceso como se detalla en cada ítem.

3.14. Tablero Interno Electrónico de control

En la Figura 3.14 se observan el diseño de tablero de control, donde se ubican los equipos eléctricos y electrónicos que permitirá la activación de la máquina. Dispone de un sistema de protección para los equipos electrónicos como el fusible de 0.7A.

El módulo de relés es un equipo electromecánico capaz de controlar un circuito de mayor potencia que el de entrada en este caso recibir una señal de 5v de la Raspberry y a su salida un voltaje de 24V que alimentan a los solenoides activando a las electroválvulas al recibir un pulso del controlador.

Figura 3.14 Diseño del interior del tablero de control

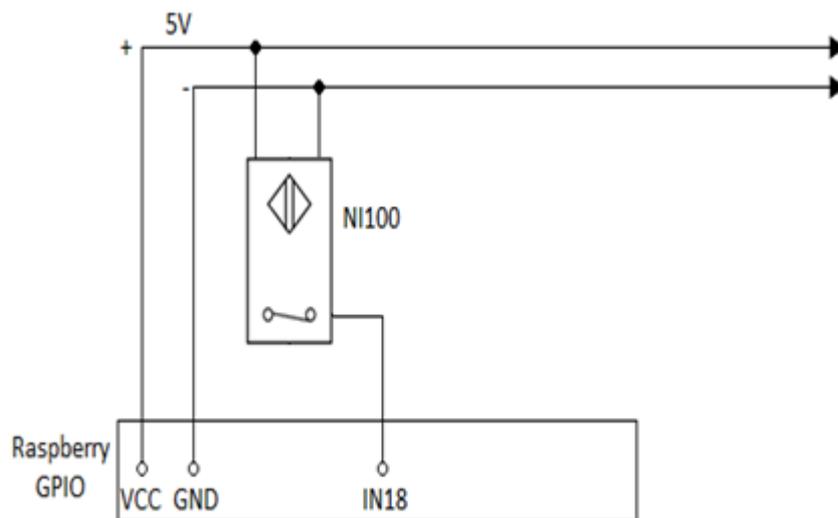


Elaborado por: Ricardo Briceño

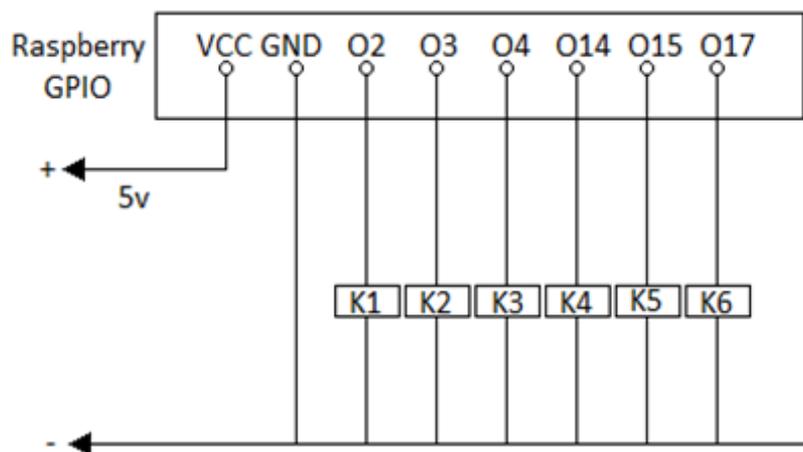
En la Figura 3.14 en la imagen A observan los distintos equipos antes mencionados que se manejarán en el proceso de control, distribuido e etiquetado de acuerdo a las distintas conexiones del proceso.

Por motivos de funcionalidad de la cámara para evitar que no haya reflejos de luz externa y posición específica en el proceso se colocó en un tablero externo junto con la Raspberry como se puede ver en la imagen B.

Figura 3.15 Esquema electrónico de entradas y salidas digitales de la Raspberry
Entrada Digital



Salida Digital



Elaborado por: Ricardo Briceño

En la Figura 3.15 se observa que la GPIO de las Raspberry será energizada con 5Vdc y conexión a GND. Las salidas digitales se conectara la entrada del módulo de relés y de la salida a cada electroválvula que serán etiquetadas para su identificación en el proceso a continuación en la Tabla 3.8 se detallara la simbología de las salidas digitales.

Tabla 3.8 Simbología de las Salidas digital de la Raspberry

Símbolo	Función
K1	Electroválvula de cilindro de ventosa
K2	Electroválvula cilindro detección y puesta de liner
K3	Electroválvula vertical de puesta de liner
K4	Electroválvula accionamiento de ventosa
K5	Electroválvula rechazo
K6	Electroválvula reciclaje

Elaborado por: Ricardo Briceño

Tabla 3.9 Elementos del tablero electrónico

Ítem	Elemento	Descripción
1	Fusible	Protección equipos electrónicos
2	Fuente	Protección de 24Vdc
3	Relé K1	Electroválvula de cilindro de ventosa
	Relé K2	Electroválvula cilindros de detección y de puesta liner
	Relé K3	Electroválvula cilindro vertical de puesta de liner
	Relé K4	Electroválvula accionamiento de ventosa
	Relé K5	Electroválvula rechazo
	Relé K6	Electroválvula reciclaje

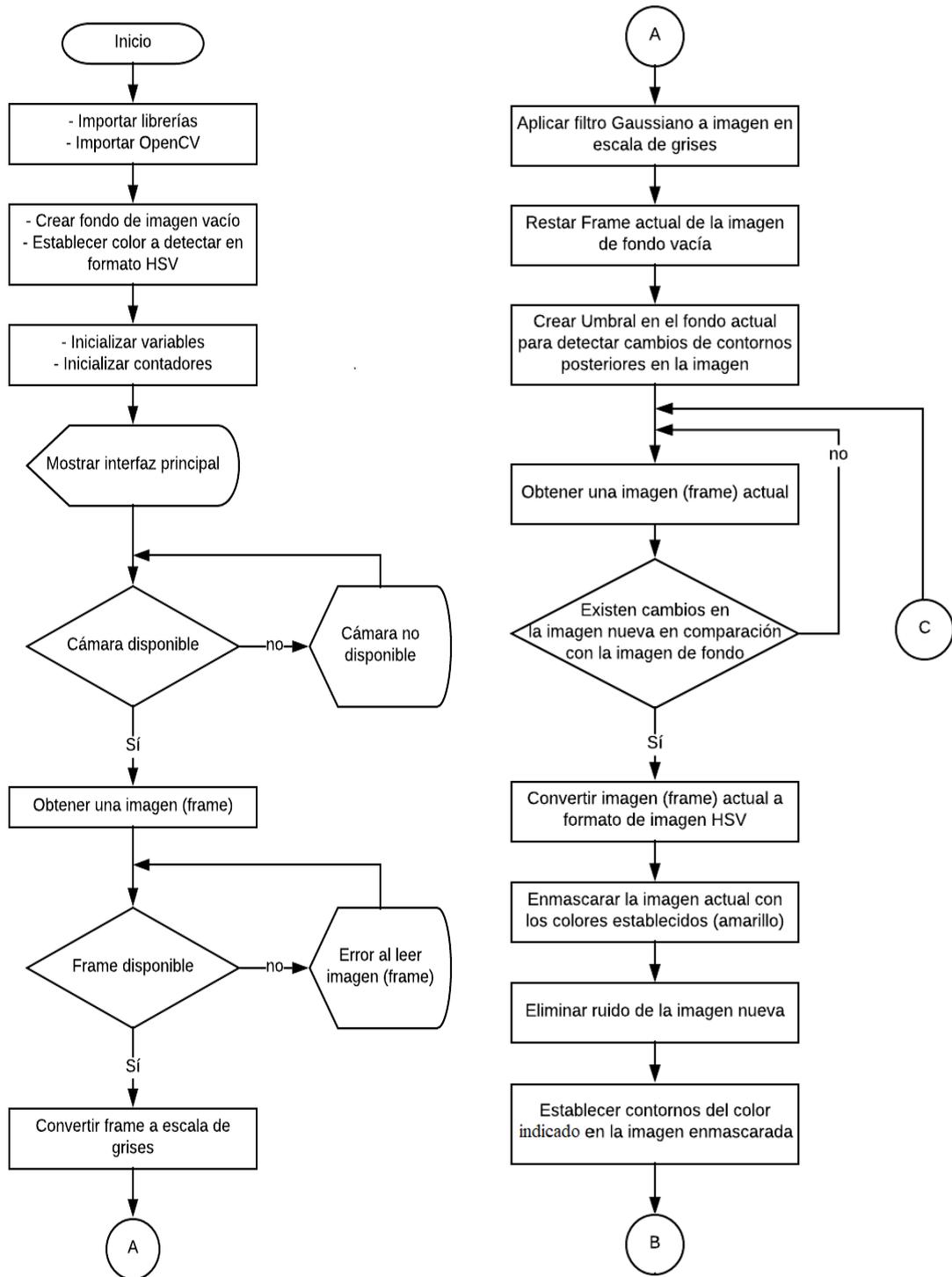
Elaborado por: Ricardo Briceño

En la Tabla 3.9 se observa en el ítem 1-2 los equipos de protección del tablero electrónico y el ítem 3 se etiquetaran los elementos en base de las salidas digitales de la Raspberry descritas en Tabla 3.8 con la finalidad de que exista una relación de simbología para llevar un orden para las conexiones.

3.15. Visión Artificial

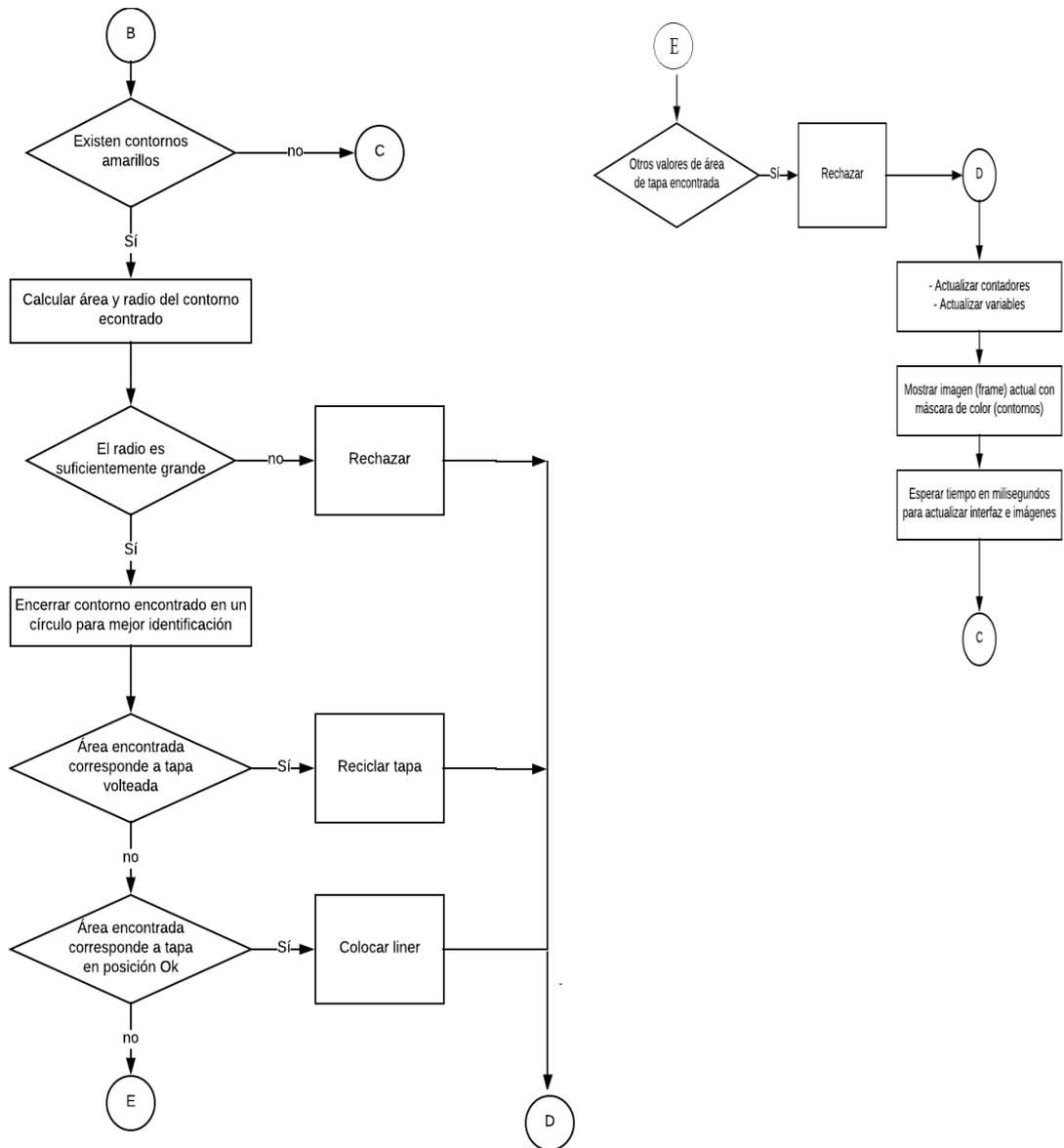
En la Figura 3.16 y Figura 3.17 se observa el diagrama de secuencia diseñado para el procesamiento digital de imágenes considerando que es un sistema que cuenta con elementos en movimiento dentro del campo de visión de la cámara, donde se toma una foto inicial del lugar este caso de la banda para que cualquier cosa que sea distinta de lo que se tomó se lo considere como un movimiento dentro del rango. Una vez que el elemento este en la rango de visión se detecta del color que se requiere, a continuación se verifica el área para determinar su posición boca abajo o boca arriba pasara al reciclaje o a colocar el *liner* y a su vez si el área es distinta a la tapa original se rechazará. La programación de la visón artificial se encentra en el Anexo 1

Figura 3.16 Diagrama de Flujo Visión Artificial



Elaborado por: Ricardo Briceño

Figura 3.17 Continuación de diagrama de flujo de visión artificial

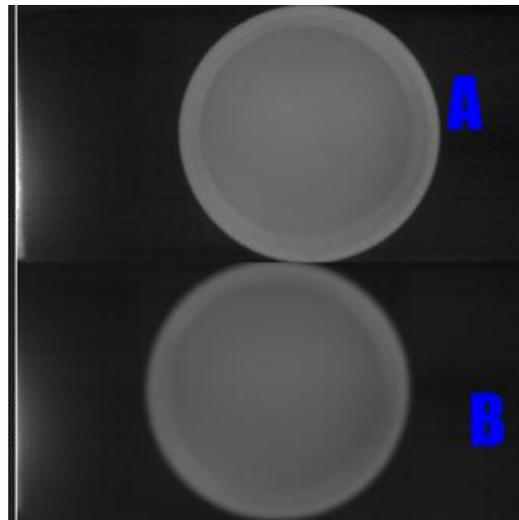


Elaborado por: Ricardo Briceño

Para el propósito en el tema de programación en cada etapa del proceso se realizara:

- **Paso 1** en la Figura 3.18 se observa la creación de una imagen fondo, se inicializara la cámara y tomara un frame (foto) de la tapa que se cambiará a gris y se le aplicará un filtro gaussiano y posterior a esto se borra los contornos para comparar con las nuevas imágenes. Se aplicara en la imagen una escala de gris ya que es más fácil manipular este tipo de acciones que con una imagen a colores.

Figura 3.18 Frame Gris (A) y filtro Gaussiano (B)

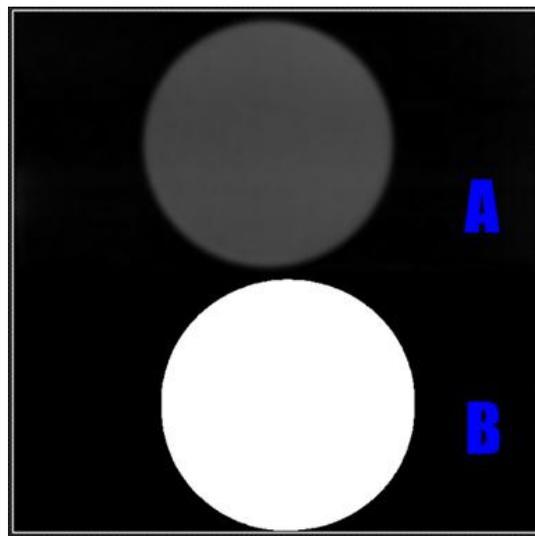


Elaborado por: Ricardo Briceño

- **Paso 2** en la Figura 3.19 se observa que para poder detectar que haya objetos en movimiento dentro de detección del rango de la cámara se va a realizar una resta de imágenes que consiste en tomar un frame, el mismo que será cambiado de colores normales a color gris para luego utilizar un filtro de difuminación gaussiano que hará que la imagen se torne un poco borrosa. Esta imagen borrosa al utilizar el filtro umbral hará que esta imagen quede vacía, por lo tanto cualquier imagen comparada será detectada como un cambio dentro de los objetos que puedan aparecer. Esta comparación la realizamos al restar la imagen nueva de la imagen de fondo, que si se toma en cuenta es una imagen que está vacía. Al decir que está vacía no significa que no hay

información si no que se trata de borrar cualquier contorno que existe asumiendo que esos elementos siempre van a estar presentes (banda transportadora y guías de tapas) y cualquier cambio nuevo en relación a este fondo vacío indicarán que existe la presencia de elementos.

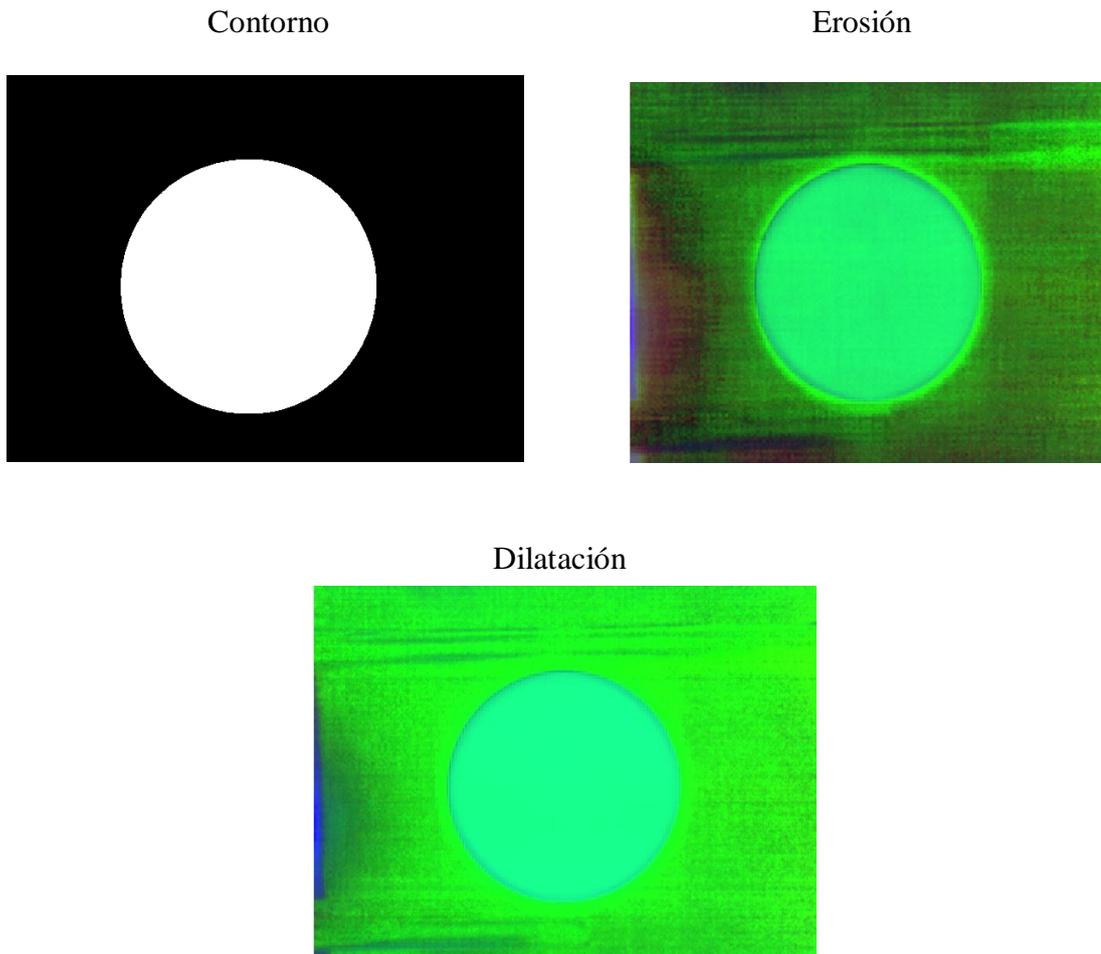
Figura 3.19 Resta (A) y Umbral (B)



Elaborado por: Ricardo Briceño

- **Paso 3** en la Figura 3.20 para saber que algo se está moviendo se encontrara los contornos en la nueva imagen que se deberá convertir en formato de imagen HSV (Matiz, Saturación, Valor), al verificar que estos objetos que están pasando son del color amarillo que se necesita para eso al frame nuevo se lo enmascara (detectar solo el color que se necesita en este caso amarillo).Una vez detectado el color amarillo se realizará una erosión y una dilatación para eliminar cualquier ruido y estar seguros que el color que se busca es el correcto.

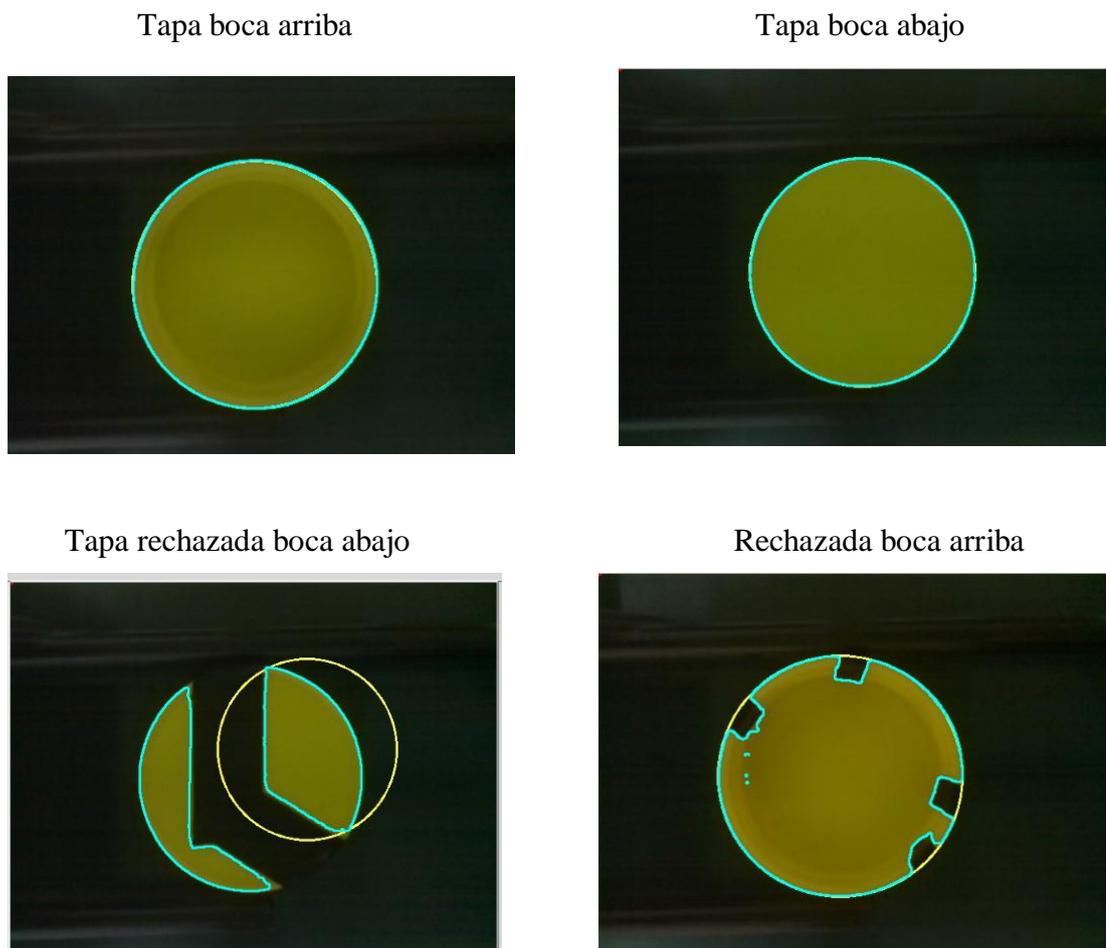
Figura 3.20 Contorno, Erosión, Dilatación



Elaborado por: Ricardo Briceño

- **Paso 4** en la Figura 3.21 una vez que se ha erosionado y dilatado es decir se ha eliminado el ruido se encontrara nuevamente los contornos, para saber si el objeto amarillo existe se hará una comparación entre objetos moviéndose dentro de la banda. Si es afirmativo verificaremos los colores que al ser amarillos a través de los momentos de OpenCv se comprobara el área y de esta manera se sabrá si la tapa esta boca arriba, boca abajo o a su vez si no corresponde a estos rangos significa que no es lo que se busca y por ende será rechazado.

Figura 3.21 Área de Tapas



Elaborado por: Ricardo Briceño

3.16. Interpretación de las imágenes luego del procedimiento

Para poder dilucidar el procedimiento de los pasos anteriores se procederá a enunciar los distintos pasos que se toma en cuenta para llegar a obtener los resultados requeridos:

- a. Se realiza una captura de imagen de fondo a la cual se aplicara filtros, se eliminara sus contornos y se la comparara con las imágenes subsiguientes que al resultar diferentes a la de fondo se puede decir que existe un objeto en movimiento.
- b. Se realiza una captura de la última imagen es decir la actual que pasa a través del rango de visión de la cámara la cual se pasara a formato HCV aplicando

filtros de enmascaramiento se tomara un rango de color amarillo para determinar si el color buscado es el requerido.

- c. Una vez identificado el color se encontrara el contorno de la tapa que servirá para delimitar el rango de visión de la cámara, a que se hace referencia con el contorno todo lo que está alrededor de la tapa, una vez encontrado el contorno se determinara el área de dicho contorno para aquello se determinará el rango en cual se encuentran las tapas boca arriba, boca abajo y caso contrario las áreas que no se encuentran en estos rangos serán rechazadas como tapas con deformación en su geometría. En la Tabla 3.10 se podrá verificar los rangos de dichas áreas al pasar por el rango de visión de la cámara.

Tabla 3.10 Áreas de tapas en posición correcta e incorrecta

PRUEBA	AREA	TAPA BOCA ARRIBA	TAPA BOCA ABAJO
1	69750	✓	
2	65020		✓
3	70000	✓	
4	68500	✓	
5	62500		✓
6	64500		✓
7	68050	✓	
8	69200	✓	
9	65840		✓
10	68350	✓	

Elaborado por: Ricardo Briceño

- d. Al finalizar la identificación y si los resultados cumplen con el color, la posición y geometría de la tapa se procederá a colocar el *liner*.

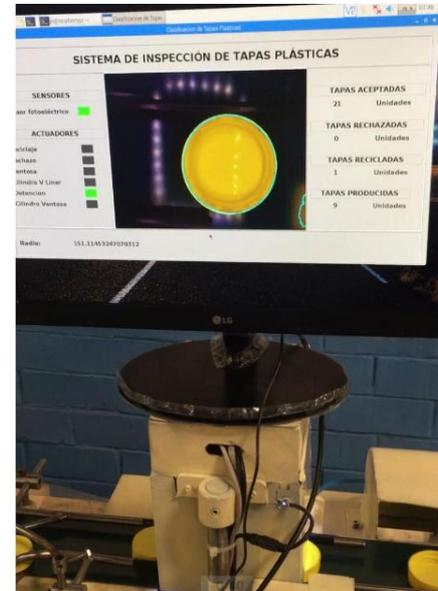
3.17. Hardware de la maquina colocadora de liner

Figura 3.22 Etapa de transporte, Etapa de visión artificial y HMI

Etapa de transporte



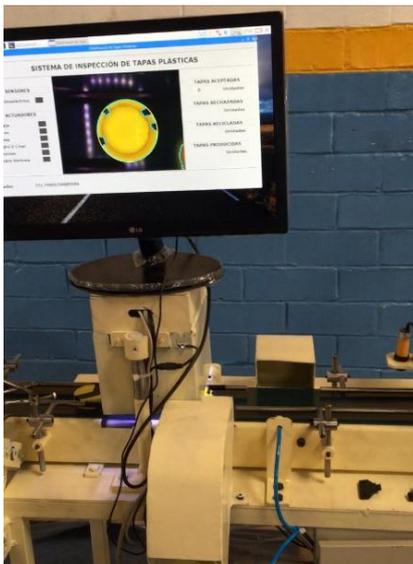
Etapa de visión artificial y HMI



Elaborado por: Ricardo Briceño

Figura 3.23 Etapa de Rechazo

Tapa falla inyección



Tapa Desechada



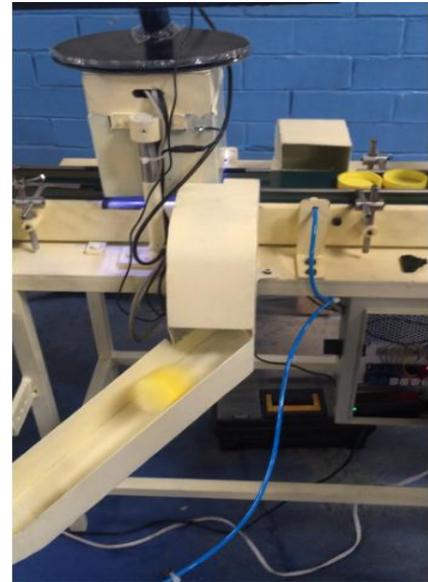
Elaborado por: Ricardo Briceño

Figura 3.24 Etapa de Reciclaje

Tapa boca abajo



Tapa Reciclada



Elaborado por: Ricardo Briceño

Figura 3.25 Etapa de colocación liner y Tablero electro-neumático y electrónico

Etapa colocar *liner*



Tablero control y Electro neumático



Elaborado por: Ricardo Briceño

CAPÍTULO 4

PRUEBAS Y RESULTADOS

En este capítulo se procederá a detallar el proceso y los equipos y herramientas que intervendrán para la construcción y funcionamiento de la máquina, que una vez concluida se realizara distintas pruebas con la finalidad de comprobar los resultados en el sistema.

4.1. Procedimiento

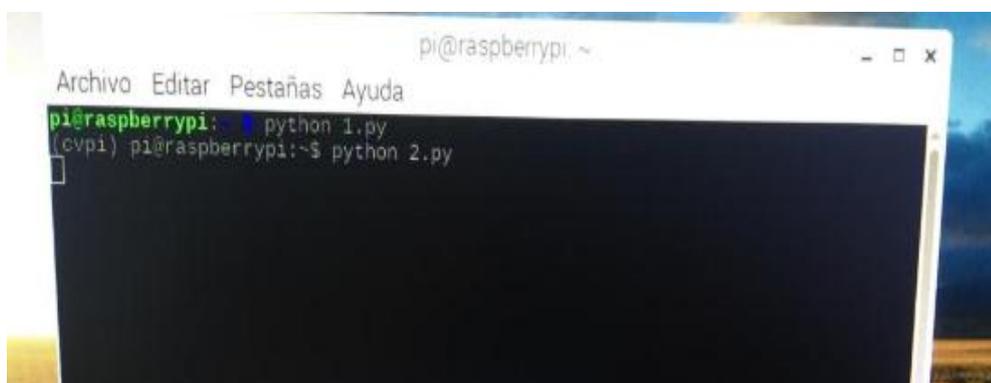
Al iniciar se hará clic en el icono de la pantalla para ingresar los códigos una vez abierto se escribirá dos líneas de código a través de teclado hasta el entorno del HMI se visualice, al finalizar se encenderá el selector del tablero eléctrico para que empiece a operar la banda transportadora.

Figura 4.1 Prueba de Ingreso al sistema



INGRESO PANTALLA DE
COMANDOS RASPBERRY

Ingreso líneas de código



Elaborado por: Ricardo Briceño

4.2. Prueba de ingreso al sistema

En la Figura 4.1 se puede observar las dos líneas de código para ingresar al entorno de la pantalla HMI, la primera se inicia el entorno virtual (Python 1.py) y la segunda es para entrar al programa (Python 2.py)

4.3. Pruebas de la etapa de transporte

Figura 4.2 Etapa de transporte

Banda Horizontal



Elaborado por: Ricardo Briceño

Banda Vertical

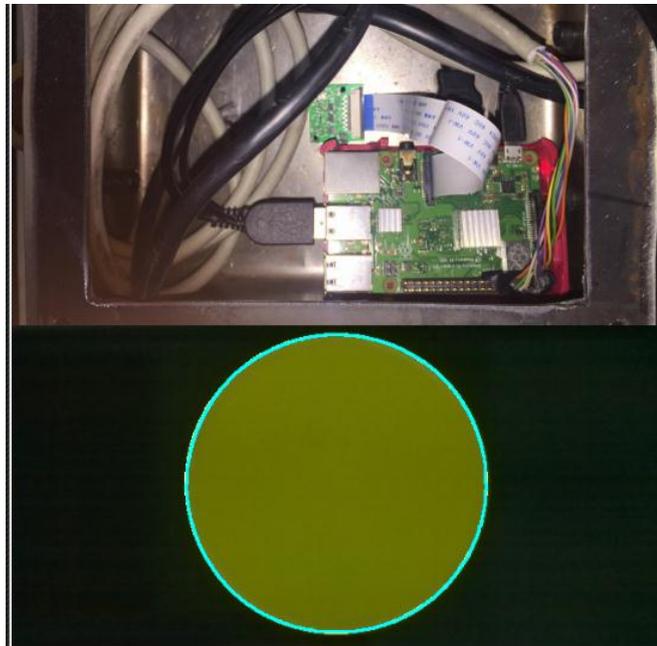


Elaborado por: Ricardo Briceño

En la Figura 4.2 se observa la etapa de transporte, donde se realizó pruebas y se determinó que la velocidad de 67rpm que entrega el motor es ineficiente, debido a que la velocidad de procesamiento de la Raspberry por medio de la cámara al momento que las tapas pasan por la banda transportadora no es suficiente para cumplir las distintas acciones del sistema, por aquello se procedió a colocar dos poleas extras de 1 y 4,5 pulgadas para reducir la velocidad del motor a 24 rpm ideal para el proceso.

4.4. Pruebas de visión artificial

Figura 4.3 Pruebas Visión Artificial



Elaborado por: Ricardo Briceño

En la Figura 4.3 se observa la etapa de visión artificial que consiste en capturar imágenes que para aquello se va a utilizar la cámara y transmitir los datos digitalmente hacia el procesador de la Raspberry que será el encargado de ejecutar las instrucciones y comandos escritos en el lenguaje de programación Python de acuerdo a los requerimientos del sistema.

La velocidad de procesamiento del sistema equivale a 3.3 segundos por cada tapa y de esta manera llega a colocar el *liner* en 18 tapas por minuto.

4.5. Prueba de tapas en posición incorrecta

Figura 4.4 Pruebas de la etapa de tapas en posición incorrecta

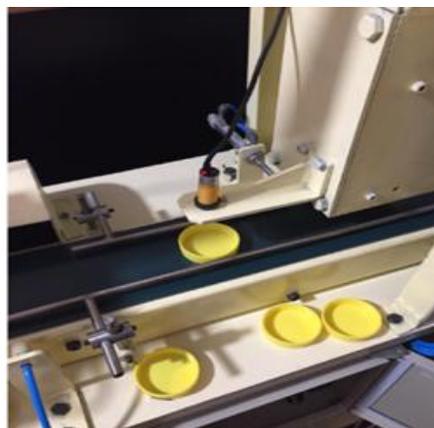


Elaborado por: Ricardo Briceño

En la Figura 4.4 se observa la etapa donde las tapas que están en posición incorrecta para colocar el *liner* se retroalimentan y caen por gravedad a la tolva a través de la rampa al ser expulsadas por una salida de aire, donde la banda vertical por medio de un mecanismo de pestañas sobrepuestas sobre ella se podrá girar las tapas y de esta forma vuelvan a la línea de producción en la posición requerida.

4.6. Pruebas de la etapa de detección de tapas en posición correcta

Figura 4.5 Pruebas de la etapa de detección de tapas en posición correcta

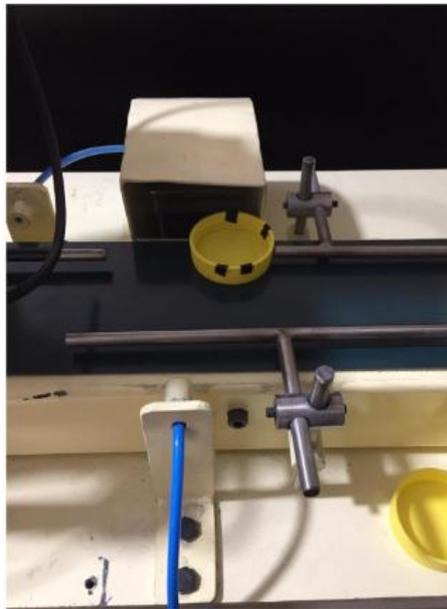


Elaborado por: Ricardo Briceño

En la Figura 4.5 se observa el sensor fotoeléctrico que se encargara de la detección de las tapas que hayan pasado de forma eficiente por el control de calidad y posición correcta (tapa boca arriba), una vez detectado transcurrirá un tiempo de 5.8 segundos para actué el cilindro doble efecto de tal forma que detenga la tapa y se coloque el *liner*.

4.7. Pruebas de desecho de tapas por falla de inyección

Figura 4.6 Pruebas de tapa por falla de inyección



Elaborado por: Ricardo Briceño

En la Figura 4.6 se observa la etapa de desecho donde las tapas al ser escaneadas por la cámara y procesadas no cumplan con los parámetros geométricos y de color serán rechazadas por medio de una salida de aire y caerán en una caja para luego ser colocadas en un molino para recuperar la materia prima.

4.8. Pruebas de colocadora de *liner*

Figura 4.7 Prueba etapa de colocación de *liner*



Elaborado por: Ricardo Briceño

En la Figura 4.7 se observa la etapa de colocación de *liner* donde mediante una señal digital, la ventosa en posición inicial activará la electroválvula (ventosa) y succionará el *liner*, activará la electroválvula (cilindro ventosa) para posicionarse sobre la tapa, una vez ahí se introducirá el *liner*. Al concluir con estas etapas se desactivarán las electroválvulas del tal forma que retorne la ventosa y el cilindro a la posición inicial.

4.9. Pruebas del HMI

Figura 4.8 Pruebas de funcionamiento HMI
Funcionamiento escaneo de tapa



Funcionamiento de indicadores actuadores, contadores, sensor



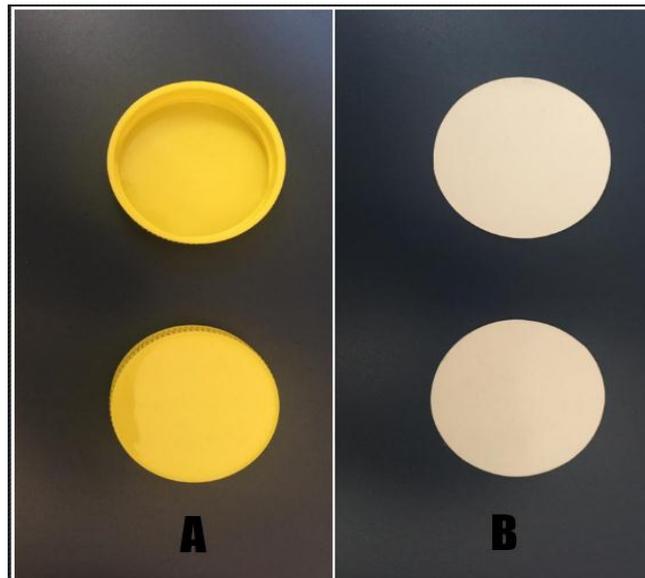
Elaborado por: Ricardo Briceño

En la Figura 4.8 se observa las pruebas realizadas del HMI, donde para este caso en particular en la pantalla de escaneo se visualiza una tapa que cumple con los requerimientos de calidad y de posición correcta para colocar el *liner* ya que se activan los indicadores detención, ventosa y cilindro vertical.

Por motivos de funcionalidad del sistema se realizaron pruebas con tapas rechazadas, recicladas y tapas aceptadas para colocar *liner*, donde se lleva el conteo de cada una de ellas como se observa al lado derecho de la pantalla.

4.10. Materia Prima

Figura 4.9 Materia Prima Liner (B) y Tapa(A)



Elaborado por: Ricardo Briceño

En la Figura 4.9 se observa en la parte A la tapa y en la parte B el *liner* que se usaran en el proceso. La experiencia se realizó con la tapa de diámetro indicado obteniendo distintos resultados de acuerdo a la previa inyección.

4.11. Funcionamiento

El sistema de control de calidad y posición de las tapas al tratarse de un proceso que se realizará mediante el análisis de imágenes, se necesitara que la velocidad de la etapa de transporte y procesamiento de imágenes se defina de tal forma integral que se pueda obtener el resultado requerido.

Para aquello es necesario que al finalizar la programación de la identificación y seguimiento de la tapa, evaluar el algoritmo midiendo el tiempo de ejecución de este y el consumo de recursos, la forma más eficaz de realizar es medir el desempeño de imágenes procesadas por segundo.

Los FDS (Frames Per Second) se conocen como un refrescamiento, ya que es la velocidad a la que un dispositivo actualiza una imagen. Donde a mayor sea el consumo de recursos del proceso una mayor refrescamiento.

Debido a los diversos refrescamientos es indispensable realizar una programación eficiente facilitando a los FDS, para que el ojo humano sea capaz de visualizar como fluye la imagen a través del HMI.

Hay que tener en cuenta la resolución de la imagen mostrada, esto es, el número de pixeles que componen la imagen, donde a mayor resolución más complejo son los cálculos. A continuación en la Tabla 4.1 se muestra un ejemplo de pruebas en distintos formatos de video para encontrar el óptimo entre la resolución de la pantalla y los FPS obtenidos. (Gonzales, 2016)

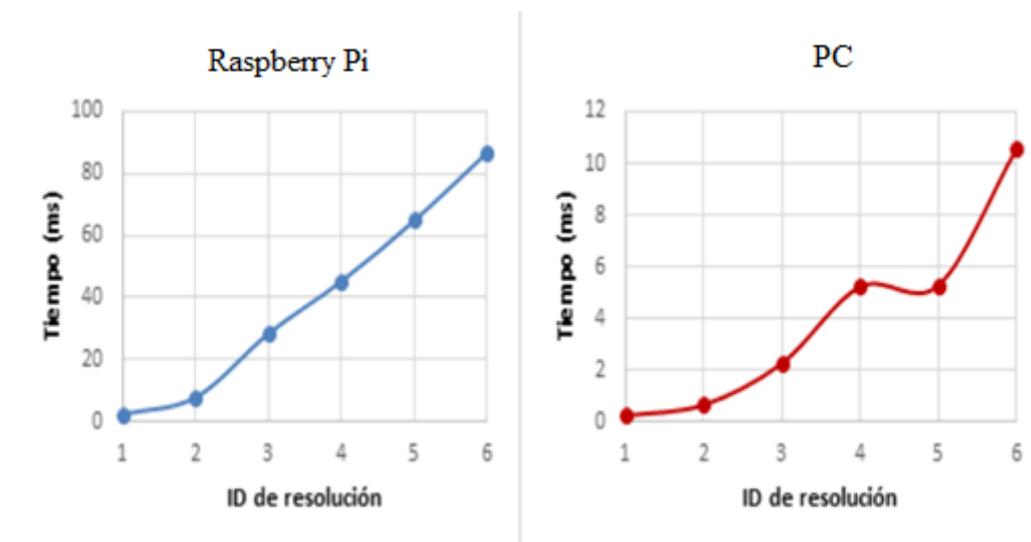
Tabla 4.1 Resolución de Pantalla Vs FPS

Veces Prueba	Pixeles	RPI	PC
		t(ms)	t(ms)
1	160x120	2,3348978	0,21601202
2	320x240	7,76519238	0,63930862
3	640x480	28,3802224	2,25667936
4	800x600	45,2535952	5,22323647
5	960x720	64,9059058	5,28137475
6	1280x720	86,8953928	10,6079639

Fuente: (Gonzales, 2016)

Este ejemplo se realizó con las Raspberry y el ordenador. Como se observa en la Figura 4.10 se puede conocer la resolución a partir de la cual el consumo de recursos es desmedido. El fin de este ejemplo es obtener la resolución que presenta una relación óptima entre la resolución adecuada y el consumo de recursos, en el presente proyecto se trabaja con una resolución de 640x480 Pixeles.

Figura 4.10 Comparación del tiempo de procesamiento en función de la resolución en las diferentes plataformas



Fuente: (Gonzales, 2016)

4.12. Producción manual

Tabla 4.2 Colocación manual de Liner en Industrias IEPESA

IPEsa COLOCACION LINER MANUAL						
TIEMPO INICIAL	ACCIONES	TIEMPO/# OPERADOR DOS	TAPAS / HORA	TAPA LISTA	TAPA VOLTEADA	TAPA RECHAZADA
16 tapas / minuto	Tapas en desorden volteo tapas manual	1 segundo / operador	960 HORA/TAPA	663 TAPAS	288 TAPAS	9 TAPAS
ciclo maquina 60 segundos	Ver fallas	3 segundo / operador				
	Colocar Liner	1 segundo / operador				
	Visualizar indicadores de inyectora	1 segundo / operador				
	Colocar materia prima en tolva de inyectora	4 minutos / operador				
	Pausas Activas	5 minutos / operador				
	Almuerzo	20 minutos / operador				

Elaborado por: Ricardo Briceño

En la Tabla 4.2 se observa el tiempo de inyección de las tapas por ciclo es decir se produce 16 tapas en 60 segundos, a partir de ahí se considera el tiempo de la colocación manual de *liner*, en base a todos proporcionado por el departamento de producción y

a las experiencias del operario en su jornada laboral como la ergonomía, pausas activas, almuerzo, con la finalidad de obtener la producción por hora y por turno.

4.13. Producción automática

Tabla 4.3 Producción automática de la máquina colocadora de *liner*

PRODUCCIÓN AUTOMÁTICA DE LA MÁQUINA COLOCADORA DE LINER							
TIEMPO INICIAL	ACCIONES	COLOCACIÓN DE LINER AUTOMÁTICO CON UN OPERADOR	TAPAS/HORA	TAPAS POR TURNO	TAPA LISTA	TAPA VOLTEADA	TAPA RECHAZADA
16 TAPAS EN 53 SEGUNDOS	Revisión de la tapa a través de visión artificial	2 segundos	1090 HORA/TAPA	8720 TAPAS	752 TAPAS/HORA	327 TAPAS/HORA	11 TAPAS/HORA
	Colocar <i>liner</i>	3.3 segundos					

Elaborado por: Ricardo Briceño

En la Tabla 4.3 se observa que el tiempo de proceso de inyección de tapas se bajó con la finalidad de incrementar la producción, al constatar que la máquina automatizada ofrece mejores prestaciones en cuanto no solo al tema de mejora de tiempos en colocación de *liner*, sino además cero manipulación de la tapa y el número de operadores se redujo de dos a uno.

4.14. Contraste de producción automática Vs manual

Tabla 4.4 Contraste de producción manual VS automático

COLOCACIÓN MAUNAL DE LINER				COLOCACIÓN AUTOMÁTICA DE LINER				INCREMENTO DE PRODUCCIÓN
Tiempo inicial del proceso	# de operadores	Tapas/hora	Tapas/ turno	Tiempo inicial del proceso	# de operadores	Tapas/hora	Tapas/ turno	
16 Tapas en 60 segundos	2	960 Tapas/hora	7680 Tapas/hora	16 Tapas en 53 segundos	1	1090 Tapas/hora	8720 Tapas/turno	130 Tapas/hora

Elaborado por: Ricardo Briceño

En la Tabla 4.4 se observa el contraste de producción entre la colocación manual del *liner* donde el tiempo inicial del proceso es de 16 tapas en 60 segundos y la automática el tiempo de proceso varia de 16 tapas en 53 segundos para así tener un incremento de producción de 130 tapas/hora, además de muchos benéficos antes mencionados.

CONCLUSIONES

El sistema diseñado en el presente trabajo permite que el proceso de colocación de *liner* en las tapas plásticas se ejecute automáticamente, de tal manera que la presencia de personas sea la mínima posible, añadiendo de igual manera un control por visión artificial que permite mejorar la cantidad y calidad de producto terminado. De esta manera se puede generar ahorro y mejora en la presentación de los productos.

El poder identificar plenamente cada etapa que compone el proceso total, permite tener la información total de cómo éste fluye, pudiendo así determinar las variables involucradas en el mismo.

La automatización integral mediante equipos mecánicos y electrónicos para el proceso de colocación de *liner* en las tapas plásticas facilitara el trabajo de los operarios ya que al realizar un proceso manual de 16 tapas por minuto se pierde eficacia a las 3 horas de trabajo, cuando una máquina realiza el mismo trabajo repetitivo sin verse afectada, e incluso se incrementará la producción a 18 tapas por minuto.

La implementación de un sistema de visión artificial para verificación de parámetros visuales como geometría, color e imperfecciones físicas de elementos añade valor técnico al sistema, puesto que se puede prescindir de recursos humanos que realicen esta actividad. La selección de los elementos es importante al momento de implementar el sistema completo, ya que de ellos depende la velocidad y la calidad de ejecución, de tal manera que en la presente maquina es necesario al menos 1.65 segundos para procesar una tapa mientras esta pasa por el rango de visión de la cámara. El uso del sistema de control Raspberry cumple con la tarea de gestionar imágenes a través de su procesador electrónico integrado, y realizar un control sencillo gracias a la presencia de un puerto de entradas y salidas digitales en su Hardware.

La búsqueda del sistema de control central para la ejecución del presente proyecto se basó en aspectos como la cantidad y calidad de información disponible, la presencia de proveedores de los equipos en el país, la facilidad de aprendizaje e implementación de lenguajes de programación, la presencia de puertos de entradas y salidas digitales de forma física, la facilidad de conexión a equipos de visualización y control, y

finalmente el precio de adquisición del mismo. Finalmente, se decidió por utilizar la Raspberry Pi 3 B+, equipo que cumple con todos los puntos anteriormente descritos. El sistema cumple con lo descrito en los enunciados, es aún un sistema de baja producción en comparación con sistemas especializados en la inspección de calidad mediante visión artificial. Esto se debe completamente a las características electrónicas de su hardware, características que limitarían al sistema si la velocidad de producción y control aumentara significativamente. Sin embargo, las características del sistema son aceptables para el proceso de verificación de tapas y colocación de *liner*.

La etapa de colocación de *liner* cumple con un tiempo de 3,3 segundos por tapa determinado por pruebas realizadas, ya que al tratarse de un proceso repetitivo y continuo se gestiona a través de las salidas y entradas digitales de la Raspberry, que mediante un algoritmo de programación lleva a cabo esta actividad sin problemas. Esta etapa está ligada completamente a la etapa de inspección de calidad, ya que de ella dependen las tapas que sean aceptadas.

Las comparaciones realizadas entre el proceso manual y automático indican claramente un incremento de producción en 130 tapas por hora, ya que al tratarse de un sistema automatizado podrá llevar a cabo un mayor número de ejecuciones de un proceso en específico siempre y cuando las características técnicas y de control sean las óptimas para así reducir errores ligados a problemas con su diseño.

RECOMENDACIONES

Para el presente proyecto se tendrá que tomar en cuenta que las líneas neumáticas para su alimentación ofrezcan una presión constante y además no exista caídas de voltaje de esta manera poder garantizar la ejecución de forma correcta de los diferentes equipos del sistema.

La correcta selección de los elementos de control permite tener los resultados esperados, es por esto que una investigación previa de la tecnología que existe en el mercado puede reducir mucho los tiempos de implementación de este tipo de proyectos.

Es muy recomendable realizar pruebas iniciales de visión artificial con la cámara del sistema con diferentes materiales y texturas para conocer cuáles de ellas son las más recomendables de usar en el sistema completo. Esto permite evitar señales erróneas al momento de procesar las imágenes. Los errores más comunes que se presentaron al desarrollar el proyecto fueron las reflexiones de luz que hacían que las imágenes tomaran colores diferentes al real y la iluminación que también distorsionaban los colores reales.

La tecnología actual permite tener a la mano equipos que pueden ser usados en diversas circunstancias, se recomienda evaluar las características de los equipos periféricos del sistema en el caso de que se necesite una actualización a futuro. En este caso se optó por usar una pantalla HDMI común para presentar la interfaz del usuario, así en una próxima oportunidad se puede cambiar por un sistema de visualización diferente tan solo con cambiar el tamaño del monitor.

REFERENCIAS

- Alejandro, V. P. (2009). *Introducción al diseño de micro robots móviles*.
- BalenaEtcher. (s.f.). Obtenido de Transfiere las imágenes del SO a tarjetas SD y unidades USB de forma segura y sencilla.: <https://etcher.io/>
- Barreño, C. I. (2012). Metodología Para la Transferencia Tecnológica en la Industria Electrónica Basada en Software Libre y Hardware Copyleft. *CASE*, 111-112-113.
- Corso, L. D. (2016). *Polipropileno*.
- Ecoplas. (2011). *poliestireno características y ventajas respecto al medio ambiente*.
- Edgar Ruiz Lizama, J. L. (2008). Desarrollo de una Interfaz hombre máquina orientada al control de procesos. *UNMSM*, 70-71.
- Enercon. (2013). *Guía para principales al sellado mediante inducción*.
- Europea, U. (2012). *Aplicación práctica de visión artificial en control de procesos industriales*. España.
- F. Ebel, S. I. (2008). Fundamentos de la técnica de automatización. *Festo*, 15.
- FESTO. (14 de 06 de 2017). *Programa Festo FluidSim Neumática 4.2 Español*. Obtenido de <http://mecatronicaayuda.blogspot.com/2017/06/les-traigo-todos-la-solucion-de-unos.html>
- Galindo, M. y. (2015). Productividad.
- Garavito, J. (2007). *Cursos de Proceso de Manufactura*. Colombia.
- Gonzales, C. P. (2016). Detección y seguimiento de objetos por colores en una plataforma raspberry. *Industriales*, 49-50.
- Mendoza, M. A. (s.f.). *Instrumentación simbología, sistema de unidades*.
- Microsof. (15 de 05 de 2017). *Como usar escritorio remoto*. Obtenido de <https://support.microsoft.com/es-ec/help/4028379/windows-10-how-to-use-remote-desktop>

- OpenCv. (s.f.). *OpenCv*. Obtenido de <https://opencv.org/>
- PILONI, M. E. (2017). *Liners, Cierres y Tapas para Envase y Embalaje*. Mexico.
- PLatero, C. (s.f.). *Introduccion a Vision artificial*. Dpto. Electrónica, Automática e Informática Industrial .
- PuTTY*. (08 de 07 de 2017). Obtenido de <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Python. (s.f.). Obtenido de <https://www.python.org/>
- Raspbian. (11 de 2018). Obtenido de <https://www.raspberrypi.org/downloads/raspbian/>
- Roca, G. I. (2005). *Estudio de las propiedades y aplicaciones industriales del politileno de alta densidad*. Guatemala.
- Ruiz, H. (s.f.). *Viosn Artificial*. España.
- Salgado, L. (s.f.). *Vision artificial fundamentos y aplicaciones*. Madrid.
- Sanchez, J. M. (2014). Diseño y Fabricacion de un molde para inyeccion de plasticos. *Comillas*, 17.
- Scanner, A. I. (s.f.). *Advanced IP Scanner*. Obtenido de <https://www.advanced-ip-scanner.com/es/>
- Soto, J. E. (2010). Modelo Integral de Productividad, aspectos importantes para su implementacion. *EAN*, 12.

ANEXOS

Anexo 1 Programación de visión

artificial

```
#Comando para permitir escritura con caracteres especiales como tildes, etc
#coding=utf-8
#Importación de librerías para generación de interfaz gráfica
from tkinter.ttk import *
from tkinter import *
from tkinter import font
from PIL import Image, ImageTk
import tkinter as tk
#Importación de librerías para procesamiento de imágenes
import cv2
import numpy as np
#Importación de librerías para uso del puerto GPIO
import RPi.GPIO as GPIO
#Importación de librerías necesarias para cálculos y procesos internos
import time
import math
import os
import threading
#Comando para seleccionar el tipo de cámara que se está utilizando
os.system('sudo modprobe bcm2835-v4l2')
#Variable que crea un frame con información "nula"
last_frame = np.zeros((480, 640, 2), dtype=np.uint8)
#Variable para capturar un frame desde la PiCamara
cap = cv2.VideoCapture(0)
#Variable para guardar el primer frame como fondo
fondo = None
#Declaración de variables y contadores
aux = 0 #1=Reciclaje 2=Rechazo 3=Buena/Producción
cntrc = 0 #Contador recicladas
cntrz = 0 #Contador rechazadas
cntac = 0 #Contador aceptadas
cntpr = 0 #Contador producidas
i=0 #Contador para tapas en cola
#Declaración de variables para definir el número de pin de cada electroválvula
mrz = 2
mrc = 3
ventosa = 4
cilvert = 17
detlin_cola = 14
cilvent = 15
sensor = 18
#Inicialización del módulo GPIO
```

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
#Inicialización de pines a usarse
GPIO.setup(mrz,GPIO.OUT) #EV
RECHAZO
GPIO.setup(mrc,GPIO.OUT) #EV
RECICLAJE
GPIO.setup(ventosa,GPIO.OUT) #EV
VENTOSA
GPIO.setup(cilvert,GPIO.OUT) #EV
CILINDRO VERTICAL LINER
GPIO.setup(detlin_cola,GPIO.OUT)
#EV DETENCIÓN
GPIO.setup(cilvent,GPIO.OUT) #EV
CILINDRO VENTOSA
GPIO.setup(sensor,GPIO.IN, GPIO.PUD_UP)
#Sensor
#Apagar todas las salidas, logica inversa
GPIO.output(mrz,GPIO.HIGH)
GPIO.output(mrc,GPIO.HIGH)
GPIO.output(ventosa,GPIO.HIGH)
GPIO.output(cilvert,GPIO.HIGH)
GPIO.output(detlin_cola,GPIO.HIGH)
GPIO.output(cilvent,GPIO.HIGH)
#Variable de valor de retardo
esp = 0.7
#SUBPROCESO PARA RECICLAJE Y RECHAZO
def rcrz():
#Definir el tipo de letra y tamaño para impresión en interfaz gráfica
cf2 = font.Font(family="Century Gothic", size=14, weight="bold")
#Llamar a las variables para un uso global y que sus valores puedan servir para todo el código
global aux
global cntrc
global cntrz
#Creación de variables tipo "tk" (Integer) para poder ser mostradas en la interfaz gráfica
recicladas=tk.IntVar(value=cntrc)
echazadas=tk.IntVar(value=cntrz)
#Si aux==1 se debe reciclar
if (aux == 1):
#Espera un tiempo de 400ms
time.sleep(0.4)
#Activa la electroválvula
GPIO.output(mrc,GPIO.LOW)
#Colorea el cuadro de color verde para indicar que está activo
```

```

Label(root, text="          ", font=cf2,
background="#00ff00").place(x=270, y=315)
#reciclaje
        #Espera un tiempo de 300ms
        time.sleep(0.3)
        #Desactiva la electroválvula

        GPIO.output(mrc,GPIO.HIGH)
#Colorea el cuadro de color gris para indicar
que está inactivo
Label(root, text="          ", font=cf2,
background="#555555").place(x=270, y=315)
#reciclaje
#Aumenta el contador de tapas recicladas
cntrc = cntrc + 1
#Asigna el valor al contador correspondiente
para ser mostrado posteriormente
        recicladas.set(cntrc)

#Encera la variable "aux" para un nuevo
control
        aux = 0
        #Imprime un mensaje en
pantalla
        Label(root, text="Accionar
Reciclaje          ",
font=cf2).place(x=20, y=590)
        #Si aux==2 se debe rechazar
        if (aux == 2):
                #Espera un tiempo de
2800ms
                time.sleep(2.8)
#Activa la electroválvula

        GPIO.output(mrz,GPIO.LOW)
#Colorea el cuadro de color verde para indicar
que está activo
Label(root, text="          ", font=cf2,
background="#00ff00").place(x=270, y=350)
#rechazo
#Espera un tiempo de 300ms
        time.sleep(0.3)
#Desactiva la electroválvula

        GPIO.output(mrz,GPIO.HIGH)
        #Colorea el cuadro de color
gris para indicar que está inactivo
Label(root, text="          ", font=cf2,
background="#555555").place(x=270, y=350)
#rechazo
#Aumenta el contador de tapas rechazadas
        cntrz = cntrz + 1
#Asigna el valor al contador correspondiente
para ser mostrado posteriormente
        rechazadas.set(cntrz)
#Encera la variable "aux" para un nuevo
control
        aux = 0
#Imprime un mensaje en pantalla

```

```

Label(root, text="Accionar Rechazo
", font=cf2).place(x=20, y=590)
#Imprime los valores de los contadores en la
interfaz
Label(root, textvariable=rechazadas, font=cf2,
width=10).place(x=1020, y=290)
Label(root, textvariable=recicladas, font=cf2,
width=10).place(x=1020, y=390)

#SUBPROCESO COLOCACIÓN DE LINER
def colocacion_liner(channel):

        #Definir el tipo de letra y tamaño para
impresión en interfaz gráfica
        cf2 = font.Font(family="Century
Gothic", size=14, weight="bold")

#Llamar a las variables para un uso global y
que sus valores puedan servir para todo el
código
        global i
        global cntpr

#Bucle para colocar el liner, estará funcionando
indefinidamente las veces que existan tapas
aceptadas
        while (i>=1):

#Imprime un mensaje en pantalla
Label(root, text="Colocación del Liner ",
font=cf2).place(x=20, y=590)

        #Resta contador de tapas
aceptadas
        i=i-1

#Espera un tiempo de 3200ms
        time.sleep(3.2)

        #Activa electroválvulas de
detención y ventosa

        GPIO.output(detlin_cola,GPIO.LOW
)

        GPIO.output(ventosa,
GPIO.LOW)

#Colorea el cuadro de color verde para indicar
que está activo
Label(root, text="          ", font=cf2,
background="#00ff00").place(x=270, y=455)
#detencion

```

```

Label(root, text="          ", font=cf2,
background="#00ff00").place(x=270, y=385)
#ventosa

#Espera un tiempo de 700ms

        time.sleep(esp)

#Activa electroválvula del cilindro vertical
“giratorio”

        GPIO.output(cilvert,
GPIO.LOW)

#Colorea el cuadro de color verde para indicar
que está activo

Label(root, text="          ", font=cf2,
background="#00ff00").place(x=270, y=420)
#cilindro vert liner

#Espera un tiempo de 1200ms

        time.sleep(1.2)

#Activa electroválvula del cilindro sujetador de
la ventosa y desactiva la ventosa

        GPIO.output(cilvent,
GPIO.LOW)

        GPIO.output(ventosa,
GPIO.HIGH)

# Colorea el cuadro de color verde para indicar
que está activo y de gris al que está inactivo

Label(root, text="          ", font=cf2,
background="#00ff00").place(x=270, y=490)
#cilindro ventosa

Label(root, text="          ", font=cf2,
background="#555555").place(x=270, y=385)
#ventosa

#Espera un tiempo de 700ms

        time.sleep(esp)

#Desactiva la electroválvula del cilindro
sujetador de la ventosa

        GPIO.output(cilvent,
GPIO.HIGH)

#Colorea el cuadro de color verde para indicar
que está activo

Label(root, text="          ", font=cf2,
background="#555555").place(x=270, y=490)
#cilindro ventosa

#Espera un tiempo de 700ms

        time.sleep(esp)

```

```

#Desactiva electroválvula del cilindro vertical
“giratorio”

        GPIO.output(cilvert,
GPIO.HIGH)

#Colorea el cuadro de color gris para indicar
que está inactivo

Label(root, text="          ", font=cf2,
background="#555555").place(x=270, y=420)
#cilindro vert liner

#Aumenta el contador de tapass producidas

        cntpr=cntpr+1

#Asigna el valor de las tapas producidas para
ser mostrado en la interfaz

        producidas=tk.IntVar(value=cntpr)

#Limpia el área de mensajes e imprime los
datos de la variable de tapas producidas

Label(root, text="          ",
font=cf2).place(x=20, y=590)

Label(root, textvariable=producidas, font=cf2,
width=10).place(x=1020, y=490)

#Comparación para saber si hay más tapas en
cola

        if (i>1): #Si hay tapas en cola
se debe dejar pasar una tapa y activar la
detención

#Desactivar electroválvula de detención

        GPIO.output(detlinCola,GPIO.HIG
H)

#Esperar 1500ms

        time.sleep(1.5)

#Activar electroválvula de detención

        GPIO.output(detlinCola,GPIO.LOW
)

#Colorea el cuadro de color verde para indicar
que está activo

Label(root, text="          ", font=cf2,
background="#00ff00").place(x=270, y=455)
#detencion

        else:

#Si no hay tapas en cola desactivar
electroválvula de detención

```

```
GPIO.output(detlin_cola,GPIO.HIGH)
H)
```

```
#Colorea el cuadro de color gris para indicar
que está inactivo
```

```
Label(root, text="", font=cf2,
background="#555555").place(x=270, y=455)
#detencion
```

```
#Comando para activar la interrupción externa
cada vez que se detecte un flanco de subida en
el sensor, esto indica que una tapa ha pasado
completamente
```

```
GPIO.add_event_detect(18, GPIO.RISING,
callback=colocacion_liner, bouncetime=800)
```

```
#SUBPROCESO PARA PROCESAMIENTO
DE IMÁGENES Y PROYECCIÓN DE
IMÁGENES EN PANTALLA
```

```
def show_vid():
```

```
#Variable para detectar cuando el sensor esté
activo o inactivo
```

```
inp = GPIO.input(18)
```

```
#Selección de tipo de fuente y tamaño
```

```
cf2 = font.Font(family="Century
Gothic", size=14, weight="bold")
```

```
#Llamar a las variables para un uso global y
que sus valores puedan servir para todo el
código
```

```
global aux
```

```
global cntac
```

```
global radio
```

```
#Creación de variables tipo "tk" (Integer) para
poder ser mostradas en la interfaz gráfica
```

```
aceptadas=tk.IntVar(value=cntac)
```

```
radio=tk.IntVar()
```

```
#Comprobar si se tiene acceso a la cámara
```

```
if not cap.isOpened():
```

```
Label(root, text="No se puede acceder a la
camara", font=cf2).place(x=20, y=590)
```

```
#Capturar un frame inicial para verificar
funcionamiento de la cámara
```

```
flag, frame = cap.read()
```

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

```
#Si la cámara funciona guardar en la variable
"last_frame" el primer frame capturado
```

```
if flag is None:
```

```
Label(root, text="Error!",
font=cf2).place(x=20, y=590)
```

```
elif flag:
```

```
global last_frame
```

```
last_frame = frame.copy()
```

```
#PROCESAMIENTO DE IMÁGENES,
GRABA UN FRAME QUE SERVIRÁ DE
FONDO Y COMPARARÁ LOS
SIGUIENTES FRAMES PARA
COMPROBAR SI EXISTE OBJETOS EN
MOVIMIENTO
```

```
#Transformar a escala de grises el frame
capturado
```

```
gris = cv2.cvtColor(frame,
cv2.COLOR_BGR2GRAY)
```

```
#Aplicar al frame gris un filtro de difusión
gaussiana
```

```
gris = cv2.GaussianBlur(gris, (21, 21),
0)
```

```
#Si la variable "fondo" no tiene información
(variable inicializada al encender el equipo)
guardar la imagen gris con el filtro gaussiano
```

```
global fondo
```

```
if fondo is None:
```

```
fondo = gris
```

```
#Restar un nuevo frame de la imagen de fondo
```

```
resta = cv2.absdiff(fondo, gris)
```

```
#Aplicar un threshold para mantener la imagen
estática y poder comprobar diferencias
```

```
umbral = cv2.threshold(resta, 25, 255,
cv2.THRESH_BINARY)[1]
```

```
#Aplicar una dilatación a la imagen
anterior para encontrar contornos nuevos
```

```
umbral = cv2.dilate(umbral, None,
iterations=2)
```

```
#Detectar contornos nuevos en comparación de
la imagen de fondo, contornos de objetos en
movimiento
```

```
contornosimg = umbral.copy()
```

```

im, contornos, hierarchy =
cv2.findContours(contornosimg,cv2.RETR_T
REE,cv2.CHAIN_APPROX_SIMPLE)

# -----
-----

#Definir los colores a detectar en formato HSV,
color amarillo

colorLower = (25, 100, 100)
colorUpper = (43, 255, 255)

#Cambiar imagen de color BGR a color HSV
hsv = cv2.cvtColor(frame,
cv2.COLOR_BGR2HSV)

#Aplicar máscara para detectar los colores
definidos

mask = cv2.inRange(hsv, colorLower,
colorUpper)

#Aplicar filtro de erosión para mejorar calidad
de imagen, eliminar ruido

mask = cv2.erode(mask, None,
iterations=2)

#Aplicar filtro de dilatación para mejorar
calidad de imagen, eliminar ruido

mask = cv2.dilate(mask, None,
iterations=2)

#Contar los contornos encontrados (objetos en
movimiento) del color definido. Contornos de
objetos amarillos

cnts = cv2.findContours(mask.copy(),
cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)[-2]

#Variable para definir el centro de los objetos
encontrados

center = None

#Definir variable para trabajar con procesos en
segundo plano para que no afecten al flujo
normal del código

d = threading.Thread(target=rcrz,
name='rcrz')

d.setDaemon(True)

#Si existen objetos en movimiento y se han
detectado contornos proceder:

if len(contornos)>0:

#Si existen contornos de objetos amarillos
proceder:

```

```

if len(cnts)>0:

#Detectar un contorno de objeto amarillo y
encontrar su área

c = max(cnts,
key=cv2.contourArea)

#Detectar las coordenadas "x" y "y" y definir
el radio del objeto detectado

((x, y), radius) =
cv2.minEnclosingCircle(c)

#Asignar el valor de radio a la variable de
formato tk para ser mostrada en la interfaz

radio.set(radius)

#Encerrar el área encontrada en las
coordenadas y el radio detectado

cv2.circle(frame,
(int(x), int(y)), int(radius), (100, 255, 255), 2)

#Colocar un punto
de color rojo en el centro del objeto detectado

cv2.circle(frame,
center, 4, (0, 0, 255), -1)

#Si el objeto está
sobre el rango de la coordenada "x" proceder:

if (x>360 and x<420
and aux==0):

#encontrar los momentos del objeto amarillo
detectado

M =
cv2.moments(c)

#Calcular el centro del objeto detectado

center = (int(M["m10"] /
M["m00"]), int(M["m01"] / M["m00"]))

#Si el dato del área indicado por "M["m00"]"
esta entre los rangos indicados proceder:

if
(M["m00"]>=68000 and M["m00"]<=70000):

aux=3 #Tapa Ok, tapa
aceptada que procederá a colocación del liner

elif
(M["m00"]>=62000 and M["m00"]<=65000):

aux=1 #Tapa reciclaje

#Activar el subproceso en segundo plano para
rechazo o reciclaje

```

```

d.start()
else:
    aux=2 #Tapa rechazo
#Activar el subproceso en segundo plano para
rechazo o reciclaje

d.start()
#Si la tapa es aceptada proceder:
if (aux == 3):
#Aumentar el contador de tapas aceptadas
    cntac = cntac + 1
#Asignar variable del contador de tapas
aceptadas para ser mostrado en pantalla
posteriormente
    aceptadas.set(cntac)
    aux = 0
#Imprimir mensaje en pantalla
    Label(root, text="Colocación
de Liner",
font=cf2).place(x=20, y=590)
#Cada vez que una tapa sea aceptada se
aumenta el contador de tapas en cola, que
permitirá al subproceso de colocación de liner
trabajar ininterrumpidamente mientras haya
tapas en cola
    global i
    i = i+1
#Si el sensor es activado o descativado,
colorear el cuadro en verde o gris
respectivamente
    if (inp == 0):
Label(root, text="", font=cf2,
background="#00ff00").place(x=270, y=200)
#sensor
    else:
Label(root, text="", font=cf2,
background="#555555").place(x=270, y=200)
#sensor
#Encontrar y dibujar contornos en las áreas del
objeto amarillo definido para ser mostrado en
pantalla, esto no influye en el procesamiento de
imágenes, solamente es para visualización
didáctica en pantalla

```

```

canny = cv2.Canny(mask, 180, 260)
(, contornos,_) =
cv2.findContours(canny.copy(),
cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(frame,contornos,-
1,(255,255,0), 2)
#Al frame actual que captura la cámara
cambiarlo de RGB a color BGR para una
muestra "normal" de colores en pantalla
    frame = cv2.cvtColor(frame,
cv2.COLOR_RGB2BGR)
#Obtener el frame actual y guardarlo en forma
de array para ser mostrado en pantalla sin
alteraciones
    img = Image.fromarray(frame)
#Transformar el frame actual en un objeto tipo
"tk" para poder ser mostrado en la interfaz
gráfica
    imgtk =
ImageTk.PhotoImage(image=img)
#Definir la pantalla principal de la interfaz
    lmain.imgtk = imgtk
#Configurar la pantalla de interfaz conforme a
los datos indicados al inicio del bucle principal
    lmain.configure(image=imgtk)
#Imprimir en pantalla los datos de las variables
del contador de tapas aceptadas y el radio
detectado de los objetos
    Label(root, textvariable=aceptadas,
font=cf2, width=10).place(x=1020, y=190)
    Label(root, textvariable=radio,
font=cf2, width=30).place(x=130, y=620)
#Refrescar la interfaz cada 100ms
    lmain.after(100, show_vid)
#SUBPROCESO PARA DIBUJAR LA
INTERFAZ VISUAL DEL HMI
def interfaz():
    #Definir tipos de letra y tamaños
    cf1 = font.Font(family="Century
Gothic", size=23, weight="bold")
    cf2 = font.Font(family="Century
Gothic", size=16, weight="bold")

```

```
cf3 = font.Font(family="Century Gothic", size=14, weight="bold")
```

```
#Colocar títulos y leyendas y dibujar recuadros en la interfaz
```

```
Label(root, text="SISTEMA DE INSPECCIÓN DE TAPAS PLÁSTICAS", font=cf1, width=61, height = 2, borderwidth=3, relief="ridge").place(x=5, y=24)
```

```
recuadro = Frame(root, width = 360, height = 483, borderwidth=3, relief="ridge").place(x=5, y=100)
```

```
recuadro = Frame(root, width = 360, height = 483, borderwidth=3, relief="ridge").place(x=995, y=100)
```

```
recuadro = Frame(root, width = 1350, height = 100, borderwidth=3, relief="ridge").place(x=5, y=582)
```

```
Label(root, text="SENSORES", font=cf2, width=22, borderwidth=2, relief="ridge").place(x=17, y=150)
```

```
Label(root, text="Sensor fotoeléctrico", font=cf3).place(x=20, y=200)
```

```
Label(root, text="ACTUADORES", font=cf2, width=22, borderwidth=2, relief="ridge").place(x=17, y=265)
```

```
Label(root, text="Reciclaje", font=cf3).place(x=20, y=315)
```

```
Label(root, text="Rechazo", font=cf3).place(x=20, y=350)
```

```
Label(root, text="Ventosa", font=cf3).place(x=20, y=385)
```

```
Label(root, text="Cilindro V Liner", font=cf3).place(x=20, y=420)
```

```
Label(root, text="Detencion", font=cf3).place(x=20, y=455)
```

```
Label(root, text="Cilindro Ventosa", font=cf3).place(x=20, y=490)
```

```
Label(root, text="TAPAS ACEPTADAS", font=cf2, width=22, borderwidth=2, relief="ridge").place(x=1007, y=150)
```

```
Label(root, text="Unidades", font=cf3).place(x=1200, y=190)
```

```
Label(root, text="TAPAS RECHAZADAS", font=cf2, width=22, borderwidth=2, relief="ridge").place(x=1007, y=250)
```

```
Label(root, text="Unidades", font=cf3).place(x=1200, y=290)
```

```
Label(root, text="TAPAS RECICLADAS", font=cf2, width=22, borderwidth=2, relief="ridge").place(x=1007, y=350)
```

```
Label(root, text="Unidades", font=cf3).place(x=1200, y=390)
```

```
Label(root, text="TAPAS PRODUCIDAS", font=cf2, width=20, borderwidth=2, relief="ridge").place(x=1007, y=450)
```

```
Label(root, text="Unidades", font=cf3).place(x=1200, y=490)
```

```
Label(root, text="Radio:", font=cf2).place(x=20, y=620)
```

```
Label(root, text=" ", font=cf3, background="#555555").place(x=270, y=200) #sensor
```

```
Label(root, text=" ", font=cf3, background="#555555").place(x=270, y=315) #reciclaje
```

```
Label(root, text=" ", font=cf3, background="#555555").place(x=270, y=350) #rechazo
```

```
Label(root, text=" ", font=cf3, background="#555555").place(x=270, y=385) #ventosa
```

```
Label(root, text=" ", font=cf3, background="#555555").place(x=270, y=420) #cilindro vert liner
```

```
Label(root, text=" ", font=cf3, background="#555555").place(x=270, y=455) #detencion
```

```
Label(root, text=" ", font=cf3, background="#555555").place(x=270, y=490) #cilindro ventosa
```

```
#PROCESO PRINCIPAL, BUCLE INFINITO
```

```
if __name__ == '__main__':
```

```
#Deshabilitar mensajes del módulo GPIO para que no se muestren en consola
```

```
GPIO.setwarnings(False)
```

#Definir nombre principal de la pantalla donde se colocarán los objetos del HMI, el nombre es "root", fijarse en el subproceso Interfaz, que todo objeto tiene de referencia el nombre "root", pues identifica la pantalla donde será colocado cada objeto

```
root=tk.Tk()
```

#Definir que la interfaz estará colocada en la pantalla "root"

```
lmain = tk.Label(master=root)
```

#Utilizar sistema de rejilla para colocar los objetos según las coordenadas que se les asignen

```
lmain.grid(column=0, rowspan=1,  
padx=1, pady=1)
```

```
#Colocar título a la ventana
```

```
root.title("Clasificación de Tapas  
Plásticas")
```

```
#Configurar resolución o tamaño de la  
ventana
```

```
lmain.configure(width=1366,  
height=680)
```

```
#Iniciar el subproceso INTERFAZ
```

```
interfaz()
```

```
#Iniciar el subproceso SHOW_Vid  
que ejecuta el procesamiento de imágenes
```

```
show_vid()
```

```
#Mantener el bucle activo  
indefinidamente
```

```
root.mainloop()
```

```
#Al salir del programa liberar la  
cámara y limpiar el puerto GPIO
```

```
cap.release()
```

```
GPIO.cleanup()
```

