



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

SISTEMA INTELIGENTE DE RECICLAJE “OPEN BOT”

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Electrónica y Comunicaciones.

ÁREA: Física y Electrónica

LÍNEA DE INVESTIGACIÓN: Sistemas Electrónicos

AUTOR: Cristian Israel Satuquinga Congacha

TUTOR: Ing. Mg. Víctor Santiago Manzano Villafuerte

AMBATO – ECUADOR

Febrero – 2021

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Investigación sobre el tema: "SISTEMA INTELIGENTE DE RECICLAJE 'OPEN BOT'", desarrollado por el señor Cristian Israel Satuquina Congacha, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, febrero de 2021



Firmado electrónicamente por:
VICTOR SANTIAGO
MANZANO
VILLAFUERTE

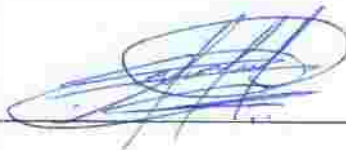
Ing. Víctor Santiago Manzano Villafuerte, Mg.

TUTOR

AUTORÍA DEL TRABAJO DE TITULACIÓN

El presente Proyecto de Investigación titulado: "SISTEMA INTELIGENTE DE RECICLAJE 'OPEN BOT'", es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero de 2021



Cristian Israel Satuquinga Congacha

C.C. 1805156724

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Satuquinga Congacha Cristian Israel estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad PROYECTO DE INVESTIGACION titulado "SISTEMA INTELIGENTE DE RECICLAJE 'OPEN BOT'" nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, febrero de 2021



Firmado electrónicamente por:
**ELSA PILAR
URRUTIA**

Ing. Elsa Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL



Firmado electrónicamente por:
**JUAN PABLO
PALLO NORONA**

Ing. Juan Pablo Pallo, Mg.
PROFESOR CALIFICADOR



Firmado electrónicamente por:
**ANA PAMELA
CASTRO**

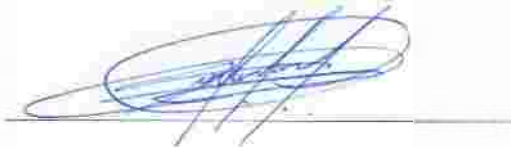
Ing. Pamela Castro, Mg.
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, febrero de 2021



Cristian Israel Satuquina Congacha

C.C. 1805156724

AUTOR

DEDICATORIA

A mi familia, especialmente a mis padres, mis hermanos, mi abuelita y mi tío quienes me supieron guiar por el camino correcto y brindar su apoyo de manera incondicional, mientras cursaba la carrera, lo cual me inspiró y motivó a seguir mirando mi objetivo adelante, nunca dejaré de agradecerlo.

A mi amiga, compañera y enamorada, por los momentos que compartimos ideas para la creación del presente proyecto.

Cristian Israel Satuquinga Congacha

AGRADECIMIENTO

A Dios por ser mi guía espiritual, brindarme salud y darme tranquilidad durante mi vida.

A mis padres, por su ayuda, su amor y apoyo económico para seguir adelante durante la carrera

A mis hermanos, mi abuelita y mi tío por su apoyo moral en el transcurso de la carrera.

A los docentes de la Facultad de Ingeniería en Sistemas Electrónica e Industrial quienes impartieron sus ideas, conocimientos y experiencias, para la formación académica en la carrera de ingeniería en Electrónica y Comunicaciones.

A mi tutor y docente Ing. Santiago Manzano quien, pese al corto tiempo supo guiarme durante la elaboración del proyecto.

A mis amigos con quienes cursaron la carrera por los momentos compartidos, lo cual generó ideas para el presente proyecto.

Cristian Israel Satuquinga Congacha

ÍNDICE GENERAL DE CONTENIDOS

Aprobación del Tutor	I
Autoría del trabajo de titulación.....	II
Derechos de Autor.....	IV
Aprobación del Tribunal de grado	III
DEDICATORIA	V
AGRADECIMIENTO.....	VI
ÍNDICE GENERAL DE CONTENIDOS.....	VII
ÍNDICE DE TABLAS	XI
ÍNDICE DE FIGURAS.....	XII
RESUMEN.....	XIV
ABSTRACT	XV
CAPÍTULO I.....	1
MARCO TEÓRICO.....	1
1.1 Antecedentes Investigativos	1
1.1.1 Contextualización del problema	4
1.2 Fundamentación teórica.....	4
1.2.1 Medio Ambiente.....	4
1.2.2 Contaminación Ambiental por PET	5
1.2.3 El Reciclaje.....	5
1.2.4 Consumo de PET.....	7
1.2.5 Sistemas Inteligentes	8
1.2.6 Sistemas Programados.....	9
a. Python.....	10
b. Librerías Python	10
1.2.7 Sistemas de Reciclado	11

a. Modo de captación PET.....	11
1.2.8 Reducción de Contaminantes mediante sistemas inteligentes.....	12
a. Incentivos de sistemas inteligentes.....	13
b. Productos de consumo.....	14
1.2.9 Efectos de Sistemas de Reciclados actuales.....	15
a. Impacto social del reciclaje.....	15
b. Impacto económico del reciclaje.....	15
1.2.10 Sistemas electromecánicos.....	16
a. Sensores.....	16
b. Actuadores.....	16
1.3. Objetivos.....	17
1.3.1 Objetivo General.....	17
1.3.2 Objetivos Específicos.....	17
CAPÍTULO II.....	19
METODOLOGÍA.....	19
2.1 Materiales.....	19
2.1.1. Acero Inoxidable.....	19
2.1.2. Raspberry Pi 4 Modelo B.....	19
2.1.3. Pantalla HDMI 10 pulgadas.....	20
2.1.4. Programación Python.....	20
2.1.5. Motor a pasos NEMA.....	20
2.1.6. Driver motor a pasos.....	21
2.1.7. Servo Motor 25Kg.....	21
2.1.8. Ultrasónico HC-SR04.....	22
2.1.9. DHT-22.....	22
2.1.10. Fotorresistor.....	22
2.1.11. Final de Carrera.....	23

2.1.12. Fuente ATX.....	23
2.1.13. Cámara Web.....	24
2.1.14. Editor de Código.....	24
2.2. Métodos.....	24
2.2.1 Modalidad de la Investigación.....	24
2.2.2 Recolección de Información.....	25
2.2.3 Procesamiento y Análisis de Datos.....	26
2.2.4. Desarrollo del Proyecto.....	26
CAPÍTULO III.....	28
RESULTADOS Y DISCUSIÓN.....	28
3.1 Análisis y discusión de los resultados.....	28
3.2 Desarrollo de la propuesta.....	28
3.2.1 Sistema de reciclaje convencionales.....	29
3.2.2 Tecnologías actuales en los sistemas de reciclajes.....	30
3.2.3 Captación de envases PET.....	34
3.2.4 Requerimiento electromecánico para el prototipo del sistema.....	34
a. Controladores del sistema.....	34
b. Sensores.....	35
c. Actuadores.....	38
d. Scanner.....	39
e. Dispositivo de Salida.....	40
3.2.5 Sistema de reciclado de envases PET.....	41
a. Medición y adquisición de datos (Sensorización).....	41
b. Sistema de gestión y visualización.....	41
c. Sistema de control mecánico.....	42
d. Sistema de dispensación.....	43
e. Desarrollo de interfaz de usuario.....	43

d. Base de Datos	50
3.2.6 Placa de Control	51
3.2.6 Construcción del contenedor “OPEN BOT”	55
3.2.9 Pruebas del funcionamiento	60
3.2.7 Factibilidad Económica	64
CAPÍTULO IV	66
CONCLUSIONES Y RECOMENDACIONES.....	66
4.1 Conclusiones.....	66
4.2 Recomendaciones	67
BIBLIOGRAFÍA.....	69
Anexos.....	74
Anexo 1: Datasheet de Raspberry Pi 4.....	74
Anexo 2: Datasheet de sensor HC-SR04	77
Anexo 3: Datasheet de final de carrea.....	79
Anexo 4: Datasheet de sensor DHT-22.....	82
Anexo 5: Datasheet de servo motor 25 Kg	84
Anexo 6: Datasheet de motor a paso Nema 17	85
Anexo 7: Datos Técnicos de Pantalla HDMI 10.1 pulgadas.....	86
Anexo 8: Manual de Usuario OPEN BOT.....	87
Anexo 9: Código de interfaz gráfica desarrollada en Python.	99

ÍNDICE DE TABLAS

Tabla 1: Relación costo y tiempo de descomposición por kilo de residuos	6
Tabla 2: Incentivos por Reciclar	14
Tabla 3: Contenedores inteligentes de reciclaje.....	31
Tabla 4: Características de OPEN BOT.....	32
Tabla 5: Comparativa técnico de miniordenadores.....	35
Tabla 6: Comparativa técnico sensores Ultrasónicos	36
Tabla 7: Comparativa técnico elemento electrónico para detección de objeto.....	36
Tabla 8: Comparativa técnico sensores de temperatura.....	37
Tabla 9: Comparativa técnico servo motor	38
Tabla 10: Comparativa técnico motor a pasos	39
Tabla 11: Comparativa técnico módulos de detención de códigos de barra.....	40
Tabla 12: Comparativa técnico dispositivo de salida	40
Tabla 13: Lista de envases aptos para el sistema.....	61
Tabla 14: Relación PET sin compresión y comprimido por el sistema OPEN BOT	62
Tabla 15: Presupuesto de construcción del sistema	64

ÍNDICE DE FIGURAS.

Figura 1: Símbolo de Reciclaje	6
Figura 2: Estructura sistema inteligente.	9
Figura 3: Acero inoxidable.	19
Figura 4: Raspberry Pi 4.	20
Figura 5: Pantalla HDMI 10.1 pulgadas.	20
Figura 6: Motor a pasos Nema 17.....	21
Figura 7: Driver A4988.....	21
Figura 8: Servo 25 Kg.....	21
Figura 9: Ultrasónico Hc-SR04	22
Figura 10: Sensor DHT-22	22
Figura 11: Fotorresistor	23
Figura 12: Final de carrera.....	23
Figura 13: Fuente ATX.....	23
Figura 14: Web Cam.....	24
Figura 15: Logo Sublime Text.....	24
Figura 16: Forma de reciclaje mediante estructura metálica	30
Figura 17: Envase PET en tacho de basura.....	33
Figura 18: Diagrama de acceso al sistema.....	44
Figura 19: Diagrama esquemático del sistema	45
Figura 20: Frame teclado del sistema	46
Figura 21: Frame Principal	46
Figura 22: Frame usuario.....	47
Figura 23: Frame administrador	48
Figura 24: Frame nuevo usuario	48
Figura 25: Preguntas se seguridad	49

Figura 26: Frame recuperar clave	49
Figura 27: Datos del sistema.....	50
Figura 28: Base de datos del sistema	50
Figura 29: Tabla administrador.....	50
Figura 30: Tabla de envases vinculados al sistema	51
Figura 31: Tabla de usuarios registrados	51
Figura 32: Fase de motores.....	52
Figura 33: Conexión de drives motores a pasos	52
Figura 34: Fase acondicionamiento de la señal	53
Figura 35: Conexión de arranque seguro del sistema.....	53
Figura 36: Representación del esquema electrónico.....	54
Figura 37: PCB del sistema	54
Figura 38: PCB del sistema vista 3D.....	55
Figura 39: Vista isométrica frontal del sistema	56
Figura 40: Vista isométrica posterior del sistema.....	56
Figura 41: Vista isométrica con puerta de acceso.....	57
Figura 42: Esquema mecánico de compresión	57
Figura 43: Conexión del sistema	58
Figura 44: Vista frontal.....	59
Figura 45: Vista en perspectiva	59
Figura 46: Mensaje de sistema de envase no registrado.....	60
Figura 47: Envases dentro del sistema.....	63
Figura 48: Incentivo expendido por el sistema.....	63
Figura 49: Diagrama Frame Buttons	99
Figura 50: Diagrama Frame Usuario	102
Figura 51: Diagrama Frame Nuevo usuario	107
Figura 52: Diagrama Cambio de Clave	117

RESUMEN

El presente proyecto de investigación, tiene el propósito de reducir los contaminantes por Tereftalato de polietileno (PET) en el medio Ambiente, fomentando una cultura de reciclaje para la sociedad, con la ayuda de tecnologías coordinadas, programadas y configuradas en el sistema. Este sistema mejora la salud ambiental en zonas urbanas y rurales, haciendo efectivo el desarrollo del proyecto sistema de reciclaje “OPEN BOT”.

El sistema desarrollado cuenta con una configuración de captación y dispensación de incentivos (regalo al usuario) por la acción de reciclar. El sistema evalúa parámetros como: si es un envase PET, el usuario se encuentra registrado, aún existe espacio en el contenedor y el usuario tienen créditos disponibles para reclamar el incentivo.

El proyecto desarrollado cuenta con materiales de calidad para el funcionamiento en exteriores, al contener materiales como acero inoxidable que es adaptable a climas de alta humedad, logrando abarcar alrededor de 350 envases PET en su interior, cumpliendo de forma óptima su propósito de gestionar residuos sólidos para posteriormente ser tratados.

El sistema de reciclaje “OPEN BOT”, es de código abierto (Open Source) y mejorable en un futuro.

La parte de control cuenta con dos configuraciones: configuración visual y configuración electrónica. La unidad central de procesos del sistema se encuentra comandada por una placa Raspberry Pi 4, su interfaz de usuario fue desarrollada con el lenguaje Python que es un lenguaje interpretado de alto nivel y a su vez gestiona actuadores para su proceso.

Palabras clave: PET, Open Source, Raspberry Pi, Python, contaminación ambiental, desechos sólidos, reciclaje.

ABSTRACT

This research project aims to reduce polyethylene terephthalate (PET) pollutants in the environment, promoting a culture of recycling for society, with the help of coordinated, programmed and configured technologies in the system. This system improves environmental health in urban and rural areas, also it makes effective development of the project recycling system "OPEN BOT".

The developed system has a configuration for capturing and dispensing incentives (gift to the user) for the action of recycling. This evaluates parameters such as: if it is a PET container, the user is registered, there is still space in the container and the user has credits available to claim the incentive.

The developed project has quality materials for outdoor operation, by containing materials such as stainless steel that is adaptable to high humidity climates, managing to cover around 350 PET containers inside, optimally fulfilling its purpose of managing solid waste to later be treated.

The recycling system "OPEN BOT" is open source (Open Source) and can be improved in the future.

The control part has two configurations: visual configuration and electronic configuration. The central process unit of the system is commanded by a Raspberry Pi 4 board, the user interface was developed with the Python language, which is a high-level interpreted language and manages actuators for your process.

Keywords: *PET, Open Source, Raspberry Pi, Python, environmental pollution, solid waste, recycling.*

CAPÍTULO I

MARCO TEÓRICO

1.1 Antecedentes Investigativos

Al finalizar la investigación bibliográfica en los principales repositorios de las universidades, así como en diversas bases de datos de artículos científicos, se encontraron los siguientes antecedentes:

En el año 2019, Mazharul Islam, Mohammad Farhan Ferdous, Md Shajidur Rahman Almazee, Md Shahriar Dipto, Kayanat Tahera Sultana, Azmir Hasan, en la Ciudad de Spokane, WA, en Estados Unidos, realizaron la Investigación de *“Depósito y reembolso de botellas de plástico reciclable Sistema (RPBDRS)”* se diseñó una máquina que pueda ayudar a reducir el aumento exponencial de la contaminación plástica. El dispositivo funciona depositando los recipientes de plástico en la entrada de la máquina. Luego, la máquina evaluará varios parámetros sobre la botella a través de una serie de sensores sofisticados. Después de la evaluación, si la botella resulta ser de plástico, alentará a la gente en general a usarla. Además, el marco tiene un sistema de energía dual, uno de los cuales está conectado a la red mientras que el otro funciona con energía solar. Esta máquina también tiene un Sistema de posicionamiento global (GPS) que ayuda a las personas a encontrarlo y usarlo. También existe un sistema de observación remoto (servidor en la nube) para el host del dispositivo [1].

En el año 2018, Pravin Dhulekar, S. T. Gandhe, Ulhas P. Mahajan, en la Ciudad de Pune en India, realizaron la Investigación de *“Desarrollo de la máquina de reciclaje de botellas utilizando algoritmo de aprendizaje automático”*, el cual utilizó un

Raspberry Pi 2 conectado con una cámara y un sistema audio visual para el proceso de incentivación al usuario que se entrega mediante tickets de recompensa impreso en una impresora térmica de recibos [2], realizando un proceso de reciclaje efectivo y a bajo costo.

En el año 2018, Yu-Ping Liao, Rwei-Chang Lu, Sheng-Ying Wu, Ping-En Cheng, Guo-Cheng Xu, en la ciudad de Taoyuan en Taiwán, realizaron la Investigación de *“El robot para el reciclaje basado en machine learning”* donde se diseñó un robot para clasificar los artículos para su reciclaje. Identifica automáticamente los objetos y sus ubicaciones en función de la imagen de profundidad, los toma con su brazo mecánico y, finalmente, los coloca en la papelera de reciclaje correspondiente. El robot se basa en tecnología de imágenes en profundidad con un algoritmo de aprendizaje profundo para reconocimiento y categorización [3]. Para ayudar a la protección del medio ambiente y reducir la contaminación.

En el año 2017, Wisdom Gen P. Dumpayan, Matthew Lawrence M. De Mesa, Nathalie Danielle F. Yucor, Eden T. Gabion, Jacqueline D. Reynoso, Gabriel Rodnei M. Geslani, en la ciudad de Manila en Filipinas, realizaron la Investigación de *“Máquina expendedora inversa de botellas bidireccionales de plástico con microcontrolador y sistema de valor almacenado mediante radiofrecuencia y Tecnología de escáner de identificación (RFID)”* en el cual utilizó un microcontrolador que dirige el funcionamiento de los diversos dispositivos de entrada (sensores y teclado) y de salida (pantalla y motores) conectados a él [4]. La máquina acepta botellas de plástico y las acredita como puntos, que, a su vez, pueden usarse para comprar productos. La operación de la máquina exhibió precisión en el reconocimiento de cuentas RFID, distinguiendo entre botellas plásticas y no plásticas.

En el año 2017, Tritubot una empresa ecuatoriana realizó una investigación de *“Diseño de una máquina trituradora de botellas plásticas PET”* en la cual se desconoce por políticas de la empresa el hardware y software utilizados, este sistema

permite al usuario insertar un envase y recibir el incentivo de \$ 0,02 dólares por envase, la cual es innovador y da grandes resultados favoreciendo al medio ambiente y a la cultura de reciclaje en el Ecuador a un costo relativamente asequible por empresas como Petroecuador [5].

En el año 2017, Marloun Sejera, Joseph Bryan Ibarra, Anrol Sarah Canare, Lyra Escano, Dianne Claudinne Mapanoo, John Phillip Suaviso, en la ciudad de Manila en Filipinas, realizaron la investigación de “*Contenedor de basura automatizado basado en frecuencia independiente y segregador de botellas de plástico y latas*”, el sistema contiene un contenedor de basura automatizado que distingue las botellas de plástico y las latas. El sistema utiliza un micrófono piezoeléctrico para la adquisición de la señal de entrada y un comparador para la eliminación del ruido, el sistema se basó en la respuesta de frecuencia promedio del objeto cuando golpea la plataforma [6]. Este estudio ayudará a la comunidad en la recuperación de residuos y tiene como objetivo hacer que el reciclaje sea más conveniente.

En el año 2016, Maofic Farhan Karin, Hasan U. Zaman, Khandaker Sharif Noor, en la Ciudad de Dhaka, Bangladesh en India, realizaron la investigación de “*Diseño basado en hardware e implementación de una máquina de reciclaje de botellas con FPGA*” El cual se utiliza una FPGA Altera DE2-115 usando verilog HDL, que permite al usuario reciclar botellas plásticas y recibir recompensas por ellas resultando un proyecto favorable al medio ambiente y a un bajo costo [7].

En el año de 2016, Witaya Srigul, Mongkol Kupimai, Prajuab Inrawong, en la ciudad de Nakhon Ratchasima en Tailandia, realizaron la investigación de “*Clasificación de plástico basada en la correlación de color RGB*” un método de correlación múltiple para clasificar los tipos de botellas de plástico, botellas PET y no PET [8]. La correlación de cada imagen de botella de plástico se calculó a partir del histograma de todos los colores (RGB) de las imágenes de botella de plástico, que consiste de una cámara digital (12 Megapíxeles) y envía una imagen en formato JPEG hacia un

ordenador con un sistema de programación basado en MATLAB, donde se realiza la clasificación de los diferentes tipos de botellas.

1.1.1 Contextualización del problema

La reducción de contaminantes en el medio ambiente de envases plásticos de forma convencional esta entre el 15% y 25%, para su gestión y de acuerdo a las medidas planteadas por el Ministerio del Ambiente y ONU Medio Ambiente, sugiere la reducción de la contaminación por plásticos mediante sistemas óptimos que oferten beneficios a la parte y contraparte.

Dando a lugar un sistema de reciclado inteligente con expendedores automáticos que genere incentivos por ejecutar la acción de reciclado, también que el sistema llame la atención en la sociedad al ser interactivos y de fácil acceso.

El presente proyecto de investigación cumplirá la finalidad de contribuir con el medio ambiente, satisfaciendo directamente la acción de la sociedad en la salud del ecosistema, utilizando tecnologías modernas para mejorar los centros turísticos y locales comerciales, de la misma forma ayudará indirectamente al gobierno local al notar la reducción de los contaminantes de envases PET en la provincia

1.2 Fundamentación teórica

1.2.1 Medio Ambiente

El programa de las Naciones Unidas para el medio ambiente definió al Medio Ambiente como un exterior físico y biológico donde vive el ser humano y otros organismos, que constituye en sí mismo un todo complejo y cuyos diversos componentes se influyen recíprocamente [9]. Con este concepto global se puede reducir la escala y los espacios de estudio, especificando las características del medio ambiente en cada nivel de análisis.

El medio ambiente natural, producido solo por fuerzas de la naturaleza, se ve afectada en muchas ocasiones por fuerzas ajenas a ella, que involucran las acciones de los seres humanos, llevando a una modificación de su medio y dando el análisis de medio ambiente antrópico, que está configurado por un conjunto de elementos físicos-naturales, sociales, culturales y económicos, que interactúan entre si con cada uno de los individuos y con la comunidad que la rodea, estableciendo su forma, carácter y relaciones, llevando a una integración directa con el ser humano como un elemento de gran trascendencia por la razón que el hombre transforma el medio ambiente natural a su beneficio [9]. Este medio ambiente antrópico es conocido medio ambiente natural en extrema alteración o una ciudad, conformada por edificaciones, vías y comercio por la necesidad de mantener un elevado nivel de interacción social, económica y cultural.

1.2.2 Contaminación Ambiental por PET

La sociedad actual cuenta con zonas altas en biodiversidad rodeadas por poblaciones urbanas y rurales, muchos cuentan con servicio de recolección de basura y son potencialmente consumidores de bebidas envasadas en PET, pero esto no disminuye la contaminación ambiental, las personas que consumen bebidas en estos lugares lo desechan en basura, lo queman o lo entierran, ocasionando grandes problemas de contaminación ambiental por ser el plástico más utilizado en el mundo actualmente. El PET tarda mucho tiempo en degradarse, cada vez habrá más envases, y la degradación de los mismos resulta cada vez más complejo [10]. Estos afectarán de manera directa la presentación de los espacios naturales, además científicos aseguran que aquellos plásticos que sean llevados por las corrientes de las lluvias terminarán en ríos y algunos otros en el mar, en donde también liberarán sus componentes químicos afectando la salud y estabilidad de especies más vulnerables.

1.2.3 El Reciclaje

En el análisis de la Real Academia Española de la lengua, el reciclaje o reciclar posee en sus acepciones que nos aproxima de forma cierta a la idea de reciclaje como acción de reciclar, su significado: “Someter un material usado a un proceso para que se pueda volver a utilizar. Someter repetidamente una materia a un mismo ciclo, para ampliar o

incrementar los efectos de ésta.” De esta manera reducir el impacto de la contaminación en el medio ambiente [11].



Figura 1: Símbolo de Reciclaje [10]

El reciclaje es un proceso que consiste en la elaboración de un material o un producto totalmente nuevo a través de la reutilización de productos desechos o ya utilizados que llevan a un ciclo de tratamiento total o parcial para obtener una materia prima, dicho ante la perspectiva del agotamiento de recursos naturales y para una eliminación de forma eficaz de los propios desechos [11].

El Reciclaje es por tanto una de las alternativas utilizadas en la reducción del volumen de los desperdicios, el cual consiste en volver a utilizar materiales que fueron desechados como PET, y que aún son aptos para elaborar otros productos o volver a elaborar los mismos.

Tabla 1: Relación costo y tiempo de descomposición por kilo de residuos [11]

Residuos Sólidos de reciclaje		
Residuo	Precio por kilo	Años de descomposición
Cartón	0,10 \$	0,25
Plástico PET	0,50 \$	450
Plástico Duro	0,20 \$	10 – 20
Plástico Soplado	0,10 \$	450
Papel (oficina)	0,10 \$	1
Periódico	0,15 \$	0,12
Cobre	2 \$	x
Aluminio	0,25 \$	10 – 100
Acero	0,25 \$	50

Bronce	1,10 \$	x
Baterías	0,25 \$	100
Placa computadoras	1 \$	1000
Placa teléfono	3 \$	1000
Vidrio	0,03 \$	4000

Elaborado por: El Investigador

En la **Tabla 1** se aprecia el precio por kilo y el tiempo de que se demora en descomponerse en años de dos residuos más desechados en la actualidad en el mercado, observando su nivel de resistencia y lo perjudicial que es para el medio ambiente.

1.2.4 Consumo de PET

En la actualidad el Tereftalato de polietileno (PET), es uno de los materiales plásticos de envasado que más ha incrementado su consumo. El abaratamiento en los costes de fabricación y el desarrollo de tecnologías que mejoran substancialmente las propiedades de las botellas de PET, han permitido un crecimiento notable del número de sus aplicaciones [11]. Entre ellas cabe destacar el envasado de agua mineral, aceite, zumos, bebidas isotónicas, detergentes, productos de higiene corporal y productos farmacéuticos entre otros.

Los envases de PET están adecuados para contener líquidos a presión, el envasado de bebidas carbónicas como su principal aplicación. Debido a esta versatilidad, su uso se ha incrementado de manera exponencial, generando un número considerable de desechos post consumo, los cuales, causan daños directos al medio ambiente, e incrementan notablemente la fracción de residuos sólidos urbanos y rurales. El uso excesivo del PET por el sector empresarial y doméstico, ha originado una forma de contaminación ambiental por desechos plásticos, por su alta resistencia a los agentes biológicos y atmosféricos, son productos nocivos al medio [11]. EL deterioro

ambiental afecta a las tres cuartas partes de la humanidad, ejemplo de ello es que, al alrededor del 50% de la población mundial carece de sistemas adecuados de saneamiento en sentido general, incluyendo los Residuos Sólidos Urbanos.

Las primeras botellas de PET aparecieron en el mercado en 1980. Desde entonces el consumo ha ido creciendo hasta alcanzar, en 1995, las 80.000 toneladas. En el país en el año 2012 se produjeron 1 406 millones de botellas, de las cuales se lograron recuperar 511 millones de los embotelladores y 624 millones de los centros de acopio y recicladores, logrando una recolección total de 1 136 millones de PET [12].

Actualmente se logra reciclar al menos 500 botellas diarias gracias a sistemas inteligentes como Tritubot [13], de las aproximadamente 12.739,01 toneladas de residuos y desechos sólidos que se generan a diario en el país. De las cuales el 11% es residuo PET que las industrias de bebidas no alcohólicas ubican en el mercado. A su vez aportan con el 70% del reciclaje de estos envases colocados en el mercado con la ayuda de recolectores humanos. [14]

1.2.5 Sistemas Inteligentes

Son sistemas complejos capaz de resolver acciones programadas y multidisciplinarias de una forma automática reduciendo tiempo a los usuarios y facilitando una forma efectiva de realizar la acción predestinada. Con el avance de la tecnología estos sistemas cada vez son más eficientes y de gran escala que contiene miniordenadores comunicados hacia una red de sensores para la toma de decisiones, a su vez poseen cloud service para el respaldo de información o ejecución de programas que requieren cloud computing [15].

Los sistemas inteligentes también son capaces de ejecutar servicios a nivel local como almacenamiento SQL y servicios web, esta facilidad para generar y almacenar datos de información ha provocado la necesidad de desarrollar eventos de factibilidad en uso

y perfeccionar sistemas inteligentes autónomos en el análisis de la información que ofrezcan instrumentos que faciliten la toma de decisiones ante una ejecución de evento, en la **Figura 2** se observa la estructura general de un sistema inteligente [15].

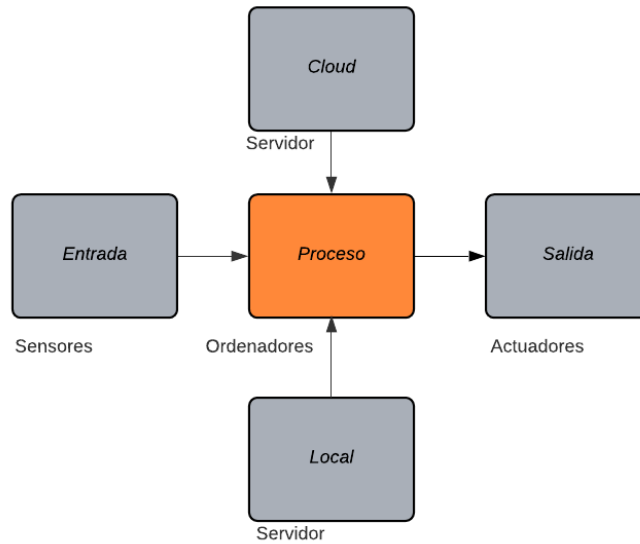


Figura 2: Estructura sistema inteligente. [11]

1.2.6 Sistemas Programados

Una forma eficiente de efectuar un sistema inteligente a gran escala y funcional esta emergida a la programación ante eventos repetitivos y de toma de decisiones programadas o generadas por el sistema, para la generación de estas acciones se considera la ayuda de lenguajes de programación de alto nivel como Python, JavaScript, C#, GO entre otras, las cuales ayudan a llevar de forma efectiva al sistema inteligente. [16]

Las ventajas que ofrecen los sistemas inteligentes programados sin duda son mayores a las desventajas, como una mención es “Aplicaciones innovadoras de robótica”, “Sistemas innovadoras de Smart Things” y “Sistemas innovadoras de Telemetría” generando investigaciones diarias en este campo gracias a la tecnología usada. [16]

a. Python

Python un lenguaje de alto nivel interpretado y compilado, enfocado principalmente a la legibilidad y facilidad de aprendizaje y uso, es un lenguaje orientado a objetos, soporta paradigmas como la programación funcional y, la programación imperativa, es un lenguaje multiplataforma. Python, por medio del uso de bibliotecas o librerías, aprovecha las posibilidades concretas que le brinda cada plataforma, de modo que esos programas funcionen indistintamente en cualquier ordenador. Además, Python es software libre, y se distribuye bajo la licencia “Python Software Foundation License”. Esto, entre otras cosas, significa que se distribuye gratuitamente y no necesita del pago de licencias para su uso, ya sea privado o comercial. [17]

b. Librerías Python

El uso de herramientas en Python día a día es más sencillo, Python 3 la última versión de este lenguaje tiene gran gama de aplicaciones y herramientas para el desarrollo, visual, electrónico, automatización e inteligencia en robótica o artificial. Para su ejecución únicamente requiere de “pip install Nombre-librería” las dependencias utilizadas con mayor frecuencias y configuraciones por defectos son las siguientes [18].

❖ Entornos Visuales

-Tkinter: Configurada en el sistema por defecto.

-PyQT: Dependencia desarrollada y requiere de instalación para su uso.

❖ Conexión SQL

-pymysql: Dependencia desarrollada y requiere de instalación para su uso.

-mysql-connector-python: Dependencia desarrollada y requiere de instalación para su uso.

-mysql-connector: Dependencia desarrollada y requiere de instalación para su uso.

❖ Interfaz GPIO

Esta dependencia está diseñada para mini ordenadores como las Raspberry Pi en todas sus versiones, la cual se encuentra configurada por defecto.

❖ Procesamiento grafico

-OpenCV: Dependencia para el tratamiento de imágenes. Esta dependencia es un método empleado para la segmentación de una imagen que debe adecuarse convenientemente a las características de las imágenes en estudio. Donde se muestra una fotografía ventral del en RGB y su descomposición por bandas, para su tratamiento en código. [19]

-pyzbar: Dependencia para el tratamiento de imágenes de códigos de barra y códigos QR, utilizado junto a la dependencia OpenCV.

1.2.7 Sistemas de Reciclado

El objetivo principal de un sistema de reciclado es la reducción de contaminantes al medio ambiente de forma sencilla y con menor impacto social, la cual ejecuta el correcto tratamiento de forma efectiva al desecho, plástico, metálico, de tipo vidrio, de tipo papel, de tipo orgánico e inorgánico, entre otros tipos. Dando el tratamiento adecuado para cada tipo de desecho que esta predestinada hacia la industria que lo requiera, los sistemas de reciclaje únicamente se encargan de clasificar estos desechos para una facilidad de tratamiento de acuerdo a su tipo, muchos de estos sistemas de reciclado pueden ser manuales o automáticos [20].

El reciclaje en la industria es un extenso campo dentro del sector para tratar de conservar el medio ambiente en el que se pueden reciclar una amplia variedad de residuos, con el objetivo de mejorar su reutilización directa e indirecta, ya sea por la transformación en materia prima reutilizable o por la producción de otros bienes de consumo masivo. Una gran parte de estos residuos son plásticos que se presenta en diversas variedades, una mención es el PET, material para los conservantes líquidos. Debido a estos problemas de necesidad se efectúa un sistema de reciclaje automático que recepte grandes cantidades de desechos con el menor impacto posible [21].

a. Modo de captación PET

Los sistemas de captación de envases PET de la actualidad tienen diferentes configuraciones, que benefician al sistema y al usuario, la más común es de

almacenamiento. Una forma óptima de almacenar un PET a reciclar sería con la reducción de volumen mediante diversos tratamientos, como compactación o trituración, de esta manera aprovechando al máximo su capacidad. Un contenedor común puede llegar a abarcar hasta 100 envases en pleno funcionamiento, dicho de otro modo, el mismo contenedor configurado con la opción de compactación puede llegar a abarcar hasta el doble.

1.2.8 Reducción de Contaminantes mediante sistemas inteligentes

El principal objetivo de los sistemas de reciclado es reducir la tasa de impacto al medio ambiente con el reciclaje facilitando dicha labor a la sociedad, incluso, motivar que el reciclaje se vuelva un hábito de cultura (o penalizar a quien no recicle). Son máquinas, robots o contenedores con las últimas tecnologías, además cuentan con sistemas que interpretan a la persona que recicla, el tipo de residuo que introduce al contenedor, y por ello dan dinero al reciclar. Los sistemas inteligentes de reciclaje alertan a la empresa encargada cuando han llegado a su capacidad máxima de almacenada de desperdicios. [22]. Los contenedores inteligentes o sistemas de reciclado inteligente compactan los residuos para una mayor captación, estos sistemas en la actualidad se adaptan a personas con discapacidad con ayuda de tecnologías innovadoras que adaptan los orificios de acceso hacia el usuario. Los sistemas actualmente modernos cuentan con pantallas de interacción en las que ofrecen información y esto hace que el reciclaje sea innovador y entretenido.

Se referencia un país del continente europeo para concretar esta idea, en España se generan 1,3 miles de millones de residuos sólidos urbanos al año y está aumentando que tiene una consecuencia evidente: los costes de su recolección también aumentan. El organismo predice que se dispararán en un 83%, 375 mil millones de dólares en 2025. Una alternativa de contrarrestar este efecto es fabricación de cubos o contenedores de basura inteligentes que puede ahorrar hasta un 40% de los precios, según indica Compology. La compañía ha creado sensores que detectan los niveles de basura en los contenedores y notifican a los camiones de eliminación de desechos cuando es el momento de recoger. [23]

Además, Getxo una empresa fabricante de contenedores inteligentes ya ha acogido una prueba preliminar en la que usuarios han podido interactuar con el contenedor. Su objetivo ha sido conocer su experiencia de uso y optimizar su funcionamiento, dando grandes resultados en su acogida. [24]

El reciclaje en España aumentó un 42 % durante 2019 respecto al año anterior, con un total de 57.520 toneladas gestionadas por Recyclia, que agrupa a las Fundaciones Ecopilas, Ecofimática, Ecoasimelec y Ecolum. EFEverde las cuales son fabricantes de contenedores inteligentes. [24]

En Valencia, una cadena hotelera prefiere sistemas de reciclajes como contenedores inteligentes que reduzcan gastos y optimicen tiempos de trasportes de residuos, dando lugar a nuevas innovaciones tecnológica para la recolección de desechos. [25]

Por los antecedentes mencionados la reducción de desechos que contaminan el medio ambiente en la última década ha generado controversia y es la causa que mayor afecta la sociedad de forma económica y cultural, es motivo suficiente para ser imprescindible un sistema inteligente para la clasificación de desechos plásticos, lo malo es que el ser humano no tiene la suficiente capacidad para usarlos por diferentes factores como: falta de conocimiento, desinterés, incapacidad o disposición [22].

a. Incentivos de sistemas inteligentes

Los contenedores o sistemas de reciclajes inteligentes están configurados para ofrecer algo por reciclar, los más frecuentes son de 0,02 dólares por envase, también podemos encontrar los que ofrecen tickets y pasajes, la **Tabla 2** enlista los incentivos dados por este tipo de sistemas.

Tabla 2: Incentivos por Reciclar

Incentivos de sistemas de reciclajes		
Incentivo	Tipo	Valor estimado
Ticket	Impresión	0,02 \$
Pasajes	Impresión	0,02 \$
Dinero	Físico	0,02 \$

Elaborado por: El Investigador

b. Productos de consumo

Para poder ofertar un incentivo al usuario se analiza los productos de consumo masivo, la fecha de durabilidad, formas y temperaturas optimas de almacenamiento. Existen productos con durabilidad de un mes como productos de durabilidad de hasta un año. Los cuales en un contenedor programable con la opción de dispensar productos se puede poner cualquier tipo de estos. Otro dato para poder establecer el tipo de producto es la frecuencia en que una persona ingiere una bebida gaseosa o mineral [26] . De acuerdo a los datos de un artículo una persona en Ecuador consume 24.6 litros de gaseosa, 41.2 litros de agua embotellada mensuales, dato de relevancia para el cálculo de límite de durabilidad en un incentivo a ser expendido si el usuario es nuevo en el sistema.

$$Sdp = \frac{\text{litros de bebidas consumidas}}{\text{litros de bebida mas consumida}} * \text{precio del envase}$$

$$Sdp = \frac{24.6 + 41.2}{0,5} * 0.02 = 2.63$$

Sdp → Saldo de durabilidad de producto

Con el análisis del resultado obtenido un sistema de reciclaje puede ofertar productos que duren fechas de un mes y estén valorados entre 0 a 2,63 dólares.

1.2.9 Efectos de Sistemas de Reciclados actuales

Los incentivos que generan estos sistemas inteligentes de reciclaje en la actualidad no parecen ser bastos para la sociedad, estos sistemas generan tickets de Metrobús, tickets de pago en supermercado, pagos en monedas como \$0,02 dólares, que ocasionan problemática a la hora de gestionar el pago, a pesar de la tecnología que posee no parece ser una correcta adaptación a la sociedad que se vive en la actualidad [27]. Con todas estas problemáticas que poseen los sistemas actuales, ha generado la necesidad de diseñar un sistema óptima de reciclaje ayudando directamente al medio ambiente.

En mención a esto en Ecuador con la ayuda de la empresa Tritubot y Petroecuador en el último año se ha logrado reciclar alrededor de 313 mil botellas en costa y sierra, dando paso a los sistemas de reciclaje inteligentes sean una medida para combatir la contaminación al medio ambiente ante los desechos plásticos de tipo PET. [27]

a. Impacto social del reciclaje

El ministerio de medio ambiente con la iniciativa recicla propone modelos de recuperación y aprovechamiento de residuos sólidos, así como también el fortalecimiento de una cultura del reciclaje en las ciudades, promoviendo un trabajo conjunto entre Municipios, recicladores y ciudadanía [28]. “Ecuador Recicla” uno de los eventos base contribuye su compromiso por el cambio del país. Juntos por una ciudad más limpia y presentable.

b. Impacto económico del reciclaje

El Gobierno busca trabajar con políticas, programas y proyectos que se enmarcan en el Acuerdo por la Competitividad, Empleo, e Innovación aplicaciones potenciales para la adecuada gestión de residuos, generando empleo digno a través de la aplicación de iniciativas de reciclaje, a su vez priorizando la prevención de la contaminación ambiental, lo cual representa ahorros económicos y reducción de impactos ambientales [28].

1.2.10 Sistemas electromecánicos

Estos sistemas contienen actuadores inherentemente mecánicos cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide) [29] [30]. Dependiendo de el origen de la fuerza el actuador se denomina “neumático”, “hidráulico” o “eléctrico”. La que permite ejecutar un sin número de tareas programadas con la ayuda de sensores y actuadores.

a. Sensores

Los sensores se definen como un dispositivo de entrada que provee una salida manipulable de la variable física medida también se analiza cómo este convierte la energía de un dominio a otro. Entonces, se puede decir que la definición del concepto de sensor está íntimamente relacionada con la definición de transductor, ya que un sensor siempre hará uso de un transductor [31]. No obstante, la principal diferencia entre un transductor y un sensor radica en que el sensor no solo cambia el dominio de la variable física medida, sino que además la salida del sensor será un dato útil para un sistema de medición.

b. Actuadores

Es capaz de convertir una variable física en otra que tiene un dominio diferente, es encargado de ejecutar la acción determinada por el sistema de procesamiento de la información, también es un dispositivo con la capacidad de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía, los actuadores se clasifican en dos grandes grupos [31]:

1. Por el tipo de energía utilizada: actuador neumático, hidráulico y eléctrico
2. Por el tipo de movimiento que generan: actuador lineal y rotatorio.

1.3. Objetivos

1.3.1 Objetivo General

Implementar un Sistema inteligente de Reciclaje “OPEN BOT”

1.3.2 Objetivos Específicos

- ❖ Analizar los sistemas actuales de reciclajes
- ❖ Diseñar un sistema electromecánico para la captación de envases PET y dispensación de incentivos.
- ❖ Diseñar una interfaz de interacción usuario máquina que evalúe condiciones de usuario y ejecute acciones acordes a sus datos.

Con los objetivos planteados se procede a su análisis correspondiente:

El objetivo principal del presente proyecto de investigación es implementar un sistema de reciclaje inteligente que capte envases PET mediante una automatización para la gestión correcta de residuos, permitiendo reducir la contaminación ambiental y promoviendo el reciclaje como una cultura en la sociedad.

Con el estudio de los sistemas que ejercen la misma acción en la actualidad, se obtienen la relevancia e importancia que los deben tener para la reducción de contaminantes en el medio ambiente y que tecnologías que utilizan para ejercer dicha acción, para esto se cumplen las siguientes actividades.

- ❖ Estudio de tecnologías actuales implementado en los sistemas de reciclajes
- ❖ Determinación del impacto social/económico de los sistemas actuales
- ❖ Estudio de Hardware y Software empleado en sistemas electromecánicos.

El sistema de captación de envases y dispensación, permite acreditar y desacreditar aportes de reciclaje, en base a su frecuencia de uso mediante una programación, para esto se cumplen las siguientes actividades.

- ❖ Análisis de comportamiento de sensores, para captación de envases PET.

- ❖ Análisis de datos para expender incentivos hacia el usuario.
- ❖ Representación esquemática del sistema de reciclado de envases PET.

Para su correcto funcionamiento del sistema con los usuarios que desean reciclar, es necesario generar una aplicación de escritorio que adquiera y escriba datos a un servidor local, administrando y gestionado sus credenciales, para esto se cumplen las siguientes actividades.

- ❖ Diseño del interfaz de usuario.
- ❖ Implementación de prototipo del sistema con un miniordenador.

CAPÍTULO II

METODOLOGÍA

2.1 Materiales

Los materiales correspondientes al proyecto se describen a continuación.

2.1.1. Acero Inoxidable

Por su aleación de hierro y carbono además por su resistencia al oxido y la corrosión se seleccionó este tipo de material para el cobertor del contenedor “OPEN BOT”, una representación gráfica se observa en la **Figura 3**.



Figura 3: Acero inoxidable.

2.1.2. Raspberry Pi 4 Modelo B

Su versatilidad y su amplia memoria con respecto a sus antecesoras da un rendimiento óptimo ante sistemas de interfaces gráficas y la interacción con los usuarios brinda una mejor experiencia en cuanto a fluidez, motivo por el cual es óptimo para “OPEN BOT”, una representación gráfica se observa en la **Figura 4**.



Figura 4: Raspberry Pi 4.

2.1.3. Pantalla HDMI 10 pulgadas

La interacción máquina-usuario es importante en cuanto se refiere a un sistema de registro e ingreso con una ejecución visual, con tecnología touch capacitiva la pantalla brinda una mejor experiencia ante este tipo de sistemas, que se incluye para “OPEN BOT”, una representación gráfica se observa en la **Figura 5**.

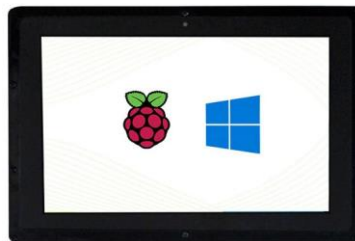


Figura 5: Pantalla HDMI 10.1 pulgadas.

2.1.4. Programación Python

Por su simplicidad, versatilidad y rapidez de desarrollo, Python interactúa como comunicador entre máquina-usuario, ejecutando ordenes que recibe mediante el sistema que se incorpora en “OPEN BOT”

2.1.5. Motor a pasos NEMA

Su utilidad en la electrónica gracias a su fácil operación es versátil, para actuar ante un evento que requiera un cierto y exacto revolución, y muchos de estos son utilizados en sistemas expendedoras y además se incluye en el sistema “OPEN BOT” , una representación gráfica se observa en la **Figura 6**.

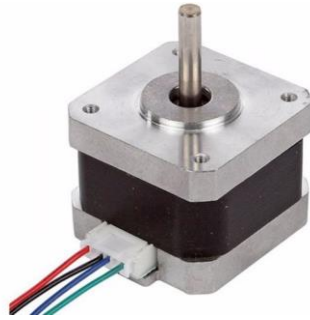


Figura 6: Motor a pasos Nema 17

2.1.6. Driver motor a pasos

En la electrónica se tiene diferentes soluciones para operar un actuador, “OPEN BOT” está constituido por el driver A4988 que brindan operaciones de corrientes acorde al motor y carga que ejecuten en ella, una representación gráfica se observa en la **Figura 7**.



Figura 7: Driver A4988

2.1.7. Servo Motor 25Kg

Para el sistema este actuador tiene la funcionalidad de operar en diferentes etapas, además gracia a su gran torque, puede aplicar fuerzas en áreas relativas, por lo cual está configurado en el sistema “OPEN BOT” , una representación gráfica se observa en la **Figura 8**.



Figura 8: Servo 25 Kg

2.1.8. Ultrasónico HC-SR04

Su utilidad en el sistema es de medir el nivel del contenedor para dar paso al ingreso de un envase a ser reciclado, enviar los datos al ordenador para su análisis y realizar su acción según lo establecido en su algoritmo, una representación gráfica se observa en la **Figura 9**.



Figura 9: Ultrasónico Hc-SR04

2.1.9. DHT-22

En el sistema este sensor tiene la funcionalidad de medir constantemente la temperatura del interior del contenedor, para adecuar la temperatura en los incentivos a ser expendidos, el dato se envía al ordenador para que el sistema “OPEN BOT” ejecute su algoritmo, una representación gráfica se observa en la **Figura 10**.



Figura 10: Sensor DHT-22

2.1.10. Fotorresistor

Este elemento electrónico en el sistema “OPEN BOT” cumple con la función de sensor de seguridad para el usuario, lee si el acceso al envase se encuentra obstruida y el sistema no cierra su acceso, una representación gráfica se observa en la **Figura 11**.



Figura 11: Fotorresistor

2.1.11. Final de Carrera

En el sistema este elemento electrónico cumple la función de detectar si algún envase su ingresada, dando paso al sistema a ejecutar a acción de compactar, una representación gráfica se observa en la **Figura 12**.



Figura 12: Final de carrera

2.1.12. Fuente ATX

Por sus niveles de corrientes y diferentes valores de tenciones, este elemento brinda al sistema la energía para su óptimo funcionamiento, una representación gráfica se observa en la **Figura 13**.



Figura 13: Fuente ATX

2.1.13. Cámara Web

En el sistema este gadget cumple la función de un scanner de código de barras, permitiendo abaratar costos, una representación gráfica se observa en la **Figura 14**.



Figura 14: Web Cam

2.1.14. Editor de Código

Es un software utilizado para la configuración y elaboración de la aplicación del sistema “OPEN BOT” , una representación gráfica se observa en la **Figura 15**.



Figura 15: Logo Sublime Text

2.2. Métodos

2.2.1 Modalidad de la Investigación

El presente proyecto para dar solución a los problemas sociales y económicos causados por la contaminación ambiental con envases PET, se tomaron en cuenta la aplicación de los conocimientos adquiridos durante el ciclo de duración en la Carrera de Ingeniería en Electrónica y Comunicaciones, que dan lugar a solucionar este problema mediante el desarrollo de un sistema de reciclaje “OPEN BOT”, el cual utiliza las siguientes técnicas:

Investigación Aplicada

Se enmarcó dentro de investigación aplicada por el análisis de las tecnologías utilizadas actualmente para el tratamiento de contaminantes al medio ambiente y a su vez la forma de incentivar a la humanidad con esta tecnología, la cual va directamente ejecutado en el lugar de desarrollo de la investigación, ejecutando nuevas técnicas de implementación para un óptimo funcionamiento del sistema.

Investigación Bibliográfica

Se enmarcó dentro de investigación bibliográfica por el nivel de información requerido para el análisis de nuevas tecnologías, métodos de reducción respecto a contaminantes plásticos, la comunicación y transición de datos entre el sistema/usuario mediante Frontend local que requieren información de fuentes como libros, páginas web, publicaciones, revistas y artículos científicos.

Investigación de Campo

El presente trabajo de investigación se enmarcó dentro de la investigación de campo ya que se requiere la recolección de información con ayuda del método de observación permitiendo realizar un estudio detallado de lugares con mayor índice de contaminación por envases plásticos, además lugares donde una solución motive al reciclaje y sea recompensada a su beneficio, observado en lugares de mayor concurrencia cotidiana de lugares público y/o privados.

2.2.2 Recolección de Información

La recolección información se obtuvo principalmente de libros, revistas científicas, paper's y proyectos de titulación desarrollado en los últimos años, cada una de estas relacionadas y vinculadas a Sistemas Inteligentes, Sistemas de Reciclaje, Sistemas Electrónicos, Servidores con Bases de Datos, Sistemas Frontend, y Sistemas de Comunicación por lo que se requiere de Investigación bibliográfica, también se realiza a través de la investigación de campo, en lugares concurridos sean públicos y/o privados, y analizando sus respectivas necesidades a través de una observación a la

sociedad, con esta información se realizará un diseño eficiente y confiable para reciclaje de envases plásticos.

2.2.3 Procesamiento y Análisis de Datos

En el proceso y análisis de datos se vincula la información de los diferentes medios digitales y físicos que documentan respecto a los puntos a analizar de una manera crítica y estos puntos son:

- ❖ Revisión de la información recopilada.
- ❖ Estudio de la propuesta de solución planteada para reducir los contaminantes de envases PET en el medio ambiente.
- ❖ Determinación de la mejor propuesta de solución.
- ❖ Planteamiento de la propuesta de solución.

2.2.4. Desarrollo del Proyecto

Para poder cumplir con los objetivos planteados y a su vez ejecutar el desarrollo del proyecto debe cumplirse las siguientes actividades que lleven a la implementación del sistema de reciclaje “OPEN BOT”:

- ❖ Análisis de sistemas de reciclaje convencionales.
- ❖ Estudio de tecnologías actuales implementado en los sistemas de reciclajes.
- ❖ Determinación del impacto social/económico de los sistemas actuales.
- ❖ Estudio de Hardware y Software empleado en sistemas electromecánicos.
- ❖ Análisis de comportamiento de sensores, para captación de envases PET.
- ❖ Análisis de datos para expender incentivos hacia el usuario.
- ❖ Análisis de requerimiento electromecánico para el prototipo del sistema.
- ❖ Representación esquemática del sistema de reciclado de envases PET.
- ❖ Determinación de niveles de tolerancia de hardware para el sistema.
- ❖ Diseño del interfaz de usuario.
- ❖ Evaluación de datos con la interacción de usuarios.

- ❖ Interpretación de pruebas del funcionamiento en Software de Programación.
- ❖ Implementación de prototipo del sistema con un miniordenador.
- ❖ Elaboración del informe final de la propuesta.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

La implementación de un sistema de reciclaje inteligente de nombre “OPEN BOT”, permite mejorar de manera óptima la forma de reciclar un envase PET, ayudando directamente a la salud del medio ambiente y fomentando el reciclaje en la sociedad, también el sistema mejora la gestión de residuos PET al para su óptimo tratamiento.

3.2 Desarrollo de la propuesta

La investigación realiza un desarrollo de un contenedor de botellas PET de diferentes tipos de bebidas circulantes en el mercado, para el reciclaje y no perjudicar el medio ambiente, adecuándose en lugares de mayor comercio o turismo.

El sistema “OPEN BOT”, permite a los usuarios a través de una interfaz gráfica, incrementar puntos en su cuenta, por medio del reciclaje con el objetivo de obtener un incentivo de la máquina expendedora programada para expulsar un producto disponible.

El sistema administra cuentas de usuarios en una base de dato local de quienes reciclan sus envases PET, lleva una opción de validación de código para no perder créditos obtenidos, un conteo de envases reciclados, una opción de cambio de clave, validación de usuarios registrados, validación de usuario para el registro, validación de preguntas de seguridad, validación de envases PET, registro de envases PET en modo

administrador, validación de almacenamiento acorde a la detención de nivel del contenedor con una capacidad aproximada de 350 envases.

En el área de captación del sistema, se encuentra un control electrónico y mecánico que válida el ingreso mediante sensores y actuadores como: sensor nivel del contenedor, sensor de seguridad de usuario, sensor de detección de envase en el área de compresión. Posteriormente es comprimido y almacenado. De esta forma sumando crédito a personas quienes reciclan.

3.2.1 Sistema de reciclaje convencionales

En diversos lugares de concurrencia masiva como en parques, colegio, universidades o lugares públicos, se ha logrado observar que únicamente existen depósitos de botellas hechos con hierro en forma de Botella gigante y cubierta de la propia envase PET a su alrededor, no obstante, los usuarios prefieren no utilizar o pasar de alto esta manera de optar por el reciclaje, por diferentes puntos como:

- ❖ No le llama la atención.
- ❖ No se siente motivado hacerlo
- ❖ No saben a dónde se dirige el envase
- ❖ No tienen beneficios de esto

Las autoridades incentivan mediante seminarios o charlas a reciclar para una mejor presentación del lugar, ubicando estas estructuras metálicas en la zona donde se pueda observar con facilidad, en muchas ocasiones estas estructuras pasan días sin recibir algún envase PET; debido a factores de mal hábito humano o simplemente deciden poner en un tacho de basura como se observa en la **Figura 16**. Pese a esto existen sistemas que reciclan mediante el transporte y almacenamiento por la actividad humana, para ser transportados al lugar de tratamiento del envase. Esta actividad tiene un promedio de pago de 100 dólares semanales [32] .



Figura 16: Forma de reciclaje mediante estructura metálica

Elaborado por: El investigador

En Ecuador existen 38 estaciones de servicios que tienen sistemas de reciclaje que interactúa con el usuario implementado 08 de febrero de 2019 que actualmente están en funcionamiento, ubicados en las ciudades de: Quito (15), Guayaquil (3), Santo Domingo (5), Manabí (6), Ambato (5) y Riobamba (4). [33]

3.2.2 Tecnologías actuales en los sistemas de reciclajes.

En la actualidad existen grandes avances tecnológicos que renuevan la forma de vida del ser humano. Estas tecnologías son desarrolladas respecto a su propósito con geolocalización, sensores de presencia, interpretación gráfica, interfaz de usuario, red de sensores o técnicas de procesamiento automático mediante ordenes preprogramadas, pese a esto la tecnología no son de provecho por el mundo al abordar los nuevos retos en la gestión de residuos. Sobre todo, aquellos que tienen que ver con su reciclaje. Como plataformas integrales para toda la cadena de gestión de residuos, que permiten un seguimiento de los camiones y contenedores, información de su contenido y así mejorar el tratamiento y la gestión de residuos. O para el desarrollo de tecnologías que facilitan el control y la optimización del ciclo de tratamiento de esos residuos [34].

Estas tecnologías permiten un mejor trazado de los residuos de entrada y que estos puedan ser convertidos en nueva materia prima con la que fabricar nuevos productos

y la forma de reciclar se va quedando obsoletas; debido a que la sociedad no pone como prioridad el reciclaje. [34]

Existen organismos como el Ministerio del Ambiente y ONU Medio Ambiente que involucran el reciclaje como una actividad obligatoria para las empresas que tengan un consumo masivo de plásticos, además para la ciudadanía sugiere que se debe practicar en la vida cotidiana como hábito [12].

Las recolectoras o sistemas de reciclajes actuales tienen la tecnología como primordial en esta acción, pero muchas de ellas únicamente captan envases utilizando tecnologías mecánicas para abrir o cerrar el contenedor sin proveer un incentivo para los usuarios que reciclan. En el país el contenedor existe uno que es “TRITUBOT”, este sistema escanea el código de barras si es un envase PET lo almacena y genera 0.02 dólares como recompensa a esta acción, utiliza un escáner de Código de Barra, un sistema mecánico y un sistema eléctrico para ejercer esta función [5].

La tecnología en el país designada para el reciclaje que es obtenida por el gobierno no incentiva a la sociedad a realizar esta opción de forma correcta, a diferencia de Ecuador, los países europeos llevan años de investigación, que una forma de combatir la contaminación es elaborando contenedores inteligentes interactivos [35, 36, 37, 38, 39], como se analiza en la **Tabla 3**.

Tabla 3: Contenedores inteligentes de reciclaje

Sistema	Sensores	Características	Recolección	Funcionalidad
Tritubot	<ul style="list-style-type: none"> Scanner 	<ul style="list-style-type: none"> Electrónico Almacena los envases en su interior mediante una compuerta que se acciona al leer el código de barra 	<ul style="list-style-type: none"> PET 	Ofrece \$ 0,02 por cada envase reciclado

BioBox	<ul style="list-style-type: none"> • Scanner 	<ul style="list-style-type: none"> • Electrónico • Almacena envases al leer un código de barra mediante un brazo mecánico 	<ul style="list-style-type: none"> • PET 	Ofrece puntos para ser cambiados por servicios
Bigbelly	<ul style="list-style-type: none"> • Presencia • Scanner 	<ul style="list-style-type: none"> • Electrónico • Compacta los desechos para más almacenamiento 	<ul style="list-style-type: none"> • Desechos 	Gestiona residuos mediante una plataforma en la nube
Vidortec	<ul style="list-style-type: none"> • Movimiento • Indicador de estado • Voltaje 	<ul style="list-style-type: none"> • Electrónico • Muestra el porcentaje del contenedor 	<ul style="list-style-type: none"> • Cartón • PET 	Separa residuos para un mejor manejo de ellos
Recyclia	<ul style="list-style-type: none"> • Sensor PET • Movimiento 	<ul style="list-style-type: none"> • Electrónico • Clasifica los desechos 	<ul style="list-style-type: none"> • PET • Cartón • Electrónica 	Almacena residuos para ser transportados
Sensoneo	<ul style="list-style-type: none"> • Distancia 	<ul style="list-style-type: none"> • Electrónico 	<ul style="list-style-type: none"> • Residuos 	Almacena residuos para ser gestionados

Elaborado por: El investigador

Estos sistemas son de derechos reservados, Los cuáles no puede ser replicados sin el consentimiento del autor, a diferencia de OPEN BOT es de código abierto. Además, OPEN BOT tiene un mecanismo de dispensador junto a un sensor de temperatura para mantener el incentivo en óptimas condiciones para su consumo, también OPEN BOT ofrece mayor prestación como se observa en la **Tabla 4**.

Tabla 4: Características de OPEN BOT

Sistema	Sensores	Características	Recolección	Funcionalidad
OPEN BOT	<ul style="list-style-type: none"> • Sensor de seguridad • Sensor de Envase • Sensor de nivel de Almacenamiento 	<ul style="list-style-type: none"> • Electrónico • Almacena usuarios en su base de datos • Gestiona un registro de 	PET	Ofrece puntos para después ser reclamados por un sistema de dispensador un incentivo de su agrado

	<ul style="list-style-type: none"> • Sensor de Temperatura 	<p>envases PET circulantes en el mercado</p> <ul style="list-style-type: none"> • Comprime el volumen de los envases • Almacena los envases mediante un código de barras • Ofrece puntos por el reciclaje que realice 		
--	---	--	--	--

Elaborado por: El investigador

Respecto a lo mencionado en Ecuador existe desinterés por reciclar al no disponer de estas tecnologías y optan por desechar en los tachos de basura como se observa en la **Figura 17**.



Figura 17: Envase PET en tacho de basura

Elaborado por: El investigador

3.2.3 Captación de envases PET.

Los sistemas de captación de envases son contenedores inteligentes utilizados en forma de recicladores programados, que utilizan diversos sensores para captar envases PET como su funcionalidad. Estos sistemas poseen sensores de presencia del envase y lectores de códigos de barra, que comunican el estado en tiempo real a los equipos de recolección. Además, la forma de captar los envases de los contenedores inteligentes, es mediante una compuerta mecánica o electrónica controlada por un medio micro procesado para su efectividad, dando acceso a su almacenamiento para posteriormente ser trasladado al centro de tratamiento [5, 36].

3.2.4 Requerimiento electromecánico para el prototipo del sistema.




El sistema OPEN BOT para el funcionamiento necesita de tarjeta controladora capaz de ejecutar un lenguaje interpretado de alto nivel para el desarrollo de su interfaz, y controlar actuadores mediante la lectura de datos de los sensores.

Los actuadores que se requieren para OPEN BOT necesitan tener la capacidad de realizar un movimiento angular para las aperturas de accesos al contenedor mediante una señal eléctrica, también necesitan generar un número de movimiento exacto para la opción de dispensar incentivos al usuario, según su requerimiento se analiza a continuación.

a. Controladores del sistema

En el diseño del sistema se consideró la accesibilidad, rendimiento y operabilidad de los dispositivos que circulan en el mercado de la electrónica actualmente, tomando en cuenta sus GPIO, corriente de salidas, tecnologías de comunicación y tensión de alimentación, en la **Tabla 5** se detalla sus características.

Tabla 5: Comparativa técnico de miniordenadores

Parámetros técnicos	Arduino Mega 	Orange Pi 	Raspberry Pi 4 
GPIO	54	26	40
Corriente GPIO	40-50 mA	16-50 mA	16-50 mA
Tensión Alimentación	5 V	5 V-micro-USB	5 V-USB tipo C
Memoria RAM	8 KB	256 / 512 MB	1-2-4 GB
Precio	25 \$	30 \$	90-110-125 \$



Elaborado por: el investigador

Al culminar el análisis entre las posibles opciones para la ejecución del sistema con un rendimiento adecuado, pese a la similitud encontrada en microcontroladores a nivel de corrientes y voltajes, se descarta el Arduino y Orange Pi, debido a su bajo configuración en Memoria RAM, por necesidad del sistema a utilizar interfaz gráfica y manejos de datos ininterrumpida, por la cual se optó la Raspberry Pi 4, sus características más detalladas se encuentran en el **Anexo 1**.

b. Sensores

Los sensores en el sistema se encargan de: nivel del contenedor, Comprobar ingreso del envase PET, comprobación de cantidad de productos en el dispensador, medidas de protección del usuario en el ingreso del envase y temperatura en el interior del contenedor en la sección productos. Los sensores existentes en el mercado son analizados **Tabla 6, Tabla 7 y Tabla 8**.



Tabla 6: Comparativa técnico sensores Ultrasónicos

Parámetros técnicos	HC-SR04 	XL-Maxsonar EZ4 	JSN Sr04t 
Distancia de medición	2 – 400 cm	0 – 765 cm	20 – 600 cm
Interfaz	2 hilos digitales	Digital y análoga	Digital
Frecuencia en pulso	40 Khz	42 Khz	40 Khz
Señal de disparo	10 us	100 ms	10 us
Tensión de alimentación	5 V	3.3 – 5 V	3.3 – 5 V
Precio	2 \$	60 \$	20 \$

Elaborado por: El investigador

Al concluir el análisis se seleccionó el sensor HC-SR04 debido a su accesibilidad, rango óptimo para el sistema y compatibilidad con diversos microcontroladores, esto llevó al descarte de XL-Maxsonar EZ4, por su alto costo y señales de disparos más retardados, las características del HC-SR04 se encuentran más detalladas en el **Anexo 2**.



Tabla 7: Comparativa técnico elemento electrónico para detección de objeto

Parámetros técnicos	Final de Carrera 	Sensores IR 
Tensión de alimentación	5A, 125/250 VAC	3.3 a 5V
Operabilidad en transparencia	Efectiva	Menos efectiva
Precio	1 \$	2.5 \$

Elaborado por: El investigador

Se toma en cuenta que los envases PET la mayoría son transparente y que un haz de luz siempre atravesara por él y no habría respuesta del sensor IR, debido a esto se descarta este sensor, y se toma en cuenta el final de carrera como la opción más viable para el sistema, con la ayuda de un servo motor que compacta el envase, también se utiliza en la comprobación de dispensar incentivos. Sus características más detalladas se encuentran en el **Anexo 3**.

Tabla 8: Comparativa técnico sensores de temperatura

Parámetros técnicos	LM-35 	DHT-22 	PT-100 
Tensión	4 – 30 V	3.3 – 6 V	3.3 – 5 V
Rango-Temperatura	-55 °C a 150 °C	-40 °C a 80 °C	0 a 150 °C
Exactitud	± 0.5 °C	± 0.5 °C	± 0.01 °C
Precio	3 \$	7 \$	20 \$

Elaborado por: El investigador

El sistema al contener alimento de consumo humano, necesita tener una función de refrigeración que se encienda automáticamente, se toma en cuenta la temperatura ambiente interior en la sección de dispensar para efectuar esta acción.

De acuerdo a los datos climáticos de Ecuador, en la región Costa la temperatura promedio esta entre 22-25 °C, la región Sierra varía entre 8-20 °C, la región oriental tiene un promedio entre 24-25 °C y en la región Insular la temperatura promedio es de 23 °C. [40]

Con estos datos se optó por el DHT-22 como opción viable debido a su operación en comparación al LM35 y PT-100. Sus características más detalladas se encuentran en el **Anexo 4**.

c. Actuadores

El sistema OPEN BOT para dar acceso al usuario necesita disponer de mecanismos como:

- Apertura de acceso PET que gire una puerta entre 0°-90° con una señal eléctrica
- Compresión de PET mediante un movimiento angular de 0°-90° con una señal eléctrica con un torque relativamente alta.
- Apertura de almacenamiento PET que gire una compuerta entre 0°-90° con una señal eléctrica
- Empuje de PET hace los extremos vacíos del contenedor mediante un movimiento angular de 90°
- Dispensación de incentivos mediante una revolución exacta de acuerdo una señal de pulsos.

Para el análisis de los actuadores que se requieren en el sistema se analizó la función requerida, forma de operación y funcionamiento, el más cercano de acuerdo a los requerimientos del Sistema, es un Servo Motor y un Motor a Pasos, en la **Tabla 9** y **Tabla 10**, se realiza una comparativa técnica para su evaluación y elección que se encuentran en el mercado actualmente.



Tabla 9: Comparativa técnico servo motor

Parámetros técnicos	SG-90 	MG-995 	DS3225-MG 
Tensión de alimentación	4.8 - 5 V	3 - 7.2V	4.8 - 6.8V
Torque	1.8 kg / cm	15 kg / cm	25kg / cm
Impermeabilidad	NO	NO	SI
Velocidad de operación	0.1 s/60°	0.17 s/60°	0.13 s/60°
Precio	3 \$	11 \$	25 \$

Elaborado por: El investigador

El servo-motor seleccionado es DS3225-MG, por el par de torque, para la ejecución de compactar los envases PET se necesita una torción alta con respecto a la resistencia del envase, se descartan los servomotores SG-90 y MG-995, las características del DS3225-MG más detalladas se encuentran en el **Anexo 5**.

Tabla 10: Comparativa técnico motor a pasos

Parámetros Técnicos	Nema 17 	Nema 23 
Tensión de alimentación	12 V	12-48 V
Torque	3.2 kg/cm	19Kg/cm
Pasos por vuelta	200 pasos	200 pasos
Corriente máxima	1.7A	2.8A
Precio	15 \$	43 \$

Elaborado por: El investigador

El sistema OPEN BOT para poder expender incentivos al usuario necesita girar un numero de revoluciones exactos mediante un número de pulsos, por estas prestaciones requeridas del sistema un motor a pasos de bajo costo es óptimo para el sistema y se eligió el motor nema 17 por su bajo precio. Sus características más detalladas se encuentran en el **Anexo 6**.

d. Scanner

Para el ingreso de envases PET adecuados el sistema posee una función de escáner de código de barra, dando acceso así al contenedor, en la **Tabla 11** se presentan las opciones para obtener dicho dato y el más óptimo para el sistema.

Tabla 11: Comparativa técnico módulos de detención de códigos de barra

Parámetros técnicos	Scanner de código de barra 	Web Cam 
Tensión de alimentación	5 V	5 V
Interfaz	USB	USB
Compatibilidad Linux	Si	Si
Modo operación	disparo	Llamado del sistema
Precio	50 \$	10 \$


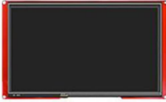

Elaborado por: El investigador

El sistema utiliza una Web Cam debido a que los lectores de códigos de barras tienen un costo elevado, y necesitan disparo para su ejecución.

e. Dispositivo de Salida

El sistema al tener la función de interactuar con el usuario, necesita un medio de comunicación con él para dar acceso a las funciones que el sistema posee, para ello en la **Tabla 12** se analiza los diferentes dispositivos que se adaptan al sistema y circulan en el mercado actual.

Tabla 12: Comparativa técnico dispositivo de salida

Parámetros técnicos	Pantalla Raspberry Pi 7” 	Pantalla Nextion 10.1” 	Pantalla HDMI LCD (B) 10.1” 
Soporte Linux	Si	Si	Si

Tensión Alimentación	5 V	5 V	5 V
Multi Touch	Si	Si	Si
Interfaz Sistema	Puerto Pi	No soportable	HDMI
Comunicación Touch	Puerto Pi	RS232	USB
Precio	98 \$	195 \$	180 \$

Elaborado por: El investigador

Para el sistema se utiliza la Pantalla HDMI LCD (B) 10.1'' por su compatibilidad y modo de operación en touch, sus características se detallan en el **Anexo 7**.

3.2.5 Sistema de reciclado de envases PET.

a. Medición y adquisición de datos (Sensorización)

El sistema para su proceso adquiere datos de los sensores:

Ultrasónico, verificando el nivel actual del contenedor para no exceder con su capacidad de 350 envases, esto se realiza cada vez que el usuario desee ingresar un envase PET.

Sensor de seguridad, verificando en cada interacción del usuario, si no obstruye el ingreso, para proceder a cerrarla y seguir con el proceso de almacenamiento.

Temperatura, verificando contantemente el ambiente interno a una precisión de 0,5 °C

b. Sistema de gestión y visualización

El sistema está encargado de administrar, gestionar y controlar los datos de usuarios por los diferentes modos de visualización:

Modo Usuario. – Si el usuario registrado ejecuta esta acción, el sistema suma puntos a su cuenta, para posterior ser usado en reclamos de regalos.

Modo Invitado. – Si el usuario que desea reciclar y no requiere una cuenta, el sistema posee una opción de reciclar en modo de no registro, pero no acredita puntos a su cuenta.

Modo Administrador. – Para poder reciclar el usuario necesita tener acceso al contenedor, el sistema al operar con una base de datos permite con este modo registrar nuevos envases en él, para posteriormente los usuarios puedan reciclar sus envases.

Modo New usuario. – Si desea reciclar y no tiene cuenta el sistema permite crearla, mediante el registro de un usuario y contraseña, además se requiere de tres preguntas básicas de seguridad, en caso de olvido de contraseña y poder recuperarlo. El sistema verifica si el usuario a registrar no es existente, caso contrario aborta la ejecución mediante información de “usuario existente”.

Modo cambio clave. – Los usuarios que olviden su contraseña de acceso, tienen la opción de cambiarla, a esto se necesita las tres preguntas de seguridad registradas, caso contrario de aborta la ejecución mediante la información de “datos incorrectos”

c. Sistema de control mecánico

El sistema de reciclaje “OPEN BOT” para almacenar los envases tiene un control mecánico que interactúa en los modos: invitado y usuario. El proceso ejecuta las siguientes acciones según su funcionalidad:

Leer código. – El sistema únicamente acepta envases PET de bebidas que circulan en el mercado y son registrados por el Administrador, lo cual se verifica en una base de datos mediante la lectura de código de barras, con la ayuda de una cámara y procesamiento gráfico en Python, si este código leído no está registrado se aborta la ejecución mediante una información de “envase no PET”

Acceso de envases al contenedor. – Antes que el usuario ingrese un envase nuevo al contenedor y después de cumplir si el envase se encuentra registrado, el sistema verifica que el contenedor este el nivel entre vacío-llevo para continuar con el proceso,

de lo contrario informa un mensaje de “contenedor lleno”, lo cual no acepta más envases para ser almacenados.

Compactación de envases. – El sistema para aprovechar al máximo su capacidad, reduce el volumen de los envases PET mediante la compresión, con la ayuda de un servo motor de 25 Kg en fuerza, ejecutar esta acción verificando si el usuario ha ingresado o no un envase, de lo contrario informa “envase no ingresada”, a su vez no acredita los puntos al reciclador.

Acceso al contenedor. – Después de cumplir las etapas de acceso y compactación, el sistema almacena el envase en su interior mediante una compuerta que da paso al contenedor, que se encuentra cerrada para el proceso de compactación.

Nivelar contenedor. – Para su almacenamiento, el sistema después de ingresar un envase nivela el contenedor enviando los envases hacia el extremo vacío con la ayuda de un servo motor.

d. Sistema de dispensación.

El sistema contiene regalos al reciclador que opte por esta manera de contribuir en el manejo de desechos sólidos, con la opción de dispensar incentivo. Open Bot ofrece a los usuarios que lleven un crédito mayor o igual a los 0.50, permitiéndose reclamar un obsequio como recompensa al reciclaje, además verifica si el usuario no dispone de crédito suficiente e informa mediante “créditos insuficientes” abortando la ejecución.

e. Desarrollo de interfaz de usuario

Para el desarrollo de la interfaz, se utiliza Python con la librería Tkinter, al ser compatible con la mayoría de sistemas operativos y ser un lenguaje de programación interpretado. El sistema “OPEN BOT” consta con una interfaz de usuario como App de escritorio y contiene diversos Frame’s de control. Su esquema se representa en la **Figura 18** y su programación se describe a continuación y se detalla en el **Anexo 9**.

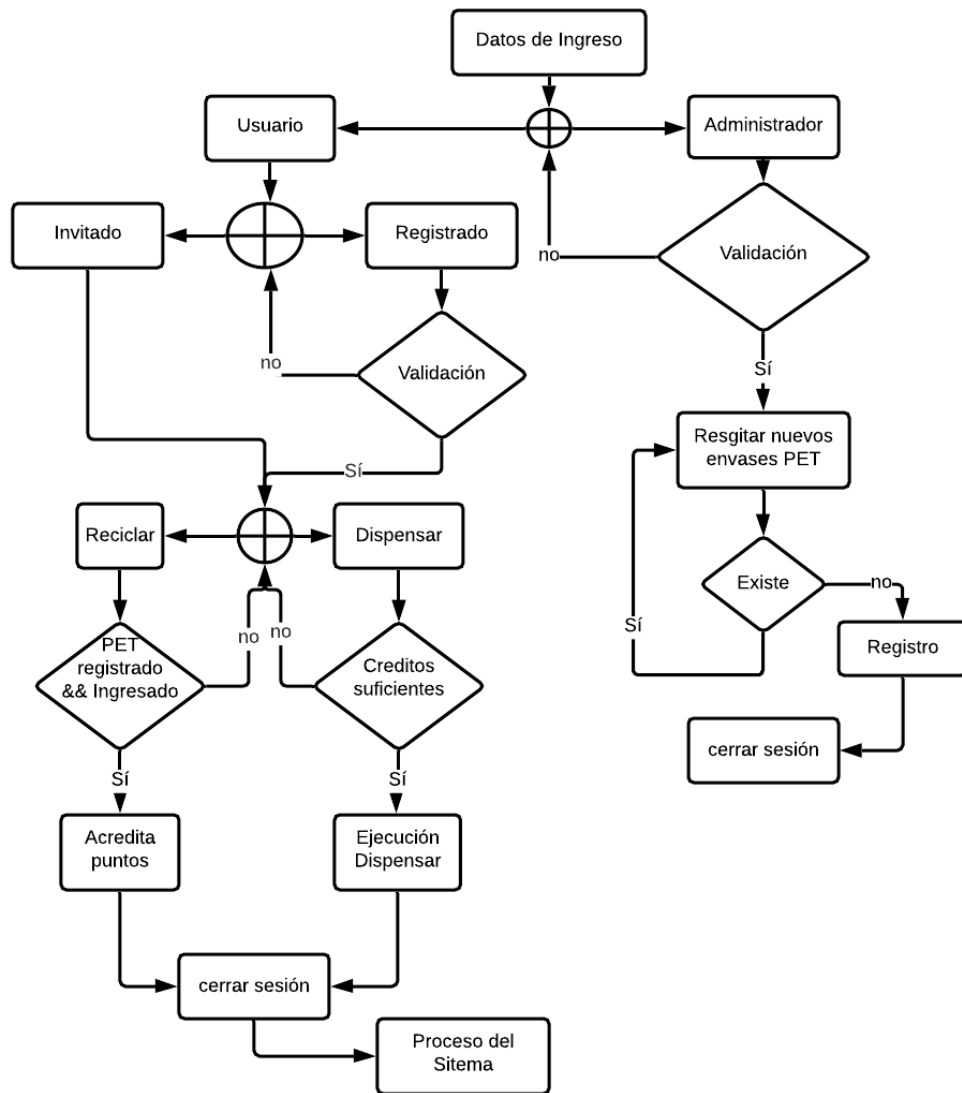


Figura 18: Diagrama de acceso al sistema

Elaborado por: El investigador

La **Figura 19** representa el diagrama de bloques del sistema “OPEN BOT”, la cual determina cada uno de las fases a desarrollar y su interacción entre sí.

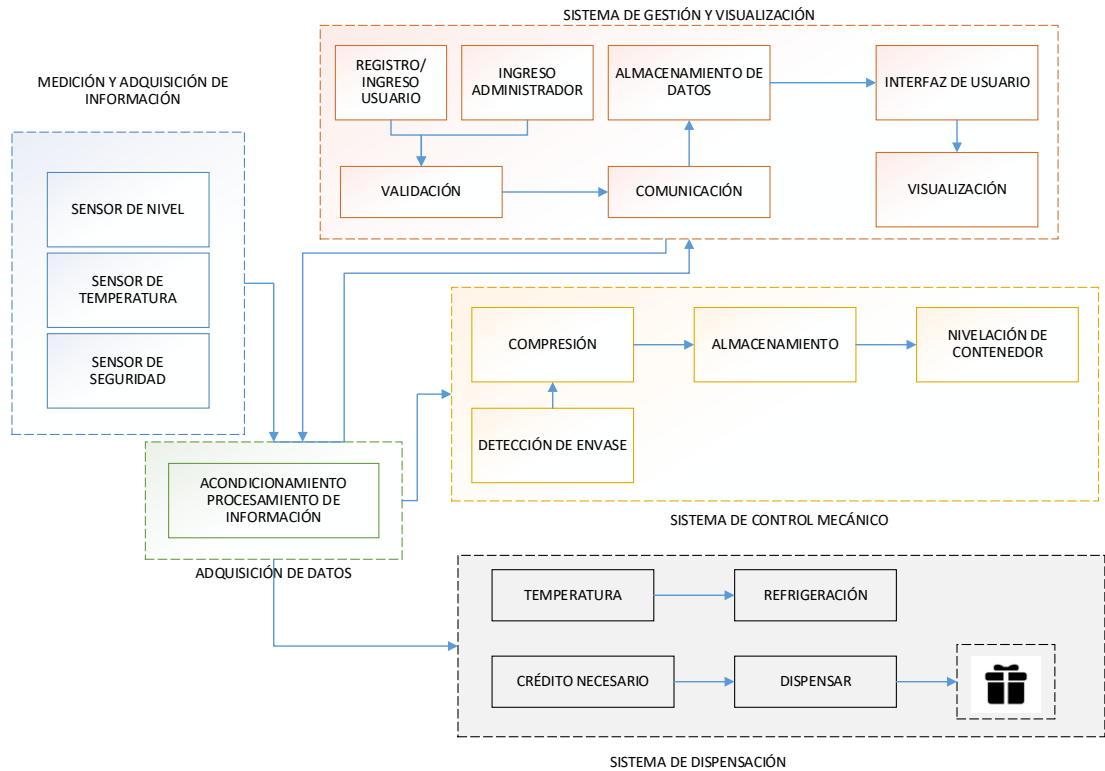


Figura 19: Diagrama esquemático del sistema

Elaborado por: El investigador

Ventana Raíz: Como primera instancia se establece un contenedor principal que genera como arranque de la aplicación para el Sistema “OPEN BOT”

Frame Teclado: Esta salida visual tienen la funcionalidad de ejercer como un teclado virtual para la adquisición de datos hacia los usuarios al momento de iniciar sesión, crear nuevo usuario, recuperar clave e inicio modo administrador. Su tecla presionada se almacena en una variable para posterior ser utilizada en el modo que se encuentre. Su parte visual se observa en a **Figura 20**.



Figura 20: Frame teclado del sistema

Elaborado por: El investigador

Frame Principal: Esta salida visual tiene la funcionalidad de obtener datos de usuario registrados y administrador para el inicio de sesión. Si los datos son nulos y selecciona la opción iniciar, accede a la cuenta invitado. Si los datos son nulos y selecciona la opción administrador, se informa mediante “datos incorrectos”. Para su ejecución comprueba con una base de datos en cada ejecución. Su parte visual se observa en la **Figura 21.**

A login form titled "Iniciar Sesión" is shown. It has a dark blue header with the title in white. Below the header, there are two input fields: the first is labeled "Usuario:" with a person icon, and the second is labeled "Clave:" with a key icon. Below the input fields, there are four buttons: "Ingresar" and "Registrar" are side-by-side; "Recuperar Clave" is centered below them; and "Iniciar Administrador" is centered below that.

Figura 21: Frame Principal

Elaborado por: El investigador

Frame Usuario: En esta salida visual se observa los créditos acumulados y el modo que se encuentra, tiene la funcionalidad de consultar datos del usuario del servidor local, mediante la comprobación de clave, además tiene las opciones de reciclar envase PET y expandir incentivo. Para ejercer las opciones reciclar verifica con la ayuda de sensores el nivel del contenedor, si el usuario ha ingresado un envase para posterior acredita y permite reciclar. Para ejercer la opción de Dispensar verifica si el usuario posee los créditos suficientes y acciona la dispensación. Su parte visual se observa en la **Figura 22**.



Figura 22: Frame usuario

Elaborado por: El investigador

Frame Administrador: Esta salida visual tiene la funcionalidad añadir nuevos códigos de envases al servidor local verificando en su base de datos si no se encuentra actualmente en lista, para que el usuario pueda hacer uso del sistema. Su parte visual se observa en la **Figura 23**.



Figura 23: Frame administrador

Elaborado por: El investigador

Frame New Usuario: Esta salida visual cumple con la función de añadir un nuevo usuario al sistema para su uso, en esta ventana realiza el control si el usuario está registrado, validación de clave y preguntas básicas de seguridad que son: mascota, color y fecha de nacimiento, para validar su registro. Su parte visual se observa en la **Figura 24** y **Figura 25**.

Figura 24: Frame nuevo usuario

Elaborado por: El investigador



Figura 25: Preguntas se seguridad

Elaborado por: El investigador

Frame Recuperar Clave: Esta salida visual tiene la funcionalidad de cambiar la clave si el usuario lo olvido, en esta ventana se comprueba si el usuario está registrado, verifica las preguntas de seguridad de la **Figura 25** para un cambio éxitos de su clave. Su parte visual se observa en la **Figura 26**.

The screenshot shows a form titled 'Recuperar Clave'. It has an orange header. Below the header are two input fields for 'Usuario*' and 'Nueva Clave*'. To the right of these fields is a robot icon labeled 'OpenBot'. Below the input fields is a section titled 'Preguntas de Seguridad' with a 'Fecha de Nacimiento*' field containing '0000-00-00'. At the bottom, there are two dropdown menus for 'Mascota*' and 'Color*', and two buttons: 'Aceptar' and 'Salir'.

Figura 26: Frame recuperar clave

Elaborado por: El investigador

Frame Datos de Sistema: Esta salida visual tiene la funcionalidad obtener la temperatura, numero de botellas registradas, hora y fecha que tienen el sistema, son de utilidad para el usuario y administrador. Su parte visual se observa en la **Figura 27**.

Figura 27: Datos del sistema

Elaborado por: El investigador

d. Base de Datos

El sistema para la administración de datos de los usuarios tiene un servidor de base de datos local desarrollada en un servidor MariaDB y para un mejor manejo y visualización también se optó por configurar un servidor Apache y PHP bajo el sistema operativo Raspbian.

El las **Figuras (28 al 31)** se observa la base de datos y Tablas creadas.

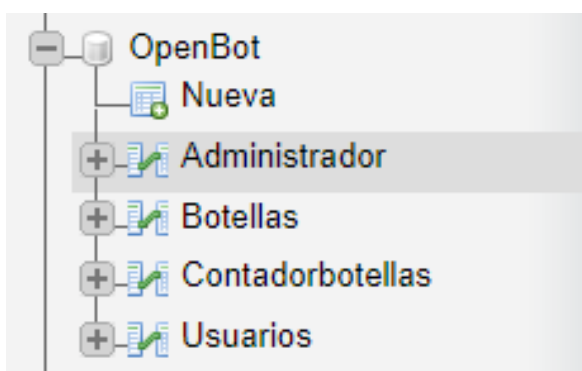


Figura 28: Base de datos del sistema

Elaborado por: El investigador

+ Opciones			
	ID	administrador	clave
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	1805156724	1032

Figura 29: Tabla administrador

Elaborado por: El investigador

+ Opciones		ID	codigo
<input type="checkbox"/>	←T→ Editar Copiar Borrar	1	0012796
<input type="checkbox"/>	←T→ Editar Copiar Borrar	2	0759494000354
<input type="checkbox"/>	←T→ Editar Copiar Borrar	3	6178794000354

Figura 30: Tabla de envases vinculados al sistema

Elaborado por: El investigador

+ Opciones		ID	userid	clave	pregunta1	pregunta2	fecha	credito
<input type="checkbox"/>	←T→ Editar Copiar Borrar	1	1234567890	1234	perro	blanco	0000-00-00	0.4
<input type="checkbox"/>	←T→ Editar Copiar Borrar	2	1805156724	1032	perro	verde	1992-09-26	0.5
<input type="checkbox"/>	←T→ Editar Copiar Borrar	3	1800000000	1234	perro	blanco	2020-08-06	0

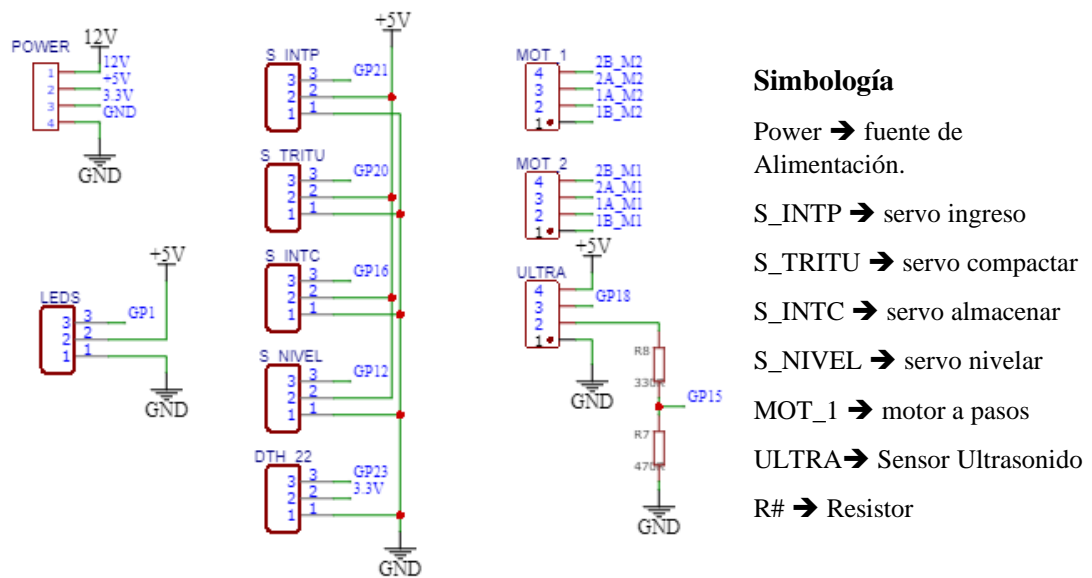
Figura 31: Tabla de usuarios registrados

Elaborado por: El investigador

3.2.6 Placa de Control

Para el control de la parte electromecánica del sistema se ha desarrollado una placa electrónica ver **Figura 37**.

El diseño esquemático del sistema se describe a continuación, para su entendimiento de describen por partes y se usa conectores NET, se aprecia en la **Figura 32** la fase de alimentación, las conexiones de los motores a pasos y conexiones de servo motores.



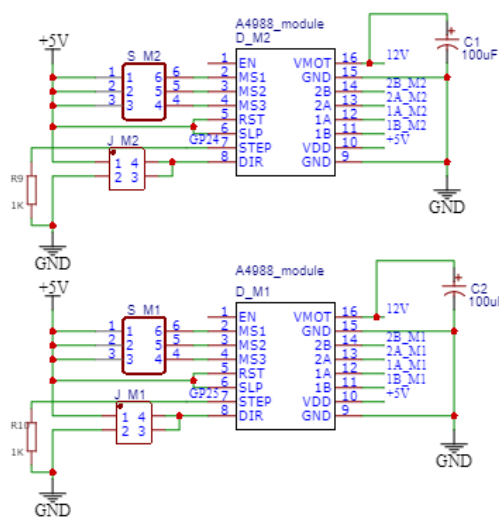
Simbología

- Power → fuente de Alimentación.
- S_INTP → servo ingreso
- S_TRITU → servo compactar
- S_INTC → servo almacenar
- S_NIVEL → servo nivelar
- MOT_1 → motor a pasos
- ULTRA → Sensor Ultrasonido
- R# → Resistor

Figura 32: Fase de motores

Elaborado por: El investigador

En la **Figura 33** se observa las conexiones de los drives de motores a pasos



Simbología

- C# → capacitor.
- D_M# → driver A4988 motor a pasos
- R# → resistor
- S_M# → selector steep
- J_M# → Dirección Motor
- MOT_1 → motor a pasos

Figura 33: Conexión de drives motores a pasos

Elaborado por: El investigador

En la **Figura 34** se observa las conexiones de acondicionamiento de las señales provenientes de los sensores

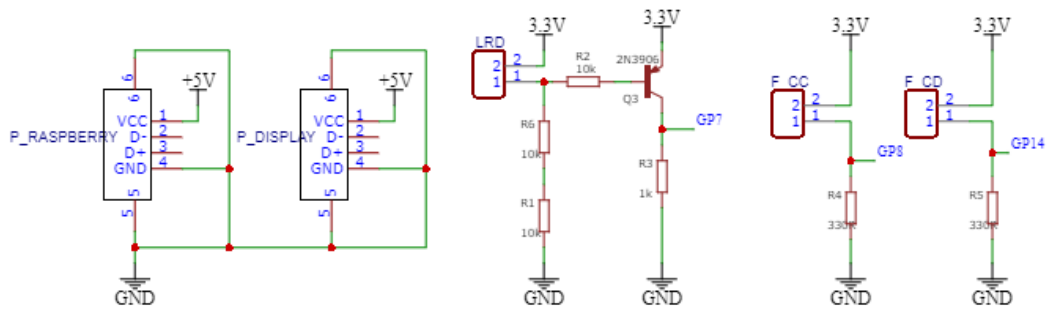


Figura 34: Fase acondicionamiento de la señal

Elaborado por: El investigador

En la **Figura 35**, se aprecia las conexiones de arranque seguro y ventilación en el área de dispensación del sistema.

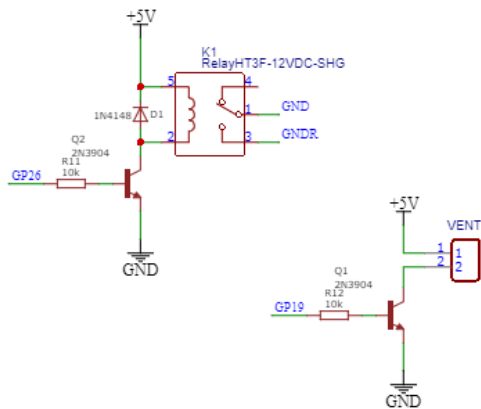


Figura 35: Conexión de arranque seguro del sistema

Elaborado por: El investigador

Su representación completa se encuentra en la **Figura 36**. Diseñado en EasyEDA

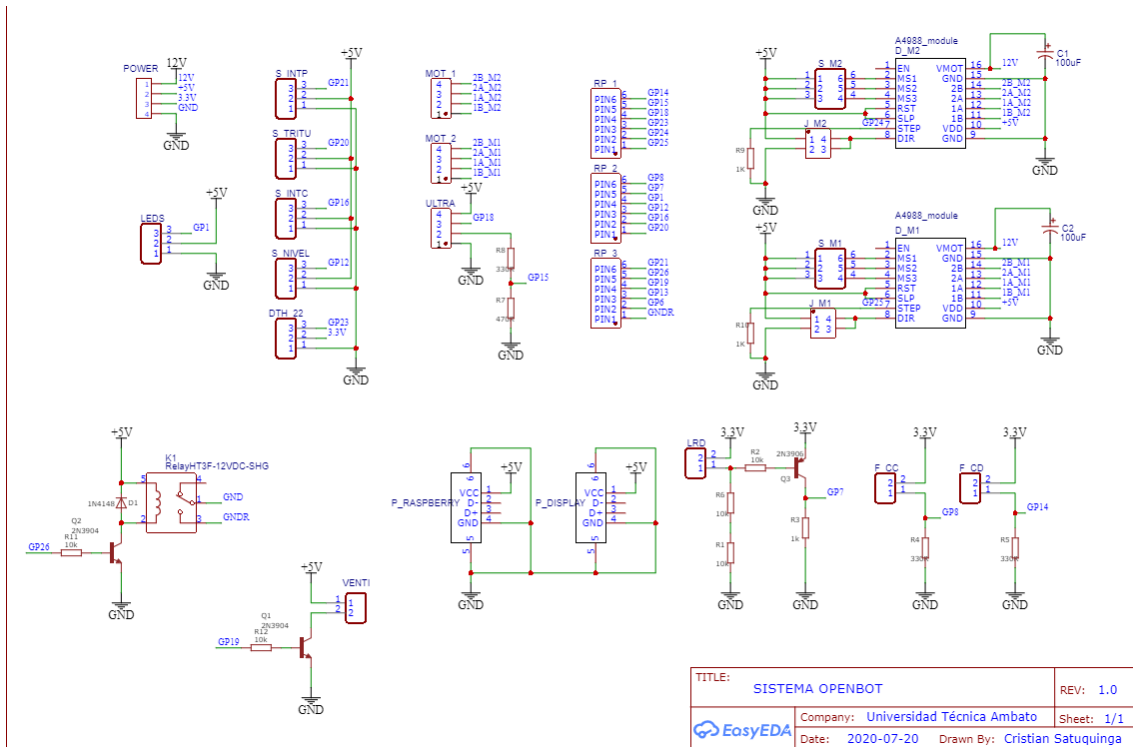


Figura 36: Representación del esquema electrónico

Elaborado por: El investigador

La **Figura 37** ilustra su configuración en PCB de vista frontal y posterior diseñado en el programa EasyEDA.

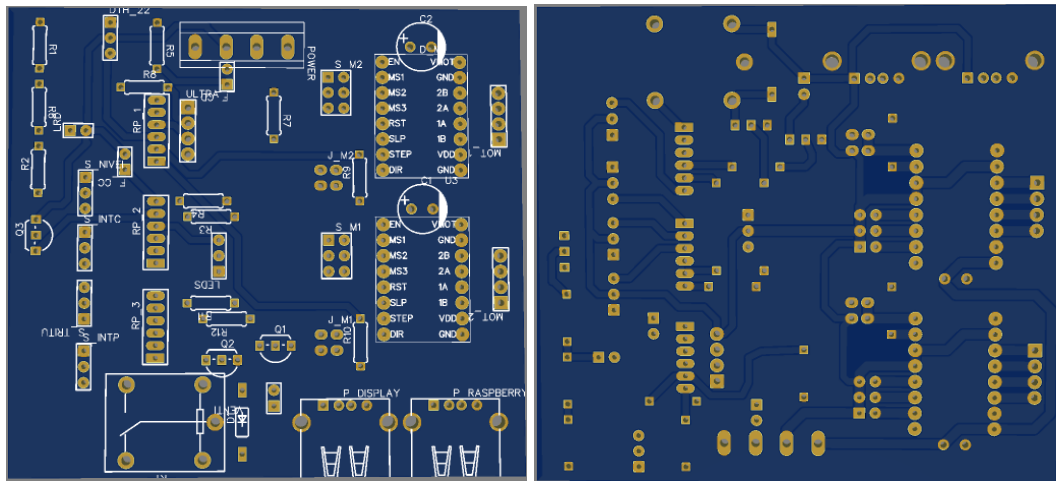


Figura 37: PCB del sistema

Elaborado por: El investigador

La **Figura 38** ilustra su configuración en PCB de vista en 3D diseñado en el programa EasyEDA.

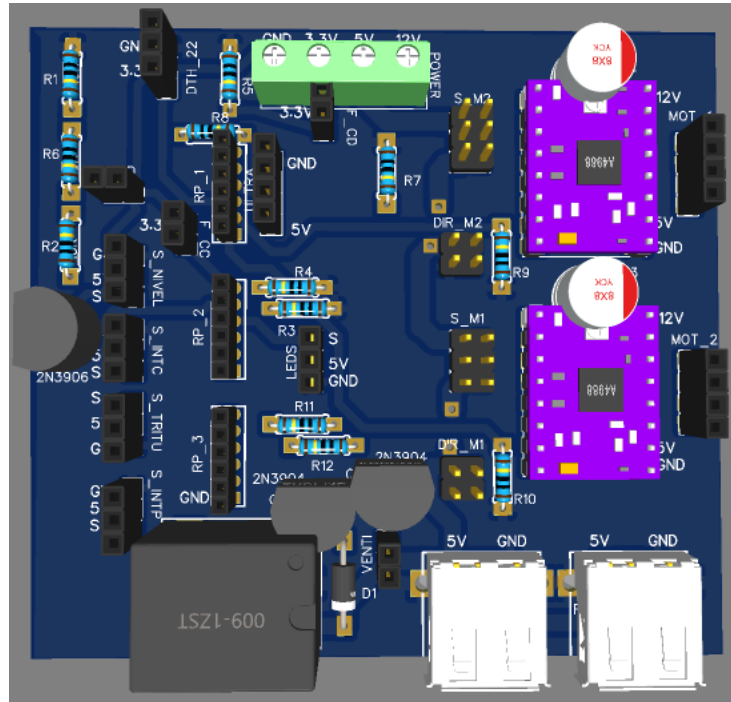


Figura 38: PCB del sistema vista 3D

Elaborado por: El investigador

3.2.6 Construcción del contenedor “OPEN BOT”

Por la configuración del sistema, el contenedor está diseñado para abarcar hasta 350 envases, las más comunes son: envases entre 0,5 litros a 1.2 litros. La **Figura 39** ilustra su composición en perspectiva que contiene una pantalla HDMI táctil, un ingreso PET, un orificio para las tapas de los envases, un área de dispensación y un lector de código de barras. Además, se observa donde ira situado el sensor de protección para el usuario.

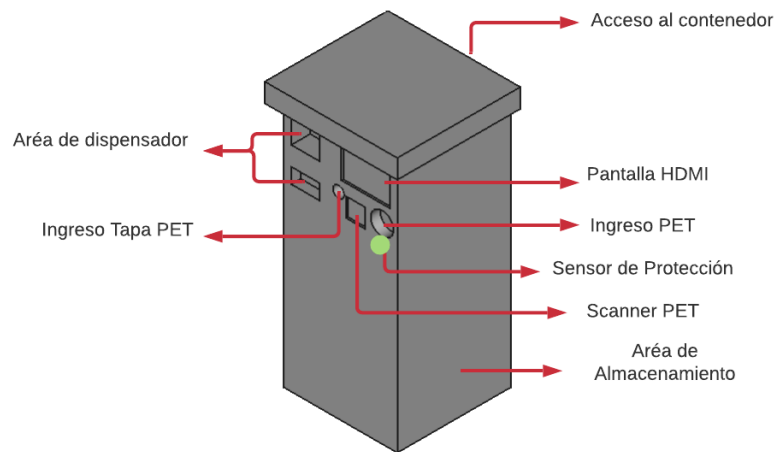


Figura 39: Vista isométrica frontal del sistema

Elaborado por: El investigador

La **Figuras 40 y 41** ilustran su composición del reverso, donde contiene un área de dispensación, un área de electrónico, un área de compresión y un área de almacenamiento, que se accede mediante una puerta para su mantenimiento o revisión del sistema.

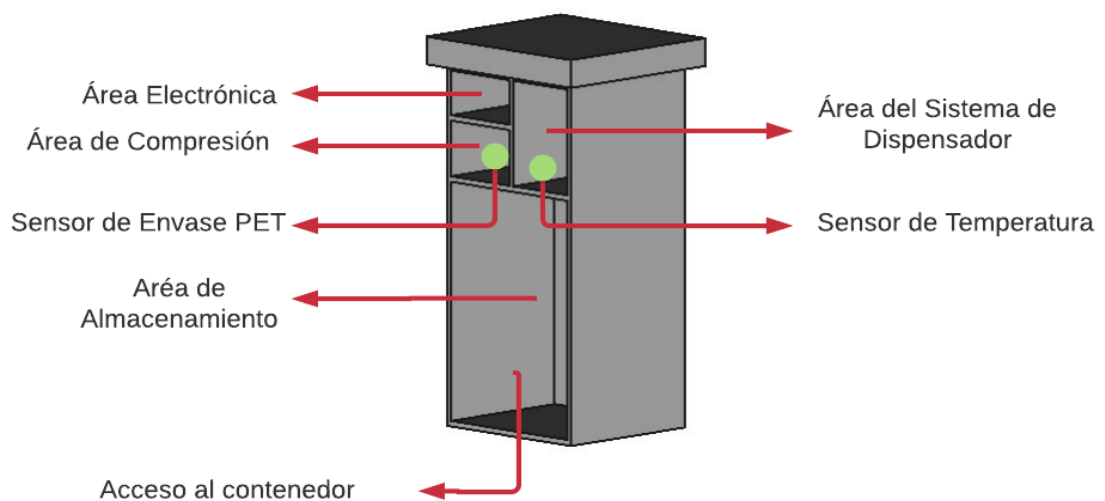


Figura 40: Vista isométrica posterior del sistema

Elaborado por: El investigador

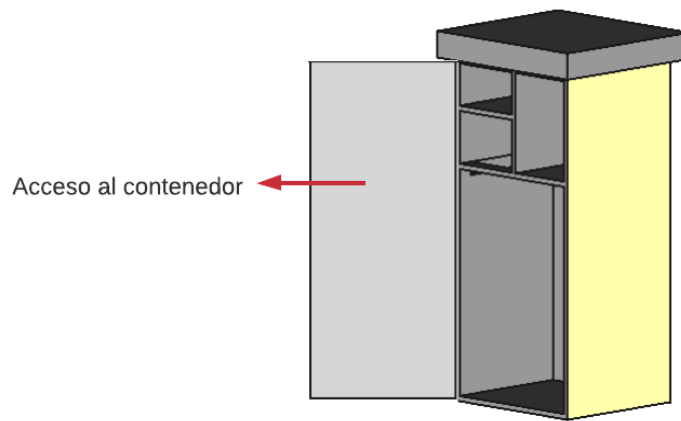


Figura 41: Vista isométrica con puerta de acceso

Elaborado por: El investigador

a. Esquema interno del área de compresión.

En la **Figura 41** se ilustra la composición interna del área de compresión donde está ubicado el sensor del envase PET junto a los actuadores que realizan la acción de abrir acceso, comprimir envase y almacenar en el contenedor.

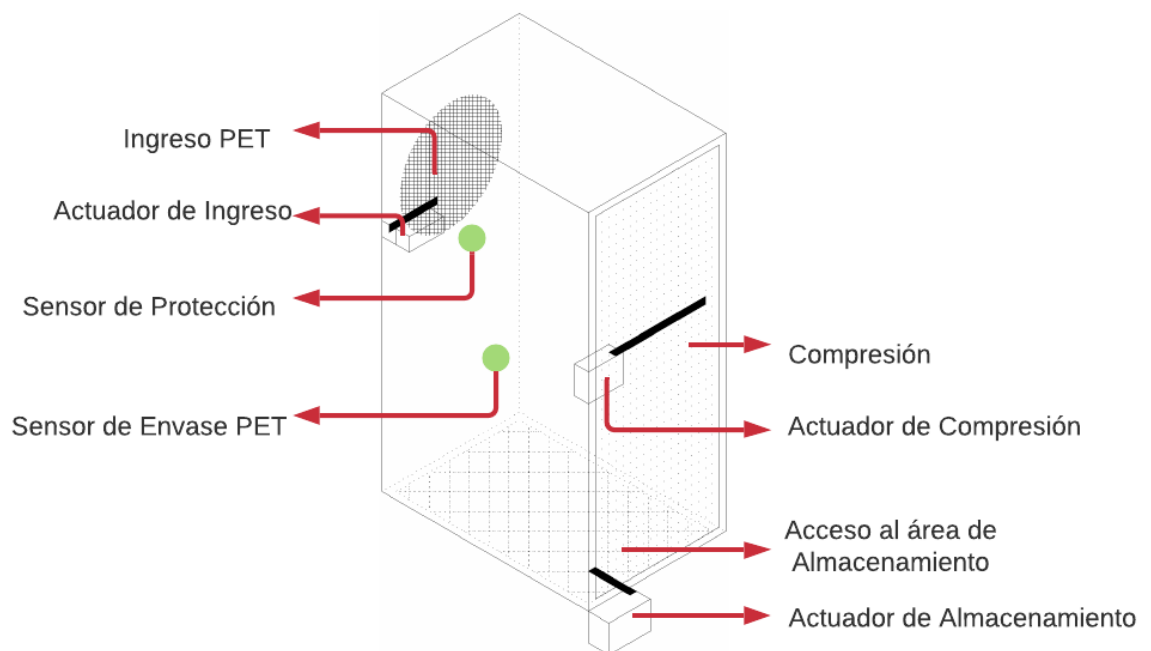


Figura 42: Esquema mecánico de compresión

Elaborado por: El investigador

b. Gráficos de conexiones

En la **Figura 43** se ilustra la configuración esquemática de conexión dinámica realizado en el programa Fritzing, donde se encuentran los sensores, actuadores y módulos reales en representación gráfica.

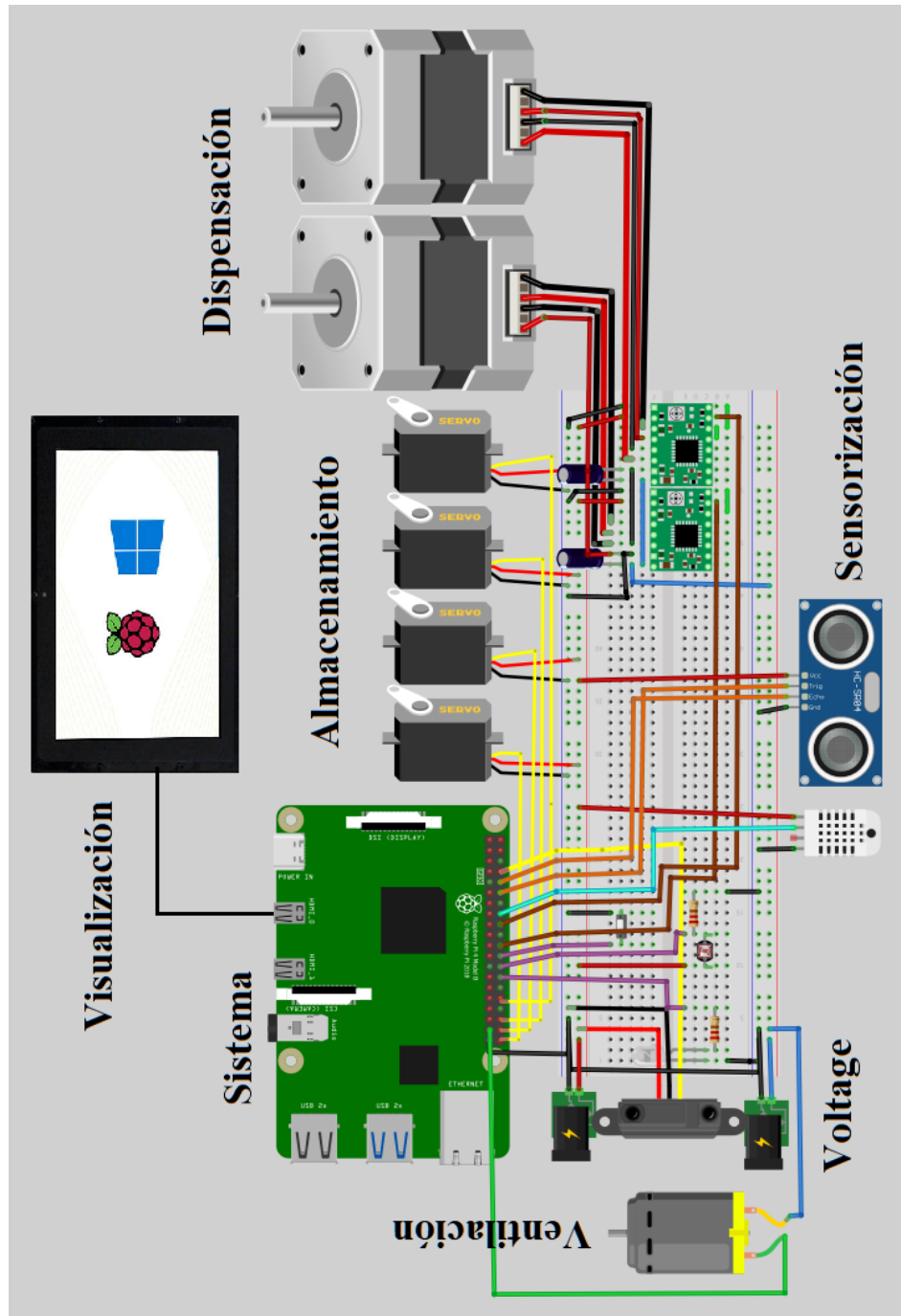


Figura 43: Conexión del sistema

Elaborado por: El investigador

Gráficos finales

En las **Figuras 44 y 45** se ilustran las imágenes del producto ensamblado en su totalidad con sus respectivas partes y ubicaciones de los dispositivos.



Figura 44: Vista frontal

Elaborado por: El investigador



Figura 45: Vista en perspectiva

Elaborado por: El investigador

3.2.9 Pruebas del funcionamiento

Los envases PET, de bebidas gaseosas, aguas minerales y energizantes, son correctamente captadas por el sistema, las cuales da un aporte al medio ambiente y a la sociedad al ser reciclado por este tipo de sistemas de gestión.

El sistema al poseer una base de datos para los envases registrados, lo cual comprueba cada que el usuario desea reciclar, desecha los que no están registrado abortando el proceso, en la **Figura 46** se observa el mensaje si un envase no se encuentra registrado.



Figura 46: Mensaje de sistema de envase no registrado

Elaborado por: El investigador

En la **Tabla 13** se observa de los envases más consumido a nivel de ecuador, con su respectivo volumen y precio, las cuales son aceptados en el sistema de reciclaje “OPEN BOT”

Tabla 13: Lista de envases aptos para el sistema

Producto	Volumen	Precio	Recompensa
Agua mineral Tesalia	625 ml	0,40 \$	0.02 \$
Agua mineral Cielo	725 ml	0,50 \$	0.02 \$
Agua mineral Dasani	600 ml	0,50 \$	0.02 \$
Bebida gaseosa Coca Cola	300 ml	0,35 \$	0.02 \$
Bebida gaseosa Fioravanti	300 ml	0,35 \$	0.02 \$
Bebida gaseosa Sprite	300 ml	0.35 \$	0.02 \$
Bebida gaseosa 7 Up	450 ml	0,40 \$	0.02 \$
Bebida gaseosa Big Cola	300 ml	0.20 S	0.02 \$
Bebida gaseosa Güitig	500 ml	0,60 S	0.02 \$
Bebida energizante 220V	600 ml	1.10 \$	0.02 \$

Elaborado por: El investigador

Pese a esta lista descrita con anterioridad no son las únicas que sistema admite para ser reciclados. El sistema al tener le modo administrador permite registrar un sin número de envases tipo PET, de acuerdo a que las entidades administradoras lo requieran y configuren.

Para verificar el volumen de compresión se ocupó una forma experimental con la relación peso-volumen del agua al calcular con su densidad.

Densidad del agua 1kg/litro

Para conocer el nivel de compresión se utilizó la relación del volumen en los envases PET, antes y después de la compresión utilizando la relación de densidad de agua.

En este experimento se mide el peso del agua antes y después de la compresión llenando el envase a su totalidad sin modificar su estructura.

Tabla 14: Relación PET sin compresión y comprimido por el sistema OPEN BOT

Relación de compresión PET por el sistema OPEN BOT				
Volumen envase PET	Peso antes de compresión	Peso después de Compresión	Porcentaje de compresión	Porcentaje de Error
1,20 litros	1,20 kg	0,54 kg	55 %	5 %
1,10 litros	1,10 kg	0,60 kg	45 %	5 %
1,00 litros	1,00 kg	0,51 kg	49 %	1 %
0,80 litros	0,08 kg	0,40 kg	50 %	0 %
0,75 litros	0,75 kg	0,37 kg	51 %	1 %
0,60 litros	0,60 kg	0,31 kg	48 %	2 %
0,50 litros	0,50 kg	0,23 kg	54 %	4 %
0,40 litros	0,40 kg	0,19 kg	53 %	3 %
0,35 litros	0,35 kg	0,17 kg	51 %	1 %
0,30 litros	0,30 kg	0,16 kg	47 %	3 %

Elaborado por: El investigador

Dando de promedio de compresión al 50%

Del cual el margen de error se obtuvo relacionando el porcentaje de compresión con el promedio de compresión.

En las pruebas realizadas se logra apreciar que la máquina almacena de forma correcta, comprime y almacena para su gestión para ser trasladados al lugar de tratamiento. Los envases registrados en el sistema hasta el momento, fomenta la cultura de reciclaje y ayuda a los usuarios a obtener beneficios por hacerlo. La **Figura 47** muestra los envases reciclados por el sistema.



Figura 47: Envases dentro del sistema

Elaborado por: El investigador

Como una de las opciones del sistema es la dispensación de incentivos, en la **Figura 48** se observa el resultado, si el usuario tiene los créditos necesarios para su reclamo.



Figura 48: Incentivo expendido por el sistema

Elaborado por: el investigador

3.2.7 Factibilidad Económica

El proyecto va dirigido hacia las empresas que fomentan el reciclaje para la reducción de contaminantes en el medio ambiente, por la cual la construcción del sistema inteligente “OPEN BOT”, este compuesto por elementos electrónicos adaptables y de bajo costo, su estructura está construido con materiales de fácil asequibilidad y estos se detallan en la **Tabla 15**.

Tabla 15: Presupuesto de construcción del sistema

Tablas de costos del sistema “OPEN BOT”			
Material	Tipo	Cantidad	Valor Total
Raspberry Pi	Modelo 4	1	110 \$
Acero inoxidable	Lamina 2mm*240cm*120cm	3	100 \$
Acero inoxidable	Tubo ½ pulgada	3	40 \$
Acero inoxidable	Lamina 2mm * 6m	3	20 \$
Motor a pasos	Nema 17	2	30 \$
Servo motor	DS3225 mg 25 Kg	4	100 \$
Driver motor	A4988	2	8 \$
Relevador	5 V	1	1 \$
Sensor ultrasónico	HC-SR04	1	2 \$
Cámara web	USB	1	10 \$
Fuente	ATX	1	20 \$
Pantalla HDMI	10.1 pulgadas	1	280 \$
Placa	Diseño el sistema	1	20 \$
Construcción	Montaje	--	300 \$
Diseño	Planos	--	100 \$
Trasporte	Uso	--	20 \$
Imprevistos	Extras	--	39 \$
Total			1,200 \$

Elaborado por: El investigador

Los costos establecidos están sujetos a valores por unidad, tomando en cuenta el precio unitario de los materiales que se adquieren en el mercado actual a nivel del país, este valor de fabricación final es \$1,200.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- ❖ El sistema de reciclaje “OPEN BOT”, está diseñado para brindar una reducción de contaminantes al medio e incentivar a la sociedad al reciclaje, utilizando tecnología disponible en el mercado. El sistema brinda al usuario la opción de dispensador de incentivos cuando tiene los créditos suficientes para hacerlo, además cuenta con la opción de compactar envases, aprovechando al máximo su capacidad de 350 contenedores.
- ❖ El sistema al verificar mediante un lector de código de barra, evalúa los envases si son PET, permitiendo captar únicamente envases que circulan en el mercado, logrando así no ingresar contenedores que no sean reciclables y evitar daños en el sistema.
- ❖ Open Bot se diferencia de otros sistemas de reciclaje, ya que utiliza software y hardware libre, a diferencia de otros, es automático y tiene capacidad sensorial, además posee un sistema de dispensación de incentivos, y compresión de envases que ayuda obtener mayor almacenamiento al reducir un 50% del volumen, con un margen de error de compresión $\pm 5\%$.
- ❖ Al sistema lo hace inteligente al contar con sensores de validación para ejecutar cada acción, siendo estos: lectura y validación de código de barras, detención de envases en el interior, detección de obstáculo en el ingreso. Logrando cumplir eventualmente su aplicación de reciclaje y dispensador de incentivos

- ❖ Open Bot es un sistema inteligente programado que realiza tareas determinadas en base a patrones de señales, gracias a los sensores de medición que utiliza para la toma de decisiones, el sistema procesa de manera eficiente ya que está dividido en áreas como: el área de dispensación, área de almacenamiento, el área de interacción del usuario y el medio de gestión de datos, todos ellos se encuentran comunicados mediante la tarjeta gráfica Raspberry Pi4 que tiene la capacidad de almacenar información en su base de datos MSQl.
- ❖ El sistema se puede mejorar relacionando supermercados y estaciones de servicios, de tal manera que los clientes pondrán hacer usos de los créditos acumulados en el sistema, como un medio de pago por los servicios y productos adquiridos.
- ❖ Actualmente con este diseño las áreas de oportunidad para OPEN BOT son centros turísticos, medios públicos/privados y unidades educativas, donde existe mayor consumo de bebidas envasadas en PET reduciendo sus áreas ante la contaminación.

4.2 Recomendaciones

- ❖ Por la configuración del sistema requiere de un punto de conexión eléctrica, la cual debe ser suministrada constantemente para el óptimo funcionamiento.
- ❖ El sistema al tener la opción de compactar mediante actuadores electrónicos sensibles es recomendable ingresar la tapa en su lugar ya que puede complicarse al momento de la reducción del volumen de los envases y no se lleve a cabo esta acción.
- ❖ Para que el sistema OPEN BOT admita la mayor cantidad de envases que circulan en el mercado, un administrador debe registrar cada una de ellas para posterior no tener complicaciones de captación y el usuario recicle de forma óptima los envases PET.
- ❖ Se recomienda ser instalado en lugares públicos de mayor frecuencia si está dirigido a empresas públicas, o en lugares de servicios de bebida si la empresa es privada.

- ❖ Se realizó un manual dirigido hacia el usuario y personal de mantenimiento con fácil entendimiento, con información necesaria en caso de requerir cambios de dispositivos o mecanismos del sistema.

BIBLIOGRAFÍA

- [1] Mazharul Islam, Mohammad Farhan Ferdous, Md Shajidur Rahman Almazee, Md Shahriar Dipto, Kayanat Tahera Sultana, Azmir Hasan, «IEEE,» 02 06 2019. [En línea]. Available: <https://ieeexplore.ieee.org/document/8858577>. [Último acceso: 12 03 2020].
- [2] Pravin Dhulekar, S. T. Gandhe, Ulhas P. Mahajan, «System of recycling, » IEEE, 09 02 2018. [En línea]. Available: <https://ieeexplore.ieee.org/document/8529483>. [Último acceso: 16 08 2019].
- [3] Yu-Ping Liao, Ruei-Chang Lu, Sheng-Ying Wu, Ping-En Cheng, Guo-Cheng Xu, «IEEE,» 04 11 2018. [En línea]. Available: <https://ieeexplore.ieee.org/document/8606758>. [Último acceso: 12 03 2020].
- [4] Wisdom Gen P. Dumpayan, Matthew Lawrence M. De Mesa, Nathalie Danielle F. Yucor, Eden T. Gabion, Jacqueline D. Reynoso, Gabriel Rodnei M. Geslani, «Máquina expendedora inversa de botellas bidireccionales de plástico con microcontrolador,» de IEEE, Philippines, 2017
- [5] Tritubot, «EP Petroecuador recicla,» Petroecuador, 04 2019. [En línea]. Available: <https://www.eppetroecuador.ec/?p=7266>. [Último acceso: 17 08 2019].
- [6] Marloun Sejera, Joseph Bryan Ibarra, Anrol Sarah Canare, Lyra Escano, Dianne Claudinne Mapanoo, John Phillip Suaviso, «Contenedor de basura automatizado basado en frecuencia independiente,» de IEEE, Filipinas, 2017.
- [7] Maofic Farhan Karin, Hasan U. Zaman, Khandaker Sharif Noor, «Recycling Machine using FPGA,» IEEE, 18 12 2016. [En línea]. Available: <https://ieeexplore.ieee.org/document/7920701>. [Último acceso: 16 08 2019].

- [8] Witaya Srigul, Mongkol Kupimai, Prajuab Inrawong, «IEEE,» 08 09 2016. [En línea]. Available: <https://ieeexplore.ieee.org/document/7561304>. [Último acceso: 13 12 2020].
- [9] Eva Martín Roda, María del Pilar Borderías, Medio Ambiente Urbano, Madrid: ISBN, 2006, pp 4-10.
- [10] Assen Toledo Argüelles, José C. Cadalso Basadre, Alyn Ferro Nieto, El envase PET: su impacto ambiental, La Habana: Universitaria, 2008, pp 6-12.
- [11] David Rosendo Ramos, El reciclaje en la sociedad actual, España: WANCEULEN, 2010.
- [12] Ministerio de Ambiente, «Ministerio de Ambiente,» GOB, 01 2020. [En línea]. Available: <https://www.ambiente.gob.ec/>. [Último acceso: 12 05 2020].
- [13] Petroecuador, «Petroecuador recicla,» 02 2020. [En línea]. Available: <https://www.eppetroecuador.ec/>. [Último acceso: 05 2020].
- [14] Vistazo, «Ecuador recicla,» 04 2020. [En línea]. Available: <https://www.vistazo.com/seccion/vida-moderna/ambiente/ecuador-avanza-en-el-camino-de-la-cultura-del-reciclaje>. [Último acceso: 05 2020].
- [15] OTRI, «Sistemas Inteligentes,» Universidad de Pablo Olavide, 2017. [En línea]. Available: <https://www.upo.es/upotec/catalogo/telecomunicaciones-electronica-e-informatica/sistemas-inteligentes-que-automatizan-el-analisis-/>. [Último acceso: 17 08 2019].
- [16] UNED, «Sistemas Inteligentes,» UNED, 2017. [En línea]. Available: http://portal.uned.es/EadmonGuiasWeb/htdocs/abrir_fichero/abrir_fichero.jsp?idGuia=86334. [Último acceso: 17 08 2019].

- [17] Ángel Pablo Hinojosa Gutiérrez, Python, Madrid, España: RA-MA, S.A., 2016, pp. 21-24.
- [18] Alberto Cuevas Álvarez, Aplicaciones gráficas con Python 3, Madrid, España: RA-MA, 2018, pp. 15-20.
- [19] Antonio Jesús Álvarez Martínez, Roció María Olivares Molina, Héctor Górriz Sáez, Procesamiento digital de imagen, España: universidad de Almería, 2018, pp. 43-45.
- [20] PLASMAQ, «Sistemas de reciclaje,» PLASMAQ, 2019. [En línea]. Available: <http://www.plasmaq.pt/es/productos/sdfg-2/>. [Último acceso: 17 08 2019].
- [21] La comunidad Valenciana, «Sistema de reciclaje,» ECOEMBES, 2018. [En línea]. Available: <http://www.reciclaconloscincosentidos.com/el-reciclaje/el-sistema-de-reciclaje/>. [Último acceso: 18 08 2019].
- [22] Ecología Verde, «Contenedores inteligentes,» Ecología Verde, 2018. [En línea]. Available: <https://www.ecologiaverde.com/contenedores-de-reciclaje-inteligentes-203.html>. [Último acceso: 18 08 2019].
- [23] La vanguardia, «Cómo los sistemas de basura inteligentes reducen los costes de recogida,» 12 2019. [En línea]. Available: <https://www.lavanguardia.com/basura-smart-ciudad.html>. [Último acceso: 04 2020].
- [24] El ágora, «Contenedores inteligentes,» 11 2019. [En línea]. Available: <https://www.elagoradiario.com/contenedores-inteligentes-residuos-electronicos/>. [Último acceso: 04 2020].
- [25] Residuos Profesional, «Contenedor inteligente,» 02 2018. [En línea]. Available: <https://www.residuosprofesional.com/contenedor-inteligente-plasticos-hoteles/>. [Último acceso: 05 2020].

- [26] El Universo, «Economía,» Consumo de agua embotellada, 05 08 2019.
- [27] Petroecuador, «Sistema Tritubot,» Petroecuador, 2018. [En línea]. Available: <https://www.eppetroecuador.ec/?p=7266>. [Último acceso: 19 08 2019].
- [28] Ministerio de Producción, comercio exterior, inversiones y pesca, «Ecuador apunta al Pacto por la Economía Circular,» 08 2019. [En línea]. Available: www.produccion.gob.ec. [Último acceso: 12 07 2020].
- [29] Eugenio. Vildósola., «<http://www.aie.cl>,» 2016. [En línea]. Available: www.aie.cl. [Último acceso: 2 10 2019].
- [30] Antonio Pérez González, José L. Iserte Vilar, Octavio Bernad Ros, Sistemas mecánicos para industrias, UNIVERSITAT JAUME I, 2012, pp. 12-20.
- [31] Leonel G. Corona Ramírez, Griselda S. Abarca Jiménez, Jesús Mares Carreño, Sensores y Actuadores, México: SBN ebook, 2014, pp 10-25.
- [32] El universo, «Economía por reciclaje,» 05 2019. [En línea]. Available: <https://www.eluniverso.com/se-puede-ganar-reciclar>. [Último acceso: 06 2020].
- [33] Petroecuador, «Ubicación de Tritubot,» 02 2019. [En línea]. Available: <https://www.eppetroecuador.ec/estaciones-tritubot.pdf>. [Último acceso: 04 2020].
- [34] Bioverde, «Tecnologías de reciclaje,» 05 2019. [En línea]. Available: <https://www.leonardo-gr.com/es/tecnolog-del-reciclaje>. [Último acceso: 06 2020].
- [35] BIOBOX, «contenedor Inteligente,» 01 2020. [En línea]. Available: <https://www.biobox.com.mx/>. [Último acceso: 06 2020].

- [36] Bigbelly, «Bigbelly,» Future Street, 2020. [En línea]. Available: <http://futurestreet.com/es/platform/>. [Último acceso: 20 07 2020].
- [37] VIDORTEC, «Contenedor Inteligente,» 02 2020. [En línea]. Available: www.vidortec.com.ec/contenedores-inteligentes/. [Último acceso: 06 2020].
- [38] Sensoneo, «Contenedor Inteligente,» 08 2019. [En línea]. Available: <https://sensoneo.com/es/solucion-inteligente-de-gestion-de-residuos-para-la-ciudad/>. [Último acceso: 06 2020].
- [39] Recyclia, «Contenedor Inteligente,» 02 2019. [En línea]. Available: www.recyclia.es/proyectos-europeos/circ4life/. [Último acceso: 06 2020].
- [40] PUCE, «Datos Climáticos,» [En línea]. Available: <https://bioweb.bio/faunaweb/amphibiaweb/GeografiaClima/>. [Último acceso: 04 2020].

Anexos

Anexo 1: Datasheet de Raspberry Pi 4



2 Features

2.1 Hardware

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

2.2 Interfaces

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



4.1 Power Requirements

The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.

5 Peripherals

5.1 GPIO Interface

The Pi4B makes 28 BCM2711 GPIOs available via a standard Raspberry Pi 40-pin header. This header is backwards compatible with all previous Raspberry Pi boards with a 40-way header.

5.1.1 GPIO Pin Assignments

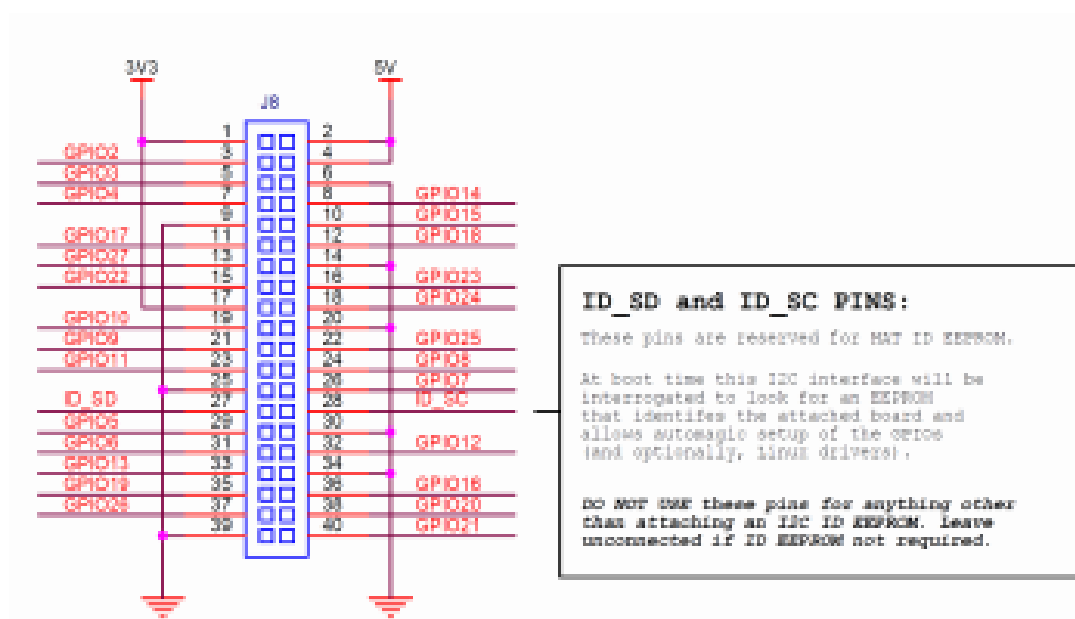


Figure 3: GPIO Connector Pinout

As well as being able to be used as straightforward software controlled input and output (with programmable pulls), GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI.

In addition to the standard peripheral options found on legacy Pis, extra I2C, UART and SPI peripherals have been added to the BCM2711 chip and are available as further mux options on the Pi4. This gives users much more flexibility when attaching add-on hardware as compared to older models.



5.1.2 GPIO Alternate Functions

GPIO	Default						
	Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	SPI3_CE0_N	TXD2	SDA6
1	High	SCL0	SA4	DE	SPI3_MISO	RXD2	SCL6
2	High	SDA1	SA3	LCD_VSYNC	SPI3_MOSI	CTS2	SDA3
3	High	SCL1	SA2	LCD_HSYNC	SPI3_SCLK	RTS2	SCL3
4	High	GPCLK0	SA1	DPLD0	SPI4_CE0_N	TXD3	SDA3
5	High	GPCLK1	SA0	DPLD1	SPI4_MISO	RXD3	SCL3
6	High	GPCLK2	SOE_N	DPLD2	SPI4_MOSI	CTS3	SDA4
7	High	SPI0_CE1_N	SWE_N	DPLD3	SPI4_SCLK	RTS3	SCL4
8	High	SPI0_CE0_N	SD0	DPLD4	-	TXD4	SDA4
9	Low	SPI0_MISO	SD1	DPLD5	-	RXD4	SCL4
10	Low	SPI0_MOSI	SD2	DPLD6	-	CTS4	SDA5
11	Low	SPI0_SCLK	SD3	DPLD7	-	RTS4	SCL5
12	Low	PWM0	SD4	DPLD8	SPI5_CE0_N	TXD5	SDA5
13	Low	PWM1	SD5	DPLD9	SPI5_MISO	RXD5	SCL5
14	Low	TXD0	SD6	DPLD10	SPI5_MOSI	CTS5	TXD1
15	Low	RXD0	SD7	DPLD11	SPI5_SCLK	RTS5	RXD1
16	Low	FL0	SD8	DPLD12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPLD13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPLD14	SPI6_CE0_N	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPLD15	SPI6_MISO	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPLD16	SPI6_MOSI	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPLD17	SPI6_SCLK	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPLD18	SD1_CLK	ARM_TRST	SDA6
23	Low	SD0_CMD	SD15	DPLD19	SD1_CMD	ARM_RTCK	SCL6
24	Low	SD0_DAT0	SD16	DPLD20	SD1_DAT0	ARM_TDO	SPI3_CE1_N
25	Low	SD0_DAT1	SD17	DPLD21	SD1_DAT1	ARM_TCK	SPI4_CE1_N
26	Low	SD0_DAT2	TE0	DPLD22	SD1_DAT2	ARM_TDI	SPI5_CE1_N
27	Low	SD0_DAT3	TE1	DPLD23	SD1_DAT3	ARM_TMS	SPI6_CE1_N

Table 5: Raspberry Pi 4 GPIO Alternate Functions

Table 5 details the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the BCM2711 Peripherals Specification document which can be downloaded from the hardware documentation section of the website.



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time*velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

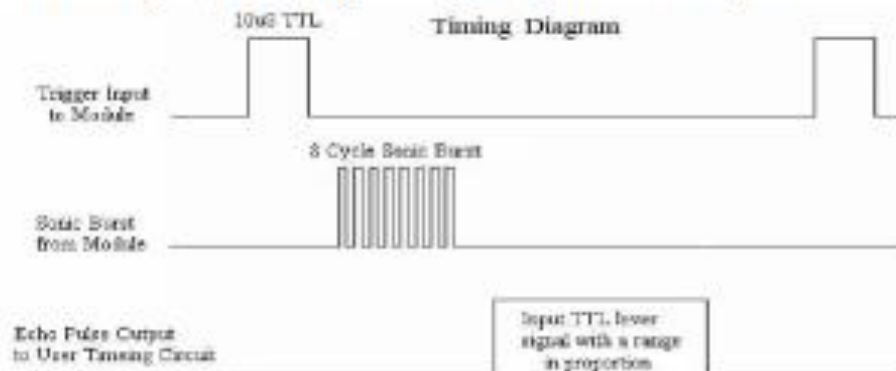
Electric Parameter

Working Voltage	DC 5V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL level signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

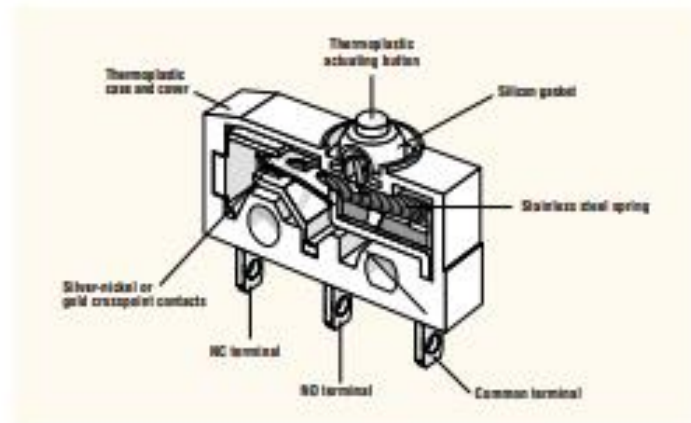




SUBMINIATURE — SEALED DC Series

Features

- Enclosed switch complying with IP 6K7
- Silicon-free version available
- Models available for operating temperatures up to 120°C
- Rated for currents up to 10 amp at 250VAC
- Range of auxiliary actuators available (can also be retrofit)
- Various contact materials available, depending on application
- Mechanical life: min. 1×10^6 operations
- Wide variety of connection options



Electrical Ratings

Switch Series	EN61058 Rating	UL1058 Rating	Electrical Life at Rated Load:	
			According to EN (Mio. Operations)	According to UL (Mio. Operations)
DC1	6A, 250V~	5A, 125/250VAC	10,000	6,000
DC2	10(1.5)A, 250V~	10.1A, 125/250VAC; 1/4HP, 125VAC	10,000	6,000
DC3	0.1A, 250V~	0.1A, 125/250VAC	50,000	6,000
DC4	3A, 250V~	3A, 125/250VAC	50,000	6,000

Specifications

Electrical

Temperature Rating: -40° to +85°C / +120°C (without wire leads)
-40° to +85°C / +105°C (with wire leads)

Flammability Rating: UL94V-0 (PET, PBT)
UL94HB (POM)

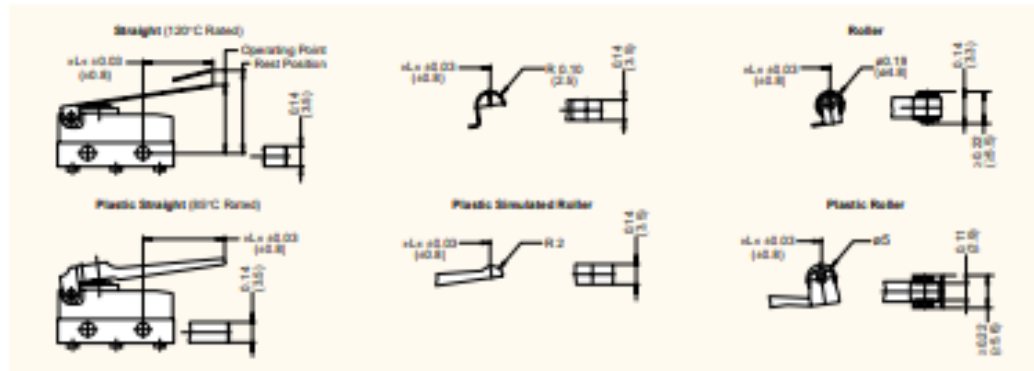
Materials

Case: PET / PBT
Actuator: POM (T85), PBT (T120)
Auxiliary Actuator: Stainless Steel or Plastic
Terminals: Silver-Plated Copper-Zinc
Contacts: Silver Alloy (DC3 — Gold Crosspoint)

PBT = Polybutylene terephthalate • PET = Polyethylene terephthalate • POM = Polyacetal



Auxiliary Actuators inches (mm)



Actuator Specifications

Actuator Code	Switch Type	Maximum Operating Force (N)		Maximum Pre-Travel inches (mm)		Operating Point inches (mm)		Minimum Over-Travel inches (mm)	Max. Movement Differential inches (mm)		Max. Rest Position inches (mm)	Actuation Length inches (mm)
		DC1, DC2	DC3	DC1, DC2	DC3	DC1, DC2	DC3		DC1, DC2	DC3		
AA		200 (1.0)	340 (1.0)	0.04 (1.0)	0.04 (1.0)	0.33±0.01 (8.4±0.3)	0.33±0.01 (8.4±0.3)	0.02 (0.5)	0.004 (0.10)	0.004 (0.10)	0.37 (9.3)	—
LB		80	150	0.18 (4.5)	0.20 (5.0)	0.42±0.05 (10.7±1.3)	0.42±0.06 (10.7±1.6)	0.06 (1.5)	0.02 (0.5)	0.03 (0.7)	0.55 (14.0)	0.189 (4.8)
LC		70	120	0.20 (5.0)	0.22 (5.5)	0.44±0.06 (11.1±1.5)	0.44±0.07 (11.1±1.8)	0.06 (1.5)	0.02 (0.6)	0.04 (1.0)	0.59 (15.0)	0.276 (7.0)
LD		force and travel available upon request										
SB		90	160	0.18 (4.5)	0.20 (5.0)	0.63±0.05 (16.0±1.3)	0.63±0.06 (16.0±1.6)	0.06 (1.5)	0.02 (0.5)	0.03 (0.7)	0.75 (19.0)	1.654 (42.0)
SC		80	130	0.20 (5.0)	0.22 (5.5)	0.65±0.06 (16.4±1.5)	0.65±0.07 (16.4±1.8)	0.06 (1.5)	0.02 (0.6)	0.04 (1.0)	0.79 (20.0)	0.185 (4.7)
SD		force and travel available upon request										
RB		90	160	0.18 (4.5)	0.20 (5.0)	0.62±0.05 (15.8±1.3)	0.62±0.06 (15.8±1.6)	0.06 (1.5)	0.02 (0.5)	0.03 (0.7)	0.75 (19.0)	0.068 (2.5)
RC		80	130	0.20 (5.0)	0.22 (5.5)	0.64±0.06 (16.2±1.5)	0.64±0.07 (16.2±1.8)	0.06 (1.5)	0.02 (0.6)	0.04 (1.0)	0.79 (20.0)	0.185 (4.7)
RD		force and travel available upon request										
WB		68	115	0.18 (4.5)	0.18 (4.5)	0.44±0.06 (11.1±1.5)	0.44±0.06 (11.1±1.5)	0.08 (2.0)	0.02 (0.6)	0.02 (0.6)	0.59 (15.0)	0.276 (7.0)
WC		50	85	0.24 (6.0)	0.24 (6.0)	0.48±0.07 (12.2±1.8)	0.48±0.07 (12.2±1.8)	0.12 (3.0)	0.03 (0.8)	0.03 (0.8)	0.67 (17.0)	0.551 (14.0)
VB		73	125	0.18 (4.5)	0.18 (4.5)	0.47±0.06 (11.9±1.4)	0.47±0.06 (11.9±1.4)	0.08 (2.0)	0.02 (0.6)	0.02 (0.6)	0.59 (15.0)	0.220 (5.6)
ZB		73	125	0.18 (4.5)	0.18 (4.5)	0.63±0.06 (16.0±1.4)	0.63±0.06 (16.0±1.4)	0.06 (1.5)	0.02 (0.6)	0.02 (0.6)	0.75 (19.0)	0.265 (5.2)

Anexo 4: Datasheet de sensor DHT-22

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

1. Feature & Application:

- * Full range temperature compensated
- * Relative humidity and temperature measurement
- * Calibrated digital signal
- * Outstanding long-term stability
- * Extra components not needed
- * Long transmission distance
- * Low power consumption
- * 4 pins packaged and fully interchangeable

2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

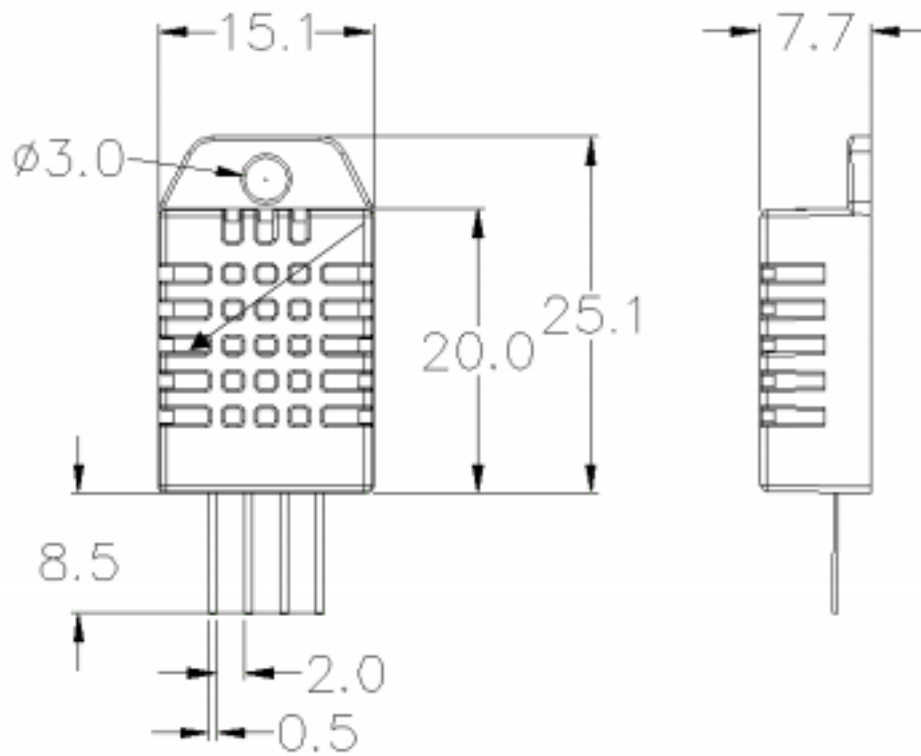
Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40-80Celsius
Accuracy	humidity +2%RH(Max +5%RH); temperature <+0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +/-1%RH; temperature +/-0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

4. Dimensions: (unit---mm)

1) Small size dimensions: (unit---mm)

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD—power supply
2	DATA—signal
3	NULL
4	GND

Thomas Liu (Business Manager)

4

Anexo 5: Datasheet de servo motor 25 Kg

Siempre versión resistente al agua

Product Description

Brand	Dsservo	
Product Name	DS3225	DS3225 Coreless Digital Servo
Torque	21 kg-cm @5V	22 kg-cm @5V
	24.5 kg-cm @6.8V	24 kg-cm @6V
		25.5 kg-cm @7.4V
Speed	0.15 sec/60° @5V	0.1 sec/60° @5V
	0.13 sec/60° @6.8V	0.08 sec/60° @6V
		0.07 sec/60° @7.4V
Note	High Torque Ensure that the BEO has enough output power Recommend using 6V	High Speed - High Efficiency Recommend using 7.4V

Especificación:

Marca: Dsservo

Artículo: DS3225 25KG alto par Digital Servo

Servo Digital de engranaje de acero inoxidable sin núcleo DS3225 25KG

Voltaje de funcionamiento: 5 ~ 6 voltios DC (DS3225)
5 ~ 7,4 DC voltios (DS3235)

Muerto marca: 3µs / 2 µs

Peso: 60g (2,12 oz)

Tipo de Motor: Motor DC Motor Sin Núcleo

Tipo de engranaje: cobre y aluminio/Acero inoxidable y aluminio

Frecuencia de funcionamiento: 50-333Hz

Tamaño: 40x20x40,5mm (1,58x0,79x1,60 en)

40x20x38,5mm (1,58x0,79x1,50 en)

Características:

- Servo estándar digital de alto rendimiento
- Engranajes de metal de alta precisión con anodizado duro
- Carcasa central de aluminio CNC
- Doble rodamientos de bolas

Crisol de grafito mini horno de oro de la antorcha de fusión del metal:

1 x DS3225 25KG alto par Digital Servo

1x25T de metal ajustable brazo de servo

O

1 x DS3225 25KG Servo Digital de engranaje de acero inoxidable sin núcleo

1x25T de metal ajustable brazo de servo

Un conjunto de accesorios

Anexo 6: Datasheet de motor a paso Nema 17

Quick Reference NEMA size 17 1.8° 2-phase stepper motor



Schneider
Electric

Notes and Warnings

Installation, configuration and maintenance must be carried out by qualified technicians only. You must have detailed information to be able to carry out this work.

- Unexpected dangers may be encountered when working with this product!
- Incorrect use may destroy this product and connected components!

For more information, go to www.lmshome.com

Specifications

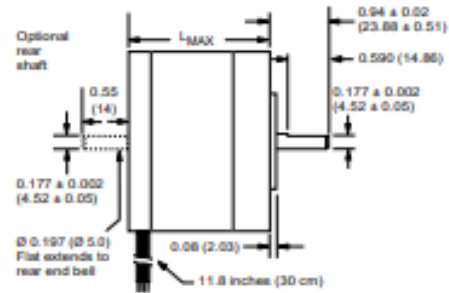
1.5 Amp motors		Single length	Double length	Triple length
Part number		M-1713-1.5 + (1)	M-1715-1.5 + (1)	M-1719-1.5 + (1)
Holding torque	oz-in	32	60	75
	N-cm	23	42	53
Detent torque	oz-in	1.7	2.1	3.5
	N-cm	1.2	1.5	2.5
Rotor inertia	oz-in-sec ²	0.000536	0.000807	0.0011562
	kg-cm ²	0.038	0.057	0.082
Weight	oz	7.4	8.1	12.7
	grams	210	230	360
Phase current	amps	1.5	1.5	1.5
Phase resistance	ohms	1.3	2.1	2.0
Phase inductance	mH	2.1	5.0	3.85

(1) indicate S for single-shaft or D for double-shaft. Example M-1713-1.5S

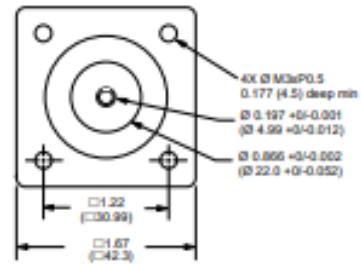
Wiring and Connections

Signals and wire colors	
Phase A	Red
Phase B	Blue
Phase S	Green
Phase G	Black

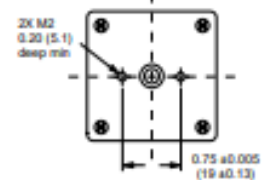
Mechanical Specifications Dimensions in inches (mm)



FRONT VIEW



REAR VIEW (Reduced)



Motor stack length inches (mm)	Single	Double	Triple
L _{MAX}	1.34 (34.0)	1.57 (40)	1.89 (48)

Part Numbers

Example:	M - 1 7 1 3 - 1.5 S
Stepper motor frame size	M - 1 7 1 3 - 1.5 S
M-17 = NEMA 17 (1.7"/42 mm)	
Motor length	M - 1 7 1 3 - 1.5 S
13 = single stack	
15 = double stack	
19 = triple stack	
Phase current	M - 1 7 1 3 - 1.5 S
1.5 = 1.5 Amps	
Shaft	M - 1 7 1 3 - 1.5 S
S = single, front shaft only	
D = double, front and rear shafts	
Optional optical encoder (1)	M - 1 7 1 3 - 1.5 E S 1 0 0
ES = Single-end	
ED = Differential	
Line count	
100, 200, 250, 400, 500 or 1000 (2)	

Anexo 7: Datos Técnicos de Pantalla HDMI 10.1 pulgadas

The LCD and Control Board

IPS screen, 1280x800 hardware resolution
Toughened glass capacitive touch panel, 6H hardness
Supports popular mini PCs such as Raspberry Pi, BB Black, as well as general desktop computers
When works with Raspberry Pi, supports Raspbian/Ubuntu/Kali/Retropie/WIN10 IOT, driver free
When work as a computer monitor, supports Windows 10/8.1/8/7, ten-points touch, and driver free
HDMI interface for displaying, USB interface for touch control
Supports 5-level backlight adjustment
Note: to use the LCD with following mini-PCs, additional cables are required and should be purchased separately:
Raspberry Pi Zero: HDMI cable, USB-type-A-receptacle-to-Micro-B-plug-cable
BB Black: HDMI cable

The Case

Material : high quality black Acrylic
Comes with back holder, 45° tilt angle
Features mounting holes for Raspberry Pi 3B+/3B/2B/B+/A+/B, BB Black, Banana Pi

Development Resources

Wiki : [www.waveshare.com/wiki/10.1inch_HDMI_LCD_\(B\)_\(with_case\)](http://www.waveshare.com/wiki/10.1inch_HDMI_LCD_(B)_(with_case))

Product information

Product Dimensions	7.09 x 3.94 x 3.54 inches
Item Weight	1.76 pounds
Manufacturer	Waveshare
ASIN	B01H013FGC
Item model number	10.1inch HDMI LCD (B) (with case)

Anexo 8: Manual de Usuario OPEN BOT



OPEN BOT **Manual de Usuario**

Versión: 1
Fecha: 20/12/2020



**OPEN BOT
Manual Técnico**

**Universidad Técnica de
Ambato**

HOJA DE CONTROL

Organismo	Universidad Técnica de Ambato		
Proyecto	Sistema de Reciclaje OPEN BOT		
Entregable	Manual de Usuario		
Autor	Cristian Satuquinga		
Versión/Edición	1	Fecha Versión	20/12/2020
Aprobado por		Fecha Aprobación	20/12/2020
		N° Total de Páginas	12

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1	Versión inicial	Cristian Satuquinga	20/12/2020

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos
Universidad Técnica de Ambato



1 DESCRIPCIÓN DEL SISTEMA

1.1 Objeto

Dar a conocer el uso adecuado del sistema Open Bot en aspectos técnicos de manera descriptiva e ilustrada sobre los componentes y funcionalidades.

1.2 Alcance

El manual técnico va dirigido a los usuarios y técnicos para dar a conocer la información necesaria para realizar la debida utilización, mantenimiento e instalación del sistema Open Bot, el cual es un sistema que permite fomentar la cultura de reciclar. Además de generar incentivos al usuario.

1.3 Funcionalidad

OPEN BOT es un sistema inteligente de reciclaje de software libre, mediante la tarjeta raspberry Pi4 y el sistema operativo raspbian, se utilizó el lenguaje de programación Python, por su simplicidad e interpretación, permitiendo la facilidad de comunicación entre el usuario y máquina, gracias a sus extensas librerías que contiene. A diferencia de otros sistemas, OPEN BOT comprime los envases, permitiendo mayor almacenamiento en sus contenedores, además gestiona créditos y genera incentivos por ejecutar la acción de reciclar, con el objetivo de contribuir con el medio ambiente, mediante la reducción de residuos. El sistema evalúa condiciones al momento de almacenar el envase plástico PET como: nivel de contenedor, ingreso del envase, registro del código de barras de las bebidas más comerciales, el registro de usuario y la acumulación de créditos disponibles para dispensar y otorgar el incentivo hacia el usuario. También contiene una interfaz gráfica que consiste en cinco Frame's: pantalla inicial, inicio de sesión del usuario, inicio de sesión del administrador, registro y recuperación de sesión, desarrollado en código de interpretación y gestionado en una base datos



2 MANUAL DE USUARIO

En la pantalla principal se visualiza información como:

- Registrar
- Iniciar sesión
- Recuperar clave
- Iniciar administrador
- La cantidad de botellas almacenadas
- Datos como: la temperatura, fecha y hora



De tal manera que el sistema cuenta con 3 modos de trabajo para que los usuarios pueden ingresar, los cuales son:

- El modo invitado: es para las personas que deseen reciclar sin la necesidad de requerir una cuenta en el sistema
- El modo usuario: es para las personas que cuentan con una cuenta registrada en el sistema
- El modo administrador: es exclusivamente para el personal técnico de mantenimiento y/o adquirente del producto

Modo Invitado

Para ingresar en modo invitado el usuario debe seleccionar "ingresar" y automáticamente ingresa al sistema





Nota: El modo invitado del sistema no acredita puntos al momento de realizar el depósito del envase.

Modo Usuario

En el modo usuario las personas deben tener una cuenta de acceso al sistema para iniciar sesión en caso contrario debes realizar el proceso de registrarse **ver modo registro en la siguiente página.**

El usuario debe ingresar al sistema con los datos de información:

- Usuario (# Cedula de Ciudadanía)
- Clave (pin de 4 dígitos)



Modo administrador

Para ingresar al modo administrador se selecciona “iniciar administrador” este modo es para el personal técnico de mantenimiento y/o adquisidor del producto, su principal función es registrar los códigos de barras de envases de bebidas.



- Selecciona la opción registrar
- Colocar el código de barras del envase en el scanner para ser captado



Registrarse en el sistema

En la pantalla principal el usuario debe seleccionar “registrar” y proceder a llenar todos los campos de información sugeridos por el sistema como:

- Usuario (# Cedula de Ciudadanía)
- Clave (pin de 4 dígitos)
- Comprobación de Clave (pin de 4 dígitos)
- Preguntas de Seguridad (color, mascota, y fecha de nacimiento)



Proceso para realizar reciclaje de un envase PET

1. El usuario debe ingresar al sistema con los datos de información:

- Usuario (# Cedula de Ciudadanía)
- Clave (pin de 4 dígitos)



O en modo invitado si desea donar el envase





Cuando el usuario haya ingresado al sistema se visualizará una pantalla donde indica:

- El número de cedula del usuario,
- Los puntos acreditados por los envases ingresados,
- La opción de dispensar en caso de que desee reclamar el incentivo
- La opción de ingresar para introducir un nuevo envase en el contenedor.



2. Depositar la tapa del envase



3. Escanear el código de barras del envase

Esperar que el sistema verifique si puede ingresar envases al contenedor e inmediatamente se abre la compuerta del almacenamiento de envase



4. Depositar el envase



Esperar a que el sistema acredite puntos en la cuenta por el envase ingresado





5. **Dispensar**

Para dispensar, el usuario tiene que seleccionar la opción “dispensar”; pero el usuario antes de seleccionar debe primero verificar que su dinero acreditado acumulado sea mayor o igual a 0.50 para obtener su incentivo



6. **Cerrar sesión**

Cierre sesión para que los puntos de su cuenta no sean utilizados por otros usuarios.

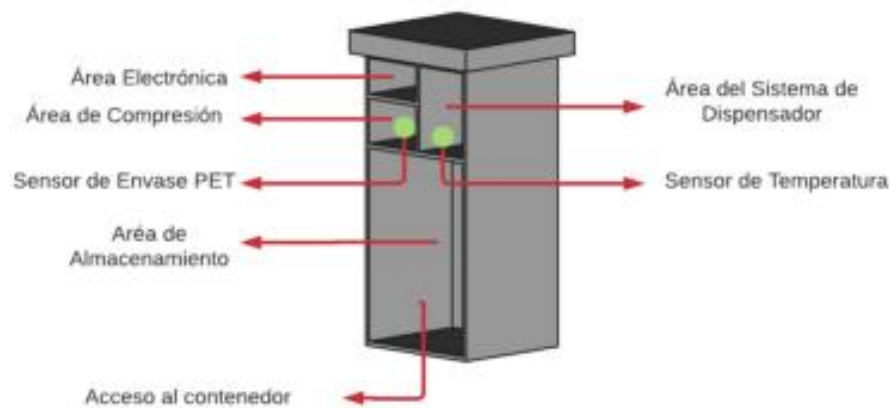
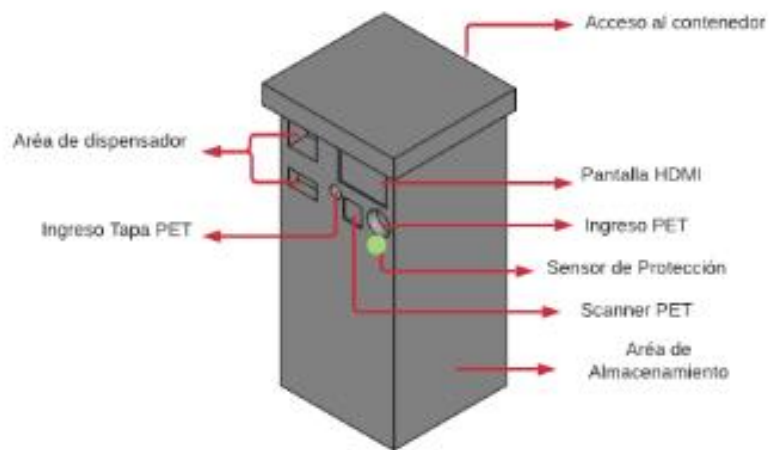


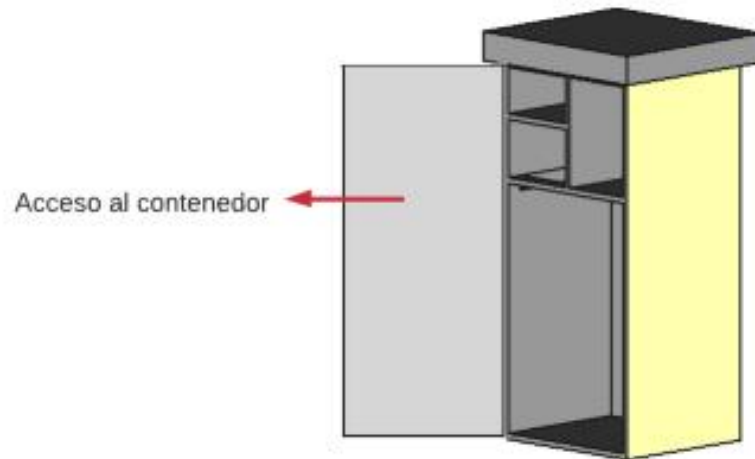


3 MANUAL TÉCNICO

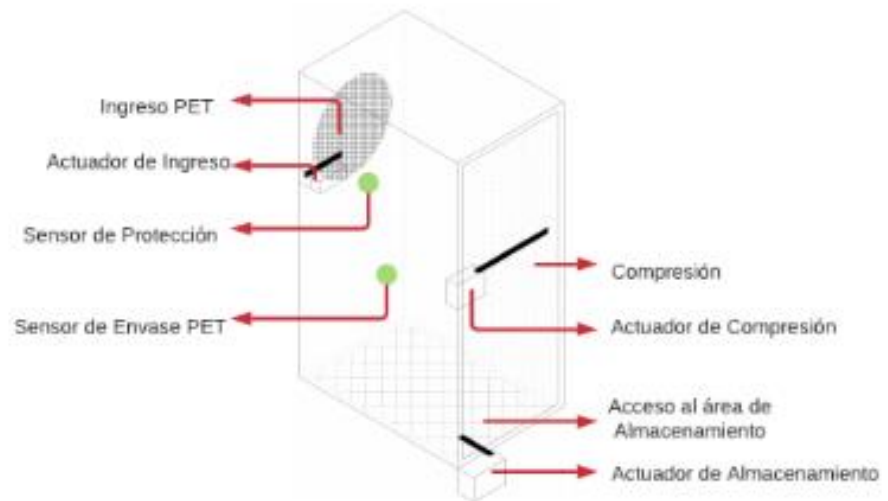
El sistema OPEN BOT Tiene sensores, actuadores y dispositivos electrónicos ubicados en el prototipo, se detallan su estructura y ubicación a continuación.

1. Partes del Sistema OPEN BOT



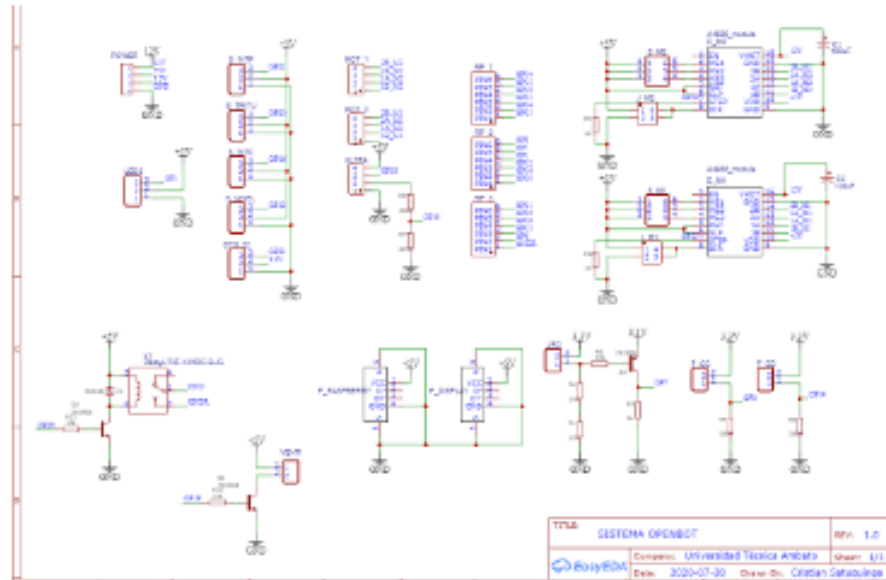


2. Partes de compresión OPEN BOT





3. Esquema de OPEN BOT



Anexo 9: Código de interfaz gráfica desarrollada en Python.

El presente anexo detalla el código que se encuentra tras la interfaz gráfica del sistema, configurada para la Raspberry pi 4. El mismo que se encarga de administrar los usuarios y controlar sensores y actuadores del sistema. Junto a sus diagramas de flujo que se describen a continuación.

Programación Python Frame Teclado. – Crea Botones para la el ingreso de datos de los usuarios hacia el sistema. Su código fuente es la siguiente y su diagrama se aprecia en la **Figura 49**.

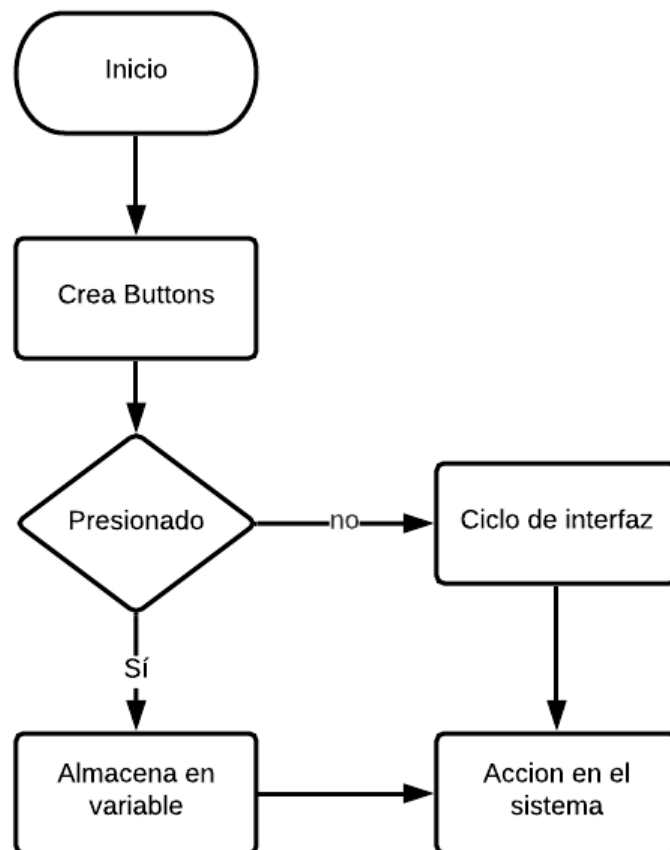


Figura 49: Diagrama Frame Buttons

Elaborado por: El investigador

```
self.button7=Button(self.labelFrameVentanaTeclado,text="7",justify="center",
command=lambda:self.numeroPulsado("7"))

self.button7.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.button7.place(x=self.filax1, y=self.columnay1)
```

```
self.button8=Button(self.labelFrameVentanaTeclado,text="8", justify="center",
command=lambda:self.numeroPulsado("8"))

self.button8.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.button8.place(x=self.filax2, y=self.columnay1)
```

```
self.button9=Button(self.labelFrameVentanaTeclado,text="9", justify="center",
command=lambda:self.numeroPulsado("9"))

self.button9.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.button9.place(x=self.filax3, y=self.columnay1)
```

```
self.button4=Button(self.labelFrameVentanaTeclado,text="4",
command=lambda:self.numeroPulsado("4"))

self.button4.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.button4.place(x=self.filax1, y=self.columnay2)
```

```
self.button5=Button(self.labelFrameVentanaTeclado,text="5",
command=lambda:self.numeroPulsado("5"))

self.button5.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.button5.place(x=self.filax2, y=self.columnay2)
```

```
self.button6=Button(self.labelFrameVentanaTeclado,text="6",
command=lambda:self.numeroPulsado("6"))

self.button6.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.button6.place(x=self.filax3, y=self.columnay2)
```

```
self.button1=Button(self.labelFrameVentanaTeclado,text="1",
command=lambda:self.numeroPulsado("1"))
```

```
self.button1.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.button1.place(x=self.filax1, y=self.columnay3)
```

```
self.button2=Button(self.labelFrameVentanaTeclado,text="2",
command=lambda:self.numeroPulsado("2"))
self.button2.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.button2.place(x=self.filax2, y=self.columnay3)
```

```
self.button3=Button(self.labelFrameVentanaTeclado,text="3",
command=lambda:self.numeroPulsado("3"))
self.button3.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.button3.place(x=self.filax3, y=self.columnay3)
```

```
self.buttonBorrarTodo=Button(self.labelFrameVentanaTeclado, text="Borrar ",
command=lambda:self.borrarTodo())
self.buttonBorrarTodo.configure(width=6, height=2, font=("Arial", 20, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.buttonBorrarTodo.place(x=self.filax1, y=self.columnay4)
```

```
self.button0=Button(self.labelFrameVentanaTeclado,text="0",
command=lambda:self.numeroPulsado("0"))
self.button0.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.button0.place(x=self.filax2, y=self.columnay4)
```

```
self.buttonC=Button(self.labelFrameVentanaTeclado, text="\u232B",
command=lambda:self.borrarCaracter())
self.buttonC.configure(width=3, height=1, font=("Arial", self.valorLetraBoton, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.buttonC.place(x=self.filax3, y=self.columnay4)
```

```
def numeroPulsado(self,num):
self.nEntrada = self.focusPrincipal()
self.nEntrada1 = self.focusRegistrar()
```

```

self.nEntrada2 = self.focusRecuperarClave()
if self.nEntrada == 1 or self.nEntrada1 == 3 or self.nEntrada2 == 5:
self.varG_usuarioView.set(self.varG_usuarioView.get() + num)
if self.nEntrada == 2 or self.nEntrada1 == 4 or self.nEntrada2 == 6:
self.varG_claveView.set(self.varG_claveView.get() + num)
if self.nEntrada1 == 7 or self.nEntrada2 == 8:
self.varG_newClaveView.set(self.varG_newClaveView.get() + num)

```

Programación Python Frame Usuario. – Permite iniciar en modo Usuario, invita o administrador comprobando datos del servidor local, su código fuente es la siguiente y su diagrama de flujo se aprecia en la **Figura 50**.

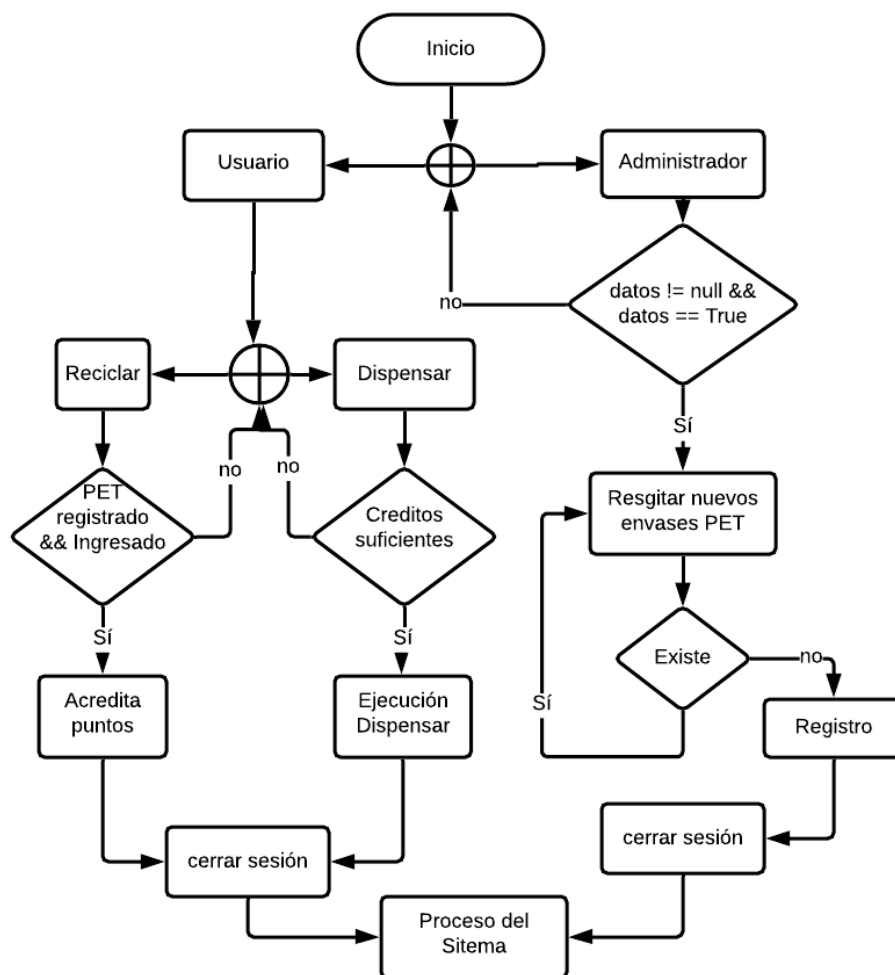


Figura 50: Diagrama Frame Usuario

Elaborado por: El investigador

```

self.labelIngresarBotella=Label(self.frameUsuario, text="Ingresar")
self.labelIngresarBotella.configure(height=1, font=("Arial", 17, "bold"), anchor="w")
self.labelIngresarBotella.place(x=423, y=240)

self.labelDispensarBotella=Label(self.frameUsuario, text="Dispensar")
self.labelDispensarBotella.configure(height=1, font=("Arial", 17, "bold"), anchor="w")
self.labelDispensarBotella.place(x=290, y=240)

self.labelUsuarioLeido=Label(self.frameUsuario,
textvariable=self.varG_usuarioLogin)
self.labelUsuarioLeido.configure(width=10, font=("Arial", 45, "bold"), anchor="w")#
anchor="w"
self.labelUsuarioLeido.place(x=50, y=120)

self.labelCreditoLeido=Label(self.frameUsuario,
textvariable=self.varG_creditoUsuario)
self.labelCreditoLeido.configure(width=5, font=("Arial", 35, "bold"))
self.labelCreditoLeido.place(x=370, y=180)

self.labelMensajeUsuario=Label(self.frameUsuario, text="Se encuentra en Modo
USUARIO")
self.labelMensajeUsuario.configure(height=3, font=("Arial", 20))

self.labelMensajeInvitado=Label(self.frameUsuario, text="Se encuentra en Modo
INVITADO \n no se acreditaran los créditos a tu cuenta")
self.labelMensajeInvitado.configure(height=3, font=("Arial", 20))

self.labelImagenMoneda=Label(self.frameUsuario, image=self.imagenCREDITO)
self.labelImagenMoneda.place(x=430, y=125)

self.labelImagenOpenBot=Label(self.frameUsuario, image=self.imagenOPENBOT)
self.labelImagenOpenBot.place(x=60,y=220)

```

```
self.labelImagenBarCode=Label(self.frameUsuario, image=self.imagenBARCODE)
self.labelImagenBarCode.place(x=30,y=450)
```

```
self.labelCodigoBarrasLeidoUsuario=Label(self.frameUsuario,
textvariable=self.varG_CodigoBarras)
self.labelCodigoBarrasLeidoUsuario.configure(width=14, font=("Arial", 30))#,
state="readonly")
self.labelCodigoBarrasLeidoUsuario.place(x=180, y=450)
```

```
self.buttonIngresarBotella=Button(self.frameUsuario,
command=lambda:self.ingresarBotellaPET())
self.buttonIngresarBotella.configure(width=84, height=95, font=("Arial", 20, "bold"),
image=self.imagenBOTELLA_VACIA,
cursor="hand2", bd=3, highlightbackground="black", relief="ridge",
activebackground="red")
self.buttonIngresarBotella.place(x=423, y=275)
```

```
self.buttonDispensarBotellaAgua=Button(self.frameUsuario,
command=lambda:self.dispensarBotellaAgua())
self.buttonDispensarBotellaAgua.configure(width=84, height=95, font=("Arial", 20,
"bold"), image=self.imagenBOTELLA_AGUA,
cursor="hand2", bd=3, highlightbackground="black", relief="ridge",
activebackground="red")
self.buttonDispensarBotellaAgua.place(x=300, y=275)
```

```
self.buttonCerrarSesion=Button(self.frameUsuario, text="Cerrar Sesión",
command=lambda:self.activarVentanaPrincipal())
self.buttonCerrarSesion.configure(width=13, height=1, font=("Arial", 20, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.buttonCerrarSesion.place(x=295, y=390)
```

```
def ingresarBotellaPET(self):
self.verificaCredito=self.varG_creditoUsuario.get()
self.verificaCredito=float(self.verificaCredito)
self.contadorBotella=self.conexion.leerCantidadBotellas()
```



```

self.varG_contadorBotellas.set(str(self.contadorBotella[0][0]))
self.contadorBotellas=int(self.varG_contadorBotellas.get())

self.botellasEnLista=self.conexion.leerCodigosBotellas()
self.nivelContenedor=self.datosGPIO.leerNivelContenedor()
self.datoUpdate=0
self.noBotella=False

if self.nivelContenedor != 2:

self.datosGPIO.activarLedScanner()
self.verificaCodigoBotella=self.codigoimagen.leerBarCodeImagen()
self.datosGPIO.desactivarLedScanner()
self.comparadorCodigoBotella=(self.verificaCodigoBotella,)
self.varG_CodigoBarras.set(str(self.verificaCodigoBotella))

for x in self.botellasEnLista:
if x == self.comparadorCodigoBotella:
self.datosGPIO.abrirIngresoBotella()
self.datosGPIO.tiempoEspera(10)

self.obstaculo=self.datosGPIO.leerObstaculoIngreso()
while self.obstaculo == False:
self.obstaculo=self.datosGPIO.leerObstaculoIngreso()
self.datosGPIO.cerrarIngresoBotella()
self.datosGPIO.tiempoEspera(5)
self.datosGPIO.verificarBotella()
self.noBotella=self.datosGPIO.leerBotellaIngresada()

if self.noBotella == True:
self.datosGPIO.comprimirBotella()
self.datosGPIO.desactivarComprimirBotella()

```

```

self.datosGPIO.abrirContenedor()
self.datosGPIO.nivelarBotellasContenedor()
self.datosGPIO.cerrarContenedor()
self.verificaCredito=round(self.verificaCredito+0.02,3)
self.conexion.updateCreditoUsuario(self.verificaCredito,
self.varG_usuarioLogin.get())
self.contadorBotellas=self.contadorBotellas + 1
self.conexion.updateContadorBotellas(self.contadorBotellas)
self.varG_contadorBotellas.set(str(self.contadorBotellas))
self.varG_creditoUsuario.set(str(self.verificaCredito))
self.datoUpdate=1
else:
self.datosGPIO.desactivarComprimirBotella()
mb.showinfo("Información", "No ingreso Ninguna Botella")
self.datoUpdate=1

if self.datoUpdate == 0:
self.varG_creditoUsuario.set(self.varG_creditoUsuario.get())
mb.showinfo("Información", "No es Botella PET")
self.datoUpdate=0
else:
mb.showinfo("Información", "Contenedor lleno")

def dispensarIncentivo(self):
self.verificaCredito=self.varG_creditoUsuario.get()
self.verificaCredito=float(self.verificaCredito)
if self.verificaCredito >= 0.5:
self.verificaCredito=round(self.verificaCredito-0.5,3)
self.verificaDis=self.datosGPIO.dispensarBotella()
if self.verificaDis == True:
self.conexion.updateCreditoUsuario(self.verificaCredito,
self.varG_usuarioLogin.get())

```

```

self.varG_creditoUsuario.set(str(self.consultarCredito(self.varG_usuarioLogin.get()))
)
else:
mb.showinfo("Información", "No disponible")
else:
self.varG_creditoUsuario.set(self.varG_creditoUsuario.get())
mb.showinfo("Información", "Creditos Insuficientes")

```

Programación Python Frame New Usuario. – Permite registrar un nuevo usuario en el sistema, evaluando si no se encuentra actualmente registrado, su código fuente es la siguiente y su diagrama de flujo se aprecia en la **Figura 51**.

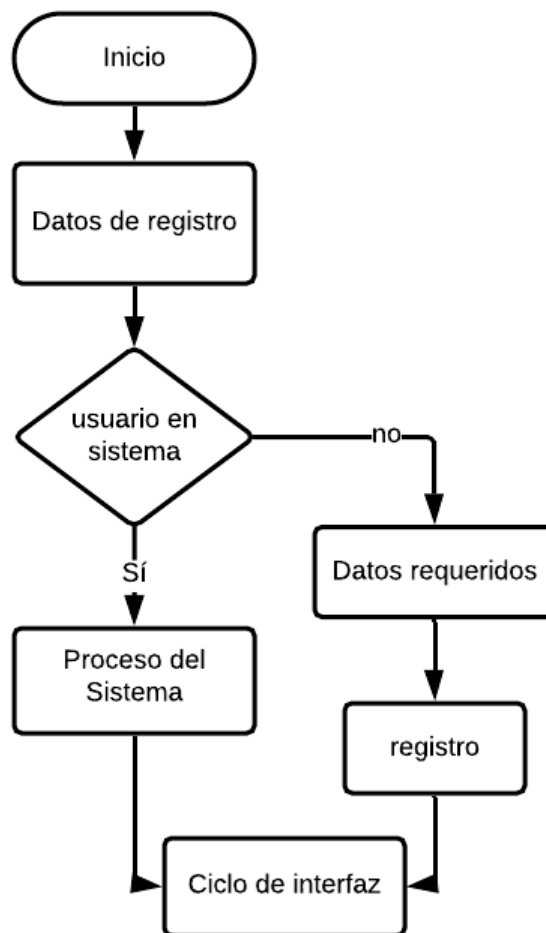


Figura 51: Diagrama Frame Nuevo usuario

Elaborado por: El investigador

```

self.labelEtiquetaLogin=Label(self.frameCrearUsuario, text="Registrarse")
self.labelEtiquetaLogin.configure(width=25, height=2,font=("Arial", 30, "bold"),
bg="#31FA6E", fg="white")
self.labelEtiquetaLogin.place(x=0, y=0)

self.labelImagenLoggin=Label(self.frameCrearUsuario, image=self.imagenLOGGIN)
self.labelImagenLoggin.place(x=40, y=100)

self.labelImagenPassword=Label(self.frameCrearUsuario,
image=self.imagenPASSWORD)
self.labelImagenPassword.place(x=40, y=200)

self.labelUsuario=Label(self.frameCrearUsuario, text="Usuario* ")
self.labelUsuario.configure(height=1, font=("Arial", 20, "bold"))
self.labelUsuario.place(x=80, y=100)
self.labelOpenBot=Label(self.frameCrearUsuario, text="OpenBot")
self.labelOpenBot.configure(height=1, font=("Arial", 25, "bold"), fg="green")
self.labelOpenBot.place(x=365, y=120)

self.labelClave=Label(self.frameCrearUsuario, text="Clave* ")
self.labelClave.configure( height=1, font=("Arial", 20, "bold"))
self.labelClave.place(x=80, y=200)
self.labelImagenRobot=Label(self.frameCrearUsuario, image=self.imagenROBOT)
self.labelImagenRobot.place(x=360, y=170)

self.labelPreguntaS1=Label(self.frameCrearUsuario, text="Preguntas de
Seguridad")
self.labelPreguntaS1.configure(height=1, font=("Arial", 20, "bold"))
self.labelPreguntaS1.place(x=30, y=370)

self.labelFechaRegistrarUsuario=Label(self.frameCrearUsuario, text="Fecha de
Nacimiento*")
self.labelFechaRegistrarUsuario.configure(height=1, font=("Arial", 16))

```

```

self.labelFechaRegistrarUsuario.place(x=30, y=420)

self.labelMascotaRegistrarUsuario=Label(self.frameCrearUsuario, text="Mascota*")
self.labelMascotaRegistrarUsuario.configure(height=1, font=("Arial", 16))
self.labelMascotaRegistrarUsuario.place(x=30, y=480)

self.labelColorRegistrarUsuario=Label(self.frameCrearUsuario, text="Color*")
self.labelColorRegistrarUsuario.configure(height=1, font=("Arial", 16))
self.labelColorRegistrarUsuario.place(x=230, y=480)

self.entryUsuarioRegistrar=ttk.Entry(self.frameCrearUsuario,
textvariable=self.varG_usuarioView)
self.entryUsuarioRegistrar.configure(width=11, font=("Arial", 30), state="readonly")
self.entryUsuarioRegistrar.place(x=60, y=140)

self.entryClaveRegistrar=ttk.Entry(self.frameCrearUsuario,
textvariable=self.varG_claveView)
self.entryClaveRegistrar.configure(show="*", width=11, font=("Arial", 30),
state="readonly")
self.entryClaveRegistrar.place(x=60, y=240)

self.entryNewClaveRegistrar=ttk.Entry(self.frameCrearUsuario,
textvariable=self.varG_newClaveView)
self.entryNewClaveRegistrar.configure(show="*", width=11, font=("Arial", 30),
state="readonly")
self.entryNewClaveRegistrar.place(x=60, y=305)

self.buttonAceptarRegistrar=Button(self.frameCrearUsuario, text="Aceptar",
command=lambda:self.registrarNuevoUsuario())
self.buttonAceptarRegistrar.configure(width=7, height=1, font=("Arial", 14, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
self.buttonAceptarRegistrar.place(x=30, y=535)

```

```
self.buttonSalirRegistrarUsuario=Button(self.frameCrearUsuario, text="Salir",
command=lambda:self.activarVentanaPrincipal())
```

```
self.buttonSalirRegistrarUsuario.configure(width=8, height=1, font=("Arial", 14,
"bold"),
```

```
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")
```

```
self.buttonSalirRegistrarUsuario.place(x=410, y=535)
```

```
self.buttonSeleccionMascotaRegistrar=Button(self.frameCrearUsuario,
image=self.imagenMASCOTA, command=lambda:self.activarFrameMascotas())
```

```
self.buttonSeleccionMascotaRegistrar.configure(width=55, height=43,
cursor="hand2", bd=1, highlightbackground="black", relief="ridge",
```

```
activebackground="red")
```

```
self.buttonSeleccionMascotaRegistrar.place(x=150, y=470)
```

```
self.buttonSeleccionColorRegistrar=Button(self.frameCrearUsuario, text="Color",
command=lambda:self.activarFrameColores())
```

```
self.buttonSeleccionColorRegistrar.configure(width=4, height=2, cursor="hand2",
bd=1, relief="ridge", activebackground="red",
```

```
font=("Arial", 12),highlightbackground="black")
```

```
self.buttonSeleccionColorRegistrar.place(x=310, y=468)
```

```
self.buttonFechaRegistrar=Button(self.frameCrearUsuario, text="0000-00-00",
command=lambda:self.activarFrameCalendario())
```

```
self.buttonFechaRegistrar.configure(width=10, height=1, font=("Arial", 14),
```

```
cursor="hand2", bd=1, highlightbackground="black", relief="ridge")
```

```
self.buttonFechaRegistrar.place(x=280, y=416)
```

```
self.frameMascotasCrearUsuario=Frame(self.frameCrearUsuario, width=250,
height=330, bd=4, relief="ridge")
```

```
self.labelFrameMascotasCrearUsuario=LabelFrame(self.frameMascotasCrearUsuar
io, text="Mascotas", bd=4, font=("Arial", 14),
```

```
width=220, height=300, relief="ridge", highlightbackground="black")
```

```
self.labelFrameMascotasCrearUsuario.place(x=10, y=10)
```

```
self.buttonPerro=Button(self.labelFrameMascotasCrearUsuario,
width=self.dimencionImagenButton,
command=lambda:self.mascotaPulsado(self.listaImágenes[1]),
```

```
height=self.dimencionImagenButton, image=self.imagenPERRO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonPerro.place(x=10, y=10)
```

```
self.buttonGato=Button(self.labelFrameMascotasCrearUsuario,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[2]),
```

```
height=self.dimencionImagenButton, image=self.imagenGATO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonGato.place(x=75, y=10)
```

```
self.buttonHamster=Button(self.labelFrameMascotasCrearUsuario,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[3]),
```

```
height=self.dimencionImagenButton, image=self.imagenHAMSTER,  
cursor="hand2", bd=3, relief="ridge", activebackground="red")
```

```
self.buttonHamster.place(x=140, y=10)
```

```
self.buttonConejo=Button(self.labelFrameMascotasCrearUsuario,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[4]),
```

```
height=self.dimencionImagenButton, image=self.imagenCONEJO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonConejo.place(x=10, y=75)
```

```
self.buttonAves=Button(self.labelFrameMascotasCrearUsuario,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[5]),
```

```
height=self.dimencionImagenButton, image=self.imagenAVES, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonAves.place(x=75, y=75)
```

```
self.buttonPez=Button(self.labelFrameMascotasCrearUsuario,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[6]),
```

```
height=self.dimencionImagenButton, image=self.imagenPEZ, cursor="hand2", bd=3,  
relief="ridge", activebackground="red")
```

```
self.buttonPez.place(x=140, y=75)
```

```
self.buttonCerdo=Button(self.labelFrameMascotasCrearUsuario,
width=self.dimencionImagenButton,
command=lambda:self.mascotaPulsado(self.listaImágenes[7]),
height=self.dimencionImagenButton, image=self.imagenCERDO, cursor="hand2",
bd=3, relief="ridge", activebackground="red")
self.buttonCerdo.place(x=10, y=140)
```

```
self.buttonVaca=Button(self.labelFrameMascotasCrearUsuario,
width=self.dimencionImagenButton,
command=lambda:self.mascotaPulsado(self.listaImágenes[8]),
height=self.dimencionImagenButton, image=self.imagenVACA, cursor="hand2",
bd=3, relief="ridge", activebackground="red")
self.buttonVaca.place(x=75, y=140)
```

```
self.buttonCaballo=Button(self.labelFrameMascotasCrearUsuario,
width=self.dimencionImagenButton,
command=lambda:self.mascotaPulsado(self.listaImágenes[9]),
height=self.dimencionImagenButton, image=self.imagenCABALLO, cursor="hand2",
bd=3, relief="ridge", activebackground="red")
self.buttonCaballo.place(x=140, y=140)
```

```
self.buttonOveja=Button(self.labelFrameMascotasCrearUsuario,
width=self.dimencionImagenButton,
command=lambda:self.mascotaPulsado(self.listaImágenes[10]),
height=self.dimencionImagenButton, image=self.imagenOVEJA, cursor="hand2",
bd=3, relief="ridge", activebackground="red")
self.buttonOveja.place(x=10, y=205)
```

```
self.buttonOso=Button(self.labelFrameMascotasCrearUsuario,
width=self.dimencionImagenButton,
command=lambda:self.mascotaPulsado(self.listaImágenes[11]),
height=self.dimencionImagenButton, image=self.imagenOSO, cursor="hand2",
bd=3, relief="ridge", activebackground="red")
self.buttonOso.place(x=75, y=205)
```

```
self.buttonTigre=Button(self.labelFrameMascotasCrearUsuario,
width=self.dimencionImagenButton,
command=lambda:self.mascotaPulsado(self.listaImágenes[12]),
```



```
height=self.dimencionImagenButton, image=self.imagenTIGRE, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonTigre.place(x=140, y=205)
```

```
self.frameColoresCrearUsuario=Frame(self.frameCrearUsuario, width=275,  
height=310, bd=4, relief="ridge")
```

```
self.labelFrameColoresCrearUsuario=LabelFrame(self.frameColoresCrearUsuario,  
text="Colores", bd=4, font=("Arial", 14),
```

```
width=245, height=280, relief="ridge", highlightbackground="black")
```

```
self.labelFrameColoresCrearUsuario.place(x=10, y=10)
```

```
self.buttonBlanco=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[0]),
```

```
bg=self.coloresMostrarSet[0], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonBlanco.place(x=10, y=10)
```

```
self.buttonNegro=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[1]),
```

```
bg=self.coloresMostrarSet[1], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonNegro.place(x=85, y=10)
```

```
self.buttonAmarillo=Button(self.labelFrameColoresCrearUsuario, width=4,  
height=2,command=lambda:self.colorPulsado(self.listaColores[2]),
```

```
bg=self.coloresMostrarSet[2], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonAmarillo.place(x=160, y=10)
```

```
self.buttonAzul=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[3]),
```

```
bg=self.coloresMostrarSet[3], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonAzul.place(x=10, y=70)
```

```

self.buttonRojo=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[4]),
bg=self.coloresMostrarSet[4], cursor="hand2", bd=3, activebackground="red",
relief="ridge")
self.buttonRojo.place(x=85, y=70)

self.buttonRosado=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[5]),
bg=self.coloresMostrarSet[5], cursor="hand2", bd=3, activebackground="red",
relief="ridge")
self.buttonRosado.place(x=160, y=70)

self.buttonVerde=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[6]),
bg=self.coloresMostrarSet[6], cursor="hand2", bd=3, activebackground="red",
relief="ridge")
self.buttonVerde.place(x=10, y=130)

self.buttonGris=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[7]),
bg=self.coloresMostrarSet[7], cursor="hand2", bd=3, activebackground="red",
relief="ridge")
self.buttonGris.place(x=85, y=130)

self.buttonCafe=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[8]),
bg=self.coloresMostrarSet[8], cursor="hand2", bd=3, activebackground="red",
relief="ridge")
self.buttonCafe.place(x=160, y=130)

self.buttonFucsia=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[9]),
bg=self.coloresMostrarSet[9], cursor="hand2", bd=3, activebackground="red",
relief="ridge")
self.buttonFucsia.place(x=10, y=190)

```

```
self.buttonCeleste=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[10]),
```

```
bg=self.coloresMostrarSet[10], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonCeleste.place(x=85, y=190)
```

```
self.buttonNaranja=Button(self.labelFrameColoresCrearUsuario, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[11]),
```

```
bg=self.coloresMostrarSet[11], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonNaranja.place(x=160, y=190)
```

```
self.frameCalendarioCrearUsuario=Frame(self.frameCrearUsuario, width=378,  
height=305, bd=4, relief="ridge", bg="#D2FABE")
```

```
self.day = int(time.strftime("%d"))
```

```
self.month = int(time.strftime("%m"))
```

```
self.year = int(time.strftime("%Y"))
```

```
self.calendarFechaCrearUsuario = Calendar(self.frameCalendarioCrearUsuario,  
selectmode='day', locale='es_EC.utf8',
```

```
cursor="hand2", year=self.year, month=self.month, day=self.day )
```

```
self.calendarFechaCrearUsuario.configure(font=("Arial", 15))
```

```
self.calendarFechaCrearUsuario.place(x=0, y=0)
```

```
self.buttonOkCalendar=Button(self.frameCalendarioCrearUsuario, text="Aceptar",  
command=lambda:self.obtenerFecha())
```

```
self.buttonOkCalendar.configure(width=10, height=1, font=("Arial", 12, "bold"),
```

```
cursor="hand2", bd=3, activebackground="red", relief="ridge", bg="#BEDCFA")
```

```
self.buttonOkCalendar.place(x=140, y=260)
```

```
def registrarNuevoUsuario(self):
```

```
self.user=self.entryUsuario.get()
```

```
self.password=self.entryClave.get()
```

```
self.newpassword=self.entryNewClaveRegistrar.get()
```

```
self.mascota=self.varG_MascotaSeleccionado.get()
```

```

self.color=self.varG_ColorSeleccionado.get()
self.fecha_envio=self.varG_FechaSeleccionado.get()

if ((self.user != "") and (self.password != "") and (self.newpassword != "") and
(self.color != "") and (self.mascota != "") and
(self.fecha_envio != "")):

self.longitudUser=len(self.user)
self.longitudPassword=len(self.password)
self.userval=int(self.user)
if (self.longitudUser == 10) and (self.userval in range(10000000,249999999)):
if self.longitudPassword in range(4,6):

if self.password == self.newpassword:
self.verificaUsuarioRegistrado=(self.user,)
self.usuariosEnLista=self.conexion.leerTodoUsuarios()
self.datoUpdate=None
for x in self.usuariosEnLista:
if x == self.verificaUsuarioRegistrado:
self.datoUpdate=1
mb.showinfo("Información", "Usuario se Encuentra Registrado")

if self.datoUpdate != 1:
self.conexion.insertarUsuarios(self.user,self.password, self.mascota, self.color,
self.fecha_envio)
mb.showinfo("Información", "Usuario Registrado con exito")
self.borrarTodo()
self.datoUpdate=None
self.activarVentanaPrincipal()
else:
mb.showinfo("Información", "Las Claves no Coinciden")
else:

```

```
mb.showinfo("Información", "La Clave debe tener entre 4 y 6 dígitos ejemplo: 1234")
```

else:

```
mb.showinfo("Información", "Usuario no valido debe ser su C.I. ejemplo:  
1800000000")
```

else:

```
mb.showinfo("Información", "Campos * Requeridos")
```

Programación Python Frame Cambio Clave. – Permite cambiar la clave si el usuario se encuentra registrado, comprobando datos del servidor local, su código fuente es la siguiente y su diagrama de flujo se aprecia en la **Figura 52**.

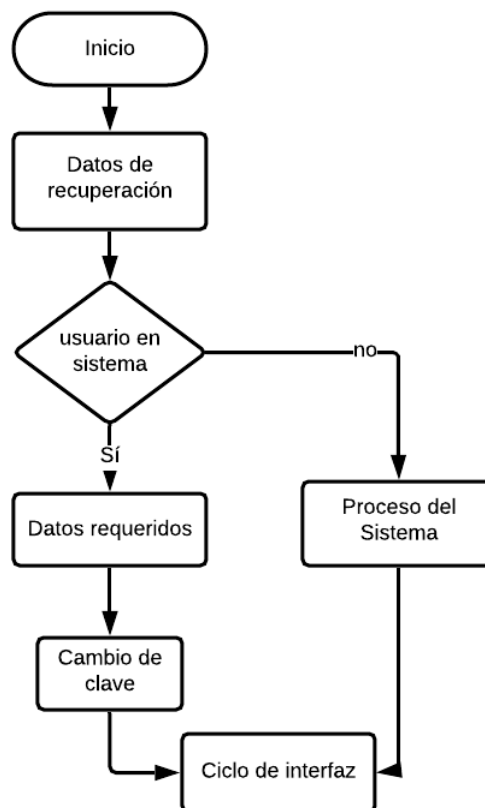


Figura 52: Diagrama Cambio de Clave

Elaborado por: El investigador

```
self.labelEtiquetaLogin=Label(self.frameRecuperarClave, text="Recuperar Clave")
self.labelEtiquetaLogin.configure(width=25, height=2,font=("Arial", 30, "bold"),
bg="#F79B17", fg="white")
self.labelEtiquetaLogin.place(x=0, y=0)
```

```
self.labelImagenLoggin=Label(self.frameRecuperarClave,
image=self.imagenLOGGIN)
```

```
self.labelImagenLoggin.place(x=40, y=100)
```

```
self.labelImagenPassword=Label(self.frameRecuperarClave,
image=self.imagenPASSWORD)
```

```
self.labelImagenPassword.place(x=40, y=200)
```

```
self.labelUsuario=Label(self.frameRecuperarClave, text="Usuario*")
```

```
self.labelUsuario.configure(height=1, font=("Arial", 20, "bold"))
```

```
self.labelUsuario.place(x=80, y=100)
```

```
self.labelClave=Label(self.frameRecuperarClave, text="Nueva Clave*")
```

```
self.labelClave.configure( height=1, font=("Arial", 20, "bold"))
```

```
self.labelClave.place(x=80, y=200)
```

```
self.labelOpenBot=Label(self.frameRecuperarClave, text="OpenBot")
```

```
self.labelOpenBot.configure(height=1, font=("Arial", 25, "bold"), fg="green")
```

```
self.labelOpenBot.place(x=365, y=120)
```

```
self.labelImagenRobot=Label(self.frameRecuperarClave,
image=self.imagenROBOT)
```

```
self.labelImagenRobot.place(x=360, y=170)
```

```
self.labelPreguntaS1=Label(self.frameRecuperarClave, text="Preguntas de
Seguridad")
```

```
self.labelPreguntaS1.configure(height=1, font=("Arial", 20, "bold"))
```

```
self.labelPreguntaS1.place(x=30, y=370)
```

```
self.labelFechaRegistrarUsuario=Label(self.frameRecuperarClave, text="Fecha de
Nacimiento*")
```

```
self.labelFechaRegistrarUsuario.configure(height=1, font=("Arial", 16))
```

```

self.labelFechaRegistrarUsuario.place(x=30, y=420)

self.labelMascotaRegistrarUsuario=Label(self.frameRecuperarClave,
text="Mascota*")
self.labelMascotaRegistrarUsuario.configure(height=1, font=("Arial", 16))
self.labelMascotaRegistrarUsuario.place(x=30, y=480)
self.labelColorRegistrarUsuario=Label(self.frameRecuperarClave, text="Color*")
self.labelColorRegistrarUsuario.configure(height=1, font=("Arial", 16))
self.labelColorRegistrarUsuario.place(x=230, y=480)

self.entryUsuarioRecuperarC=ttk.Entry(self.frameRecuperarClave,
textvariable=self.varG_usuarioView)

self.entryUsuarioRecuperarC.configure(width=11, font=("Arial", 30),
state="readonly")

self.entryUsuarioRecuperarC.place(x=60, y=140)

self.entryClaveRecuperarC=ttk.Entry(self.frameRecuperarClave,
textvariable=self.varG_claveView)

self.entryClaveRecuperarC.configure(show="*", width=11, font=("Arial", 30),
state="readonly")

self.entryClaveRecuperarC.place(x=60, y=240)

self.entryNewClaveRecuperarC=ttk.Entry(self.frameRecuperarClave,
textvariable=self.varG_newClaveView)

self.entryNewClaveRecuperarC.configure(show="*", width=11, font=("Arial", 30),
state="readonly")

self.entryNewClaveRecuperarC.place(x=60, y=305)

self.buttonAceptar=Button(self.frameRecuperarClave,text="Aceptar",
command=lambda:self.actualizarClaveUsuario())

self.buttonAceptar.configure(width=7, height=1, font=("Arial", 14, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.buttonAceptar.place(x=30, y=535)

self.buttonSalirRecuperarC=Button(self.frameRecuperarClave, text="Salir",
command=lambda:self.activarVentanaPrincipal())

self.buttonSalirRecuperarC.configure(width=8, height=1, font=("Arial", 14, "bold"),
cursor="hand2", bd=3, highlightbackground="black", relief="ridge")

self.buttonSalirRecuperarC.place(x=410, y=535)

```

```
self.buttonSeleccionMascotaRecuperarC=Button(self.frameRecuperarClave,  
image=self.imagenMASCOTA, command=lambda:self.activarFrameMascotas())
```

```
self.buttonSeleccionMascotaRecuperarC.configure(width=55, height=43,  
cursor="hand2", bd=1, highlightbackground="black", relief="ridge",  
activebackground="red")
```

```
self.buttonSeleccionMascotaRecuperarC.place(x=150, y=470)
```

```
self.buttonSeleccionColorRecuperarC=Button(self.frameRecuperarClave,  
text="Color",command=lambda:self.activarFrameColores())
```

```
self.buttonSeleccionColorRecuperarC.configure(width=4, height=2, cursor="hand2",  
bd=1, relief="ridge", activebackground="red",
```

```
font=("Arial", 12),highlightbackground="black")
```

```
self.buttonSeleccionColorRecuperarC.place(x=310, y=468)
```

```
self.buttonFechaRecuperarC=Button(self.frameRecuperarClave, text="0000-00-00",  
command=lambda:self.activarFrameCalendario())
```

```
self.buttonFechaRecuperarC.configure(width=10, height=1, font=("Arial", 14),  
cursor="hand2", bd=1, highlightbackground="black", relief="ridge")
```

```
self.buttonFechaRecuperarC.place(x=280, y=416)
```

```
self.frameMascotasRecuperarC=Frame(self.frameRecuperarClave, width=250,  
height=330, bd=4, relief="ridge")
```

```
self.labelFrameMascotasRecuperarC=LabelFrame(self.frameMascotasRecuperarC,  
text="Mascotas", bd=4, font=("Arial", 14),
```

```
width=220, height=300, relief="ridge", highlightbackground="black")
```

```
self.labelFrameMascotasRecuperarC.place(x=10, y=10)
```

```
self.buttonPerro=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[1]),
```

```
height=self.dimencionImagenButton, image=self.imagenPERRO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonPerro.place(x=10, y=10)
```



```
self.buttonGato=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[2]),  
height=self.dimencionImagenButton, image=self.imagenGATO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonGato.place(x=75, y=10)
```

```
self.buttonHamster=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[3]),  
height=self.dimencionImagenButton, image=self.imagenHAMSTER,  
cursor="hand2", bd=3, relief="ridge", activebackground="red")
```

```
self.buttonHamster.place(x=140, y=10)
```

```
self.buttonConejo=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[4]),  
height=self.dimencionImagenButton, image=self.imagenCONEJO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonConejo.place(x=10, y=75)
```

```
self.buttonAves=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[5]),  
height=self.dimencionImagenButton, image=self.imagenAVES, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonAves.place(x=75, y=75)
```

```
self.buttonPez=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[6]),  
height=self.dimencionImagenButton, image=self.imagenPEZ, cursor="hand2", bd=3,  
relief="ridge", activebackground="red")
```

```
self.buttonPez.place(x=140, y=75)
```

```
self.buttonCerdo=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImágenes[7]),
```

```
height=self.dimencionImagenButton, image=self.imagenCERDO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonCerdo.place(x=10, y=140)
```

```
self.buttonVaca=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImagenes[8]),
```

```
height=self.dimencionImagenButton, image=self.imagenVACA, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonVaca.place(x=75, y=140)
```

```
self.buttonCaballo=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImagenes[9]),
```

```
height=self.dimencionImagenButton, image=self.imagenCABALLO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonCaballo.place(x=140, y=140)
```

```
self.buttonOveja=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImagenes[10]),
```

```
height=self.dimencionImagenButton, image=self.imagenOVEJA, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonOveja.place(x=10, y=205)
```

```
self.buttonOso=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImagenes[11]),
```

```
height=self.dimencionImagenButton, image=self.imagenOSO, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonOso.place(x=75, y=205)
```

```
self.buttonTigre=Button(self.labelFrameMascotasRecuperarC,  
width=self.dimencionImagenButton,  
command=lambda:self.mascotaPulsado(self.listaImagenes[12]),
```

```
height=self.dimencionImagenButton, image=self.imagenTIGRE, cursor="hand2",  
bd=3, relief="ridge", activebackground="red")
```

```
self.buttonTigre.place(x=140, y=205)
```

```

self.frameColoresRecuperarC=Frame(self.frameRecuperarClave, width=275,
height=310, bd=4, relief="ridge")

self.labelFrameColoresRecuperarC=LabelFrame(self.frameColoresRecuperarC,
text="Colores", bd=4, font=("Arial", 14),
width=245, height=280, relief="ridge", highlightbackground="black")

self.labelFrameColoresRecuperarC.place(x=10, y=10)

self.buttonBlanco=Button(self.labelFrameColoresRecuperarC, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[0]),
bg=self.coloresMostrarSet[0], cursor="hand2", bd=3, activebackground="red",
relief="ridge")

self.buttonBlanco.place(x=10, y=10)

self.buttonNegro=Button(self.labelFrameColoresRecuperarC, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[1]),
bg=self.coloresMostrarSet[1], cursor="hand2", bd=3, activebackground="red",
relief="ridge")

self.buttonNegro.place(x=85, y=10)

self.buttonAmarillo=Button(self.labelFrameColoresRecuperarC, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[2]),
bg=self.coloresMostrarSet[2], cursor="hand2", bd=3, activebackground="red",
relief="ridge")

self.buttonAmarillo.place(x=160, y=10)

self.buttonAzul=Button(self.labelFrameColoresRecuperarC, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[3]),
bg=self.coloresMostrarSet[3], cursor="hand2", bd=3, activebackground="red",
relief="ridge")

self.buttonAzul.place(x=10, y=70)

self.buttonRojo=Button(self.labelFrameColoresRecuperarC, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[4]),
bg=self.coloresMostrarSet[4], cursor="hand2", bd=3, activebackground="red",
relief="ridge")

self.buttonRojo.place(x=85, y=70)

self.buttonRosado=Button(self.labelFrameColoresRecuperarC, width=4, height=2,
command=lambda:self.colorPulsado(self.listaColores[5]),

```

```
bg=self.coloresMostrarSet[5], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonRosado.place(x=160, y=70)
```

```
self.buttonVerde=Button(self.labelFrameColoresRecuperarC, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[6]),
```

```
bg=self.coloresMostrarSet[6], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonVerde.place(x=10, y=130)
```

```
self.buttonGris=Button(self.labelFrameColoresRecuperarC, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[7]),
```

```
bg=self.coloresMostrarSet[7], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonGris.place(x=85, y=130)
```

```
self.buttonCafe=Button(self.labelFrameColoresRecuperarC, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[8]),
```

```
bg=self.coloresMostrarSet[8], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonCafe.place(x=160, y=130)
```

```
self.buttonFucsia=Button(self.labelFrameColoresRecuperarC, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[9]),
```

```
bg=self.coloresMostrarSet[9], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonFucsia.place(x=10, y=190)
```

```
self.buttonCeleste=Button(self.labelFrameColoresRecuperarC, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[10]),
```

```
bg=self.coloresMostrarSet[10], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```
self.buttonCeleste.place(x=85, y=190)
```

```
self.buttonNaranja=Button(self.labelFrameColoresRecuperarC, width=4, height=2,  
command=lambda:self.colorPulsado(self.listaColores[11]),
```

```
bg=self.coloresMostrarSet[11], cursor="hand2", bd=3, activebackground="red",  
relief="ridge")
```

```

self.buttonNaranja.place(x=160, y=190)

self.day = int(time.strftime("%d"))
self.month = int(time.strftime("%m"))
self.year = int(time.strftime("%Y"))

self.frameCalendarioRecuperarC=Frame(self.frameRecuperarClave, width=378,
height=305, bd=4, relief="ridge", bg="#D2FABE")

self.calendarFechaRecuperarC = Calendar(self.frameCalendarioRecuperarC,
selectmode='day', locale='es_EC.utf8',
cursor="hand2", year=self.year, month=self.month, day=self.day)

self.calendarFechaRecuperarC.configure(font=("Arial", 15))

self.calendarFechaRecuperarC.place(x=0, y=0)

self.buttonOkCalendar=Button(self.frameCalendarioRecuperarC, text="Aceptar",
command=lambda:self.obtenerFecha())

self.buttonOkCalendar.configure(width=10, height=1, font=("Arial", 12, "bold"),
cursor="hand2", bd=3, activebackground="red", relief="ridge", bg="#BEDCFA")

self.buttonOkCalendar.place(x=140, y=260)

actualizarClaveUsuario(self):
self.user=self.entryUsuario.get()
self.password=self.entryClave.get()
self.newpassword=self.entryNewClaveRecuperarC.get()
self.mascota=self.varG_MascotaSeleccionado.get()
self.color=self.varG_ColorSeleccionado.get()
self.fecha_envio=self.varG_FechaSeleccionado.get()

if ((self.user != "") and (self.password != "") and (self.newpassword != "") and
(self.color != "") and (self.mascota != "") and
(self.fecha_envio != "")):

self.longitudUser=len(self.user)
self.longitudPassword=len(self.password)

```

```

self.verificaUsuarioRegistrado=(self.user,)
self.usuariosEnLista=self.conexion.leerTodoUsuarios()
self.datoUpdate=None

for x in self.usuariosEnLista:
if x == self.verificaUsuarioRegistrado:
self.datoUpdate=1

if self.datoUpdate == 1:
self.datosUsuariosLeidos=self.conexion.leerDatosUsuario(self.user)
self.mascotaLeido=self.datosUsuariosLeidos[0][0]
self.colorLeido=self.datosUsuariosLeidos[0][1]
self.fechaLeido=self.datosUsuariosLeidos[0][2]
if self.longitudPassword in range(4,6):
if self.password == self.newpassword:
if (str(self.mascotaLeido)==str(self.mascota) and str(self.colorLeido)==str(self.color)
and str(self.fechaLeido)==str(self.fecha_envio)):
self.conexion.actualizarClaveUsuario(self.password,self.user)
print(self.mascotaLeido,self.colorLeido,self.fechaLeido)
mb.showinfo("Información", "Clave Actualizado con Exito de USUARIO:
{a}".format(a=self.user))
self.borrarTodo()
self.datoUpdate=None
self.activarVentanaPrincipal()
else:
mb.showinfo("Información", "Datos Incorrectos")
else:
mb.showinfo("Información", "Las Claves no Coinciden")
else:
mb.showinfo("Información", "La Clave debe tener entre 4 y 6 dígitos ejemplo: 1234")
else:
mb.showinfo("Información", "Usuario No Registrado")

```

else:

```
mb.showinfo("Información", "Campos * Requeridos")
```