



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Intercambiador automático de muestreo
para una trampa de succión para insectos
de alta movilidad con un enfoque de
Internet de las Cosas (IoT)**

TESIS

Que para obtener el título de

INGENIERA MECATRÓNICA

P R E S E N T A

Aline Arlette Lozano Moctezuma

DIRECTOR DE TESIS

Dr. Victor Manuel Lomas Barrie



Ciudad Universitaria, Cd. Mx., 2019

Agradecimientos

A Dios, por darme la vida y guiar cada uno de mis pasos en esta etapa.

A mi mamá y a mi papá, quienes desde pequeña me educaron y me inculcaron los principios y valores que me llevaron hasta aquí. Quienes aplaudieron mis triunfos y cuestionaron mis fracasos y errores, pero sin dejar de creer en mí. Entre muchas cosas, gracias por su paciencia, apoyo y el amor que ambos me brindan. Mamá, por enseñarme a siempre pelear por lo que quiera, aunque parezca imposible; por mostrarme que no hay edad para dejar de divertirse y tener aventuras. Papá, por ser firme, por mostrarme que hay tiempo para relajarse y tiempo para trabajar; que hay que esforzarse y enfocarse para alcanzar los objetivos. Un sinfín de cosas les agradezco, y aunque este espacio no es suficiente para hacerlo, recuerden siempre lo mucho que los Quiero.

A mi hermana Annie y a mi hermano Yarick, que, si bien hay varios años de diferencia, hemos vivido momentos muy buenos, y otros no tanto, pero al final siempre recordando el lazo que nos une. Gracias por acompañarme durante la carrera, por aconsejarme y dejarse asombrar con mis proyectos; por las diversas travesuras, que me recuerdan lo bonito de tenerlos como hermanos. Los Quiero y estaré siempre para ustedes.

A Milton, que llegaste a mi vida de una manera inexplicable, todo el camino universitario lo vivimos juntos: inscripciones, clases, proyectos, desveladas, exámenes, comidas, cursos, entre otras cosas. Te agradezco por ser el impulso que me motiva siempre que lo necesito. Gracias por apoyarme y ayudarme, por aconsejarme y extenderme tu mano para salir adelante juntos, y por todo lo que sin duda no terminaría de listar. Eres muy importante para mí y te agradezco el acompañarme todos estos años. Te AMo infinitAMente Super Mucho MilA.

A bita, que me viste y cuidaste desde pequeña, mucho he aprendido de ti. Gracias por todos estos años de apapachos, por escucharme siempre, y ayudarme siempre que lo necesito.

A bito, por asombrarme siempre con tus historias, gracias por tu cariño, por mostrarme la importancia de aprender, y despertar en mí la sed de conocimiento en diferentes ámbitos.

A mis tías, les agradezco que cada una ha estado para mí en diferentes momentos de mi vida, gracias por todo su apoyo incondicional, sus consejos y su motivación.

A mis tíos, con quienes recuerdo muchas aventuras desde pequeña, gracias por también acompañarme en estos años.

A mis primas y primos.



A mi amiga Kat, con quién tuve el gusto de coincidir en los primeros semestres en la facultad y quién me eleva los ánimos para continuar adelante.

A mis amistades de la secundaria, de la preparatoria, de la universidad, de karate, de la gimnasia y a los que he conocido de otras maneras, gracias porque más de una vez me motivaron para terminar este ciclo.

Al personal del IIMAS, quienes me echaron porras a diario, gracias Margarita.

Por su tiempo, a cada uno de mis sinodales:

Dr. Juan Mario Peña Cabrera

Dr. Victor Manuel Lomas Barrie

Dr. Eduardo Espinosa Ávila

Dra. Tamara Iskra Alcántara Concepción

Mtro. Rubén Valenzuela Montes

Lalo, gracias por tu forma de enseñar tan única, aprendí mucho y mi gusto por la ingeniería fue mayor.

Dra. Tamara, le agradezco por todo lo que me ha enseñado y por su apoyo en esta etapa.

Dr. Victor, gracias por su paciencia, he aprendido mucho en este tiempo.

A la Dra. Claudia Tania Lomas Barrie, del INIFAP, por permitirme formar parte del proyecto.

Al Ing. Luis Rafael Jiménez Ugalde, por demostrarme que, si se quiere, se puede.

A la Facultad de Ingeniería.

A la Universidad Nacional Autónoma de México.



Índice

Introducción	1
Objetivo	3
Objetivos particulares	3
Planteamiento del problema	3
Descripción del contenido	4
Capítulo 1 - Antecedentes	5
Trampa de succión para insectos de alta movilidad	6
Intercambiador automático de muestreo	7
Internet de las Cosas	18
Arquitectura	18
Proceso de desarrollo del prototipo	21
Capítulo 2 - Primera iteración del ciclo	23
Etapa de Planificar	24
Etapa de Hacer	32
Etapa de Verificar	53
Etapa de Actuar	55
Capítulo 3 - Segunda iteración del ciclo	56
Etapa de Planificar	57
Etapa de Hacer	58
Etapa de Verificar	62
Etapa de Actuar	64
Capítulo 4 - Tercera iteración del ciclo	65
Etapa de Planificar	66
Etapa de Hacer	67
Etapa de Verificar	78
Etapa de Actuar	79
Capítulo 5 - Integración final	80
Conexiones del sistema	81
Bibliotecas para los actuadores y sensores	89



MQTT	91
<i>Framework</i>	93
Pruebas finales de operación	95
Conclusiones	96
Trabajo a Futuro	98
Trabajos citados	99
Otras fuentes de consulta	105
ANEXO A: Piezas fotografiadas de la primera iteración	108
ANEXO B: Piezas fotografiadas de la segunda iteración	113
ANEXO C: Piezas fotografiadas de la tercera iteración	114
ANEXO D: Código para los sensores	117
ANEXO E: Código para mover el émbolo	120
ANEXO F: Código para mover el carrusel	123
ANEXO G: Código del cliente en la SBC	128
ANEXO H: Fracción del config.json de Pimatic	131



Índice de Figuras

<i>Figura 1: Características de la trampa tipo Rothamsted. Imagen elaborada a partir de los planos de la trampa instalada en Mocochá.</i>	7
<i>Figura 2: Última versión probada del intercambiador automático de muestreo, antes de comenzar con la presente investigación. Fotografía tomada por personal del INIFAP.</i>	8
<i>Figura 3: Piezas de la versión que falló del intercambiador automático de muestreo. Las letras coinciden con la lista en el texto.</i>	9
<i>Figura 4: Circuito para el ECS1030. Diagrama obtenido en el tutorial de un colaborador de SparkFun Electronics (rwaldron, 2016).</i>	16
<i>Figura 5: IoT en 3 capas. Diagrama elaborado a partir de la información obtenida en (Sethi & Sarangi, 2017) y (Al-Qaseemi, Almulhim, Almulhim, & Rasool, 2016).</i>	19
<i>Figura 6: Elementos de las capas de IoT considerados en esta investigación.</i>	20
<i>Figura 7: Ciclo de Deming. Diagrama elaborado a partir de la información obtenida en (García, 2016).</i>	21
<i>Figura 8: Modelo del inmovilizador del motor.</i>	32
<i>Figura 9: Modelos de las paletas del carrusel. En la izquierda se muestra la pieza anterior y en la derecha la pieza solución.</i>	33
<i>Figura 10: Modelo del engrane para servomotor construido para ser fijado al cople comercial del servomotor.</i>	33
<i>Figura 11: Modelos de la escuadra sujetadora del frasco de recolección. En la izquierda se muestra la pieza anterior y en la derecha la pieza solución.</i>	34
<i>Figura 12: Modelo del soporte para el frasco de recolección.</i>	35
<i>Figura 13: Modelo del cople para motor.</i>	35
<i>Figura 14: Modelos del porta servomotor. En la izquierda se muestra la pieza anterior y en la derecha la pieza solución.</i>	36
<i>Figura 15: Boceto del eje del carrusel.</i>	36
<i>Figura 16: Bocetos de opciones de sujeción del sistema.</i>	37
<i>Figura 17: Modelo de la opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Los orificios son para la introducción de tornillos. Parte frontal.</i>	38
<i>Figura 18: Modelo de la opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Los orificios son para la introducción de tornillos. Parte posterior.</i>	38
<i>Figura 19: Modelo de tapa - sujetador. Esta pieza es la que une al sistema con la opción de sujeción seleccionada.</i>	38
<i>Figura 20: Pieza cilíndrica con boca cónica, unida a la estructura de la trampa.</i>	39
<i>Figura 21: Estados del frasco. A la izquierda se muestra el estado “arriba”; a la derecha el estado “abajo”</i>	39
<i>Figura 22: Boceto de opción para sellar los frascos con tapas a presión.</i>	41
<i>Figura 23: Modelos de las diferentes versiones de las tapas a presión para sellar los frascos.</i>	41
<i>Figura 24: Fotografía de las diferentes versiones de las tapas a presión para sellar los frascos.</i>	41
<i>Figura 25: Bocetos de cople para unir la tapa.</i>	41
<i>Figura 26: Modelo de la tapa con el cople.</i>	42



<i>Figura 27: Modelo del porta - sensores vertical en forma de abrazadera montado sobre la guía - porta motor. Se muestra el modelo, de abajo hacia arriba: parte posterior, parte inferior, parte frontal y parte superior.</i>	<i>43</i>
<i>Figura 28: Boceto del porta - sensores utilizando un riel.</i>	<i>44</i>
<i>Figura 29: Modelo del porta - sensores vertical utilizando un riel montado sobre la guía - porta motor. Se muestra el modelo, de abajo hacia arriba: parte inferior, parte frontal, parte superior y parte lateral.</i>	<i>44</i>
<i>Figura 30: Disipador de calor para SSR. Obtenida de (Alibaba, 2019).</i>	<i>45</i>
<i>Figura 31: Disipador de calor para SSR. Obtenida de (Banggood, 2019).</i>	<i>45</i>
<i>Figura 32: Proceso para fabricar piezas utilizando manufactura aditiva. El archivo STL es “una versión simplificada de un archivo CAD, con menos cantidad de información y con la geometría final resumida, optimizada, reducida al mínimo posible para que no se pierda la geometría y estructura del objeto” (Regidor, 2016). El archivo g es el tipo de archivo necesario para imprimir, contiene las “instrucciones para la máquina que determinan medidas como la velocidad, la dirección y la profundidad a la que debe moverse” (ProyectosCAD, 2018).</i>	<i>51</i>
<i>Figura 33: Piezas auxiliares para imprimir en 3D, elaboradas a partir de plastilina o herramientas forradas con el Masking Tape mostrado.</i>	<i>52</i>
<i>Figura 34: Proceso para fabricar piezas utilizando corte láser. SVG es un “formato gráfico basado en XML para crear archivos vectoriales en 2D, con un lenguaje de marcado por medio de etiquetas” (ATGU, 2014). DXF es un “formato de dibujo desarrollado por Autodesk como un tipo de formato universal para almacenar modelos CAD” (iX Corp, 2018).</i>	<i>52</i>
<i>Figura 35: Intercambiador fijado a una mesa del Laboratorio de Electrónica y Automatización para Industria 4.0, en el IIMAS.</i>	<i>53</i>
<i>Figura 36: Porta - micro - motorreductor.</i>	<i>60</i>
<i>Figura 37: Cople para motor modificado.</i>	<i>61</i>
<i>Figura 38: Intercambiador de la segunda iteración del ciclo, fijado a una mesa del Laboratorio de Electrónica y Automatización para Industria 4.0.</i>	<i>62</i>
<i>Figura 39: Prueba de la opción de sujeción seleccionada en la primera iteración, en la trampa de Mococho.</i>	<i>63</i>
<i>Figura 40: Cople dañado. La parte donde se introduce uno de los tornillos se separó debido a que no soportó los esfuerzos.</i>	<i>63</i>
<i>Figura 41: Piezas en forma de "C" y "T" para limitar el movimiento del cople.</i>	<i>67</i>
<i>Figura 42: Bocetos del cople para la tapa del sellado de frascos.</i>	<i>67</i>
<i>Figura 43: Bocetos de las nuevas opciones de sujeción del sistema.</i>	<i>68</i>
<i>Figura 44: Otro boceto para la nueva opción de sujeción del sistema.</i>	<i>69</i>
<i>Figura 45: Modelo de la nueva opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Parte frontal.</i>	<i>69</i>
<i>Figura 46: Modelo de la nueva opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Parte posterior.</i>	<i>69</i>
<i>Figura 47: Modelo de la tapa - sujetador. Esta pieza es la que une al sistema con la nueva opción de sujeción seleccionada.</i>	<i>69</i>
<i>Figura 48: Ensamble entre la opción de sujeción seleccionada y la tapa - sujetador.</i>	<i>70</i>
<i>Figura 49: Boceto de la tapa deslizante sobre rieles.</i>	<i>70</i>
<i>Figura 50: Modelo de la figura deslizada sobre sus rieles en la tapa.</i>	<i>71</i>



<i>Figura 51: Prototipo de las piezas para sostener los sensores de efecto Hall de tipo lineal, se sostiene de la tapa a diferentes distancias, los sensores se colocan en una barra donde se deslizan vertical y horizontalmente para finalmente ajustarse.</i>	72
<i>Figura 52: Bocetos de las piezas para sostener los sensores de efecto Hall de tipo lineal.</i>	73
<i>Figura 53: Modelo de las piezas para sostener los sensores de efecto Hall de tipo lineal.</i>	73
<i>Figura 54: Pieza para limitar el movimiento de las piezas para sostener los sensores de efecto Hall de tipo lineal. En la izquierda, se muestra sólo la pieza. En la derecha se muestra el modelo de la pieza montada.</i>	73
<i>Figura 55: Modelo de la tapa final.</i>	74
<i>Figura 56: Modelo final del cople para unir la tapa.</i>	74
<i>Figura 57: Guía para alinear el frasco.</i>	74
<i>Figura 58: Primera pieza del cople en forma de caja del micro - motorreductor.</i>	76
<i>Figura 59: Segunda pieza del cople en forma de caja del micro - motorreductor (Tapa).</i>	76
<i>Figura 60: Pieza creada para compensar el error en el diseño del cople en forma de caja con tapa del micro - motorreductor.</i>	77
<i>Figura 61: Intercambiador fijado a perfil de 1 [in], a la vez, el perfil fijado a la mesa del laboratorio en el IIMAS. Se muestra el sistema, de izquierda a derecha: parte frontal y parte posterior.</i>	79
<i>Figura 62: Diagrama de las conexiones en la placa perforada.</i>	83
<i>Figura 63: Placa perforada alambrada.</i>	84
<i>Figura 64: Placa soldada. Se indica el componente que se conecta a cada header polarizado (incluyendo la fuente externa de 9 [V]), las bases con los circuitos integrados y el header de pines para la Raspberry Pi.</i>	84
<i>Figura 65: Conexiones en el conector DE-15. Imagen elaborada a partir del esquema de (Mobius, 2006). Los colores que se indican corresponden al color del cable o alambre de cada componente en el intercambiador.</i>	85
<i>Figura 66: Cable UTP forrado con manguera ranurada, asegurado con cinturones de nylon. En sus extremos se une a conectores DE-15 macho.</i>	85
<i>Figura 67: Conector DE-15 hembra.</i>	86
<i>Figura 68: Conectores Jack de 3.5 [mm] macho. De izquierda a derecha: Relevador, LM35, ECS1030.</i>	86
<i>Figura 69: Conectores Jack de 3.5 [mm] hembra. De izquierda a derecha: ECS1030, LM35, Relevador.</i>	86
<i>Figura 70: Conector DE-9 hembra para sensores. De izquierda a derecha: BMP180, DHT11. Se inhabilitan físicamente dos pines del conector para evitar equivocación al conectar.</i>	87
<i>Figura 71: LED unidos a la caja.</i>	87
<i>Figura 72: Conector DC hembra de 5 [mm].</i>	87
<i>Figura 73: Cable que atraviesa una pared de la caja para alimentar a la SBC.</i>	88
<i>Figura 74: Caja de componentes electrónicos final. A la izquierda, el interior de la caja. A la derecha, vista desde el exterior.</i>	88
<i>Figura 75: Estructura de los topics para suscribirse y publicar. Se consideran cinco niveles donde el topic “trampa” es el primer nivel.</i>	92
<i>Figura 76: Capas de IoT para este proyecto.</i>	94
<i>Figura 77: Aplicación web configurada utilizando Pimatic.</i>	94



Índice de Tablas

<i>Tabla 1: Características de la SBC (Raspberry Pi Zero W). Tabla construida con base en la información en (Raspberry Pi Foundation, 2018a) y (Raspberry Pi Foundation, 2019a).</i>	10
<i>Tabla 2: Características del motor de CA. Tabla construida con base en la información de la etiqueta.</i>	10
<i>Tabla 3: Características del servomotor (Tower Pro MG995). Tabla construida con base en la información en (330 ohms, 2019a) y (Torq Pro & Tower Pro, 2014).</i>	11
<i>Tabla 4: Características del motorreductor de plástico. Tabla construida con base en la información en (330ohms, 2019c).</i>	11
<i>Tabla 5: Características del puente H (L293D). Tabla construida con base en la información en (Texas Instruments, 2016).</i>	12
<i>Tabla 6: Características del BMP180 (GY-68). Tabla construida con base en la información en (Bosch Sensortec, 2011).</i>	12
<i>Tabla 7: Características del DHT11. Tabla construida con base en la información en (D-Robotics UK, 2010).</i>	13
<i>Tabla 8: Características del MCP3008. Tabla construida con base en la información en (Microchip Technology, 2008).</i>	13
<i>Tabla 9: Características del LM35DZ. Tabla construida con base en la información en (Texas Instruments, 2017).</i>	14
<i>Tabla 10: Características del ECS1030. Tabla construida con base en la información en (Echun Electronic Co, s.f.) y (330ohms, 2019b).</i>	15
<i>Tabla 11: Información recopilada del intercambiador automático de muestreo.</i>	24
<i>Tabla 12: Necesidades de los usuarios. Etiquetadas según los tres subsistemas: Electrónico, Mecánico y Software.</i>	26
<i>Tabla 13: Especificaciones del intercambiador. Definidas como métricas a partir de las necesidades de la Tabla 12. Se indican las unidades y valores para cada métrica.</i>	28
<i>Tabla 14: Puntos críticos de la primera iteración del ciclo.</i>	30
<i>Tabla 15: Solución para el punto crítico 1 de la primera iteración del ciclo.</i>	32
<i>Tabla 16: Solución para el punto crítico 2 de la primera iteración del ciclo.</i>	32
<i>Tabla 17: Solución para el punto crítico 3 de la primera iteración del ciclo.</i>	33
<i>Tabla 18: Solución para el punto crítico 4 de la primera iteración del ciclo.</i>	34
<i>Tabla 19: Solución para el punto crítico 5 de la primera iteración del ciclo.</i>	35
<i>Tabla 20: Solución para el punto crítico 6 de la primera iteración del ciclo.</i>	35
<i>Tabla 21: Solución para el punto crítico 7 de la primera iteración del ciclo.</i>	35
<i>Tabla 22: Solución para el punto crítico 8 de la primera iteración del ciclo.</i>	36
<i>Tabla 23: Solución para el punto crítico 9 de la primera iteración del ciclo.</i>	37
<i>Tabla 24: Solución para el punto crítico 10 de la primera iteración del ciclo.</i>	40
<i>Tabla 25: Solución para el punto crítico 11 de la primera iteración del ciclo.</i>	42
<i>Tabla 26: Características del sensor de efecto Hall (U18). Tabla construida con base en la información en (Unisonic Technologies Company, 2012).</i>	42
<i>Tabla 27: Solución para el punto crítico 12 de la primera iteración del ciclo.</i>	43



<i>Tabla 28: Características del relevador de estado sólido (SSR - 40 DA). Tabla construida con base en la información en (FOTEK, s.f.).</i>	45
<i>Tabla 29: Solución para el punto crítico 13 de la primera iteración del ciclo.</i>	46
<i>Tabla 30: Voltaje y corriente de alimentación de los componentes para la primera iteración del ciclo. En la última columna se indica como sería alimentado el componente, por la tarjeta o por una fuente externa. La información de los LED se obtiene en (finsuka, 2017), se utilizan dos resistores debido a que se requieren menos de 3.3 [V] para su encendido. Tabla elaborada con base en las características, descritas con anterioridad, de los componentes.</i>	47
<i>Tabla 31: Solución para el punto crítico 14 de la primera iteración del ciclo.</i>	47
<i>Tabla 32: Solución para el punto crítico 15 de la primera iteración del ciclo.</i>	47
<i>Tabla 33: Solución para el punto crítico 16 de la primera iteración del ciclo.</i>	48
<i>Tabla 34: Solución para el punto crítico 17 de la primera iteración del ciclo.</i>	48
<i>Tabla 35: Solución para el punto crítico 18 de la primera iteración del ciclo.</i>	49
<i>Tabla 36: Protocolos de red. Tabla construida con base en la información de Aron Semle (Semle, 2016).</i>	50
<i>Tabla 37: Solución para el punto crítico 19 de la primera iteración del ciclo.</i>	51
<i>Tabla 38: Puntos críticos de la segunda iteración del ciclo.</i>	57
<i>Tabla 39: Solución para el punto crítico 1 de la segunda iteración del ciclo.</i>	58
<i>Tabla 40: Características del micro - motorreductor. Tabla construida con base en las características que se obtuvieron en (Pololu Corporation, 2019) y (CDMX Electrónica, 2019a). Para conocer los valores de corriente máxima y corriente al arrancar se hicieron diferentes pruebas, colocando voltajes de 6 [V], 9 [V] y 12 [V], variando la carga, e intercalando el movimiento de un lado a otro de acuerdo con lo necesario para el proyecto.</i>	59
<i>Tabla 41: Solución para el punto crítico 2 de la segunda iteración del ciclo.</i>	59
<i>Tabla 42: Voltaje y corriente de alimentación de los componentes para la segunda iteración del ciclo. En la última columna se indica como sería alimentado el componente, por la tarjeta o por una fuente externa. Tabla elaborada con base en las características descritas con anterioridad.</i>	60
<i>Tabla 43: Solución para el punto crítico 3 de la segunda iteración del ciclo.</i>	60
<i>Tabla 44: Solución para el punto crítico 4 de la segunda iteración del ciclo.</i>	61
<i>Tabla 45: Puntos críticos en la tercera iteración.</i>	66
<i>Tabla 46: Solución para el punto crítico 1 de la tercera iteración del ciclo.</i>	67
<i>Tabla 47: Solución para el punto crítico 2 de la tercera iteración del ciclo.</i>	68
<i>Tabla 48: Solución para el punto crítico 3 de la tercera iteración del ciclo.</i>	70
<i>Tabla 49: Solución para el punto crítico 4 de la tercera iteración del ciclo.</i>	71
<i>Tabla 50: Características del micro - motorreductor. Tabla construida con base en las características que se obtuvieron en (Pololu Corporation, 2019) y (CDMX Electrónica, 2019b). Para conocer los valores de corriente máxima y corriente al arrancar se hicieron diferentes pruebas, colocando voltajes de 6 [V], 9 [V] y 12 [V], variando la carga, e intercalando el movimiento de un lado a otro de acuerdo con lo necesario para el proyecto.</i>	72
<i>Tabla 51: Características del sensor de efecto hall 49E. Tabla construida con base en las características que se obtuvieron de su hoja de especificaciones (Yangzhou Positioning Tech). El rango de voltaje en la señal de salida se obtuvo experimentalmente.</i>	72
<i>Tabla 52: Solución para el punto crítico 5 de la tercera iteración del ciclo.</i>	74



Tabla 53: Voltaje y corriente de alimentación de los componentes para la tercera iteración del ciclo. En la última columna se indica como sería alimentado el componente, por la tarjeta o por una fuente externa. Tabla elaborada con base en las características descritas con anterioridad.

..... 75

Tabla 54: Solución para el punto crítico 6 de la tercera iteración del ciclo. 75

Tabla 55: Solución para el punto crítico 7 de la tercera iteración del ciclo. 76

Tabla 56: Forma de conexión de los componentes. Tabla construida con base en la información en las hojas de especificaciones de cada componente. 83



Introducción

Según la definición de la Real Academia Española (Real Academia Española, 2018) se conoce como plaga a la “aparición masiva y repentina de seres vivos de la misma especie que causan graves daños a poblaciones animales o vegetales”.

Para evitar los daños provocados por una plaga (ecológicos y/o económicos), es necesario detectar el crecimiento repentino e intempestivo de la población. Lo anterior se logra monitoreando la zona geográfica de interés, para que se puedan tomar las medidas necesarias para proteger la zona oportunamente, y en su caso, controlar la plaga.

En el año 2017, investigadores del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) y del Instituto Nacional de Investigaciones Forestales Agrícolas y Pecuarias (INIFAP) establecieron un convenio para instalar una trampa de succión tipo Rothamsted, en el Campo Experimental de Mocochoá, Yucatán, como iniciativa para ayudar a los agricultores a proteger sus cultivos de las plagas conformadas por insectos de alta movilidad¹, en específico aquellos que se puedan encontrar a una altura mayor a los 12 metros.

En Yucatán, la temperatura varía entre 16 [°C] y 36 [°C], además, 85.5% de la superficie presenta un clima cálido subhúmedo, lo que permite el desarrollo de diferentes tipos de cultivo como el henequén, maíz, frijol, entre otros. Por lo anterior, 22% de su superficie está destinado a la agricultura (INEGI, 2015).

Con ayuda de la trampa colocada, los investigadores del INIFAP comenzaron a tomar muestras. Además, se incorporaron diversas versiones de un sistema que intercambiara automáticamente los frascos de recolección de muestras, para conseguir más de una, cada vez que se acudiera a la trampa. No obstante, las versiones del intercambiador tuvieron problemas para continuar trabajando, en este escrito se describirá la última versión que estuvo en funcionamiento.

Con el interés y propósito de incorporar nuevamente un intercambiador automático de muestreo, se comenzó a trabajar esta investigación a partir de la última versión que falló, tomando las partes funcionales y sustituyendo las que no lo fueran. Además, en lugar de desarrollar la investigación en Yucatán, se realizó en el Laboratorio de Electrónica y Automatización para Industria 4.0, en el IIMAS, debido a una mayor accesibilidad al equipo, material y software necesario.

Cabe mencionar que, se pretende que se pueda instalar un sistema de alertamiento temprano a lo largo de las costas del país, este sistema estaría conformado por trampas de succión como la descrita más adelante, así como otros tipos de trampas, estaciones

¹ Insectos de alta movilidad: se refiere a las especies que son capaces de trasladarse grandes distancias para ir de un territorio a otro de manera intencional o accidental.



meteorológicas y sistemas automáticos para el monitoreo. Por ello, cada trampa y estación meteorológica sería una IoT (*Internet of Things*, en español: Internet de las Cosas) eventualmente; para que con *cloud computing* se acceda y se almacene la información de cada trampa y del intercambiador remotamente.

En las siguientes líneas de este escrito se describen los detalles de cada paso que se empleó, consideraciones, justificación de cambios, soluciones a problemas, entre otros; que llevaron a la obtención de una nueva versión del intercambiador automático de muestreo para ser instalado en la trampa, y monitoreado a través de una aplicación web.



Objetivo

Diseñar, desarrollar e implementar un intercambiador automático de muestreo para una trampa de succión para insectos de alta movilidad, utilizando tecnologías de Internet de las Cosas (IoT), permitiendo su control y monitoreo.

Objetivos particulares

- Detectar los puntos críticos en la versión anterior del intercambiador automático de muestreo.
- Proponer un diseño del intercambiador que corrija y/o satisfaga los puntos críticos detectados, considerando el aspecto económico.
- Definir los elementos de la arquitectura de IoT aplicables e implementarlos.
- Comprobar el control del intercambiador remotamente, así como la obtención de los datos de los sensores instalados en el sistema.

Planteamiento del problema

El monitoreo de una plaga es esencial para prevenir los daños ecológicos y económicos que esta puede generar en zonas de cultivo. Para lograrlo se instalan trampas para tomar muestras cada determinado tiempo, que después serán estudiadas, según los investigadores del INIFAP.

En el Campo Experimental de Mocochá del INIFAP se instaló una trampa de succión tipo Rothamsted para monitorear insectos de alta movilidad que se encuentren a una altura mayor a los 12 metros.

Dentro de la trampa se encontraba un sistema para obtener muestras de manera automática, el cual dejó de funcionar por varios problemas mecánicos y electrónicos.

Para este proyecto se realizó una versión funcional, con mejoras incrementales, del intercambiador automático de muestreo, en el Laboratorio de Electrónica y Automatización para Industria 4.0, adecuándolo a los retos y necesidades de una IoT.

Cabe indicar, que se trata de un prototipo, debido a los aspectos que no se consideraron y que forman parte del Trabajo a futuro.



Descripción del contenido

Este escrito se encuentra estructurado en cinco capítulos. Se explican en las siguientes líneas.

En el capítulo 1 se describen los temas relacionados directamente con la tesis, como son:

- Trampa de succión para insectos de alta movilidad: se define qué es, su utilidad y como se compone la instalada en el Campo Experimental de Mocochá. Por otra parte, se explica la versión anterior del intercambiador automático de muestreo, es decir, la última versión que estuvo en funcionamiento y falló. Se incluye la descripción de sus elementos físicos y electrónicos, algunos de estos se retomaron para la nueva versión.
- Internet de las Cosas: se define y se especifican los elementos que conforman su arquitectura. Además, se indican los que fueron considerados para esta implementación.
- Proceso de desarrollo del prototipo: se explica el proceso que se efectuó para desarrollar el prototipo, describiendo las etapas del modelo en el que se basó (Ciclo de Deming) e incluyendo la justificación de haberlo utilizado.

Tomando en cuenta el modelo utilizado, se consideraron tres iteraciones del Ciclo de Deming. Cada iteración se desarrolla en un capítulo distinto, incluyendo la primera iteración en el capítulo 2, la segunda iteración en el capítulo 3, y la tercera iteración en el capítulo 4. Es necesario indicar que las etapas que se incluyen en cada ciclo abarcan pruebas y su análisis, por lo que no se tiene un capítulo específico para definirlos.

El capítulo 5: Integración final, reúne elementos que fueron mencionados en las iteraciones del Ciclo de Deming pero que no fueron explicados hasta ese momento, resume la versión obtenida del intercambiador automático de muestreo con el enfoque de IoT, así como las pruebas finales que aún forman parte de la última iteración del ciclo.

Posteriormente, se encuentran las conclusiones, el trabajo a futuro, los trabajos citados y otras fuentes de consulta.

Para terminar, se observan los anexos, los cuales incluyen las fotografías de las piezas manufacturadas para el intercambiador, así como el código que cumple con diferentes funciones del sistema.



Capítulo 1 - Antecedentes

Este capítulo introduce a los temas relacionados con la tesis:

- Trampa de succión para insectos de alta movilidad, incluyendo el intercambiador automático de muestreo.
- IoT y su arquitectura.
- Proceso de desarrollo del prototipo, Ciclo de Deming.



Trampa de succión para insectos de alta movilidad

De acuerdo con los investigadores del INIFAP, para poder llevar a cabo el monitoreo de los insectos que pueden convertirse en plaga, es necesario capturar una muestra en la zona geográfica en cuestión, que posteriormente será analizada. Esto se logra al colocar trampas, en donde se incluye un sistema de muestreo.

Existen trampas que funcionan de diferente manera, en este caso, se colocó una trampa de succión, que como su nombre lo dice, su función es succionar insectos a través de un tubo hacia un frasco de recolección para su estudio.

Las trampas de succión se pueden dividir en diferentes tipos de acuerdo con la potencia y la longitud o diámetro del tubo por donde se succionan las especies a recolectar, el prototipo instalado en el Campo Experimental del INIFAP de Mocochoá, Yucatán, es tipo Rothamsted, llamada así, al ser desarrollada por la *Rothamsted Experimental Station* de Gran Bretaña.

Una trampa de succión tipo Rothamsted colecta insectos que están siendo transportados por el canal de viento a una determinada altura sobre la superficie.

Según Juan Nieto y María Seco (Nieto & Seco, 1990) este tipo de trampa se caracteriza por un tubo de aspiración de 25 [cm] de diámetro que alcanza los 12.2 [m] de altura, en su parte superior se encuentra una malla metálica que impide el acceso de especies más grandes que las de estudio; en la parte inferior del tubo se encuentra una cabina cerrada herméticamente de 3 [m] de altura; en el interior de esta cabina se encuentra un extractor, así como un cono de red metálica, cuya parte más ancha se une al tubo de aspiración, mientras que la contraparte desemboca en un frasco de recolección. Estas características se señalan en la Figura 1.



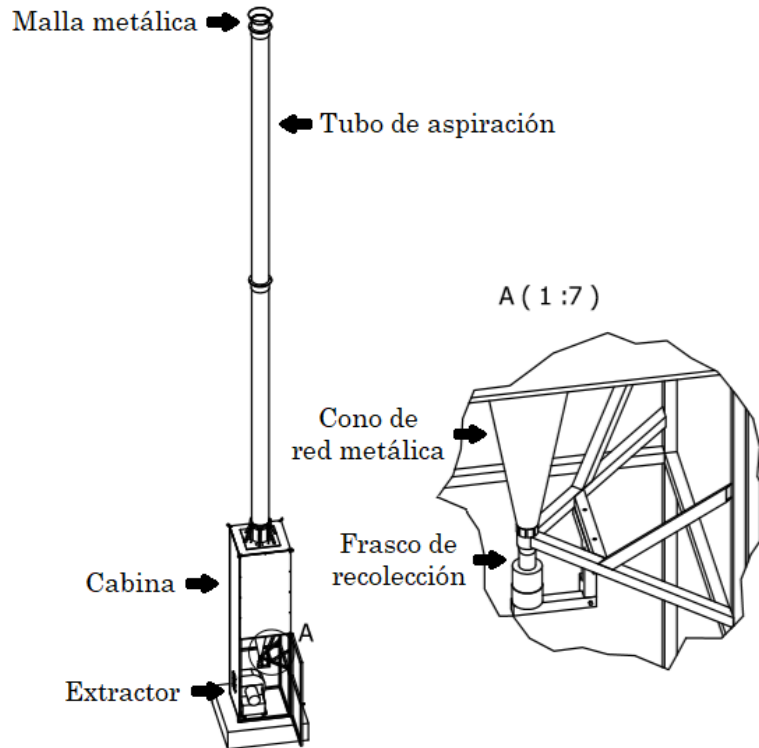


Figura 1: Características de la trampa tipo Rothamsted. Imagen elaborada a partir de los planos de la trampa instalada en Mocochá.

Describiendo el funcionamiento de la trampa, el extractor se enciende tomando el aire a través del tubo de aspiración, al sobrevolar los insectos por encima del tubo y al utilizar como medio de transporte el aire, estos son succionados a través del tubo hacia el cono de red metálica, para lo cual, al pasar el aire a través de esta red hacia la cabina, los insectos terminan cayendo al interior del frasco de recolección; finalmente, el aire es devuelto al exterior gracias al extractor por un lateral de la cabina.

Para que la especie recolectada desemboque en un frasco de recolección se diseñó un sistema intercambiador de muestras, para coleccionar especímenes diariamente en frascos con una solución de alcohol al 70%.

Intercambiador automático de muestreo

En el año 2017, como parte del convenio entre el IIMAS y el INIFAP, se diseñó un sistema intercambiador de muestras debido a las siguientes razones:

- Evitar el cambio manual de uno por uno de los frascos de recolección.
- Acudir al campo donde se encuentra la trampa a recolectar más de una muestra, reduciendo el gasto de recursos, ya que no es sencillo llegar a la trampa.

- Tener un mayor control sobre el horario y fecha en que se toma cada muestra.
- Obtener datos sobre factores relacionados con la muestra, así como las condiciones ambientales que podrían afectar a la trampa. Algunos factores son temperatura interna y humedad.
- Suspender la toma de muestras cuando llueva o en caso de una tormenta o huracán.

Se diseñaron y fabricaron diferentes versiones de dicho sistema, mejorando diferentes aspectos entre cada una. La última versión que se probó y utilizó se muestra en la Figura 2. Con esta versión se comenzó a trabajar.



Figura 2: Última versión probada del intercambiador automático de muestreo, antes de comenzar con la presente investigación. Fotografía tomada por personal del INIFAP.

La mayoría de las piezas de esta versión se fabricaron con manufactura aditiva, a continuación se enlistan y se señalan en la Figura 3:

- Aro sujetador del frasco de recolección.
- Escuadra sujetadora del frasco de recolección.
- Paleta del carrusel.
- Eje del carrusel.
- Engrane para girar el carrusel.
- Engrane para servomotor.
- Porta servomotor.
- Émbolo.
- Tornillo de unión entre el eje del carrusel, el engrane para girar el carrusel y el émbolo. Además, se propuso una tuerca para sujetar un balero que permite el giro del carrusel.
- Guía - porta motor.

- k. Tornillo sin fin, emulado a partir de un balero y tuercas; se conecta y atraviesa al émbolo.
- l. Cople para motor, que conecta el motor de CD con el tornillo sin fin que se emuló.
- m. Tapa - sujetador
- n. Sujeción del sistema a la estructura de la trampa, a través de una escuadra con nervio y tornillos.



Figura 3: Piezas de la versión que falló del intercambiador automático de muestreo. Las letras coinciden con la lista en el texto.

Por otra parte, los componentes electrónicos, que se utilizaron en la última versión que funcionó, se escriben a continuación; algunos de estos se retomaron para la nueva versión del intercambiador que se desarrolló con esta investigación:

- SBC: es una tarjeta de tamaño reducido que contiene las características de una computadora (microprocesador, memoria, puertos de entrada/salida, entre otros). Se utilizó la SBC de origen británico, Raspberry Pi, que fue desarrollada por la *Raspberry Pi Foundation* con el propósito de utilizarla para enseñar la informática de una forma atractiva. Sin embargo, el gran impacto que tuvo la convirtió en una tarjeta popular y útil para desarrollar diversos proyectos. Con el paso de los años se han desarrollado diferentes versiones y modelos de la Raspberry Pi, las cuales cumplen diferentes necesidades según el proyecto a desarrollar. En este caso se utilizó la Raspberry Pi Zero W, gracias a su bajo costo y tamaño, además, contiene los módulos de WiFi y Bluetooth integrados. En la Tabla 1, se indican las características de este modelo. Es necesario considerar que esta tarjeta no tiene entradas dedicadas a señales analógicas.

SBC: Raspberry Pi Zero W	
Característica	Descripción
Microprocesador	Broadcom BCM2835 de 1 núcleo a 1 [GHz]
RAM	512 [MB]
Puertos	Mini - HDMI, micro - USB (uno para alimentación y otro para datos), conector CSI para cámara
Almacenamiento	Externo, utilizando una micro - SD
GPIO	40, acomodadas en un <i>header</i> de pines de 20x2, de las cuales algunas no son reconfigurables
Alimentación	5 [V] 2.5 [A]
Dimensiones	65 [mm] x 30 [mm] x 5 [mm]
Voltaje máximo suministrado en los GPIO	3.3 [V]
Corriente máxima suministrada en los GPIO	0.5 [mA]
Voltaje que se consume en los GPIO	3.3 [V]
Corriente máxima que se puede consumir en los GPIO	100 [mA] en total de los GPIO. Considerar máximo 50 [mA] en un GPIO.
Temperatura de operación	0 a 70 [°C]
Costo	\$290.00

Tabla 1: Características de la SBC (Raspberry Pi Zero W). Tabla construida con base en la información en (Raspberry Pi Foundation, 2018a) y (Raspberry Pi Foundation, 2019a).

- Motor de CA: el extractor que permite absorber los insectos al interior de la trampa de succión funciona mediante un motor de corriente alterna. Las características eléctricas más importantes de este motor se encuentran en la Tabla 2:

Motor de CA	
Característica	Descripción
Voltaje de alimentación	127 / 220 [VCA]
Corriente de alimentación	14.20 / 6.83 [A]
Potencia	746 [W]
Temperatura máxima de operación	40 [°C]

Tabla 2: Características del motor de CA. Tabla construida con base en la información de la etiqueta.

- Servomotor: la ventaja de este dispositivo electromecánico es que su posición puede ser controlada mediante la modulación por ancho de pulsos (PWM). Se utilizó el Tower Pro MG995, con engranaje de metal, que modifica su posición hasta 180°. Para su conexión, cuenta con un conector tipo “S” cuyas tres terminales se identifican según el color de su cable:
 - Cable de color anaranjado: PWM.
 - Cable de color rojo: Vcc.
 - Cable de color café: Tierra.



Este componente se utiliza para girar el eje del carrusel, utilizando el juego de engranes para girar más de 180°. Algunas de sus características se muestran en la Tabla 3.

Servomotor: Tower Pro MG995	
Característica	Descripción
Rango de voltaje de alimentación	4.8 a 6.6 [V]
Corriente sin carga	170 [mA]
Consumo de corriente máximo (con carga máxima)	1200 [mA]
Par permitido	9.4 [kg/cm] para 5 [V]; 11 [kg/cm] para 6 [V]
Masa	65 [g]
Temperatura de operación	0 a 55 [°C]
Costo	\$255.00

Tabla 3: Características del servomotor (Tower Pro MG995). Tabla construida con base en la información en (330 ohms, 2019a) y (Torq Pro & Tower Pro, 2014).

- Motor de CD: se encarga de mover el tornillo sin fin que se emuló, con la finalidad de elevar y bajar el émbolo. Para obtener una velocidad reducida se utilizó un motorreductor de plástico de 140 [RPM], el cual cuenta con un motor de corriente directa acoplado. Las principales características de este motor se observan en la Tabla 4.

Motor de CD: Motorreductor de plástico	
Característica	Descripción
Voltaje de alimentación	4.5 [V]
Corriente sin carga	190 [mA]
Corriente máxima con carga	250 [mA]
Par permitido	800 [g/cm]
Masa	27 [g]
Costo	\$35.00

Tabla 4: Características del motorreductor de plástico. Tabla construida con base en la información en (330ohms, 2019c).

- Puente H: se empleó para controlar el giro del motor de corriente directa y proteger a la SBC. En específico, se utilizó el circuito integrado L293D que, a diferencia de otros L293, incluye diodos de protección. Las principales características se muestran en la Tabla 5.



Puente H: L293D	
Característica	Descripción
Rango de voltaje de alimentación del circuito integrado	4.5 a 7 [V]
Corriente de alimentación máxima	100 [μ A] para los pines A y 10 [μ A] para los pines EN, teniendo un voltaje de alimentación de 7 [V]
Rango de voltaje de alimentación para el motor	De voltaje de alimentación del circuito integrado a 36 [V]
Corriente máxima de salida al motor	600 [mA] por canal
Temperatura de operación	0 a 70 [°C]
Costo	\$40.00

Tabla 5: Características del puente H (L293D). Tabla construida con base en la información en (Texas Instruments, 2016).

- BMP180: este sensor se colocó para medir la presión barométrica y la temperatura dentro de la cabina de la trampa. Además, con él se puede obtener de manera indirecta el valor de la altitud, aunque este no es preciso ya que se trata de una estimación con base en los valores de la presión y temperatura. Se utilizó el módulo GY-68 que ocupa el bus I²C y que consta sólo de cuatro pines de conexión. Las características principales de este sensor se observan en la Tabla 6.

Sensor: BMP180 (Módulo GY-68)	
Característica	Descripción
Rango de voltaje de alimentación	1.8 a 3.6 [V]
Consumo de corriente máximo	1000 [μ A]
Resolución de la presión	0.01 [hPa]
Resolución de la temperatura	0.1 [°C]
Temperatura de operación	-40 a 85 [°C] (0 a 65 [°C] para precisión completa)
Costo	\$160.00

Tabla 6: Características del BMP180 (GY-68). Tabla construida con base en la información en (Bosch Sensortec, 2011).

- DHT11: se trata de un sensor digital que mide la temperatura y la humedad del ambiente. Se colocó para obtener estos valores en el interior de la cabina de la trampa. Sus características principales se muestran en la Tabla 7.



Sensor: DHT11	
Característica	Descripción
Rango de voltaje de alimentación	3 a 5.5 [V]
Consumo de corriente máximo	2.5 [mA]
Periodo de muestreo	1 [s]
Resolución de la humedad	1 [%RH]
Precisión de la humedad	±5 [%RH]
Resolución de la temperatura	1 [°C]
Precisión de la temperatura	±2 [°C]
Temperatura de operación	0 a 50 [°C]
Costo	\$55.00

Tabla 7: Características del DHT11. Tabla construida con base en la información en (D-Robotics UK, 2010).

- MCP3008: se trata de un dispositivo que contiene un ADC (Analog-to-Digital Converter, traducido al español como convertidor analógico-digital). El objetivo del ADC es transformar de una señal analógica (un voltaje) a una señal digital (bits), permitiendo que, en este caso, la Raspberry Pi, con entradas digitales, obtenga el equivalente en número binario de la señal analógica recibida. La cantidad de bits que otorgará como salida digital depende del fabricante del dispositivo que contiene el ADC, en este caso, el ADC es de 10 bits. Además, este dispositivo tiene ocho canales para obtener la conversión de ocho señales analógicas, y para el envío de información utiliza el bus SPI. Otras características se observan en la Tabla 8.

ADC: MCP3008	
Característica	Descripción
Rango del voltaje de operación	2.7 a 5.5 [V]
Rango del voltaje de referencia	0.25 [V] a voltaje de operación
Rango de voltaje en las entradas analógicas	0 [V] a voltaje de referencia
Máxima corriente de operación	550 [μA] para un voltaje de operación = voltaje de referencia = 5 [V]
Máxima corriente en la salida	10 [μA]
Temperatura de operación	-40 a 85 [°C]
Costo	\$80.00

Tabla 8: Características del MCP3008. Tabla construida con base en la información en (Microchip Technology, 2008).

Para conocer el número binario equivalente al voltaje recibido, se debe conocer la resolución, para obtenerla se utiliza la Ecuación 1. Posteriormente, se utiliza la Ecuación 2 para conocer el valor en decimal equivalente, para finalmente, ser convertido a un número binario.



$$\text{Resolución} = \frac{V_{ref^+} - V_{ref^-}}{2^n}$$

Donde:

V_{ref^+} = Voltaje de referencia positivo.

V_{ref^-} = Voltaje de referencia negativo.

n = número de bits indicados por el fabricante.

Ecuación 1: Resolución del convertidor analógico-digital.

$$\text{Valor}_{Decimal} = \frac{\text{Voltaje}_{Recibido}}{\text{Resolución}}$$

Donde:

$\text{Voltaje}_{Recibido}$ = Voltaje recibido (señal analógica).

Ecuación 2: Valor equivalente en decimal.

Si para este ADC se selecciona el voltaje de referencia de 3.3 [V], su resolución se calcularía como se muestra en Ecuación 3.

$$\text{Resolución} = \frac{V_{ref^+} - V_{ref^-}}{2^n} = \frac{3.3 - 0}{2^{10}} = 0.00322 [V]$$

Ecuación 3: Resolución del ADC en cuestión, al sustituir los valores correspondientes en Ecuación 1.

- **LM35:** en general se refiere a una serie de componentes de precisión para medir la temperatura, entregando un voltaje proporcional a esta (señal analógica). Dependiendo del dispositivo de la serie es el rango de temperatura que mide, aunque el factor de escala que define el valor de la temperatura con respecto al voltaje es el mismo: 10 [mV] por cada grado centígrado. El objetivo de utilizar este componente fue medir la temperatura del motor de CA, por lo que se seleccionó el modelo LM35DZ, cuyo rango de valores por medir abarca los que se podrían tener, y su tipo de empaquetado TO-92 sirve para estar en contacto directo con la superficie del motor. Las principales características de este componente se observan en la Tabla 9. Cabe resaltar que, la señal debe ingresar al ADC antes que a la SBC.

Sensor: LM35DZ	
Característica	Descripción
Rango de voltaje de alimentación	4 a 30 [V]
Corriente de alimentación máxima	60 [μA]
Corriente de salida máxima	10 [mA]
Precisión	±0.6 [°C] para 25 [°C]; ±0.9 [°C] para 0 y 100 [°C]
Temperatura de operación	0 a 100 [°C]
Costo	\$40.00

Tabla 9: Características del LM35DZ. Tabla construida con base en la información en (Texas Instruments, 2017).



- ECS1030: este componente pertenece a la serie de componentes ECS10, conocidos como “Transformadores de corriente de núcleo dividido”. Se utiliza para medir la corriente de una línea de suministro de carga eléctrica de manera no-invasiva, es decir, sin la necesidad de interrumpir la línea, ya que se coloca alrededor de esta. Tiene como características las que se muestran en la Tabla 10. No cuenta con resistencia de carga, por lo que es necesario colocarle un resistor a la salida, para obtener un voltaje medible correspondiente a la corriente medida con el sensor. La razón de utilizar dicho sensor fue medir la corriente del motor de CA, para apagarlo en el caso en que supere la corriente máxima. Cabe resaltar que, la señal que se obtiene es analógica, por lo que se necesita conectar al ADC antes de llegar a la SBC.

Sensor: ECS1030	
Característica	Descripción
Corriente máxima en línea de suministro	30 [A]
Relación de giro	$N_p:N_s = 1:2000$
Relación de corriente	30 [A] / 15 [mA]
Temperatura de operación	-40 a 65 [°C]
Costo	\$350.00

Tabla 10: Características del ECS1030. Tabla construida con base en la información en (Echun Electronic Co, s.f.) y (330ohms, 2019b).

En un tutorial de un colaborador de *SparkFun Electronics* (rwaldron, 2016) se muestra el circuito necesario para utilizar este sensor (Figura 4), acoplándolo al sistema en cuestión con ayuda de la información en (Learn Open Energy Monitor, s.f.), con la que se realizaron los cálculos para obtener los valores adecuados de los componentes de su circuito, tal como se describe a continuación.



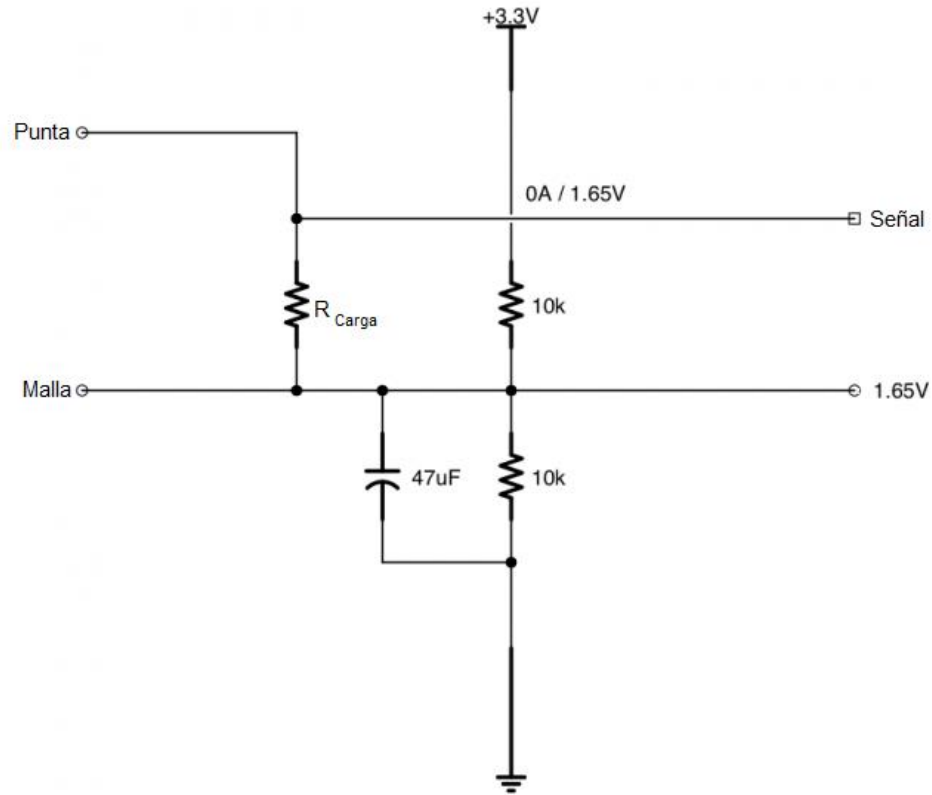


Figura 4: Circuito para el ECS1030. Diagrama obtenido en el tutorial de un colaborador de *SparkFun Electronics* (rwaldron, 2016).

Se sabe que el rango de corriente RMS en el devanado primario es de 0 a 30 [A], tomando la Ecuación 4 y considerando la máxima corriente, se tiene un valor pico de $30\sqrt{2}$ [A].

$$Corriente_{pico} = Corriente_{RMS} * \sqrt{2}$$

Ecuación 4: Corriente pico.

Considerando la relación de giro, la corriente en el devanado secundario se obtiene con la Ecuación 5, el valor que se tendría sería de $\frac{30\sqrt{2}}{2000}$ [A].

$$Corriente_{Secundario} = \frac{Corriente_{primario}}{N_s}$$

Ecuación 5: Corriente pico en el devanado secundario.

Es importante indicar que para esta corriente se tendría un valor correspondiente de voltaje máximo que, debido al divisor de voltaje y recordando el rango de voltaje en las entradas analógicas del ADC, si se considera el voltaje de referencia del ADC de 3.3 [V], el voltaje máximo sería la mitad de este voltaje.



Con los valores anteriores y utilizando la Ley de Ohm, se obtiene la resistencia de carga como se muestra en Ecuación 6. Ya que el resultado no es una resistencia comercial, se seleccionó un valor comercial menor: 68 [Ω].

$$R = \frac{V}{I} = \frac{\frac{3.3}{2}}{\frac{30\sqrt{2}}{2000}} = 77.78 [\Omega]$$

Ecuación 6: Resistencia de carga obtenida gracias a la Ley de Ohm.

En cuanto al capacitor, se continuó utilizando el de 47 [μF].

Por otro lado, si bien se conocía cómo debía comportarse el sistema del intercambiador, no se tuvo acceso al código o bibliotecas² utilizadas para describirlo como antecedente.

Por lo anterior, para desarrollar este proyecto, se consideró dividir el sistema en tres subsistemas:

- I. Mecánico: incluye los mecanismos y estructuras.
- II. Electrónico: abarca las conexiones entre la SBC (*Single-Board Computer*, en español: computadora en sólo una tarjeta), actuadores, sensores y todos los componentes electrónicos necesarios.
- III. Software: se encarga de controlar el comportamiento del intercambiador.

² Biblioteca: es “un conjunto de recursos (algoritmos) prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones” (Zator Systems, 2016).



Internet de las Cosas

El término IoT fue propuesto en 1999 por Kevin Ashton, cuyo campo de investigación era la identificación por radiofrecuencia en red (RFID) y que debido a la popularidad que tenía la Internet³ en ese momento, decidió utilizar el término Internet de las Cosas al hacer una presentación para Procter & Gamble (P&G) para referirse al uso de RFID en su cadena de suministro (Ashton, 2009). Con los años, el significado de IoT se ha ampliado.

Aún no existe un consenso en cuanto a la definición, pero tomando en cuenta la que se encuentra en (Van Kranenburg, 2007): “una infraestructura de red global dinámica con capacidades de autoconfiguración basadas en protocolos de comunicación estándar e interoperables donde cosas físicas y virtuales tienen identidades, atributos físicos, y personalidades virtuales y usan interfaces inteligentes, y están perfectamente integrados en la red de información” y en (Gartner, 2019): “La Internet de las cosas (IoT) es la red de objetos físicos que contienen tecnología integrada para comunicarse y detectar o interactuar con sus estados internos o el entorno externo”, la IoT se puede definir como un concepto dirigido a una red de dispositivos conectados (sensores, actuadores, entre otros) que se comunican e interactúan entre sí, utilizando una familia de protocolos para intercambiar información a la que los humanos tienen acceso.

Arquitectura

La arquitectura hace referencia a la manera en la que se organizan los elementos de una IoT. Para esto se utiliza un modelo por capas que durante años se ha modificado y adecuado a los requerimientos de cada persona, por lo que aún no se tiene una arquitectura estándar. A pesar de ello, se puede partir de la arquitectura más básica de 3 capas (Figura 5), como la describe Pallavi Sethi y Smruti Sarangi (Sethi & Sarangi, 2017). A continuación se explica cada capa:

- Capa de Percepción (*Perception Layer*): también conocida como la capa de los objetos. Incluye dos tipos de objetos:
 - Los que contienen sensores, cuya información captada será procesada.
 - Los que contienen actuadores, cuyas acciones provocarán el cambio de estado del objeto.

³ Internet: “es una red global de área amplia que conecta sistemas informáticos en todo el mundo” (3G E-Learning LLC, 2018b). Usa la familia de protocolos TCP/IP que “se utiliza para resolver la transmisión de datos que se genera en una red ya sea de forma interna o externa. La unión de estos protocolos se realiza para asegurar que la información llegue siempre por la mejor ruta y de forma correcta a destino” (Qloudea, 2017).



- Capa de Red (*Network Layer*): su objetivo es transferir la información mediante diferentes tecnologías. Incluye el *Gateway*⁴, protocolos de red⁵, protocolos de seguridad, entre otros. En esta capa la información es filtrada y preparada antes de ser enviada a la nube.
- Capa de Aplicación (*Application Layer*): es la capa de los servicios web⁶. Incluye el software para conectar a los usuarios con los datos del sistema, permitiendo que sea monitoreado y controlado mediante comandos enviados desde diferentes aplicaciones. Además, incluye el almacenamiento de la información en bases de datos, su consulta, la creación de gráficas y diagramas para su evaluación, e incluso, en algunos sistemas recientes se ha incluido el análisis y control mediante inteligencia artificial.



Figura 5: IoT en 3 capas. Diagrama elaborado a partir de la información obtenida en (Sethi & Sarangi, 2017) y (Al-Qaseemi, Almulhim, Almulhim, & Rasool, 2016).

Cabe señalar, que a partir de las tres capas anteriores se pueden desprender más capas, en tanto que, mientras más capas se tengan definidas, se tendrá una mejor comprensión de los elementos del sistema y un mayor control sobre su funcionamiento antes de ser integradas; no obstante, conlleva definir más detalles sobre los elementos que bien podrían quedarse como los establecidos por defecto cuando se trata de aplicaciones sencillas, invirtiendo en más recursos de los necesarios.

⁴ Gateway: traducido al español como puerta de enlace, es un “dispositivo o conjunto de dispositivos que conecta dos o más redes, permitiendo la transferencia de datos entre ellas” (3G E-Learning LLC, 2018a).

⁵ Protocolo de red: “Conjunto de reglas y procedimientos que rigen la transmisión entre los componentes de una red” (3G E-Learning LLC, 2018c).

⁶ Servicios web: “aplicaciones autónomas modulares que se pueden describir, publicar, localizar e invocar a través de una red” (IBM, 2019), utilizan “un conjunto de protocolos y estándares facilitando el intercambio de información entre sistemas” (Del Valle, 2015). Su nombre proviene del hecho de ser diseñados para residir en un servidor web ya que fueron creados para resolver el problema que surgió tras buscar intercambiar información entre aplicaciones que habían sido creadas con diferentes lenguajes de programación y que residían en diferentes sistemas operativos, esto se convierte en la principal característica de los servicios web: la interoperabilidad.

Para esta investigación se utilizó el modelo de 3 capas, y se definieron los elementos para cada capa que se muestran en la Figura 6.

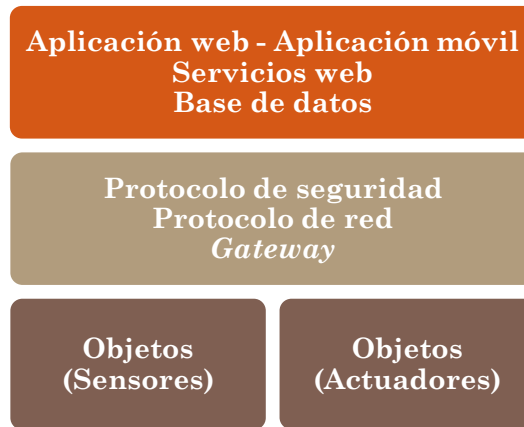


Figura 6: Elementos de las capas de IoT considerados en esta investigación.

Proceso de desarrollo del prototipo

En primera instancia, se pensó en el intercambiador de frascos de esta investigación como un producto, por lo que se consultó el libro “Diseño y desarrollo de productos” de Karl Ulrich y Steven Eppinger (Ulrich & Eppinger, 2013) para emplear el proceso de desarrollo del producto que en él se indicara, con el antecedente de haber utilizado este libro durante los estudios en la licenciatura. Dado que el intercambiador se podía considerar como un producto de rápida elaboración, debido a la disponibilidad de modelar y generar prototipos mediante manufactura aditiva o con corte láser, era posible aplicar el Proceso de desarrollo del producto en espiral, como en el texto se señalaba. No obstante, se descubrió que el Modelo en espiral, en el que se basa el proceso anterior, fue propuesto por Barry Boehm en 1988 (Sommerville, 2011), para ser usado, principalmente, en ingeniería de software. Si bien en este proyecto se incluye el desarrollo de software, se debía considerar la inclusión de componentes electrónicos y piezas manufacturadas. Además, al indagar sobre las etapas mencionadas en el libro sobre este proceso de desarrollo, se optó por no utilizar el proceso tal cual se indicaba en el texto, ya que la versión del intercambiador que se desarrolló debía considerarse como un prototipo y no como un producto, puesto que su producción en masa aún no se definió, e incluso, hubo detalles que no se consideraron, estos se explican en el Trabajo a futuro de este escrito.

Por lo anterior, un modelo parecido, pero que se puede aplicar de manera más general, no sólo al software, es el Ciclo de Deming, también conocido como Espiral de mejora continua. Fue utilizado en 1950 por Edwards Deming, quien se basó en las ideas de Walter Shewhart sobre la mejora continua de la calidad (Alemany, 2004). Este modelo consta de cuatro etapas, mostradas en la Figura 7, que se repiten continuamente hasta completar el número de iteraciones necesarias para cumplir con las expectativas del proyecto en cuestión.

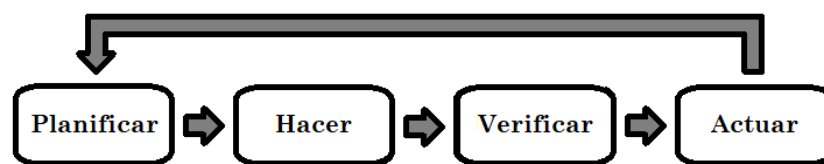


Figura 7: Ciclo de Deming. Diagrama elaborado a partir de la información obtenida en (García, 2016).

La etapa Planificar consiste en la recopilación y análisis de la información que se tiene del proyecto, se establecen los recursos a ocupar y las acciones que se llevarán a cabo. En la siguiente etapa, Hacer, se efectúan las acciones referentes a los cambios, según se haya planeado. En la etapa Verificar, se comprueba que los cambios llevados a cabo cumplan con los objetivos y especificaciones del proyecto. En la última etapa del ciclo, Actuar, se decide si implantar la mejora de manera definitiva o si hay cambios por realizar para ajustar los resultados.



Considerando el Ciclo de Deming, en los siguientes capítulos se describen las etapas de las tres iteraciones que se realizaron para constituir el proceso de desarrollo del sistema del intercambiador de frascos.



Capítulo 2 - Primera iteración del ciclo

Correspondientes a la primera iteración, en este capítulo se incluyen las cuatro etapas del Ciclo de Deming: Planificar, Hacer, Verificar y Actuar.



Etapa de Planificar

Para esta primera etapa, se recopiló la información del intercambiador, como: su función; la propuesta de valor para la versión del intercambiador de esta investigación; los objetivos clave referentes a este prototipo, más allá de todo el proyecto; los principales beneficiarios; y las suposiciones y restricciones que afectarían la situación final del intercambiador. La información anterior se muestra en la Tabla 11.

Intercambiador automático de muestreo	
Descripción del prototipo	Colocar un frasco de recolección por debajo del cono de red metálica, de tal manera que los insectos puedan desembocar dentro del frasco sin escapar. Una vez obtenida la muestra, intercambia automáticamente el frasco para tomar otra muestra.
Propuesta de valor	Enfoque de Internet de las Cosas. Corrección de problemas con versiones anteriores.
Objetivos clave	Rediseñar las piezas que generaron problemas en el intercambiador, y así, mejorar su calidad de funcionamiento. Reducir costos empleando métodos de manufactura de piezas más económicos. Obtener los subsistemas Mecánico y Electrónico funcionando antes del 18 de junio del 2019, fecha en la que se llevará a cabo la instalación en Yucatán.
Usuarios	Investigadores del Campo Experimental del INIFAP. Agricultores del área. Investigadores del IIMAS.
Suposiciones y restricciones	Se trata de una mejora incremental a un producto existente. Enfocar el sistema a formar una IoT. Considerar las medidas de la trampa instalada en Yucatán, de los de los frascos de recolección (incluyendo su peso con la solución), y de la caja que contiene parte de los componentes electrónicos. Las piezas de la versión anterior fueron manufacturadas en una impresora 3D. La impresora 3D se puede utilizar para la manufactura de piezas. Ser de sencilla instalación, mantenimiento, y en su caso, reparación. El ambiente donde se instalará la trampa es húmedo y su temperatura promedio puede variar entre 16° y 36° (INEGI, 2015). Tomar en cuenta las restricciones de temperatura, voltaje y corriente del subsistema Electrónico. Reducir costos.

Tabla 11: Información recopilada del intercambiador automático de muestreo.



Los beneficiarios de este proyecto son los principales usuarios, satisfacer en lo posible sus necesidades era indispensable para que los resultados fueran satisfactorios. En Tabla 12 se describen las necesidades reunidas de los usuarios mencionados, además, cada una se clasificó de acuerdo con el subsistema al que pertenecen.

No. Necesidad	Subsistema al que pertenece	Necesidad
1	Mecánico	El intercambiador cabe dentro del espacio designado para este.
2	Mecánico	Las piezas del intercambiador soportan esfuerzos sin dañarse.
3	Mecánico	Las piezas del intercambiador soportan la introducción de los tornillos sin dañar la estructura.
4	Mecánico	El sistema permanece ajustado a pesar del movimiento, es decir, las piezas no se aflojan a pesar del movimiento.
5	Mecánico	El engrane para servomotor se mueve conforme al servomotor, es decir, sin barrerse o deslizarse.
6	Mecánico	Los frascos están sujetos al sistema, en su lugar, a pesar del movimiento del intercambiador.
7	Mecánico	La sujeción del sistema a la estructura de la trampa es a través de otro tipo de pieza(s).
8	Mecánico	Los frascos permanecen sellados mientras no se encuentran debajo de la boca del cono para recolectar las muestras.
9	Electrónico	Los actuadores soportan la masa que deben mover.
10	Electrónico	Los componentes electrónicos funcionan dentro de su voltaje límite.
11	Electrónico	Los componentes electrónicos funcionan dentro de su corriente límite.
12	Electrónico	Los componentes electrónicos funcionan dentro de su temperatura límite.
13	Electrónico	Se apaga automáticamente el extractor cuando sobrepase los 30 [A] de corriente o los 40 [°C] de temperatura; así como cuando llueve; e incluso cuando esté en movimiento el intercambiador automático de muestreo.
14	Electrónico	Hay indicador para cuando la SBC está encendida.
15	Electrónico	Hay indicador para cuando la SBC se integra a la red.
16	Software	Existen los estados del frasco "arriba" y "abajo". Cuando el frasco está "arriba" no se puede intercambiar; cuando está "abajo", si se puede intercambiar.
17	Software	El intercambiador coloca el frasco de recolección exactamente debajo del cono de red metálica para tomar una muestra.



18	Software	Se identifica el número de frasco que se encuentra debajo del cono.
19	Software	Se controla el movimiento del intercambiador a través de una aplicación móvil.
20	Software	Se controla el movimiento del intercambiador a través de una aplicación web.
21	Software	Se obtienen los datos de los sensores a través de una aplicación móvil.
22	Software	Se obtienen los datos de los sensores a través de una aplicación web.
23	Software	Se obtienen los datos del estado del proceso del intercambiador a través de una aplicación móvil.
24	Software	Se obtienen los datos del estado del proceso del intercambiador a través de una aplicación web.

Tabla 12: Necesidades de los usuarios. Etiquetadas según los tres subsistemas: Electrónico, Mecánico y Software.

Con el propósito de expresar en términos técnicos cada una de las necesidades reunidas, se establecieron las especificaciones del intercambiador en forma de métricas, de forma que las necesidades pudieran medirse de manera cuantitativa o cualitativa, tal como se muestran en Tabla 13.

No. Métrica	No. Necesidad	Métrica	Unidades	Valor
1	1, 7	Distancia entre el final de la sujeción del sistema y el centro del embudo de la trampa.	mm	< 166.61
2	1	Ancho de la cabina donde se sitúa el intercambiador.	mm	711.24
3	1	Largo de la cabina donde se sitúa el intercambiador.	mm	710.82
4	2	Pruebas de todas las piezas ante esfuerzos a los que serán sometidos.	Binaria	Soporta.
5	2	Grosor de pared de la guía - porta motor.	mm	> 1.9
6	2	Grosor de material en las piezas (paredes en las piezas).	mm	> 1.4
7	2	Diámetro de los agujeros ahorra - material de la paleta del carrusel.	mm	< 10
8	2	Espacio entre los orificios de las paletas de unión del eje del carrusel.	mm	> 4.5
9	3	Pruebas de introducción y extracción de tornillos en todas las piezas que los requieren.	Binaria	Soporta.
10	3	Ancho de la cavidad del eje del carrusel para colocar tuerca.	mm	12.4 - 13.4



11	3	Ancho de la sección del émbolo donde gira la cabeza del tornillo del carrusel	mm	> 15
12	4	Pruebas de movimiento en el sistema para comprobar que las piezas no se desajustan.	Binaria	Ajustado.
13	4	Unión del servomotor al porta servomotor.	Binaria	Ajustado.
14	4	Unión del cople para motor con el motor y el tornillo sin fin que se emuló.	Binaria	Ajustado.
15	4, 6	Unión entre el aro sujetador del frasco y la escuadra sujetadora del frasco de recolección.	Binaria	Alineados.
16	5	Unión entre el engrane para servomotor y el servomotor.	Binaria	No se barre.
17	6	Diámetro interior del aro sujetador del frasco de recolección.	mm	61.2
18	7, 16, 17	Distancia entre la boca del embudo y la boca del frasco cuando el estado del frasco es "arriba".	mm	< 0
19	7, 16	Distancia entre la boca del embudo y la boca del frasco cuando el estado del frasco es "abajo".	mm	> 10
20	8	Estado de los frascos mientras no recolectan muestras.	Binaria	Sellado.
21	9	Masa del frasco de recolección.	g	32
22	9	Masa en conjunto de los seis frascos, los seis aros sujetadores de los frascos, las seis escuadras sujetadoras, las seis paletas del carrusel, el eje del carrusel, el engrane para girar el carrusel, el engrane para el servomotor, el porta servomotor, el émbolo, tornillos y baleros necesarios.	g	< 800
23	9	Masa en conjunto de los seis frascos, los seis aros sujetadores de los frascos, las seis escuadras sujetadoras, las seis paletas del carrusel, el eje del carrusel, el engrane para girar el carrusel, el engrane para el servomotor y tornillos necesarios.	g	< 9,400
24	10	Voltaje límite de cada componente.	Binaria	No supera.
25	11	Corriente límite de cada componente.	Binaria	No supera.
26	12	Temperatura límite de cada componente.	Binaria	No supera.
27	13	Corriente del motor de CA del extractor para apagarlo.	A	> 29



28	13	Temperatura para apagar el extractor.	°C	> 39
29	13	Estado del extractor cuando llueve.	Binaria	Apagado.
30	13	Estado del extractor cuando el intercambiador automático de muestreo está en movimiento.	Binaria	Apagado.
31	14, 15	Indicadores en la SBC.	Subjetiva	SBC encendida. SBC integrada a la Red.
32	16	Intercambio del frasco cuando su estado es "arriba".	Binaria	Prohibido.
33	17	Distancia entre el centro de la boca del embudo y el centro de la boca del frasco.	mm	0
34	18	Conocimiento del número de frasco en el que se encuentra.	Binaria	Verdadero.
35	19, 20	Control de movimiento a través de aplicaciones.	Subjetiva	App web. App móvil.
36	21, 22	Obtención de datos de los sensores.	Subjetiva	App web. App móvil.
37	23, 24	Obtención de estado del proceso.	Subjetiva	App web. App móvil.

Tabla 13: Especificaciones del intercambiador. Definidas como métricas a partir de las necesidades de la Tabla 12. Se indican las unidades y valores para cada métrica.

Posteriormente, se identificaron los puntos críticos que arriesgaban el cumplimiento de lo definido en la Tabla 11, la Tabla 12 y la Tabla 13. Estos puntos y su descripción se encuentran en la Tabla 14.

No. Punto crítico	Punto crítico	Descripción
1	Grosor de la guía - porta motor y de la tapa - sujetador	El grosor de las paredes de estas piezas no fue suficiente para evitar dañarse mientras se manipulaban, las capas que se formaron a partir de la impresión 3D se separaron; además, la introducción de los tornillos también dañó estas piezas debido al grosor de las paredes de los orificios donde debían introducirse los tornillos y formar la cuerda. Además, para la tapa - sujetador, la parte que más daño sufrió fue la que rodeaba al motor para inmovilizarlo.
2	Agujeros de la paleta del carrusel y de la escuadra sujetadora del frasco de recolección	Como ya se mencionó, en la Tabla 13, los agujeros ahorra - material de las paletas del carrusel debían modificarse, esto debido a que estas piezas no soportaron los esfuerzos al ser manipuladas, obteniendo dos piezas dañadas. En cuanto a la escuadra sujetadora del frasco de recolección, esta pieza no se había dañado, pero corría el riesgo de hacerlo de la misma manera que las paletas del carrusel.



3	Engrane para servomotor barrido	Los coples comerciales del servomotor se adhieren a este gracias a su orificio dentado y a un tornillo. El engrane que se había manufacturado en la versión antecesora buscaba seguir el mismo principio, sin embargo, los dientes del orificio se barrieron por el tipo de material.
4	Alineación y fijación entre la paleta del carrusel con la escuadra sujetadora del frasco de recolección, y a su vez, con el aro sujetador del frasco de recolección.	La principal razón de que estas piezas no estuvieran fijas fue para permitir que se pudieran ajustar en caso de alguna corrección no esperada, referente con la alineación del frasco de recolección con el extremo por donde caerían los insectos, ya que en ese momento no se contaba con la suficiente información para fijar estos objetos, como el diámetro del frasco. Además, para la unión de la escuadra con el aro, se utilizaba un tornillo, pero éste era introducido por debajo, dejando la tuerca arriba, la cual fácilmente era aflojada por el movimiento.
5	Ancho de la cavidad del eje del carrusel y de la sección del émbolo donde gira la cabeza del tornillo del carrusel	Nuevamente, como se lee en la Tabla 13, el ancho de la cavidad del eje del carrusel, para colocar la tuerca, se debía modificar, pues no había suficiente espacio en la cavidad para que ingresara la tuerca. En cuanto a la sección del émbolo donde gira la cabeza del tornillo del carrusel, a diferencia de la pieza anterior, en esta sección era posible que la cabeza del tornillo ingresara, no obstante, era necesario que la cabeza del tornillo girara, lo cual no era posible.
6	Fijación del cople para motor	Este cople permitía conectar el eje del motor con la cabeza del tornillo sin fin que se emuló, sincronizando el movimiento de ambos. El problema que se tenía con el cople fue que después de cierto tiempo de uso, el cople se recorría, por lo que se desconectaba de la cabeza del tornillo deteniendo su movimiento.
7	Fijación del servomotor al porta servomotor	El servomotor era fácilmente montable y desmontable en el sistema, lo cual resultaba ser una ventaja, sin embargo, tenía el problema de que el servomotor se movía de lugar durante su funcionamiento, lo que provocaba que el engrane para servomotor y el engrane para girar el carrusel ya no se tocaran y el movimiento dejara de transmitirse.
8	Ampliación del espacio entre los orificios de las paletas de unión del eje del carrusel	El eje del carrusel se diseñó con paletas cortas entre las cuales se introducían las paletas del carrusel para unir las al eje, además, se había planeado que las paletas del carrusel se fijaran a este eje a través de tornillos, sin embargo, a pesar de que el tamaño de los orificios era adecuado para los tornillos seleccionados, el espacio para introducirlos entre las paletas cortas del eje del carrusel no fue suficiente.
9	Sujeción del sistema dentro de la cabina de la trampa	Como ya se mencionó, para sujetar el sistema se ocupaba una escuadra con nervio, tornillos y tuercas. Esto fue una solución improvisada.



10	Agregar sistema de sellado	Los frascos contienen una solución que, dadas las condiciones climáticas, puede evaporarse; además, si llueve y los frascos no se cubren, la solución se llena de agua y se desborda, provocando la pérdida de la muestra capturada.
11	Caracterización del ángulo del servomotor	Debido al servomotor con el que se contaba, se debían definir los valores de la señal que dicho servomotor recibiría para posicionarse en los ángulos deseados, para mover los engranes y definir el lugar de posición de cada frasco.
12	Conocimiento del estado del frasco (“Arriba” y “Abajo”)	Se debía conocer el estado en el que se encontraba el frasco de recolección.
13	Voltaje y corriente	Dadas las restricciones que tiene la Raspberry Pi Zero W con respecto al voltaje y corriente que puede entregar en sus GPIO, se debía verificar que todos los componentes se alimentaran dentro de estos límites.
14	Temperatura	Se debía verificar que los componentes eran aptos para funcionar en el Campo Experimental del INIFAP de Mochochá, donde la temperatura varía entre 16 [°C] y 36[°C].
15	Verificación de encendido y de la conexión a la red	La SBC contiene un LED que indica su encendido. Sin embargo, se decidió que el circuito se encontraría dentro de una caja, por lo que no se podía ver este LED y el usuario desconocía si la SBC se encontraba encendida. Además, también faltaba un indicador para que el usuario conociera cuando la Raspberry Pi se integraba a la red.
16	Número de frasco	Se debía determinar el número de frasco que se encontraba debajo del cono metálico para recolectar la muestra.
17	Forma de obtención de los datos de los sensores y del estado del sistema	Se debían conocer los datos otorgados por cada sensor y los datos del sistema. Los datos del sistema incluyen el encendido y apagado de los motores, y el punto en el que se encuentra el sistema dentro de su ciclo de funcionamiento.
18	Definición de los elementos de la capa de Red	Se debían definir los elementos de la capa considerados en esta investigación.
19	Aplicación web y aplicación móvil	Este punto forma parte de la tercera capa de IoT. Se debía considerar para definir la forma en que los datos se almacenarían y como el usuario tendría acceso a ellos.

Tabla 14: Puntos críticos de la primera iteración del ciclo.

Se planeó resolver cada uno de los puntos críticos generando bocetos, modelos, diagramas y/o lluvias de ideas.



Como software para realizar los modelos, se seleccionó el CAD⁷ llamado Tinkercad, un software de diseño basado en la nube creado por Autodesk, con la ventaja de ser gratuito.

Además, para la solución de los puntos críticos que incluía el diseño de piezas, se optó por considerar, como primera opción, el mismo método de manufactura con el que se había fabricado la pieza antecesora: manufactura aditiva. El otro proceso de manufactura que era posible utilizar fue el corte láser. Es necesario indicar que cada uno de estos procesos tenían restricciones que debieron ser tomadas en cuenta para el diseño de las piezas. Estas restricciones se describen a continuación:

- Restricciones del corte láser:
 - Piezas en 2D con grosores constantes por pieza, y específicos conforme al material.
 - *Kerf*⁸. En este caso se consideró de 0.18 [mm] para acrílico de 3 [mm], y de 0.23 [mm] para acrílico de 9 [mm].
- Restricciones de la manufactura aditiva:
 - Piezas con inclinación con ángulo mayor a 45°.
 - Piezas sin techo, o bien, que su techo pueda ser sostenido por otro tipo de material.
 - Grosor del material extruido. En este caso se consideró de 0.30 [mm].
 - Material disponible para la impresora: filamento PLA⁹.

Cabe mencionar, que la primera capa de la arquitectura de IoT, la capa de los objetos, se compone de los subsistemas Mecánico y Electrónico, a excepción de la SBC que se utiliza como Gateway (elemento de la segunda capa).

⁷ CAD: *Computer-Aided Design*, traducido al español como Diseño Asistido por Computadoras. “consiste en el uso de programas para crear, modificar, analizar y documentar representaciones gráficas bidimensionales o tridimensionales (2D o 3D) de objetos físicos como una alternativa a los borradores manuales y a los prototipos de producto” (Siemens Industry Software Inc., 2019).

⁸ Kerf: porción de material removido durante el proceso de corte. Depende de la máquina y material a utilizar. (CutLaserCut Ltd., 2015).

⁹ PLA: “también llamado ácido poliláctico, se deriva de materias primas naturales y renovables, como el maíz, y pertenece a los poliésteres como un polímero sintético... Es uno de los materiales de impresión 3D más populares” (Contreras, 2019). “En referencia a la temperatura, cualquier objeto o pieza impresa en PLA se vuelve endeble a temperaturas entorno a (60-70) °C” (Impresoras3D.com, 2018).



Etapa de Hacer

Para cada punto crítico, identificado en la etapa anterior, se encontró su solución como se observa en cada una de las tablas siguientes.

No. Punto crítico	Punto crítico	Operación efectuada
1	Grosor de la guía - porta motor y de la tapa - sujetador	Se optó por cambiar el grosor de las paredes dañadas de esta pieza a 2 [mm]. Cabe señalar que, este cambio también impactó sobre otras piezas que debían quedar alineadas con ésta, como fue el caso de la tapa - sujetador, por lo que también se le modificó su grosor. Por otra parte, se optó por quitarle la función de inmovilizar al motor a la tapa - sujetador, así, se diseñó una nueva pieza que cumpliera esta función. En la Figura 8 se encuentra el modelo de esta pieza

Tabla 15: Solución para el punto crítico 1 de la primera iteración del ciclo.



Figura 8: Modelo del inmovilizador del motor.

No. Punto crítico	Punto crítico	Operación efectuada
2	Agujeros de la paleta del carrusel y de la escuadra sujetadora del frasco de recolección	La solución encontrada para corregir este problema fue hacer un rediseño de los agujeros ahorra - material, se reubicaron y se modificó su tamaño. En la Figura 9, se muestra el modelo obtenido de la paleta del carrusel con esta solución.

Tabla 16: Solución para el punto crítico 2 de la primera iteración del ciclo.

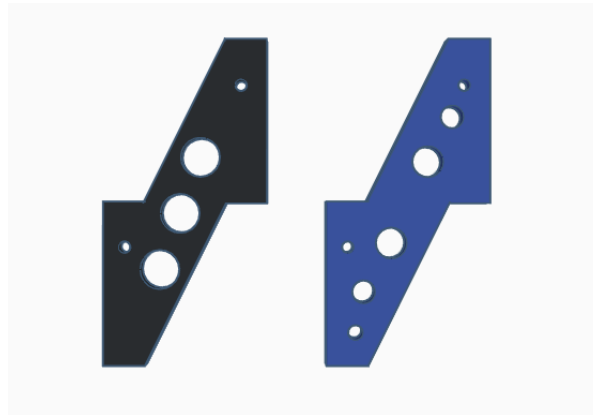


Figura 9: Modelos de las paletas del carrusel. En la izquierda se muestra la pieza anterior y en la derecha la pieza solución.

No. Punto crítico	Punto crítico	Operación efectuada
3	Engrane para servomotor barrido	Se decidió utilizar un cople del servomotor para generar un engrane que no sufriera el mismo desgaste y quedara fijo al servomotor. En la Figura 10, se encuentra el modelo obtenido con esta solución.

Tabla 17: Solución para el punto crítico 3 de la primera iteración del ciclo.

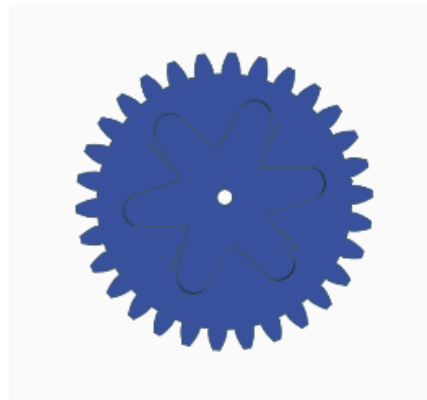


Figura 10: Modelo del engrane para servomotor construido para ser fijado al cople comercial del servomotor.

No. Punto crítico	Punto crítico	Operación efectuada
4	Alineación y fijación entre la paleta del carrusel con la escuadra sujetadora del frasco de recolección, y a su vez, con el aro sujetador del frasco de recolección.	Para esta versión del intercambiador ya se obtuvo información suficiente para fijar estas piezas, por lo que se hizo. Para esto, en la escuadra sujetadora del frasco de recolección se colocó otra pared como límite, reduciendo material en la pared con la que ya se contaba, esta pared impedía el movimiento entre la escuadra sujetadora del frasco de recolección con la paleta del carrusel, unidos por un tornillo, el modelo de esta pieza se muestra en la Figura 11. En cuanto a la unión de la escuadra con el aro, se optó por invertir la manera en que se introducía el tornillo, colocando una cavidad para mantener la tuerca estática por debajo. Mientras, del lado del aro sujetador, se colocaron paredes limitantes que rodearon parte de la escuadra sujetadora del frasco de recolección, impidiendo el movimiento entre estas dos piezas; en tanto que, se ajustó el diámetro para que el frasco pudiera introducirse con facilidad, y a su vez, quedara fijo y alineado. Por otro lado, se decidió agregar una base para el frasco de recolección que le diera mayor soporte, uniéndolo con la escuadra bajo el mismo principio de paredes limitantes y un tornillo, ocupando poco material; el modelo de este soporte se ilustra en la Figura 12.

Tabla 18: Solución para el punto crítico 4 de la primera iteración del ciclo.

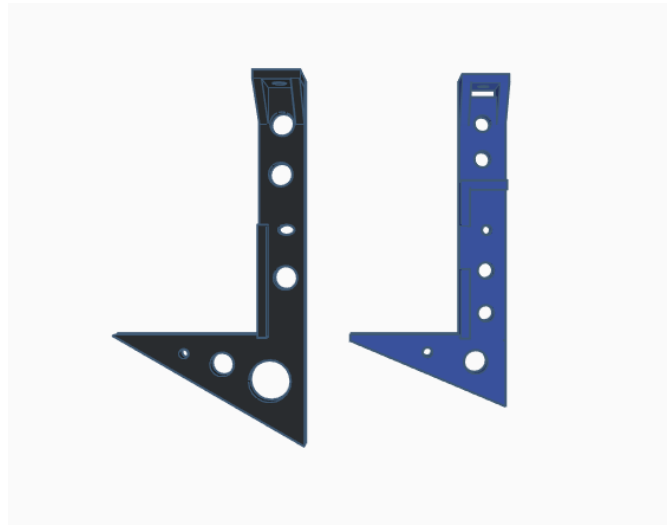


Figura 11: Modelos de la escuadra sujetadora del frasco de recolección. En la izquierda se muestra la pieza anterior y en la derecha la pieza solución.

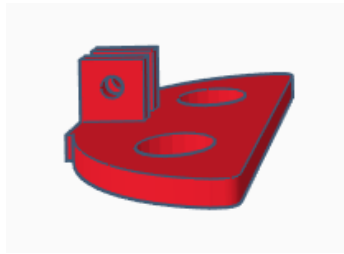


Figura 12: Modelo del soporte para el frasco de recolección.

No. Punto crítico	Punto crítico	Operación efectuada
5	Ancho de la cavidad del eje del carrusel y de la sección del émbolo donde gira la cabeza del tornillo del carrusel	Para ambos casos, se agrandó el ancho. En el modelo del eje del carrusel se definió un ancho de la cavidad de 12.8 [mm], que al tomar en cuenta el grosor del material extruido por la impresora 3D, la cavidad obtenida permitiría ingresar la tuerca. En tanto que, el incremento del ancho de la sección del émbolo permitiría que la cabeza del tornillo girara.

Tabla 19: Solución para el punto crítico 5 de la primera iteración del ciclo.

No. Punto crítico	Punto crítico	Operación efectuada
6	Fijación del cople para motor	Se buscó impedir que el cople se recorriera, colocando una capa de material entre el orificio con la forma del eje del motor y el orificio con la forma de la cabeza del tornillo. El modelo se observa en la Figura 13.

Tabla 20: Solución para el punto crítico 6 de la primera iteración del ciclo.



Figura 13: Modelo del cople para motor.

No. Punto crítico	Punto crítico	Operación efectuada
7	Fijación del servomotor al porta servomotor	Se modificó el porta servomotor para que tuviera dos puntos de sujeción en lugar de uno, tal como se muestra en la Figura 14.

Tabla 21: Solución para el punto crítico 7 de la primera iteración del ciclo.



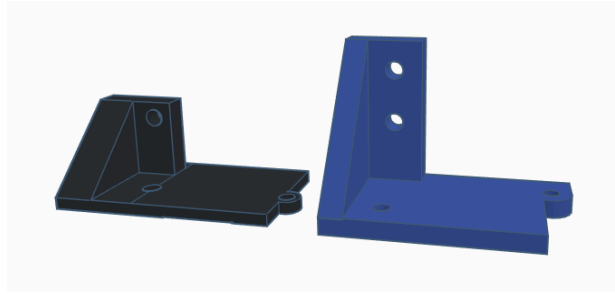


Figura 14: Modelos del porta servomotor. En la izquierda se muestra la pieza anterior y en la derecha la pieza solución.

No. Punto crítico	Punto crítico	Operación efectuada
8	Ampliación del espacio entre los orificios de las paletas de unión del eje del carrusel	El espacio para introducir los tornillos entre las paletas cortas del eje del carrusel se amplió. Para hacerlo se optó por alargar las paletas cortas del eje del carrusel proponiendo una figura que ocupara poco material, además, compensando parte de este material agregado al reducir el del centro de la pieza, el boceto se observa en la Figura 15. Cabe mencionar que, a esta pieza se adhirió el engrane para girar el carrusel, ya que al tener estas piezas separadas, se aflojaban y el movimiento no se transmitía correctamente.

Tabla 22: Solución para el punto crítico 8 de la primera iteración del ciclo.

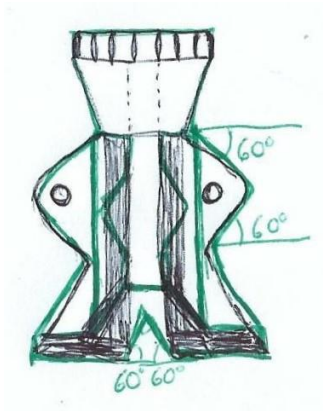


Figura 15: Boceto del eje del carrusel.

No. Punto crítico	Punto crítico	Operación efectuada
9	Sujeción del sistema dentro de la cabina de la trampa	Se buscó fabricar una nueva opción, la cual permitiera ajustar el sistema a diferentes alturas, con la posibilidad de deslizarse sobre la estructura de la trampa sobre la cual se fijaría, para colocarlo a una distancia deseada del cono de red metálica. Cabe notar que, la estructura de la trampa está fabricada con perfiles de 1 [in]. Los bocetos creados se muestran en la Figura 16. Finalmente, se seleccionó la opción que formó el modelo ilustrado en la Figura 17 y la Figura 18. Es necesario mencionar, que esta opción sujeta al sistema a través de tornillos que atraviesan la pieza tapa - sujetador, el modelo de esta última pieza se observa en la Figura 19.

Tabla 23: Solución para el punto crítico 9 de la primera iteración del ciclo.

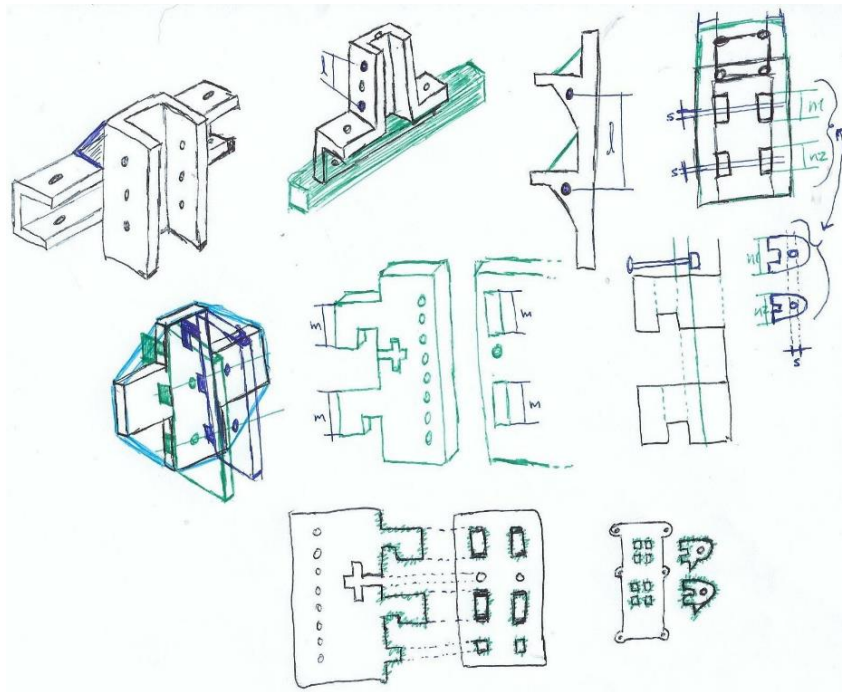


Figura 16: Bocetos de opciones de sujeción del sistema.

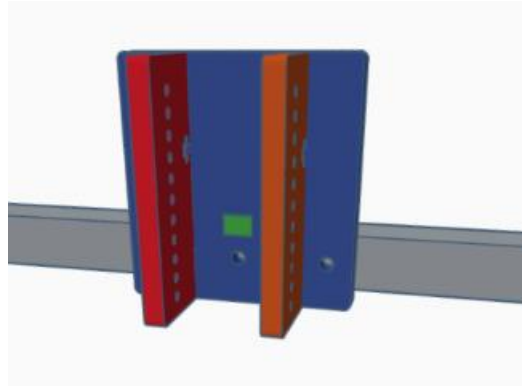


Figura 17: Modelo de la opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Los orificios son para la introducción de tornillos. Parte frontal.

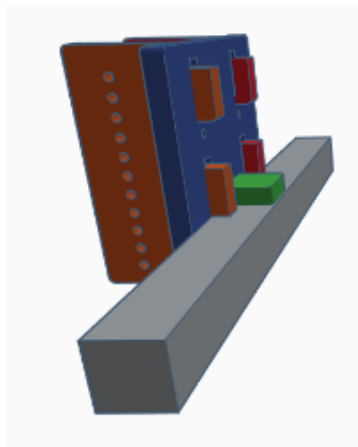


Figura 18: Modelo de la opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Los orificios son para la introducción de tornillos. Parte posterior.

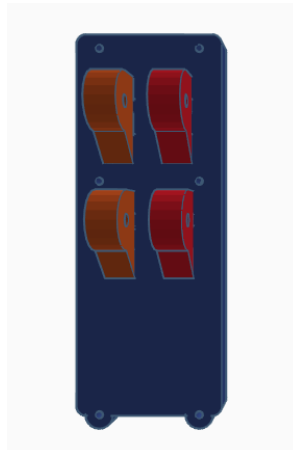


Figura 19: Modelo de tapa - sujetador. Esta pieza es la que une al sistema con la opción de sujeción seleccionada.

Antes de continuar con la solución de los puntos críticos restantes, es necesario definir los estados del frasco, “arriba” y “abajo”.

Unido al extremo de radio menor del cono de red metálica y a la estructura de la trampa, existe una pieza cilíndrica con boca cónica (Figura 20), a través de la cual desembocan los insectos al frasco de recolección. Al colocar un frasco por debajo de la pieza anterior, éste se encuentra en estado “arriba”, cuando no hay ningún espacio entre la boca del frasco y la base de radio menor de la pieza. En tanto que, el estado del frasco es “abajo” cuando existe el espacio. Ambos estados se ilustran en la Figura 21.

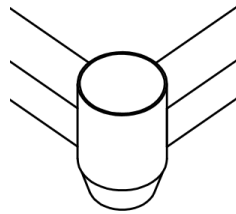


Figura 20: Pieza cilíndrica con boca cónica, unida a la estructura de la trampa.

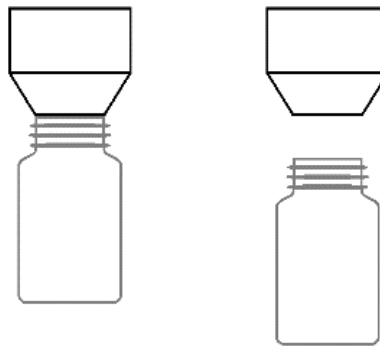


Figura 21: Estados del frasco. A la izquierda se muestra el estado “arriba”; a la derecha el estado “abajo”

No. Punto crítico	Punto crítico	Operación efectuada
10	Agregar sistema de sellado	<p>Entre las propuestas, se encontró la de colocar una tapa a cada frasco mediante un mecanismo con engranes y una leva, cuyo accionamiento por un motorreductor sería preprogramado; para finalmente cerrarlo a presión; el boceto de esta opción se muestra en la Figura 22. Para comprobar que tan viable era esta opción, una serie de tapas a presión se diseñaron y manufacturaron con ayuda de la impresora 3D, como se observan en la Figura 23 y la Figura 24, probando si era posible generar suficiente presión para sellar el frasco sin dañar alguna parte del intercambiador, además de que las tapas evitarían la salida del fluido. Como resultado, esta opción fue descartada debido a que, para evitar la salida del fluido, se debía utilizar la tapa que mayor presión necesitaba para sellar el frasco. Al final, se determinó utilizar una opción más sencilla, que consistió en una tapa en general para todos los frascos. Sin embargo, la tapa debía cubrir cinco de los seis frascos cuando el estado del sexto frasco fuera “arriba”, por lo cual la mejor opción para fijar la tapa fue a la pieza de la Figura 20. Para este objetivo, se diseñó un cople que encajara con la tapa y la parte cónica de la pieza mencionada, uniendo este cople con la parte cilíndrica de la pieza usando un trozo de manguera recta servicio pesado tipo Charter, de 14.4 [mm] (1 3/4 [in]) de diámetro y presión de trabajo de 2.11 [kg/cm²] (38 [lb/in²]), y dos abrazaderas de media presión sin fin.</p> <p>Para el cople en cuestión, se realizaron bocetos donde se propuso el uso de tornillos para fijar la tapa, tal como se ilustra en la Figura 25. En cuanto a la tapa se decidió colocar tres orificios, uno para introducir el cople, otro que atravesaría al sistema sin intervenir con su funcionamiento y, por último, otro para poder extraer los frascos, incluyendo su propia tapa. Se consideró poner el último orificio en un espacio entre dos frascos, de tal manera que, para extraer el frasco, la alineación a este orificio impidiera la alineación de cualquier otro frasco con el cono de red metálica, esto resolvía el problema de evitar que los frascos se llenaran de agua al llover puesto que en esta posición se mantenían sellados todos los frascos. El modelo de esta solución propuesta se muestra en la Figura 26. Es necesario mencionar que el material de la tapa debía ser rígido, y a la vez, debía ser capaz de sellar los frascos al contacto, por ello, se decidió utilizar dos capas de materiales, una rígida y otra capaz de crear un efecto de impermeabilidad; de las tres opciones pensadas (corcho, foami y polifoam) se seleccionó el polifoam¹⁰, cuya característica de impermeabilidad funcionó mejor que los otros materiales al probarlos.</p>

Tabla 24: Solución para el punto crítico 10 de la primera iteración del ciclo.

¹⁰ Polifoam: conocido como *foam* de polietileno, está compuesto por celdas cerradas con aire atrapado en su interior. Provee acolchonamiento, por lo que es capaz de absorber vibraciones, compresiones y golpes (INNOVAPACK, 2012).



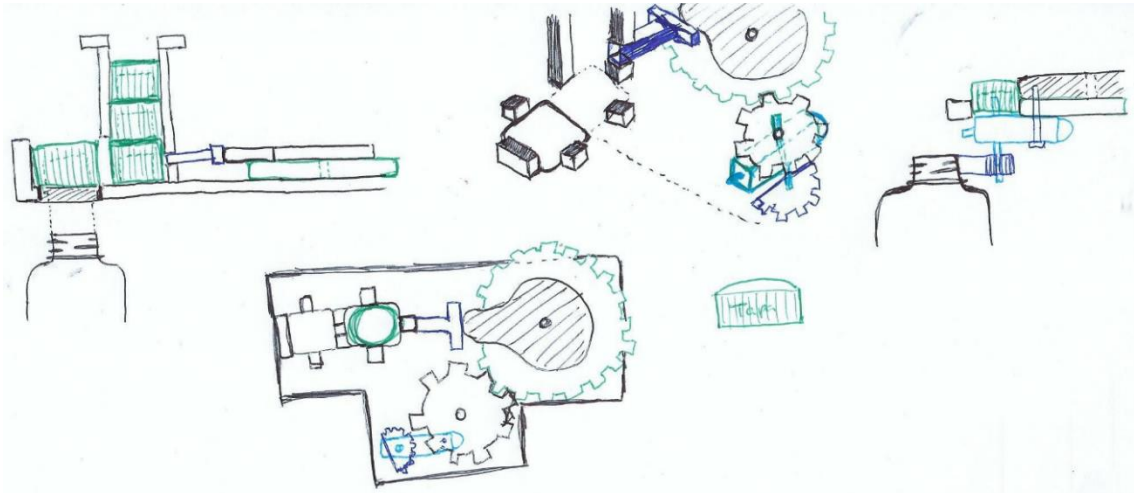


Figura 22: Boceto de opción para sellar los frascos con tapas a presión.

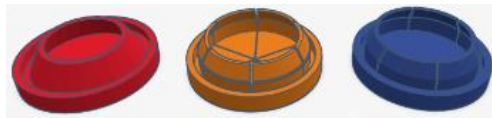


Figura 23: Modelos de las diferentes versiones de las tapas a presión para sellar los frascos.



Figura 24: Fotografía de las diferentes versiones de las tapas a presión para sellar los frascos.

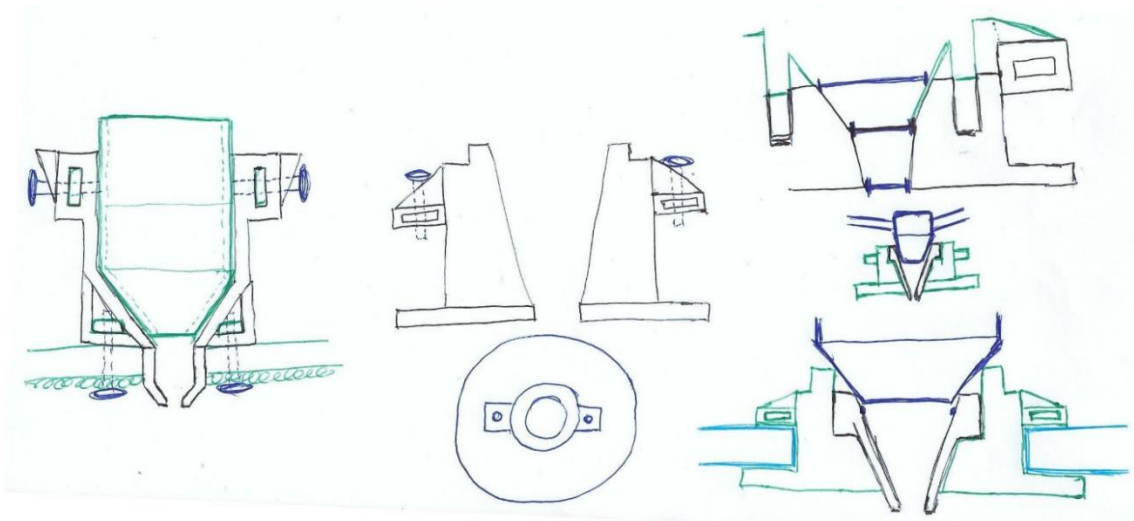


Figura 25: Bocetos de cople para unir la tapa.

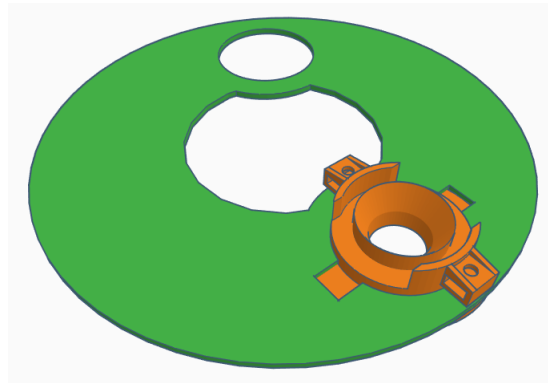


Figura 26: Modelo de la tapa con el cople.

No. Punto crítico	Punto crítico	Operación efectuada
11	Caracterización del ángulo del servomotor	Se definieron los valores de la señal que dicho servomotor recibiría para posicionarse en los ángulos deseados, para mover los engranes y definir el lugar de posición de cada frasco.

Tabla 25: Solución para el punto crítico 11 de la primera iteración del ciclo.

Antes de describir la solución al punto crítico siguiente, es necesario describir los componentes que se agregaron al subsistema Electrónico: sensores de efecto Hall.

Estos componentes se encargan de detectar o medir campos magnéticos gracias al efecto Hall. La señal de salida que se obtiene con este sensor es de dos tipos:

- Lineal: Proporciona un voltaje proporcional al campo magnético que está captando. Se trata de una señal analógica.
- Digital: Proporciona el voltaje equivalente a dos estados de acuerdo con un umbral: detecta campo magnético, no detecta campo magnético. Se trata de una señal digital.

Se optó por utilizar un sensor de tipo digital, U18, cuyas características esenciales se muestran en la Tabla 26.

Sensor: U18	
Característica	Descripción
Rango de voltaje de alimentación	2.5 a 20 [V]
Consumo de corriente	5.5 [mA] para un voltaje de alimentación de 3.6 V; 6 [mA] para un voltaje de alimentación de 16 [V]
Corriente en la salida	12 [mA]
Temperatura de operación	-20 a 125 [°C]
Costo	\$25.00

Tabla 26: Características del sensor de efecto Hall (U18). Tabla construida con base en la información en (Unisonic Technologies Company, 2012).



No. Punto crítico	Punto crítico	Operación efectuada
12	Conocimiento del estado del frasco (“Arriba” y “Abajo”)	<p>Se decidió utilizar los sensores de efecto Hall de tipo digital para conocer el estado en el que se encontraba el frasco de recolección; se utilizaron imanes redondos permanentes de neodimio de 8 [mm] de diámetro y 1.8 [mm] de altura. Por lo anterior, se buscaron opciones para poder montar dichos sensores en el sistema. Debido a que estos sensores no se quedarían fijos, la opción seleccionada debía cumplir con esta característica. Se decidió diseñar un modelo en forma de abrazadera a la guía - porta motor (Figura 27), siendo sujeta a través de la presión de tornillos, no obstante, la opción se descartó debido a que la distancia entre los dos sensores era fija, además, la sujeción era inestable por lo que podría cambiar de posición por las vibraciones, y existía la posibilidad de dañarse la guía - porta motor debido a los tornillos que ejercerían presión sobre ésta.</p> <p>Así, se buscó otra alternativa, donde cada sensor pudiera fijarse de manera independiente a diferentes distancias, llegando a la conclusión de utilizar un riel sobre el cual se posicionaría cada sensor. Tomando lo anterior en cuenta, se pensó en la manera de diseñar el perfil del riel con sus contrapartes (las bases de los sensores), obteniendo el boceto de la Figura 28. Finalmente, el modelo que se obtuvo con esta propuesta se muestra en la Figura 29.</p> <p>Cabe mencionar que, la forma que tuviera el porta - sensores, afectaría la forma de la guía - porta motor, esto se puede observar en los diferentes modelos de las opciones anteriores.</p>

Tabla 27: Solución para el punto crítico 12 de la primera iteración del ciclo.

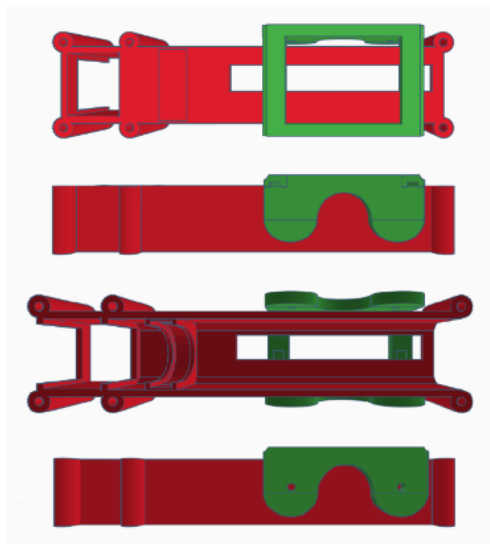


Figura 27: Modelo del porta - sensores vertical en forma de abrazadera montado sobre la guía - porta motor. Se muestra el modelo, de abajo hacia arriba: parte posterior, parte inferior, parte frontal y parte superior.

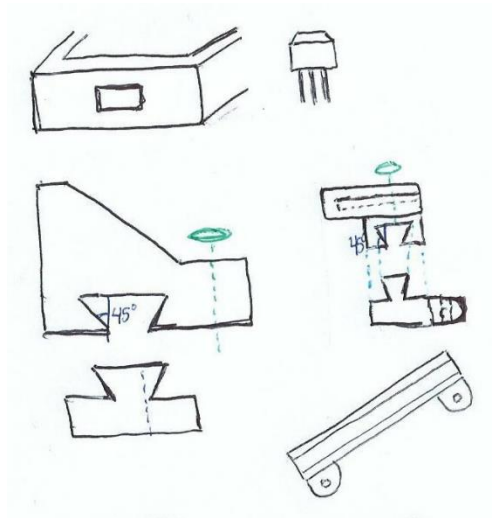


Figura 28: Boceto del porta - sensores utilizando un riel.

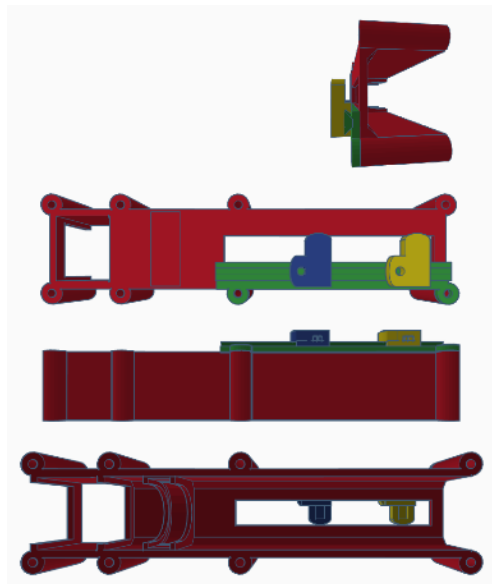


Figura 29: Modelo del porta - sensores vertical utilizando un riel montado sobre la guía - porta motor. Se muestra el modelo, de abajo hacia arriba: parte inferior, parte frontal, parte superior y parte lateral.

Previo a la descripción de la siguiente solución, es necesario recordar que el extractor se debe apagar automáticamente en ciertas circunstancias (cuando sobrepase los 30 [A] de corriente o los 40 [°C] de temperatura; al llover; si está en movimiento el intercambiador automático de muestreo). Para lograr esto es necesario controlar el apagado del motor de CA a través de la SBC utilizando un relevador. Se seleccionó un relevador de estado sólido, SSR (Solid State Relay), modelo 40 DA, ya que soporta hasta 40 A (se consideraron 30 A de corriente alterna que se podrían tener en la salida del relevador) y se controla con voltaje de corriente directa (característica requerida debido a que sería controlado por la SBC). Algunas de sus características se encuentran en la Tabla 28.

Relevador: SSR - 40 DA	
Característica	Descripción
Rango de voltaje de alimentación	3 a 32 [VDC]
Corriente de alimentación para 12 [V]	7.5 [mA]
Rango de voltaje en la salida	24 a 380 [VAC]
Corriente nominal en la salida	40 [A]
Tiempo de respuesta	10 [ms]
Temperatura de operación	-20 a 80 [°C]
Costo	\$158.00

Tabla 28: Características del relevador de estado sólido (SSR - 40 DA). Tabla construida con base en la información en (FOTEK, s.f.).

Además, se recomienda utilizar un disipador de calor de aleación de aluminio, como los de la Figura 30 y la Figura 31.



Figura 30: Disipador de calor para SSR. Obtenida de (Alibaba, 2019).



Figura 31: Disipador de calor para SSR. Obtenida de (Banggood, 2019).

No. Punto crítico	Punto crítico	Operación efectuada
13	Voltaje y corriente	<p>Se realizó la Tabla 30 considerando los valores de corriente y voltaje de todos los componentes del sistema. Se seleccionaron los voltajes de alimentación de acuerdo con los necesarios para que cada componente funcionara y de acuerdo con los 3.3 [V] y 5 [V] que puede ofrecer la SBC. Para algunos componentes no se tenían los valores exactos de corriente correspondientes, por lo que se tomaron los valores máximos o inmediatos superiores, o se obtuvieron experimentalmente. Un ejemplo es el puente H, cuyo valor de corriente total consumida por el circuito integrado se obtuvo experimentalmente, se realizaron pruebas combinando diversas situaciones, que incluían la habilitación e inhabilitación de los pines EN y el giro de uno o dos motores hacia ambos lados; el valor máximo que se obtuvo fue de 47.58 [mA] teniendo habilitados ambos pines EN, mientras que para un pin EN habilitado la corriente fue de 31.96 [mA].</p> <p>Es necesario destacar que en la tabla se incluyen dos LED cuyas características no se definieron con anterioridad, pero que fue necesario que se tomaran en cuenta, ya que se utilizaron para solucionar el punto crítico número 15.</p> <p>Como se observa, con esta tabla se probó que las conexiones se pueden llevar a cabo sin dañar a la tarjeta, la única consideración importante fue alimentarse de los 5 [V] de los pines dedicados específicamente a otorgarlos. En cuanto a la corriente se consume un total de 60.7 [mA], lo que está por debajo de la corriente máxima en los GPIO, incluso por pin el valor es menor a los 50 [mA], por lo que nuevamente se comprobó que los componentes pueden funcionar sin problema alimentándose de la SBC.</p> <p>Así mismo, esta tabla fue de utilidad para conocer que fuentes de alimentación eran necesarias, para los componentes que en la tabla se indica la alimentación por una fuente externa, se dividieron en 3 fuentes externas diferentes: una fuente de 5 [V] y 2.5 [A] para la SBC; una fuente de 127 [VCA] y 30 [A] para alimentar el extractor; y una fuente de 5 [V] y 2 [A] para alimentar los motores.</p>

Tabla 29: Solución para el punto crítico 13 de la primera iteración del ciclo.



Componente	Voltaje de alimentación [V]	Corriente de alimentación [mA]	Fuente de alimentación
SBC	5	2500	Externa
Extractor	127 [VCA]	14200	Externa
Relevador	3.3	7.5	SBC
Motorreductor de plástico	5	250	Externa
Puente H	5	31.96	SBC
Servomotor	5	1200	Externa
BMP180	3.3	1	SBC
DHT11	5	2.5	SBC
LM35	5	0.06	SBC
Circuito para ECS1030	3.3	0.33	SBC
U18	2(3.3)	2(6)	SBC
MCP3008	3.3	0.55	SBC
LED Amarillo	1.95	3	SBC
LED Azul	3.04	1.8	SBC

Tabla 30: Voltaje y corriente de alimentación de los componentes para la primera iteración del ciclo. En la última columna se indica como sería alimentado el componente, por la tarjeta o por una fuente externa. La información de los LED se obtiene en (finsuka, 2017), se utilizan dos resistores debido a que se requieren menos de 3.3 [V] para su encendido.

Tabla elaborada con base en las características, descritas con anterioridad, de los componentes.

No. Punto crítico	Punto crítico	Operación efectuada
14	Temperatura	Con base en la información recolectada de cada componente, de las hojas de especificaciones o de sus proveedores, se verificó que los componentes fueran aptos para funcionar dentro del rango de temperatura de 16 [°C] a 36 [°C].

Tabla 31: Solución para el punto crítico 14 de la primera iteración del ciclo.

No. Punto crítico	Punto crítico	Operación efectuada
15	Verificación de encendido y de conexión a la red	Se decidió fijar los dos LED, mencionados en la Tabla 30, para que fueran visibles para el usuario, fuera de la caja donde se encontraría el circuito. Para el encendido se colocó el LED amarillo, en tanto que, se situó el LED azul para indicarle al usuario la integración de la Raspberry Pi a la red.

Tabla 32: Solución para el punto crítico 15 de la primera iteración del ciclo.



No. Punto crítico	Punto crítico	Operación efectuada
16	Número de frasco	El objetivo de este punto era utilizar la caracterización del ángulo del servomotor para determinar el número de frasco que se encontraba debajo del cono metálico para recolectar la muestra. Esta caracterización se llevó a cabo hasta la etapa de Verificación.

Tabla 33: Solución para el punto crítico 16 de la primera iteración del ciclo.

Aunado a las operaciones efectuadas para los puntos críticos siguientes, es necesario mencionar que, si bien en esta iteración se seleccionó el sistema operativo de la SBC, las bibliotecas necesarias para programar los actuadores y los sensores del sistema, y otros elementos de la arquitectura de IoT, se describen en los apartados Bibliotecas para los actuadores y sensores, MQTT, y Framework, del capítulo 5: Integración final.

No. Punto crítico	Punto crítico	Operación efectuada
17	Forma de obtención de los datos de los sensores y del estado del sistema	Se seleccionaron las bibliotecas que serían útiles para para escribir el código que obtuviera los datos otorgados por cada sensor y los datos del sistema.

Tabla 34: Solución para el punto crítico 17 de la primera iteración del ciclo.

No. Punto crítico	Punto crítico	Operación efectuada
18	Definición de los elementos de la capa de Red	<p>Como ya se mencionó, se consideró la SBC como <i>Gateway</i>. Para seleccionar el protocolo de red, fue necesario conocer su clasificación. Los protocolos de red se pueden clasificar en siete capas según el modelo OSI (<i>Open System Interconnection</i>, en español: Interconexión de Sistemas Abiertos). Enfocándose en la capa 7 de este modelo, “Nivel de Aplicación”, existen los protocolos de comunicación que se pueden dividir en diferentes categorías dependiendo de sus características. Tomando en cuenta la arquitectura de su mensajería se pueden tener dos categorías:</p> <ul style="list-style-type: none"> • Cliente - Servidor: Se requiere que el cliente mande solicitudes al servidor y este responda a sus pedidos. La información se queda almacenada en el servidor hasta que el cliente lo solicite. • Suscriptor - Publicador: Existe un intermediario que se ocupa de recibir la información que el publicador manda, almacenándola hasta que el suscriptor la solicita. <p>Aron Semle (Semle, 2016) menciona que la arquitectura de mensajería Suscriptor - Publicador es más útil que la de Cliente - Servidor cuando se desconoce la infraestructura de la red, además, es más escalable gracias a la división que los hace</p>



		<p>actuar de manera independiente, es decir, el publicador puede estar continuamente publicando información aun si el suscriptor no busca obtenerla. Su principal desventaja ante la otra arquitectura es la inseguridad y el formato de los mensajes que debe estar acordado entre todos los clientes.</p> <p>Algunos ejemplos de los protocolos de comunicación con ambas arquitecturas de mensajería se observan en la Tabla 36. Cada uno de estos protocolos tiene sus ventajas y desventajas, por lo que no se puede definir uno como el mejor ante cualquier aplicación.</p> <p>Para este proyecto, se seleccionó el protocolo de comunicación MQTT debido a las características mencionadas, además de que, es simple, popular (por lo que se puede obtener amplia información sobre su uso y aplicación) y utiliza un ancho de banda bajo (característica útil hablando de IoT). Cabe mencionar, que utiliza un protocolo de transporte TCP¹¹ que transfiere datos de manera fiable y en primera instancia se debe tener el puerto TCP:1883 abierto en el firewall, aunque posteriormente se puede configurar otro puerto para una mayor seguridad. Retomando la información de la tabla, sigue una topología en estrella, donde el nodo central es el intermediario al que se le conoce como <i>broker</i> y los nodos periféricos son los clientes (de dos tipos: suscriptores o publicadores), que intercambian información a través de este <i>broker</i>. Esta información, en forma de mensajes, es gestionada por el <i>broker</i>, que la filtra de acuerdo con el <i>topic</i> al que pertenezca, indicado por los clientes, de esta manera, los clientes publicadores pueden enviar mensajes indicando diferentes <i>topics</i>, y los suscriptores sólo obtienen la información referente a los <i>topics</i> a los que se hayan suscrito previamente.</p> <p>Un punto importante respecto a los <i>topics</i>, es que pueden tener una estructura jerárquica, donde un <i>topic</i> puede contener a varios, así, se puede obtener la información de todos los <i>topics</i> hijos cuando el cliente está suscrito a un <i>topic</i> padre, o bien, se puede obtener sólo la información de un <i>topic</i> hijo en particular. Finalmente, para el último elemento de la capa de Red, una recomendación era asegurar la comunicación con TLS (<i>Transport Layer Security</i>, en español: Seguridad en Capas de Transporte), pero al implicar un costo computacional alto, se optó por la definición de un usuario y contraseña para el envío y recepción de los mensajes.</p>
--	--	---

Tabla 35: Solución para el punto crítico 18 de la primera iteración del ciclo.

¹¹ TCP: *Transmission Control Protocol*, en español se traduce como Protocolo de Control de Transmisión. Este protocolo “se encarga de crear conexiones entre sí para que se cree un flujo de datos. Este proceso garantiza que los datos sean entregados en destino sin errores y en el mismo orden en el que salieron. También se utiliza para distinguir diferentes aplicaciones en un mismo dispositivo” (Qloudea, 2017).



Protocolo de Red	Arquitectura de Mensajería	Descripción
OPC UA (Unified Architecture)	Cliente - Servidor	Permite una comunicación “amigable” con sensores y PLC’s a partir de sistemas existentes como MES (<i>Manufacturing Execution System</i>) y SCADA (<i>Supervisory Control and Data Acquisition</i>). Es seguro gracias a sus mensajes firmados y encriptados.
HTTP (Hypertext Transfer Protocol)	Cliente - Servidor	Durante años ha sido el más utilizado. Sobre todo, si se requiere enviar grandes cantidades de información, sin embargo, para mensajes pequeños es ineficiente. Gira en torno al modelo REST (<i>Representational State Transfer</i>). La comunicación se debe asegurar con HTTPS (<i>Hypertext Transfer Protocol Secure</i>).
MQTT (Message Queuing Telemetry Transport)	Suscriptor - Publicador	Durante el transcurso de los años ha tomado popularidad. Se utiliza cuando se tiene un ancho de banda bajo. Su intermediario está centralizado, mostrando una topología en estrella. La comunicación se debe asegurar con TLS.
CoAP (Constrained Application Protocol)	Cliente - Servidor	Es similar a HTTP. Útil para proveer la compatibilidad de HTTP con una mínima carga. Se prefiere cuando el utilizar HTTP demanda ancho de banda intenso. La comunicación se recomienda asegurar con DTLS (<i>Datagram Transport Layer Security</i>).
DDS (Data Distribution Service)	Suscriptor - Publicador	Su intermediario está descentralizado. Es una buena opción cuando se busca la entrega de información de forma confiable y en tiempo real. Se recomienda como protocolo secundario.
AMQP (Advanced Message Queuing Protocol)	Suscriptor - Publicador	Garantiza transacciones completas, aunque esta ventaja no es necesariamente requerida cuando se habla de IoT, ya que es un protocolo “pesado”.

Tabla 36: Protocolos de red. Tabla construida con base en la información de Aron Semle (Semle, 2016).



No. Punto crítico	Punto crítico	Operación efectuada
19	Aplicación web y aplicación móvil	Se probó un <i>framework</i> ¹² , aunque no fue unificado al sistema debido a que primero se debía comprobar el funcionamiento de los elementos de la primera y segunda capa de IoT.

Tabla 37: Solución para el punto crítico 19 de la primera iteración del ciclo.

Las piezas anteriormente descritas se fabricaron con una impresora 3D, en específico se utilizó el modelo *Ultra 3D Printer* de la marca *PowerSpec*. Además, se utilizó el software original de la impresora, *PowerPrint*, para crear el tipo de archivo necesario para imprimir y para visualizar una aproximación de cómo se llevaría a cabo la manufactura de cada capa, permitiendo la corrección previa a la impresión de la pieza o de sus propiedades. El proceso que se llevó a cabo se muestra en la Figura 32.

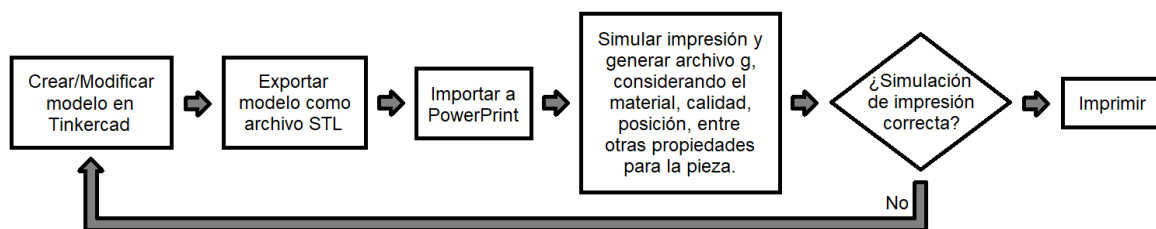


Figura 32: Proceso para fabricar piezas utilizando manufactura aditiva. El archivo STL es “una versión simplificada de un archivo CAD, con menos cantidad de información y con la geometría final resumida, optimizada, reducida al mínimo posible para que no se pierda la geometría y estructura del objeto” (Regidor, 2016). El archivo g es el tipo de archivo necesario para imprimir, contiene las “instrucciones para la máquina que determinan medidas como la velocidad, la dirección y la profundidad a la que debe moverse” (ProyectosCAD, 2018).

Como ya se mencionó, parte de las restricciones al manufacturar piezas mediante manufactura aditiva es que las piezas no tengan techo y un ángulo de inclinación superior a los 45°, no obstante, para algunas piezas esto no fue posible. Una solución era utilizar como material de impresión un filamento lavable, pero implicaba un mayor costo, por lo que se decidió insertar piezas que pudieran ser removidas al término de la impresión; algunas de estas piezas fueron elaboradas con plastilina forradas con cinta de enmascarar (*Masking Tape*), otras eran herramientas o tornillos forrados de la misma manera. En la Figura 33 se observan algunas de las piezas utilizadas.

¹² *Framework*: es “un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación” (Jordisan, 2006); “puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto” (EcuRed, 2019).

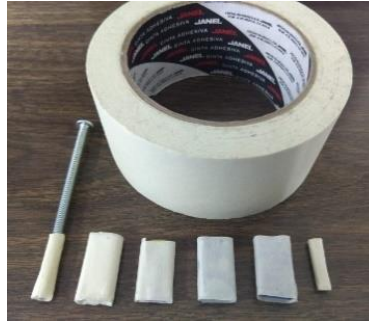


Figura 33: Piezas auxiliares para imprimir en 3D, elaboradas a partir de plastilina o herramientas forradas con el *Masking Tape* mostrado.

Para las piezas de la sujeción del sistema, así como la tapa del sellado de frascos, se optó por utilizar corte láser; seleccionando como material acrílico de 9 [mm] para la sujeción del sistema, y acrílico de 3 [mm] para la tapa del sellado de frascos. Debido a que se requería un tipo de archivo DXF para poder mandar las piezas al corte láser, se utilizó un software de edición de gráficos, Inkscape, para obtenerlo. Así, similar al proceso de manufactura anterior, se llevó a cabo el proceso que se observa en la Figura 34.

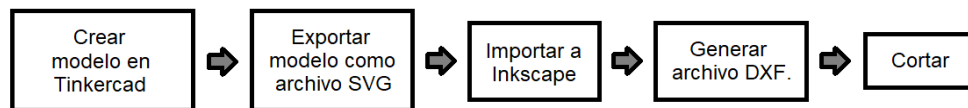


Figura 34: Proceso para fabricar piezas utilizando corte láser. SVG es un “formato gráfico basado en XML para crear archivos vectoriales en 2D, con un lenguaje de marcado por medio de etiquetas” (ATGU, 2014). DXF es un “formato de dibujo desarrollado por Autodesk como un tipo de formato universal para almacenar modelos CAD” (iX Corp, 2018).

Las fotografías de las piezas manufacturadas y ensambladas se muestran en el ANEXO A.

Por otra parte, se escribió el código necesario para complementar el subsistema de Software. Sin embargo, no se describe debido a que en esta primera iteración del ciclo:

1. Se hicieron programas no finales, por ejemplo, se hizo un programa por cada componente para comprobar su funcionamiento por separado; posteriormente para las pruebas se hicieron tres programas dividiéndolos en uno para los sensores de la versión anterior, otro para el motorreductor y los sensores de efecto Hall, y el último para el servomotor.
2. Hubo modificaciones en el código en las iteraciones posteriores.
3. Primero se debía comprobar el funcionamiento de los elementos de la primera y segunda capa de IoT, por lo que sólo se probaron algunos elementos de la arquitectura de IoT relacionados con el subsistema de Software, y no se definieron por completo, ni aún fueron incluidos en el sistema. Por ejemplo, se probó el envío y recepción de mensajes con el protocolo de red MQTT entre la SBC y una computadora de escritorio, sólo para conocer y aprender sobre su funcionamiento.



Etapa de Verificar

Para comprobar el funcionamiento del sistema, se incorporaron los tres subsistemas, ensamblando todas las piezas manufacturadas y los componentes electrónicos.

Para fijar el intercambiador de la manera más parecida a como se instalaría en la trampa, se utilizó una mesa del Laboratorio de Electrónica y Automatización para Industria 4.0, en el IIMAS. De esta manera, se colocó como se muestra en la Figura 35, y se hicieron las conexiones necesarias para energizarlo. Se accedió a la SBC para ejecutar cada comando definido para mover los actuadores y obtener los valores de los sensores.



Figura 35: Intercambiador fijado a una mesa del Laboratorio de Electrónica y Automatización para Industria 4.0, en el IIMAS.

Los sensores se pudieron probar adecuadamente, a excepción del sensor de corriente, ya que el motor no se encuentra en el laboratorio.

Debido a que no se simuló el embudo de la trampa, no se colocó la tapa para sellar los frascos. A pesar de lo anterior, se probó el movimiento del servomotor para verificar que la caracterización realizada fue correcta, se encontró que algunos frascos se encontraban desfasados. En cuanto al movimiento del motorreductor, éste elevó y bajo los frascos correctamente y con una velocidad lenta, deteniendo el movimiento de acuerdo con los sensores de efecto Hall conectados.

Las pruebas anteriores resultaron favorables, sin embargo, los frascos no contenían ningún líquido. A continuación, se realizaron las mismas pruebas con los frascos llenos de agua. Como resultado se observó un nuevo desfase con el servomotor; en tanto que, para el motorreductor, el movimiento no se llevó a cabo.

Tomando como antecedente que la versión del intercambiador, anterior a esta investigación, estuvo en funcionamiento algunos meses, no se verificaron los valores máximos de masa que podría mover cada actuador del sistema. En las hojas de especificaciones, los valores máximos de masa que podría mover cada actuador indican que el servomotor puede mover 9.4 [kg], cuando esta masa se encuentra a máximo 1 [cm] de su eje, mientras que el motorreductor soporta 800 [g]. Si bien, al principio no se contaba con el equipo para medir masa, se consiguió; de igual manera se solicitó al personal del INIFAP la masa del frasco de recolección con la solución de alcohol, que fue de 186 [g]. Así, se midieron las piezas que debía mover cada actuador, considerando los frascos llenos. Se midió 1.328 [kg] para el servomotor y 1.471 [kg] para el motorreductor; encontrando la razón de falta de movimiento de este último durante las pruebas.



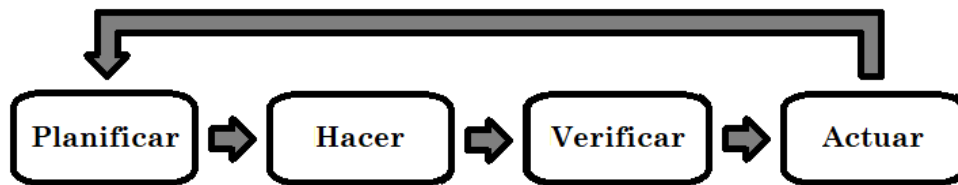
Etapa de Actuar

Las operaciones realizadas para los puntos críticos dieron resultados suficientes, durante las pruebas en el Laboratorio de Electrónica y Automatización para Industria 4.0, para que se consideraran definitivas las mejoras. No obstante, no se consideró como punto crítico la masa que soportaba mover el motorreductor, por lo que falló durante las pruebas y se debió descartar su uso, dando continuidad a la siguiente iteración del ciclo.



Capítulo 3 - Segunda iteración del ciclo

Este capítulo incluye la segunda iteración del Ciclo de Deming. En esta iteración se buscó corregir los errores de la iteración anterior, incluyendo las cuatro etapas del ciclo: Planificar, Hacer, Verificar y Actuar.



Etapa de Planificar

Debido a que el uso del motorreductor de plástico se descartó, nuevamente se consideraron los puntos críticos de la etapa de Planificar de la iteración anterior, referentes a los límites de funcionamiento (temperatura, voltaje y corriente), además, se consideró otro límite de funcionamiento: la masa. Asimismo, se retomó el punto crítico correspondiente al voltaje y corriente del prototipo en general. Por otra parte, dependiendo del nuevo actuador que se utilizaría, se modificarían dos piezas: el cople para motor y la guía - porta motor; por lo que se definieron dos puntos críticos más, de acuerdo con la manera como serían afectadas estas piezas. Estos puntos y su descripción se muestran en Tabla 38.

No. Punto crítico	Punto crítico	Descripción
1	Nuevo actuador dentro de los límites de funcionamiento	Debido al problema con el motorreductor de plástico, el nuevo actuador debía elevar y descender los frascos de recolección llenos sin problema. Asimismo, debía funcionar en el Campo Experimental del INIFAP de Mochochá, donde la temperatura varía entre 16 [°C] y 36[°C]; además de considerarse su voltaje y corriente de alimentación.
2	Voltaje y corriente	Dadas las restricciones que tiene la Raspberry Pi Zero W con respecto al voltaje y corriente que puede entregar en sus GPIO, nuevamente se debía verificar que todos los componentes se alimentaran dentro de estos límites.
3	Afectación en la guía - porta motor	Esta pieza tenía una cavidad donde se ingresaba y se sujetaba el motorreductor de plástico, fue diseñada específicamente para este motorreductor. El cambio de actuador afectó la utilidad de la pieza, puesto que podía cumplir con su función de guiar al émbolo, pero no sostener al nuevo actuador debido a sus dimensiones.
4	Afectación en el cople para motor	Este cople permitía conectar el eje del motor con la cabeza del tornillo sin fin que se emuló, sincronizando el movimiento de ambos. El cople tenía un diseño que le permitía ajustarse al eje del motor, por lo que ahora se tenía que ajustar al eje del nuevo actuador.

Tabla 38: Puntos críticos de la segunda iteración del ciclo.

Se planeó seleccionar la opción que reemplazaría al motorreductor de plástico, utilizando los límites de funcionamiento mencionados para el primer punto crítico, considerando, además, el costo del nuevo actuador y su facilidad de compra.

En tanto que, para el segundo punto crítico, se realizaría otra tabla como la de la iteración del ciclo previo.

Para los puntos críticos restantes, se planeó generar bocetos, modelos, diagramas y/o lluvias de ideas.



Etapa de Hacer

Nuevamente, para cada punto crítico identificado en la etapa anterior, se encontró su solución como se observa en cada una de las tablas siguientes.

No. Punto crítico	Punto crítico	Operación efectuada
1	Nuevo actuador dentro de los límites de funcionamiento.	<p>Se optó por utilizar un micro - motorreductor Pololu de 100 [RPM] el cual, a diferencia del motorreductor de plástico, tiene un mayor par y consume menor corriente, además, es más pequeño (de aquí el prefijo “micro”), y a pesar de ser de metal es menos pesado; no obstante, su costo y voltaje de alimentación es mayor. Los valores de sus características se muestran en la Tabla 40. Como se observa, los valores están dentro de los límites de masa, temperatura, voltaje y corriente.</p> <p>Cabe notar que, se utilizaba una misma fuente para alimentar al servomotor y al motorreductor de plástico; sin embargo, debido a que el servomotor y el micro - motorreductor se energizan con valores diferentes, era necesario utilizar distintas fuentes de alimentación. El problema, es que no se cuenta con suficientes tomas de corriente en el lugar donde se encuentra la trampa. Así, se decidió incluir un regulador en el circuito que, si bien su principal función es mantener un voltaje constante evitando que los picos de voltaje dañen un circuito, también ofrece la posibilidad de obtener uno de los voltajes deseados a partir del otro. Se empleó un regulador de la serie LM78. Según su hoja de especificaciones (Fairchild Semiconductor Corporation, 2008), una de las características de esta serie es que se pueden obtener voltajes de salida de 5, 6, 8, 9, 10, 12, 15, 18 y 24, según el voltaje deseado será el componente de la serie seleccionado. Otra característica es su voltaje de alimentación máximo, que está definido de 35 [V] si se desea un voltaje de salida de 5 a 18 [V], y de 40 [V] si se desea un voltaje de salida de 24 [V]. El componente seleccionado de esta serie fue el LM7805, que permitió obtener 5 [V] a partir de la fuente de 9 [V], además, su rango de temperatura de operación es amplio, desde los -40 [°C] hasta los 125 [°C].</p>

Tabla 39: Solución para el punto crítico 1 de la segunda iteración del ciclo.



Motor de CD: Micro - motorreductor Pololu 100 [RPM]	
Característica	Descripción
Rango de voltaje de alimentación	6 a 12 [V]
Corriente máxima	82 [mA]
Corriente al arrancar	171 [mA]
Par permitido	6.5 [kg/mm]
Masa	10 [g]
Costo	\$110.00

Tabla 40: Características del micro - motorreductor. Tabla construida con base en las características que se obtuvieron en (Pololu Corporation, 2019) y (CDMX Electrónica, 2019a). Para conocer los valores de corriente máxima y corriente al arrancar se hicieron diferentes pruebas, colocando voltajes de 6 [V], 9 [V] y 12 [V], variando la carga, e intercalando el movimiento de un lado a otro de acuerdo con lo necesario para el proyecto.

No. Punto crítico	Punto crítico	Operación efectuada
2	Voltaje y corriente	Como se planeó, se realizó otra tabla (Tabla 42) considerando los valores de corriente y voltaje de todos los componentes en esta iteración. Con esta tabla, se comprobó que las conexiones no dañan la SBC, considerando la alimentación de los 5 [V] de los pines dedicados específicamente a otorgarlos. En cuanto a la corriente, se consume un total de 60.7 [mA], lo que está por debajo de la corriente máxima en los GPIO, incluso por pin el valor es menor a los 50 [mA], por lo que nuevamente se comprobó que los componentes pueden funcionar sin problema alimentándose de la SBC. Además, la tabla se utilizó para conocer que fuentes de alimentación eran necesarias, para los componentes que en la tabla se indica la alimentación por una fuente externa, nuevamente se dividieron en 3 fuentes externas diferentes: una fuente de 5 [V] y 2.5 [A] para la SBC; una fuente de 127 [VCA] y 30 [A] para alimentar el extractor; y una fuente de 9 [V] y 2 [A] para alimentar los motores.

Tabla 41: Solución para el punto crítico 2 de la segunda iteración del ciclo.



Componente	Voltaje de alimentación [V]	Corriente de alimentación [mA]	Fuente de alimentación
SBC	5	2500	Externa
Extractor	127 [VCA]	14200	Externa
Relevador	3.3	7.5	SBC
Motorreductor de metal 100 [RPM]	9	171	Externa
Puente H	5	31.96	SBC
Servomotor	5	1200	Regulador
Regulador	9	1200	Externa
BMP180	3.3	1	SBC
DHT11	5	2.5	SBC
LM35	5	0.06	SBC
Circuito para ECS1030	3.3	0.33	SBC
U18	2(3.3)	2(6)	SBC
MCP3008	3.3	0.55	SBC
LED Amarillo	1.95	3	SBC
LED Azul	3.04	1.8	SBC

Tabla 42: Voltaje y corriente de alimentación de los componentes para la segunda iteración del ciclo. En la última columna se indica como sería alimentado el componente, por la tarjeta o por una fuente externa. Tabla elaborada con base en las características descritas con anterioridad.

No. Punto crítico	Punto crítico	Operación efectuada
3	Afectación en la guía - porta motor	Una solución era modificar el diseño de la cavidad para sostener al actuador y manufacturar nuevamente la pieza, sin embargo, para evitar el desperdicio de la pieza anterior, se optó por diseñar otra pieza que funcionara como cople para el micro - motorreductor, acoplándolo a la guía - porta motor con la que se contaba. El modelo de esta pieza se muestra en la Figura 36.

Tabla 43: Solución para el punto crítico 3 de la segunda iteración del ciclo.

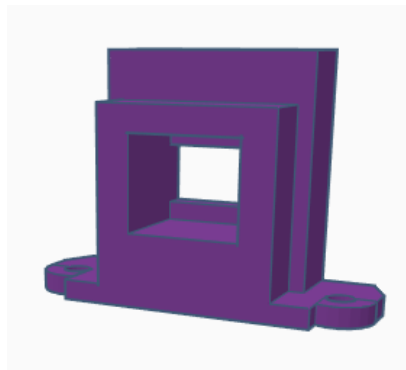


Figura 36: Porta - micro - motorreductor.

No. Punto crítico	Punto crítico	Operación efectuada
4	Afectación en el cople para motor	Debido a que la forma del eje era distinta, entre el motorreductor de plástico y el micro – motorreductor Pololu, se cambió al que se observa en la Figura 37.

Tabla 44: Solución para el punto crítico 4 de la segunda iteración del ciclo.



Figura 37: Cople para motor modificado.

Utilizando nuevamente manufactura aditiva, se fabricó la pieza diseñada como cople entre el micro - motorreductor y la guía - porta motor. De la misma manera, el cople se imprimió. Las piezas obtenidas se muestran en el ANEXO B.

Cabe notar que, no se modificó el código del motorreductor, ya que, a pesar del cambio del actuador, se probó que el código funcionaba para ambos. Para el código del servomotor, se modificaron los valores de caracterización.

Etapa de Verificar

En esta iteración del ciclo se tuvo la oportunidad de realizar dos pruebas en el Laboratorio de Electrónica y Automatización para Industria 4.0, del IIMAS, y una en la trampa instalada en Mocochá.

Para la primera prueba en el laboratorio, se fijó el intercambiador a la mesa y se hicieron las conexiones necesarias para energizarlo (Figura 38). Nuevamente, se utilizó el protocolo SSH para acceder a la Raspberry Pi y ejecutar los comandos necesarios. Se realizó la prueba del ascenso y descenso de los frascos sin agua, para lo cual el funcionamiento fue efectivo; posteriormente, se realizó la misma prueba con los frascos llenos de agua y el funcionamiento también fue efectivo.



Figura 38: Intercambiador de la segunda iteración del ciclo, fijado a una mesa del Laboratorio de Electrónica y Automatización para Industria 4.0.

Para la prueba en la trampa, se buscó fijar el intercambiador utilizando la opción de sujeción seleccionada en la primera iteración del ciclo, no obstante, el diseño fue erróneo al considerar que el intercambiador se trasladaría sobre un eje y no sobre dos, lo que provocó que ningún frasco pudiera posicionarse debajo del extremo por donde caerían los insectos, esto se observa en la Figura 39. Por esta razón, no se probó el funcionamiento del intercambiador en Mocochá.



Figura 39: Prueba de la opción de sujeción seleccionada en la primera iteración, en la trampa de Mocochá.

Con el antecedente de la prueba en la trampa, la segunda prueba en el laboratorio se realizó para analizar y encontrar todas las fallas que aún podría tener el intercambiador con respecto a los puntos críticos definidos en las etapas de planificar de ambas iteraciones. Para esto, se colocó el intercambiador de la misma manera que en la primera iteración del ciclo, para hacer más de doce iteraciones en relación con el movimiento de ambos actuadores, por lo que se detectó que, a pesar de la caracterización del servomotor, los frascos no se colocaban en su lugar todas las veces. Además, al energizar el sistema, el servomotor también era energizado y al no tener una señal, realizaba un pequeño movimiento.

Por otro lado, debido a que no se había probado la solución dada para el sellado de frascos en la etapa de Verificación de la iteración anterior, ni en Mocochá, se fijó el trozo de manguera recta de servicio pesado tipo Charter sobre una viga, y con ayuda del cople diseñado y tornillos, se fijó la tapa de acrílico y polifoam a la manguera. De manera no intencionada, en diversas ocasiones se colocó y se quitó la tapa, atornillando y desatornillando el cople, por lo que éste no soportó, y finalmente, se dañó como se muestra en la Figura 40.

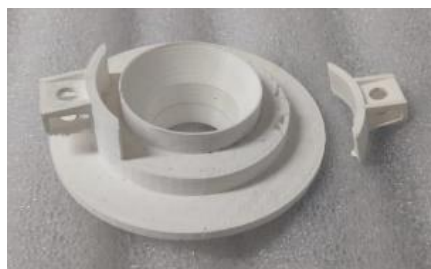


Figura 40: Cople dañado. La parte donde se introduce uno de los tornillos se separó debido a que no soportó los esfuerzos.

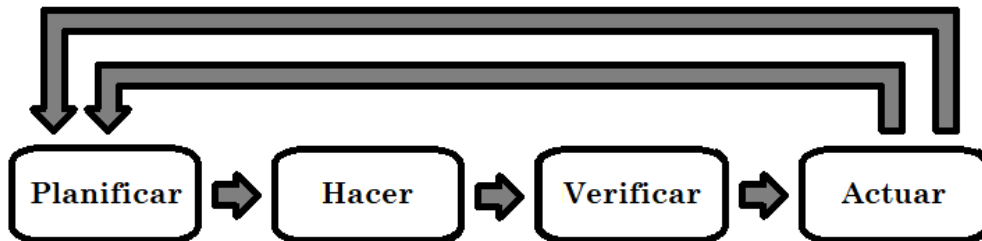
Etapa de Actuar

Con los resultados de las tres pruebas en la etapa anterior, se identificaron nuevos inconvenientes, como el cople dañado, la necesidad de diseñar una nueva opción de sujeción, y el movimiento del servomotor involuntario al energizar el sistema; este último se podía evitar si se colocaba un relevador para energizar al servomotor después de mandarle la señal, no obstante, se prefirió descartar el servomotor para esta función y corregir todo lo anterior en una nueva iteración del ciclo.



Capítulo 4 - Tercera iteración del ciclo

Con el objetivo de corregir los problemas de la iteración anterior, este capítulo incluye la tercera y última iteración, considerando las cuatro etapas del Ciclo de Deming: Planificar, Hacer, Verificar y Actuar.



Etapa de Planificar

De acuerdo con los problemas identificados en la iteración anterior, se definieron los puntos críticos de la Tabla 45.

No. Punto crítico	Punto crítico	Descripción
1	Distribución de esfuerzos en el cople para la tapa del sellado de frascos	El error que se tuvo con esta pieza fue concentrar la fuerza en dos puntos que provocaron que la pieza no soportara los esfuerzos; la dirección en que se imprimió la pieza influyó también en este aspecto.
2	Rediseño de la sujeción del sistema dentro de la cabina de la trampa	Las piezas de sujeción, diseñadas en la primera iteración, no funcionaron como se esperaba durante las pruebas dentro de la trampa, que se llevaron a cabo durante la segunda iteración.
3	Modificación de la tapa del sistema de sellado de frascos de acuerdo con el cople	El diseño del cople y la forma de sujetarlo a la tapa, influía en esta última pieza. El rediseño del cople causaría el rediseño de la tapa.
4	Sustituto de la función del servomotor	El movimiento del servomotor no fue satisfactorio para colocar los frascos en su lugar, por lo que se descartó. Además, tomando en cuenta que una de las características del servomotor era detenerse según la señal de PWM dada, indicando que un frasco estaba alineado con el cono de red metálica, se tenía que encontrar la manera de sustituir esta función.
5	Voltaje y corriente	Dadas las restricciones que tiene la Raspberry Pi Zero W con respecto al voltaje y corriente que puede entregar en sus GPIO, nuevamente se debía verificar que todos los componentes se alimentaran dentro de estos límites.
6	Temperatura y masa	Todos los nuevos componentes agregados debían ser aptos para funcionar dentro del rango de temperatura de 16° a 36°, a la vez, los actuadores debían mover la masa para la que fueron colocados.
7	Número de frasco	Aún con el cambio de actuador, se debía determinar el número de frasco que se encontraba debajo del cono metálico para recolectar la muestra.

Tabla 45: Puntos críticos en la tercera iteración.

Nuevamente, se planeó resolver cada uno de los puntos críticos generando bocetos, modelos, diagramas y/o lluvias de ideas.



Etapa de Hacer

Para cada uno de los puntos críticos descritos en la etapa anterior, se aplicó la operación que se explica en las tablas siguientes.

No. Punto crítico	Punto crítico	Operación efectuada
1	Distribución de esfuerzos en el cople para la tapa del sellado de frascos	Se diseñó otro cople que, incluso, evita el uso de tornillos, colocando piezas que limitaran el movimiento en forma de "C" y "T" (Figura 41), fijando el cople con la tapa. En la Figura 42 se observan los bocetos.

Tabla 46: Solución para el punto crítico 1 de la tercera iteración del ciclo.

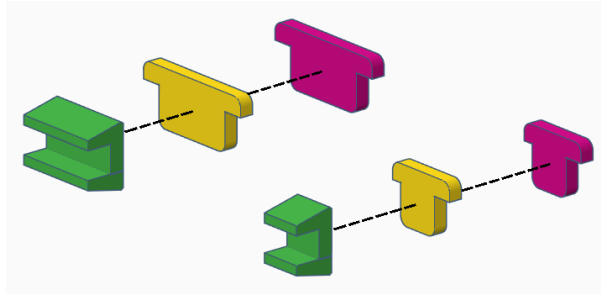


Figura 41: Piezas en forma de "C" y "T" para limitar el movimiento del cople.

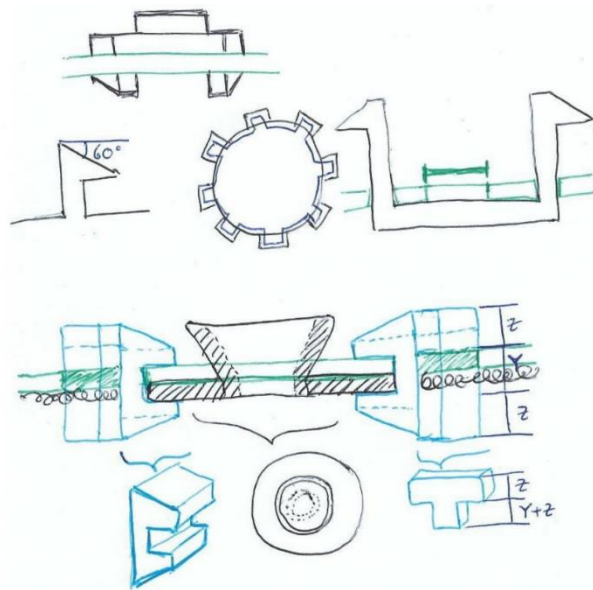


Figura 42: Bocetos del cople para la tapa del sellado de frascos.

No. Punto crítico	Punto crítico	Operación efectuada
2	Rediseño de la sujeción del sistema dentro de la cabina de la trampa	<p>Para el nuevo diseño se consideró que la sujeción no debía sobrepasar el ancho de la guía - porta motor, o bien, sobrepasarlo al lado contrario de donde se encontraría el cono de red metálica. Además, debía buscarse en lo posible que fuera mínimo el espacio entre la guía - porta motor y el perfil de la estructura donde se sostendría. De la misma manera que con la opción de sujeción anterior, se buscó la posibilidad de posicionar el sistema a diferentes alturas y distancias desde el cono; esto se solucionó con ayuda de dos tornillos que pudieran deslizarse sobre una guía en la pieza hasta ser apretados con arandelas y tuercas. Siguiendo el mismo principio, se colocaron cuatro tornillos que pudieran apretarse al perfil cuando se quisiera fijar el intercambiador. Los bocetos que se hicieron se observan en la Figura 43 y la Figura 44, y finalmente, el modelo obtenido en la Figura 45 y la Figura 46.</p> <p>Cabe señalar, que la pieza tapa - sujetador también se modificó, de acuerdo con el diseño de la pieza de sujeción. Su modelo se ilustra en la Figura 47, y el de su ensamble con la pieza de sujeción en la Figura 48.</p>

Tabla 47: Solución para el punto crítico 2 de la tercera iteración del ciclo.

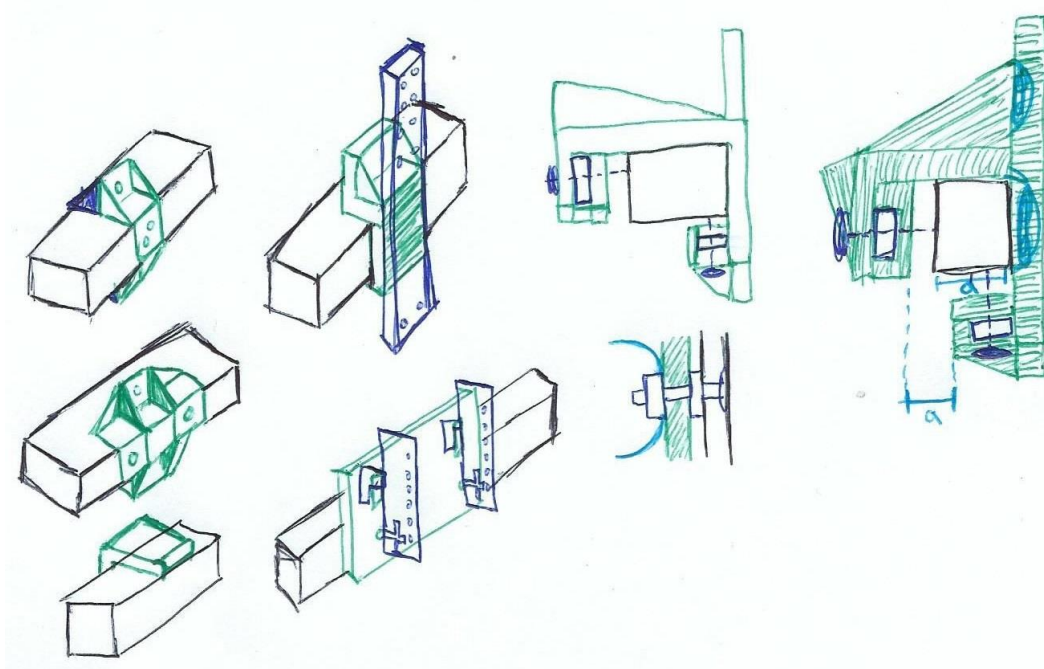


Figura 43: Bocetos de las nuevas opciones de sujeción del sistema.

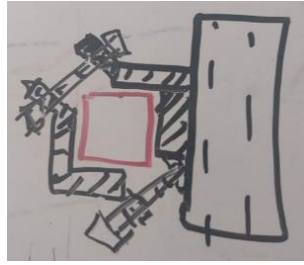


Figura 44: Otro boceto para la nueva opción de sujeción del sistema.

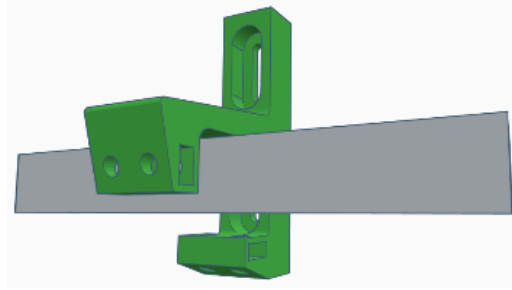


Figura 45: Modelo de la nueva opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Parte frontal.

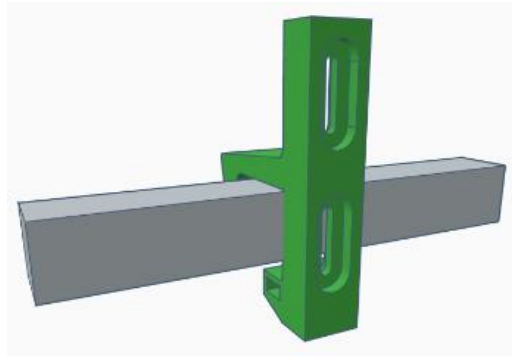


Figura 46: Modelo de la nueva opción de sujeción seleccionada, montada sobre perfil de 1 [in]. Parte posterior.

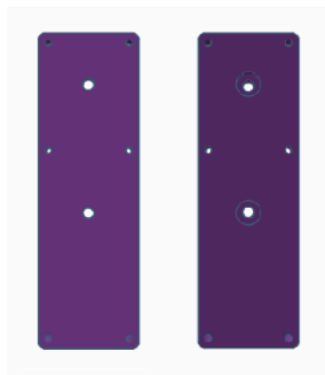


Figura 47: Modelo de la tapa - sujetador. Esta pieza es la que une al sistema con la nueva opción de sujeción seleccionada

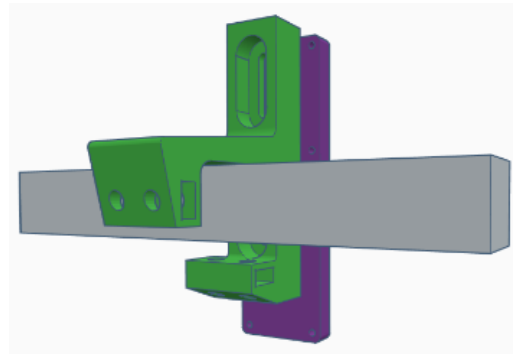


Figura 48: Ensamble entre la opción de sujeción seleccionada y la tapa - sujetador.

No. Punto crítico	Punto crítico	Operación efectuada
3	Modificación de la tapa del sistema de sellado de frascos de acuerdo con el cople	Para el nuevo diseño de la tapa, se colocaron dos orificios, uno para introducir el nuevo cople, y otro para atravesar al sistema sin intervenir con su funcionamiento. El tercer orificio del diseño de la primera iteración del ciclo fue sustituido por una figura que modificaba el diseño circular de la tapa original, permitiendo que un espacio quedara libre para poder introducir el frasco; en tanto que, al insertar otra pieza con esta figura, se completaba la circunferencia de la tapa, cubriendo el frasco que se hubiera introducido con anterioridad. La manera en que se decidió que la figura diseñada se introdujera a la tapa fue deslizándose a través de rieles en forma de "H". En la Figura 49 se muestran los bocetos de la figura y sus rieles, mientras que en la Figura 50 se muestran los modelos la figura deslizada sobre sus rieles en la tapa.

Tabla 48: Solución para el punto crítico 3 de la tercera iteración del ciclo.

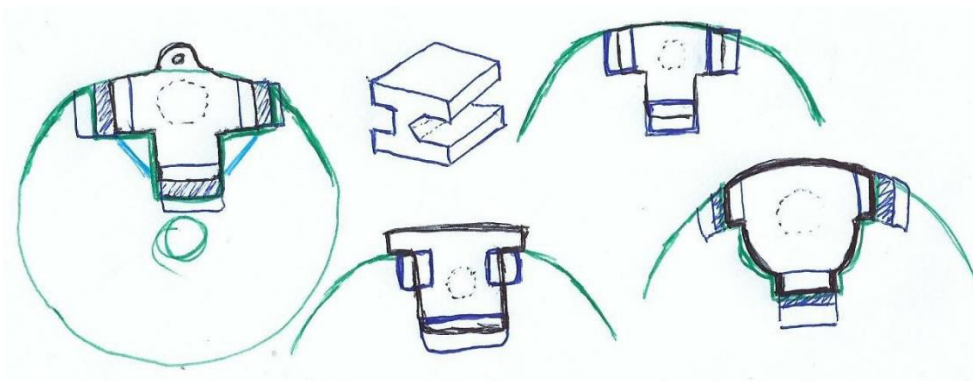


Figura 49: Boceto de la tapa deslizante sobre rieles.

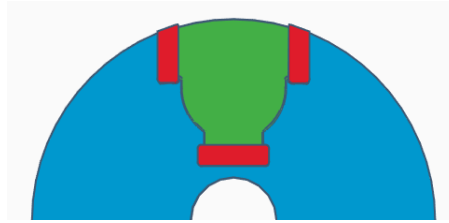


Figura 50: Modelo de la figura deslizada sobre sus rieles en la tapa.

No. Punto crítico	Punto crítico	Operación efectuada
4	Sustituto de la función del servomotor	<p>Como sustituto, se utilizó otro mico - motorreductor Pololu, pero de 30 [RPM], con las características de la Tabla 50. Se conectó directamente al eje del carrusel, en lugar de transmitir el movimiento a través de engranes. Para evitar volver a imprimir el émbolo, se diseñó un cople de dos piezas en forma de caja con tapa, para insertar y sostener este micro - motorreductor, el balero y el tornillo que se une al eje del carrusel.</p> <p>Por otra parte, se utilizaron sensores de efecto Hall de tipo lineal, para indicar que un frasco estaba alineado con el cono de red metálica; sus características se muestran en la Tabla 51. El objetivo de utilizar estos sensores fue obtener una señal para conocer cuando un frasco se colocara por debajo del cono para recibir los insectos. Para poder fijar los sensores al sistema, se pensó en la manera de colocarlos con la posibilidad de ajustar su posición, en tanto que, las piezas que sostuvieran a los sensores se unieran a la tapa del sistema. Así, se siguió el principio utilizado con los otros sensores de efecto Hall, para poder ajustar la posición por medio de rieles.</p> <p>Para comprobar la factibilidad de esta idea, se construyó el prototipo que se muestra en la Figura 51, y a cada aro sujetador del frasco de recolección se le adhirió un imán del mismo tipo que se utilizó para los otros sensores. Se realizaron pruebas para conocer si los sensores detectaban los imanes en movimiento y podían reaccionar para detener los frascos. Con estas pruebas se comprobó la factibilidad de la idea, con la condición de hacer una caracterización con los valores obtenidos de los sensores de efecto Hall. Los bocetos de las piezas para sostener los sensores se encuentran en la Figura 52; mientras que los modelos se muestran en la Figura 53. Además, para ajustarlas a la tapa y limitar su movimiento, se diseñó otra pieza en forma de “T” con orificios, como se observa en la Figura 54. Dado que estas piezas influyeron en el diseño de la tapa y el cople para unir la tapa, los modelos finales de ambas piezas se muestran en la Figura 55 y la Figura 56. Además de la caracterización de los sensores de efecto Hall, se diseñó una guía para el caso en que el frasco no se detuviera en el lugar exacto, permitiendo su alineación con ayuda de esta guía; su modelo se encuentra en la Figura 57.</p>

Tabla 49: Solución para el punto crítico 4 de la tercera iteración del ciclo.



Motor de CD: Micro - motorreductor Pololu 30 [RPM]	
Característica	Descripción
Rango de voltaje de alimentación	6 a 12 [V]
Corriente máxima	29 [mA]
Corriente al arrancar	247 [mA]
Par permitido	5.4 [kg/mm]
Masa	10 [g]
Costo	\$110.00

Tabla 50: Características del micro - motorreductor. Tabla construida con base en las características que se obtuvieron en (Pololu Corporation, 2019) y (CDMX Electrónica, 2019b). Para conocer los valores de corriente máxima y corriente al arrancar se hicieron diferentes pruebas, colocando voltajes de 6 [V], 9 [V] y 12 [V], variando la carga, e intercalando el movimiento de un lado a otro de acuerdo con lo necesario para el proyecto.

Sensor: 49E	
Característica	Descripción
Rango de voltaje de alimentación	2.3 a 10 [V]
Consumo de corriente	4 [mA] para un voltaje de alimentación de 5 [V]
Consumo de corriente máximo	6.5 [mA]
Rango de voltaje en la señal de salida	1.16 a 2.4 [V]
Temperatura de operación	-40 a 100 [°C]
Costo	\$25.00

Tabla 51: Características del sensor de efecto hall 49E. Tabla construida con base en las características que se obtuvieron de su hoja de especificaciones (Yangzhou Positioning Tech). El rango de voltaje en la señal de salida se obtuvo experimentalmente.



Figura 51: Prototipo de las piezas para sostener los sensores de efecto Hall de tipo lineal, se sostiene de la tapa a diferentes distancias, los sensores se colocan en una barra donde se deslizan vertical y horizontalmente para finalmente ajustarse.

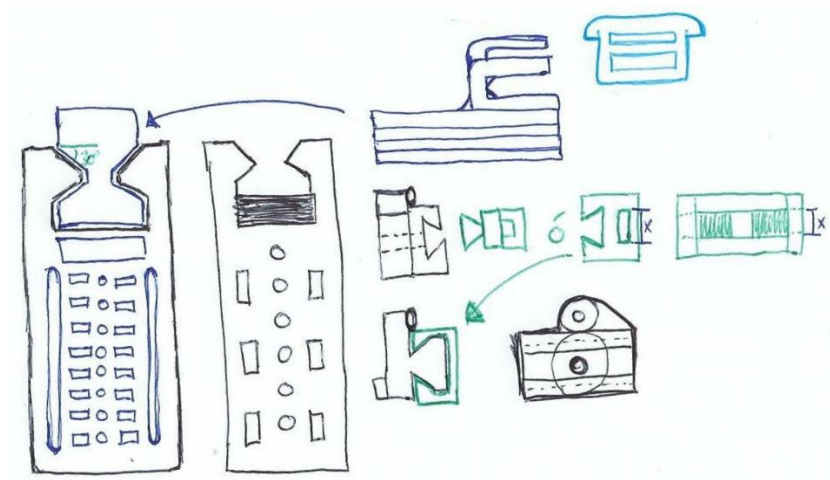


Figura 52: Bocetos de las piezas para sostener los sensores de efecto Hall de tipo lineal.

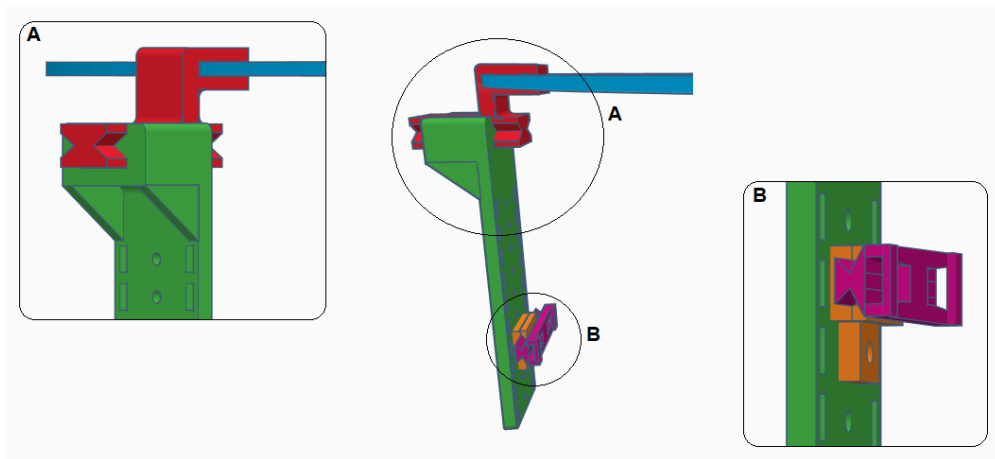


Figura 53: Modelo de las piezas para sostener los sensores de efecto Hall de tipo lineal.

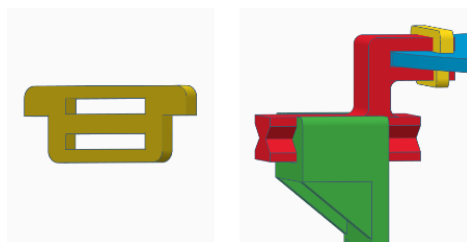


Figura 54: Pieza para limitar el movimiento de las piezas para sostener los sensores de efecto Hall de tipo lineal. En la izquierda, se muestra sólo la pieza. En la derecha se muestra el modelo de la pieza montada.

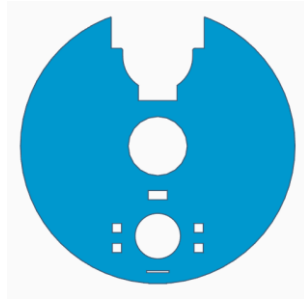


Figura 55: Modelo de la tapa final.

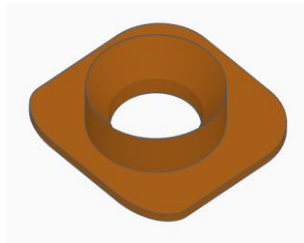


Figura 56: Modelo final del cople para unir la tapa.



Figura 57: Guía para alinear el frasco.

No. Punto crítico	Punto crítico	Operación efectuada
5	Voltaje y corriente	<p>Para este punto se realizó otra tabla considerando los valores de corriente y voltaje de todos los componentes del sistema incluidos en esta iteración, cabe indicar, que al ya no ocupar el servomotor, el regulador se descartó (Tabla 53). Se comprobó que las conexiones de estos componentes no dañan la SBC, considerando la alimentación de los 5 [V] de los pines dedicados específicamente a otorgarlos. En cuanto a la corriente, se consume un total de 84.43 [mA], lo que está por debajo de la corriente máxima en los GPIO, sin embargo, se debe tener precaución con el puente H para habilitar sólo un pin EN al mismo tiempo, ya que su corriente consumida, con ambos pines EN habilitados, está cerca de los 50 [mA] máximos que permite la Raspberry Pi. Por otra parte, para los componentes que en la tabla se indica la alimentación por una fuente externa, nuevamente se dividieron en 3 fuentes diferentes: una fuente de 5 [V] y 2.5 [A] para la SBC; una fuente de 127 [VCA] y 30 [A] para alimentar el extractor; y una fuente de 9 [V] y 1 [A] para alimentar los motores.</p>

Tabla 52: Solución para el punto crítico 5 de la tercera iteración del ciclo.



Componente	Voltaje de alimentación [V]	Corriente de alimentación [mA]	Fuente de alimentación
SBC	5	2500	Externa
Extractor	127 [VCA]	14200	Externa
Relevador	3.3	7.5	SBC
Motorreductor de metal 100 [RPM]	9	171	Externa
Puente H	5	47.69	SBC
Motorreductor de metal 30 [RPM]	9	247	Externa
BMP180	3.3	1	SBC
DHT11	5	2.5	SBC
LM35	5	0.06	SBC
Circuito para ECS1030	3.3	0.33	SBC
U18	2(3.3)	2(6)	SBC
49E	2(3.3)	2(4)	SBC
MCP3008	3.3	0.55	SBC
LED Amarillo	1.95	3	SBC
LED Azul	3.04	1.8	SBC

Tabla 53: Voltaje y corriente de alimentación de los componentes para la tercera iteración del ciclo. En la última columna se indica como sería alimentado el componente, por la tarjeta o por una fuente externa. Tabla elaborada con base en las características descritas con anterioridad.

No. Punto crítico	Punto crítico	Operación efectuada
6	Temperatura y masa	Como se observó en Tabla 50 y Tabla 51, ambos componentes son aptos para funcionar dentro del rango de temperatura que se puede tener en Mocochá. Por otra parte, considerando los frascos con contenido y los nuevos componentes y piezas, se midió la masa que movería cada micro - motorreductor, obteniendo 1.321 [kg] para el que sustituyó al servomotor, y 1.444 [kg] para el micro - motorreductor que eleva y baja los frascos. Para ambos micro - motorreductores, la masa a mover se encuentra dentro de sus límites. Además, comparándola con la que se tenía en la primera iteración del ciclo, fue menor.

Tabla 54: Solución para el punto crítico 6 de la tercera iteración del ciclo.



No. Punto crítico	Punto crítico	Operación efectuada
7	Número de frasco	Para determinar el número de frasco que se encuentra debajo del cono metálico para recolectar la muestra, se optó por llevar un registro del movimiento de los frascos, es decir, cada que los sensores de efecto Hall detectan un frasco debajo del cono, se incrementa o decrementa un contador que va de 1 a 6, según sea el giro en sentido de las manecillas del reloj o en contrasentido.

Tabla 55: Solución para el punto crítico 7 de la tercera iteración del ciclo.

El cople para la tapa del sellado de frascos y sus piezas limitadoras de movimiento en forma de “C”, los rieles de la tapa, la pieza de sujeción del sistema, la tapa - sujetador, el cople de dos piezas en forma de caja con tapa para el micro - motorreductor, y las piezas para sostener los sensores de efecto Hall fueron manufacturadas con la impresora 3D.

Al imprimir la primera pieza del cople en forma de caja del micro - motorreductor (Figura 58) se encontró el error en una magnitud que provocaba que el eje del micro - motorreductor no estuviera centrado, por lo que se modificó la segunda pieza de este cople (Figura 59) y se creó otra pieza para compensar este error (Figura 60).

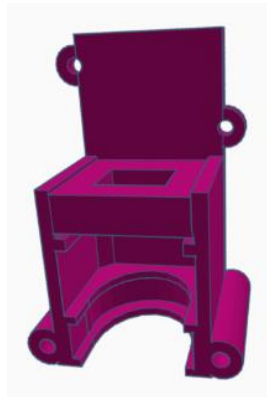


Figura 58: Primera pieza del cople en forma de caja del micro - motorreductor.

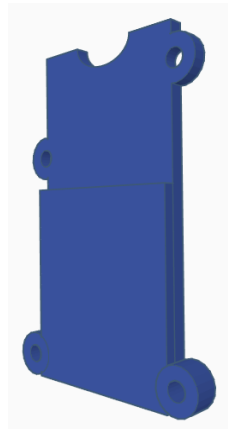


Figura 59: Segunda pieza del cople en forma de caja del micro - motorreductor (Tapa).

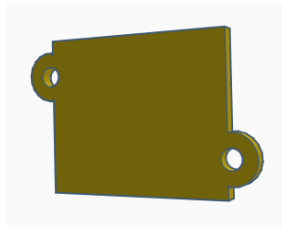


Figura 60: Pieza creada para compensar el error en el diseño del cople en forma de caja con tapa del micro - motorreductor.

Para la tapa del sellado de frascos, las piezas limitadoras de movimiento en forma de “T”, y la pieza con la figura que complementa a la tapa, se utilizó corte láser sobre acrílico de 3 [mm].

Todas las fotografías de las piezas manufacturadas en esta iteración se muestran en el ANEXO C.

En necesario indicar que, en cuanto a la programación en esta iteración, se descartó el código del servomotor y se creó un archivo para escribir el código referente al movimiento del nuevo micro - motorreductor; además, se definieron valores de caracterización para los sensores de efecto Hall de tipo lineal.

Etapa de Verificar

Las pruebas se llevaron a cabo en el Laboratorio de Electrónica y Automatización para Industria 4.0, en el IIMAS. Dada la sujeción del sistema diseñada en esta iteración, el intercambiador se fijó a un perfil de 1 [in], como el que se encuentra en la estructura de la trampa, en tanto que el perfil se fijó a la mesa del laboratorio. Se hicieron las conexiones necesarias para energizarlo y se modificó la caracterización de los sensores de efecto Hall para alinear cada frasco de recolección bajo el cono de red metálica, no obstante, se comprobó la utilidad del riel para alinear cada frasco ya que en ocasiones se presentaba un pequeño desfase. Además, se logró llevar el conteo de los frascos cuando éstos eran percibidos por los sensores. En cuanto a las piezas de sujeción del sistema, no presentaron ningún daño por la carga o movimiento del sistema.



Etapa de Actuar

De acuerdo con las pruebas realizadas en el laboratorio del IIMAS en esta iteración, el prototipo está listo para instalarse y probarse en la trampa de Mocochoá, lo que forma parte del trabajo a futuro.

Por lo anterior, se armó la caja de componentes electrónicos final, que se encontraría dentro de la trampa. En lugar de hacer un circuito impreso, se utilizó una placa perforada y alambre para hacer las conexiones. Asimismo, se utilizaron conectores que facilitarían el montaje y desmontaje de los componentes en el caso de ser necesario. Se profundiza sobre las conexiones en el capítulo 5: Integración final, dentro del apartado Conexiones del sistema.

Por otra parte, dada la comprobación del funcionamiento de los elementos anteriores, quedaron concretados los elementos de la primera y segunda capa de IoT, por lo que se prosiguió con su unificación, agregando los elementos de la tercera capa (incluyendo el *framework*, para las aplicaciones, que se ocupó desde la primera iteración del ciclo), para finalmente, probar todo el sistema. Esta unificación y pruebas se muestran en los apartados restantes del siguiente capítulo. Las fotografías del intercambiador conectado se observan en la Figura 61.

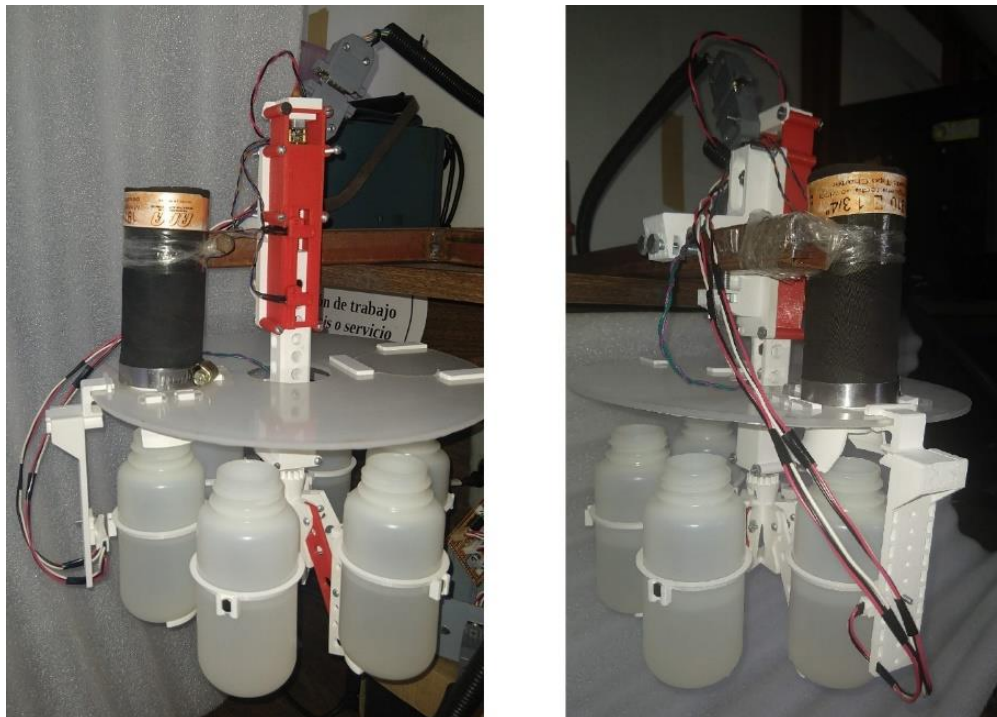


Figura 61: Intercambiador fijado a perfil de 1 [in], a la vez, el perfil fijado a la mesa del laboratorio en el IIMAS. Se muestra el sistema, de izquierda a derecha: parte frontal y parte posterior.

Capítulo 5 - Integración final


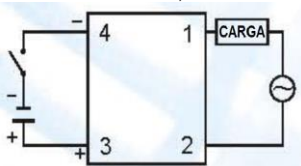

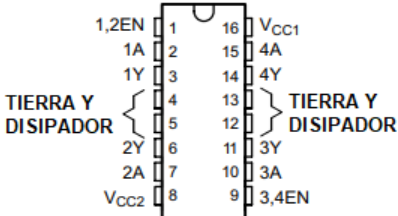
En este capítulo se describen:


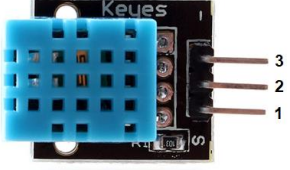



- Las conexiones entre todos los componentes del subsistema Electrónico.
- El sistema operativo de la SBC, las bibliotecas necesarias para programar los actuadores y los sensores, así como los programas finales utilizados.
- La manera en que se utilizó el protocolo de red, incluyendo los elementos de seguridad tomados en cuenta, así como la estructura de los mensajes.
- El *framework* que se mencionó en la primera iteración, utilizado para construir la aplicación web - aplicación móvil, donde el usuario puede tener acceso a los datos almacenados y controlar el intercambiador.
- Las pruebas finales del sistema integrado.



Conexiones del sistema

Para poder hacer las conexiones del sistema es necesario conocer la forma en que se hace con cada componente; es importante identificar exactamente el modelo del componente que se adquiere para no errar en las conexiones, por ejemplo, el sensor DHT11 consta de 4 pines y se puede encontrar integrado en diferentes módulos cuyas conexiones varían. En la Tabla 56 se describen las conexiones de algunos de los componentes del sistema.

Componente	Imagen y/o diagrama	Forma de conexión
<p>Relevador SSR - 40 DA</p>	 <p>Imagen obtenida de (indiamart, 2019)</p>  <p>Imagen obtenida de (FOTEK, s.f.)</p>	<p>1: Un extremo de la línea interrumpida. 2: El otro extremo de la línea interrumpida. 3: Señal de control. 4: Tierra.</p>
<p>Puente H L293D</p>	 <p>Imagen obtenida de (NoMADA Store, 2019)</p>  <p>Imagen obtenida de (Texas Instruments, 2016)</p>	<p>1,2EN: <i>Enable</i> del canal 1 y 2. Para habilitar e inhabilitar el motor. Además, se puede controlar la velocidad del motor. 1A: Entrada al controlador para el primer motor. 1Y: Salida del controlador a un polo del primer motor. TIERRA Y DISIPADOR: Disipador de calor y tierra. 2Y: Salida del controlador al otro polo del primer motor. 2A: Entrada al controlador para el primer motor. VCC2: Fuente de alimentación del motor. 3,4EN: <i>Enable</i> del canal 3 y 4. Con las mismas funciones que el <i>enable</i> del canal 1 y 2, pero para el segundo motor. 3A: Entrada al controlador para el segundo motor.</p>

		<p>3Y: Salida del controlador a un polo del segundo motor.</p> <p>4Y: Salida del controlador al otro polo del segundo motor.</p> <p>4A: Entrada al controlador para el segundo motor.</p> <p>VCC1: Fuente de alimentación del circuito integrado.</p>
<p>Sensor BMP180 Módulo GY-68</p>	 <p>Imágenes obtenidas de (Makerlab-Electronics, 2015)</p>	<p>VIN: Fuente de alimentación.</p> <p>GND: Tierra.</p> <p>SCL: Reloj del bus I²C.</p> <p>SDA: Datos del bus I²C.</p>
<p>Sensor DHT11</p>	 <p>Imagen obtenida de (Lovely Electronics, 2018)</p>	<p>1: Señal.</p> <p>2: Fuente de alimentación.</p> <p>3: Tierra.</p>
<p>Sensor LM35</p>	 <p>Imagen obtenida de (AFEL, 2019).</p>	<p>1: Fuente de alimentación.</p> <p>2: Señal.</p> <p>3: Tierra.</p>
<p>Sensor ECS1030</p>	 <p>Imagen obtenida de (330ohms, 2019b)</p>	<p>1: Malla.</p> <p>2: Anillo.</p> <p>3: Punta.</p>
<p>Sensores de efecto Hall (U18 y 49E)</p>	 <p>Imagen obtenida de (Yangzhou Positioning Tech).</p>	<p>1: Fuente de alimentación.</p> <p>2: Tierra.</p> <p>3: Señal.</p>


<p>ADC MCP3008</p>	 <p>Imagen obtenida de (Core Electronics, 2019).</p> <table border="1" data-bbox="570 447 813 636"> <tr><td>CH0</td><td>1</td><td>16</td><td>V_{DD}</td></tr> <tr><td>CH1</td><td>2</td><td>15</td><td>V_{REF}</td></tr> <tr><td>CH2</td><td>3</td><td>14</td><td>AGND</td></tr> <tr><td>CH3</td><td>4</td><td>13</td><td>CLK</td></tr> <tr><td>CH4</td><td>5</td><td>12</td><td>D_{OUT}</td></tr> <tr><td>CH5</td><td>6</td><td>11</td><td>D_{IN}</td></tr> <tr><td>CH6</td><td>7</td><td>10</td><td>CS/SHDN</td></tr> <tr><td>CH7</td><td>8</td><td>9</td><td>DGND</td></tr> </table> <p>Imagen obtenida de (Microchip Technology, 2008).</p>	CH0	1	16	V _{DD}	CH1	2	15	V _{REF}	CH2	3	14	AGND	CH3	4	13	CLK	CH4	5	12	D _{OUT}	CH5	6	11	D _{IN}	CH6	7	10	CS/SHDN	CH7	8	9	DGND	<p>CH0 - CH7: Entradas Analógicas. Son ocho canales a los que se les conectan las señales analógicas.</p> <p>DGND: Tierra Digital.</p> <p>CS/SHDN: Selección de Chip / Apagado. Se conecta al pin SPI0_CE0.</p> <p>DIN: Entrada de Datos. Se conecta al pin SPI0_MOSI.</p> <p>DOUT: Salida de Datos. Se conecta al pin SPI0_MISO.</p> <p>CLK: Reloj. Se conecta al pin SPI0_SCLK.</p> <p>AGND: Tierra Analógica.</p> <p>VREF: Voltaje de Referencia.</p> <p>VDD: Fuente de alimentación.</p>
CH0	1	16	V _{DD}																															
CH1	2	15	V _{REF}																															
CH2	3	14	AGND																															
CH3	4	13	CLK																															
CH4	5	12	D _{OUT}																															
CH5	6	11	D _{IN}																															
CH6	7	10	CS/SHDN																															
CH7	8	9	DGND																															

Tabla 56: Forma de conexión de los componentes. Tabla construida con base en la información en las hojas de especificaciones de cada componente.

Antes de realizar el diagrama de conexiones, se estableció el tipo de conectores que se utilizarían para facilitar el montaje y desmontaje entre los componentes y la placa perforada. La construcción de *header* polarizados fue la mejor opción debido a que este conector no se desconecta con facilidad, además, por la polaridad no se puede errar en el sentido en que se conecta. Se realizó el diagrama a mano que se transcribió como se muestra en la Figura 62.

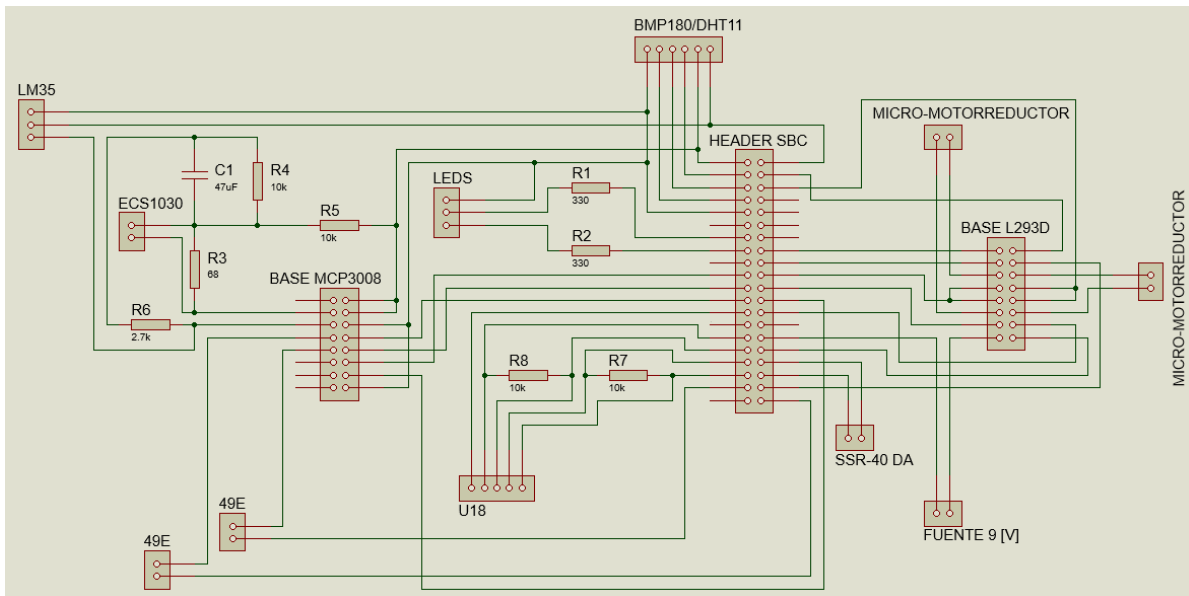


Figura 62: Diagrama de las conexiones en la placa perforada.

La placa perforada se alambró con base en el diagrama anterior (Figura 63), soldando los *header* polarizados macho de diferentes tamaños, los resistores y capacitores, el *header* de

pinos macho de 20x2 para conectar la SBC, y las bases para los circuitos integrados (L293D y MCP3008); obteniendo la placa de la Figura 64.

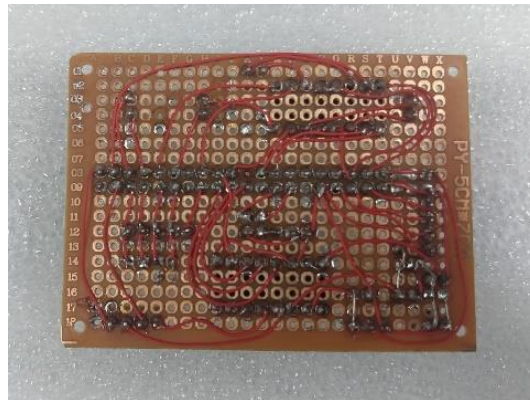


Figura 63: Placa perforada alamburada.

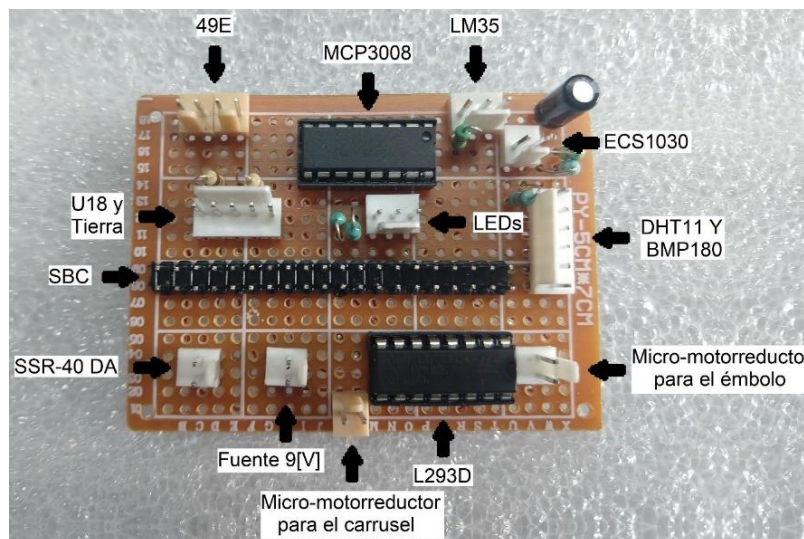
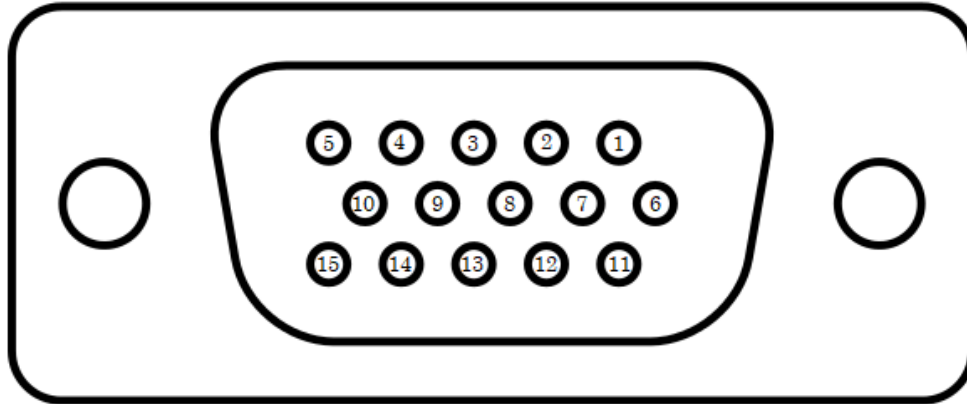


Figura 64: Placa soldada. Se indica el componente que se conecta a cada *header* polarizado (incluyendo la fuente externa de 9 [V]), las bases con los circuitos integrados y el *header* de pines para la Raspberry Pi.

En contraparte, se debían soldar los *header* polarizados hembra a cada componente. No obstante, no todos los componentes se encontrarían cerca de la caja por lo que las conexiones se llevaron a cabo de la siguiente manera:

- Para los componentes que pertenecen al intercambiador automático de muestreo se les soldó cable o alambre, conectándolos a un conector DE-15 hembra como en la Figura 65. Por otra parte, se utilizó cable UTP categoría 5E, forrado con manguera ranurada, y dos conectores DE-15 macho para construir el cable de la Figura 66. En la caja se unió otro conector DE-15 hembra (Figura 67), al cual se soldaron los *header* polarizados correspondientes.





1. Sin usar.
2. Fuente de alimentación (rojo) del 49E dos.
3. Señal (blanco) del 49E dos.
4. Fuente de alimentación (rojo) del 49E uno.
5. Señal (blanco) del 49E dos.
6. Fuente de alimentación (anaranjado) del U18 arriba.
7. Señal (azul) del U18 arriba.
8. Fuente de alimentación (anaranjado) del U18 abajo.
9. Señal (azul) del U18 abajo.
10. Tierra (negro).
11. Sin usar.
12. Polo 4Y (morado) del micro - motorreductor para carrusel.
13. Polo 3Y (verde) del micro - motorreductor para carrusel.
14. Polo 1Y (negro) del micro - motorreductor para émbolo.
15. Polo 2Y (rojo) del micro - motorreductor para émbolo.

Figura 65: Conexiones en el conector DE-15. Imagen elaborada a partir del esquema de (Mobius, 2006). Los colores que se indican corresponden al color del cable o alambre de cada componente en el intercambiador.



Figura 66: Cable UTP forrado con manguera ranurada, asegurado con cinturoncillos de nylon. En sus extremos se une a conectores DE-15 macho.



Figura 67: Conector DE-15 hembra.

- En cuanto al LM35, el ECS1030 y el relevador, se optó por utilizar conectores Jack de 3.5 [mm] (hembra y macho). En la Figura 68 se muestran los conectores macho, mientras que en la Figura 69 se observan los conectores hembra unidos a la caja, a los cuales también se les soldaron los *header* polarizados correspondientes.

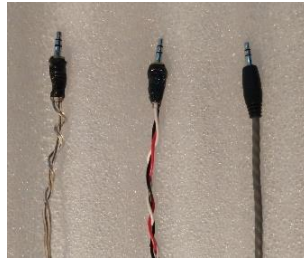


Figura 68: Conectores Jack de 3.5 [mm] macho. De izquierda a derecha: Relevador, LM35, ECS1030.

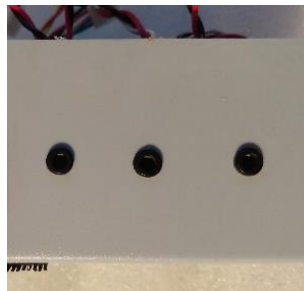


Figura 69: Conectores Jack de 3.5 [mm] hembra. De izquierda a derecha: ECS1030, LM35, Relevador.

- Los *header* polarizados correspondientes a los sensores DHT11 y BMP180 se unieron a un conector hembra DE-9 (Figura 70), donde se conectaron directamente estos sensores.



Figura 70: Conector DE-9 hembra para sensores. De izquierda a derecha: BMP180, DHT11. Se inhabilitan físicamente dos pines del conector para evitar equivocación al conectar.

- En cuanto a los LED, se unieron a la caja (Figura 71) y se le soldaron los *header* polarizados hembra.

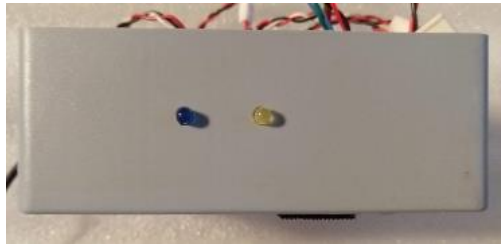


Figura 71: LED unidos a la caja.

- El *header* polarizado hembra para la fuente externa de 9 [V] se soldó a un conector DC hembra de 5.5 [mm], el cuál se fijó a la caja (Figura 72).



Figura 72: Conector DC hembra de 5 [mm].

- Para conectar la SBC al *header* de pines de la placa perforada se utilizó un cable plano de extensión de 40 pines. En cuanto a su fuente de alimentación, se atravesó un cable por la caja (Figura 73).



Figura 73: Cable que atraviesa una pared de la caja para alimentar a la SBC.

Finalmente, la caja de componentes electrónicos se muestra en Figura 74.

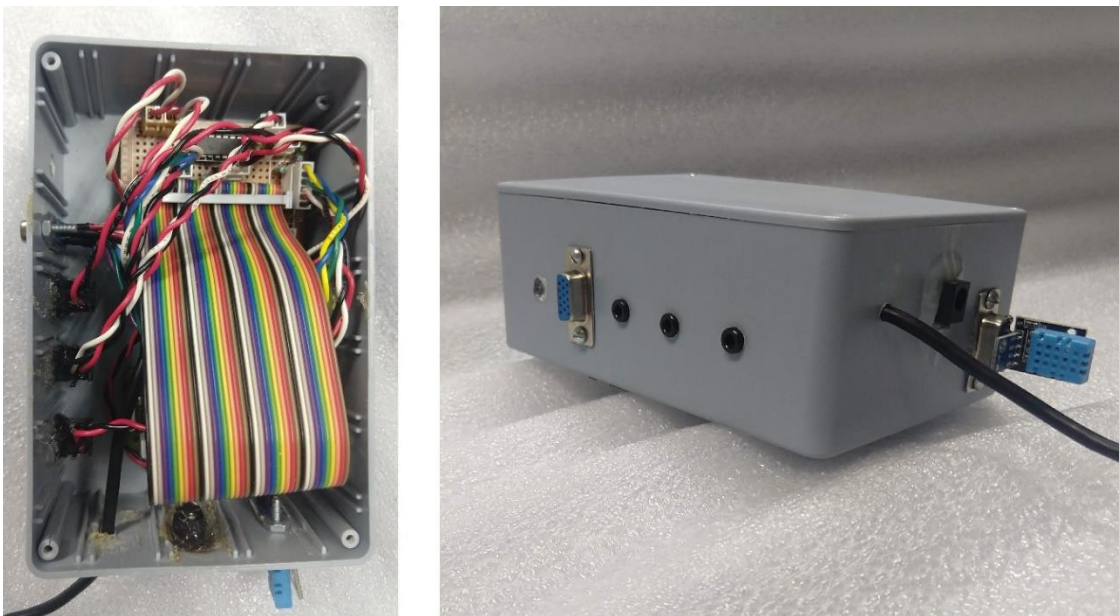


Figura 74: Caja de componentes electrónicos final. A la izquierda, el interior de la caja. A la derecha, vista desde el exterior.

Bibliotecas para los actuadores y sensores

Al utilizar la Raspberry Pi como tarjeta de desarrollo, se instaló el sistema operativo oficial: *Raspbian*. La versión disponible fue *Raspbian Stretch*, versión de noviembre del 2018 (Raspberry Pi Foundation, 2018b), la cual ofrecía tres opciones:

- *Raspbian Stretch with desktop and recommended software*
- *Raspbian Stretch with desktop*
- *Raspbian Stretch Lite*

Se seleccionó la última opción ya que contiene sólo lo esencial para correr el sistema operativo, ahorrando espacio y permitiendo la instalación sólo del software necesario. Dado que se requiere una tarjeta SD donde almacenar este sistema operativo, se utilizó una tarjeta de marca ADATA con capacidad de 16 GB, clase 10.

Existen diferentes maneras para trabajar con la Raspberry Pi, una es a través de la conexión de periféricos (monitor, teclado, *mouse*), otra es con una conexión vía SSH¹³, mediante una configuración donde la tarjeta se convierte en el servidor, mientras que otra computadora se convierte en el cliente. En este caso se optó por utilizar la conexión vía SSH.

Por otra parte, se accedió a la configuración de la Raspberry para habilitar las interfaces I²C y SPI, debido a que el BMP180 utiliza el bus I²C, y el MCP3008 el SPI.

Con respecto a las bibliotecas que se utilizaron, se creó una carpeta para colocarlas, los siguientes comandos se ejecutan estando dentro de ella.

Como lenguaje de programación se optó por utilizar Python (Python Software Foundation, 2001), por su sencillez, relevancia en los últimos años, y la disponibilidad de una biblioteca llamada GPIO Zero (Nuttall & Jones, 2015), la cual facilita la configuración y uso de los diferentes componentes conectados a la tarjeta. Los comandos para instalar esta biblioteca y las dependencias necesarias para Python fueron:

```
sudo apt-get install -y python3 python3-pip Python-dev
sudo apt-get install git
sudo apt install python3-gpiozero
```

Existe una biblioteca creada por Adafruit (DiCola, 2018b) para obtener los valores del BMP180 de una manera sencilla. Para obtenerla se ejecutaron los siguientes comandos:

¹³ SSH: *Secure SHell*. Es un “protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente.” (MIT, s.f.).



```
wget
https://files.pythonhosted.org/packages/63/36/15914e164f7d4ce82e89be3ca72b62
4aa1062302462d66feb7136657196e/Adafruit_BMP-1.5.4.tar.gz

sudo tar -xzf Adafruit_BMP-1.5.4.tar.gz
cd Adafruit_BMP-1.5.4/
sudo python3 setup.py install
```

Otra biblioteca creada por Adafruit (DiCola, 2018a) es la que facilita la obtención de los valores del DHT11. Se obtuvo ejecutando los siguientes comandos:

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT/
sudo python3 setup.py install
```

Cabe recordar que, las señales que se obtienen del LM35 y del ECS1030 son analógicas por lo que se ocupó el ADC. Dado que el MCP3008 utiliza el bus SPI, también se descargó la biblioteca Spidev (doceme, 2015) con los siguientes comandos:

```
git clone https://github.com/doceme/py-spidev.git
cd py-spidev
sudo python3 setup.py install
```

Además, se utilizó la clase dedicada al MCP3008 de la biblioteca GPIO Zero, con la que se indica el canal de interés y la manera en que se desea obtener el valor en cuestión. Otras clases que se utilizaron de esta biblioteca fueron:

- Motor: Para el movimiento en ambos sentidos de los micro - motorreductores.
- PWMLED: Para controlar la velocidad de los micro - motorreductores.
- LED: Para encender y apagar los LED.
- OutputDevice: Para asignar los pines de la SBC como salidas.
- InputDevice: Para asignar los pines de la SBC como entradas.

En el ANEXO D se muestra el código que se estableció para los sensores.

Para los micro - motorreductores se escribieron dos códigos, uno para el micro - motorreductor que mueve al émbolo (ANEXO E) y otro para el micro - motorreductor que mueve al carrusel (ANEXO F).



MQTT

Recordando las características de este protocolo, se tiene un *broker* y más de dos clientes (de tipo publicador y suscriptor).

En el Laboratorio de Electrónica y Automatización para Industria 4.0, se cuenta con un servidor con una distribución de Linux como sistema operativo, Ubuntu 18.04.2 LTS, en él se descargó el *broker* de código abierto Eclipse Mosquitto, creado por IBM en 2001. En la página oficial (Eclipse Foundation, 2004) se indica que ya existe en el repositorio de Ubuntu por lo que basta con escribir `apt-get install`, sin embargo, la versión que se instala no es reciente por lo que se ejecutaron primero los comandos:

```
sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
sudo apt-get update
```

Posteriormente, se instaló el *broker*:

```
sudo apt-get install mosquitto
```

Debido a que en el servidor existirían clientes, además del *broker*, se instaló Mosquitto - clients con el siguiente comando:

```
sudo apt-get install mosquitto-clients
```

Por otra parte, ya que la SBC funcionaría como cliente, se descargó de la misma manera Mosquitto - clients. Además, para poder utilizar MQTT con Python y publicar mensajes, se descargó la biblioteca llamada Paho (Eclipse Foundation, 2019) utilizando el siguiente comando:

```
sudo pip install paho-mqtt
```

Un punto clave de ocupar el protocolo MQTT, es respecto a la jerarquía de los *topics*, para lo cual se debe establecer la estructura por niveles de los *topics* a los que se suscribirían y publicarían los clientes; se definió la mostrada en la Figura 75.



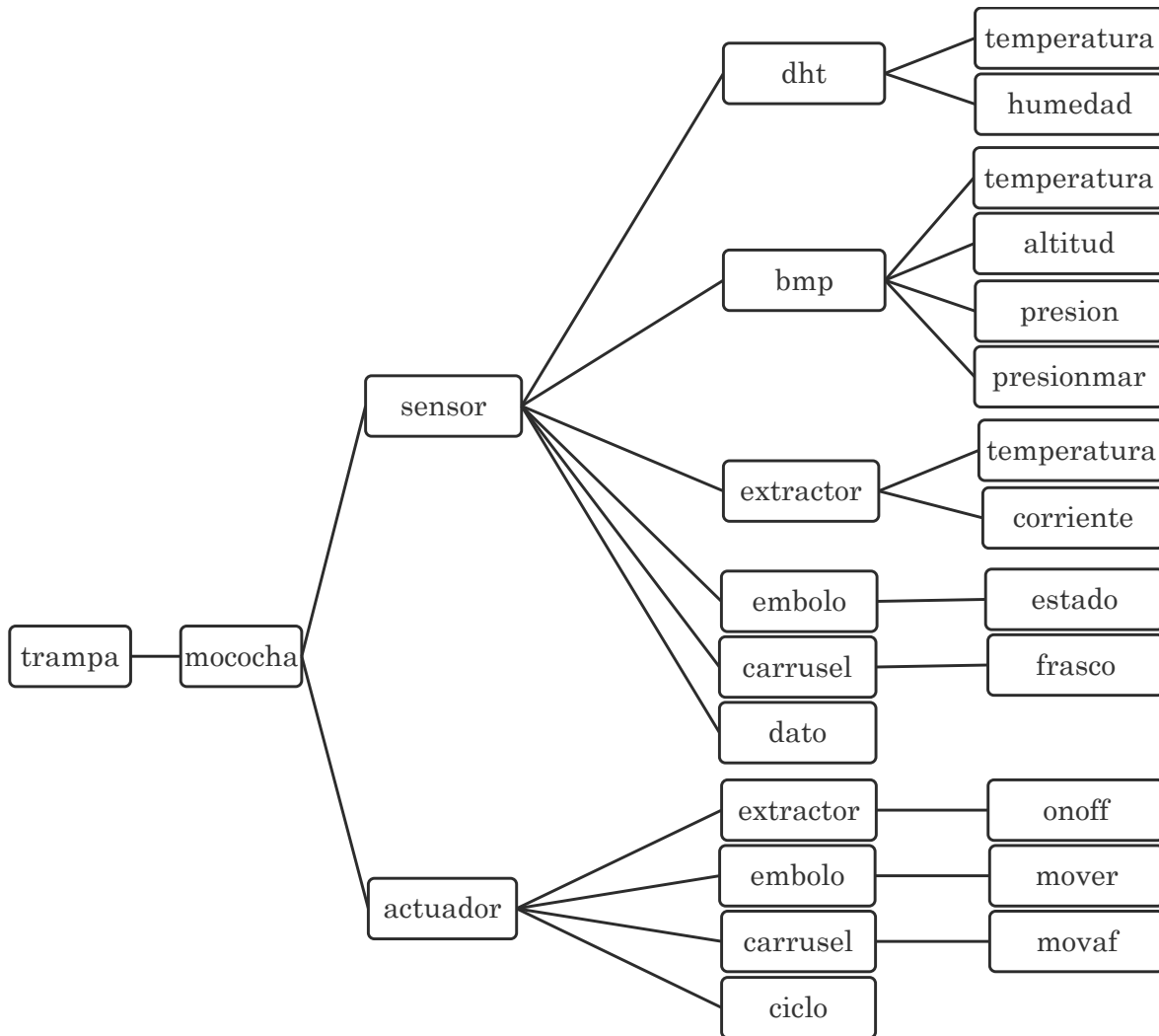


Figura 75: Estructura de los *topics* para suscribirse y publicar. Se consideran cinco niveles donde el *topic* “trampa” es el primer nivel.

Cabe recordar que, para el envío y recepción de los mensajes, se colocó un usuario y contraseña de seguridad por lo que para que un cliente se suscribiera y publicara a un *topic*, debía autenticarse. De esta manera se escribió el código en la SBC para que como cliente se autenticara, estableciera la conexión con el *broker*, se suscribiera a los *topics* de todos los niveles, y publicara a diferentes *topics* de acuerdo con el mensaje que recibiera. Además, se estableció un QoS¹⁴ nivel 1 para asegurar que el *broker* recibiera y enviara los mensajes, sin importar que estos se duplicaran.

Para este código, se utilizaron las bibliotecas Paho y GPIO Zero, así como el código escrito para los sensores y actuadores; se muestra en el ANEXO G.

¹⁴ QoS: *Quality of Service*. “En MQTT es el nivel de garantía de entrega de un mensaje específico. El cliente establece un nivel de QoS, cada vez que se suscribe a un *topic* o publica un mensaje a un *topic* en un *broker*” (Tolia, 2018). Existen tres niveles de QoS de MQTT: 0, 1 y 2.

Framework

Se decidió utilizar el *framework* Pimatic (Schneider, 2019) para crear la aplicación web. Este *framework* está enfocado en desarrollar sistemas domóticos con la ayuda de la Raspberry Pi. Además, con ayuda de diferentes *plugins*¹⁵ incluye características para controlar y automatizar tareas, visualizar y procesar información, entre otras. En tanto que su esquema le permite incluir todos los dispositivos del sistema en la misma plataforma. Cabe destacar que, es gratuito, de código abierto y corre en node.js. No obstante, su principal desventaja es que continua en desarrollo por lo que se encuentra en estatus beta.

Se puede configurar escribiendo sobre su archivo config.json, para lo cual es necesario identificar sus secciones:

- *Settings*: esta sección está definida para las configuraciones del servidor HTTP y HTTPS, además, de las configuraciones de la base de datos, entre otras.
- *Plugins*: en esta se listan los plugin a cargar.
- *Devices*: esta contiene todos los dispositivos que se registran en el *framework*, pueden ser de diferentes clases, como botones, sensores, entradas, entre otros. A cada uno se le asigna un identificador.
- *Rules*: en ella se describen las reglas que se ejecutarán dadas ciertas condiciones.
- *Variables*: en esta sección se definen las variables del *framework*.
- *Pages*: se puede incluir una o más páginas, se deben incluir los identificadores de los dispositivos pertenecientes a cada página.
- *Groups*: los dispositivos, reglas y variables se pueden agrupar. En esta sección se definen los grupos a través de sus identificadores.
- *Roles*: se definen los permisos para cada rol que se asigna a cada usuario.
- *Users*: se asignan los usuarios, incluyendo su rol y contraseña.

Además, según el usuario y su rol, existe la opción de configurar las secciones anteriores directo en la aplicación, lo que hace que el *framework* sea amigable con el programador.

En este caso, se instalaron tres plugin: *cron*, *mobile - frontend* y *mqtt*. Así, la configuración del envío y recepción de mensajes fue más sencilla, asimismo, se configuró la hora para automatizar la ejecución de diferentes tareas, y cumplió con la característica de ser responsiva, por lo que se pudo visualizar adecuadamente en un dispositivo móvil.

Por otra parte, este *framework* cuenta con la posibilidad de configurar una base de datos con los mensajes obtenidos, lo que también forma parte de la tercera fase de IoT. Sin embargo, no se modificó la configuración establecida por Pimatic, aunque se comprobó que los mensajes se almacenaron.

¹⁵ Plugin: es un “fragmento o componente de código hecho para ampliar las funciones de un programa o de una herramienta” (NeoAttack, 2018).



Incluyendo lo anterior, la arquitectura IoT para el sistema, se terminó de definir como se resume en la Figura 76.

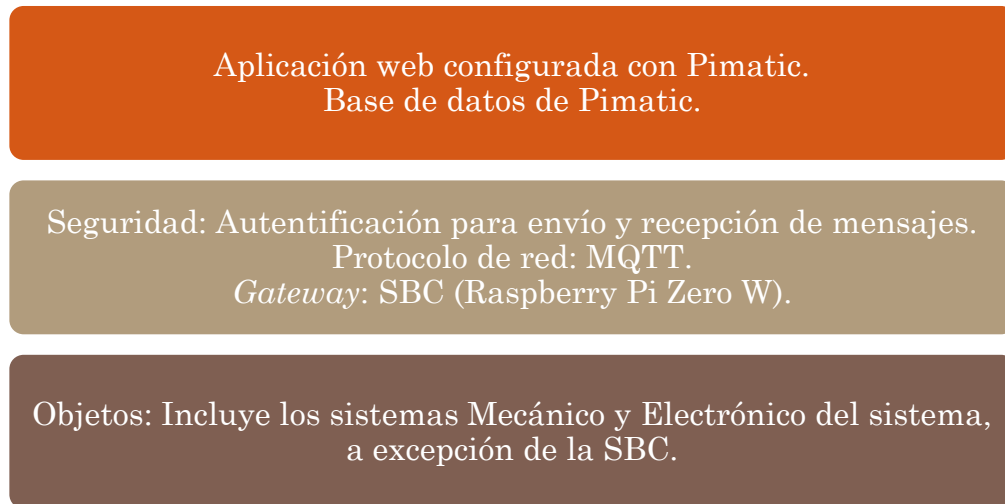


Figura 76: Capas de IoT para este proyecto.

En el ANEXO H se muestra una parte del archivo config.json, con las configuraciones de las secciones *Devices* y *Rules* que se utilizaron para realizar las pruebas finales.

La aplicación que se configuró se muestra en la Figura 77.



Figura 77: Aplicación web configurada utilizando Pimatic.



Pruebas finales de operación

Las últimas pruebas que se realizaron fueron en el Laboratorio de Electrónica y Automatización para Industria 4.0, en el IIMAS, se utilizó la aplicación web configurada para:

- Obtener los datos de los sensores, a excepción del de corriente y temperatura del extractor.
- Completar 100 iteraciones de ciclos completos. Cada ciclo consistía en bajar el frasco 1, girar el carrusel al frasco 2, subirlo para que se encontrara debajo del cono y simulara la recolección de muestra.
- Realizar 100 iteraciones de subir y bajar el frasco.
- Ejecutar 100 iteraciones de girar los frascos, llegando al frasco 6 a partir del frasco 1.

Los resultados que se obtuvieron con las pruebas anteriores para el prototipo fueron satisfactorios.



Conclusiones

En conclusión, se utilizaron tres iteraciones del Ciclo de Deming para diseñar, desarrollar e implementar un sistema funcional del intercambiador automático de muestreo, para la trampa de succión tipo Rothamsted para insectos de alta movilidad. Para esto, en la primera iteración se identificaron las especificaciones del sistema con base en las necesidades del cliente (INIFAP y agricultores del área), y posteriormente, se detectaron los puntos críticos de la última versión que estuvo en funcionamiento antes de comenzar con esta investigación. Estos primeros puntos críticos fueron indispensables para proponer un diseño adecuado para el intercambiador. Además, con las siguientes iteraciones, se identificaron nuevos puntos críticos debido a fallas en algunas propuestas previas de diseño, o se retomaron los que no se habían satisfecho con anterioridad, para finalmente, obtener un resultado satisfactorio durante las pruebas en las etapas de Verificar de las iteraciones.

Cabe destacar que, para todas las piezas manufacturadas, los nuevos componentes electrónicos, y el software utilizado, se consideró el aspecto económico como se explica más adelante.

Durante la manufactura de las piezas del intercambiador automático de muestreo, además de las consideraciones y restricciones mencionadas, se detectaron elementos que pueden modificar la buena fabricación de una pieza, como son las tolerancias y el *Kerf* que varía en cada proceso y máquina.

Otro ejemplo, al ocupar manufactura aditiva, es como se colocan los huecos estructurales en las piezas, es este caso *honeycomb* (en forma de panal), lo que puede volver más frágil una pieza. El factor humano también interviene en la calidad, pues la calibración de esta impresora (*Ultra 3D Printer* de la marca *PowerSpec*) se hace mediante el apriete y afloje de tres tornillos que mueven la plataforma, acercándola o alejándola de los extrusores, donde la persona que calibra determina cuando es suficiente. Por otro lado, los errores se pueden corregir dando un acabado con un lijado ligero o aplicando acetona pura que, si bien no es la indicada para PLA, funcionó con las piezas sobre las que se probó.

Referente al corte láser, se notó que las placas de material no siempre son de grosor constante, lo que resulta grave si la tolerancia para las piezas manufacturadas es prácticamente nula. Por otro lado, es de gran importancia redondear todas las esquinas de las piezas que se fabricarán con este proceso para evitar la concentración de esfuerzos y su ruptura, por ejemplo, al caerse. Un punto que vale la pena mencionar es que, si bien una pieza mal manufacturada se puede compensar lijándola, implica un gran esfuerzo.

Para ambos procesos de manufactura se tomaron en cuenta aspectos para reducir costos, por ejemplo, en la manufactura aditiva, no se consideró una densidad del 100% para las piezas y se colocaron agujeros cuya función sólo era la de ahorrar material. En cuanto al corte láser, se consideró al ser una opción más económica que la manufactura aditiva,



diseñando las piezas para que no hubiera cortes innecesarios en las piezas. En cuanto al software de diseño de las piezas, se utilizó una opción gratuita de CAD: Tinkercad.

En cuanto a los componentes electrónicos, algunos de ellos estaban establecidos, sin embargo, se comprobó su utilidad y funcionamiento, por lo que algunos de ellos se tuvieron que sustituir al no cumplir con las especificaciones. Un ejemplo fue el motorreductor de plástico que fue remplazado al no poder mover los seis frascos con contenido. Las hojas de especificaciones fueron esenciales para la comprobación de los componentes, no obstante, algunos datos no incluidos se obtuvieron mediante la experimentación, como fue el caso de la corriente en el puente H. Asimismo, el funcionamiento que se describe de cada componente puede variar por factores externos, por ejemplo el servomotor, que fue descartado debido a que, si bien tenía la capacidad de mover todos los frascos con contenido, no se detenía para colocar cada frasco exactamente debajo del cono de red metálica.

Otro punto, referente a los componentes electrónicos, es que algunos de ellos se utilizaron para solucionar los puntos críticos, como fue el caso de los sensores de efecto Hall, que se seleccionaron por ser una opción económica y con la que se obtuvo un funcionamiento conveniente al momento de colocar los frascos en su lugar.

Económicamente, los actuadores y sensores que reemplazaron al motorreductor y al servomotor, resultaron ser una mejor opción, pues su costo total fue de \$270; mientras que en conjunto, por el motorreductor de plástico y el servomotor, se pagaron \$290.

En cuanto a la SBC que se empleó (Raspberry Pi Zero W), fue útil para programar el comportamiento del sistema gracias a las bibliotecas ya existentes, además, se ocupó Python como lenguaje de programación, que tiene la ventaja de ser un lenguaje de alto nivel.

Por otra parte, se hizo la adecuación para definir una arquitectura de IoT con elementos básicos a partir de una arquitectura de tres capas. En cuanto al protocolo de red se investigó sobre diferentes opciones, siendo MQTT la mejor opción para una IoT como esta. Para la última capa se utilizó el *framework* Pimatic, con el que se creó la aplicación desde la cual el cliente puede controlar el sistema remotamente y obtener los datos de los sensores y estados del sistema en general. Esto se comprobó durante las pruebas finales de operación.

Con todo lo anterior se cumplieron los objetivos de la tesis, no obstante, se utilizaron sólo algunas herramientas para que este sistema tuviera un enfoque de la IoT y se pudiera controlar y monitorear, por lo que en un trabajo futuro se utilizaran las herramientas faltantes para configurar un sistema IoT completo.



Trabajo a Futuro

El proceso que se siguió para la elaboración de este proyecto concluyó, sin embargo, aún está pendiente la prueba en la trampa de Mocochá, y el sistema puede ser mejorado si se consideran algunos de los siguientes puntos:

- Agregar el *encoder* magnético disponible para el micro - motorreductor que gira al carrusel para integrar un controlador PID o utilizar un motor a pasos para girar el carrusel en lugar del micro - motorreductor de metal.
- Quitar todos los tornillos, para tener sólo piezas que no sufran corrosión o malgaste debido a las condiciones ambientales.
- Programar una aplicación móvil que funcione vía bluetooth, para poder controlar el intercambiador y obtener los datos en cuestión, cuando se acceda a la cabina de la trampa.
- Ampliar la configuración de los elementos de la arquitectura de IoT para obtener un sistema completo.



Trabajos citados

- 330 ohms. (2019a). *Servo Metálico MG995 - Posición 180 Grados*. Recuperado de <https://www.330ohms.com/products/servo-de-180-grados-mg995>
- 330ohms. (2019b). *Sensor de Corriente No Invasivo 30A*. Recuperado de <https://www.330ohms.com/products/sensor-de-corriente-no-invasivo-30a>
- 330ohms. (2019c). *Motorreductor de Plástico 140 RPM*. Recuperado de <https://www.330ohms.com/products/motorreductor-de-plastico-140-rpm>
- 3G E-Learning LLC. (2018a). Gateway. En *Illustrated Dictionary of Computer Science* (pág. 120). USA.
- 3G E-Learning LLC. (2018b). Internet. En *Illustrated Dictionary of Computer Science* (pág. 145). USA.
- 3G E-Learning LLC. (2018c). Protocol. En *Illustrated Dictionary of Computer Science* (pág. 215). USA.
- AFEL. (2019). *Chips LM35DZ Sensor Temperatura*. Recuperado de <https://afel.cl/producto/chips-lm35dz-sensor-temperatura/?v=2e6507f70a9c>
- Al-Qaseemi, S., Almulhim, H., Almulhim, M., & Rasool, S. (2016). IoT architecture challenges and issues: Lack of standardization. *2016 Future Technologies Conference (FTC)*. San Francisco, CA, USA. doi:10.1109/FTC.2016.7821686
- Aleman, J. (2004). *El Ciclo Shewhart o el Ciclo Deming*. Recuperado de <https://www.explicacion.net/wp-content/uploads/2018/10/El-Ciclo-Shewhart-o-el-Ciclo-Deming-Aleman.pdf>
- Alibaba. (2019). *XS-W 125*70*50mm SSR disipador de calor de aluminio 25A 30A disipador de calor del radiador para la fase de estado sólido relé de*. Recuperado de <https://spanish.alibaba.com/product-detail/xs-w-125-70-50mm-ssr-aluminum-heat-sink-25a-30a-heat-sink-radiator-for-single-phase-solid-state-relay-60528117219.html>
- Ashton, K. (2009). That 'Internet of Things' Thing. *RFID Journal*, 22 de Junio. Recuperado de <https://www.rfidjournal.com/articles/view?4986>
- ATGU, J. (2014, 5 de Febrero). *Qué es SVG*. Recuperado de <https://desarrolloweb.com/articulos/que-es-svg.html>
- Banggood. (2019). *Disipador de Calor de Aleación de Aluminio Para Disipación de Calor SSR del Relé de Estado Sólido*. Recuperado de <https://es.banggood.com/Wholesale->



Aluminum-Alloy-Heat-Sink-For-Solid-State-Relay-SSR-Small-Type-Heat-Dissipation-p-48066.html?cur_warehouse=USA

- Bosch Sensortec. (2011). *Data sheet BMP180 Digital pressure sensor*. Recuperado de https://cdn.shopify.com/s/files/1/1040/8806/files/BST-BMP180-DS000-07_datesheet.pdf?7329687540796527219
- CDMX Electrónica. (2019a). *Motorreductor 6-12 V 100 RPM (1654) Pololu*. Recuperado de <https://www.cdmxelectronica.com/producto/motorreductor-6-12-v-100rpm/>
- CDMX Electrónica. (2019b). *Motorreductor 6-12 V 30 RPM (1333) Pololu*. Recuperado de <https://www.cdmxelectronica.com/producto/motorreductor-6-12-v-30rpm/>
- Contreras, L. (2019, 23 de Julio). *¿Qué tan «ecológico» es realmente el filamento PLA?*. Recuperado de <https://www.3dnatives.com/es/ecologico-realmente-filamento-pla-230720192/>
- Core Electronics. (2019). *MCP3008 - 8-Channel 10-Bit ADC With SPI Interface*. Recuperado de <https://core-electronics.com.au/mcp3008-8-channel-10-bit-adc-with-spi-interface.html>
- CutLaserCut Ltd. (2015). *Understanding the "kerf" of the laser*. Recuperado de <http://www.cutlasercut.com/resources/tips-and-advice/what-is-laser-kerf>
- DiCola, T. (2018a, 7 de Noviembre). *Adafruit_Python_DHT 1.4.0*. Recuperado de https://pypi.org/project/Adafruit_Python_DHT/
- DiCola, T. (2018b, 30 de Agosto). *Adafruit-BMP 1.5.4*. Recuperado de <https://pypi.org/project/Adafruit-BMP/#description>
- doceme. (2015). *Python Spidev*. Recuperado de <https://github.com/doceme/py-spidev>
- D-Robotics UK. (2010, 30 de Julio). *DHT11 Humidity & Temperature Sensor*. Recuperado de <http://www.circuitbasics.com/wp-content/uploads/2015/11/DHT11-Datasheet.pdf>
- Del Valle, L. (2015). *59. Qué son los servicios web*. Recuperado de <https://programarfacil.com/podcast/59-que-son-los-servicios-web/>
- Echun Electronic Co. (s.f.). *Split Core Current Transformer ECS1030-L72*. Recuperado de <https://cdn.sparkfun.com/datasheets/Sensors/Current/ECS1030-L72-SPEC.pdf>
- Eclipse Foundation. (2019). *eclipse paho*. Recuperado de <https://www.eclipse.org/paho/>
- Eclipse Foundation (2004). *Download*. Recuperado de <https://mosquitto.org/download/>
- EcuRed. (2019, 14 de Agosto). *Framework*. Recuperado de <https://www.ecured.cu/Framework>



- Fairchild Semiconductor Corporation. (2008, Marzo). *LM78XX/LM78XXA 3-Terminal 1A Positive Voltage Regulator*. Recuperado de <http://ee-classes.usc.edu/ee459/library/datasheets/LM7805.pdf>
- finsuka. (2017, 07 de Noviembre). *¿Cuánta energía consumen los LEDs?*. Recuperado de https://www.youtube.com/watch?v=c_Egixv_5vY
- FOTEK. (s.f.). *SSR Series Solid State Relay*. Recuperado de <https://cdn.sparkfun.com/datasheets/Components/General/SSR40DA.pdf>
- García, E. (2016, 10 de Noviembre). *El Ciclo de Deming: La gestión y mejora de procesos*. Recuperado de <https://equipo.altran.es/el-ciclo-de-deming-la-gestion-y-mejora-de-procesos/>
- Gartner. (2019). Internet Of Things (iot). En *Gartner Glossary*. Recuperado de <https://www.gartner.com/it-glossary/internet-of-things/>
- IBM. (2019). *Servicios Web*. Recuperado de https://www.ibm.com/support/knowledgecenter/es/SS7K4U_9.0.0/com.ibm.websphere.zseries.doc/ae/cwbs_wbs2.html
- Impresoras3D.com. (2018, 03 de Enero). *Filamento PLA: consejos, características y mucho más*. Recuperado de <https://www.impresoras3d.com/filamento-pla-consejos-caracteristicas-y-mucho-mas/>
- indiamart. (2019). *Fotek SSR, Model Number: SSR-40 DA*. Recuperado de <https://www.indiamart.com/proddetail/fotek-ssr-16359011591.html>
- INEGI. (2015). *Cuéntame... de México*. Recuperado de <http://cuentame.inegi.org.mx/monografias/informacion/yuc/territorio/clima.aspx?tema=me&e=31>
- INNOVAPACK. (2012). *POLIFOAM*. Recuperado de <http://www.innovapack.mx/polifoam.html>
- iX Corp. (2018). *¿Qué Es Un archivo DXF?* Recuperado de <https://arreglatupc.site/index.php/tutoriales/item/2362-que-es-un-archivo-dxf>
- jordisan. (2006, 29 de Septiembre). *¿Qué es un 'framework'?* Recuperado de <https://jordisan.net/blog/2006/que-es-un-framework/>
- Learn Open Energy Monitor. (s.f.). *CT Sensors - Interfacing with an Arduino*. Recuperado de <https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/interface-with-arduino?redirected=true>



- Lovely Electronics. (2018). *DHT11 Digital Relative Humidity & Temperature Sensor Module*. Recuperado de <https://robotskull.in/product/dht11-digital-relative-humidity-temperature-sensor-module/>
- Makerlab-Electronics. (2015). *BMP180 Barometric Pressure Sensor*. Recuperado de <https://www.makerlab-electronics.com/product/bmp180-barometric-pressure-sensor/>
- Microchip Technology. (2008). *MCP3004/3008*. Recuperado de <http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf>
- Mobius. (2006, 12 de Junio). *File: DE15 Connector Pinout.svg*. Recuperado de https://commons.wikimedia.org/wiki/File:DE15_Connector_Pinout.svg
- NeoAttack. (2018). Concepto de Plugin. En *Neowiki*. Recuperado de <https://neoattack.com/neowiki/plugin/>
- Nieto, J., & Seco, M. (1990). Pulgones y su captura mediante trampas: La red Euraphid. *Bol. San. Veg. Plagas*, 16(3), 22 de Febrero, 593-603.
- NoMADA Store. (2019). *Driver de Baja Potencia para Motores [L293D]*. Recuperado de <https://nomada-e.com/store/actuadores-y-drivers/173-driver-para-motores-l293d.html>
- Nuttall, B., & Jones, D. (2015). *gpiozero*. Recuperado de <https://gpiozero.readthedocs.io/en/stable/index.html>
- Pololu Corporation. (2019, Junio). *Micro Metal Gearmotors*. Recuperado de <https://www.pololu.com/file/0J1487/pololu-micro-metal-garmotors-rev-4-1.pdf>
- ProyectosCAD. (2018, 23 de Agosto). *Qué archivos puedo convertir a G-Code*. Recuperado de <https://proyectoscad.com/convertir/g-code/que-archivos-puedo-convertir-a-g-code/#targetText=Actualmente%20el%20formato%20de%20archivo,%C2%ABLenguaje%20de%20teselaci%C3%B3n%20est%C3%A1ndar%C2%BB>
- Python Software Foundation. (2001). *Python*. Recuperado de <https://www.python.org/>
- Qloudea. (2017, 8 de Junio). *¿Qué es TCP/IP?* Recuperado de <https://soporte.qloudea.com/hc/es/articles/115003659065--Qu%C3%A9-es-TCP-IP-#:~:targetText=El%20conjunto%20de%20protocolos%20TCP,cree%20un%20flujo%20de%20datos>
- Raspberry Pi Foundation. (2018a). *GPIO*. Recuperado de <https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/README.md>



- Raspberry Pi Foundation. (2018b). *Raspbian*. Recuperado de <https://www.raspberrypi.org/downloads/raspbian/>
- Raspberry Pi Foundation. (2019a). *FAQs*. Recuperado de <https://www.raspberrypi.org/documentation/faqs/#pi-performance>
- Real Academia Española. (2018). Plaga. En *Diccionario de la lengua española Edición Tricentenario*. Recuperado de <https://dle.rae.es/?id=TIW9QYM|TIWEbvP>
- Regidor, A. (2016, 22 de Noviembre). *¿Qué es un archivo .STL?* Recuperado de <https://www.impression3daily.es/que-es-un-archivo-stl/>
- rwaldron. (2016, 13 de Octubre). *Environmental Monitoring with the Tessel 2*. Recuperado de <https://learn.sparkfun.com/tutorials/environmental-monitoring-with-the-tessel-2>
- Schneider, O. (2019). *pimatic*. Recuperado de <https://pimatic.org/>
- Semle, A. (2016). Protocolos IIoT para considerar. *AADECA REVISTA*, SEPTIEMBRE-OCTUBRE, edición no. 2, 32-35. Recuperado de https://www.editores-srl.com.ar/sites/default/files/aa2_semle_protocolos_ilot.pdf
- Sethi, P., & Sarangi, S. (2017). Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, 2017. Recuperado de <https://www.hindawi.com/journals/jece/2017/9324035/#B54>
- Siemens Industry Software Inc. (2019). *Diseño asistido por ordenador (CAD)*. Recuperado de <https://www.plm.automation.siemens.com/global/es/our-story/glossary/computer-aided-design-cad/12507>
- Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). Estado de México, México: Pearson Educación de México.
- Texas Instruments. (2016, Enero). *L293x Quadruple Half-H Drivers*. Recuperado de <http://www.ti.com/lit/ds/symlink/l293.pdf>
- Texas Instruments. (2017, Diciembre). *LM35 Precision Centigrade Temperature*. Recuperado de Sensors: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- Tolia, M. (2018, 17 de Septiembre). *MQTT QoS Levels (Explained with Examples & When to use Them)*. Recuperado de <https://mntolia.com/mqtt-qos-levels-explained/>
- Torq Pro & Tower Pro. (2014). *MG995*. Recuperado de <https://www.towerpro.com.tw/product/mg995/>
- Ulrich, K., & Eppinger, S. (2013). *Diseño y desarrollo de productos* (Quinta ed.). Mc Graw Hill Education.



- Unisonic Technologies Company. (2012). *U18 LINEAR INTEGRATED CIRCUIT*. Recuperado de <http://www.utc-ic.com/uploadfile/2012/0331/20120331125747992.pdf>
- Van Kranenburg, R. (2007). *The Internet of Things: A Critique of Ambient Technology and the All-Seeing Network of RFID*. Amsterdam, The Netherlands: Institute of Network Cultures.
- Yangzhou Positioning Tech. (s.f.). *49E Hall-Effect Linear Position Sensor*. Recuperado de <https://p.globalsources.com/IMAGES/PDT/SPEC/440/K1139513440.pdf>
- Zator Systems. (2016). *5 Librería Estándar*. Recuperado de <https://www.zator.com/Cpp/E5.htm>



Otras fuentes de consulta

Aosong(Guangzhou) Electronics Co. (s.f.). *Temperature and humidity module*. Recuperado de <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>

Autodesk, Inc. (2019). *TINKERCAD*. Recuperado de <https://www.tinkercad.com/>

Cope, S. (2019). *Mosquitto Username and Password Authentication -Configuration and Testing*. Recuperado de <http://www.steves-internet-guide.com/mqtt-username-password-example/>

Cope, S. (2019a). *How to Install The Mosquitto MQTT Broker on Linux (Ubuntu)*. Recuperado de <http://www.steves-internet-guide.com/install-mosquitto-linux/>

Cope, S. (2019b). *Paho Python MQTT Client Objects*. Recuperado de <http://www.steves-internet-guide.com/client-objects-python-mqtt/>

Cope, S. (2019c). *Python MQTT Client Connections– Working with Connections*. Recuperado de <http://www.steves-internet-guide.com/client-connections-python-mqtt/>

Cope, S. (2019d). *Using The Mosquitto_pub and Mosquitto_sub MQTT Client Tools- Examples*. Recuperado de http://www.steves-internet-guide.com/mosquitto_pub-sub-clients/

Del Valle, L. (2017). Cómo utilizar el sensor DHT11 para medir la temperatura y humedad con Arduino. Recuperado de <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>

Granados, G. (2001). Manejo integrado de plagas. En R. Paliwal, *El maíz en los trópicos. Mejoramiento y producción*. (pp. 295-313) Roma: Organización de las Naciones Unidas para la Agricultura y la Alimentación.

Inkscape. (2019). *INKSCAPE Draw freely*. Recuperado de <https://inkscape.org/>

Kail, M. (2016). *pimatic-mqtt*. Recuperado de <https://pimatic.org/plugins/pimatic-mqtt/>

Llamas, L. (2016, 13 de Septiembre). *MEDIR PRESIÓN DEL AIRE Y ALTITUD CON ARDUINO Y BARÓMETRO BMP180*. Recuperado de <https://www.luisllamas.es/medir-presion-del-aire-y-altitud-con-arduino-y-barometro-bmp180/>

Light, R. (2015, 28 de Septiembre). *Paho Python - MQTT Client Library Encyclopedia*. Recuperado de <https://www.hivemq.com/blog/mqtt-client-library-paho-python/>



- Light, R. (2016). *mosquitto_sub man page*. Recuperado de https://mosquitto.org/man/mosquitto_sub-1.html
- manuti. (2017, 12 de Agosto). *Cómo configurar el Wi-Fi antes de iniciar la Raspberry Pi por primera vez*. Recuperado de <https://raspberryparatorpes.net/comandos/como-configurar-el-wi-fi-antes-de-iniciar-la-raspberry-pi-por-primera-vez/>
- Mecafenix, F. (2017, 18 de Abril). *Servomotor ¿Que es y como funciona?* Recuperado de <https://www.ingmecafenix.com/electricidad-industrial/servomotor/>
- Mecafenix, F. (2019, 9 de Mayo). *¿Qué es una abrazadera? y sus tipos*. Recuperado de <https://www.ingmecafenix.com/otros/tipos-abrazaderas/>
- MecanicApp. (2017, 13 de Enero). *Sensores de Efecto HALL*. Recuperado de <http://mecanicappweb.com/sensores-de-efecto-hall/>
- Moreau, C. (2016, 29 de Septiembre). *5 Key Considerations for Your Laser Cut Design*. Recuperado de <https://makezine.com/2016/09/29/5-important-things-consider-laser-cutting/>
- Narang, S. (2016, 11 de Junio). *Raspberry Pi 3 Tutorial 11 - GPIO Analog Sensor + MCP3008 / MCP3004*. Recuperado de http://invent.module143.com/daskal_tutorial/raspberry-pi-3-gpio-analog-sensor-mcp3008-mcp3004/
- Nuttall, B., & Jones, D. (2015). *Gpiozero*. Recuperado de <https://gpiozero.readthedocs.io/en/stable/>
- pimatic. (2019). *Configuration*. Recuperado de <https://pimatic.teamemo.com/Guide/Configuration>
- pimatic. (2019). *Database Configuration*. Recuperado de <https://pimatic.teamemo.com/Advanced-Configuration/Database-Configuration>
- PowerSpec. (2019). *PowerSpec 3D Printers*. Recuperado de http://www.powerspec.com/3D_printers/3Dindex.phtml
- Pumanet. (2019). *PUM@NET*. Recuperado de <http://www.pumanet.mx/>
- Raspberry Pi Foundation. (2019b). *Raspberry Pi Zero W*. Recuperado de <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>
- Red Hat Inc. (2005). *Capítulo 20. Protocolo SSH*. Recuperado de <https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>
- Schneider, O. (2014a). *pimatic cron plugin*. Recuperado de <https://pimatic.org/plugins/pimatic-cron/>



- Schneider, O. (2014b). *pimatic mobile-frontend plugin*. Recuperado de <https://pimatic.org/plugins/pimatic-mobile-frontend/>
- Schneider, O. (2019). *Configuration Options*. Recuperado de <https://pimatic.org/pages/config/>
- SD Association. (2019). *SD Memory Card Formatter*. Recuperado de https://www.sdcard.org/downloads/formatter_4/
- Sunrom Technologies. (2012, 20 de Junio). *DHT11 - Humidity and Temperature Sensor*. Recuperado de <http://robocraft.ru/files/datasheet/DHT11.pdf>
- The Pi Hut. (2015, 17 de Abril). *Sensors - Pressure, Temperature and Altitude with the BMP180*. Recuperado de <https://thepihut.com/blogs/raspberry-pi-tutorials/18025084-sensors-pressure-temperature-and-altitude-with-the-bmp180>
- Vega, R. (2016, 07 de Abril). *PRIMEROS PASOS CON MQTT*. Recuperado de <https://ricveal.com/blog/primeros-pasos-mqtt/>
- T-Systems International. (2018, 02 de Abril). *Estas son las capas del Internet de las Cosas*. Recuperado de <https://www.t-systemsblog.es/estas-son-las-capas-del-internet-de-las-cosas/>

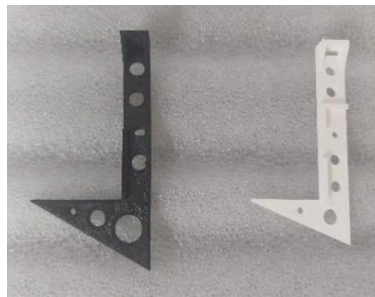


ANEXO A: Piezas fotografiadas de la primera iteración

En este anexo se muestran las fotografías de las piezas que se manufacturaron en la primera iteración del Ciclo de Deming, comparándolas con las piezas que se tenían de la última versión del intercambiador que funcionó en Mococho y a partir del cuál se partió esta investigación.



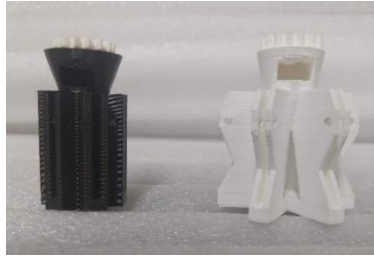
ANEXO A - Figura 1: Soporte para el frasco de recolección.



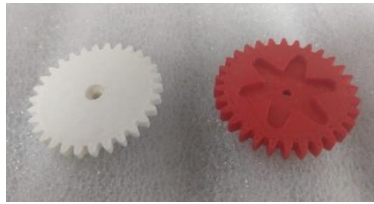
ANEXO A - Figura 2: Escuadra sujetadora del frasco de recolección. A la izquierda la pieza de la versión anterior; a la derecha la nueva pieza.



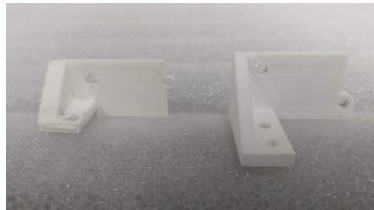
ANEXO A - Figura 3: Paleta del carrusel. A la izquierda la pieza de la versión anterior; a la derecha la nueva pieza.



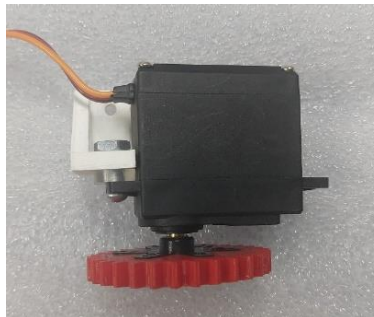
ANEXO A - Figura 4: Eje del carrusel con engrane. A la izquierda el eje del carrusel y el engrane para girar el carrusel de la versión anterior; a la derecha la nueva pieza donde se unen ambas.



ANEXO A - Figura 5: Engrane para servomotor. A la izquierda la pieza de la versión anterior; a la derecha la nueva pieza.



ANEXO A - Figura 6: Porta servomotor. A la izquierda la pieza de la versión anterior; a la derecha la nueva pieza.



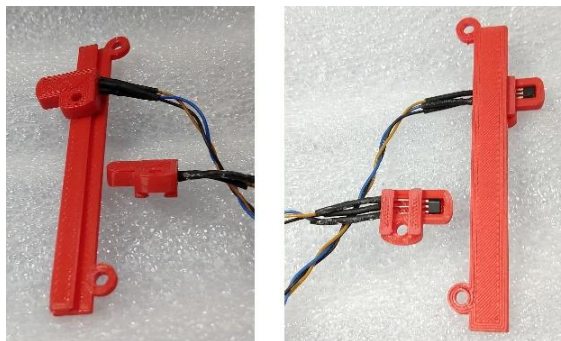
ANEXO A - Figura 7: Ensamble de las partes relacionadas con el servomotor.



ANEXO A - Figura 8: Émbolo. A la izquierda la pieza de la versión anterior; a la derecha la nueva pieza.



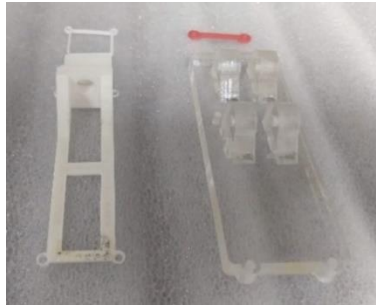
ANEXO A - Figura 9: Guía - porta motor. A la izquierda la pieza de la versión anterior; a la derecha la nueva pieza.



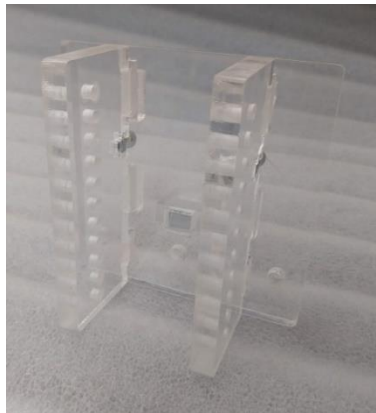
ANEXO A - Figura 10: Porta - sensores vertical. A la izquierda la vista frontal; a la derecha la vista posterior.



ANEXO A - Figura 11: Cople para motor. A la izquierda la pieza de la versión anterior; a la derecha la nueva pieza.



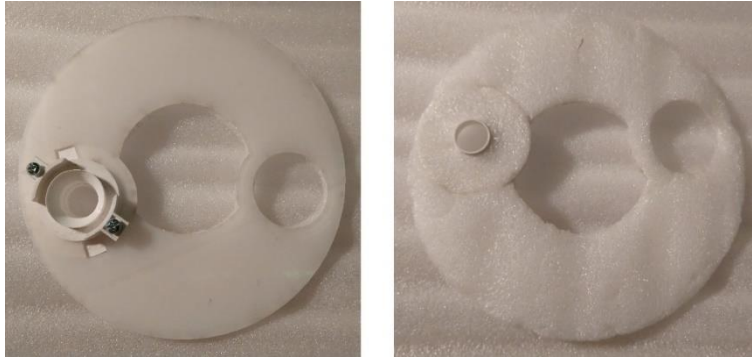
ANEXO A - Figura 12: Tapa - sujetador. A la izquierda la pieza de la versión anterior; a la derecha las nuevas piezas (Tapa - sujetador e inmovilizador del motor).



ANEXO A - Figura 13: Sujeción del sistema.



ANEXO A - Figura 14: Cople para la tapa.



ANEXO A - Figura 15: Tapa con cople. A la Izquierda vista desde arriba; a la derecha vista desde abajo con la capa de polifoam.

ANEXO B: Piezas fotografiadas de la segunda iteración

En este anexo se muestran las fotografías de las piezas que se manufacturaron en la segunda iteración del Ciclo de Deming.



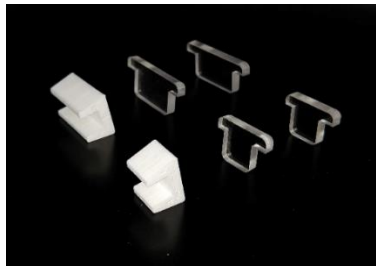
ANEXO B - Figura 1: Porta - micro - motorreductor, micro - motorreductor, y su cople.



ANEXO B - Figura 2: Porta - micro - motorreductor, micro - motorreductor, y su cople, ensamblados al intercambiador.

ANEXO C: Piezas fotografiadas de la tercera iteración

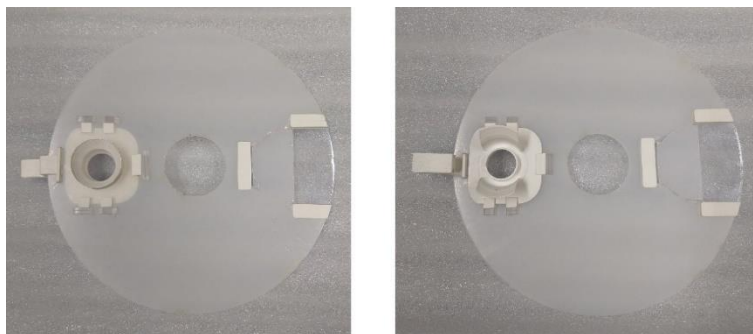
En este anexo se muestran las fotografías de las piezas que se manufacturaron en la tercera iteración del Ciclo de Deming.



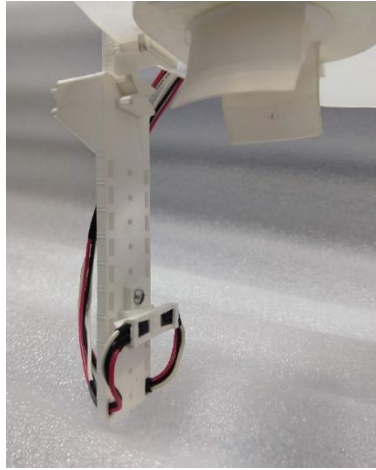
ANEXO C - Figura 1: Piezas en forma de "C" y "T" para limitar el movimiento del cople de la tapa.



ANEXO C - Figura 2: Pieza para limitar el movimiento de las piezas para sostener los sensores de efecto Hall de tipo lineal.



ANEXO C - Figura 3: Tapa con el cople ajustado con las piezas en forma de "C" y "T"; también se muestra la figura que se desliza sobre los rieles en forma de H; así mismo, se muestra una de las piezas para sostener los sensores de efecto Hall de tipo lineal, ajustada a la tapa con la otra pieza limitante. Además, a la izquierda, la vista desde arriba muestra el cople; a la derecha, en la vista desde abajo se observa la guía para alinear el frasco.



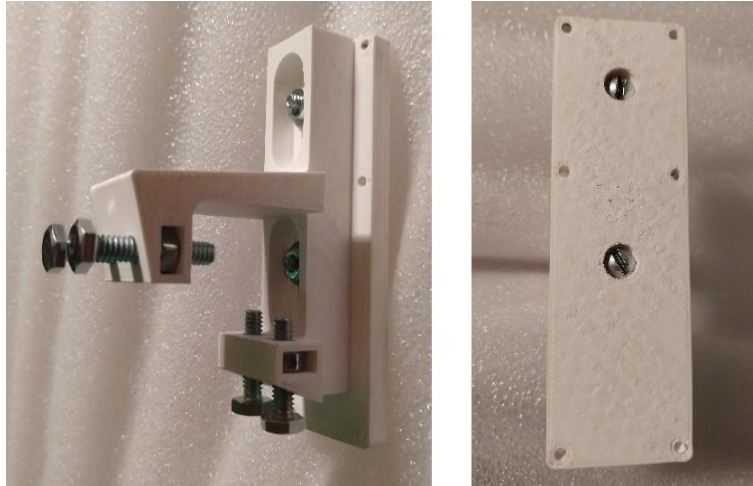
ANEXO C - Figura 4: piezas montadas para sostener los sensores de efecto Hall de tipo lineal.



ANEXO C - Figura 5: Aro sujetador del frasco de recolección modificado para poder introducir el imán. Vista desde abajo.



ANEXO C - Figura 6: Caja para micro - motorreductor que gira al carrusel, ensamblada al intercambiador.



ANEXO C - Figura 7: A la izquierda piezas para sujeción del sistema, incluyendo la Tapa - sujetador. A la derecha sólo la tapa - sujetador.

ANEXO D: Código para los sensores

En este anexo se muestra el código para los sensores de temperatura, humedad, presión y corriente. Fue escrito en la SBC utilizando Python.

```
import Adafruit_BMP.BMP085 as BMP
import Adafruit_DHT as DHT
from gpiozero import MCP3008, InputDevice, OutputDevice
from time import sleep
#Imprime si el émbolo esta arriba o abajo
def SenEmbolo():
    #Asigna el pin de la raspberry que habilitera y deshabilitara el sensor de
    efecto hall de arriba
    senArriba_enable = OutputDevice(19)
    #Asigna el pin de la raspberry que habilitera y deshabilitara el sensor de
    efecto hall de abajo
    senAbajo_enable = OutputDevice(6)
    #Asigna el pin de la raspberry que leera el valor del sensor de efecto hall
    de arriba
    senArriba = InputDevice(13,pull_up=True)
    #Asigna el pin de la raspberry que leera el valor del sensor de efecto hall
    de abajo
    senAbajo = InputDevice(5,pull_up=True)
    #Habilita pines de los sensores
    senArriba_enable.on()
    senAbajo_enable.on()
    if(senArriba.is_active == False):
        return 'arriba'
    elif(senAbajo.is_active == False):
        return 'abajo'
    else:
        return 'en posible movimiento'

#Imprime los datos sensados por el sensor BMP180
def SenBmp(parametro):
    sensorBmp = BMP.BMP085()
    if parametro == 'T':
        tempBmp = sensorBmp.read_temperature()
        print('Temperatura = {0:0.2f} *C'.format(tempBmp))
        return tempBmp
    elif parametro == 'P':
        presBmp = sensorBmp.read_pressure()
```




```

    print('Presion = {0:0.2f} Pa'.format(presBmp))
    return presBmp
elif parametro == 'A':
    altBmp = sensorBmp.read_altitude()
    print('Altitud = {0:0.2f} m'.format(altBmp))
    return altBmp
elif parametro == 'PM':
    presmarBmp = sensorBmp.read_sealevel_pressure()
    print('Presion a nivel del Mar = {0:0.2f} Pa'.format(presmarBmp))
    return presmarBmp
else:
    print("Parametros del sensor BMP: T -> Temperatura, P -> Presion, A ->
Altitud, PM -> Presion a nivel del mar")
    return 0

#Imprime los datos sensados por el sensor DHT11
def SenDht(parametro):
    sensorDht = DHT.DHT11
    pinDht = 4
    humDht, tempDht = DHT.read_retry(sensorDht, pinDht)
    if parametro == 'T':
        print('Temperatura = {0:0.2f} *C'.format(tempDht))
        return tempDht
    elif parametro == 'H':
        print('Humedad = {0:0.2f} %'.format(humDht))
        return humDht
    else:
        print("Parametros del sensor DHT11: T -> Temperatura, H -> Humedad")
        return 0

#Imprime la temperatura del motor sensada por el lm35
def SenLm(parametro):
    if parametro == 'T':
        #Se indica el canal en el Convertidor Analogico Digital MCP3008 por el
que se leera el dato
        lm = MCP3008(2)
        #voltage es el valor de voltaje en el canal indicado. Dado que 1C es
equivalente a 10 mV, se multiplica el valor de voltaje por 100
        tempLm = lm.voltage*100
        print('Temperatura = {0:0.2f} *C'.format(tempLm))
        return tempLm
    else:
        print("Parametro del sensor LM35: T -> Temperatura")
        return 0

#Imprime la corriente del motor sensada por el ECS1030

```



```
def SenEcs(parametro):
    #res es la resistencia de carga colocada y rgiro es la relacion de giro.
    res = 68
    rgiro= 2000
    if parametro == 'C':
        ecs = MCP3008(1)
        cEcs = (ecs.voltage/res)*rgiro
        print('Corriente = {0:0.2f} A'.format(cEcs))
        return cEcs
    else:
        print("Parametro del sensor ECS1030: C -> Corriente")
        return 0
```



ANEXO E:

Código para mover el émbolo

En este anexo se muestra el código para elevar y descender los frascos de recolección. Fue escrito en la SBC utilizando Python.

```

from gpiozero import Motor, OutputDevice, InputDevice, PWMLED
from time import sleep

#Mueve el embolo hacia arriba hasta que el sensor de efecto hall lo detenga
def Arriba():
    #Asigna los pines de la raspberry para el motorreductor
    motorv = Motor(24, 25)
    #Asigna el pin de la raspberry que va al enable del puente H
    motorv_enable = PWMLED(23)
    motorv_enable.value = 0
    #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor de
    efecto hall de arriba
    senArriba_enable = OutputDevice(19)
    #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor de
    efecto hall de abajo
    senAbajo_enable = OutputDevice(6)
    #Asigna el pin de la raspberry que leera el valor del sensor de efecto hall
    de arriba
    senArriba = InputDevice(13,pull_up=True)
    #Asigna el pin de la raspberry que leera el valor del sensor de efecto hall
    de abajo
    senAbajo = InputDevice(5,pull_up=True)

    #on habilita el pin de la raspberry del sensor de efecto hall de arriba
    senArriba_enable.on()
    #Si el sensor tiene un valor de true
    if (senArriba.is_active):
        #Mientras que el sensor tenga un valor "True" movera el motor. El sensor
        tendra un valor "False" en cuanto detecte el campo magnetico
        while (senArriba.is_active):
            print("Motor encendido hacia arriba")
            #on habilita la senal para el enable del puente H
            motorv_enable.value=1
            #backward gira el motor hacia atras, en este caso hacia arriba
            motorv.backward()
            sleep(0.1)
        #stop detiene el motor

```



```

motorv.stop()
#off deshabilita la senal para el enable del puente H
motorv_enable.value = 0
print("Vaso esta Arriba")
else:
    print("Vaso esta Arriba")
#off deshabilita el pin de la raspberry del sensor de efecto hall de
arriba
senArriba_enable.off()
#Libera los pines llamando a Libera
motorv.close()
motorv_enable.close()
senArriba_enable.close()
senAbajo_enable.close()
senArriba.close()
senAbajo.close()

#Mueve el embolo hacia abajo hasta que el sensor de efecto hall lo detenga
def Abajo():
    #Asigna los pines de la raspberry para el motorreductor
    motorv = Motor(24, 25)
    #Asigna el pin de la raspberry que va al enable del puente H
    motorv_enable = PWMLED(23)
    motorv_enable.value = 0
    #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor de
efecto hall de arriba
    senArriba_enable = OutputDevice(19)
    #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor de
efecto hall de abajo
    senAbajo_enable = OutputDevice(6)
    #Asigna el pin de la raspberry que leera el valor del sensor de efecto hall
de arriba
    senArriba = InputDevice(13,pull_up=True)
    #Asigna el pin de la raspberry que leera el valor del sensor de efecto hall
de abajo
    senAbajo = InputDevice(5,pull_up=True)

senAbajo_enable.on()
if(senAbajo.is_active):
    while (senAbajo.is_active):
        print("Motor encendido hacia abajo")
        motorv_enable.value = 1
        #forward gira el motor hacia adelante, en este caso hacia abajo
        motorv.forward()
        sleep(0.1)
    motorv.stop()

```



```
    motorv_enable.value = 0
    print("Vaso esta Abajo")
else:
    print("Vaso esta Abajo")
senAbajo_enable.off()

motorv.close()
motorv_enable.close()
senArriba_enable.close()
senAbajo_enable.close()
senArriba.close()
senAbajo.close()
```



ANEXO F:

Código para mover el carrusel

En este anexo se muestra el código para girar el carrusel y llevar el control del frasco que se coloca debajo del cono de red metálica para tomar una muestra. Fue escrito en la SBC utilizando Python

```

from gpiozero import MCP3008, Motor, OutputDevice, PWMLED
from time import sleep

#Mueve al frasco deseado, vasofin es el no. de frasco deseado
def Mov(vasofin):
    try:
        vasoact = 1
        #En VActual.txt se lee el no. de frasco en el que se encuentra
        archivo = open("VActual.txt")
        vasoact = int(archivo.readline())
        archivo.close()
        #Se comprueba que sea un no. valido
        if (vasofin >= 1) and (vasofin <= 6):
            #Si el frasco deseado es el siguiente se ejecuta Sig. Si es el
            anterior se ejecuta Ant. Si es el mismo lo indica.
            if (vasoact < vasofin):
                while (vasoact < vasofin):
                    Sig()
                    vasoact = vasoact + 1
                    print(vasoact)
            elif (vasoact > vasofin):
                while (vasoact > vasofin):
                    Ant()
                    vasoact = vasoact - 1
                    print(vasoact)
            elif (vasoact == vasofin):
                print("Ya se encuentra en el vaso ", vasofin)
        else:
            print("Opcion invalida")

    except KeyboardInterrupt:
        print("INTERRUPCION MANUAL")

    finally:
        #Guarda en el archivo el no. de frasco en que se quedo antes de la
        interrupcion

```



```

archivo = open("VActual.txt","w")
archivo.write(str(vasoact))
archivo.close()
#Mueve al frasco siguiente
def Sig():
    try:
        #Asigna los pines de la raspberry para el motorreductor
        motorh = Motor(7, 20)
        #Asigna el pin de la raspberry que va al enable del puente H
        motorh_enable = PWMLED(12)
        motorh_enable.value = 0
        #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor
de efecto hall de arriba
        senUno_enable = OutputDevice(21)
        #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor
de efecto hall de abajo
        senDos_enable = OutputDevice(26)
        #Se indica el canal en el Convertidor Analogico Digital MCP3008 por el
que se leera el dato
        senUno = MCP3008(3)
        #Se indica el canal en el Convertidor Analogico Digital MCP3008 por el
que se leera el dato
        senDos = MCP3008(4)
        #Habilita los sensores
        senUno_enable.on()
        senDos_enable.on()
        #Si el frasco esta antes del centro, arrancalo hasta estar fuera del
umbral de los sensores
        if (((senUno.voltage < 1.64) and (senDos.voltage < 1.64)) or
(senUno.voltage > 1.95)):
            motorh_enable.value = 0.2
            motorh.forward()
            sleep(0.6)
        #Si el frasco esta en el centro, arranca y si llega al umbral de los
sensores, sal
        else:
            motorh_enable.value = 0.2
            motorh.forward()
            sleep(0.3)
            while ((senUno.voltage >= 1.63) or (senDos.voltage >= 1.63)):
                motorh_enable.value = 0.2
        #Baja la velocidad
        motorh_enable.value = 0.1
        while (senDos.voltage < 1.665):
            n=1
        #Detente cuando llegue al sensor y espera a que se estabilice

```



```

motorh_enable.value = 0
sleep(0.4)
#Arranca en sentido contrario. Detente al llegar al sensor y espera a
que se estabilice
while senUno.voltage < 1.664:
    motorh_enable.value = 0.14
    motorh.backward()
motorh_enable.value = 0
sleep(0.4)
#Arranca hasta llegar al sensor, detente y espera a que se estabilice
while senDos.voltage < 1.658:
    motorh_enable.value = 0.13
    motorh.forward()
motorh_enable.value = 0
sleep(0.4)
#Arranca en sentido contrario, detente y espera
while senUno.voltage < 1.644:
    motorh_enable.value = 0.12
    motorh.backward()
motorh_enable.value = 0
sleep(0.4)
#Arranca, detente y espera
while senDos.voltage < 1.642:
    motorh_enable.value = 0.11
    motorh.forward()
motorh_enable.value = 0
sleep(0.4)
#Arranca en sentido contrario, detente y espera mas tiempo para
estabilizacion completa.
while senUno.voltage < 1.63:
    motorh_enable.value = 0.1
    motorh.backward()
motorh_enable.value = 0
sleep(1)
#Detente totalmente
motorh.stop()
except:
    print("Error en Sig")
finally:
    #Cierra los pines usados
    motorh.close()
    motorh_enable.close()
    senUno_enable.close()
    senDos_enable.close()
    senUno.close()
    senDos.close()

```




```

#Mueve al frasco anterior
def Ant():
    try:
        #Asigna los pines de la raspberry para el motorreductor
        motorh = Motor(7, 20)
        #Asigna el pin de la raspberry que va al enable del puente H
        motorh_enable = PWMLED(12)
        motorh_enable.value = 0
        #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor
de efecto hall de arriba
        senUno_enable = OutputDevice(21)
        #Asigna el pin de la raspberry que habilitara y deshabilitara el sensor
de efecto hall de abajo
        senDos_enable = OutputDevice(26)
        #Se indica el canal en el Convertidor Analogico Digital MCP3008 por el
que se leera el dato
        senUno = MCP3008(3)
        #Se indica el canal en el Convertidor Analogico Digital MCP3008 por el
que se leera el dato
        senDos = MCP3008(4)
        #Habilita los sensores
        senUno_enable.on()
        senDos_enable.on()
        #Si el frasco esta antes del centro
        if (((senUno.voltage < 1.64) and (senDos.voltage < 1.64)) or
(senDos.voltage > 1.95)):
            motorh_enable.value = 0.2
            motorh.backward()
            sleep(0.6)
        #Si el frasco esta en el centro
        else:
            motorh_enable.value = 0.2
            motorh.backward()
            sleep(0.3)
            while ((senUno.voltage >= 1.63) or (senDos.voltage >= 1.63)):
                motorh_enable.value = 0.2
        #Baja la velocidad
        motorh_enable.value = 0.1
        while senUno.voltage < 1.664:
            n=1
            motorh_enable.value = 0
            sleep(0.4)
        #Arranca en sentido contrario, detente y espera
        while senDos.voltage < 1.666:
            motorh_enable.value = 0.14
            motorh.forward()

```



```
motorh_enable.value = 0
sleep(0.4)
#Arraca, detente y espera
while senUno.voltage < 1.645:
    motorh_enable.value = 0.13
    motorh.backward()
motorh_enable.value = 0
sleep(0.4)
#Arranca en sentido contrario, detente y espera
while senDos.voltage < 1.648:
    motorh_enable.value = 0.12
    motorh.forward()
motorh_enable.value = 0
sleep(0.4)
#Arranca, detente y espera
while senUno.voltage < 1.643:
    motorh_enable.value = 0.11
    motorh.backward()
motorh_enable.value = 0
sleep(0.4)
#Arranca en sentido contrario, detente y espera mas
while senDos.voltage < 1.64:
    motorh_enable.value = 0.10
    motorh.forward()
motorh_enable.value = 0
sleep(1)
#Detente totalmente
motorh.stop()
except:
    print("Error en Ant")
finally:
    #Cierra los pines usados
    motorh.close()
    motorh_enable.close()
    senUno_enable.close()
    senDos_enable.close()
    senUno.close()
    senDos.close()
```



ANEXO G:

Código del cliente en la SBC

En este anexo se muestra el código que considera a la SBC como cliente para suscribirse y publicar en los diferentes tópicos. Se utilizó Python.

```
import paho.mqtt.client as mqtt
from time import sleep
from gpiozero import LED
import Moto
import Act
import Sens
#Se asignan los pines de la raspberry pi para los LED
amarillo = LED(27)
azul = LED(22)
#Se define la ip del broker
broker_ip="XXX.XXX.XX.XXX"
#Se verifica el exito de la conexion, en una conexion exitosa rc(result
code) es 0
def on_connect(client, userdata, flags, rc):
    if rc==0:
        print("Conexion exitosa, codigo: ", rc)
        client.connected_flag = True
        azul.on()
    else:
        print("Error de conexion, codigo: ", rc)
        azul.off()
#Se define lo que se hará con los sensores cuando llegue el mensaje al topic
sensor
def on_message_sensores(client, userdata, message):
    #Publica todo lo de los sensores
    print("Mensaje en /dato: "+message.payload.decode())
    client.publish("trampa/mococha/sensor/dht/temperatura", Sens.SenDht('T'),
qos=1)
    client.publish("trampa/mococha/sensor/dht/humedad", Sens.SenDht('H'),
qos=1)
    client.publish("trampa/mococha/sensor/bmp/temperatura", Sens.SenBmp('T'),
qos=1)
    client.publish("trampa/mococha/sensor/bmp/presion", Sens.SenBmp('P'),
qos=1)
    client.publish("trampa/mococha/sensor/bmp/altitud", Sens.SenBmp('A'),
qos=1)
```



```

    client.publish("trampa/mococho/sensor/bmp/presionmar", Sens.SenBmp('PM'),
qos=1)
    client.publish("trampa/mococho/sensor/extractor/temperatura",
Sens.SenLm('T'), qos=1)
    client.publish("trampa/mococho/sensor/extractor/corriente",
Sens.SenEcs('C'), qos=1)
    client.publish("trampa/mococho/sensor/embolo/estado", Sens.SenEmbolo(),
qos=1)
    archivo = open("VActual.txt")
    mensaje = int(archivo.readline())
    archivo.close()
    client.publish("trampa/mococho/sensor/carrusel/frasco", mensaje, qos=1)
#Se define el ciclo completo de bajar, girar y subir un frasco
def on_message_ciclo(client, userdata, message):
    print("Mensaje en /ciclo: "+message.payload.decode())
    Moto.Abajo()
    if message.payload.decode() == '1':
        Act.Mov(1)
        Moto.Arriba()
    elif message.payload.decode() == '2':
        Act.Mov(2)
        Moto.Arriba()
    elif message.payload.decode() == '3':
        Act.Mov(3)
        Moto.Arriba()
    elif message.payload.decode() == '4':
        Act.Mov(4)
        Moto.Arriba()
    elif message.payload.decode() == '5':
        Act.Mov(5)
        Moto.Arriba()
    elif message.payload.decode() == '6':
        Act.Mov(6)
        Moto.Arriba()
#Se define lo que se hará con el embolo según el mensaje recibido
def on_message_movembolo(client, userdata, message):
    print("Message Embolo: "+message.payload.decode())
    if message.payload.decode() == 'arriba':
        Moto.Arriba()
    elif message.payload.decode() == 'abajo':
        Moto.Abajo()
#Se define lo que se hará con el carrusel según el mensaje recibido
def on_message_movcarrusel(client, userdata, message):
    print("Mensaje Carrusel: "+message.payload.decode())
    Moto.Abajo()
    archivo = open("VActual.txt")

```



```

fact = int(archivo.readline())
archivo.close()
fobj = int(message.payload.decode())
if (fobj >= 1) and (fobj <= 6):
    Act.Mov(vasofin)
elif fobj == 7:
    Act.Mov(fact+1)
elif fobj == 8:
    Act.Mov(fact-1)
#Inicio
amarillo.on() #Enciende LED amarillo que indica que la raspberry inicio
correctamente y empezo a ejecutar script
mqtt.Client.connected_flag = False #Crea bandera y asigna como False
client = mqtt.Client() #Crea nueva instancia
client.on_connect = on_connect #Llama a funcion para verificar conexion
exitosa
client.loop_start() #Empieza loop
client.username_pw_set("*****", "*****") #Autenticacion
client.connect(broker_ip, port=$$$$) #conectar al broker
while not client.connected_flag: #Mientras que la bandera siga como False y
no True
    print("En espera de conexión")
    sleep(1) #Espera a que se conecte
print("Conectado")
client.subscribe("trampa/#", qos=1) #Se suscribe a todos los topics
sleep(1)
client.message_callback_add("trampa/mococha/sensor/dato",
on_message_sensores) #Llama a funcion que se hará si llega mensaje al topic
sensores
client.message_callback_add("trampa/mococha/actuador/ciclo",
on_message_ciclo) #Llama a funcion que se hará si llega mensaje al topic
actuador
client.message_callback_add("trampa/mococha/actuador/extractor/onoff",
on_message_onoff)
client.message_callback_add("trampa/mococha/actuador/embolo/mover",
on_message_movembolo) #Llama a funcion que se hará si llega mensaje al topic
.../mover
client.message_callback_add("trampa/mococha/actuador/carrusel/movaf",
on_message_movcarrusel) #Llama a funcion que se hará si llega mensaje al
topic .../movaf
client.loop_forever() #Estara siempre en espera de mensajes

```



ANEXO H:

Fracción del config.json de Pimatic

En este anexo se muestra la fracción del archivo config.json de Pimatic, donde se configuraron los tipos de dispositivos, los *topics* a los que pertenecen, así como algunas reglas.

```

...
"devices": [
  {
    "brokerId": "Mosquitto",
    "attributes": [
      {
        "name": "temperatura en grados centigrados",
        "topic": "trampa/mococha/sensor/dht/temperatura",
        "qos": 1,
        "type": "string",
        "acronym": "Temperatura [°C]:",
        "messageMap": {}
      },
      {
        "name": "porcentaje de humedad",
        "topic": "trampa/mococha/sensor/dht/humedad",
        "qos": 1,
        "type": "string",
        "acronym": "Humedad [%]:",
        "messageMap": {}
      }
    ],
    "xAttributeOptions": [],
    "id": "sen-dht",
    "name": "DHT11",
    "class": "MqttSensor"
  },
  {
    "brokerId": "Mosquitto",
    "attributes": [
      {
        "name": "temperatura en grados centigrados",
        "topic": "trampa/mococha/sensor/bmp/temperatura",
        "qos": 1,
        "type": "string",
        "acronym": "Temperatura [°C]:",

```



```

    "messageMap": {}
  },
  {
    "name": "altitud en metros",
    "topic": "trampa/mococha/sensor/bmp/altitud",
    "qos": 1,
    "type": "string",
    "acronym": "Altitud [m]:",
    "messageMap": {}
  },
  {
    "name": "presión en pascales",
    "topic": "trampa/mococha/sensor/bmp/presion",
    "qos": 1,
    "type": "string",
    "acronym": "Presión [Pa]:",
    "messageMap": {}
  },
  {
    "name": "presion a nivel del mar en pascales",
    "topic": "trampa/mococha/sensor/bmp/presionmar",
    "qos": 1,
    "type": "string",
    "acronym": "Presión a nivel del mar [Pa]:",
    "messageMap": {}
  }
],
"xAttributeOptions": [],
"id": "sen-bmp",
"name": "BMP180",
"class": "MqttSensor"
},
{
  "brokerId": "Mosquitto",
  "attributes": [
    {
      "name": "temperatura en grados centigrados",
      "topic": "trampa/mococha/sensor/extractor/temperatura",
      "qos": 1,
      "type": "string",
      "acronym": "Temperatura [°C]:",
      "messageMap": {}
    },
    {
      "name": "corriente en amperes",
      "topic": "trampa/mococha/sensor/extractor/corriente",

```



```

        "qos": 1,
        "type": "string",
        "acronym": "Corriente [A]:",
        "messageMap": {}
    }
],
"xAttributeOptions": [],
"id": "sen-extractor",
"name": "Extractor",
"class": "MqttSensor"
},
{
    "brokerId": "Mosquitto",
    "attributes": [
        {
            "name": "estado",
            "topic": "trampa/mococha/sensor/embolo/estado",
            "qos": 1,
            "type": "string",
            "messageMap": {}
        }
    ],
    "xAttributeOptions": [],
    "id": "sen-embolo",
    "name": "Estado:",
    "class": "MqttSensor"
},
{
    "brokerId": "Mosquitto",
    "attributes": [
        {
            "name": "numero",
            "topic": "trampa/mococha/sensor/carrusel/frasco",
            "qos": 1,
            "type": "string",
            "acronym": "no.",
            "messageMap": {}
        }
    ],
    "xAttributeOptions": [],
    "id": "sen-carrusel",
    "name": "Frasco actual",
    "class": "MqttSensor"
},
{
    "brokerId": "Mosquitto",

```




```

"topic": "trampa/mococha/actuador/extractor/onoff",
"offMessage": "2",
"qos": 1,
"xOnLabel": "On",
"xOffLabel": "Off",
"id": "act-extractor",
"name": "Encendido/Apagado",
"class": "MqttSwitch"
},
{
"brokerId": "Mosquitto",
"buttons": [
{
"id": "arriba",
"text": "Subir",
"topic": "trampa/mococha/actuador/embolo/mover",
"message": "arriba",
"qos": 1,
"confirm": true
},
{
"id": "abajo",
"text": "Bajar",
"topic": "trampa/mococha/actuador/embolo/mover",
"message": "abajo",
"qos": 1,
"confirm": true
}
],
"id": "act-embolo",
"name": "Movimiento",
"class": "MqttButtons"
},
{
"brokerId": "Mosquitto",
"enableActiveButton": true,
"buttons": [
{
"id": "frasco-ant",
"text": "Anterior",
"topic": "trampa/mococha/actuador/carrusel/movaf",
"message": "8",
"qos": 1,
"confirm": true
}
],
}

```



```
"id": "frasco-sig",
"text": "Siguiente",
"topic": "trampa/mococha/actuador/carrusel/movaf",
"message": "7",
"qos": 1,
"confirm": true
},
{
  "id": "frasco-1",
  "text": "1",
  "topic": "trampa/mococha/actuador/carrusel/movaf",
  "message": "1",
  "qos": 1,
  "confirm": true
},
{
  "id": "frasco-2",
  "text": "2",
  "topic": "trampa/mococha/actuador/carrusel/movaf",
  "message": "2",
  "qos": 1,
  "confirm": true
},
{
  "id": "frasco-3",
  "text": "3",
  "topic": "trampa/mococha/actuador/carrusel/movaf",
  "message": "3",
  "qos": 1,
  "confirm": true
},
{
  "id": "frasco-4",
  "text": "4",
  "topic": "trampa/mococha/actuador/carrusel/movaf",
  "message": "4",
  "qos": 1,
  "confirm": true
},
{
  "id": "frasco-5",
  "text": "5",
  "topic": "trampa/mococha/actuador/carrusel/movaf",
  "message": "5",
  "qos": 1,
  "confirm": true
}
```



```

    },
    {
      "id": "frasco-6",
      "text": "6",
      "topic": "trampa/mococho/actuador/carrusel/movaf",
      "message": "6",
      "qos": 1,
      "confirm": true
    }
  ],
  "id": "act-carrusel",
  "name": "Mover a frasco",
  "class": "MqttButtons"
}
],
"rules": [
  {
    "id": "reg-sensa",
    "name": "Recibe mqtt",
    "rule": "when its 13:00 then publish mqtt message \"sensando\" on
topic \"trampa/mococho/sensor/dato\" on broker Mosquitto qos: 1",
    "active": true,
    "logging": true
  },
  {
    "id": "reg-ciclof1",
    "name": "Baja, gira y sube frasco 1",
    "rule": "when its 13:10 then publish mqtt message \"1\" on topic
\"trampa/mococho/actuador/ciclo\" on broker Mosquitto qos: 1",
    "active": true,
    "logging": true
  },
  {
    "id": "reg-ciclof2",
    "name": "Baja, gira y sube frasco 2",
    "rule": "when its 13:20 then publish mqtt message \"2\" on topic
\"trampa/mococho/actuador/ciclo\" on broker Mosquitto qos: 1",
    "active": true,
    "logging": true
  },
  {
    "id": "reg-ciclof3",
    "name": "Baja, gira y sube frasco 3",
    "rule": "when its 13:30 then publish mqtt message \"3\" on topic
\"trampa/mococho/actuador/ciclo\" on broker Mosquitto qos: 1",
    "active": true,

```



```
"logging": true
},
{
  "id": "reg-ciclof4",
  "name": "Baja, gira y sube frasco 4",
  "rule": "when its 13:40 then publish mqtt message \"4\" on topic
  \\\"trampa/mococho/actuador/ciclo\\\" on broker Mosquitto qos: 1",
  "active": true,
  "logging": true
},
{
  "id": "reg-ciclof5",
  "name": "Baja, gira y sube frasco 5",
  "rule": "when its 13:50 then publish mqtt message \"5\" on topic
  \\\"trampa/mococho/actuador/ciclo\\\" on broker Mosquitto qos: 1",
  "active": true,
  "logging": true
},
{
  "id": "reg-ciclof6",
  "name": "Baja, gira y sube frasco 6",
  "rule": "when its 14:00 then publish mqtt message \"6\" on topic
  \\\"trampa/mococho/actuador/ciclo\\\" on broker Mosquitto qos: 1",
  "active": true,
  "logging": true
}
],
...
```

