

Universidad Autónoma del Estado de Morelos "UAEM"

Facultad de Ciencias Químicas e Ingeniería "FCQel"

Programa Educativo: Ing. Eléctrica - Electrónica

Diseño de un módulo de monitoreo para un laboratorio remoto mediante dispositivos IoT

Tesis para Obtener el Grado de:

Ingeniería Eléctrica - Electrónica

AUTOR:

Andrés Michel Rodríguez Gordillo

ASESOR: DR. JOSE GERARDO VERA DIMAS

SINODALES:

- DR. JOSÉ GERARDO VERA DIMAS.
- DR. OUTMANE OUBRAM.
- DR. LUIS CISNEROS VILLALOBOS.
- DRA. ALINA MARTÍNEZ OROPEZA.
- MTRO. FCO. JAVIER BECERRA GONZAÁLEZ.

CUERNAVACA, MORELOS

MAYO 2024

DEDICATORIA

A mis padres

Este presente trabajo de grado es dedicado con un inmenso amor a mis padres Sr. Andrés Rodríguez Anzures y Sra. Magnolia Gordillo Ríos los cuales con mucha entrega, esmero y sacrificio me han permitido alcanzar y ver cumplidas mis metas.

A mis Hermanas

Sharon Magnolia Rodríguez Gordillo, Shaila Magnolia Rodríguez Gordillo y Magnolia Shareni Rodríguez Gordillo. A ustedes les debo todo en mi vida, es por ello por lo que en este escrito plasmo mi más sincera admiración y agradecimiento a quienes me dieron la vida y me han acompañado en ella, sin los cuales este logro no sería posible.

AGRADECIMIENTOS

En primer lugar, quiero agradecer A mis padres y hermanas, gracias, por acompañarme en todo mi trayecto académico, por su apoyo incondicional, dedicación, sacrificio y esfuerzo a lo largo de toda mi vida.

A la Universidad Autónoma del Estado de Morelos (UAEM), y la Facultad de Ciencias Químicas e Ingeniería (FCQel), gracias, por permitirme formar parte de su prestigiosa institución, por brindarme su apoyo y formarme profesionalmente en esta apasionante carrera.

A mi comité sinodal, por su tiempo, paciencia y apoyo en mi proyecto de tesis. Gracias por todos sus consejos y asesoramientos.

A mi novia, amigos y compañeros del Laboratorio, gracias por impulsarme, apoyarme y alentarme a seguir adelante en todo momento a lo largo de la carrera y proyectos.

INDICE DE CONTENIDO GENERAL

TEMA	PAGINA
INDICE DE FIGURAS	6
RESUMEN	11
ABSTRACT	12

1. CAPITULO PRIMERO

1.1 Introducción	13
1.2 Justificación	14
1.3 Planteamiento del Problema	15
1.4 Hipótesis	16
1.5 Objetivos	16
1.5.1. Objetivo Principal	16
1.5.2 Objetivos Específicos	16
1.6 Alcances	17
1.7 Limitaciones	17

2. CAPITULO SEGUNDO

2.1 El Internet de las Cosas	18
2.1.1. ¿Cómo Funciona el Internet de las Cosas?	21
2.1.2. IoT Empresarial	21
2.1.3. IoT Industrial (IoT).....	22
2.1.4. Privacidad y Seguridad en la IoT	24
2.1.3. Microcontroladores IoT	25
2.2 Sistemas de Monitoreo	27
2.2.1. Módulos de Monitoreo	31
2.3 Laboratorios Remotos	34
2.3.1. Laboratorios remotos para Ingeniería Eléctrica, Electrónica y Mecatrónica	38

3. CAPITULO TERCERO

3.1 Desarrollo y Evaluación del Proyecto	39
3.1.1. Diagrama de Flujo de la Metodología para el Diseño de un Sistema de Monitoreo	39
3.1.2. Diagrama de Flujo del Funcionamiento del sistema	40
3.1.3. Diseño, Modelado, toma de Medidas y Adquisición del Material para desarrollar la estructura General del Módulo de Monitoreo	42
3.2 Armado y Ensamblado de la Estructura del Módulo de Monitoreo	45

3.2.1. Dispositivos y Equipos Empleados en la Elaboración del Proyecto	48
3.2.2. Selección y Adquisición de los Driver para el Control de los Motores a Pasos	60
3.2.3. Ensamblado y Conexión de los Drivers y los Motores a Pasos	62
3.3 Integración de un Microcontrolador Arduino UNO R4 Wifi	64
3.3.1. Diseño de un Código de Programación para el Control del Movimiento de los Motores a Pasos mediante un Microcontrolador Arduino UNO R4 Wifi	65
3.3.2. Pruebas del Código de Movilidad de los Motores a Pasos	68
3.3.3. Diseño de un Diagrama de Flujo de la programación para el Control de los Motores a Pasos	69
3.3.3. Diseño de un Código de programación para el Control de la Movilidad de los Motores a Pasos mediante un Dashboard y un Microcontrolador Arduino UNO R4 Wifi	76
3.4 Integración de un Microcontrolador Esp32 Cam.....	81
3.4.1. Diseño de un Código de programación para el Control del Microcontrolador Esp32 Cam	82
3.4.2. Diagrama de Flujo del Funcionamiento del Código para el Control del Microcontrolador Esp32 Cam	84
3.5 Diseño, Desarrollo y Fabricación de unas Placas de circuito Impreso o “PCB”.....	86
3.5.1. Diseño de un Código de programación para el Control General del Sistema de Monitoreo	90
3.5.2. Diagrama de Flujo del Funcionamiento General del Sistema de Monitoreo	92
3.5.3. Descripción del Proceso de Funcionamiento del sistema de Monitoreo.....	94
3.6 Análisis General de la Estructura del Sistema de Monitoreo	86
3.6.1. Determinación de las Problemáticas	107
3.6.1.1 Diseño de Soportes y Refuerzos de Impresiones en 3D.....	107
3.6.2. Impresión en 3D de los Soportes y Refuerzos diseñados	113
3.6.3. Inserción de las Piezas Impresas en 3D, a la Estructura General del sistema de Monitoreo	123

4. CAPITULO CUARTO

4.1 Resultados Finales	126
4.2 Discusión de los Resultados Finales	128
4.3 Conclusiones	134
4.4 Bibliografía	138
4.5 Anexos	140

INDICE DE FIGURAS

No. De Figuras	No. Paginas
Figura 1. Ejemplos de diversas Aplicaciones del IoT.	16
Figura 2. Esquematación del Concepto de IoT.	17
Figura 3. Número de Conexiones en IoT del 2010 al 2022.	18
Figura 4. Funcionamiento del Internet de las Cosas IoT.....	19
Figura 5. Internet de las Cosas Empresarial	20
Figura 6. Factores que Impulsan a las Industrias a Implementar Soluciones IoT.....	20
Figura 7. Internet de las Cosas IoT Industrial, controlando un Brazo Robótico.....	21
Figura 8. Internet de las Cosas IoT Industrial, controlando un Brazo Robótico.....	21
Figura 9. Seguridad empleada en el Internet de las Cosas “IoT”.....	22
Figura 10. Ilustración de los pines pertenecientes a los dispositivos GPIO del ESP32 y ESP8266.....	22
Figura 11. Ilustración de un Circuito para la Operación IoT de un ventilador.....	24
Figura 12. Ilustración de una Actualización Remota de Firmware y Script en Sistemas de “IoT”.....	24
Figura 13. Arquitectura de un Sistema de Monitoreo empleando el Internet de las Cosas “IoT”.	25
Figura 14. Diagrama del Funcionamiento de un Sistema de Monitoreo con “IoT”.....	26
Figura 15. Arquitectura de un Sistema de Monitoreo y la plataforma “IoT”.....	27
Figura 16. Arquitectura de un Sistema de Monitoreo y la plataforma “IoT”.....	28
Figura 17. Diagrama de Bloques del Módulo de procesamiento de señales Analógicas y Monitoreo	29
Figura 18. Diagrama de Bloques del Módulo de procesamiento del Sistema.	31
Figura 19. Esquema de un Laboratorio Virtual.....	33
Figura 20. Configuración típica de un Laboratorio Remoto.	34
Figura 21. Arquitectura de referencia para la Construcción de un Laboratorio Remoto.....	35
Figura 22. Arquitectura de referencia para la Construcción de un Laboratorio Remoto.....	36
Figura 23. Diagrama de Flujo de la Metodología para el diseño de un Sistema de Monitoreo	38
Figura 24. Diagrama de Flujo Principal del Proyecto.....	39
Figura 25. Tiras de Madera para la elaboración de la Estructura del Modulo de Monitoreo.....	40
Figura 26. Cortes en Tiras de Aluminio para la Estructura del Modelo de Monitoreo.	40
Figura 27. Perfil de Extrusión de modelo “6105-t5”.....	41
Figura 28. Toma de Medidas Correspondientes para la construcción del Sistema de Monitore	41
Figura 29. Laboratorio Remoto sobre el cual se construirá el Sistema de Monitoreo	41
Figura 30. 4 Tiras de aluminio de un Perfil de Extrusión de modelo “6105-t5”.....	42
Figura 31. 8 Tiras ya cortadas de aluminio de un Perfil de Extrusión de modelo “6105-t5”	42
Figura 32. Estructura ya Acoplada y Armada de la Estructura para el Módulo de Monitoreo	43

Figura 33. Estructura de Movilidad y Desplazamiento del Módulo de Monitoreo.....	43
Figura 34. Montaje de los rieles y Tornillo sin Fin para la movilidad del Módulo de Monitoreo.	44
Figura 35. Placas de Perfil de aluminio de 3mm para el ensamblaje del Módulo de Monitoreo.....	44
Figura 36. Placas cortadas a la medida de Perfil de aluminio de 3mm.....	45
Figura 37. Ensamble y acoplamiento de las placas cortadas de aluminio a la estructura del Modulo de Monitoreo.....	45
Figura 38. “Fuente de Alimentación de Mesa modelo 72-13600 TENMA”, usada en el proyecto.	48
Figura 39. Multímetro Digital con cables de prueba de Sonda.	48
Figura 40. Multímetro Digital con cables de prueba de Sonda.	49
Figura 41. Cables de tipo Caiman-Caiman.	50
Figura 42. Cables de tipo Banana-Caiman, conectados a la salida de la fuente y a un conductor.	50
Figura 43. Cable Rj45	51
Figura 44. Jumpers (puentes).....	52
Figura 45. Tarjeta electrónica digital Arduino Uno, modelo “Atmega328p”.....	52
Figura 46. Modulo Esp32 Cam.	53
Figura 47. Limit switch o sensor final de carrera.	54
Figura 48. Placa Protoboard.....	54
Figura 49. FTDI TTL USB Serial Converter FT23RL 7	55
Figura 50. Conector de Alimentacion/2300 “Jack Hembra”	55
Figura 51. Conector de Alimentacion/2300 “Jack Macho”	55
Figura 52. Divisor Extensor de Cable de Conector Hembra RJ45	56
Figura 53. Armazón y estructura del Módulo de Monitoreo con su respectivo riel	56
Figura 54. Motores a pasos o “passo a passo” Modelo Nenma-17.	57
Figura 55. Acoplamiento de los Motores Paso a Paso Nenma-17.....	58
Figura 56. Drivers adquiridos con Modelo TB6600.....	58
Figura 57. Especificación Técnica de los Drivers	59
Figura 58. Ensamble de los Drivers.	60
Figura 59. Cableado para la comunicación entre los Drivers y los Motores paso a paso.	60
Figura 60. Diagrama de Conexión entre los Drivers y los Motores paso a paso.	61
Figura 61. Diagrama de Conexión entre los Drivers y los Motores paso a paso 2.	61
Figura 62. Conexión entre el Arduino UNO R4 Wifi, los Drives y los Motores a pasos.....	62
Figura 63. Diagrama de Conexión entre el Arduino UNO R4 Wifi, los Drives y los Motores a pasos	62
Figura 64. Código de programación en Arduino para control de los Motores con Arduino R4.....	63
Figura 65. Código de programación en Arduino para control de los Motores con Arduino R4.....	63
Figura 66. Código de programación en Arduino para control de los Motores con Arduino R4.....	64
Figura 67. Código de programación en Arduino para control de los Motores con Arduino R4.....	64
Figura 68. Código de programación en Arduino para control de los Motores con Arduino R4.....	64
Figura 69. Código de programación en Arduino para control de los Motores con Arduino R4.....	65
Figura 70. Código de programación en Arduino para control de los Motores con Arduino R4.....	65

Figura 71. Código de programación en Arduino para control de los Motores con Arduino R4.....	65
Figura 72. Código de programación en Arduino para control de los Motores con Arduino R4.....	66
Figura 73. Pruebas del Código de programación para la Movilidad de los Motores.	66
Figura 74. Prueba del Código de programación para la Movilidad de los Motores.....	66
Figura 75. Diagrama de Flujo Principal de la programación para el Control de los Motores a Pasos	68
Figura 76. Prueba de Movilidad del Módulo de Monitoreo	69
Figura 77. Primer Problema detectado en la estructura del Módulo de Monitoreo.....	69
Figura 78. Segundo Problema detectado en la estructura del Módulo de Monitoreo.	70
Figura 79. Cambio del Resorte en el tornillo sin Fin	70
Figura 80. Micro Switch empleado en el proyecto.	71
Figura 81. Adquisición de 3 Micro Switch mas para su inserción en el Módulo de Monitoreo.	71
Figura 82. Cableado de los Micro Switch.	72
Figura 83. Empleo de 4 Push Button para el control de la Movilidad del Modulo.....	72
Figura 84. Dashboard generado en Arduino Cloud para el control del Modulo.	73
Figura 85. Prueba de Movilidad del Modulo de Monitoreo.	73
Figura 86. Código de programación para el control mediante un Dashboard y Arduino R4.	74
Figura 87. Código de programación para el control mediante un Dashboard y Arduino R4.	74
Figura 88. Código de programación para el control mediante un Dashboard y Arduino R4.	74
Figura 89. Código de programación para el control mediante un Dashboard y Arduino R4.	75
Figura 90. Código de programación para el control mediante un Dashboard y Arduino R4.	75
Figura 91. Código de programación para el control mediante un Dashboard y Arduino R4.	75
Figura 92. Código de programación para el control mediante un Dashboard y Arduino R4.	76
Figura 93. Código de programación para el control mediante un Dashboard y Arduino R4.	76
Figura 94. Código de programación para el control mediante un Dashboard y Arduino R4.	76
Figura 95. Código de programación para el control mediante un Dashboard y Arduino R4.	77
Figura 96. Código de programación para el control mediante un Dashboard y Arduino R4.	77
Figura 97. Código de programación para el control mediante un Dashboard y Arduino R4.	77
Figura 98. Código de programación para el control mediante un Dashboard y Arduino R4.	78
Figura 99. Dashboard Generado en Arduino Cloud para control de los Motores a pasos	78
Figura 100. Microcontrolador Esp32 Cam.....	79
Figura 101. Microcontrolador Arduino UNO R4 en conexión con Microcontrolador Esp32	79
Figura 102. Código de programación en para el control del Microcontrolador Esp32 Cam	80
Figura 103. Código de programación en para el control del Microcontrolador Esp32 Cam	80
Figura 104. Código de programación en para el control del Microcontrolador Esp32 Cam	80
Figura 105. Código de programación en para el control del Microcontrolador Esp32 Cam	81
Figura 106. Código de programación en para el control del Microcontrolador Esp32 Cam	81
Figura 107. Código de programación en para el control del Microcontrolador Esp32 Cam	81
Figura 108. Código de programación en para el control del Microcontrolador Esp32 Cam	82
Figura 109. Diagrama de Flujo del Código de programación para el Control del Microcontrolador Esp32 Cam	83
Figura 110. PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo	84
Figura 111. PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo	85
Figura 112. PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo	85

Figura 113. PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo	86
Figura 114. PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo	86
Figura 115. Diagrama de Conexión para el diseño de una Placa para el Arduino UNO R4 Wifi.	87
Figura 116. Diagrama de Conexión para el diseño de una Placa para el Esp32 Cam.	88
Figura 117. Código de programación para el Control general del Sistema de Monitoreo	89
Figura 118. Código de programación para el Control general del Sistema de Monitoreo	89
Figura 119. Diagrama de Flujo del Código de programación para el Control general del Sistema de Monitoreo	91
Figura 120. Fuente de Alimentación Modelo S-360-24.....	92
Figura 121. Arranque del Microcontrolador Arduino UNO R4 Wifi y el Esp32Cam.Cam	92
Figura 122. Cableado para la Comunicación del Microcontrolador Arduino UNO R4 Wifi y el Microcontrolador Esp32Cam, mediante un cable RJ45.....	93
Figura 123. Código puesto en Marcha del Microcontrolador Arduino UNO R4 Wifi.	93
Figura 124. Código puesto en Marcha del Microcontrolador Esp32Cam.	94
Figura 125. Microcontrolador Arduino UNO R4 Wifi.	94
Figura 126. Microcontrolador Arduino UNO R4 Wifi “Encendido”.....	95
Figura 127. Microcontrolador Arduino UNO R4 Wifi “Encendido, pero sin Recibir una Señal”.....	95
Figura 128. Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.	96
Figura 129. Microcontrolador Esp32 Cam “Encendido Recibiendo una Señal”.	96
Figura 130. Comunicación y conexión con la red y los Microcontroladores.	97
Figura 131. Comunicación y conexión con la red y los Microcontroladores.....	97
Figura 132. Imagen generada por el Dashboard para el control de los motores a pasos.....	98
Figura 133. Imagen generada por el Dashboard para la visualización del laboratorio.....	99
Figura 134. Imagen generada por el Dashboard para la visualización del laboratorio.....	99
Figura 135. Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.	100
Figura 136. Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.	100
Figura 137. Microcontrolador Arduino UNO R4 Wifi y los Drivers para el control de los Motores.....	101
Figura 138. Motores a pasos desplazando por los rieles.....	101
Figura 139. Movilización y desplazamiento del Microcontrolador Esp32 Cam.	102
Figura 140. Desplazamiento del Microcontrolador Esp32 Cam.....	102
Figura 141. Modulo de control para el Control del Modulo de Monitoreo	103
Figura 142. Sistema de Monitoreo Funcionando	103
Figura 143. Soporte-Grapa en forma de “U” para la Fijación del Cableado.....	104
Figura 144. Soporte-Cople en forma de Pirámide para interconectar el cable RJ45.	106
Figura 145. Soporte-Tope Final de Carrera.....	107
Figura 146. Soporte-Tope de Eje de Banda	108
Figura 147. Soporte-Final de Carrera en forma de “L” Soportes en los que se fijan los Micro Switch	109
Figura 148. Soporte Principal para la Inserción del Microcontrolador Esp32 Cam.	110
Figura 149. Tapa del Soporte Principal para la Inserción del Microcontrolador Esp32 Cam.....	110
Figura 150. Proceso de Impresión de las Piezas en 3D.	111
Figura 151. Proceso de Impresión de las Piezas en 3D.	112
Figura 152. Pinzas de punta empleadas para retirar los soportes de las piezas impresas en 3D.....	112

Figura 153. Soporte-Grapa en forma de “U” para la Fijación del Cableado, “Diseño”	113
Figura 154. Soporte-Grapa en forma de “U” para la Fijación del Cableado, “Impresión”	114
Figura 155. Soporte-Cople en forma de Pirámide para interconectar el cable RJ45, “Diseño”	114
Figura 156. Soporte-Cople en forma de Pirámide para interconectar el cable RJ45, “Impreso”	115
Figura 157. Soporte-Tope Final de Carrera, “Diseño”	115
Figura 158. Soporte-Tope de Eje de Banda, “Diseño”	116
Figura 159. Soporte-Tope de Eje de Banda, “Diseño”	116
Figura 160. Soporte-Tope de Eje de Banda, “Impreso” ..	117
Figura 161. Soporte-Final de Carrera en forma de “U”, “Diseño”	117
Figura 162. Soporte-Final de Carrera en forma de “U”, “Impresión” ..	118
Figura 163. Soporte Principal para la Inserción del Microcontrolador Esp32 Cam, “Diseño” ..	118
Figura 164. Tapa del Soporte Principal para la Inserción del Microcontrolador Esp32 Cam, “Diseño” ..	119
Figura 165. Tapa del Soporte Principal para la Inserción del Microcontrolador Esp32 Cam, “Diseño” ..	119
Figura 166. Tapa del Soporte Principal para la Inserción del Esp32 Cam, “Diseño Grabado”	120
Figura 167. Soporte-Grapa en forma de “U” para la Fijación del Cableado, “Acoplado”	121
Figura 168. Soporte-Cople en forma de Pirámide para interconectar el cable RJ45, “Acoplado” ..	121
Figura 169. Soporte-Tope Final de Carrera, “Acoplado” ..	122
Figura 170. Soporte-Final de Carrera en forma de “U”, “Acoplado”	122
Figura 171. Soporte-Tope de Eje de Banda, “Acoplado” ..	123
Figura 172. Soporte y Tapa Principal para la Inserción del Microcontrolador Esp32 Cam, “Acoplado” ..	123
Figura 173. Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal” ..	125
Figura 174. Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal” ..	128
Figura 175. Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal” ..	128
Figura 176. Tapa Principal para la Inserción del Microcontrolador Esp32 Cam ..	128
Figura 177. Soporte y Tapa Principal para la Inserción del Microcontrolador Esp32 Cam, “Acoplado” ..	129
Figura 178. Diagrama de Conexión entre el Arduino UNO R4 Wifi, los Drives y los Motores a pasos.	130
Figura 179. Laboratorio Remoto en función del Sistema de Monitoreo ..	130
Figura 180. Modulo de control para el Control del Modulo de Monitoreo ..	131
Figura 181. Dashboard Generado en Arduino Cloud para el control de los Motores Paso a Paso.	131

RESUMEN

En la última década, el uso de la tecnología en todos los sectores de la vida cotidiana ha tenido un fuerte impulso por parte de investigadores, sin embargo, el surgimiento del Internet de las Cosas (IoT) ha permitido ampliamente ser usado en la ingeniería como en el monitoreo ambiental, ciudades inteligentes, entre otras.

En este escrito se presenta el desarrollo de un Sistema de Monitoreo para un Laboratorio Remoto Mediante Dispositivos IoT, con la característica principal de tener un bajo costo y consumo energético, siendo visible dicha monitorización mediante una interfaz que permite a su vez tener un control general de todos aquellos procesos que se realizan, por lo cual florece como solución a esta necesidad de desarrollar un sistema de monitoreo constante para un tipo de Laboratorio Remoto, con la capacidad de, permitir a aquellas personas que se encuentren visualizando, el control de dicho monitoreo de manera remota mediante una interfaz a cualquiera que pretenda observar y analizar el entorno de dicho laboratorio siendo capaz de desplazar la cámara encargada de dicho monitoreo mediante una serie de rieles, permitiendo así un control total sobre su desplazamiento de forma arriba abajo, de izquierda a derecha en torno a toda la superficie a visualizar, teniendo entonces un control total de este proceso, de manera eficiente y precisa la visualización y el monitoreo en tiempo real y forma de dicho entorno.

Entre las ventajas de este tipo de desarrollos es que el sistema de desarrollo es de bajo costo y un bajo sistema energética, además con este tipo de proyectos se puede fortalecer la infraestructura de equipos de laboratorio en el área de ingeniería para poder ser manipulados de forma remota.

Palabras Clave: IoT, Internet de las Cosas, Sistema, Monitoreo, Laboratorio Remoto, Dispositivos IoT, Interfaz.

ABSTRACT

In the last decade, the use of technology in all sectors of daily life has had a strong push by researchers, however, the emergence of the Internet of Things (IoT) has allowed it to be widely used both in engineering as in environmental monitoring, artificial intelligence. cities, among others.

This work presents the development of a Monitoring System for a Remote Laboratory Using IoT Devices, with the main characteristic of having a low cost and energy consumption, said monitoring being visible through an interface that in turn allows general monitoring of all those processes that are carried out. carried out, so it arises as a solution to this need to develop a constant monitoring system for a kind of Remote Laboratory, with the ability to allow those people who are viewing, the control of said monitoring remotely through a Interface. to any person who intends to observe and analyze the environment of said laboratory, being able to move the camera in charge of said monitoring through a series of rails, thus allowing total control over its movement from top to bottom, from left to right throughout the entire space. around. the entire surface to be visualized, then having full control of this process, efficiently and accurately visualizing and monitoring said environment in real time.

Among the advantages of this type of development is that the server is free and the development system is low-cost. Additionally, with this type of project, the infrastructure of the laboratory equipment in the engineering area can be strengthened to be able to be manipulated in an efficient way. a remote way.

Keywords: IoT, Internet of Things, System, Monitoring, Remote Laboratory, IoT Devices, Interface.

CAPITULO PRIMERO

1.1 Introducción

Hoy en día la tecnología ha tenido un avance significativo entorno a la infinidad de posibilidades existentes de comunicación, y esto debido a la contingencia sanitaria vivida a partir del 27 de febrero del 2020 en México, este evento afecto considerablemente los entornos de aprendizaje grupal, siendo uno de los campos más afectados principal mente el del entorno educativo.

A causa de esto surgió la necesidad de un salto tecnológico evolutivo por parte de las Tecnologías de la Información y la Comunicación, permitiendo una gran variedad de formas de comunicación y distribución del conocimiento, así como el desarrollo de nuevas tecnologías para permitir el acceso e interacción a los entornos de conocimiento practico, tal es el caso de los laboratorios remotos, los cuales permitieron de manera eficiente el proceso de aprendizaje en dichos entornos, sin la necesidad de que el estudiante se encontrase de manera presencial en estos.

Sin embargo, actual mente estamos viviendo el posicionamiento de nuevas tecnologías que a través de un sin fin de dispositivos electrónicos, tecnologías como el Internet de las cosas, las cuales están revolucionando el mundo, siendo capaces de ofrecer modelos tecnológicos en todas las áreas de trabajo en la sociedad.

Este evento dio el paso agigantado para que tanto los dispositivos electrónicos como las estructuras de comunicación evolucionaran a tal grado de poder cambiar el entorno en el que vivíamos y comenzar una carrera de evolución de constantes cambios tecnológicos y digitales. Sin embargo a raíz de la gran demanda que esto conlleva, ha surgido como una necesidad indispensable el desarrollo de prototipos que permitan a su vez tener un monitoreo general de todos aquellos procesos que se realizan en aquellas plataformas de aprendizaje, por lo cual florece como solución a esta necesidad el implementar y desarrollar un sistema de monitoreo constante para este tipo de Laboratorios Remotos, con la capacidad de como su nombre lo indica, permitir a aquellas personas que se encuentren visualizando, el

control de dicho monitoreo de manera remota mediante una interfaz a cualquiera que pretenda observar y analizar el entorno de dicho laboratorio siendo capaz de desplazar la cámara encargada de dicho monitoreo mediante una serie de rieles, permitiendo así un control total sobre su desplazamiento de forma arriba abajo, de izquierda a derecha en torno a toda la superficie a visualizar, teniendo entonces un control total de este proceso, de manera eficiente y precisa la visualización y el monitoreo en tiempo real y forma de dicho entorno.

1.2 Justificación

Hoy en día los laboratorios Remotos son considerados como una herramienta tecnológica que permite a docentes y estudiantes realizar prácticas como si estos se encontraran en un entorno de laboratorio tradicional, pero a través de internet, permitiendo una mayor flexibilidad entorno a los horarios, el acceso a una gran número de datos remotos y un aprendizaje personalizado. Es decir, son una estrategia para la optimización de la eficiencia operativa, mejorando la seguridad y ofreciendo una mayor flexibilidad, siendo fundamentales para afrontar desafíos en los que la presencia a lugares cerrados llega a ser un problema cuando se trata de dar cavidad a un gran grupo de personas, tal fue el caso de la Pandemia. No obstante, a pesar de ser una solución a este tipo de problemáticas, es necesario e indispensable la monitorización de estos, el monitoreo en todo tipo de procesos es fundamental para determinar y detectar los problemas a tiempo, siendo este aspecto crucial para la toma de medidas de corrección antes de que los problemas pasen de una escala menor a una escala mayor, permitiendo un ahorro en los tiempos y costos realizando ajustes rápidos y precisos en el proceso de producción.

En el presente escrito, se propone el desarrollo de un Sistema de Monitoreo para un Laboratorio Remoto empleando dispositivos IoT, el cual sea capaz de permitir a aquellas personas que se encuentren visualizando, el control de dicho monitoreo de manera remota mediante una interfaz a cualquiera que pretenda observar y analizar el entorno de dicho laboratorio siendo capaz de desplazar la cámara encargada de dicho monitoreo mediante una serie de rieles, permitiendo así un control total sobre

su desplazamiento de forma arriba abajo, de izquierda a derecha en torno a toda la superficie a visualizar, teniendo entonces un control total de este proceso, de manera eficiente y precisa la visualización y el monitoreo en tiempo real y forma de dicho entorno.

1.3 Planteamiento del Problema

Hoy en día una de las necesidades más prioritarias en cualquier entorno de producción industrial y en cualquier laboratorio de investigación y desarrollo es la de tener un sistema de monitoreo que le permita evaluar la calidad de los procesos y resultados, identificar los riesgos permitiendo implementar medidas de seguridad, ahorrar recursos al identificar ineficiencias, siendo este aspecto crucial para la toma de medidas de corrección antes de que los problemas pasen de una escala menor a una escala mayor, permitiendo un ahorro en los tiempos y costos realizando ajustes rápidos y precisos en el proceso de producción, así mismo permite el ahorro de energía, esto mediante sensores con los cuales se puede controlar la ocupación de espacios en donde el encendido de las luminarias se hace al detectar la presencia de movimiento así mismo es su apagado contribuyendo a la reducción del consumo de energía, final mente influye en el control de calidad, permitiendo la detección de problemas y la garantía de una alta calidad en los productos finales. Por ello que se deduce que un sistema de monitoreo en un Laboratorio es fundamental para garantizar la calidad, la seguridad, la eficiencia y el cumplimiento normativo.

1.4 hipótesis

Este proyecto propone el desarrollo de un Sistema de Monitoreo para un Laboratorio Remoto empleando dispositivos IoT, que tenga la capacidad de permitir el control de dicho monitoreo de manera remota mediante una interfaz para cualquiera que pretenda observar y analizar el entorno de dicho laboratorio siendo capaz de desplazar la cámara encargada de dicho monitoreo mediante una serie de rieles, permitiendo así un control total sobre su desplazamiento de forma arriba abajo, de izquierda a derecha en torno a toda la superficie a visualizar, teniendo entonces un control total de este proceso, de manera eficiente y precisa la visualización y el monitoreo en tiempo real y forma de dicho entorno.

1.5 Objetivos

1.5.1 Objetivo General

Diseño e implementación de un Sistema de Monitoreo para un Laboratorio Remoto empleando dispositivos IoT, que permita supervisar y controlar mediante una plataforma web, el control del monitoreo en tiempo real.

1.5.2 Objetivos Específicos

- Diseñar y caracterizar un Sistema de Monitoreo mediante dispositivos IoT.
- Diseñar un programa lógico que permita el monitoreo constante de un Laboratorio Remoto, así como el control de su movilidad mediante rieles.
- Diseñar un programa mediante la implementación de dispositivos IoT que permitan el acceso al laboratorio y permita programar los microcontroladores sin necesidad de estar de forma presencial.

- Crear en una plataforma web con una interfaz que permita la supervisión y control del monitoreo al usuario mediante la tecnología IoT.
- Se permita que el usuario pueda leer las especificaciones y las Fichas técnicas

1.6 Alcances

Con la elaboración de este proyecto se pretende su implementación en diversas áreas de una industria, así como su aplicación en diversos laboratorios de investigación y desarrollo de las universidades, no obstante cabe mencionar que este proyecto se encuentra enfocado estrictamente a un área docente-educativa, ya que su empleamiento permite al usuario apreciar de manera clara aquellos procesos que se encuentren llevando a cabo a la hora de la monitorización con la finalidad de apreciar cada proceso en tiempo real, sin embargo, se espera que este prototipo “Sistema de Monitoreo” se pueda implementar a un nivel Industrial. De tal forma que, al emplear este sistema en una planta de operación de producción con los sensores adecuados, se pueda monitorear de manera tanto preventiva como correctiva, con el objetivo de detectar errores en equipos, dispositivos y o en procesos de producción, ayudando así a prevenir fallos en estos.

1.7 Limitaciones

Este proyecto se ha desarrollado partiendo de la necesidad de monitorear los procesos que se realicen entorno aun laboratorio remoto, por tal motivo es necesario el diseñar y construir un un Laboratorio Remoto empleando dispositivos IoT, el cual sea capaz de permitir a aquellas personas que se encuentren visualizando, el control de dicho monitoreo de manera remota mediante una interfaz a cualquiera que pretenda observar y analizar el entorno de dicho laboratorio siendo capaz de desplazar la cámara encargada de dicho monitoreo mediante una serie de rieles, permitiendo así un control total sobre su desplazamiento de forma arriba abajo, de

izquierda a derecha en torno a toda la superficie a visualizar, teniendo entonces un control total de este proceso, de manera eficiente y precisa la visualización y el monitoreo en tiempo real del entorno.

CAPITULO SEGUNDO

2.1 El Internet de las Cosas “IoT”

El Internet de las Cosas también conocido por sus siglas como “IoT”, se refiere a una red información y comunicación, integrada por dispositivos físicos, integrados con sensores, software y una conectividad permanentemente a internet que les permiten recopilar, capturar y compartir un gran número de datos.



Figura 1: Ejemplos de diversas Aplicaciones del IoT.
Fuente: Salazar, J., & Silvestre, S. (2016).

Todo esto con el propósito de automatizar actividades y procesos diarios en nuestra vida cotidiana, así como la de analizar todos los datos generados aportándonos información útil que nos ayude a la correcta toma de decisión frente las situaciones que se nos presentan diariamente (Salazar, J., & Silvestre, S. 2016).

En otras palabras, el Internet de las Cosas se define como un proceso que permite la conexión de una serie de elementos físicos cotidianos al Internet, estos pueden ser desde objetos domésticos como bombillas de luz hasta dispositivos médicos empleados en hospitales, prendas y accesorios personales inteligentes, e incluso sistemas de ciudades inteligentes, sin embargo estas debe ser capaces de modificar los servicios para que estos se adapten, sin intervención directa a las posibilidades de las terminales y de las redes sobre las que se transmiten. Sin embargo, los dispositivos IoT se dividen general mente en dos categorías, “Interruptores y Sensores”, los primeros se encargan de enviar instrucciones a un objeto, mientras que los Sensores son los encargados de recopilar los datos y enviarlos.

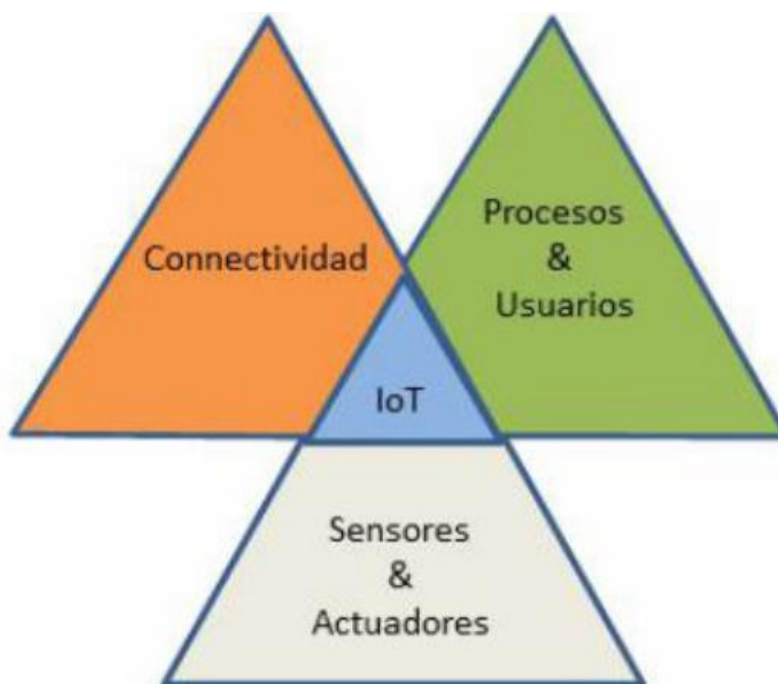


Figura 2: Esquematización del Concepto de IoT.
Fuente: Salazar, J., & Silvestre, S. (2016).

Cabe mencionar que una de las características claves de estas IoT es que esta no solo sirve para conectar objetos entre ellos y manejarlos desde un dispositivo remoto, sino que la información precisa, automatizada y en tiempo real, por lo que el funcionamiento del Internet de las Cosas implica la transferencia de un gran número de datos entre distintos dispositivos físicos mediante una serie de redes inalámbricas, con mínima intervención humana. Un ejemplo claro puede ser el de

un termostato inteligente, el cual puede recibir datos desde una ubicación u otra sin la necesidad de estar en la misma ubicación de su instalación, por lo que se cumple el que los sistemas IoT envían, reciben y analizan datos en un ciclo de retroalimentación.

Gracias a este tipo de tecnologías todos los datos enviados o recibidos pueden ser analizados, de manera inmediata por individuos o sistemas que emplean la inteligencia artificial, es decir que el IoT a revolucionado todos aquellos procesos y formas en las que el ser humano interactúa con su entorno, con el objetivo de permitir una interacción conectada, eficiente y basada en datos, haciendo que cada proceso de la vida sea más fácil.

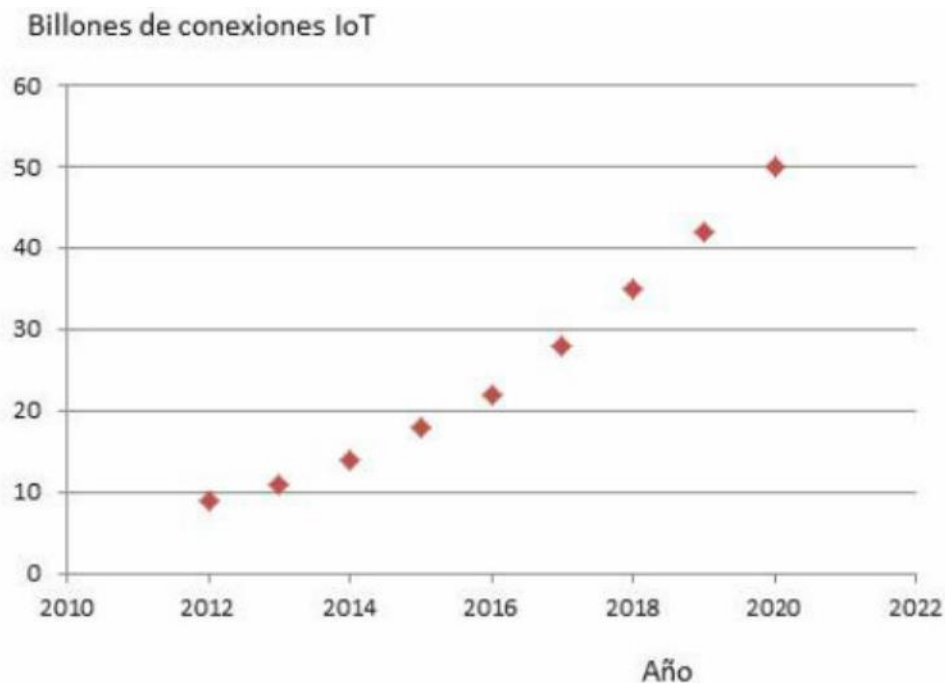


Figura 3: Número de Conexiones en IoT del 2010 al 2022.
Fuente: Salazar, J., & Silvestre, S. (2016).

Cabe mencionar que en 2010, el número de objetos físicos cotidianos y los dispositivos conectados a Internet fue de alrededor de 12,5 mil millones. Se espera que este número se duplique hasta 25 mil millones en el año 2015 con lo que el número de dispositivos inteligentes por persona aumenta, y otros 50 millones en 2020.

2.1.1 ¿Cómo funciona el Internet de las Cosas?

El funcionamiento de los sistemas de Internet de las Cosas o también conocidos como “IoT” tradicionales consiste en enviar, recibir y analizar los datos de forma permanente en un ciclo de retroalimentación, por lo que, según su tipo de tecnología aplicada, las personas o los sistemas de inteligencia artificial y aprendizaje automático (IA/ML) pueden analizar estos datos casi de inmediato o durante cierto tiempo (Muñoz, O. Q. 2019).

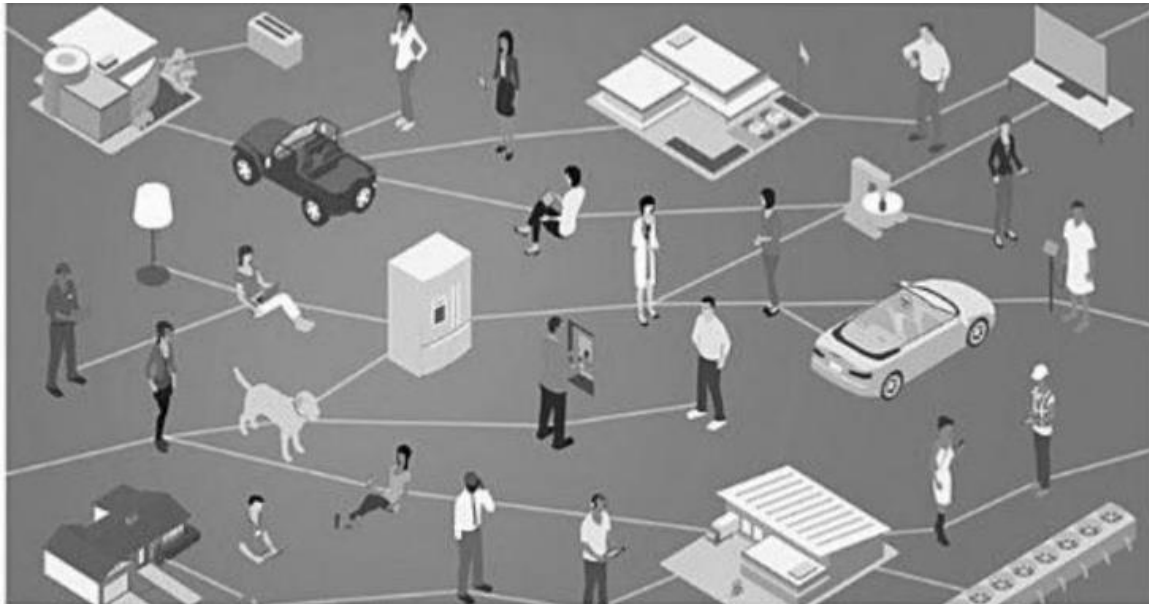


Figura 4: Funcionamiento del Internet de las Cosas IoT.
Fuente: Muñoz, O. Q. (2019).

Un ejemplo muy común es por ejemplo el del control de un Termostato a distancia con el objetivo de poder manipularlo de manera automática o a voluntad.

2.1.2 IoT Empresarial

El Internet de las cosas capta la atención de los consumidores, cuyas experiencias con las tecnologías, como los relojes inteligentes, los cuales se ven afectadas por las preocupaciones en torno a la seguridad y la privacidad que supone el hecho de estar conectados de forma permanente.

Este tipo de perspectiva se aplica a todos los tipos de proyectos de IoT empresarial, especialmente cuando el usuario final es el público en general. Esto les permite a dichas empresas mejorar los modelos comerciales actuales y entablar nuevas

relaciones con los clientes, sin embargo, el volumen de datos que genera un sistema de dispositivos inteligentes, mejor conocido como “big data” puede ser abrumador y complicado debido a su integración en los sistemas actuales y la configuración del análisis de datos.

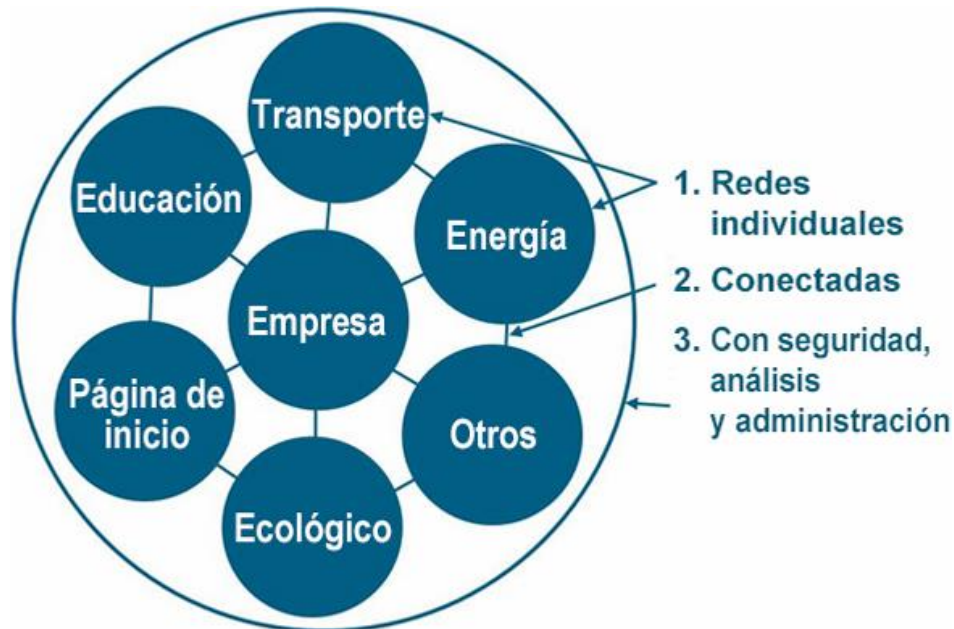


Figura 5: Internet de las Cosas Empresarial.
Fuente: Cisco IBSG, abril de 2011, Evans, D. (2011).

2.1.3 IoT industrial (IIoT)

El Internet de las cosas industrial (IIoT) hace referencia a la aplicación de la tecnología IoT en todos aquellos entornos industriales o pertenecientes a una industria, centrándose particularmente en la instrumentación de sensores y dispositivos de control que utilizan tecnologías en la nube (Valencia, A., & Portilla, P. 2019).



Figura 6: Factores que Impulsan a las Industrias a Implementar Soluciones IIoT.
Fuente: Valencia, A., & Portilla, P. (2019).

Este “IIoT” se considera abiertamente como una intersección entre la tecnología y la industria, donde las ideas impulsadas por datos mejoran la eficiencia, la seguridad y la productividad en diversos sectores, como la monitorización de maquinaria en fábricas y la gestión de manera inteligente de la cadena de suministro, transformado de manera radical la forma en la que operan las empresas y las industrias en el mundo.



Figura 7: Internet de las Cosas IIoT Industrial, controlando un Brazo Robótico.
Fuente: Cisco IBSG, abril de 2011, Evans, D. (2011).

Para la solución de problemáticas en máquinas y o dispositivos en una industria se emplean este tipo de sistemas IIoT, agregando sensores especializados a las partes de la maquinaria que son más propensas a sufrir daños o a utilizarse de forma excesiva, siendo empleados con la finalidad de realizar un mantenimiento predictivo, mejorando el desempeño del personal (siendo esto un ejemplo de la recopilación y el análisis inmediatos de los datos). Por este tipo de casos se considera que el IIoT industrial (IIoT) abarca todo tipo de casos prácticos en las actividades de las fábricas, las empresas de energía y otros sectores, teniendo un papel importante en llevar la automatización a las máquinas industriales.

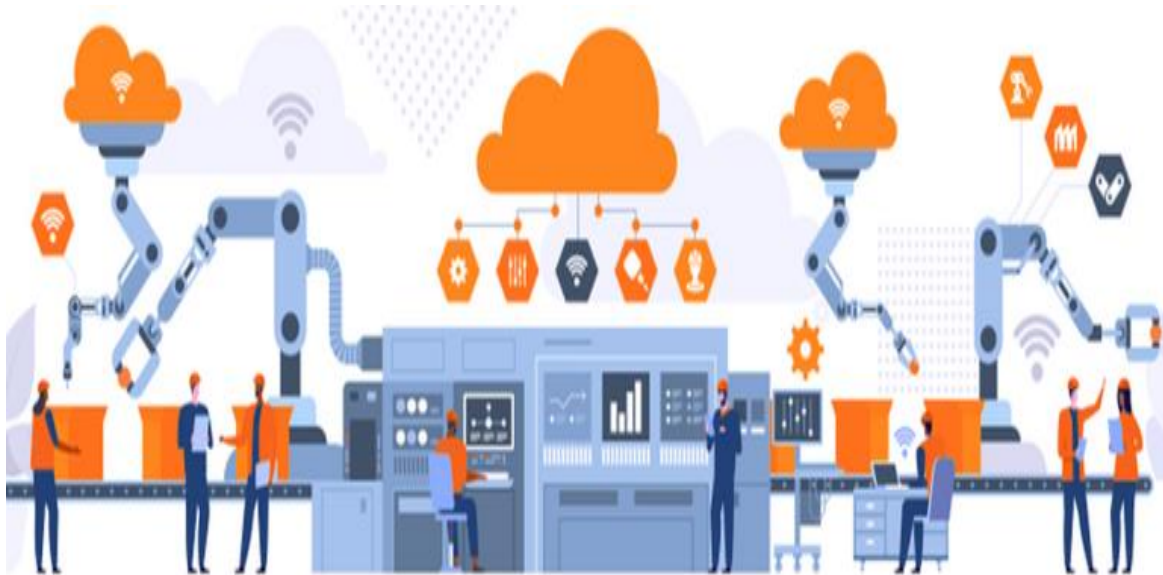


Figura 8: Internet de las Cosas IIoT Industrial, controlando un Brazo Robótico.
 Fuente: Valencia, A., & Portilla, P. (2019).

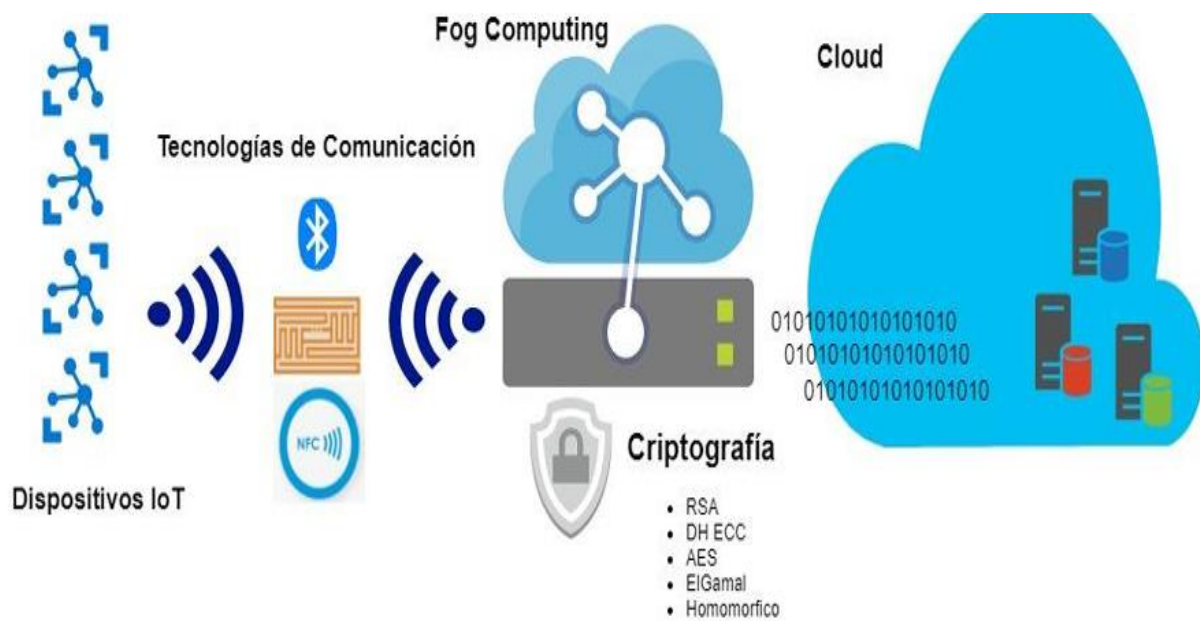


Figura 9: Seguridad empleada en el Internet de las Cosas "IoT".
 Fuente: Montoya Ponce, R. J. (2019).

2.1.4 Microcontroladores IOT

Hoy en día el empleo de los microcontroladores, es fundamental para llevar a cabo funciones de control, monitoreo y automatización.

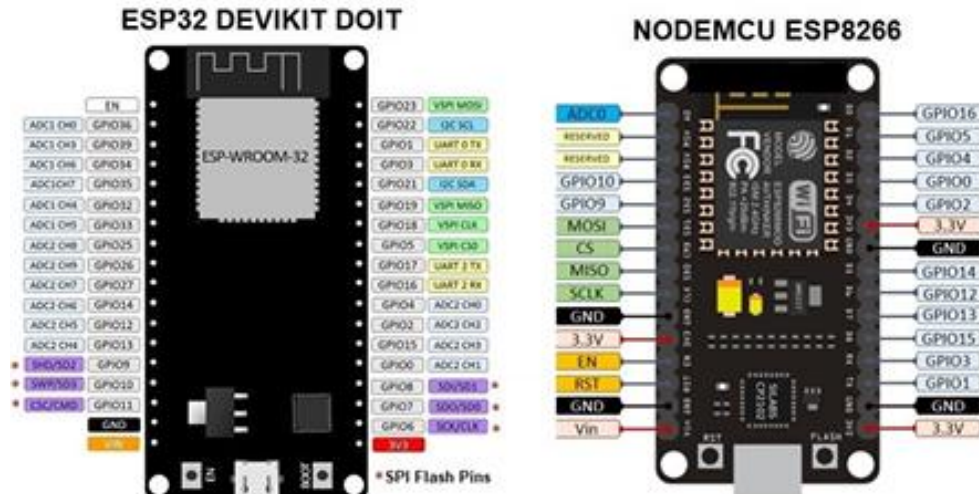


Figura 10: Ilustración de los pines pertenecientes a los dispositivos GPIO del ESP32 y ESP8266.
Fuente: Pereira, R., De Souza, C., Patino, D., & Lata, J., (2022).

Sin embargo, el avance en torno a las nuevas tecnologías que se desarrollan a pasos agigantados día con día, apunta al desarrollo, formación y evolución de aquello que se conoce como Internet de las Cosas (IoT), buscando así la comunicación entre los equipos de las industrias, de los hogares y de los usuarios de modo automático, es por ello que estos dispositivos nos permiten una mejor comunicación con las plataformas de IoT.

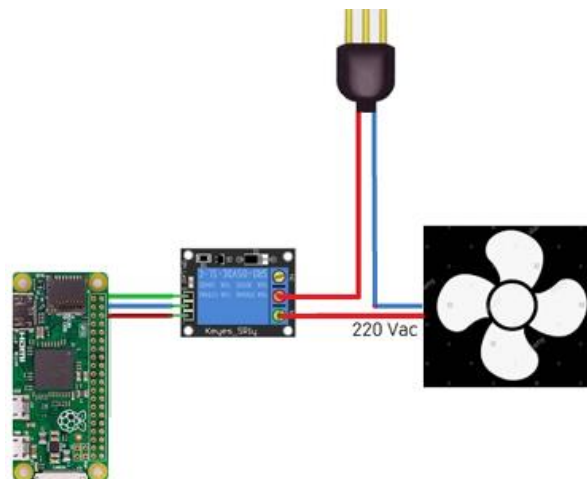


Figura 11: Ilustración de un Circuito para la Operación IoT de un ventilador.
Fuente: Pereira, R., De Souza, C., Patino, D., & Lata, J., (2022).

Cabe mencionar que estos dispositivos destacan debido a su gran eficiencia, caracterizándose por no ser un simple microcontrolador sino que estos son microcontroladores que además de brindar un entorno programable para llevar a cabo sistemas de control, mediciones, estructuras de programaciones lógicas, brindando una nueva conexión dada entre el internet y el dispositivo, creando así un sistema en donde los datos recogidos por el microcontrolador son llevados a través de servidores web, plataformas o páginas web un usuario en concreto, es decir tienen construido internamente la funcionalidad de conexión entre la nube web y el dispositivo, funcionando por medio de protocolos de conexión IPV4/ IPV6 (Pereira, R., De Souza, C., Patino, D., & Lata, J., 2022).

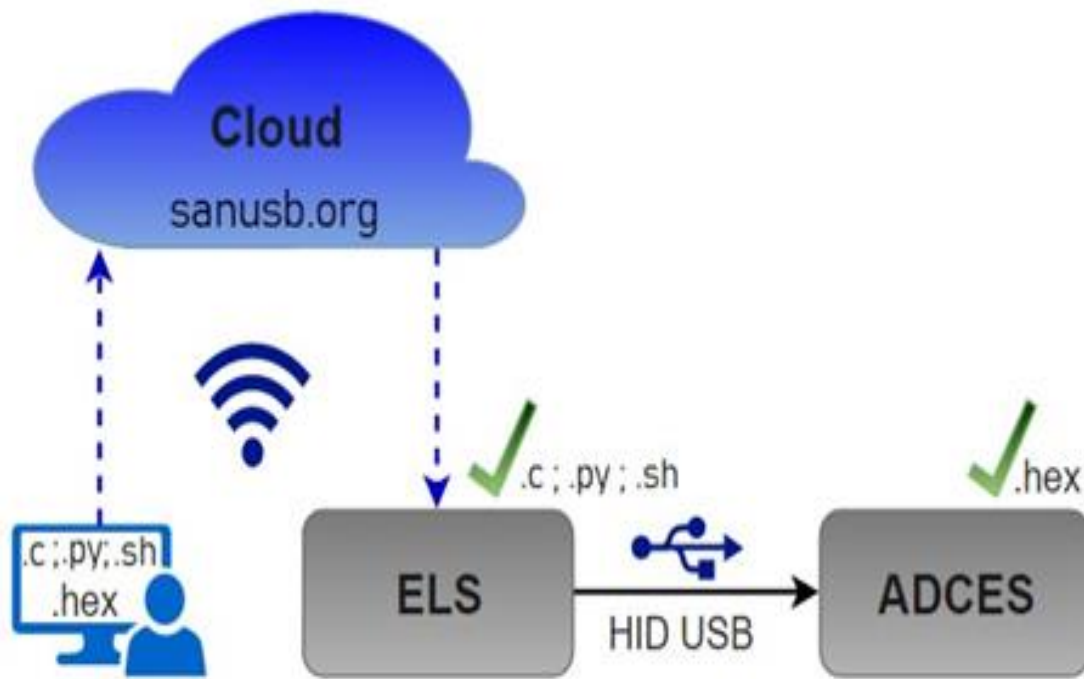


Figura 12: Ilustración de una Actualización Remota de Firmware y Script en Sistemas de "IoT"
Fuente: Pereira, R., De Souza, C., Patino, D., & Lata, J., (2022).

2.2 Sistemas de Monitoreo

Un sistema de Monitoreo es una infraestructura compuesta por diferentes dispositivos, sensores y software diseñados para la recopilación de datos mediante la implementación de un conjunto de procesos y procedimientos en tiempo y forma real, de diversas fuentes y analizarlos de forma sistemática, permitiendo hacer un seguimiento de su progreso, por lo que su objetivo es el de aplicar todos los ajustes necesarios para la mejora de la eficiencia y la eficacia.

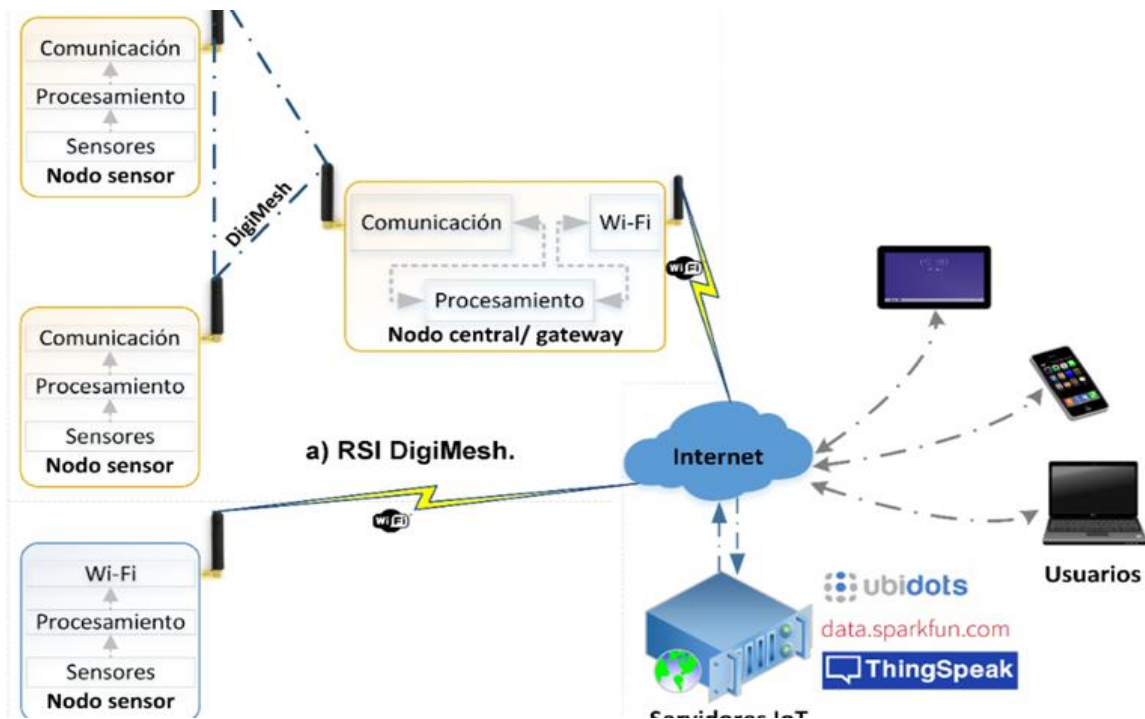


Figura 13: Arquitectura de un Sistema de Monitoreo empleando el Internet de las Cosas "IoT".
Fuente: Escobar Gallardo, E., & Villazón, A. (2018).

Este tipo de sistemas son esenciales para garantizar la eficiencia en el funcionamiento de la infraestructura tecnológica, por lo que poseen una serie de ventajas, de entre las cuales destacan, el Acceso al estado de los sistemas informáticos en tiempo real, la detección del origen de los incidentes, la información ejecutiva sobre el estado de las instalaciones y activos informáticos críticos, la mejora en las labores de mantenimiento del sistema, una mejor configuración de eventos y alarmas.

Estos sistemas compuestos por dispositivos, sensores y software empleados para la recopilación de datos en tiempo real de diversas fuentes y analizarlos de forma sistemática, desempeñan un papel crucial en diversos contextos, en el contexto de un Monitoreo de Tecnologías de la Información o IT la implementación de este sistema de monitoreo es indispensable para garantizar un funcionamiento correcto y la detección de problemas de forma preventiva y predictiva, por lo que esto permite mantener desde la disponibilidad hasta el rendimiento y la seguridad de los sistemas, donde los componentes que son monitoreados pueden incluir servidores, redes, aplicaciones etc.

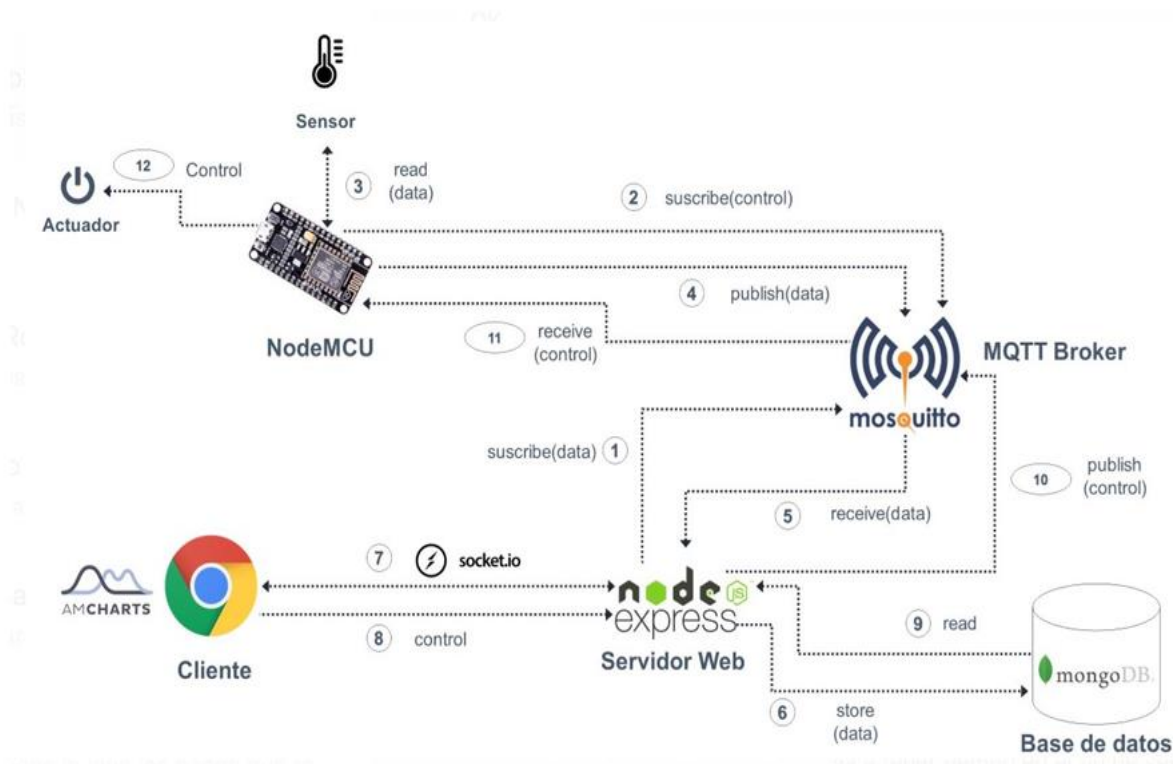


Figura 14: Diagrama del Funcionamiento de un Sistema de Monitoreo con "IoT".
Fuente: Quiñones-Cuenca, M., González-Jaramillo, V., Torres, R., & Jumbo, M. (2017).

En el contexto de un Monitoreo y Evaluación de Políticas Públicas los sistemas de monitoreo y evaluación son herramientas integradas que permiten hacer seguimiento de todos aquellos resultados en la implementación de políticas públicas e iniciativas gubernamentales a lo largo del tiempo, ayudando a medir el impacto de las acciones gubernamentales y a ajustar estrategias según los resultados obtenidos.

Final mente desde el contexto de un Monitoreo General se implica que este debe de ser consciente del estado en el que se encuentre un sistema siendo capaz de observar todos los cambios que puedan producirse con el tiempo, siendo posible de realizarse mediante una forma manual o de una manera automatizada, esto mediante los dispositivos de instrumentación y de la transmisión de una cierta cantidad de datos, es por estos contextos de monitoreo que se define entonces que los sistemas de monitoreo son esenciales para poder mantener el funcionamiento eficiente de todos aquellos sistemas tecnológicos, así como evaluar políticas y a su vez poder garantizar la seguridad y el rendimiento en diversas áreas. Cabe mencionar que la monitorización de sistemas es el proceso de supervisar y controlar constantemente el rendimiento y el estado de un sistema informático o una red de sistemas para detectar posibles problemas, errores u anomalías, por lo tanto para poder realizar una buena monitorización de un sistema, se debe seleccionar una herramienta de monitorización, así como también se debe de establecer todos y cada uno de los protocolos de resolución de incidencias para la resolución de problemas, para final mente monitorizar de forma continua con el objetivo de aprovechar lo recursos existentes, así como también prevenir los problemas y reducir los costos, por lo tanto se concluye que la monitorización es clave para mantener la infraestructura en las mejores y óptimas condiciones.

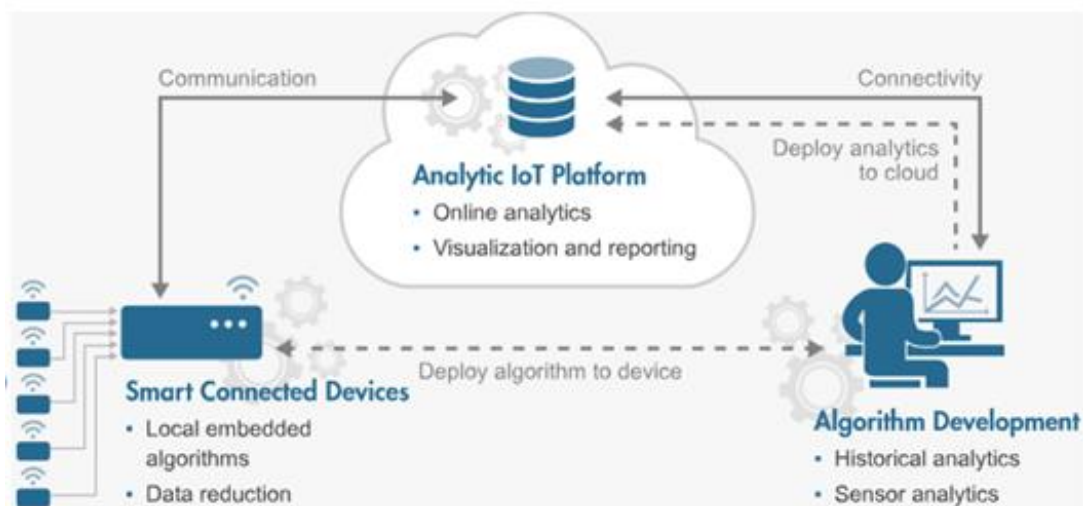


Figura 15: Arquitectura de un Sistema de Monitoreo y la plataforma "IoT".
Fuente: Quiñones-Cuenca, M., González-Jaramillo, V., Torres, R., & Jumbo, M. (2017).

Hoy en día la monitorización es esencial para el funcionamiento eficiente de las infraestructuras tecnológicas, sin embargo dentro de las ventajas que se obtiene al implementar un sistema de monitoreo se encuentran el tener acceso en tiempo real accediendo al estado del sistema a monitoreo en tiempo real, así como la detección de incidentes teniendo la facultad para poder detectar estos y permitir tomar medidas preventivas para su incurrimento, la eficiencia en el mantenimiento mejorando la eficiencia y la eficacia de las labores de mantenimiento del sistema, el inventariar sistemas creando mapas o listados de los sistemas para la planificación y el crecimiento en base al uso real y final mente la reducción de costos identificando oportunidades para reducir costos y omitir recursos.

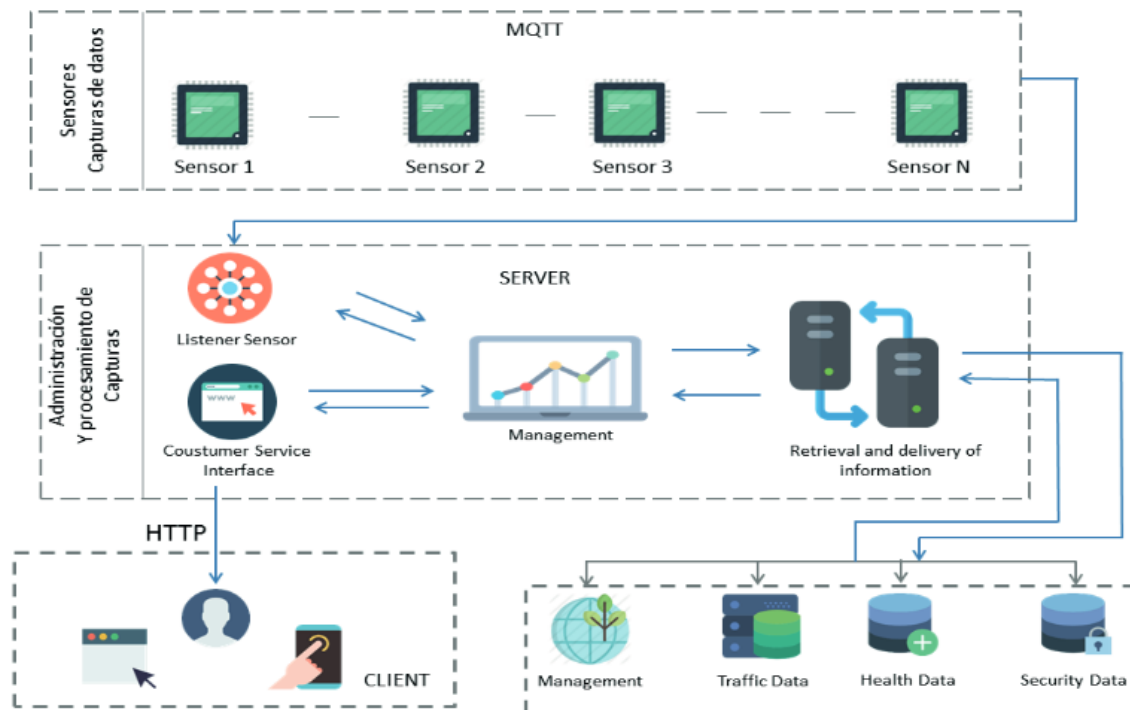


Figura 16: Arquitectura de un Sistema de Monitoreo y la plataforma "IoT".
 Fuente: Gómez, J. E., Castaño, S., Mercado, T., Fernandez, A., & Garcia, J. (2017).

No obstante, para poder llevar a cabo una buena monitorización de sistemas se debe de seleccionar una buena herramienta de monitorización la cual debe de ser adecuada a las necesidades de quien la requiera, así como establecer protocolos de adecuación con la finalidad de decir cómo se resolverán las incidencias detectadas, para final mente Monitorizar de forma constante supervisando los sistemas de manera regular para mantener su rendimiento Optimo. Sin embargo,

es indispensable el tener en cuenta que la monitorización es esencial para garantizar el buen funcionamiento de tus infraestructuras tecnológicas y prevenir problemas antes de que se agraven.

2.2.1 Módulos de Monitoreo

En el mundo de hoy en día la tecnología desempeña un papel fundamental en la vida cotidiana de las persona, es indispensable, contar y emplear un sistema de monitoreo eficiente, se ha vuelto indispensable para con su empleamiento garantizar la seguridad y el correcto funcionamiento de diversos entornos, no obstante los módulos de monitoreo son aquellos componentes esenciales en un sistema de seguridad y detección, este tipo de dispositivos permiten la configuración y verificación de diferentes tipos de cableado en distintos sistemas, asegurando un monitoreo efectivo de los elementos conectados al sistema. Estos desempeñan un papel crucial en la seguridad, la eficiencia, la tranquilidad y la supervisión de sistemas y entornos.

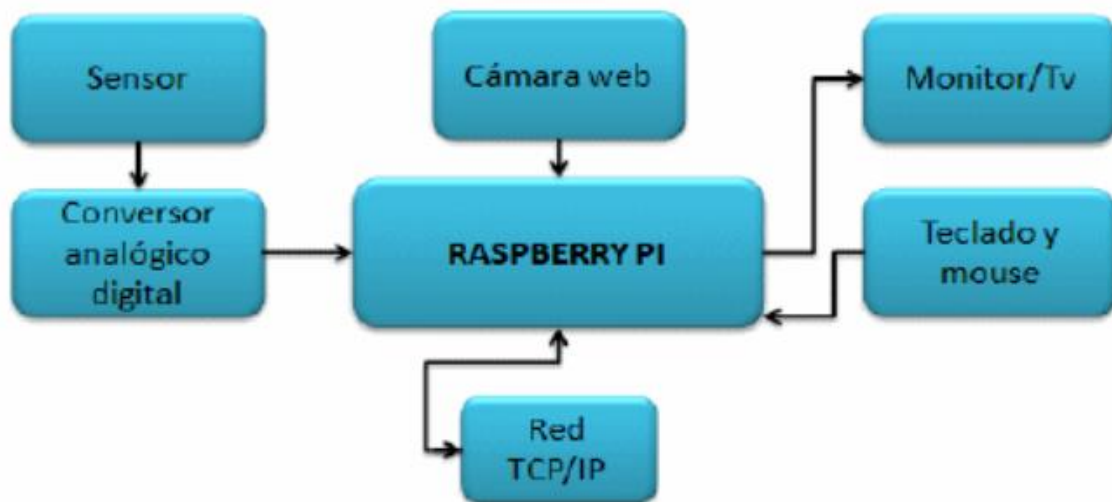


Figura 17: Diagrama de Bloques del Módulo de procesamiento de señales Analógicas y Monitoreo.
Fuente: Velásquez, S., & Gómez, C. (2013).

Sin embargo, los componentes esenciales de un sistema de monitoreo son sensores, la unidad de adquisición de datos, la transmisión de datos, la comunicación de alertas, la integración con otros sistemas, la usabilidad y presentación de los datos y la flexibilidad y adaptabilidad.

Los sensores son los dispositivos que capturan los datos del entorno y a su vez los convierte en señales eléctricas o digitales, pudiendo medir una amplia gama de variables, tales como la temperatura, la presión, la humedad, el movimiento, etc. Es por ello que la selección de estos adecuadamente depende indiscutiblemente del tipo de monitoreo, requerido y de las variables a medir. La unidad de adquisición de datos se encarga de la recopilación de los datos proveniente de los sensores y a su vez estos los convierte en un formato digital procesable y analizable, de tal forma que pueden presentarse como un dispositivo físico o una aplicación de software, según las necesidades del sistema de monitoreo. Sin embargo, una vez adquiridos los datos, es necesario su transmisión hacia un sistema central o hacia un dispositivo de visualización, esta transmisión puede ser realizada mediante un sistema de cableado, una serie de redes inalámbricas o también mediante una red de internet, en este proceso la seguridad y la confiabilidad de dicha transmisión son un punto fundamental para evitar pérdidas de la información. No obstante, para la comunicación de alertas, el sistema de monitoreo debería de ser capaz de notificar automáticamente sobre eventos críticos por lo que las alertas pueden enviarse por diversos medios de comunicación, los cuales pueden ser correo electrónico, mensajes de texto u notificaciones en una interfaz.

Para la integración con otros sistemas, un módulo de monitoreo debe de ser compatible con otros sistemas y diversas herramientas, pudiendo a su vez integrarse con una serie de sistemas de gestión, de bases de datos y servidores en la nube, es por ello que para la usabilidad y la presentación de los datos, la interfaz de usuarios debe de ser intuitiva y fácil de usar, debido a que los datos deben de presentarse de manera clara y de forma comprensible para todos los usuarios, para finalmente verse entorno a la mayor flexibilidad y adaptabilidad que se pueda obtener, donde se pretende que el sistema deba de poder adaptarse a diferentes contextos y diversos requerimientos específicos, con la caracterización principal de ser escalable y permitir la incorporación de nuevos sensores o cambios en la configuración. A raíz de todo lo anterior se concibe que un módulo de monitoreo eficiente debe de combinar estos componentes para así poder proporcionar

información valiosa en tiempo real y así poder prevenir los problemas antes de que estos se conviertan en situaciones críticas, Sin embargo su implementación ofrece un conjunto de ventajas que en conjunto, le permiten acceder al estado de los sistemas informáticos en tiempo real, facilitar la detección de una forma anticipada o temprana de una extensa variedad de problemas y la toma de decisiones de forma clara y concisa, la identificación de el origen de los incidentes y las fallas presentadas, la facilitación de la resolución rápida y eficiente de dichos problemas, asi como proporcionar información sobre el estado de las instalación y los activos informáticos críticos, la facilitación de la toma de decisiones estratégicas, la mejora en torno a la eficacia y la eficiencia de las labores de mantenimiento además de permitir la planificación del crecimiento y la expansión basados en datos reales presentes en tiempo real, por lo que con la reducción u prevención de los problemas a demás de la optimización de los recursos se logra un ahorro económico extremadamente significativo.

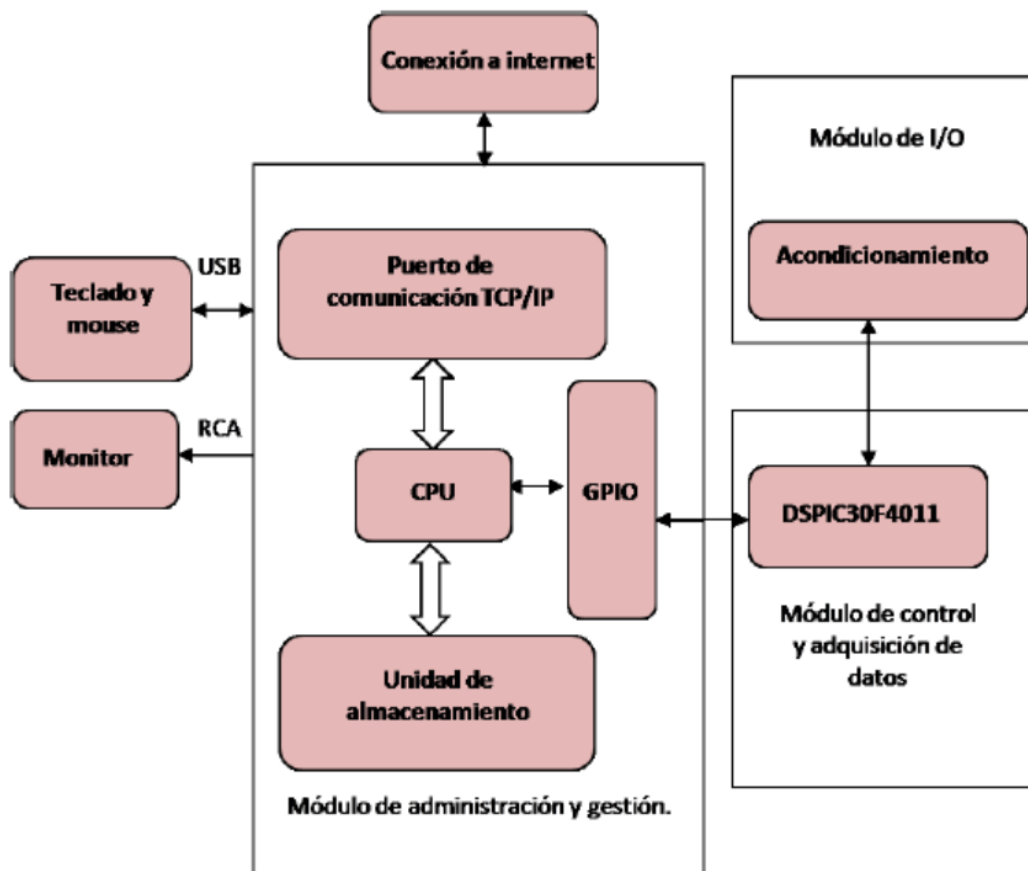


Figura 18: Diagrama de Bloques del Módulo de procesamiento del Sistema.
Fuente: Velásquez, S., & Gómez, C. (2013).

No obstante no todo es una ventaja en su implementación, debido a que se presentan con una complejidad inicial bastante alta ya que al configurar y mantener un sistema de monitoreo requiere un muy alto grado de conocimientos técnicos, así como una inversión inicial y un conjunto de recursos muy amplio, además de presentar una posible sobrecarga de información ya que si no se filtran adecuadamente las alertas puede presentarse una sobre carga de las notificaciones que el sistema envía para su resolución dificultando la identificación de problemas críticos, así como también un alto costo de implementación ya que la adquisición de las herramientas de monitoreo llegan a ser muy costosas, por lo que debe de ser considerado y ajustado por los beneficios presentados a largo plazo, por ultimo presentan una posible falta de adaptabilidad ya que algunos sistemas de monitoreo pueden no adaptarse fácilmente a cambios en torno a su infraestructura o su tecnología, por lo que es fundamental elegir una solución flexible y escalable.

2.3 Laboratorios Remotos

Una de las herramientas tecnológicas más eficientes hoy en día que combina tanto software como hardware, son los Laboratorios Remotos , la cual es una herramienta diseñada y desarrollada a raíz de la contingencia sanitaria del COVID 19 en el año 2020, orientada a raíz de este acontecimiento a docentes y estudiantes para poder llevar a cabo practicas de laboratorio a través de internet, como si estuvieran en un laboratorio de forma presencial, este tipo de herramientas tecnológicas existen físicamente en algún lugar y permiten a los usuarios controlar experimentos a través de internet ofreciendo una ventana para interactuar con recursos físicos reales que pueden estar a miles de kilómetros de distancia, fungiendo así como una instalación física equipada con equipos y dispositivos electrónicos los cuales pueden ser accedidos, controlados y manipulados de forma remota a través de internet, permitiendo a los estudiantes realizar experimentos y prácticas en tiempo real, sin la necesidad de estar presentes físicamente en el laboratorio, brindando una experiencia de aprendizaje similar a la de un laboratorio físico, pero de forma virtual y accesible desde cualquier lugar con conexión a Internet.

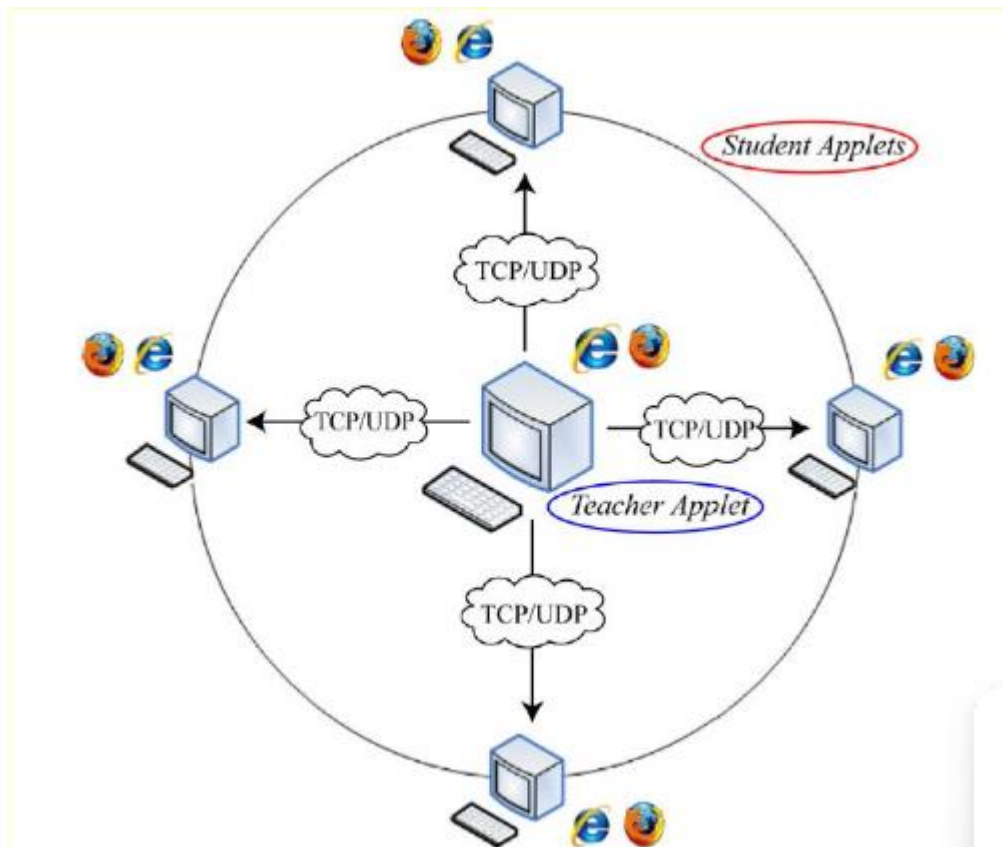


Figura 19: Esquema de un Laboratorio Virtual.

Fuente: Ortiz, K. N. T., Muñoz, D. C. H., & Mendoza, W. N. M. (2020).

Por lo tanto sus principales características son, permiten el acceso a recursos físicos en donde los estudiantes y docentes pueden acceder a instrumentos y maquinas de laboratorio sin importar la hora o la ubicación geográfica, así como también permiten una simulación extensa de experiencias presenciales a través de la red en donde los usuarios pueden realizar practicas como si estos se encontraran en un laboratorio físico, permiten el control del mismo de manera remota debido a que los usuarios pueden manipular equipos y realizar experimentos utilizando computadoras y o cualquier tipo de dispositivos móviles. Es por ello que se les considera como una herramienta valiosa para el aprendizaje práctico, brindando oportunidades de experimentación y descubrimiento sin las limitaciones geográficas o temporales de los laboratorios físicos.

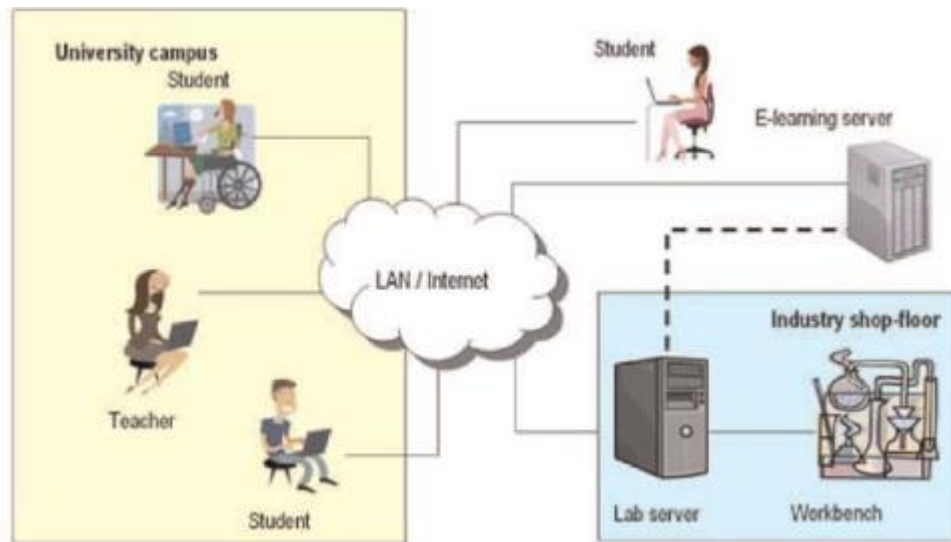


Figura 20: Configuración típica de un Laboratorio Remoto.
Fuente: (Cooper y Ferreira, 2009).

Dentro de todas las ventajas que se pueden encontrar en la implementación de este tipo de tecnologías destacan 4 principalmente, la de una mejor accesibilidad, en donde los estudiantes pueden acceder a los equipos y dispositivos desde cualquier lugar con conexión a internet, siendo así esencialmente útil para quienes no pueden estar físicamente presentes en el laboratorio, un ahorro de costos ya que al eliminar las necesidades de viajar o de mantener instalaciones físicas, los laboratorios remotos reducen los gastos asociados con los laboratorios tradicionales, la de una mejor seguridad ya que los estudiantes pueden realizar experimentos sin riesgos físicos, ya que no están en el mismo lugar que los equipos, siendo relevante para experimentos peligrosos o costosos, y finalmente una mejor flexibilidad ya que los laboratorios remotos permiten a los estudiantes trabajar en su propio horario y ritmo, adaptándose a sus necesidades y responsabilidades.

No obstante así como estas tecnologías permiten a los usuarios una amplia gama de ventajas, también presentan algunas desventajas, las cuales son las limitaciones de los usuarios, ya que en algunos casos solo un usuario puede acceder a un laboratorio físico remoto a la vez, lo que limita el número de estudiantes que pueden utilizarlo diariamente, además de presentar un elevado costo de su infraestructura ya que aun que los laboratorios remotos ahorran costos en comparación con los físicos, aún requieren inversión en infraestructura tecnológica y mantenimiento, otra

desventaja muy relevante suele ser la disponibilidad del equipo debido a que esta puede ser un desafío, especialmente si hay alta demanda o si los recursos son limitados, y final mente se presenta un desafío que aborda muchos contrastes el cual es una menor interacción física y es que aunque los laboratorios remotos brindan acceso a recursos reales, no ofrecen la misma interacción física que un laboratorio tradicional.

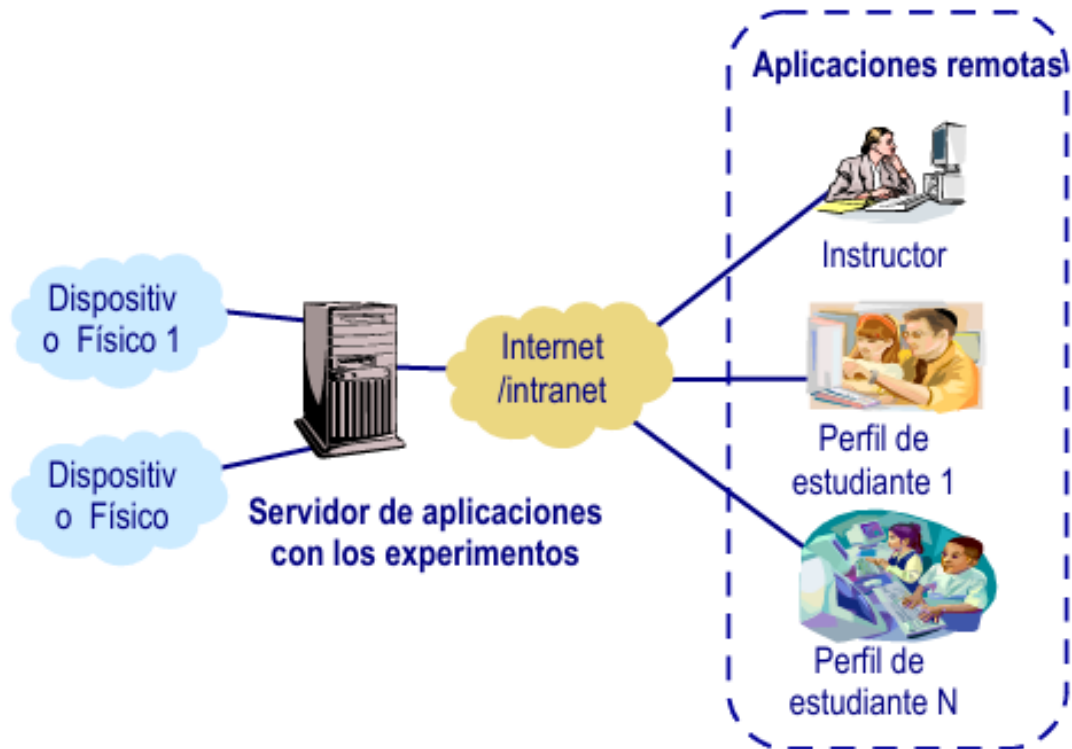


Figura 21: Arquitectura de referencia para la Construcción de un Laboratorio Remoto.
Fuente: Calvo Gordillo, I., Zulueta Guerrero, E., & López Guede, J. M. (2008).

Estas tecnologías se clasifican según su enfoque y su funcionalidad, dentro de los cuales destacan 3, estos son, los "Laboratorios remotos para Ingeniería Eléctrica, Electrónica y Mecatrónica" ya que estos laboratorios se utilizan en programas de ingeniería, permitiendo a los estudiantes acceder a equipos y dispositivos específicos relacionados con la electrónica, circuitos eléctricos y mecatrónica, siendo principal mente útiles para experimentos prácticos y proyectos de diseño, los "M00C Laboratorio Remotos de uso Masivo" ya que estos laboratorios están diseñados para un amplio público y están abiertos a todos proporcionando así una experiencia de aprendizaje práctico en línea y permiten a los usuarios realizar experimentos sin restricciones geográficas o temporales y final mente los "Remote

Labs (Laboratorios Remotos Especializados)” debido a que estos laboratorios están dirigidos a investigadores y estudiantes de posgrado, ofreciendo accesos a equipos y recursos especializados para proyectos de investigación avanzada y experimentación avanzada, siendo una herramienta valiosa para el aprendizaje práctico, brindando oportunidades de experimentación y descubrimiento sin las limitaciones geográficas o temporales de los laboratorios físicos.

2.3.1 Laboratorios Remotos para Ingeniería Eléctrica Electrónica y Mecatrónica.

Este tipo de laboratorios son una herramienta valiosa en la educación tecnológica, estos tipos de laboratorios se clasifican general mente en 3 Tipos Lab / TeleLab, M00C Lab y RemoteLab. Los primeros laboratorios “Lab/TeleLab” se han implementado en nueve áreas clave, incluyendo Instrumentación, Circuitos Eléctricos, Electrónica, Máquinas Eléctricas, Automatización, Redes Industriales, Robótica y Control. Estos están desplegados en cuatro campus del Tecnológico de Monterrey: Monterrey, Laguna, Tampico y Aguascalientes: atendiendo así aproximadamente a 200 alumnos por semestre, lo que equivale a alrededor de 1,200 alumnos en los últimos tres años.

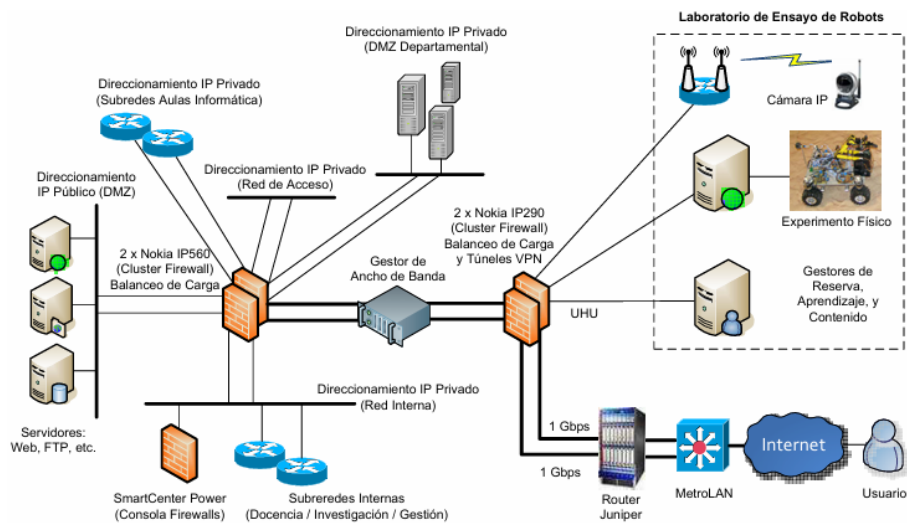


Figura 22: Arquitectura de referencia para la Construcción de un Laboratorio Remoto.
Fuente: Calvo Gordillo, I., Zulueta Guerrero, E., & López Guede, J. M. (2008).

CAPITULO TERCERO

3.1 Desarrollo y Evaluación del Proyecto

Ya hechas las investigaciones teóricas correspondientes para el desarrollo de este proyecto, se dio inicio al capítulo 3 en el cual se llevó a cabo el proceso de ejecución del proyecto. En este capítulo se pretende describir el diseño y la construcción e implementación de todos aquellos elementos, equipos, materiales y o dispositivos que conforman el prototipo a desarrollar. Cabe mencionar que este proyecto surgió a partir de un verano de investigación en el año 2023, en donde se planteó que uno de los principales problemas ocasionados por la emergencia sanitaria “Pandemia” del año 2020 fue la pérdida de conocimientos prácticos adquiridos principal mente en los laboratorios en los cuales se impartían las clases y mediante diversas practicas se aprendía y se le familiarizaba poco a poco al estudiante en un entorno de trabajo a nivel industria, es por ello que se tomó la decisión de desarrollar un prototipo de Sistema de Monitoreo controlado a distancia mediante el cual se pudiese observar todos aquellos procesos que se encuentren realizándose en el laboratorio en tiempo Real, desde cualquier lugar y sin ningún tipo de riesgo.

3.1.1 Diagrama De Flujo de la Metodología para el diseño de un Sistema de Monitoreo

En el siguiente diagrama de flujo, se representa de forma esquemática la metodología a seguir para el diseño y desarrollo de un Sistema de Monitoreo y Control basado en IoT



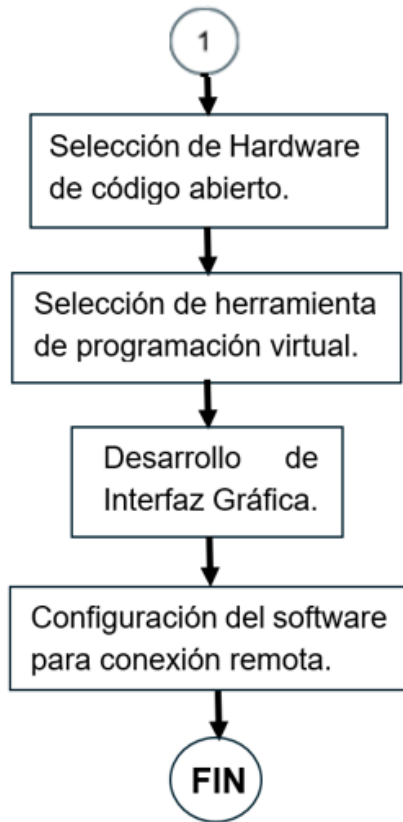
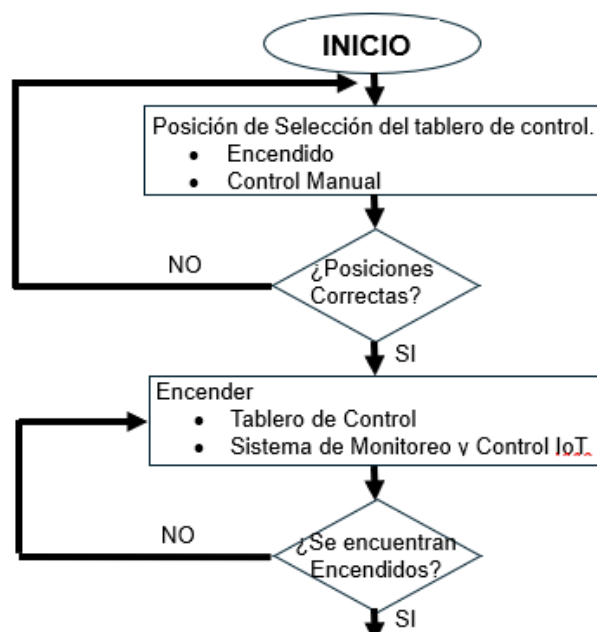


Figura 23. Diagrama de Flujo de la Metodología para el diseño de un Sistema de Monitoreo
Fuente: Elaboración Propia.

3.1.2 Diagrama De Flujo del Funcionamiento del Sistema



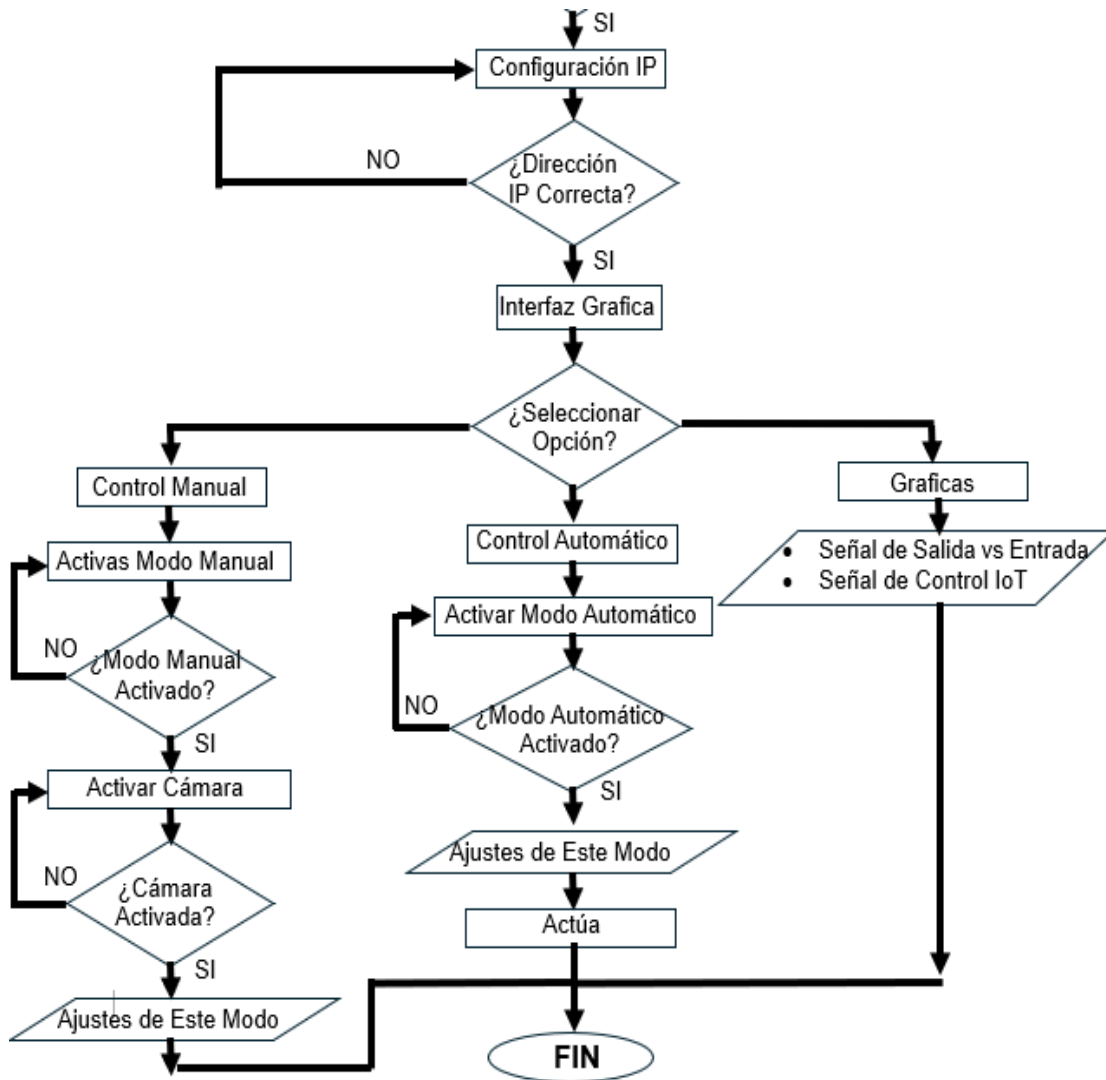


Figura 24. Diagrama de Flujo Principal del Proyecto
Fuente: Elaboración Propia.

En este diagrama de flujo, se plasmó y represento de forma esquemática en términos concretos, el proceso por el cual paso la construcción y elaboración el proyecto, es decir el funcionamiento que debe de llevar el Sistema de Monitoreo. Sin embargo, al ser una representación general, fue indispensable la elaboración de otro diagrama de flujo en donde se especificasen todas las funciones e instrucciones necesarias que el Sistema de Monitoreo debería realizar con las cuales una vez plasmadas en este diagrama, fuese más fácil el desarrollo de un código de programación de todo el funcionamiento del sistema en General.

3.1.3 Diseño, Modelado, Toma de Medidas y Adquisición del Material para desarrollar la Estructura General del Módulo de Monitoreo.

Como primer paso se realizó el diseño y desarrollo del armazón de la estructura en la que se monetaria el Sistema de Monitoreo, este en primera instancia se planteó que fuese hecho de madera.



Figura 25: Tiras de Madera para la elaboración de la Estructura del Modulo de Monitoreo
Fuente: Elaboración Propia 2024

Sin embargo después de un meticuloso análisis se concluyó que tanto por un concepto de estética, peso y funcionalidad, no eran tan eficientes en este material, es por ello que se determinó que el material más viable para su implementación debía de ser Aluminio.



Figura 26: Cortes en Tiras de Aluminio para la Estructura del Modelo de Monitoreo.
Fuente: Elaboración Propia 2024

Con el material ya decidido se procedió con la toma de medidas que se pretende tendría el esqueleto del sistema de monitoreo es decir el tamaño y la longitud de movilidad que este podría presentar.



Figura 27: Perfil de Extrusión de modelo "6105-t5"
Fuente: Elaboración Propia 2024

Debido a que el sistema de Monitoreo sería acoplado a un Laboratorio Remoto se determinó que este mismo debía tener ciertas medidas específicas para la visualización del todo el proceso que podría llevarse a cabo en el Laboratorio Remoto.



Figura 28: Toma de Medidas Correspondientes para la construcción del Sistema de Monitoreo
Fuente: Elaboración Propia 2024



Figura 29: Laboratorio Remoto sobre el cual se construirá el Sistema de Monitoreo
Fuente: Elaboración Propia 2024.

Una vez obtenidas las medidas pertenecientes a las dimensiones del Laboratorio Remoto las cuales fueron de 130cm de largo y 62cm de ancho, se procedió con la adquisición de 4 tiras de aluminio de un Perfil de Extrusión de modelo "6105-t5" de

4 mm de espesor, de una longitud de 150cm, con el objetivo de formar el esqueleto sobre el cual descansarían los rieles que le darían movilidad al módulo de Monitoreo.

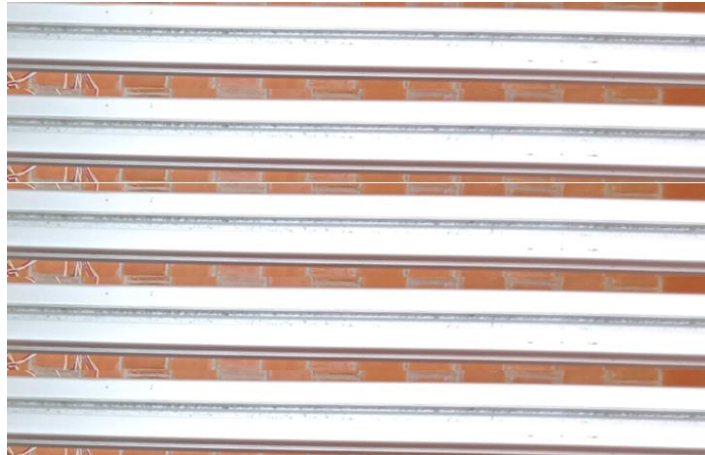


Figura 30: 4 Tiras de aluminio de un Perfil de Extrusión de modelo "6105-t5"
Fuente: Elaboración Propia 2024

Con las 4 Tiras del perfil de aluminio ya adquiridas y las medidas de la superficie el Laboratorio Remoto en el cual se ensamblaría la Estructura del Módulo de Monitoreo, se procedió con el proceso de cortado de las tiras de perfil de aluminio de 4 mm de espesor, para ello se implementó una cierra de banco con la que se realizaron los siguientes cortes a 90 "grados" 4 piezas de 90 cm, 2 piezas de 44.3cm y 2 piezas de 20.5 cm.

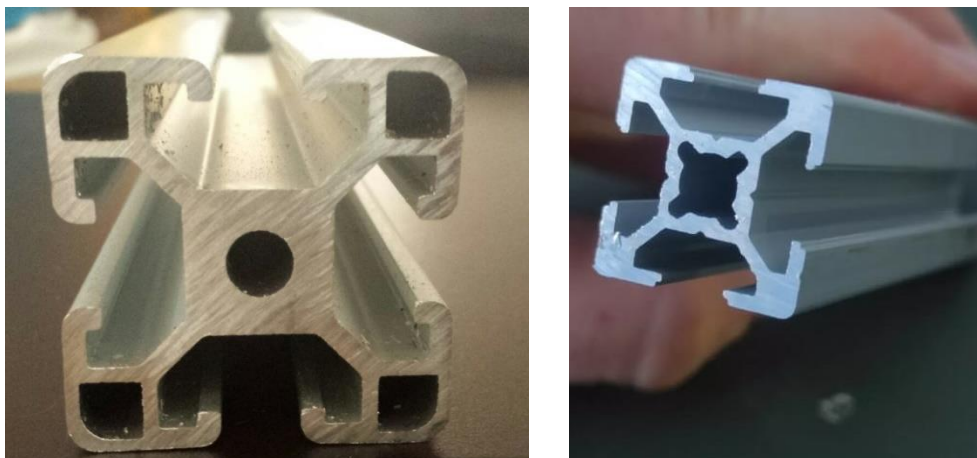


Figura 31: 8 Tiras ya cortadas de aluminio de un Perfil de Extrusión de modelo "6105-t5"
Fuente: Elaboración Propia 2024

3.2 Armado y Ensamblado de la Estructura del Módulo de Monitoreo

Yan con las 8 piezas ya cortadas a 90 grados, se procedió con el ensamble y armado de la estructura para el Módulo de Monitoreo, sin embargo, para la unión de las 8 piezas de aluminio se implementaron 8 escuadras para perfil de aluminio de 20mm x 20 mm, las cuales fueron adquiridas mediante una plataforma de ventas en línea “Amazon”, con un precio de \$296 por 10 escuadras.



Figura 32: Estructura ya Acoplada y Armada de la Estructura para el Módulo de Monitoreo.
Fuente: Elaboración Propia 2024

Con la estructura ya armada, se procedió con el diseño, desarrollo del dispositivo de movilidad del Módulo de Monitoreo, para ello se utilizó y adquirió un “Tornillo Sinfn” de una distancia de 62cm de longitud de placa a placa, junto con 2 rieles de movilidad lineal con sus respectivos 2 bloques de rodamiento con una distancia de 62cm de longitud y 2 rieles de movilidad lineal con sus respectivos 2 bloques de rodamiento para cada uno con una distancia de 1m de longitud.



Figura 33: Estructura de Movilidad y Desplazamiento del Módulo de Monitoreo.
Fuente: Elaboración Propia 2024

Con los rieles y el tornillo sin fin ya adquiridos se procedió con el armado de la estructura, sin embargo, la estructura de cada uno de ellos no se puede unir, ya que no cuenta con ningún tipo de perforación para su ensamble.



Figura 34: Montaje de los rieles y Tornillo sin Fin para la movilidad del Módulo de Monitoreo.
Fuente: Elaboración Propia 2024

Esto presento una serie de problemas debido a que el objetivo del módulo de Monitoreo es que este se pueda desplazar de izquierda a derecha, adelante y atrás, por ello la necesidad de implementar 4 rieles.

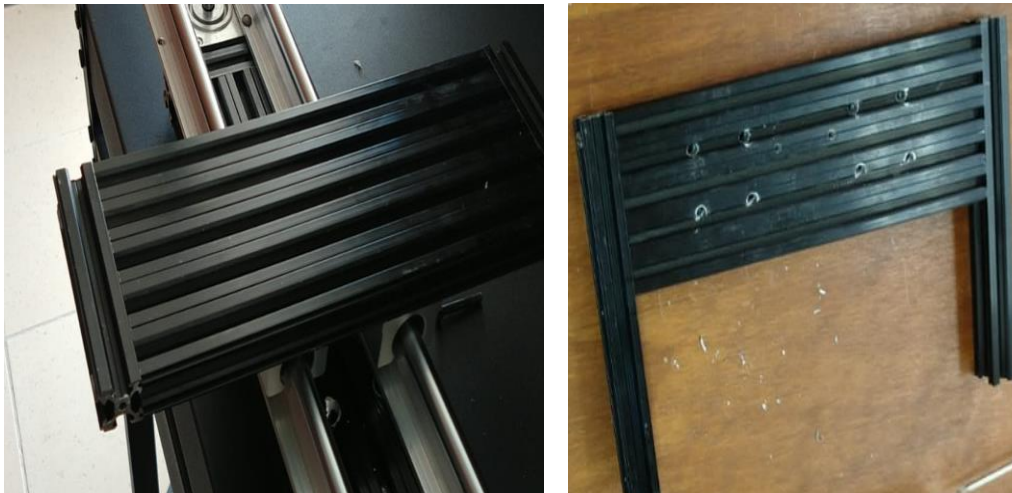


Figura 35: Placas de Perfil de aluminio de 3mm para el ensamblaje del Módulo de Monitoreo.
Fuente: Elaboración Propia 2024

Por este motivo se tomó la decisión de diseñar y desarrollar 2 placas de perfil estructural de Aluminio de 3mm de espesor, una con 30cm de largo y 10cm de ancho, y otra con 10cm de ancho y 16cm de largo.

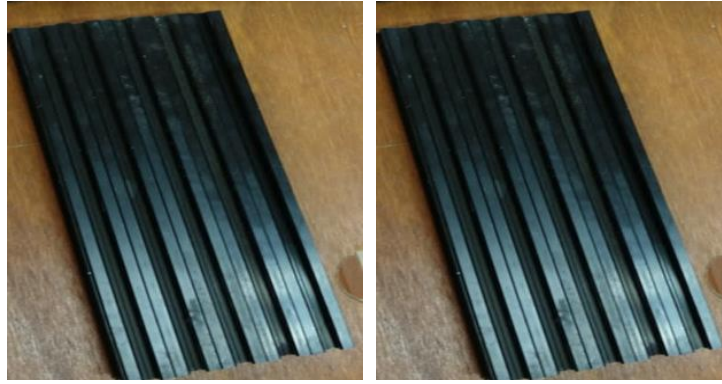


Figura 36: Placas cortadas a la medida de Perfil de aluminio de 3mm.
Fuente: Elaboración Propia 2024

La implementación de estas dos planchas de aluminio les permitiría a los 4 rieles darle una movilidad al Módulo de Monitoreo de todo el entorno, por ello se procedió con su perforación y ensamble.



Figura 37: Ensamble y acoplamiento de las placas cortadas de aluminio a la estructura del Modulo de Monitoreo.
Fuente: Elaboración Propia 2024

3.2.1 Dispositivos y Equipos Empleados en la Elaboración del Proyecto

Habiendo elaborado el diagrama de flujo se procedió con el empleamiento y adquisición de todos los dispositivos y equipo empleados para la elaboración del proyecto, así como también para realizar las pruebas y simulaciones pertinentes en cada proceso, dichos materiales fueron: 1 Fuente de Alimentación, 1 Multímetro Digital, 1 Protoboard, 1 Arduino Uno R4 Wifi, 1 Esp32 Cam, 4 Limit Swith, 2 Motores a Pasos Modelo Nema17, 1 Cabel Rj45 de 50cm, 1 Cable UTP 26AWG4PAIR 2m, 2 Cables Caimán-Caimán y 2 Cables Banana-Caimán y 1 Cable UART-USB, 1 Cople de Red, 1 Pluc de alimentación Dc, 1 Divisor extensor de Cable de conector RJ45.

- **Fuente de Alimentación**

Una fuente de alimentación también conocida como fuente de poder es aquel dispositivo eléctrico que permite convertir la corriente alterna (AC) en una corriente continua (DC), esto es gracias a un conversor de corriente. No obstante, además de convertir la corriente alterna en corriente continua, la fuente se encarga de proporcionar el voltaje necesario mediante una salida de la fuente, para la alimentación de circuitos y componentes electrónicos. Este equipo es esencial para dispositivos que funcionan con una DC y necesitan una alimentación constante, como la alimentación de una computadora, ya que utilizan la tecnología de conmutación en cuyo proceso se convierte la entrada de la corriente alterna a voltajes de corriente continua de salida. Existen diversos tipos de fuentes de alimentación (**Tabla III**), y todas ellas son clasificadas de acuerdo a las características de funcionamiento que tienen (Hart-Bautista, 2001; Boylestad et al., 2003).

Tabla III. Clasificación de los diferentes tipos de fuentes de Alimentación

Clasificación	Características
Tensión fija o tensión variable	Son el tipo de fuente de alimentación más sencillo ya que solo cuentan en su interior con un transformador, un rectificador un filtro y un regulador.
De una sola o Múltiples salidas	Son similares a las fuentes de alimentación fijas o variables, sin embargo, pueden llegar a contar con múltiples salidas, facilitando su uso, dependiendo los requerimientos.

Programables o de Forma convencional	Son capaces de reducir la tensión, convirtiendo el voltaje de forma completa, reduciéndolo y regulándolo, permitiendo un flujo constante, con la ventaja de poder ser manipulado por el usuario.
Conmutadas o conmutativas	Convierten la potencia por medio de sus transistores de potencia, implementadas para un sistema compacto y de bajo costo, diseñadas para trabajos de larga duración.
De alta potencia	Son capaces de mantener alimentados un gran número de componentes, que requieren de una alta potencia para su funcionamiento, soportando hasta 1000W de alta potencia.
De altas tensiones	Son capaces de poder dar hasta 1000v y son comúnmente utilizadas en los circuitos industriales.
Simuladores Fotovoltaicos	Tienen la capacidad de poder simular cada comportamiento de un panel fotovoltaico y batería.

En la elaboración de este proyecto se involucró la utilización de una fuente de alimentación (**Figura 17**), que le proporcionase la energía necesaria a los actuadores y sensores del brazo robótico poder realizar su función correctamente, por lo que fue indispensable usar una fuente de alimentación que saticiese los niveles de voltaje y amperaje necesarios para probar el funcionamiento de los componentes electrónicos y así hacer las primeras pruebas de su funcionamiento pertinentes.

Por tal motivo se optó por implementar otro tipo de fuente al proyecto, esta fue el modelo de “Fuente de Alimentación de mesa 72-13600 de la marca TENMA”. Con esta fuente de alimentación fue posible reducir la tensión de entrada a la fuente (125v) por medio de un transformador y permitiendo tener una salida con el voltaje mínimo necesario para la prueba de los componentes, así como también evito las oscilaciones de voltaje, y permitió una estabilidad de la señal continua.

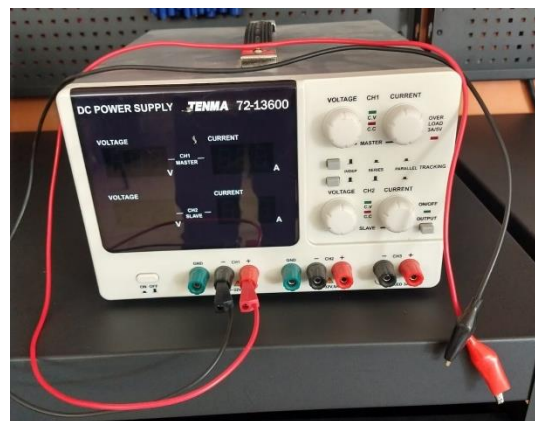


Figura 38: “Fuente de Alimentación de Mesa modelo 72-13600 TENMA”, usada en el proyecto.
Fuente: Elaboración Propia.

- **Multímetro Digital**

Un multímetro es un instrumento que se emplea para la medición de las intensidades de la corriente y los voltajes, este proceso de medición puede ser posible en corrientes continuas, en corrientes alternas y en resistencias, no obstante, existen algunos modelos que pueden llegar a medir, capacitancias, frecuencias y temperaturas (Caballero, *et al.* 2010). Estos instrumentos se dividen en dos tipos, los “analógicos”, los cuales pueden realizar sus lecturas mediante una aguja que se desplaza sobre una cierta escala y los “digitales”, los cuales no dependen de una aguja, sino que estos reflejan los valores por medio de una pantalla en la que se muestra de forma numérica el valor medido.



Figura 39: Multímetro Digital con cables de prueba de Sonda.
Fuente: Elaboración Propia.

Este dispositivo se utiliza con el único propósito de medir dos o más valores eléctricos o electrónicos, principalmente Tensiones (Voltios), corrientes (Amperios) y resistencias (Ohmios), fungiendo como una herramienta de diagnóstico para su empleador (Andrade, *et al.* 2010). Este tipo de multímetro (Imagen 19), son considerados como una mejor opción debido a que son más precisos y fáciles de poder visualizar sus valores, lo que los hace más prácticos y eficientes que los medidores analógicos que emplean una aguja, esto debido a la gran capacidad que estos manejan para medir con una mayor precisión, exactitud, fiabilidad (Barón Garcia. 2020).

El multímetro electrónico usado para el desarrollo de este proyecto (**Figura 18**), se consideró que funcionara de manera automática, por lo que las lecturas digitales

arrojadas serán determinadas de manera automática, según el tipo de escala que se maneje y los valores de los rangos máximos que soporta el instrumento.

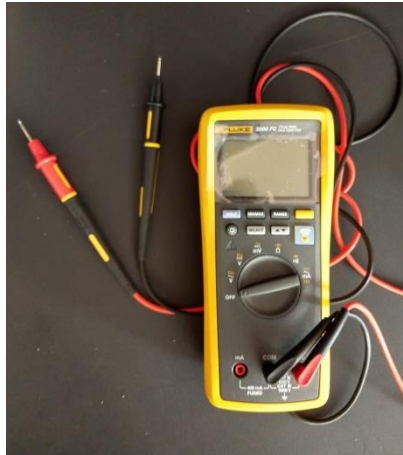


Figura 40: Multímetro Digital con cables de prueba de Sonda.
Fuente: Elaboración Propia.

Este es un multímetro digital con auto rango de modelo “mul-285” de la marca Steren, el cual permite medir con gran precisión, sin la necesidad de escoger las escalas, gracias a su función de auto rango, midiendo desde resistencia, voltaje de corriente alterna o directa, intensidad de corriente, capacitancia y continuidad. Ideal para la toma de mediciones necesarias de este proyecto (Tapia-Manzano, 2013).

- **Cables tipo Caimán-Caimán**

Los cables de tipo Caiman-Caiman son aquellos compuestos por 2 pinzas de metal una en cada uno de sus extremos, las cuales cuentan con resortes, mandíbulas largas y dentadas que permiten al conectarse a un material conductor, hacer puentes, pruebas, pasar corriente, alimentar circuitos eléctricos y/o electrónicos, sin la necesidad de soldar cables. Común mente este tipo de cables (**Figura 19**), se usan como tierras en las puntas de los equipos de pruebas como osciloscopios y multímetros, sin embargo, en este proyecto se implementaron 4 cables de este tipo para hacer un puente de continuidad entre los sensores Limit swith y los pines de un protoboard.



Figura 41: Cables de tipo Caiman-Caiman.
Fuente: Elaboración Propia.

- **Cables tipo Banana-Caimán**

Los cables de tipo Banana-Caimán, son aquellos compuestos por dos diferentes tipos de puntas, en un extremo las pinzas caimán que permiten morder o sujetar de manera fija aquellos dispositivos u conductores y por otro extremo las puntas tipo banana que se introducen dentro de una fuente de alimentación que permite la continuidad de la corriente en todo el cable hasta las pinzas caimán que al sujetarse de manera fija propician la continuidad de la energía proporcionada por la fuente de alimentación. Por tal motivo, se implementaron dos de estos tipos de cables (**Figura 20**), para puentear la energía proveniente de la fuente de alimentación hacia el protoboard y así poder alimentar todo el prototipo.



Figura 42. Cables de tipo Banana-Caiman, conectados a la salida de la fuente y a un conductor.
Fuente: Elaboración Propia.

- **Cable RJ45**

El cable RJ45 es un conector utilizado para conectar dispositivos en redes mediante un cable que puede tener hasta 8 hilos en su interior, sin embargo, aunque el conector tiene 8 pines, no siempre se utilizan todos, dentro de los dispositivos que puede conectar se encuentran las computadoras, los routers, los switches y los módems. Sirven principalmente para establecer conexiones más fiables y seguras en redes y estos pueden encontrarse en cables de red, televisores, videoconsolas, impresoras y otros periféricos.

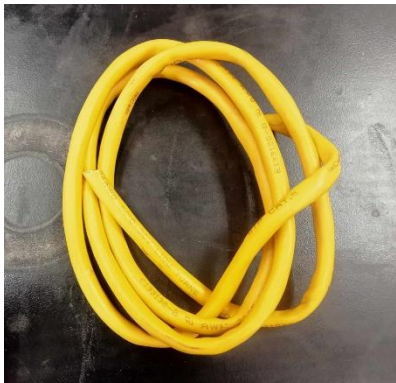


Figura 43: Cable Rj45
Fuente: Elaboración Propia 2024.

La implementación de este cable permitirá la conexión y comunicación entre los dos microcontroladores “Arduino UNO R4 Wifi” y “Esp32 Cam” obteniendo una mayor velocidad de conexión y de comunicación.

- **Jompers y Cable**

Estos pequeños cables mejor conocidos como Jompers, están compuestos por dos puntas metálicas conectadas por medio de un cable delgado, el cual propicia la conductividad entre cada una de ellos, estos pequeños cables se implementan para alimentar y/o comunicar dos dispositivos electrónicos, sin embargo la conductividad se encuentra limitada debido a que carecen de una larga distancia, sin embargo sus pines metálicos en los extremos del cable se acoplan perfectamente a las entradas de las tarjetas Arduino y los protoboard. En este proyecto se implementó un paquete de jompers, para la conexión y alimentación de la palca.



Figura 44. Jompers (puentes).
Fuente: Elaboración Propia.

- **Arduino UNO R4 Wifi**

Es una tarjeta electrónica digital así mismo es la cuarta generación de la placa Arduino Uno, y como su nombre lo indica, integra capacidades WiFi, lo que le permite conectarse y comunicarse de manera inalámbrica con otros dispositivos y la web. Esta placa cuenta con un módulo ESP32 para conectividad WiFi y Bluetooth, además de contar con una Matriz LED 12x8 y un Qwiic Connector, además de una interfaz de comunicación, un regulador de voltaje, protecciones, un oscilador y contiene principalmente un microcontrolador el cual puede ser programado, así mismo cuenta con puertos de entrada y salida, entradas analógicas, protocolos de comunicación, y cuenta, así como también tiene un solo idioma, el cual es el código binario, sin embargo, utiliza un lenguaje de programación propio de uso libre para que exista una comunicación con el programador.

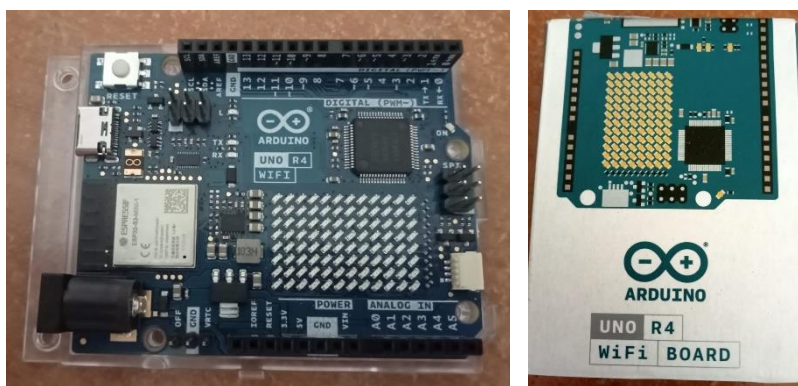


Figura 45: Tarjeta electrónica digital Arduino Uno, modelo "Atmega328p".
Fuente: Elaboración Propia.

Este tipo de placas, se utilizan para desarrollo de elementos autónomos, conectarse a otros dispositivos o interactuar con otros programas, por lo que su principal función

es la de automatizar procesos digitales, esto puede ser para aplicaciones caseras o de ambiente industrial.

- **Modulo Esp32 Cam**

El Modulo ESP32-CAM es un módulo compacto que combina el chip ESP32 con una cámara integrada, este dispositivo combina el procesamiento de imágenes y capacidades de Wi-Fi/Bluetooth, siendo una herramienta versátil que se puede implementar en proyectos de monitoreo remoto.



Figura 46: Modulo Esp32 Cam.
Fuente: Elaboración Propia 2024

Este dispositivo fue implementado en el proyecto debido a que se requiere un módulo con el cual se realizaran todos los procesos de Monitoreo, por lo que al analizar este Microcontrolador, se concluyó que su implementación en el proyecto era indispensable, fungiendo como herramienta de control.

- **MicroSwitch**

Un microswitch, es un interrupto eléctrico pequeño, llamado también interruptor rápido, el cual es accionado por medio de una presión física sobre él, sin embargo su principal característica, es la sensibilidad que tiene con respecto a otros interruptores, ya que un microswitch (**Figura 27**), se acciona con muy poca fuerza física, esto debido a que la fuerza que se ejerce para su accionamiento recae en una placa rectangular de lámina (lengüeta de acción), la cual hace palanca con la estructura del switch y le permite accionarlo de una manera más rápida.

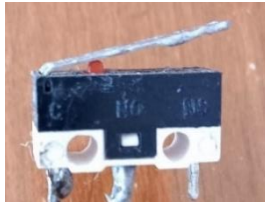


Figura 47: Limit switch o sensor final de carrera.
Fuente: Elaboración Propia.

Este dispositivo fue implementado en el proyecto debido a que microswitch puede adaptarse a espacios muy reducidos, lo que lo hace ideal debido a que los contenedores en los cuales se depositara el material proveniente de la línea de producción son pequeños, además al ser muy sensibles al peso, al caer el material sobre él, lo accionara fácilmente, haciendo funcionar al brazo robótico.

- **Protoboard**

El Protoboard también llamado breadboard, es una “placa” dispositivo electrónico de forma rectangular la cual cuenta con pequeños orificios que se encuentran conectados entre sí por medio de láminas metálicas, que se utilizan para las pruebas de funcionamiento de diversos dispositivos electrónicos, en esta placa pueden insertarse diversos dispositivos electrónicos y cables con los cuales se pueden hacer pruebas de continuidad y funcionabilidad, así como también elaborar circuitos sin la necesidad de soldar y desoldar ninguno de los componentes, estas tarjetas pueden utilizarse tantas veces como se necesite además de que su manipulación es muy sencilla,

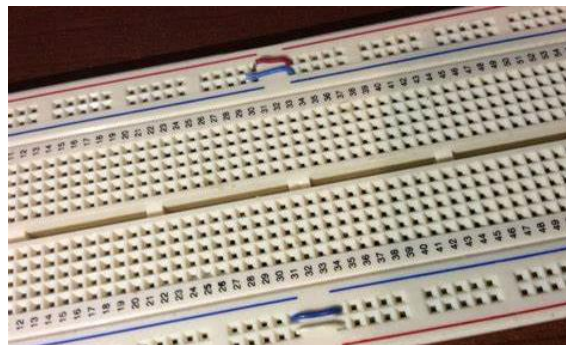


Figura 48. Placa Protoboard.
Fuente: Elaboración Propia.

El uso de esta placa es imprescindible para el proyecto, ya que sobre ella se harán las conexiones pertinentes para alimentar a los actuadores (servomotores, push

button, y microswitch), los cuales se conectarán directamente en los buces (canales positivo y negativo), y del mismo modo se conectará la fuente de alimentación, con el fin de proporcionarle a cada componente el voltaje necesario para su funcionamiento. De forma adicional se emplearon los siguientes dispositivos, los cuales fueron acoplados para el desarrollo del Módulo de Monitoreo:

- **FTDI TTL USB Serial Converter FT232RL 7**



Figura 49: FTDI TTL USB Serial Converter FT232RL 7
Fuente: Elaboración Propia 2024

Este dispositivo es esencial para la programación del Microcontrolador Esp32 Cam

- **Conector de Alimentación/2300 “Jack dc hembra de 25mm”**



Figura 50: Conector de Alimentación/2300 “Jack Hembra”
Fuente: Elaboración Propia 2024

Este dispositivo es esencial ya que mediante su implementación se podrá energizar el sistema, por medio de su instalación soldada en una placa de Circuito Impreso.

- **Conector de Alimentación/2300 “Jack Macho 55mm x 21mm”**



Figura 51: Conector de Alimentación/2300 “Jack Macho”
Fuente: Elaboración Propia 2024

Este dispositivo es esencial ya que mediante su implementación se podrá energizar el sistema, por medio de su instalación soldada en una placa de Circuito Impreso.

- **Divisor Extensor de Cable de Conector Hembra RJ45**



Figura 52: Divisor Extensor de Cable de Conector Hembra RJ45
Fuente: Elaboración Propia 2024

Este dispositivo, también conocido como (Adaptador de Cable Ethernet de 2 Piezas, Divisor Extensor de Cable de Conector LAN de Interfaz Hembra RJ45 de 1 Entrada Y 2 Salidas para Cable de Internet, Divisor Extensor de Cable ABS Negro), es un adaptador que permite la interconexión de dos Cables RJ45, empleados en el desarrollo de este proyecto.

3.2.2 Selección y Adquisición de los Motores a Pasos

Con los respectivos rieles y las placas ya montados en la estructura del Módulo de Monitoreo, se procedió con la adquisición de 2 Motores a Pasos, con la finalidad de ser las piezas claves en el desplazamiento del Módulo, ya que estos le brindarían la movilidad gradual ideal para el monitoreo constante del Laboratorio Remoto.



Figura 53: Armazón y estructura del Módulo de Monitoreo con su respectivo riel
Fuente: Elaboración Propia 2024

Después de una exhaustiva investigación sobre los motores a pasos, se decidió por la adquisición de 2 Motores modelo Nema-17.



Figura 54: Motores a pasos o “passo a passo” Modelo Nema-17.
Fuente: Elaboración Propia 2024.

Tabla IV. Especificaciones Generales de los Motores a Pasos

Step Angle (°)	1.8
Temperature Rise (°C)	80 Max (rated current, 2 phase on)
Ambient temperature (°C)	-20~+50
Number of Phase	2
Insulation Resistance	100MΩ, Min (500VDC)
Insulation Class	Class B
Max.radial force (N)	28 (20mm from the flange)
Max.axial force (N)	10

Fuente: Elaboración Propia.

Debido a que estos poseen un voltaj nominal de 12v y trabajan a una corriente de 1.2 A a 4v, además de que su ángulo de paso es de 1.8 grados, contando con 4 faces y 200 pasos por revolución, se concluyó que todas estas características lo hacen superior a otros modelos de motores a pasos.

Tabla V. Especificaciones Eléctricas de los Motores a Pasos

Rated Voltage (V)	4
Rated Current (A)	1.2
Resistance Per Phase ($\pm 10\% \Omega$)	3.3 (25°C)
Inductance Per Phase ($\pm 20\% \text{mH}$)	2.8
Holding Torque (Kg.cm)	3.17
Detent Torque (g.cm)	200
Rotor Inertia (g. cm ²)	68
Weight (Kg)	0.365

Fuente: Elaboración Propia.

Cabe mencionar que este modelo cuenta con un Torque de 3.2kg por cm, esto quiere decir que la velocidad y fuerza de desplazamiento del Módulo de Monitoreo sería más que eficiente y le brindara una mayor estabilidad al momento de desplazarse con una mayor velocidad.



Figura 55: Acoplamiento de los Motores Paso a Paso Nema-17.
Fuente: Elaboración Propia 2024.

3.2.3 Selección y Adquisición de los Drivers para el Control de los Motores a Pasos.

Posterior mente se procedió con la adquisición de 2 Drivers TB6600 para controlar la movilidad gradual de los 2 motores a pasos.

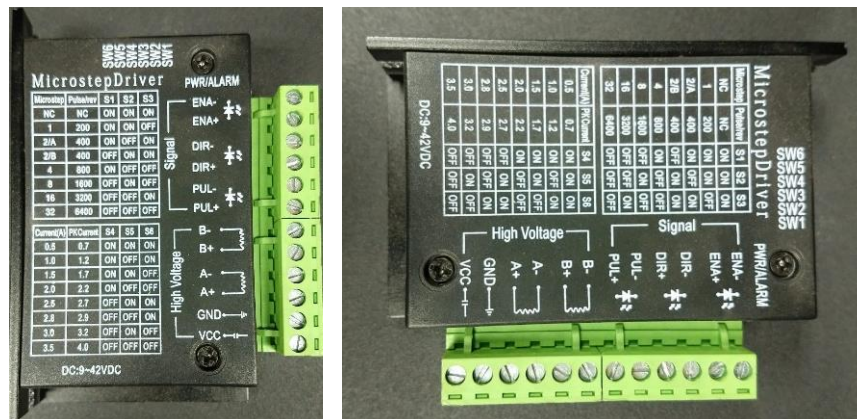


Figura 56: Drivers adquiridos con Modelo TB6600.
Fuente: Elaboración Propia 2024.

Estos trabajan de 9v a 42v Dc poseen 4 salidas las cuales serán usadas por las 4 fases de cada uno de los Motores a pasos.

Tabla VI. Especificaciones Eléctricas de los Drivers TB6600.

Input Current	0~5.0A
Output Current	0.5-4.0A
Power (MAX)	160W
Micro Step	1, 2/A, 2/B, 4, 8, 16, 32
Temperature	-10 ~ 45°C
Humidity	No Condensation
Weight	0.2 kg
Dimension	96*56*33 mm

Fuente: Elaboración Propia.

Adicional estos Drivers cuentan con 6 entradas 2 PWM y dos para el sentido del giro del Motor, contando con 6 selectores para los micropasos y la corriente a entregar a los motores la cual va desde 0.5 hasta 4 A.

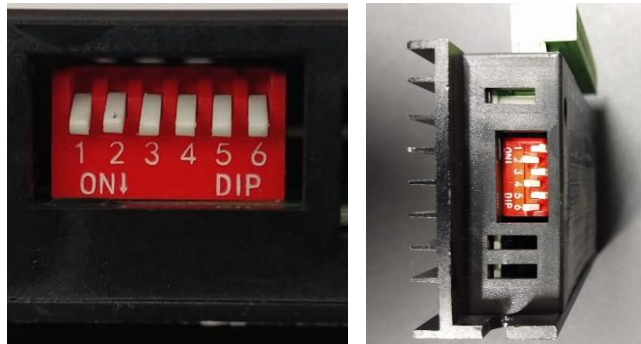


Figura 57: Especificación Técnica de los Drivers
Fuente: Elaboración Propia 2024

Tabla VII. Configuración de la Corriente a Entregar a los Motores a Pasos de los Drivers TB6600.

Current (A)	S4	S5	S6
0.5	ON	ON	ON
1.0	ON	OFF	ON
1.5	ON	ON	OFF
2.0	ON	OFF	OFF
2.5	OFF	ON	ON
2.8	OFF	OFF	ON
3.0	OFF	ON	OFF
3.5	OFF	OFF	OFF

Fuente: Elaboración Propia.

3.2.4 Ensamblado y Conexión de los Drivers y los Motores a Pasos

Ya adquiridos estos se, procedió con el montaje, cableado y configuración de estos, por lo que para su montaje se implementaron dos pijas a la base principal del Laboratorio Remoto, con la finalidad de que no tuviera juego ni movilidad.



Figura 58:
Fuente: Elaboración Propia 2024.

Posterior a esto se cablearon las 6 entradas de los Drivers y las 4 Salidas a cada uno de los 2 Motores a pasos.

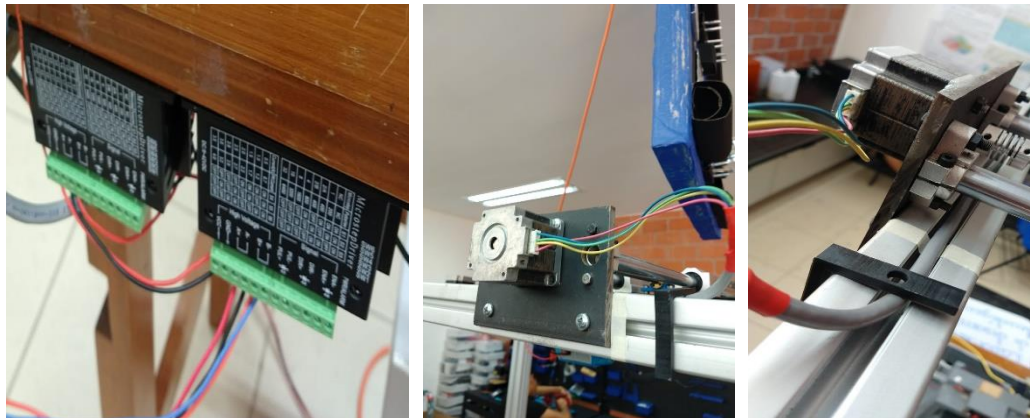


Figura 59:
Fuente: Elaboración Propia 2024

Para final mente configurarlos para determinar la cantidad de pasos y para definir la cantidad de corriente que se entrega, esto mediante la hoja de datos de las especificaciones de los motores a pasos.

Tabla VIII. Configuración de la Cantidad de Pasos por Revolución del Motor a Pasos de los Drivers TB6600.

Micro Step	Pulse/Rev	S1	S2	S3
NC	NC	ON	ON	ON
1	200	ON	ON	OFF
2/A	400	ON	OFF	ON
2/B	400	OFF	ON	ON
4	800	ON	OFF	OFF
8	1600	OFF	ON	OFF
16	3200	OFF	OFF	ON
32	6400	OFF	OFF	OFF

Fuente: Elaboración Propia.

- **Diagrama de Conexiones y Datos de los Drivers**

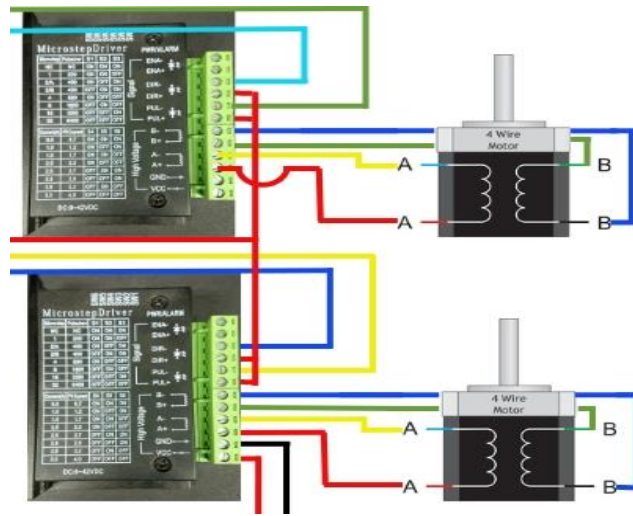


Figura 60: Diagrama de Conexión entre los Drivers y los Motores paso a paso.
Fuente: Elaboración Propia 2024

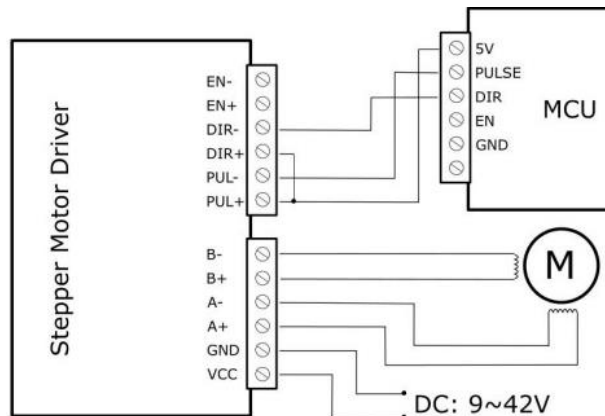


Figura 61: Diagrama de Conexión entre los Drivers y los Motores paso a paso 2.
Fuente: Elaboración Propia 2024

3.3 Integración de un Microcontrolador Arduino UNO R4 Wifi

Con las conexiones ya hechas entre los Drivers y los motores a pasos se procedió con la integración de un Microcontrolador Arduino UNO R4, por lo que se hicieron las conexiones para su comunicación entre los drivers y los Motores a Pasos.

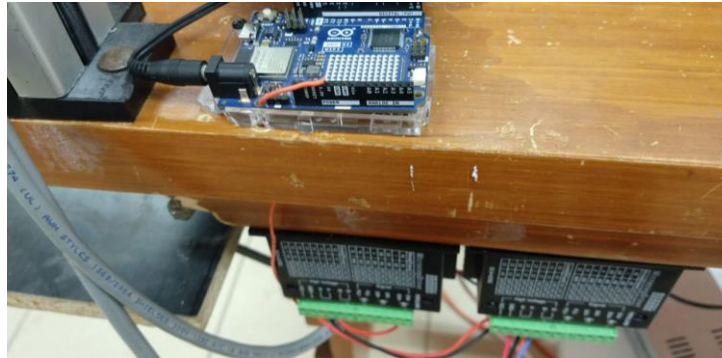


Figura 62: Conexión entre el Arduino UNO R4 Wifi, los Drives y los Motores a pasos.
Fuente: Elaboración Propia 2024

Se integro para las pruebas de movilidad gradual de cada Motor a pasos una serie de 4 Push botón conectados en un protoboard, con el objetivo de fungir como controlador de la activación del funcionamiento de cada motor, asimilando la orientación y/o dirección adelante, atrás, derecha, izquierda del campo de movilidad del Módulo de monitoreo. Esto con el fin de evidenciar la correcta movilidad y comunicación entre el microcontrolador Arduino UNO R4, los Drivers, y los 2 motores a pasos.

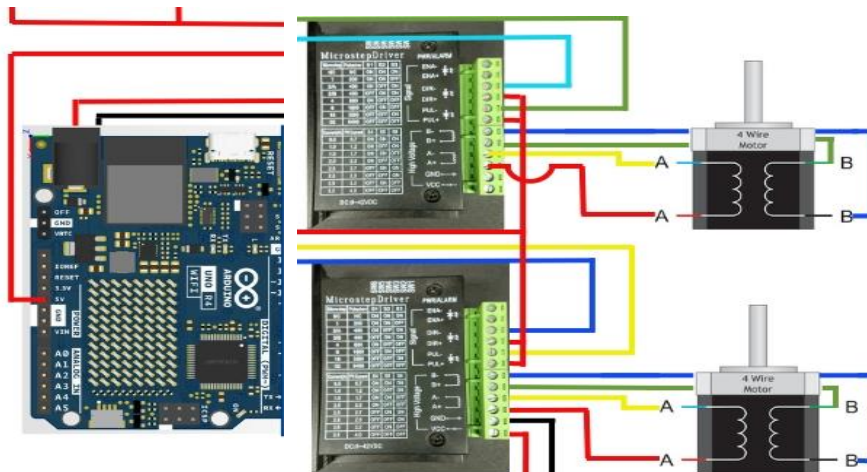


Figura 63: Diagrama de Conexión entre el Arduino UNO R4 Wifi, los Drives y los Motores a pasos
Fuente: Elaboración Propia 2024.

3.3.1 Diseño de un Código de Programación para el Control de Movilidad Gradual de los Motores a Pasos mediante un Microcontrolador Arduino UNO R4

Una vez diseñada la estructura por la cual se desplazaría el Módulo de Monitoreo, y con los dispositivos eléctricos y electrónicos ya adquiridos e implementados en el sistema, se procedió con la elaboración, diseño y desarrollo de un código de programación en Arduino, esto con la finalidad de evidenciar la correcta movilidad y comunicación entre el microcontrolador Arduino UNO R4, los Drivers, y los 2 motores a pasos

No obstante, cabe mencionar que este código no necesita de Arduino Cloud, ni un “Dashboard”, simplemente fugiendo como un código de prueba para la movilidad de los motores, es por ello que para su representación se hizo uso de 4 Push button, con los cuales se pretende que el módulo se mueva adelante, atrás, derecha, izquierda, permitiendo así un control del mismo al presionar un botón, consiguiendo así que se mueva a la dirección dependiendo la entrada a la que este conectada.

Como primer procedimiento se declararon las entradas digitales, con el fin de asignar un número del pin con estructura booleana del microcontrolador, numero del pin del microcontrolador, es decir se está declarando una entrada booleana.

```
1 //-----Declaración de entradas digitales
2 bool arriba    = 2;
3 bool abajo    = 3;
4 bool derecha  = 4;
5 bool izquierda = 5;
```

Figura 64: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Posterior mente se declararon los pulsos de control para cada uno de los Motores a Pasos.

```
6 //-----Declaración de pulsos de control de motor
7 const uint8_t boton    = 10; //boton de final de recorrido
8 const uint8_t PulsoX   = 9;  //Pulso X
9 const uint8_t DireccionX = 8; //Direccion X
10 const uint8_t DireccionY = 7; //Direccion Y
11 const uint8_t PulsoY   = 6;  //Pulso Y
12 int D;//delay
```

Figura 65: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Con los pulsos de control ya declarados para cada motor a pasos, se procedió con la declaración de las funciones empleadas para el control y desplazamiento.

```
13 //-----Declaración de funciones
14 void up() {
15     digitalWrite (DireccionY, HIGH);
16     delayMicroseconds (10);
17     for(int j=0;j<2000;j++){
18         digitalWrite (PulsoY, HIGH);
19         delayMicroseconds (300);
20         digitalWrite (PulsoY, LOW);
21         if (digitalRead (boton)==true) {
22             digitalWrite (DireccionY, HIGH);
23         }
24     }
25 }
```

Figura 66: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Para el cuarto proceso se propuso la declaración de las primeras funciones, ejecutadas para poder mandar a llamar la función, empleando una escritura digital, enviando el sentido de los pulsos.

```
27 void down() {
28     digitalWrite (DireccionY, LOW);
29     delayMicroseconds (10);
30     for(int j=0;j<2000;j++){
31         digitalWrite (PulsoY, HIGH);
32         delayMicroseconds (300);
33         digitalWrite (PulsoY, LOW);
34         if (digitalRead (boton)==true) {
35             digitalWrite (DireccionY, LOW);
36         }
37     }
38 }
```

Figura 67: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Para el quinto proceso se declararon las funciones de movimiento mediante ciclos “for” con el fin de mandarlas a llamar para posteriormente desplazar los motores.

```
40 void right() {
41     digitalWrite (DireccionX, HIGH);
42     delayMicroseconds (10);
43     for(int j=0;j<10000;j++){
44         digitalWrite (PulsoX, HIGH);
45         delayMicroseconds (300);
46         digitalWrite (PulsoX, LOW);
47         if (digitalRead (boton)==true) {
48             digitalWrite (DireccionX, HIGH);
49         }
50     }
51 }
```

Figura 68: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Para el sexto proceso se definieron los niveles y velocidades de los Motores a Pasos, así como la velocidad de la lectura para los pulsos y las direcciones.

```

53 void left() {
54     digitalWrite(DireccionX, LOW);
55     delayMicroseconds(10);
56     for(int j=0; j<10000; j++) {
57         digitalWrite(PulsoX, HIGH);
58         delayMicroseconds(300);
59         digitalWrite(PulsoX, LOW);
60         if (digitalRead(boton) == true) {
61             digitalWrite(DireccionX, LOW);
62         }
63     }
64 }

```

Figura 69: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Posterior mente se procedió con la declaración de los pines de entradas y salidas, con la finalidad de declarar los puntos de dirección de movilidad del módulo, Adelante, atrás, izquierda, derecha, linterna, y los pulsos para la dirección.

```

66 void setup() {
67     pinMode(arriba, INPUT);           //declaracion de pines de entrada y salida
68     pinMode(abajo, INPUT);
69     pinMode(PulsoX, OUTPUT);
70     pinMode(DireccionX, OUTPUT);
71     pinMode(DireccionY, OUTPUT);
72     pinMode(PulsoY, OUTPUT);
73     pinMode(derecha, INPUT);
74     pinMode(izquierda, INPUT);
75     pinMode(boton, INPUT);
76 }

```

Figura 70: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Por último, se procedió con el proceso de mandar a llamar todas las funciones anterior mente declaradas para su ejecución, es decir darle el sentido a cada pulso proporcionado por la declaración de cada una de las direcciones de movilidad del módulo de Monitoreo.

```

77 void loop(){
78     arriba = digitalRead(2);
79     abajo = digitalRead(3);
80     derecha = digitalRead(4);
81     izquierda = digitalRead(5);
82     if (arriba==true && abajo==false && derecha==false && izquierda==false){
83         up();
84     }
85     else if (arriba==false && abajo==true && derecha==false && izquierda==false){
86         down();
87     }
88     else if (arriba==false && abajo==false && derecha==true && izquierda==false){
89         right();
90     }

```

Figura 71: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

Final mente se procedió a simular el código para corregir todos los posibles errores de programación y a su vez observar el funcionamiento mediante el desplazamiento del soporte para el módulo de monitoreo.

```
Prueba_motores_a_pasos Arduino 1.8.4
Archivo Editar Programa Herramientas Ayuda

1 //-----Declaración de entradas digitales
2 bool arriba = 2;
3 bool abajo = 3;
4 bool derecha = 4;
5 bool izquierda = 5;
6 //-----Declaración de pulsos de control de motor
7 const uint8_t boton = 10; //boton de final de recorrido
8 const uint8_t PulsoX = 9; //Pulso X
9 const uint8_t DireccionX = 8; //Direccion X
10 const uint8_t DireccionY = 7; //Direccion Y
11 const uint8_t PulsoY = 6; //Pulso Y
12 int D;//delay
13 //-----Declaración de funciones
14 void up() {
15     digitalWrite(DireccionY,HIGH);
16     delayMicroseconds(10);
17     for(int j=0;j<2000;j++){
18         digitalWrite(PulsoY,HIGH);
19         delayMicroseconds(300);
20         digitalWrite(PulsoY,LOW);
21         if (digitalRead(boton)==true){
22             digitalWrite(DireccionY,HIGH);
23         }
24     }
25 }
26
27 void down() {
```

Figura 72: Código de programación en Arduino para el control de los Motores con Arduino R4.
Fuente: Elaboración Propia 2024.

3.3.2 Pruebas del Código de Movilidad de los Motores a Pasos.

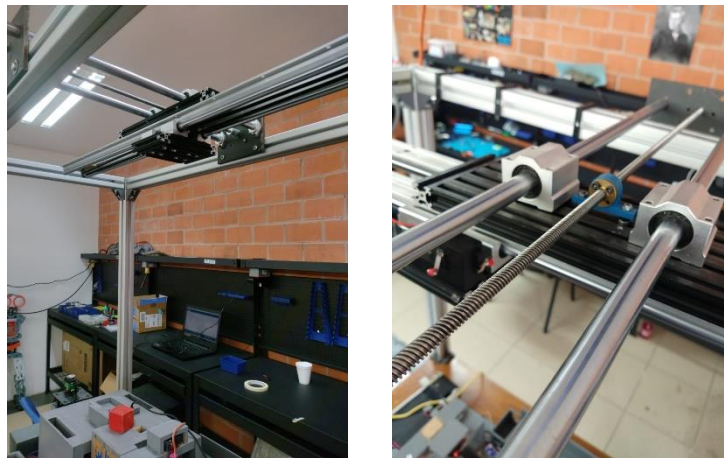


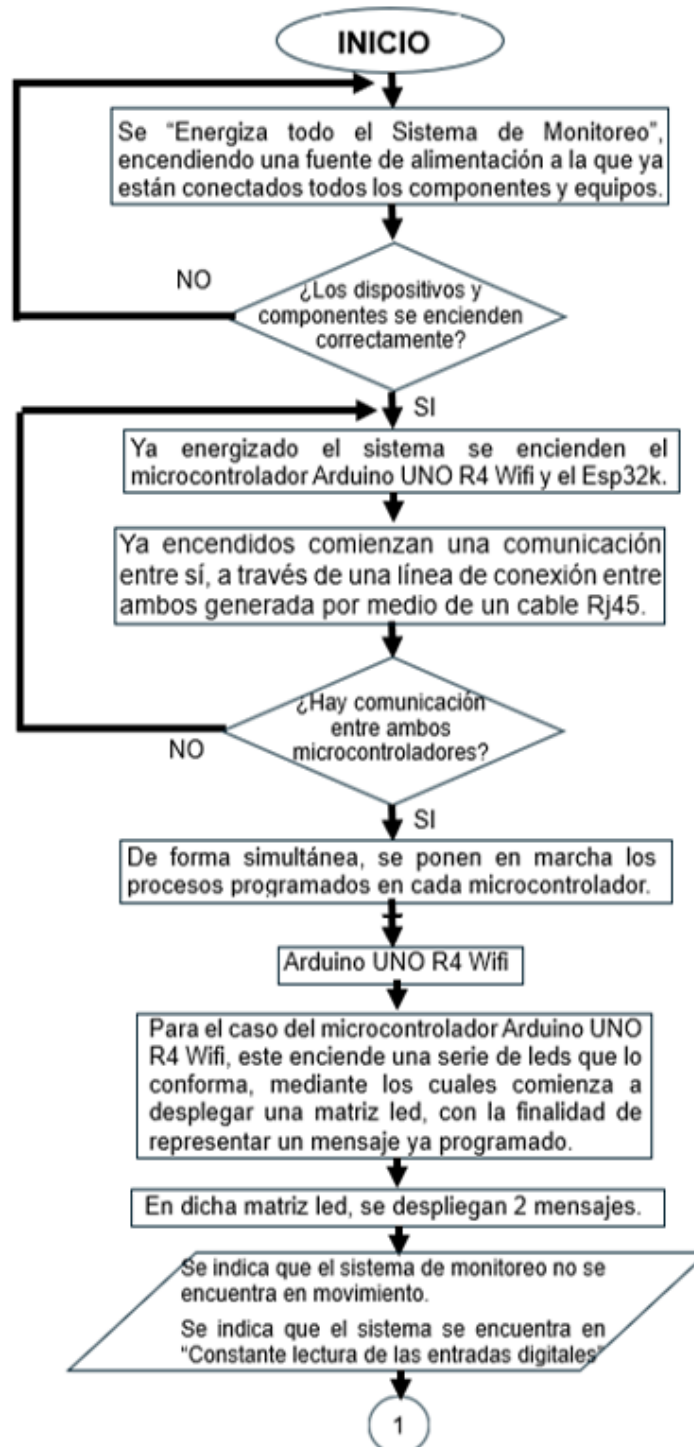
Figura 73: Pruebas del Código de programación para la Movilidad de los Motores.
Fuente: Elaboración Propia 2024.



Figura 74: Prueba del Código de programación para la Movilidad de los Motores.
Fuente: Elaboración Propia 2024.

3.3.3 Diseño de un Diagrama de Flujo de la Programación para el Control de los Motores a Pasos

En el siguiente “Diagrama de Flujo” (Figura 73), se explica de manera general la composición del código y la estructura el proceso que llevo a la programación de este.



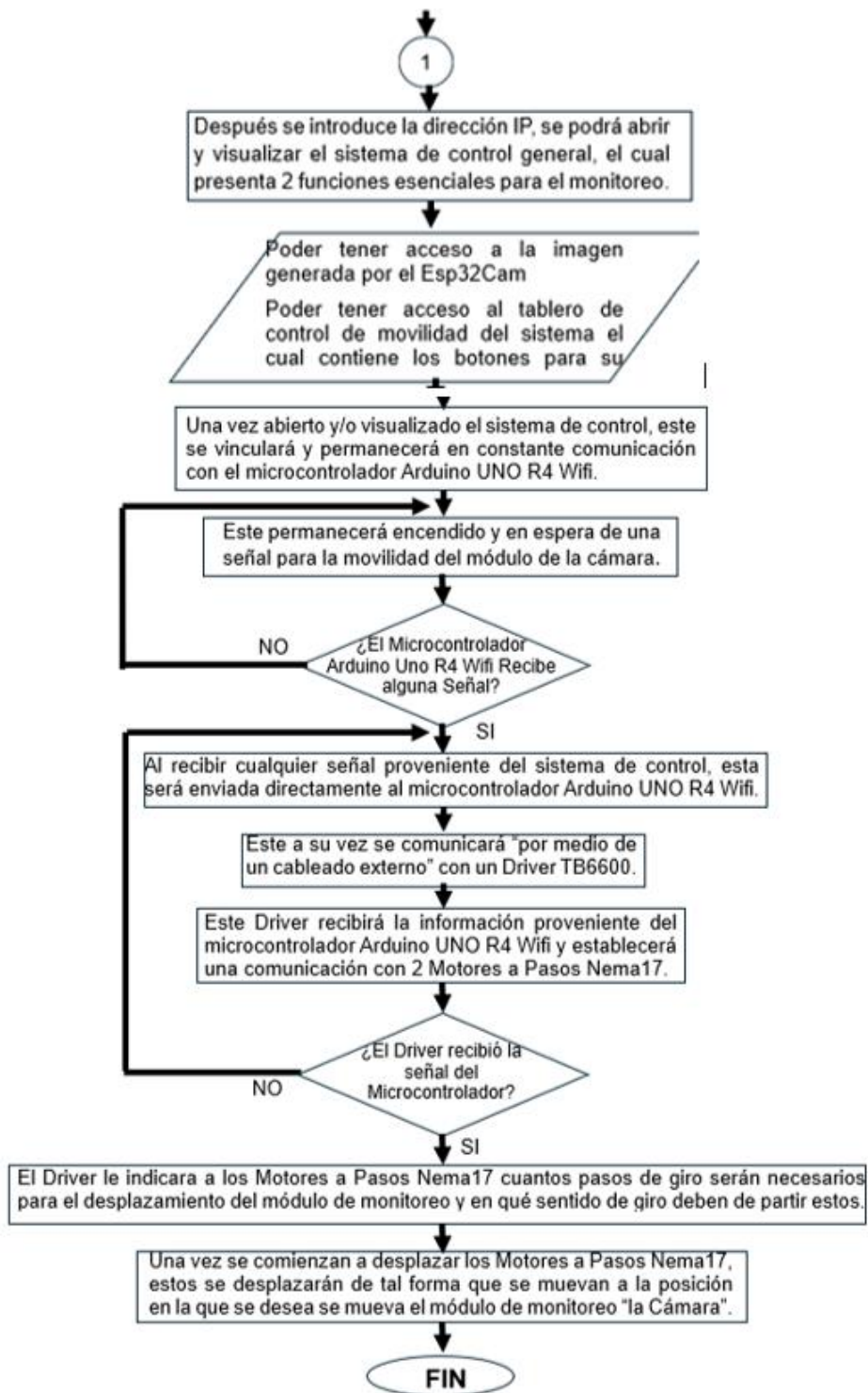


Figura 75. Diagrama de Flujo Principal de la programación para el Control de los Motores a Pasos
Fuente: Elaboración Propia.

En este diagrama de flujo, se pretende entender la cronología de cada paso u proceso a realizar del diseño, desarrollo y modelado del proyecto, con el fin de servir como una guía para su entendimiento.

Una vez corroborado el código y verificado que el sistema funciona mediante la comunicación del Microcontrolador Arduino UNO R4, los 2 Drivers y los 2 Motores a Pasos, se pudo observar durante las pruebas de movilidad una serie de inconvenientes para el buen funcionamiento del Sistema de Monitoreo.



Figura 76: Prueba de Movilidad del Módulo de Monitoreo
Fuente: Elaboración Propia 2024

Como primer punto se pudo apreciar que, si se llegaba al final del recorrido que podría ofrecer el tornillo sin fin, o la banda, el borne que une el tornillo sin fin con el motor a pasos, se lastimaba, se doblaba por lo que generaba un pico de corriente debido a que el motor se detenía pero el tornillo sin fin no giraba por lo que se generaba un esfuerzo mayor aumentando así la fuerza de trabajo del mismo y por consiguiente el amperaje de trabajo del motor aumentaba de golpe



Figura 77: Primer Problema detectado en la estructura del Módulo de Monitoreo.
Fuente: Elaboración Propia 2024

Sin embargo, el voltaje no se desplomaba debido a que su alimentación provenía de una fuente de alimentación regulable, que lo impedía, no obstante, esto generaba un sobrecalentamiento del motor, con riesgo de falla.

Como segundo punto se pudo observar que el cople de aluminio flexible, se estiro pudiendo llegar hasta el punto de romperse, debido a que se sigue efectuando la tención de torque del tronillo sin fin, forzando el cople al grado de deformarlo.



Figura 78: Segundo Problema detectado en la estructura del Módulo de Monitoreo.
Fuente: Elaboración Propia 2024

Posterior mente esta pieza se cambió en los dos extremos del tornillo sin fin, es decir se colocó una pieza nueva.



Figura 79: Cambio del Resorte en el tornillo sin Fin
Fuente: Elaboración Propia 2024

Sin embargo, con la finalidad de evitar este mismo daño en el cople por segunda ocasión se determinó que si se colocaba un micro switch final de carrera, este enviaría la señal de pulso al microcontrolador, el cual le indica al driver la inversión de giro del motor. En otras palabras, cuando la placa de aluminio insertada en los 2 rieles de movilidad lineal con sus respectivos 2 bloques de rodamiento se desplaza hacia el final de su dirección “Final del Marco” esta presionará el micro switch y por consiguiente el motor invertirá su sentido de giro, con el objetivo de evitar el error anterior.



Figura 80: Micro Switch empleado en el proyecto.
Fuente: Elaboración Propia 2024

Es por ello que con la implementación de este dispositivo se concluye que es posible evitar así el exceso de torque del tornillo sin fin, y erradicando el error de deformación del cople.



Figura 81: Adquisición de 3 Micro Switch mas para su inserción en el Módulo de Monitoreo.
Fuente: Elaboración Propia 2024

Por lo tanto, se tomó la decisión de implementar 2 Microswitch en cada uno de los extremos laterales de la Placa desplazada, de tal forma que al acercarse y hacer presión sobre el marco de la estructura del sistema de Monitore, esta acción enviaría

el pulso correspondiente al microcontrolador, y este a su vez invertiría el giro del motor. Estos microswitch se pegaron en cada uno de los extremos laterales de la Placa desplazada, esto por medio de Silicón Frio. Para posterior a esto se procedió con el cableado para su comunicación con el microcontrolador Arduino UNO R4.



Figura 82: Cableado de los Micro Switch.
Fuente: Elaboración Propia 2024

Una vez realizado el cableado entre el Arduino UNO R4 Wifi y los Micro Switch se determinó que debido al rigor de estética y eficiencia el sistema de movilidad no debía ser operado mediante 4 pushbotton, además haciendo énfasis en la premisa fundamental del que el objetivo del sistema de monitoreo debería ser controlado de forma remota, se tomó la decisión de eliminar este modo de control e implementar un control remoto mediante un Dashboard diseñado en la plataforma de Arduino Cloud.

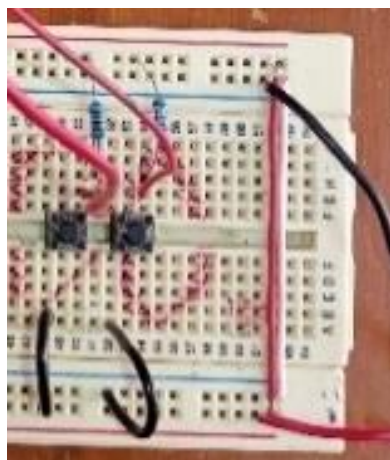


Figura 83: Empleamiento de 4 Push Button para el control de la Movilidad del Modulo.
Fuente: Elaboración Propia 2024

Con esta premisa enfundada, se procedió con la elaboración de un Dashboard, en Arduino Cloud

d, el cual se vincula a través de phings para generar la comunicación a un nivel digital a través del Arduino R4 mediante su sp32 integrada hacia los drivers y estos a su vez hacia los Motores a pasos.

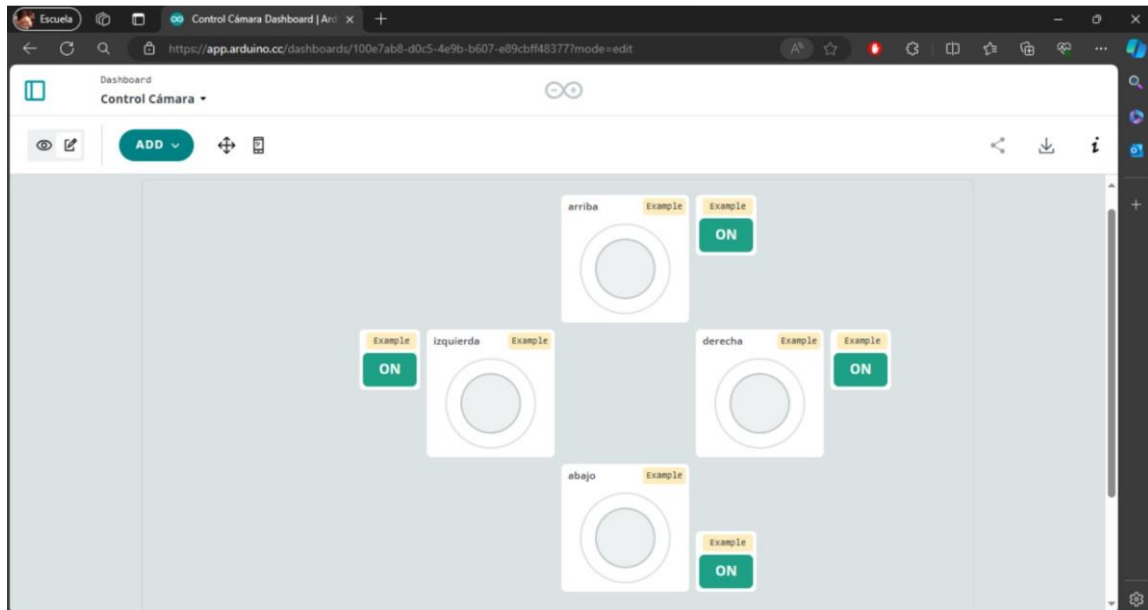


Figura 84: Dashboard generado en Arduino Cloud para el control del Modulo.
Fuente: Elaboración Propia 2024

Con este Dashboard, se pudo realizar el control de movilidad gradual perteneciente a cada motor a pasos, esto se pudo observar el control de la movilidad que se tenía por medio del Dashboard.

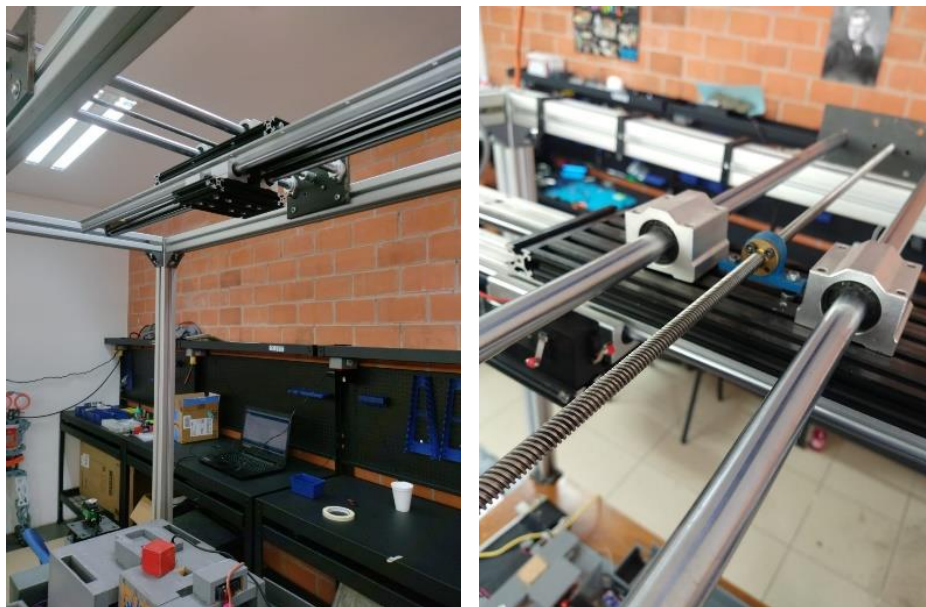


Figura 85: Prueba de Movilidad del Modulo de Monitoreo.
Fuente: Elaboración Propia 2024

3.3.4 Diseño de un Código de Programación para el Control de Movilidad de los Motores a Pasos mediante un Dashboard y un Microcontrolador Arduino UNO R4

```
18 #include "Arduino_LED_Matrix.h"
19 #include "saludo.h" //tab de capas de matriz
20 #include "frames.h" //tab de capas de matriz
21 #include "thingProperties.h"
22
23 int boton = 2;
24 int D;
25 ArduinoLEDMatrix matrix;
26
27 const uint8_t relayPin = 5;
28 const uint8_t PulsoX = 9; //Pulso X
29 const uint8_t DireccionX = 8; //Direccion X
30 const uint8_t DireccionY = 7; //Direccion Y
31 const uint8_t PulsoY = 6; //Pulso Y
32
```

Figura 86: Código de programación para el control de los Motores con un Dashboard y Arduino R4.
Fuente: Elaboración Propia 2024

Se desarrolló un código de programación en Arduino Cloud, en el cual se declararon al inicio del Código las librerías necesarias para la configuración del código, la parte del control de red.

```
33 void setup() {
34 // Initialize serial and wait for port to open:
35 Serial.begin(115200);
36 matrix.begin();
37 // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
38 delay(1500);
39
40 pinMode(PulsoX,OUTPUT);
41 pinMode(DireccionX,OUTPUT);
42 pinMode(DireccionY,OUTPUT);
43 pinMode(PulsoY,OUTPUT);
44
45 // Defined in thingProperties.h
46 initProperties();
47
48 // Connect to Arduino IoT Cloud
49 ArduinoCloud.begin(ArduinoIoTPreferredConnection);
50
```

Figura 87: Código de programación para el control de los Motores con un Dashboard y Arduino R4.
Fuente: Elaboración Propia 2024

Posterior mente se inicia la comunicación serial y se inicia la función de matriz, la cual se incluyó en la librería Arduino Led, se declararon los pines de salida para cada uno de los ejes y se inicia la función que genera una tabla en la cual se almacenan las variables que necesita el "Dashboard" para funcionar.

```
57 */
58 setDebugMessageLevel(2);
59 ArduinoCloud.printDebugInfo();
60 saludo();
61 }
62
63 void loop() {
64 ArduinoCloud.update();
65 // Your code here
66
67
68 }
```


Figura 88: Código de programación para el control de los Motores con un Dashboard y Arduino R4.

Fuente: Elaboración Propia 2024

Se declara la función de saludo fuera del “Void Loop” y dentro de este se encuentra declarada una función en la cual se manda a actualizar constantemente para la lectura de las funciones del “Dashboard”.

```
74 void onArribaChange() {
75   if(arriba){
76     matrix.loadFrame(arr);
77     digitalWrite(DireccionY,LOW);
78     delayMicroseconds(10);
79     for(int j=0;j<2000;j++){
80       digitalWrite(PulsoY,HIGH);
81       delayMicroseconds(300);
82       digitalWrite(PulsoY,LOW);
83       if (digitalRead(boton)==true){
84         matrix.loadFrame(aba);
85         digitalWrite(DireccionY,HIGH);
86       }
87     }
88   }
89 }
```

Figura 89: Código de programación para el control de los Motores con un Dashboard y Arduino R4.

Fuente: Elaboración Propia 2024

Después se describe la función que determina el movimiento del motor paso a paso, junto con la impresión de la figura en la matriz led indicando la dirección hacia arriba y con la condicional, que si este llega al final de carrera cambia la dirección de giro del motor.

```
88 }
89 else{
90   matrix.loadFrame(danger);
91 }
92 // Add your code here to act upon Arriba change
93 }
94
95 /*
96  Since Abajo is READ_WRITE variable, onAbajoChange() is
97  executed every time a new value is received from IoT Cloud.
98  */
99 void onAbajoChange() {
100  if(abajo){
101    matrix.loadFrame(aba);
102    digitalWrite(DireccionY,HIGH);
103    delayMicroseconds(10);
104    for(int j=0;j<2000;j++){
105      digitalWrite(PulsoY,HIGH);
106      delayMicroseconds(300);
107      digitalWrite(PulsoY,LOW);
108    }
109  }
110  }
111 }
112 }
113 }
114 else{
115   matrix.loadFrame(danger);
116 }
117 // Add your code here to act upon Abajo change
118 }
```

Figura 90: Código de programación para el control de los Motores con un Dashboard y Arduino R4.

Fuente: Elaboración Propia 2024

Posterior mente se describe la función que determina el movimiento del motor paso a paso, junto con la impresión de la figura en la matriz led indicando la dirección hacia abajo.

```
108   if (digitalRead(boton)==true){
109     matrix.loadFrame(der);
110     digitalWrite(DireccionY,LOW);
111   }
112 }
113 }
114 else{
115   matrix.loadFrame(danger);
116 }
117 // Add your code here to act upon Abajo change
118 }
```

Figura 91: Código de programación para el control de los Motores con un Dashboard y Arduino R4.
Fuente: Elaboración Propia 2024
Se representa la condicional, que si este llega al final de carrera cambia la dirección de giro del motor.

```

124 void onDerechaChange() {
125   if(derecha){
126     matrix.loadFrame(der);
127     digitalWrite(DireccionX, LOW);
128     delayMicroseconds(10);
129     for(int j=0; j<10000; j++){
130       digitalWrite(PulsoX, HIGH);
131       delayMicroseconds(300);
132       digitalWrite(PulsoX, LOW);
133       if (digitalRead(boton)==true){
134         matrix.loadFrame(izq);
135         digitalWrite(DireccionX, HIGH);
136       }
137     }
138   }
139   else{

```

Figura 92: Código de programación para el control de los Motores con un Dashboard y Arduino R4.
Fuente: Elaboración Propia 2024

En esta parte del código se describe la función que determina el movimiento del motor paso a paso, junto con la impresión de la figura en la matriz led indicando la dirección hacia la derecha y con la condicional, que si este llega al final de carrera cambia la dirección de giro del motor.

```

149 void onIzquierdaChange() {
150   if(izquierda){
151     matrix.loadFrame(izq);
152     digitalWrite(DireccionX, HIGH);
153     delayMicroseconds(10);
154     for(int j=0; j<10000; j++){
155       digitalWrite(PulsoX, HIGH);
156       delayMicroseconds(300);
157       digitalWrite(PulsoX, LOW);
158       if (digitalRead(boton)==true){
159         matrix.loadFrame(der);
160         digitalWrite(DireccionX, LOW);

```

Figura 93: Código de programación para el control de los Motores con un Dashboard y Arduino R4.
Fuente: Elaboración Propia 2024

Para después describir la función que determina el movimiento del motor paso a paso, junto con la impresión de la figura en la matriz led indicando la dirección hacia la izquierda y con la condicional, que si este llega al final de carrera cambia la dirección de giro del motor.

```

165   else{
166     matrix.loadFrame(danger);
167   }
168   // Add your code here to act upon Izquierda change
169 }
170 void saludo(){
171   D=60;
172   matrix.loadFrame(f1);
173   delay(D);
174   matrix.loadFrame(f2);
175   delay(D);
176   matrix.loadFrame(f3);
177   delay(D);
178   matrix.loadFrame(f4);
179   delay(D);

```

Figura 94: Código de programación para el control de los Motores con un Dashboard y Arduino R4.

Fuente: Elaboración Propia 2024

Así mismo se representa la función en la cual se guardan cada uno de los fotogramas con los que se realizó el saludo, el cual lleva movimiento a la hora de inicializar el Arduino, proyectado en la matriz led de este.

```
266 matrix.loadFrame(f48);
267 delay(D);
268 matrix.loadFrame(f49);
269 delay(300);
270 matrix.loadFrame(heart);
271 delay(500);
272 Serial.println(millis());
273
274 }
```

Figura 95: Código de programación para el control de los Motores con un Dashboard y Arduino R4.
Fuente: Elaboración Propia 2024

Para después representar la función en la cual se guardan cada uno de los fotogramas con los que se realizó el saludo, el cual lleva movimiento a la hora de inicializar el Arduino, proyectado en la matriz led de este.

```
3 #include <ArduinoIoTCloud.h>
4 #include <Arduino_ConnectionHandler.h>
5
6 const char DEVICE_LOGIN_NAME[] = "2ca1b33f-0f8b-4633-b509-6a2edc4f0c6f";
7
8 const char SSID[] = SECRET_SSID; // Network SSID (name)
9 const char PASS[] = SECRET_OPTIONAL_PASS; // Network password
10 const char DEVICE_KEY[] = SECRET_DEVICE_KEY; // Secret device password
11
12 void onAbajoChange();
13 void onArribaChange();
14 void onDerechaChange();
15 void onIzquierdaChange();
16
17 bool abajo;
18 bool arriba;
19 bool derecha;
20 bool izquierda;
```

Figura 96: Código de programación para el control de los Motores con un Dashboard y Arduino R4.
Fuente: Elaboración Propia 2024

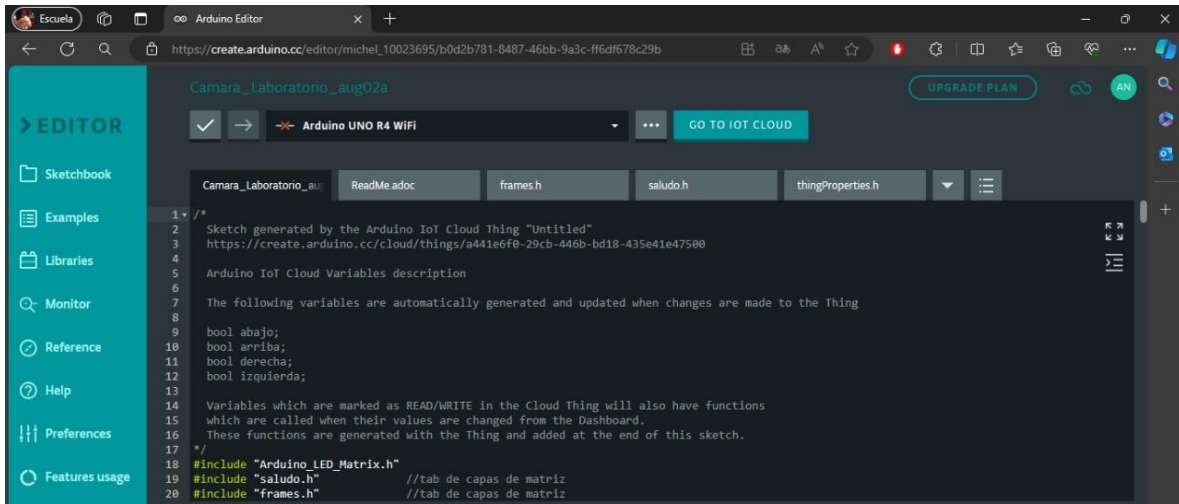
representando la tabla que se agrega en el código en la cual se guardan las contantes y variables utilizadas para el funcionamiento del “Dashboard” así como las constantes que tienen la red wifi a la cual se va a conectar y la contraseña.

```
21
22 void initProperties(){
23
24     ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
25     ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
26     ArduinoCloud.addProperty(abajo, READWRITE, ON_CHANGE, onAbajoChange);
27     ArduinoCloud.addProperty(arriba, READWRITE, ON_CHANGE, onArribaChange);
28     ArduinoCloud.addProperty(derecha, READWRITE, ON_CHANGE, onDerechaChange);
29     ArduinoCloud.addProperty(izquierda, READWRITE, ON_CHANGE, onIzquierdaChange);
30
31 }
32
33 WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
34
```

Figura 97: Código de programación para el control de los Motores con un Dashboard y Arduino R4.

Fuente: Elaboración Propia 2024

Para finalmente describir la función en la cual se mandan a llamar, las acciones que genera la librería para el funcionamiento de los botones del “Dashboard”.



```
1 /*  
2 Sketch generated by the Arduino IoT Cloud Thing "Untitled"  
3 https://create.arduino.cc/cloud/things/a441e6f0-29cb-446b-bd18-435e41e47500  
4  
5 Arduino IoT Cloud Variables description  
6  
7 The following variables are automatically generated and updated when changes are made to the Thing  
8  
9 bool abajo;  
10 bool arriba;  
11 bool derecha;  
12 bool izquierda;  
13  
14 Variables which are marked as READ/WRITE in the Cloud Thing will also have functions  
15 which are called when their values are changed from the Dashboard.  
16 These functions are generated with the Thing and added at the end of this sketch.  
17 */  
18 #include "Arduino_LED_Matrix.h"  
19 #include "saludo.h" //tab de capas de matriz  
20 #include "frames.h" //tab de capas de matriz
```

Figura 98: Código de programación para el control de los Motores con un Dashboard y Arduino R4.

Fuente: Elaboración Propia 2024

Cabe mencionar que este código genera las funciones con las cuales el “Dashboard” funciona, es decir son las líneas de código, comandos y configuraciones necesarias para hacer funcionar al “Dashboard”, ligadas a las variables declaradas.

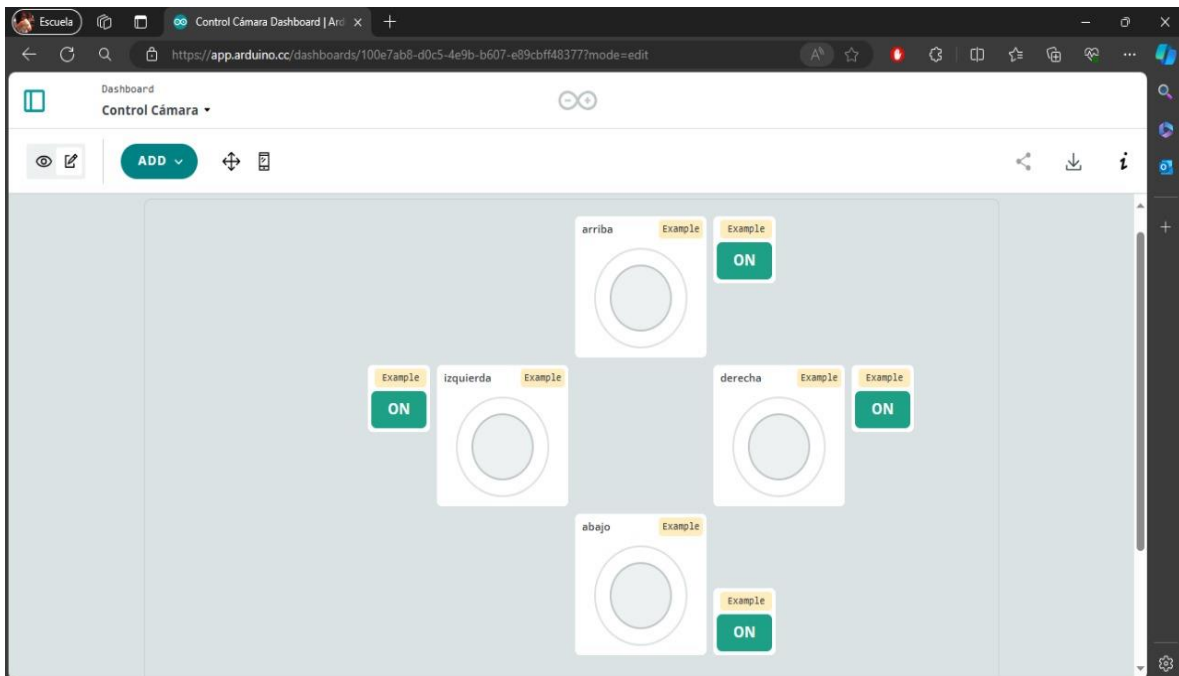


Figura 99: Dashboard Generado en Arduino Cloud para el control de los Motores Paso a Paso.

Fuente: Elaboración Propia 2024.

3.4 Integración de un Microcontrolador Esp32 Cam

Posterior mente se concluye que con la implementación del “Dashboard” se manda a llamar la función para que se muevan los motores a pasos desde Arduino Cloud. No obstante, después de un meticuloso análisis e investigación profunda, se concluyó que la plataforma de Arduino Cloud no permite la visualización de una cámara en el “Dashboard”, o de una imagen transmita por una cámara en el “Dashboard”. Es por ello por lo que se optó por una segunda opción la cual fue, la implementación de un segundo microcontrolador, con el objetivo de generar una página web que permitiera el poder transmitir una imagen y poder modificar a libertad el “Dashboard” de control.



Figura 100: Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024

Este microcontrolador fue el ESP32 Cam es decir un externo del que ya viene integrado en el Arduino R4, debido a que, de acuerdo con sus características y especificaciones, es el ideal para el sistema de Monitoreo, debido a que en su composición tiene integrado un módulo de cámara. Posterior a esto se realizaron las conexiones pertinentes para la comunicación entre ambos microcontroladores.

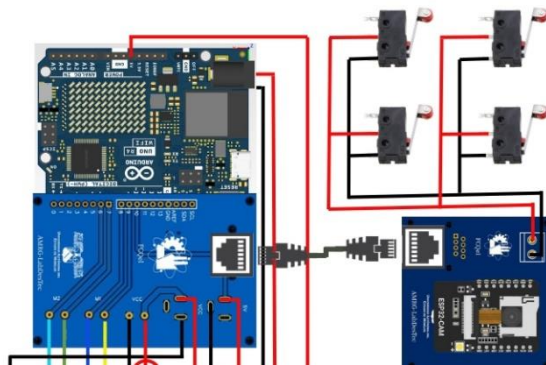


Figura 101: Microcontrolador Arduino UNO R4 Wifi en conexión con el Microcontrolador Esp32 Cam

Fuente: Elaboración Propia 2024.

3.4.1 Diseño de un Código de Programación para el Control del Microcontrolador Esp32 Cam

```
#include "esp_wifi.h" // Declaracion de librerias
#include "esp_camera.h"
#include <WiFi.h>
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h" // disable brownout problems
#include "soc/stc_ctrl_reg.h" // disable brownout problems
#include "esp_http_server.h"

// constantes generadas para identificación de la red wifi y contraseña de acceso
//const char *ssid = "Telcel_B311_AA4A";
const char *password = "PS[Zz#ib";
const char *ssid = "LabDesTecMich";
const char *password = "16023695";

#define PART_BOUNDARY "12345678900000000000000987654321"

#define CAMERA_MODEL_AI_THINKER
```

Figura 102: Código de programación en Arduino para el control del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024

En esta primera parte del código se muestra la declaración de las librerías necesarias para la configuración del código, la parte del control de red y los pines a utilizar en el Esp32 Cam para la configuración con el modelo de la cámara y se definen a su vez los pines de salida de la señal que se envía al microcontrolador Arduino UNO R4 Wifi.

```
-----Codigo html y css a utilizar
static const char PROGMEM INDEX_HTML[] = R"rawLiteral(
<html>
<head>
<title>FCQeI LabDesTec</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body { font-family: Arial; text-align: center; margin: 0px auto; padding-top: 30px;}
table (margin-left: auto; margin-right: auto;)
td { padding: 3px;}
.button{
background-color: #2f4468;
border-radius: 34px;
color: white;
padding: 10px 20px;
text-align: center;
```

Figura 103: Código de programación en Arduino para el control del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024

Posterior mente se guarda el código HTML, en una constante de caracteres, es decir La creación de la página web.


```

        background-color: #0ca513;
    }
</style>
</head>
<body>
<h1>FCQeI</h1>
<h1>Laboratorio de Desarrollos Tecnológicos</h1>
<h2>Sistema de monitoreo para laboratorio remoto</h2>
<img src="" id="photo" >
<table>
<tr><td colspan="3" align="center"><button class="button" onmousedown="toggleCheckba
<tr><td align="center"><button class="button" onmousedown="toggleCheckbox('left');".

```

Figura 104: Código de programación en Arduino para el control del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024

A continuación, se ve la generación del cuerpo de la página web, los títulos, la imagen mandada a llamar con el Id de foto para la transmisión de la cámara y la tabla en donde se generan los botones de acción mejor conocido como “Dashboard”.

```

Serial.println("Camera capture failed");
res = ESP_FAIL;
} else {
    if (fb->width > 400) {
        if (fb->format != PIXFORMAT_JPEG) {
            bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
            esp_camera_fb_return(fb);
            fb = NULL;
            if (!jpeg_converted) {
                Serial.println("JPEG compression failed");
                res = ESP_FAIL;
            }
        } else {
            _jpg_buf_len = fb->len;
            _jpg_buf = fb->buf;
        }
    }
}
if (res == ESP_OK) {

```

Figura 105: Código de programación en Arduino para el control del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024

En esta sección del Código se aprecia el formato para la inicialización de la cámara y mensajes de error en el caso de no cumplirlo.

```

//-----dec
if (!strcmp(variable, "forward")) {
    Serial.println("Arriba");
    digitalWrite(arriba, HIGH);
    digitalWrite(abajo, LOW);
    digitalWrite(derecha, LOW);
    digitalWrite(izquierda, LOW);
} else if (!strcmp(variable, "right")) {
    Serial.println("Derecha");
    digitalWrite(arriba, LOW);
    digitalWrite(abajo, LOW);
    digitalWrite(derecha, HIGH);
    digitalWrite(izquierda, LOW);
} else if (!strcmp(variable, "rightshort")) {
    Serial.println("Dex");

```

Figura 106: Código de programación en Arduino para el control del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024

Posterior mente se declara las condicionales de accionamiento a la hora de utilizar los botones de “Dashboard” y la inicialización del servidor en la cámara.

```

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout de
  //-----configuración de pines a utilizar
  pinMode(arriba, OUTPUT);
  pinMode(abajo, OUTPUT);
  pinMode(izquierda, OUTPUT);
  pinMode(derecha, OUTPUT);
  pinMode(FLASH_LED_PIN, OUTPUT);
  digitalWrite(FLASH_LED_PIN, flashState);

  Serial.begin(115200); //-----inicialización de comunicación s
  Serial.setDebugOutput(false);

  camera_config_t config; // configuración de pines
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;

```

Figura 107: Código de programación en Arduino para el control del Microcontrolador Esp32 Cam.

Fuente: Elaboración Propia 2024

Se establece la configuración donde se declara el modo y la configuración de los pines para las salidas digitales que se envían en el R4 wifi, así como el flash integrado, la inicialización de la comunicación serial, y la configuración de los pines de la cámara.

```

//-----inicialización de
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}
// Wi-Fi connection
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

Serial.print("Camera Stream Ready! Go to: 'http://'"); //---
Serial.println(WiFi.localIP());

//----inicia al servidor de la pagina web con la camara
startCameraServer();
}

void loop() {
}

```

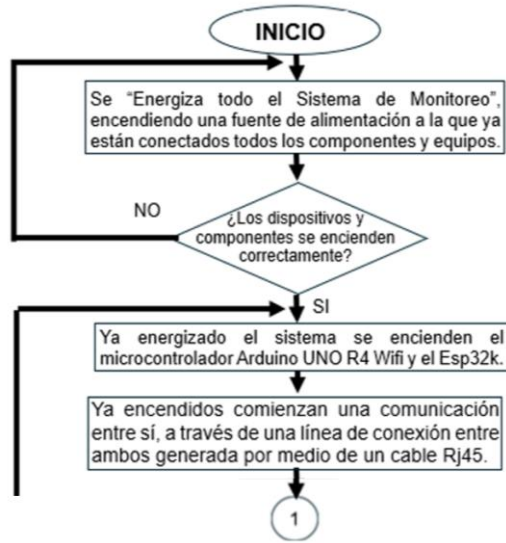
Figura 108: Código de programación en Arduino para el control del Microcontrolador Esp32 Cam.

Fuente: Elaboración Propia 2024

Para final mente declarar la conexión wifia y la escritura de la comunicación serial de la dirección IP generada para le servidor de la cámara y se inicia la transmisión de la cámara, es decir que se inicia todo el sistema de monitoreo.

3.4.2 Diagrama de Flujo del Funcionamiento del Código para el Control del Microcontrolador Esp32 Cam

Una vez terminado el código de programación, se elaboró un diagrama de flujo en donde se explica detalladamente dicho código.



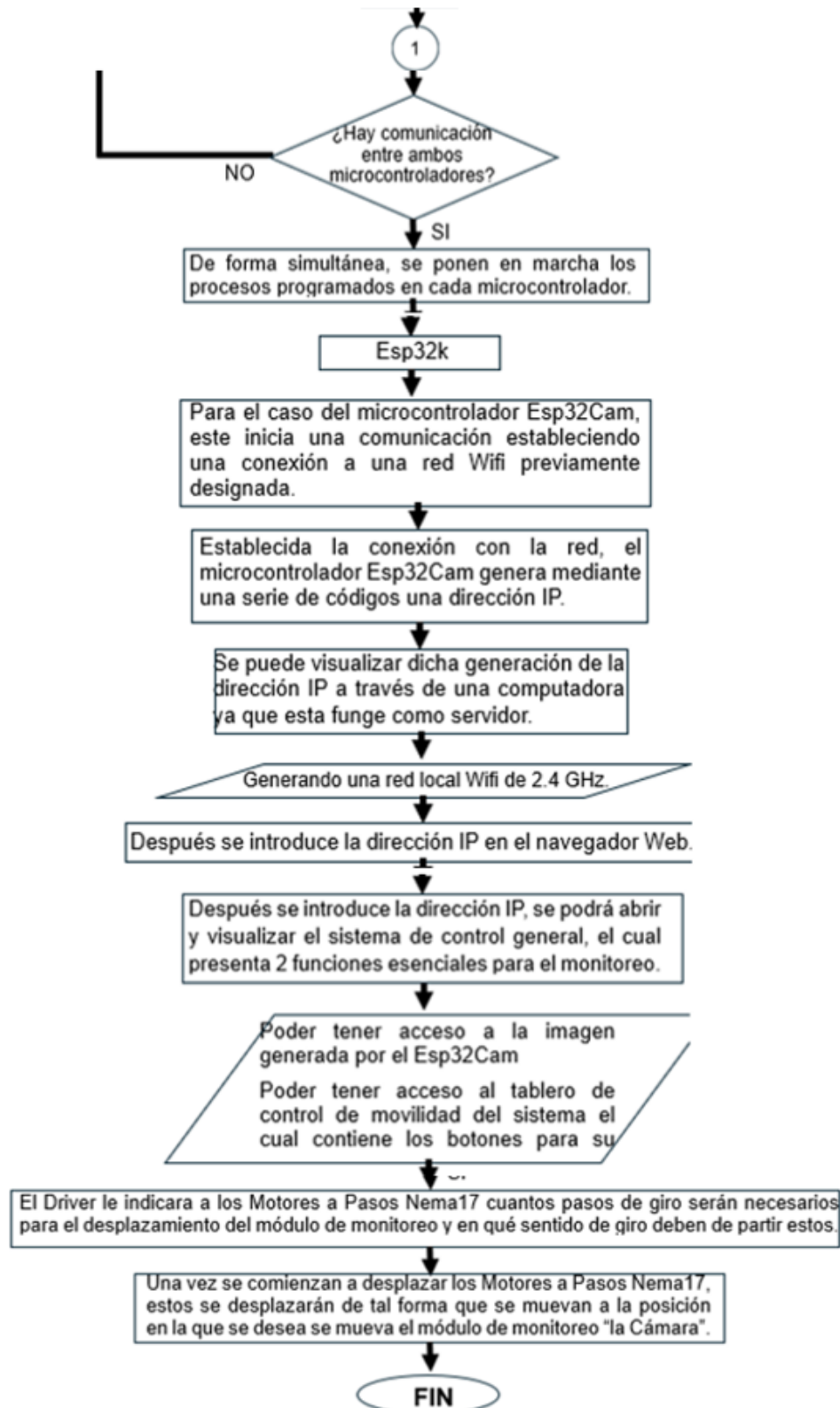


Figura 109. Diagrama de Flujo del Código de programación para el Control del Microcontrolador Esp32 Cam
Fuente: Elaboración Propia.

3.5 Diseño, Desarrollo y Fabricación de unas Placas de Circuito Impreso o “PCB”

Con el Sistema de Monitoreo funcionando ahora con la integración y acoplamiento en conjunto de los 2 Microcontroladores “Arduino UNO R4 Wifi” y “Esp32 Cam”, con las conexiones echas y el correcto funcionamiento de los Motores paso a paso mediante la implementación de 2 Drivers de control, y con el “Dashboard” creado, desarrollado y listo en la plataforma de Arduino Cloud se procedió con la elaboración de placas de circuito impreso para la interconexión de estos componentes, con la finalidad de obtener un mejor rendimiento en torno a la eficiencia y la optimización de espacio y exceso de cableado, así como una mejor comunicación entre ambos microcontroladores.

Para la elaboración de las placas de circuito impreso (PCB), se diseñaron en un software llamado Fritzing, el cual fue proporcionado y mostrado su funcionamiento en el diplomado del internet de las cosas impartido por código IoT. Eston con el propósito de realizar pruebas de interconexión con los materiales propuestos (conector y cable Rj45), después de realizar las conexiones en los diseños de las placas se exportaron en formato png a tamaño real, los cuales se invirtieron de sentido (espejo) en adobe ilustrator, para así poder imprimirlos en una impresora0020láser en papel transfer, con el método de planchado se transfirió la tinta de las impresiones a cada una de las dos placas fenólicas.

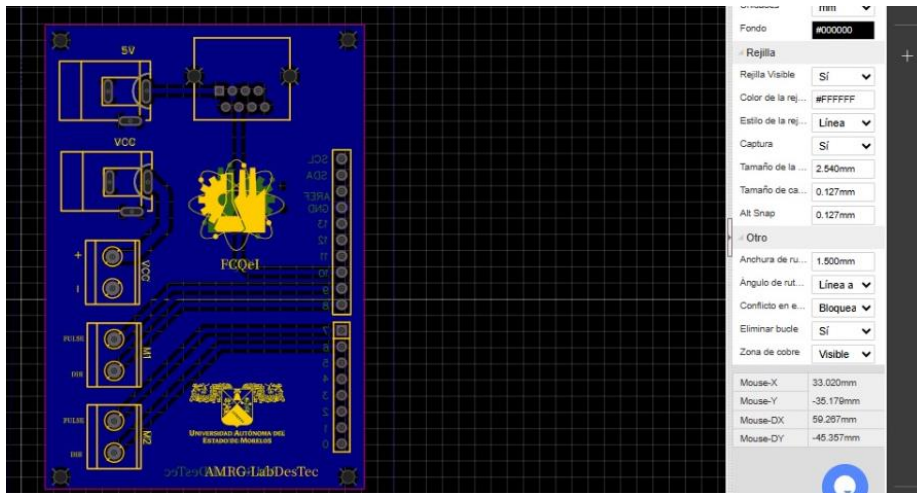


Figura 110: PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo Fuente: Elaboración Propia.

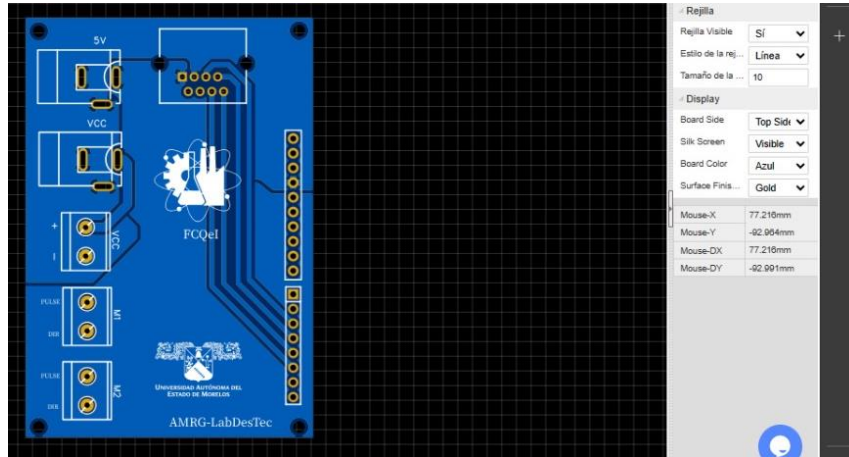


Figura 111: PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo
Fuente: Elaboración Propia.

Una vez obtenidas las placas con el diseño plasmado se retocaron detalles para después sumergirlas en ácido férrico activado con una cuarta parte de agua caliente, dejando únicamente el cobre que se encuentra debajo de la tinta del diseño, después, se realizaron las perforaciones necesarias para soldar los componentes (Jack Rj45 PCB, headers hembra para colocar el Esp32Cam, Jack atornillable doble, headers macho, Jack de alimentación).

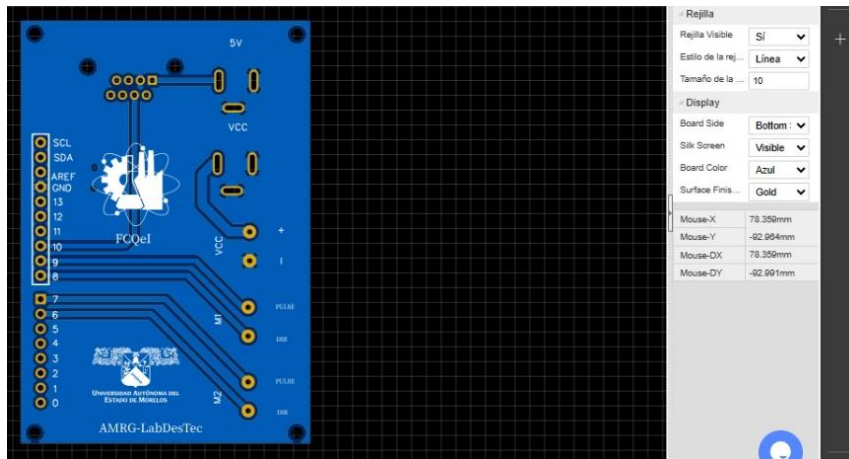


Figura 112: PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo
Fuente: Elaboración Propia.

Con las pruebas ya realizadas se corroboró el funcionamiento de estas era muy deficiente, lo cual llevó al rediseño para lograr una mejor calidad de placa, fue necesario cambiar de software de diseño de placas PCB para obtener mayor precisión, además de obtener herramientas de diseño necesarias para su elaboración. Sin embargo, el software Fritzing no contaba con las hermanitas

necesarias para la realización de las placas, por lo cual se realizaron los diseños en software “easyEDA”, con la ventaja de que no necesariamente debe de ser descargado y puede utilizarse en línea.

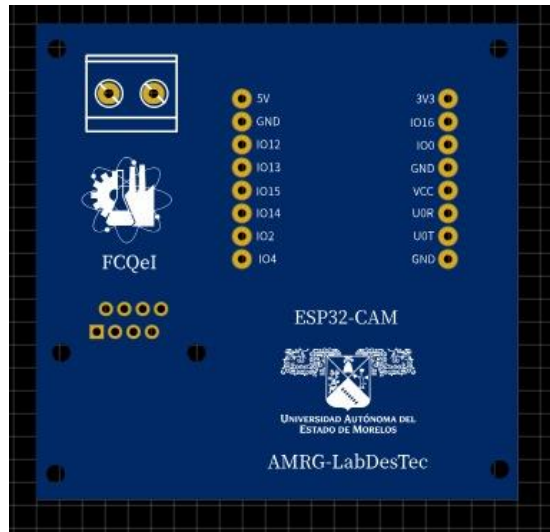


Figura 113: PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo
Fuente: Elaboración Propia.

La implementación de este software permitió editar aspectos de manera más precisa como lo son el tamaño de la placa, el grosor de la pista y libertad de trazado de esta, numero de capas, así como la personalización de la serigrafía para identificación de la placa.

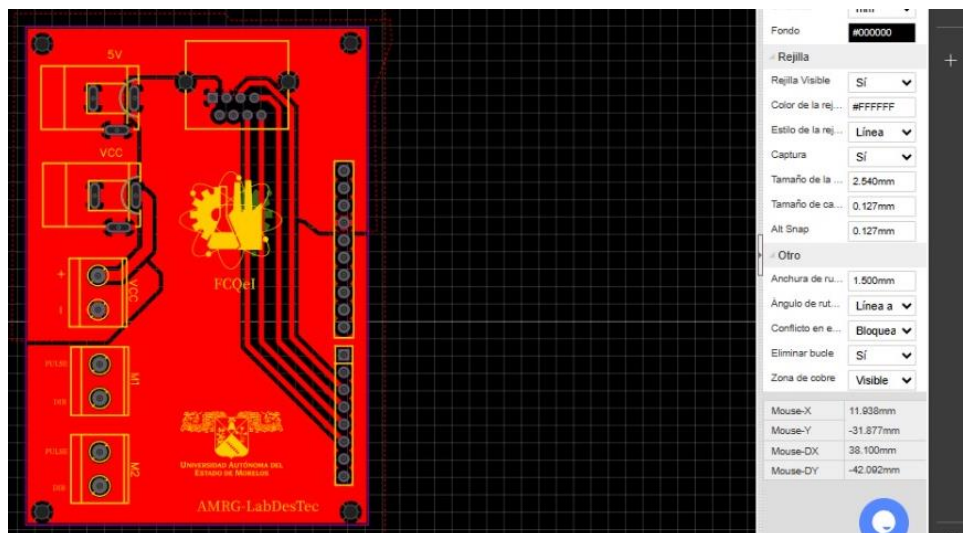


Figura 114: PCB “Placa de Circuito Impreso” empelada en el Modulo de Monitoreo
Fuente: Elaboración Propia.

Otro aspecto importante fue la cantidad de librerías elaboradas que contaban con el diseño y separación y tamaño de las pistas de los componentes, así como la calidad y accesibilidad del entorno para elaborar los diagramas esquemáticos, el mismo entorno cuenta con la posibilidad de enviar a imprimir las placas PCB directamente a la fábrica ubicada en China por un bajo costo por la cantidad mínima de 5 piezas por diseño, el archivo (GERBER) se exporta para posteriormente subirlo a la página en línea para hacer el pedido (JLCPcb), en el cual para estas dos placas tardó 15 días desde que se realizó el pedido, al llegar las placas se procedió al montaje de los componentes para probar su correcto funcionamiento, el cual se comportó de la manera esperada.

- **Diagrama de Conexión para Diseño de una Placa de Inserción de un Arduino UNO R4 Wifi**

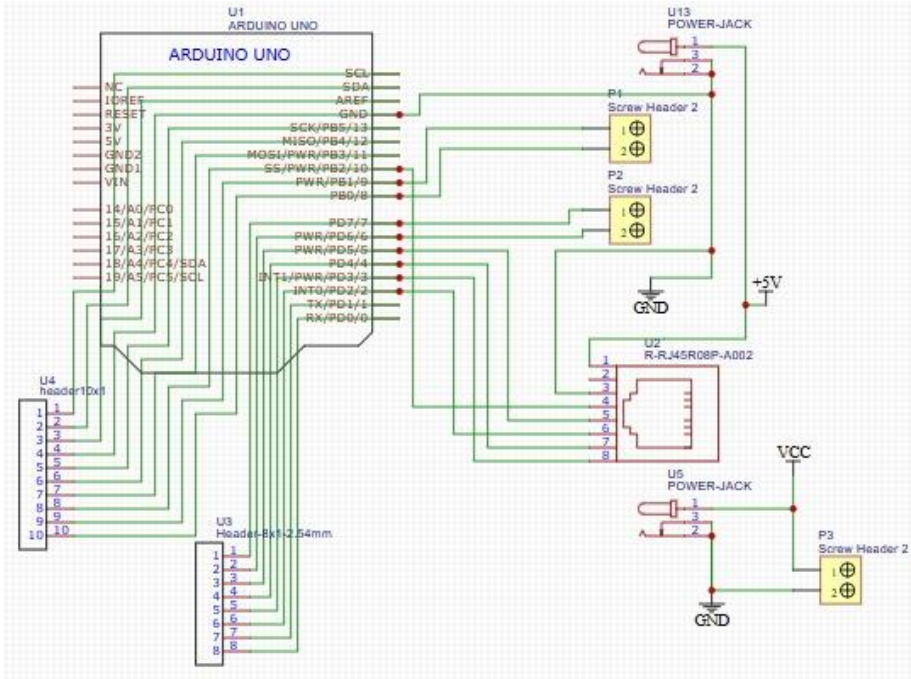


Figura 115: Diagrama de Conexión para el diseño de una Placa para el Arduino UNO R4 Wifi. Fuente: Elaboración Propia.

- **Conexión de la Placa del Módulo de Monitoreo Esp32Cam**

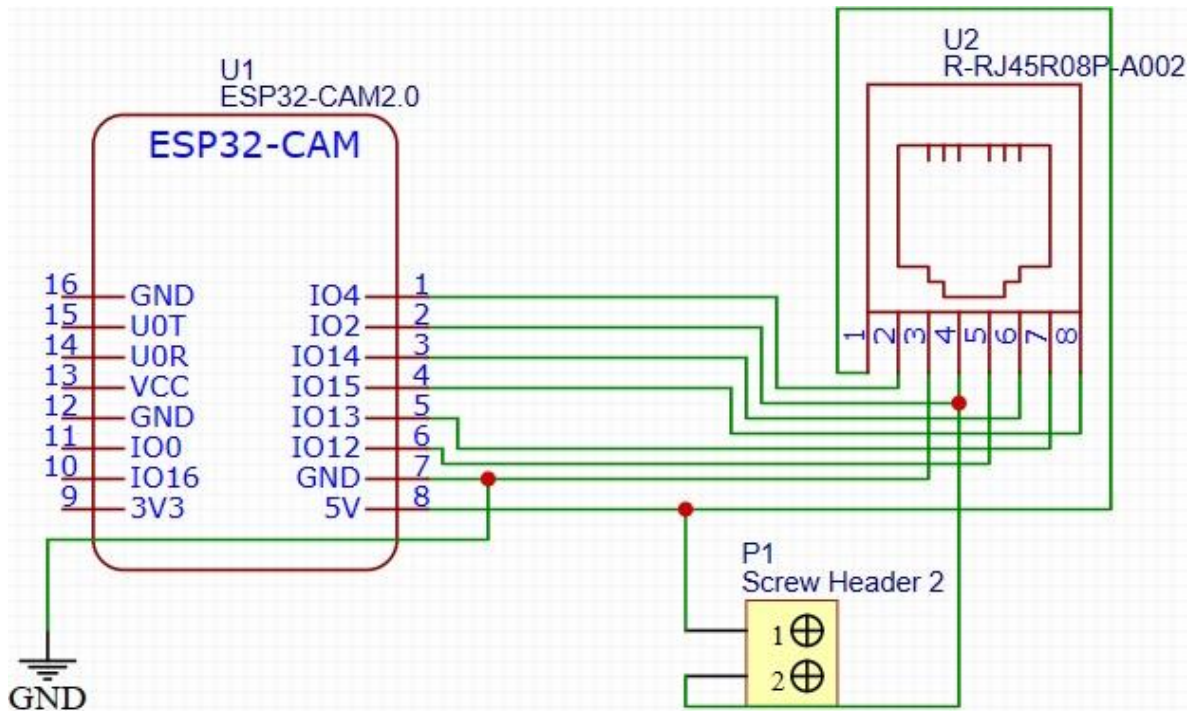


Figura 116: Diagrama de Conexión para el diseño de una Placa para el Esp32 Cam.
Fuente: Elaboración Propia.

Final mente con las placas ya terminadas se procedió con el diseño de un Código de programación que permitiese la integración de todos y pudiese permitirse así el control general de todo el Sistema de Monitoreo.

3.5.1 Diseño de un Código para el control General del Sistema de Monitoreo.

Es indispensable argumentar que para la elaboración de este Código se empleó de los 3 códigos anteriores, con la finalidad de juntarlos, realizando un compendio de estos, para final mente o obtener un Código de programación con el cual se pudiera controlar de forma eficiente y precisa todo el Sistema de Monitoreo.


```

3 #include "frames.h" //tab de capas de matriz
4
5 ArduinoLEDMatrix matrix;
6
7 int D;//delay
8 //-----declaracion de entradas digitales
9 bool arriba = 2;
10 bool abajo = 3;
11 bool derecha = 4;
12 bool izquierda = 5;
13 //-----declaracion de pulsos de control de motor
14 const uint8_t boton = 10; //boton de final de recorrido
15 const uint8_t PulsoX = 9; //Pulso X
16 const uint8_t DireccionX = 8; //Direccion X
17 const uint8_t DireccionY = 7; //Direccion Y
18 const uint8_t PulsoY = 6; //Pulso Y
19

```

Figura 117: Código de programación para el Control general del Sistema de Monitoreo
Fuente: Elaboración Propia.

Declaración de las funciones, de los pines a utilizar, declaraciones de función de movimiento.

```

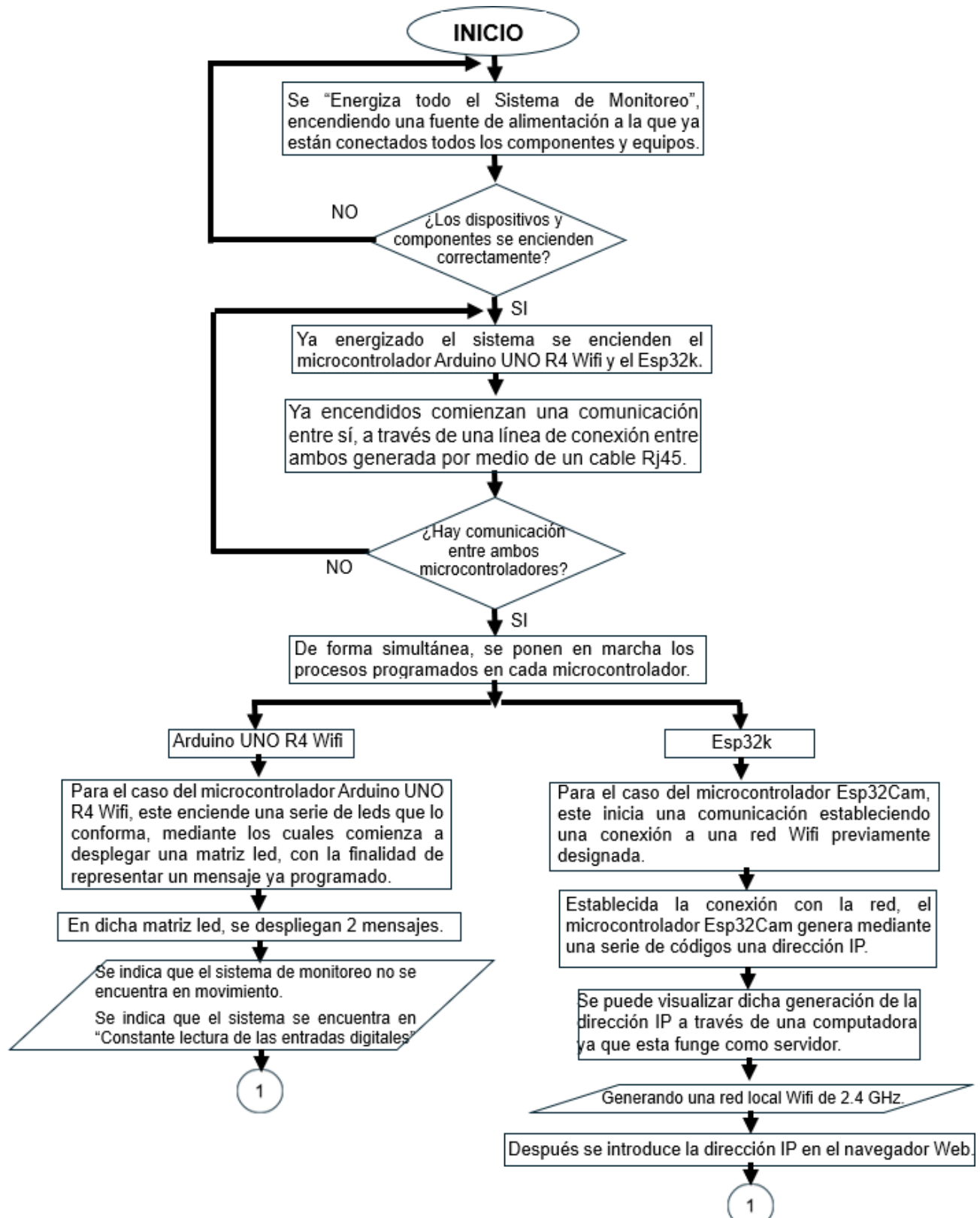
21 //Declaracion de funciones
22 void up() {
23     matrix.loadFrame(arr);
24     digitalWrite(DireccionY,HIGH);
25     delayMicroseconds(10);
26     for(int j=0;j<2000;j++){
27         digitalWrite(PulsoY,HIGH);
28         delayMicroseconds(300);
29         digitalWrite(PulsoY,LOW);
30         if (digitalRead(boton)==true){
31             matrix.loadFrame(aba);
32             digitalWrite(DireccionY,HIGH);
33         }
34     }
35 }

```

Figura 118: Código de programación para el Control general del Sistema de Monitoreo
Fuente: Elaboración Propia.

Declaración de la función que le mandan las señales al driver para genera el movimiento en la dirección hacia arriba con la cantidad de pasos que realiza el motor de paso a paso, así como el sentido de giro con la condición de que si llega al final de carrera la entrada digital del botón esta activada este v a cambiar la dirección para completar los pasos restantes. No obstante, es indispensable argumentar que el primer código, funciona a través de las acciones que se generan en el Dashboard las cuales manda a llamar a las funciones de acciones de movimientos de los motores paso a paso. Mientras que en el segundo funciona con lecturas digitales que son enviadas por el Esp32 Cam a través de Dashboard con condicional de selección en función a las señales digitales enviadas, para indicar la función de accionamiento, es decir hacia donde se mueve.

3.5.2 Diagrama De Flujo del Proceso de Funcionamiento del Sistema de Monitoreo



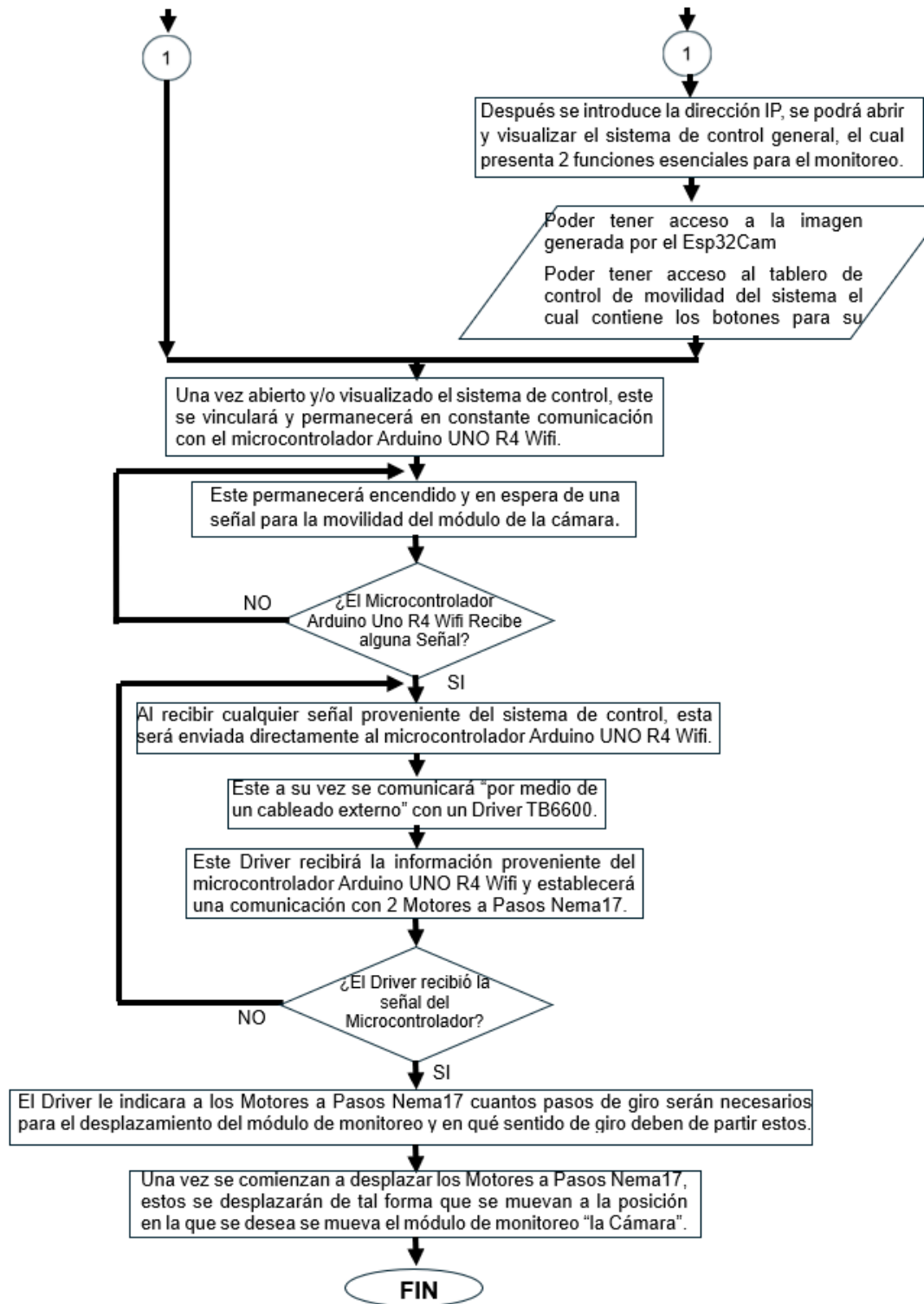


Figura 119: Diagrama de Flujo del Código de programación para el Control general del Sistema de Monitoreo

Fuente: Elaboración Propia.

En el siguiente diagrama de flujo (Figura 23), se pretende entender la cronología de cada paso u proceso a realizar del diseño, desarrollo y modelado del proyecto, con el fin de servir como una guía para su entendimiento.

3.5.3 Descripción del Proceso de Funcionamiento del Sistema de Monitoreo

Como primer paso se “Energiza todo el Sistema de Monitoreo”, encendiendo una fuente de alimentación modelo “ “, en la cual ya están conectados todos los componentes y equipos, que deben ser energizados para el Sistema de Monitoreo, como dato significativo se debe hacer mención que la fuente de alimentación arroja un voltaje de 24v la cual alimenta directamente el Arduino UNO R4.



Figura 120: Fuente de Alimentación Modelo “S-360-24”.
Fuente: Elaboración Propia 2024.

Al momento de ser energizado se enciende el microcontrolador Arduino UNO R4 Wifi y el microcontrolador Esp32Cam

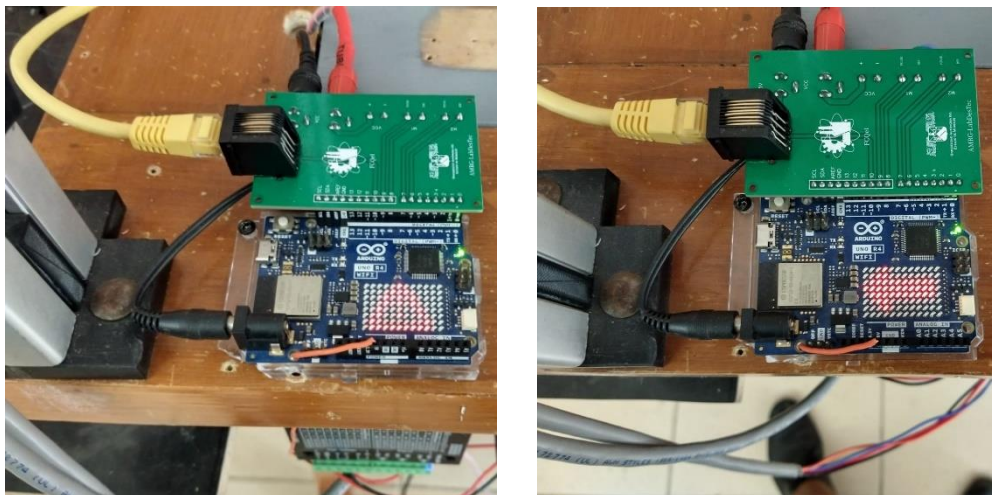


Figura 121: Arranque del Microcontrolador Arduino UNO R4 Wifi y el Microcontrolador Esp32Cam.
Fuente: Elaboración Propia 2024.

Ya encendidos, estos microcontroladores comienzan una comunicación entre sí, a través de una línea de conexión entre ambos generada por medio de un cable Rj45.

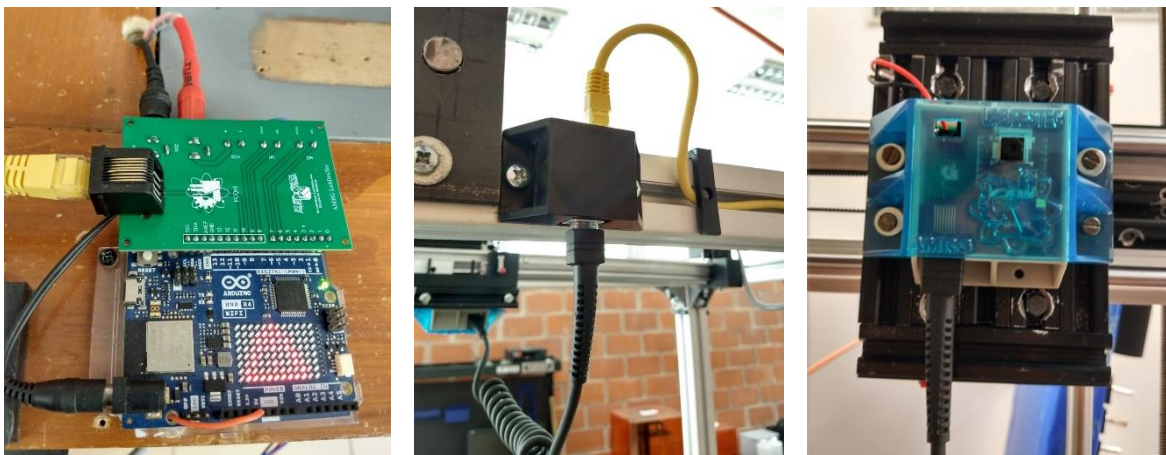


Figura 122: Cableado para la Comunicación del Microcontrolador Arduino UNO R4 Wifi y el Microcontrolador Esp32Cam, mediante un cable RJ45.
Fuente: Elaboración Propia 2024.

De forma inmediata y casi simultánea, se ponen en marcha los procesos programados en cada microcontrolador “Arduino UNO R4 Wifi y Esp32Cam”.

```
control_camara_esclavo Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

control_camara_esclavo frames.h saludo.h
1 #include "Arduino_LED_Matrix.h"
2 #include "saludo.h" //tab de capas de matriz
3 #include "frames.h" //tab de capas de matriz
4
5 ArduinoLEDMatrix matrix;
6
7 int Dr;//delay
8 //-----declaracion de entradas digitales
9 bool arriba = 2;
10 bool abajo = 3;
11 bool derecha = 4;
12 bool izquierda = 5;
13 //-----declaracion de pulsos de control de motor
14 const uint8_t boton = 10; //boton de final de recorrido
15 const uint8_t PulsoX = 9; //Pulso X
16 const uint8_t DireccionX = 8; //Direccion X
17 const uint8_t DireccionY = 7; //Direccion Y
18 const uint8_t PulsoY = 6; //Pulso Y
19
20
21 //Declaracion de funciones
22 void up(){
23     matrix.loadFrame(arr);
24     digitalWrite(DireccionY,HIGH);
25     delayMicroseconds(10);
26     for(int i=0;i<2000;i++){
```

Figura 123: Código puesto en Marcha del Microcontrolador Arduino UNO R4 Wifi.
Fuente: Elaboración Propia 2024.


```
camaraLabMaster Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
camaraLabMaster
1 #include "esp_wifi.h"
2 #include "esp_camera.h"
3 #include <WiFi.h>
4 #include "esp_timer.h"
5 #include "img_converters.h"
6 #include "Arduino.h"
7 #include "fb_gfx.h"
8 #include "soc/soc.h" // disable brownout problems
9 #include "soc/rtc_cntl_reg.h" // disable brownout problems
10 #include "esp_http_server.h"
11
12 // Replace with your network credentials
13 const char *ssid = "IZZI-D4D2";
14 const char *password = "C863FC39D4D2";
15
16 #define PART_BOUNDARY "1234567890000000000000987654321"
17
18 #define CAMERA_MODEL_AI_THINKER
19
20 #define PWDN_GPIO_NUM 32
21 #define RESET_GPIO_NUM -1
22 #define XCLK_GPIO_NUM 0
23 #define SIOD_GPIO_NUM 26
24 #define SIOC_GPIO_NUM 27
25
Compilando programa...
```

Figura 124: Código puesto en Marcha del Microcontrolador Esp32Cam.
Fuente: Elaboración Propia 2024.

Para el caso del microcontrolador Arduino UNO R4 Wifi, este enciende una serie de leds que lo conforma, mediante los cuales comienza a desplegar una matriz led, con la finalidad de representar un mensaje ya programado.

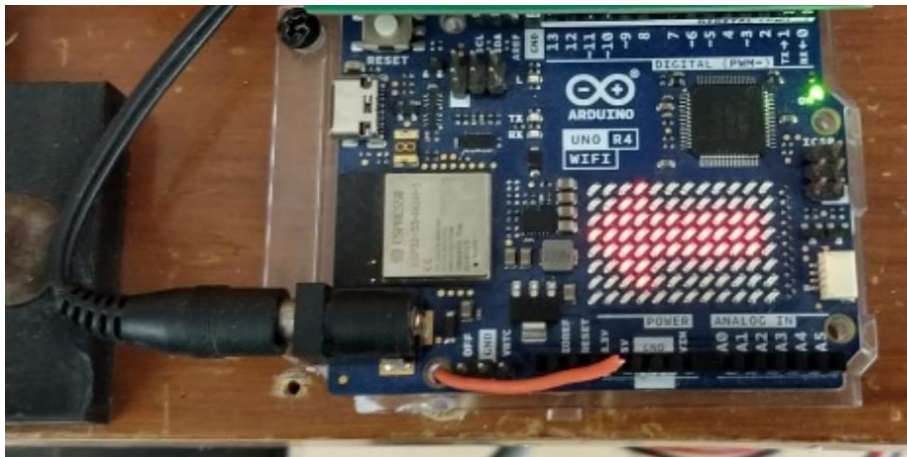


Figura 125: Microcontrolador Arduino UNO R4 Wifi.
Fuente: Elaboración Propia 2024.

Cabe mencionar que para esto el Microcontrolador Esp32Cam envía una señal y genera una IP en la que se abre un Dashboard, en el cual se le da la indicación, es decir cuando este se presiona se envía un pulso que regresa al Microcontrolador Esp32Cam el cual mediante la comunicación de un cable Rj45, envía la indicación de su desplazamiento al Microcontrolador Arduino UNO R4, este activa su protocolo de funcionamiento anterior mente ya programado y controla la velocidad y la dirección del movimiento de los Motores a pasos mediante 2 Drivers.

En dicha matriz led, se encuentran programados una serie de despliegues de mensajes los cuales sirven como indicadores del proceso que está realizando el Microcontrolador, estos son los siguientes:

- El primero indica que el sistema de monitoreo se encuentra encendido y en funcionamiento.



Figura 126: Microcontrolador Arduino UNO R4 Wifi “Encendido”.
Fuente: Elaboración Propia 2024.

- El Segundo es un indicador de que el sistema de monitoreo no se encuentra en movimiento, es decir que este no ha recibido ninguna señal perteneciente al Microcontrolador Esp32Cam.

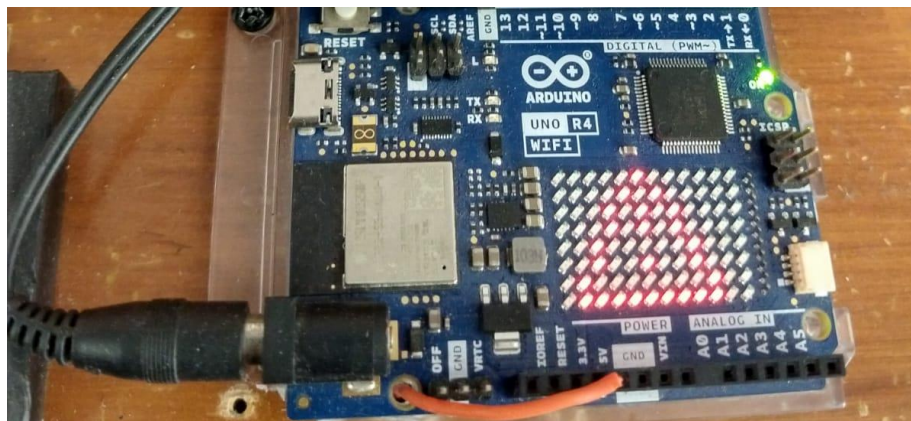


Figura 127: Microcontrolador Arduino UNO R4 Wifi “Encendido, pero sin Recibir una Señal”.
Fuente: Elaboración Propia 2024.

- El tercero indica que el sistema se encuentra en “Constante lectura de las entradas digitales”, es decir en espera de alguna instrucción de mando del microcontrolador Esp32Cam.

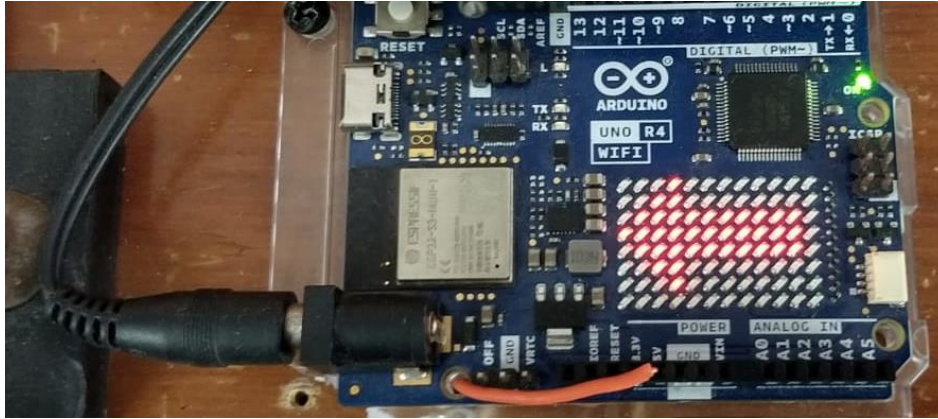


Figura 128: Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.
Fuente: Elaboración Propia 2024.

Para el caso del microcontrolador Esp32Cam, este inicia una comunicación estableciendo una conexión a una red Wifi previamente designada.

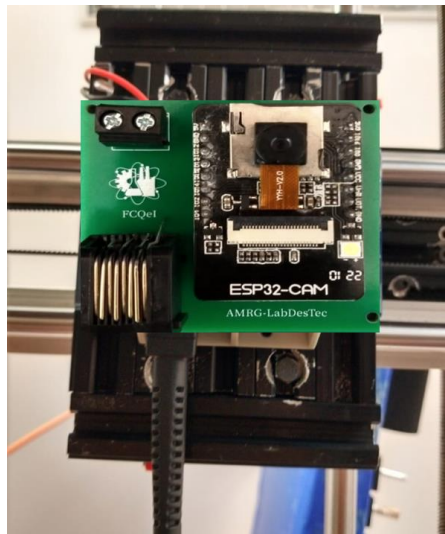


Figura 129: Microcontrolador Esp32 Cam “Encendido Recibiendo una Señal”.
Fuente: Elaboración Propia 2024.

Una vez establecida la comunicación y conexión con la red, el microcontrolador Esp32Cam genera mediante una serie de códigos una dirección IP. Cabe mencionar que la red wifi a la cual se conecta la Esp32Cam es generada a través de la computadora del Laboratorio debido a problemas con una red abierta externa.

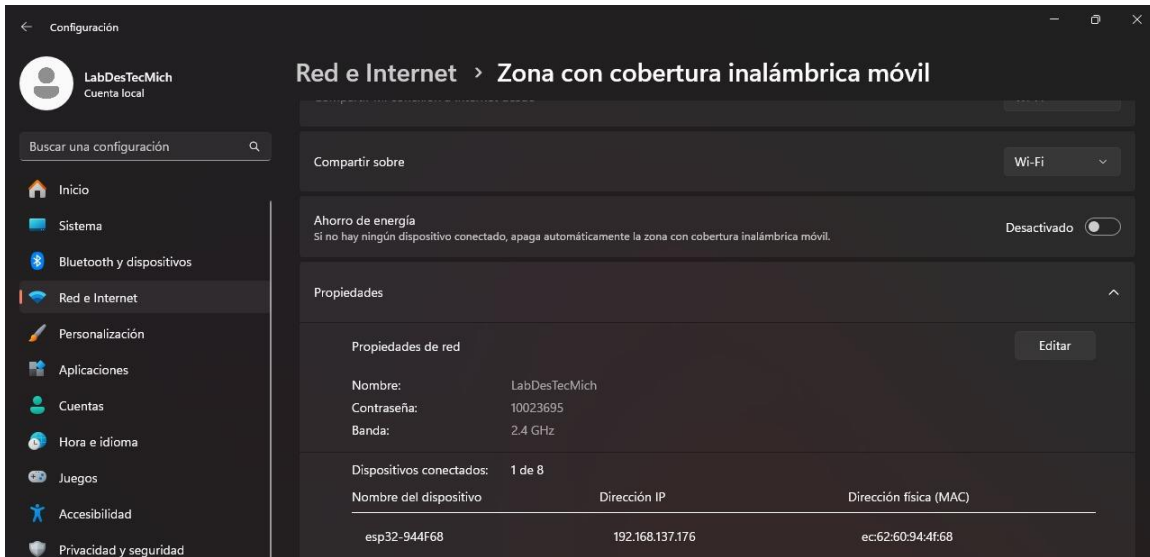


Figura 130: Comunicación y conexión con la red y los Microcontroladores.
Fuente: Elaboración Propia 2024.

De forma inmediata debido a la exitosa comunicación y conexión con la red, se puede visualizar dicha generación de la dirección IP a través de una computadora ya que esta funge como servidor. Cabe mencionar que este servidor en la configuración de la zona de cobertura móvil genera una red local Wifi de 2.4 GHz. Después se introduce la dirección IP en el navegador Web, ya que es esencial para el control del sistema de monitoreo.

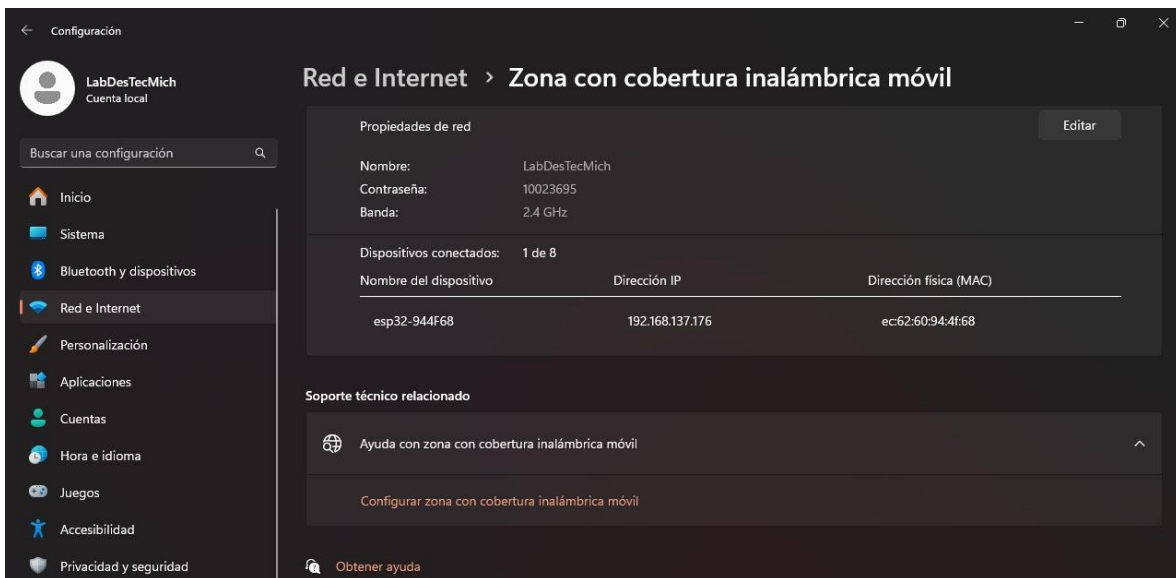


Figura 131: Comunicación y conexión con la red y los Microcontroladores.
Fuente: Elaboración Propia 2024.

Con la dirección IP introducida en el navegador web se podrá abrir y visualizar el sistema de control general, el cual presenta 2 funciones esenciales para el monitoreo, estas son:

- La primero es el poder tener acceso a la imagen generada por el Esp32Cam



Figura 132: Imagen generada por el Dashboard para el control de los motores a pasos.
Fuente: Elaboración Propia 2024.

- La segunda es el poder tener acceso al tablero de control de movilidad del sistema el cual contiene los botones (arriba, abajo, derecha, izquierda y control del encendido y apagado de una luz)



Figura 133: Imagen generada por el Dashboard para la visualización del laboratorio.
Fuente: Elaboración Propia 2024.

Una vez abierto y/o visualizado el sistema de control, este se vinculará y permanecerá en constante comunicación con el microcontrolador Arduino UNO R4 Wifi, el cual permanecerá encendido y en espera de una señal para la movilidad del módulo de la cámara.

FCQeI

Laboratorio de Desarrollos Tecnologicos

Sistema de monitoreo para laboratorio remoto

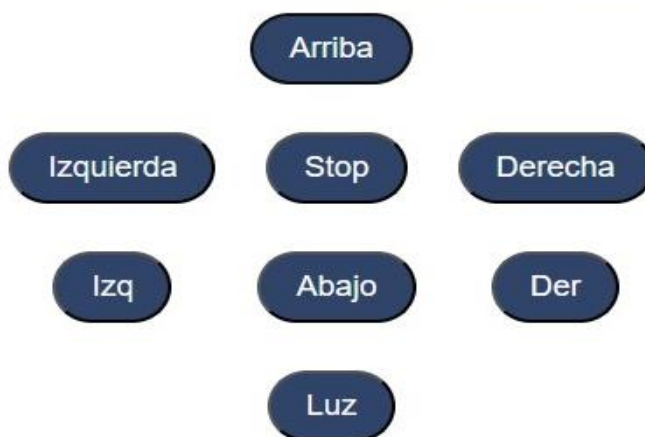


Figura 134: Imagen generada por el Dashboard para la visualización del laboratorio.
Fuente: Elaboración Propia 2024.

Al recibir cualquier señal proveniente del sistema de control para el desplazamiento del módulo de monitoreo, esta será enviada directamente al microcontrolador Arduino UNO R4 Wifi, el cual a su vez se comunicará “Esto por medio de un cableado externo” con un Driver TB6600.

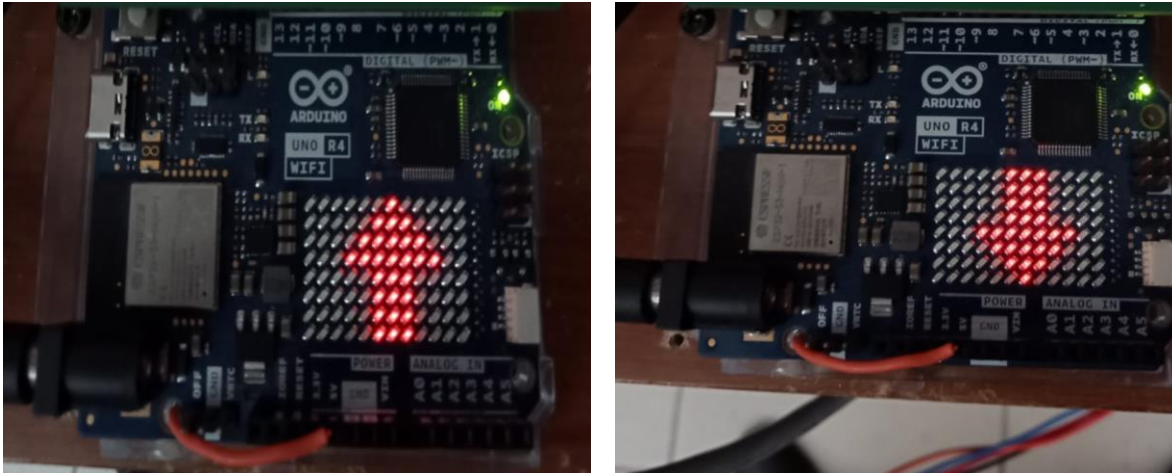


Figura 135: Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.
Fuente: Elaboración Propia 2024.

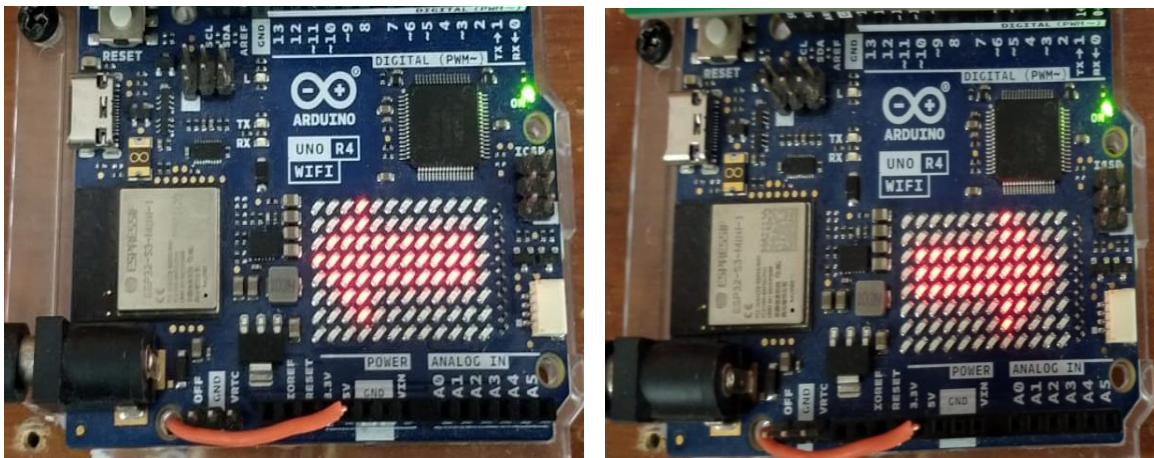


Figura 136: Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.
Fuente: Elaboración Propia 2024.

En esta secuencia de imágenes se puede apreciar la matriz led del Microcontrolador Arduino UNO R4 Wifi, con sus respectivos indicadores de dirección, “Adelante, atrás, izquierda y derecha”.

Este Driver recibirá la información proveniente del microcontrolador Arduino UNO R4 Wifi y establecerá una comunicación con 2 Motores a Pasos Nema17.

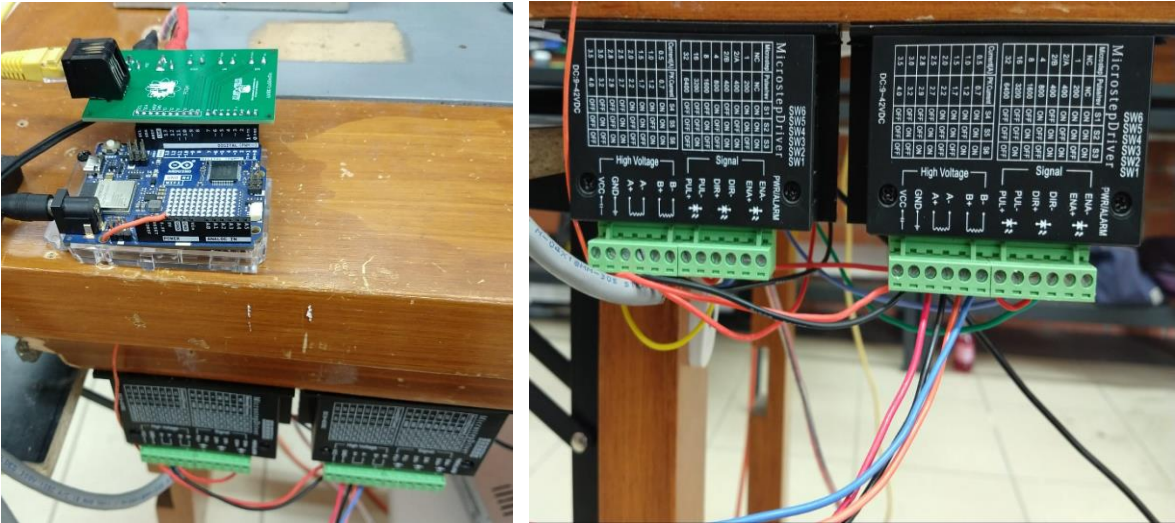


Figura 137: Microcontrolador Arduino UNO R4 Wifi y los Drivers para el control de los Motores.
Fuente: Elaboración Propia 2024.

Una vez establecida la comunicación el Driver, le indicara a los Motores a Pasos Nema17 cuantos pasos de giro serán necesarios para el desplazamiento del módulo de monitoreo y en qué sentido de giro deben de partir estos.

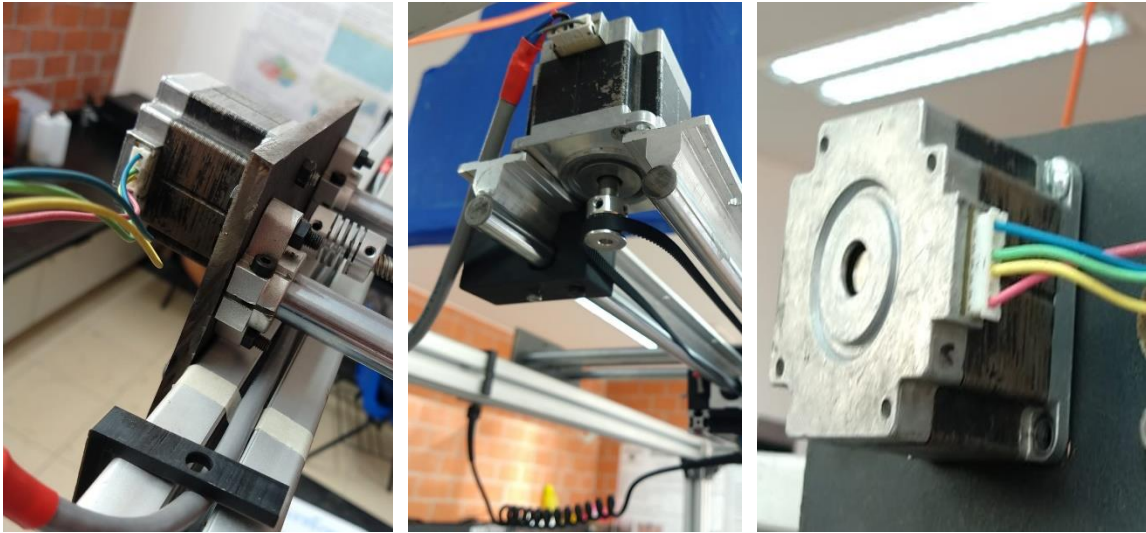


Figura 138: Motores a pasos desplazando por los rieles.
Fuente: Elaboración Propia 2024.

Una vez se comienzan a desplazar los Motores a Pasos Nema17, estos se desplazarán de tal forma que se muevan hacia la posición en la se le solicite desde el Dashboard, desplazando el módulo de monitoreo es decir “la Cámara”.

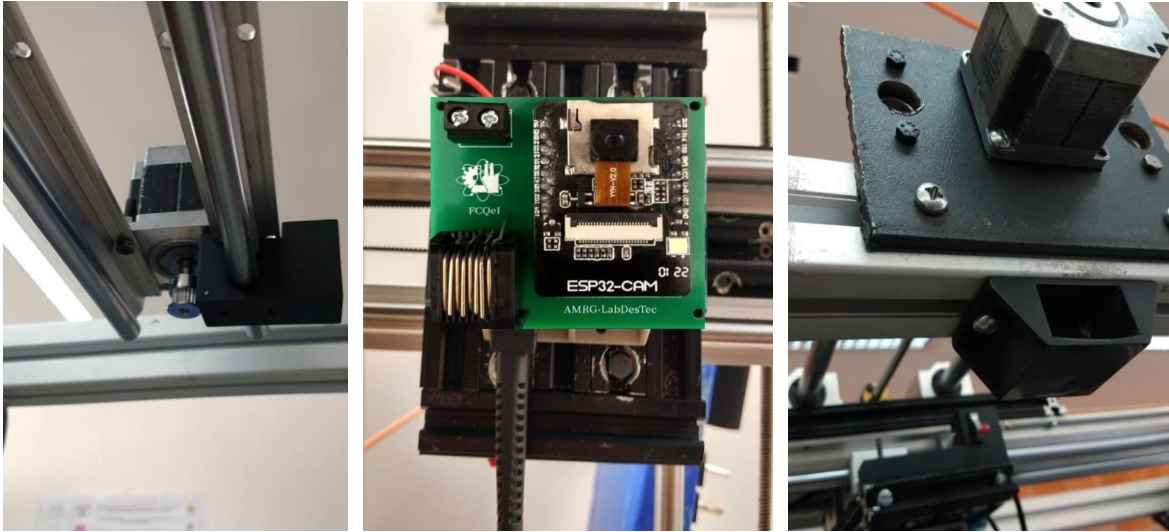


Figura 139: Movilización y desplazamiento del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024.

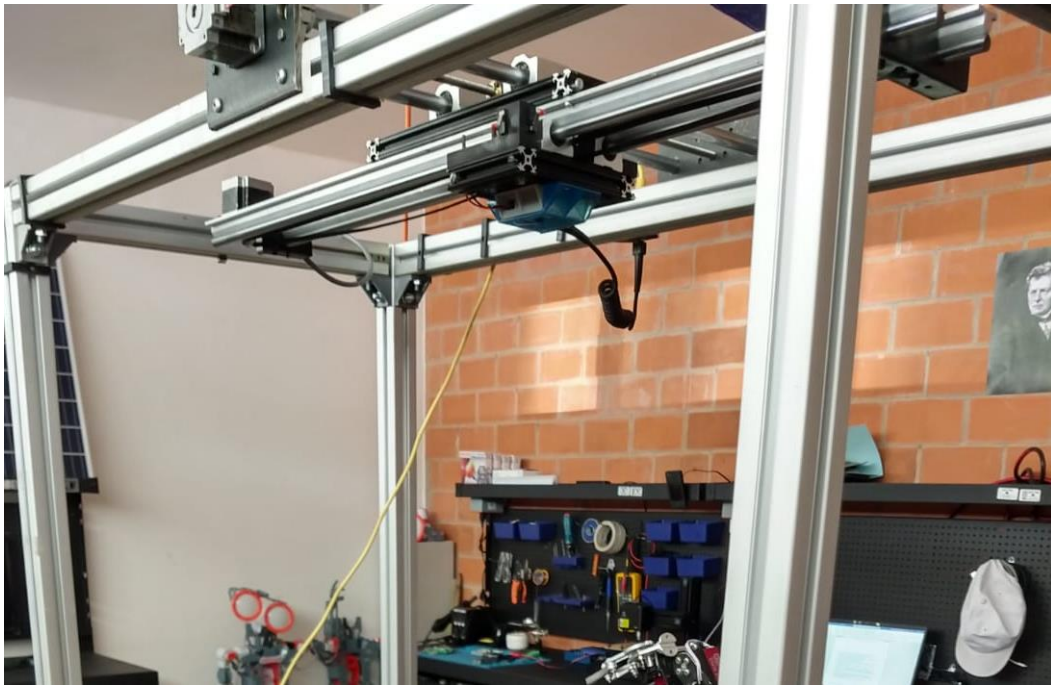


Figura 140: Desplazamiento del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024.

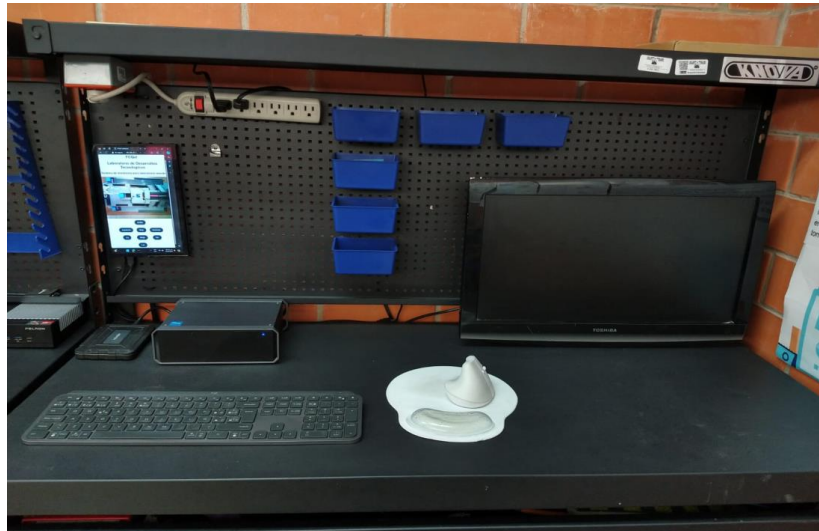


Figura 141: Modulo de control para el Control del Modulo de Monitoreo
Fuente: Elaboración Propia 2024.

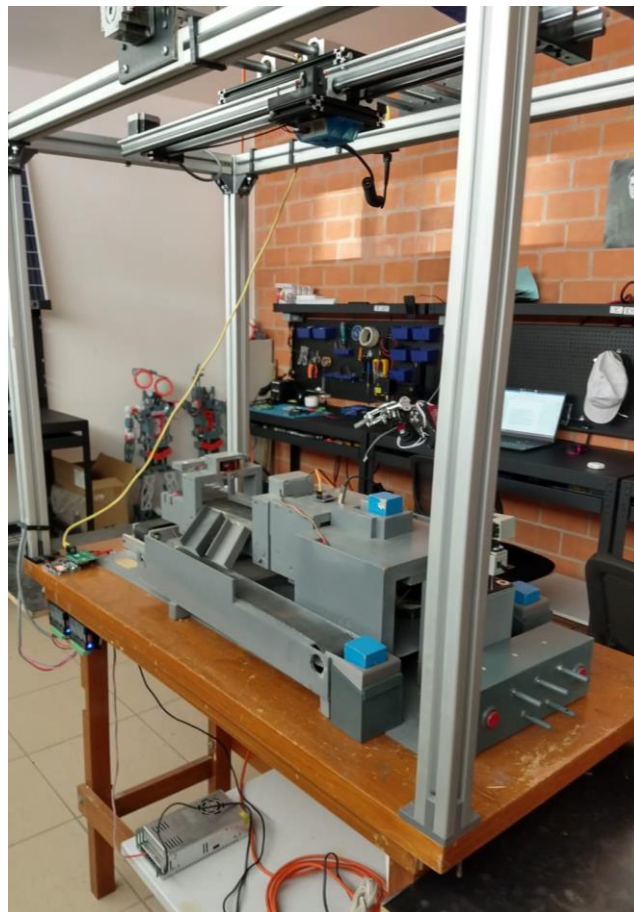


Figura 142: Sistema de Monitoreo Funcionando
Fuente: Elaboración Propia 2024.

3.6 Análisis General de la Estructura del Sistema de Monitoreo

Después de realizar las pruebas de funcionalidad de todo el sistema de Monitoreo, las cuales abarcaron las velocidades de desplazamiento máximas y mínimas de la velocidad en la que se desplazaría el Módulo de Monitoreo, e identificar cual era la mejor programación para que este realizase su funcionamiento de la manera más rápida y eficiente, se obtuvo como resultado una movilidad síncrona del mismo. No obstante, al observar detenidamente el proceder de los distintos tipos de movimientos que este, se pudo detectar que todo el prototipo ya funcionando, sufría de algunos fallos en su funcionamiento, por tal motivo, se realizó un análisis general de todo el prototipo con el objetivo de determinar la ubicación exacta del punto en el que se gestaba el error y así idear una forma de resolverlo, considerando la funcionalidad y el esfuerzo mecánico que se pudieran efectuar sobre ellas. Después de realizar dicho análisis se determinaron los puntos más sensibles, poco eficientes, más recurrentes a fallas o errores y poco estéticos, dichas partes de la estructura del Sistema de Monitoreo se enlistaron de la siguiente manera:

1. Problema en la sujeción del cableado que comunica los dispositivos
2. Problema con la sujeción e inserción del cable RJ45
3. Problema en el final del riel de desplazamiento Delantero y Trasero
4. Problema en el final del eje de la banda des desplazamiento
5. Problema en la detención del Modulo desplazado
6. Problema con la fijación del Microcontrolador Esp32 Cam

Estos puntos para modificar, se planteó fueran reforzados por medio de impresiones en 3d, es decir diseñar piezas a la medida con la finalidad de ser insertadas en la estructura del modulo de Monitoreo y ayudar a si a desaparecer estas deficiencias en su eficiencia, por lo tanto, se procedió con la toma de medidas y un minucioso análisis sobre como diseñar una pieza, la forma, las características para el reforzamiento en dicho punto de quiebre. Sin embargo, teniendo en cuenta que cada uno de estos puntos a modificar deberán ser diseñados minuciosamente se

procedió con la implementación de una herramienta “Vernier” para la toma de medidas exactas, con la finalidad de evitar errores de impresión.

3.6.1 Determinación de las Problemáticas

1. Problema en la sujeción del cableado que comunica los dispositivos

Para este primer punto de problemática a modificar, se planteó la implementación, diseño y fabricación de una pieza capaz de sujetar el cableado que comunica los componentes que componen el Sistema de Monitoreo sin dañarlos, es por ello que se optó por la fabricación de la siguiente pieza en forma de “L”.

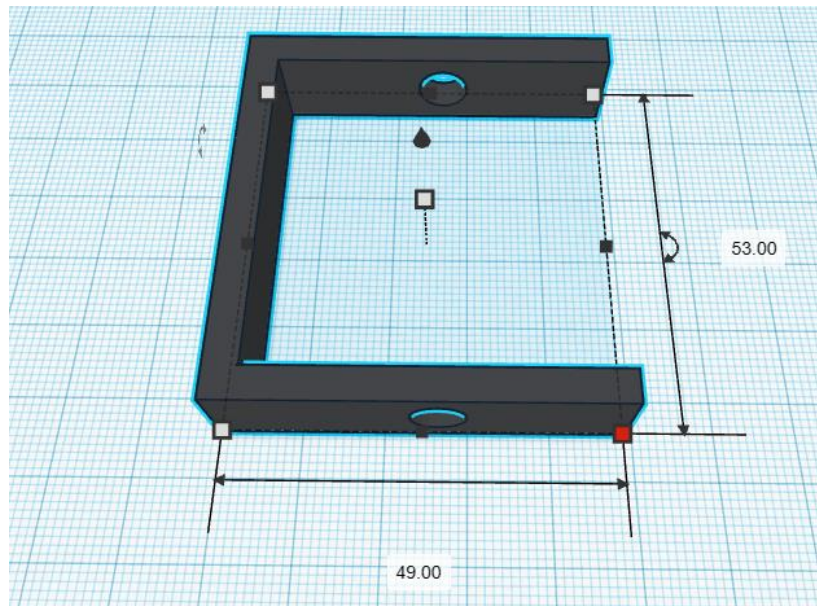


Figura 143: Soporte-Grapa en forma de “U” para la Fijación del Cableado.
Fuente: Elaboración Propia 2024.

Esta pieza en forma de “L” fungiría como, con la finalidad de hacer presión en sus extremidades, para posterior mente ser fijadas a alguno de los pilares de la estructura general del Modulo de Monitoreo, por lo que al introducir el cableado dentro de esta y posterior mente fijarla, se evitaría el daño a ese cableado y se le proporcionaría un estiramiento y rigidez al mismo cableado, evitando así curvas en el mismo y elongaciones, dándole una vista más estética al Sistema.

2. Problema con la sujeción e inserción del cable RJ45

Para el segundo punto de problemática a modificar, se planteó la implementación, diseño y fabricación de una pieza capaz de fungir como un soporte que se insertara en uno de los pilares superiores de la estructura del Modulo de Monitoreo y a su vez permitiera la conexión entre dos cables RJ45 con la suficiente presión para mantener esta unión, de tal forma que si este se estirase con fuerza, la pieza al momento de ser insertada a la estructura de aluminio y al momento de insertar en ella las 2 entradas del cable, este soporte sirviera como cople y a través de su fuerza y estructura, impidiera que dicho cableado se desconectara.

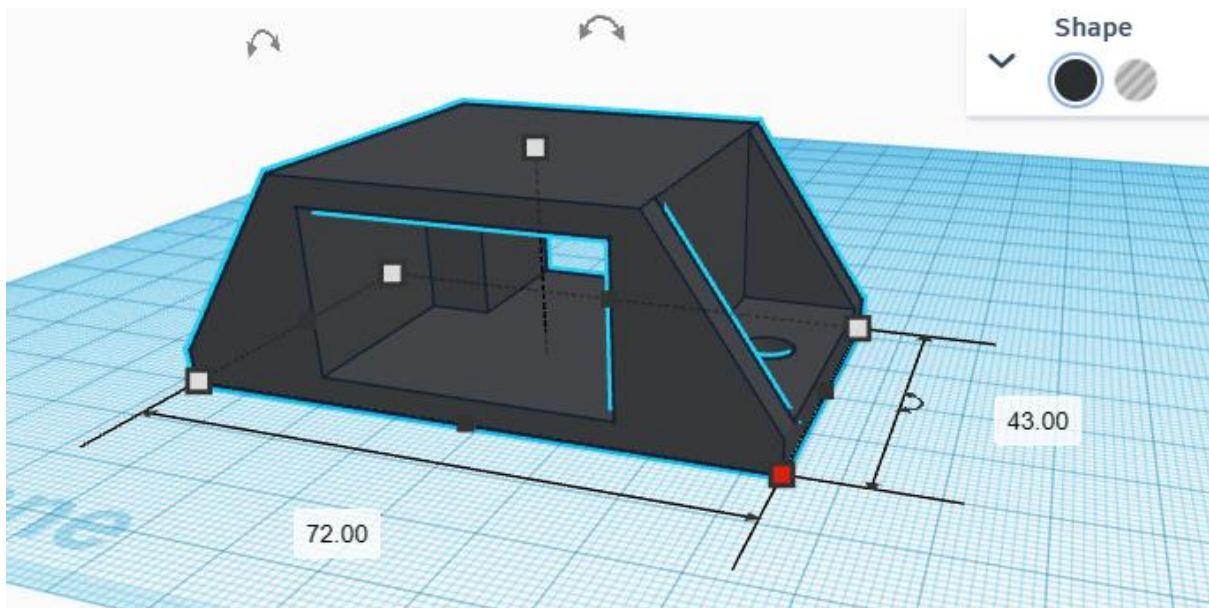


Figura 144: Soporte-Cople en forma de Pirámide para interconectar el cable RJ45.
Fuente: Elaboración Propia 2024.

Esta pieza en forma de “Pirámide” tiene la finalidad de sujetar el cableado y ser un cople al mismo tiempo, permitiendo la inserción de las dos entradas en sus dos extremos de el cableado RJ45, brindando así la posibilidad de estirar dicho cableado brindándole aún más estética a la estructura del módulo de Monitoreo.

3. Problema en el final del riel de desplazamiento Delantero y Trasero

Para el tercer punto de problemática a modificar, se planteó la implementación, diseño y fabricación de una pieza capaz de fungir como un tope final de carrera es decir, que esta pieza permitiera dos cosas, la primera fijarse al riel de

desplazamiento del Módulo de Monitoreo de tal forma que no se moviera aun que esta pieza fuese empujada y la segunda que fuese lo suficiente mente resistente y rígida, como para permanecer fija e intacta cuando el Módulo de Monitoreo la empujase evitando su avance y que este no saliera del riel.

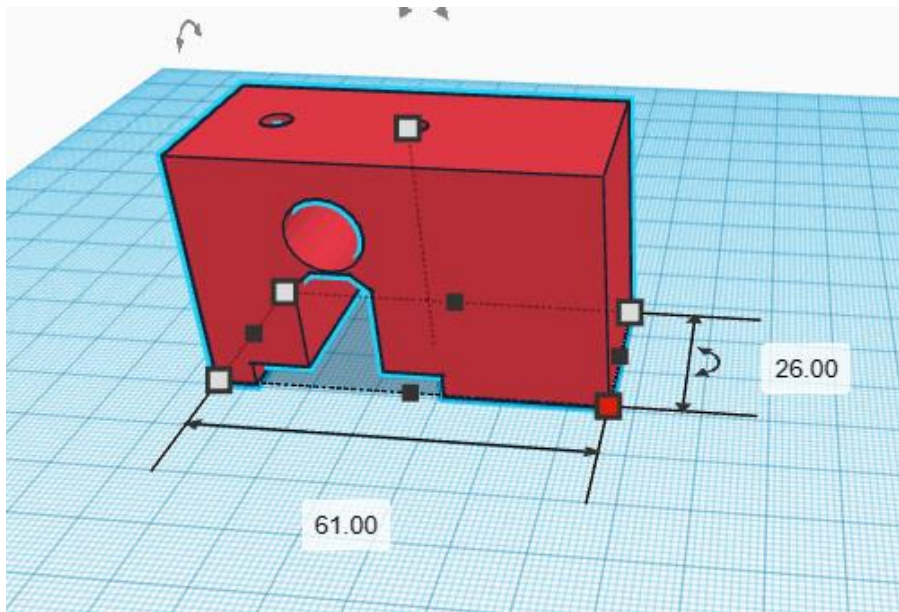


Figura 145: Soporte-Tope Final de Carrera.
Fuente: Elaboración Propia 2024.

Esta pieza en forma de “Rectángulo” tiene la finalidad de sujetarse del riel de desplazamiento de tal forma que al insertarlo en el este se mantenga fijo y lo suficientemente rígido para que cuando el módulo de Monitoreo lo toque y empuje querido avanzar o desplazarse aún más, este lo frene y le impida el paso, adicional a esto.

4. Problema en el final del eje de la banda des desplazamiento

Para el cuarto punto de problemática a modificar, se planteó la implementación, diseño y fabricación de una pieza capaz de fungir como un tope final de carrera y un eje de la Banda de desplazamiento, esto debido a que tanto la banda como los rieles de desplazamiento serian insertados en ella, dándole la separación necesaria a estos para un mejor funcionamiento de desplazamiento del Módulo de Monitoreo, además de permanecer como una pieza fija y rígida de tal forma que cuando el

módulo de Monitoreo lo toque y empuje querido avanzar o desplazarse aún más, este lo frene y le impida el paso, adicional a esto.

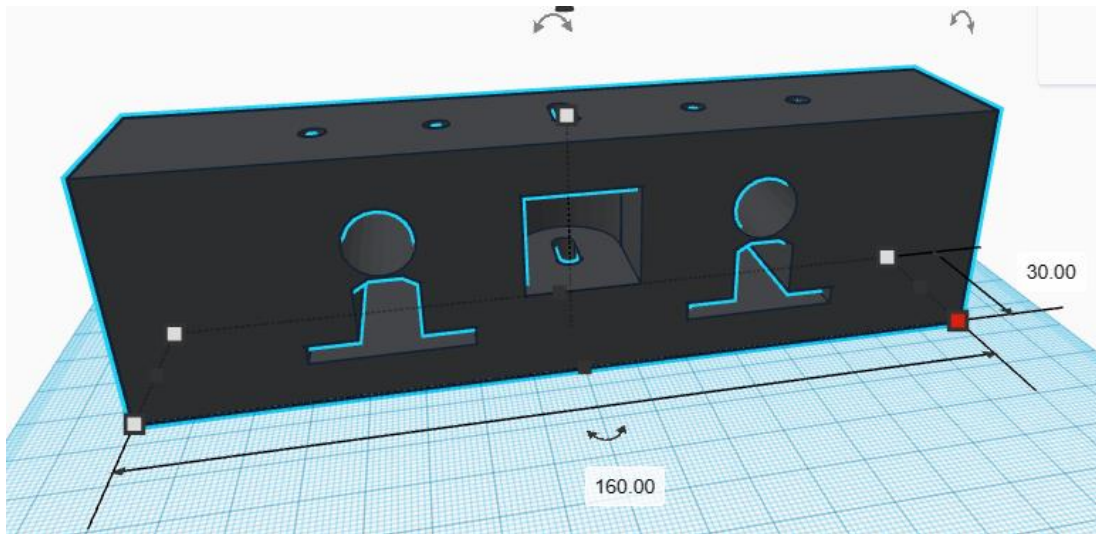


Figura 146: Soporte-Tope de Eje de Banda
Fuente: Elaboración Propia 2024.

Esta pieza en forma de “Rectángulo” tiene la finalidad de sujetarse ambos rieles de desplazamiento y la banda, de tal forma que al insertarlo en el este se mantenga fijo y lo suficientemente rígido para que cuando el módulo de Monitoreo lo toque y empuje querido avanzar o desplazarse aún más, este lo frene y le impida el paso, adicional a esto, así como también darles la separación necesaria a estos para un mejor funcionamiento de desplazamiento del Módulo de Monitoreo.

5. Problema en la detención del Módulo desplazado

Para el quinto punto de problemática a modificar, se planteó la implementación, diseño y fabricación de una pieza capaz de fungir como un tope final de carrera no obstante este tendría la capacidad de poder tener insertado en el un micro switch final de carrera y a su vez estar ensamblado y acoplado a la misma placa del Modulo de monitoreo, esto se decidió fuese así debido a que cuando el modulo hiciera contacto con el “Soporte-Tope Final de carrera” y con el “Soporte-Tope de eje de

Banda”, lo primero que hiciera contacto en estos fuese el micro switch Final de carrera, en tornas palabras, cuando el Modulo de Monitoreo avanzar y llegara casi al final del riel de desplazamiento, antes de que la placa del Modulo tocara el Tope, el Final de Carrera seria presionado y esto enviaría un pulso al microcontrolador, indicándole que ha llegado el punto máximo de desplazamiento y que no puede desplazarse más adelante.

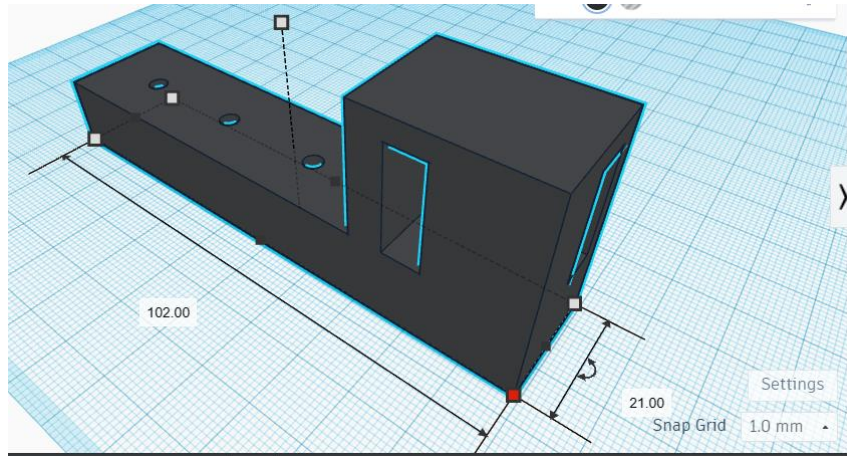


Figura 147: Soporte-Final de Carrera en forma de “L” Soportes en los que se fijan los Micro Switch
Fuente: Elaboración Propia 2024.

Estas piezas en forma de “L” tiene la finalidad de sujetarse de la plancha en la que se encuentra el Microcontrolador Esp32 Cam, para ser accionados antes que la placa se salga del eje, es importante mencionar que para esta impresión se determino fuesen 2 piezas ya que de esta forma se abarcarían todos los ángulos en los que se pudiera salir del riel de desplazamiento.

6. Problema con la fijación del Microcontrolador Esp32 Cam

Para el sexto y último punto de problemática a modificar, se planteó la implementación, diseño y fabricación de 2 piezas, la primera capaz de fungir como soporte principal del Microcontrolador Esp32 Cam, en la cual este dispositivo fuese insertado para su correcto desplazamiento, cabe mencionar que este mismo soporte seria insertado previamente a la plancha de desplazamiento central del Modulo de Monitoreo y sobre esta ya insertada correctamente, se presentaría el microcontrolador para posteriormente ser fijado en ella, fijándola por completo y evitando que esta misma se callera y dañara.

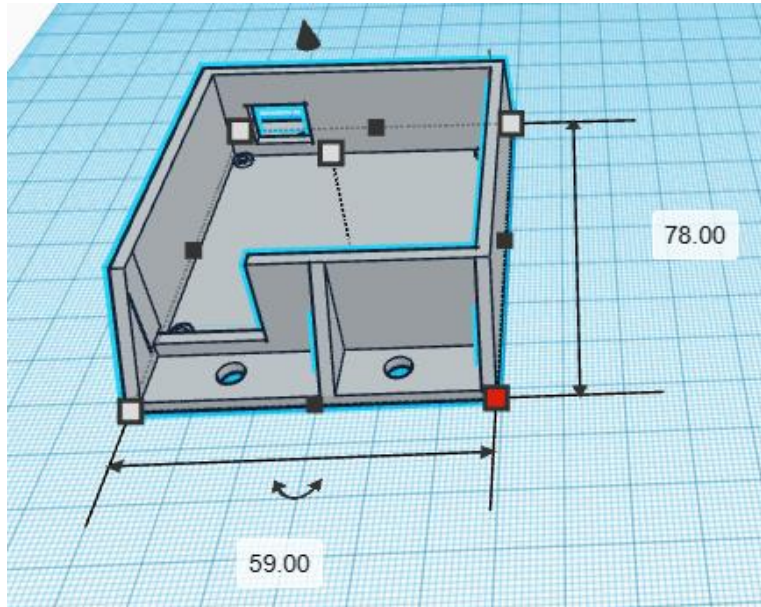


Figura 148: Soporte Principal para la Inserción del Microcontrolador Esp32 Cam.
 Fuente: Elaboración Propia 2024.

Con el soporte principal diseñado, se llegó a la conclusión de que este mismo debía requerir de una tapa que protegiera en su totalidad al microcontrolador y evitando que este se encuentre expuesto a daños externos, por tal motivo se diseñó y desarrollo una pieza con las medidas exactas del microcontrolador y el Soporte principal, de tal forma que fungiera como una tapa de protección para el microcontrolador, contacto clara mente con las perforaciones necesarias para la cámara.

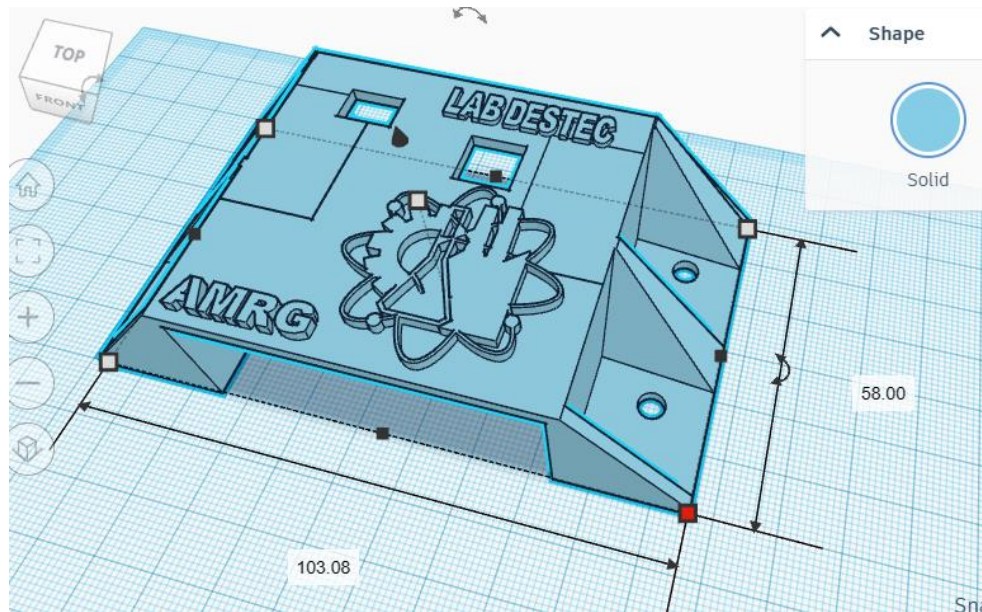


Figura 149: Tapa del Soporte Principal para la Inserción del Microcontrolador Esp32 Cam.
 Fuente: Elaboración Propia 2024.

3.6.2 Impresión en 3D de los Soportes y Refuerzos diseñados

Con el análisis sobre los puntos de error en el Módulo de Monitoreo, las medidas pertinentes y las piezas diseñadas correctamente en el software de programación, se procedió con el proceso de impresión de las piezas, esto mediante la implementación de un software llamado CHITUBOX, proporcionado por el fabricante de la impresora 3D CREALITY, en el software se realizaron las configuraciones según las especificaciones de la resina a utilizar (resina ABS color gris claro).

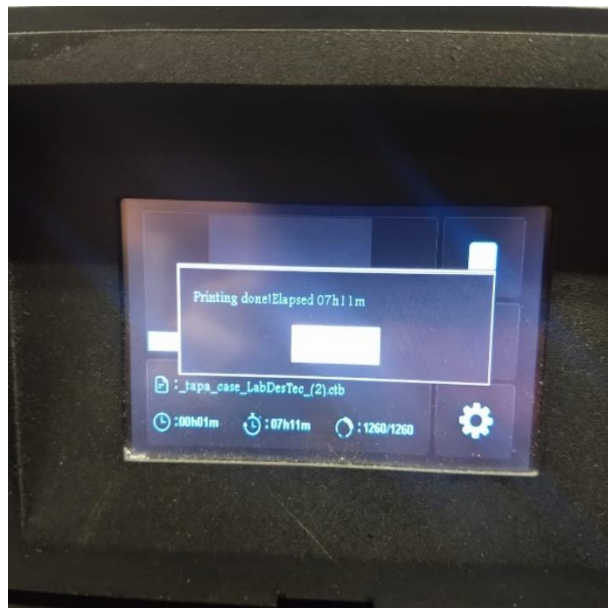


Figura 150: Proceso de Impresión de las Piezas en 3D.
Fuente: Elaboración Propia 2024.

Posteriormente se revisó la orientación óptima para la impresión y revisar si es necesario la colocación de soportes, la cual en este caso por el tiempo de orientación solo fueron necesarios dos para el hueco de la entrada de los cables de los interruptores de final de carrera, después de configurar la pieza para la impresora se exporta el archivo en formato .ctb, se guarda en una memoria usb; posteriormente se prepara la impresora colocando en la charola la resina a utilizar considerando el gramaje de la pieza especificado en el archivo .ctb, después se inserta en la impresora y se selecciona la pieza a imprimir. Ya seleccionada la pieza la impresora comenzara con el proceso de impresión, cabe mencionar que este proceso se realizo con cada una de las piezas diseñadas, por lo que queda implícito

que este proceso de impresión de cada pieza se llevó mucho tiempo, aproximadamente de 7 a 8 horas por pieza.



Figura 151: Proceso de Impresión de las Piezas en 3D, en la Impresora.
Fuente: Elaboración Propia 2024

Una vez terminada la impresión se retira de la plancha de la impresión con una espátula de metal, y su vez se retiran los soportes añadidos a cada pieza con la finalidad de que le den la rigidez suficiente para que esta tome forma durante el proceso de impresión, este proceso de retirar los soportes se hace con pinzas de corte lo más cerca a la pieza posible, tratando de dejar la pieza liza o los cortes casi al ras de la superficie de dicha pieza impresa.



Figura 152: Pinzas de punta empleadas para retirar los soportes de las piezas impresas en 3D.
Fuente: Elaboración Propia 2024.

Con los soportes de las Piezas impresas ya retirados, se procede con la colocación de dichas piezas en la máquina “Tina” de lavado, en ella se introduce la pieza y alcohol isopropílico durante un periodo no mayor a diez “10” minutos en la “Tina”, esto se hace con la finalidad de poder retirar cualquier residuo de resina sin curar sobre la pieza. Ya con el proceso de lavado terminado, se cambia el recipiente por una base metálica para realizar el proceso de curado final el cual tiene una duración en promedio de 5 a 10 min con luz es decir rayos “UV”, con la finalidad de garantizar la rigidez estructural de la pieza, evitando así que se torne a un color amarillento y poder conservar la transparencia.

Teniendo la pieza curada, se procedió a lijar con 3 diferentes tipos de grano, lija de agua número de número 400, 600,1200 y 2000, debido a que la pieza perdió transparencia con el lavado de alcohol isopropílico, además de eliminar marcas de los soportes colocados para la impresión, obteniendo una pieza con acabado liso lista para colocarle tres capas de barniz acrílico y terminar de recuperar la transparencia deseada, una vez obtenida la pieza terminada se realizó el montaje con la pieza que sujeta la placa con el microcontrolador en la placa de soporte móvil. A continuación, se presentan todas las piezas en su formato de diseño y ya Impresas:

Soporte-Grapa en forma de “U” para la Fijación del Cableado

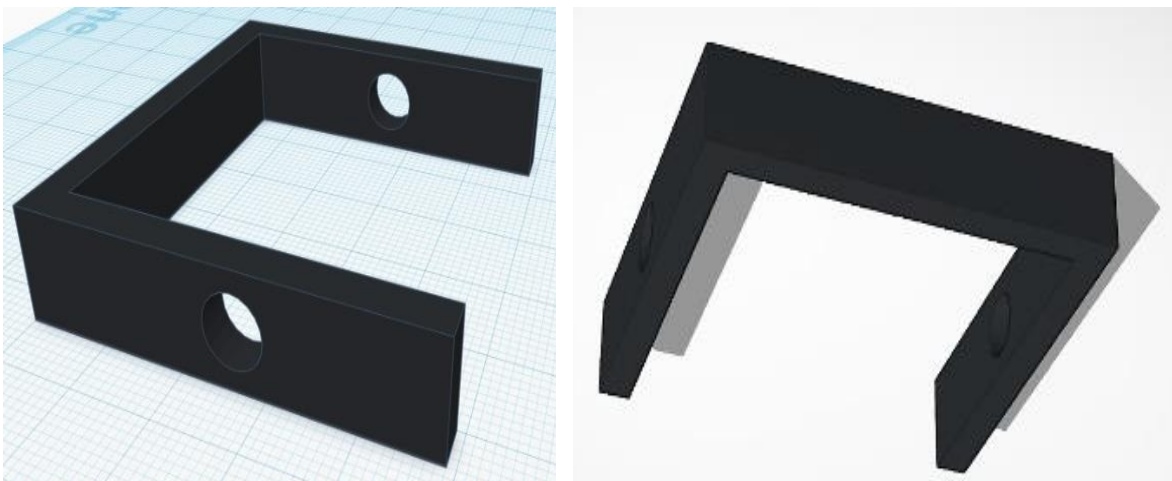


Figura 153: Soporte-Grapa en forma de “U” para la Fijación del Cableado, “Diseño”.
Fuente: Elaboración Propia 2024.

Esta pieza en forma de “L” fungiría como, con la finalidad de hacer presión en sus extremidades, para posterior mente ser fijadas a alguno de los pilares de la estructura general del Módulo de Monitoreo, por lo que al introducir el cableado dentro de esta y posterior mente fijarla, se evitaría el daño a ese cableado y se le proporcionaría un estiramiento y rigidez al mismo cableado, evitando así curvas en el mismo y elongaciones, dándole una vista más estética al Sistema.

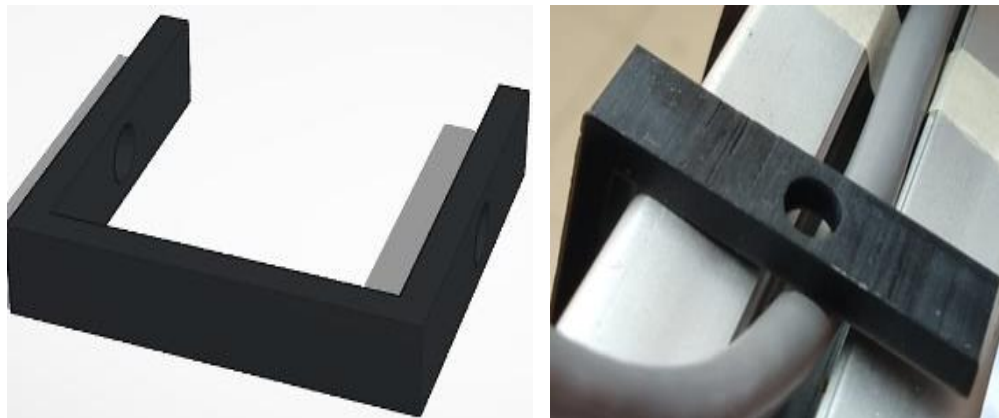


Figura 154: Soporte-Grapa en forma de “U” para la Fijación del Cableado, “Impresión”.
Fuente: Elaboración Propia 2024.

Soporte-Cople para la Inserción del Cable RJ45

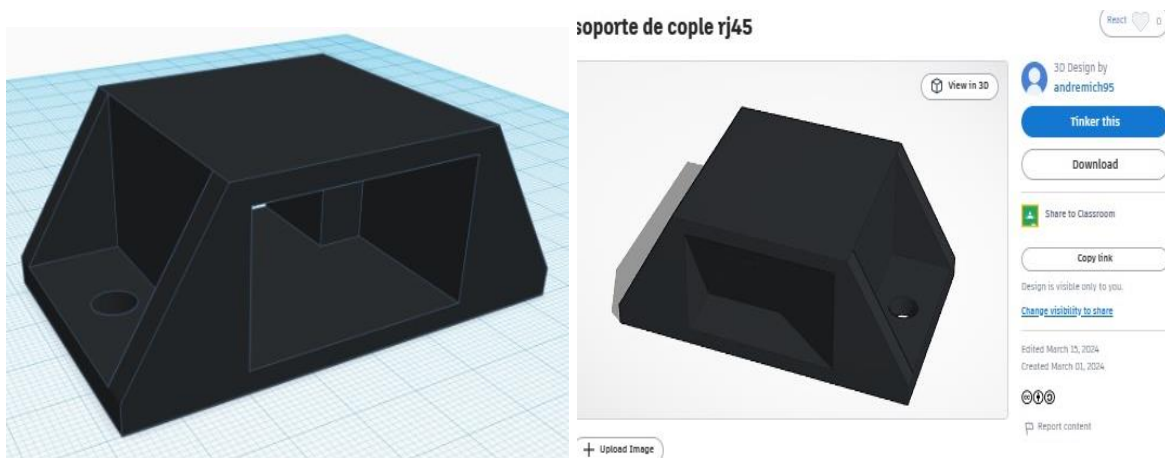


Figura 155: Soporte-Cople en forma de Pirámide para interconectar el cable RJ45, “Diseño”.
Fuente: Elaboración Propia 2024.

Esta pieza en forma de “Pirámide” tiene la finalidad de sujetar el cableado y ser un cople al mismo tiempo, permitiendo la inserción de las dos entradas en sus dos

extremos del cableado RJ45, brindando así la posibilidad de estirar dicho cableado brindándole aún más estética a la estructura del módulo de Monitoreo.

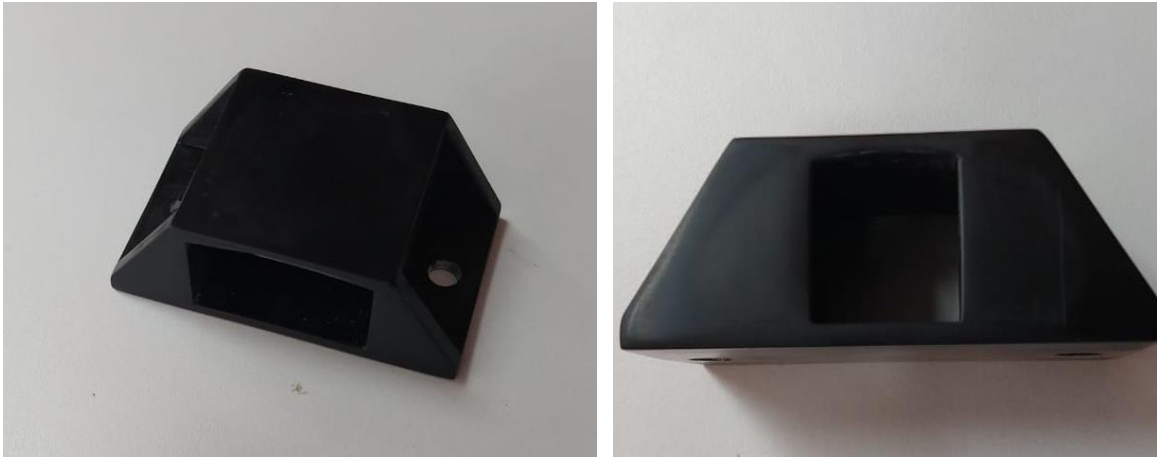


Figura 156: Soporte-Cople en forma de Pirámide para interconectar el cable RJ45, "Impreso".
Fuente: Elaboración Propia 2024.

Soporte-Tope de Final de Carrera

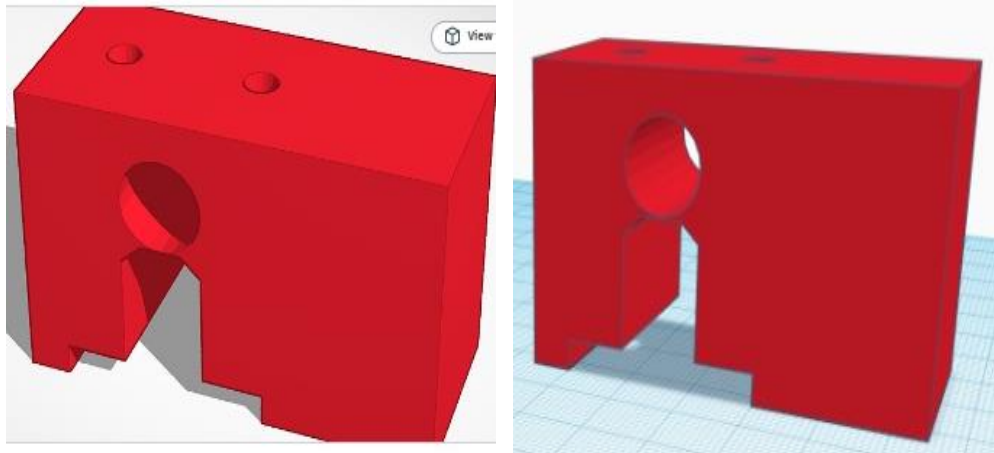


Figura 157: Soporte-Tope Final de Carrera, "Diseño".
Fuente: Elaboración Propia 2024.

Esta pieza en forma de "Rectángulo" tiene la finalidad de sujetarse del riel de desplazamiento de tal forma que al insertarlo en el este se mantenga fijo y lo suficientemente rígido para que cuando el módulo de Monitoreo lo toque y empuje querido avanzar o desplazarse aún más, este lo frene y le impida el paso, adicional a esto.

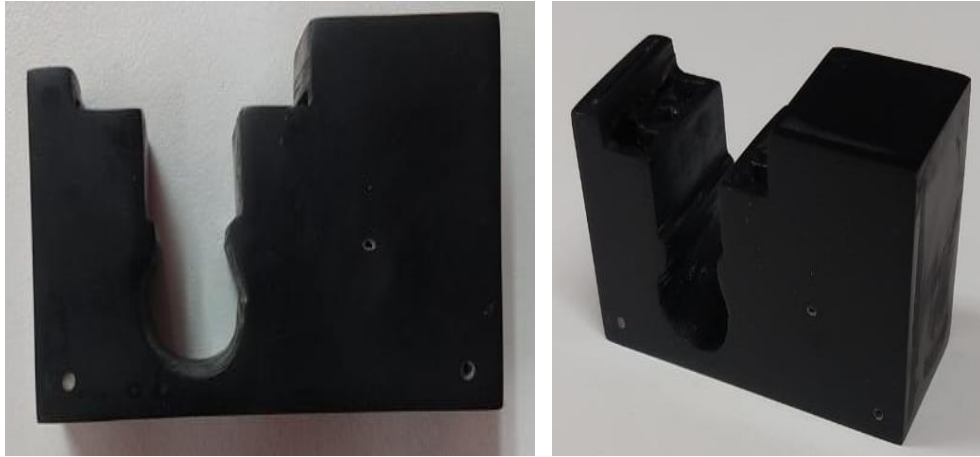


Figura 158: Soporte-Tope Final de Carrera, "Impreso".
Fuente: Elaboración Propia 2024.

Soporte-Tope de Eje de Banda

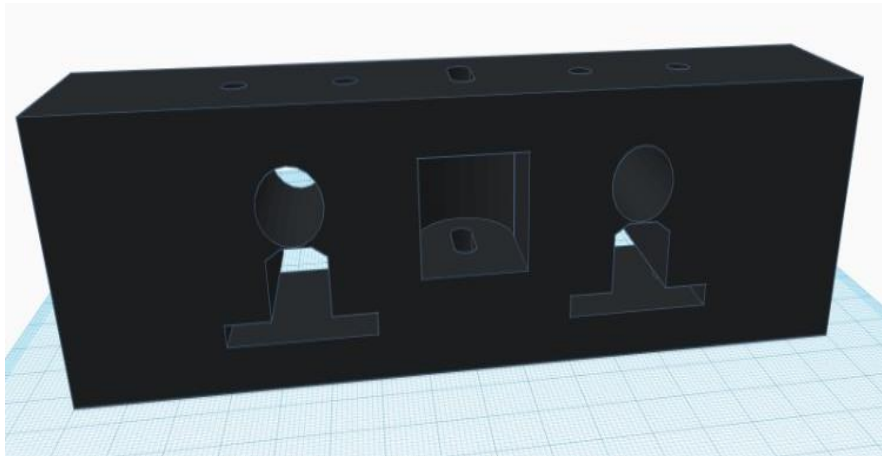


Figura 159: Soporte-Tope de Eje de Banda, "Diseño".
Fuente: Elaboración Propia 2024.

Esta pieza en forma de "Rectángulo" tiene la finalidad de sujetarse ambos rieles de desplazamiento y la banda, de tal forma que al insertarlo en el este se mantenga fijo y lo suficientemente rígido para que cuando el módulo de Monitoreo lo toque y empuje querido avanzar o desplazarse aún más, este lo frene y le impida el paso, adicional a esto, así como también darles la separación necesaria a estos para un mejor funcionamiento de desplazamiento del Módulo de Monitoreo.

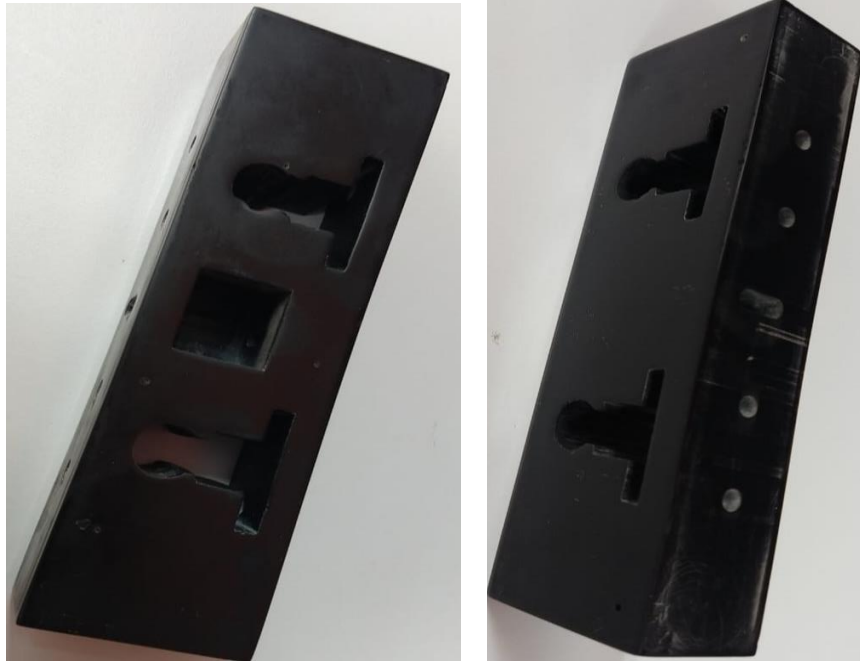


Figura 160: Soporte-Tope de Eje de Banda, "Impreso".
Fuente: Elaboración Propia 2024.

Soporte-Final de Carrera en forma de "L"

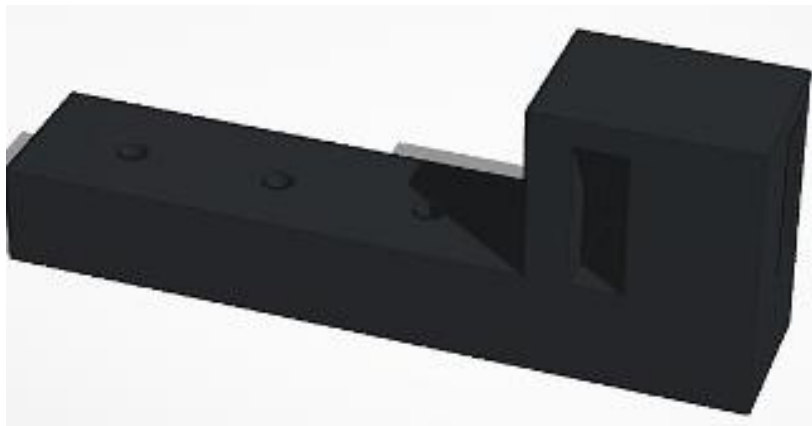


Figura 161: Soporte-Final de Carrera en forma de "U", "Diseño".
Fuente: Elaboración Propia 2024.

Estas piezas en forma de "L" tiene la finalidad de sujetarse de la plancha en la que se encuentra el Microcontrolador Esp32 Cam, para ser accionados antes que la placa se salga del eje, es importante mencionar que para esta impresión se determinó fuesen 2 piezas ya que de esta forma se abarcarían todos los ángulos en los que se pudiera salir del riel de desplazamiento.



Figura 162: Soporte-Final de Carrera en forma de “U”, “Impresión”.
Fuente: Elaboración Propia 2024.

Soporte Principal para la Inserción del Microcontrolador Esp32 Cam y Tapa

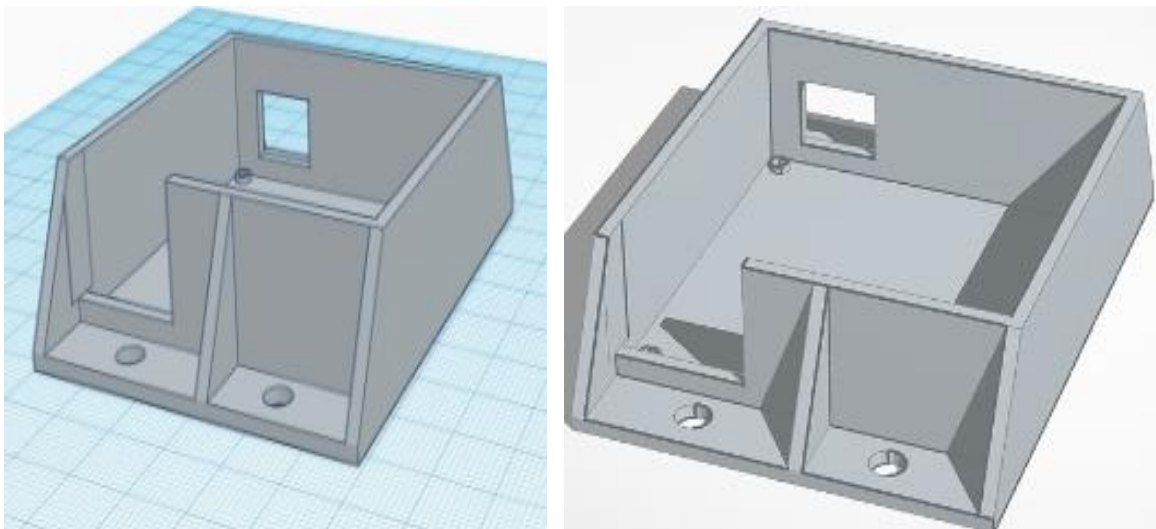


Figura 163: Soporte Principal para la Inserción del Microcontrolador Esp32 Cam, “Diseño”.
Fuente: Elaboración Propia 2024.

Con el soporte principal diseñado, se llegó a la conclusión de que este mismo debía requerir de una tapa que protegiera en su totalidad al microcontrolador y evitando que este se encuentre expuesto a daños externos, por tal motivo se diseñó y desarrollo una pieza con las medidas exactas del microcontrolador y el Soporte principal, de tal forma que fungiera como una tapa de protección para el microcontrolador, contacto clara mente con las perforaciones necesarias para la cámara.

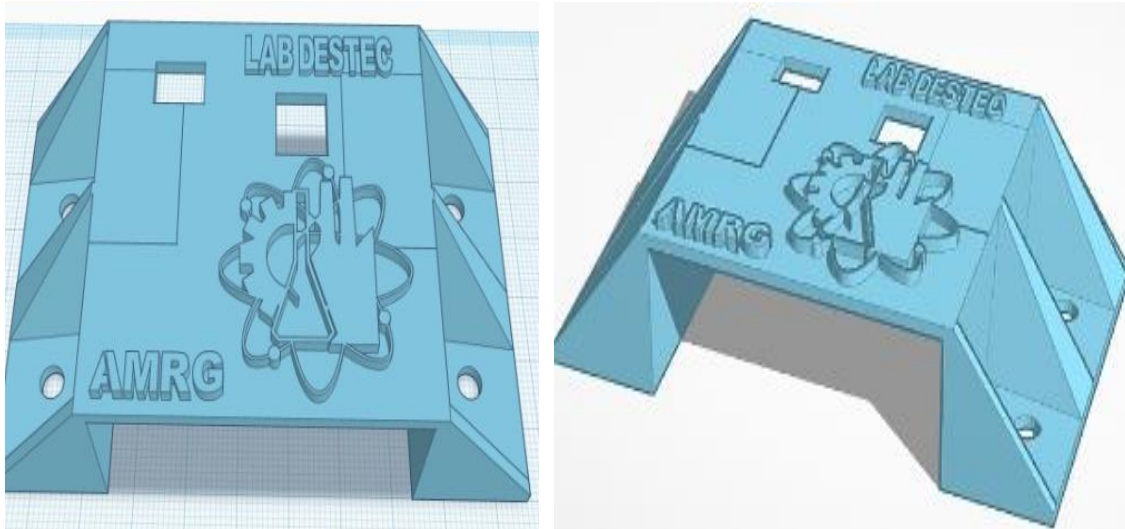


Figura 164: Tapa del Soporte Principal para la Inserción del Microcontrolador Esp32 Cam, "Diseño".
Fuente: Elaboración Propia 2024.

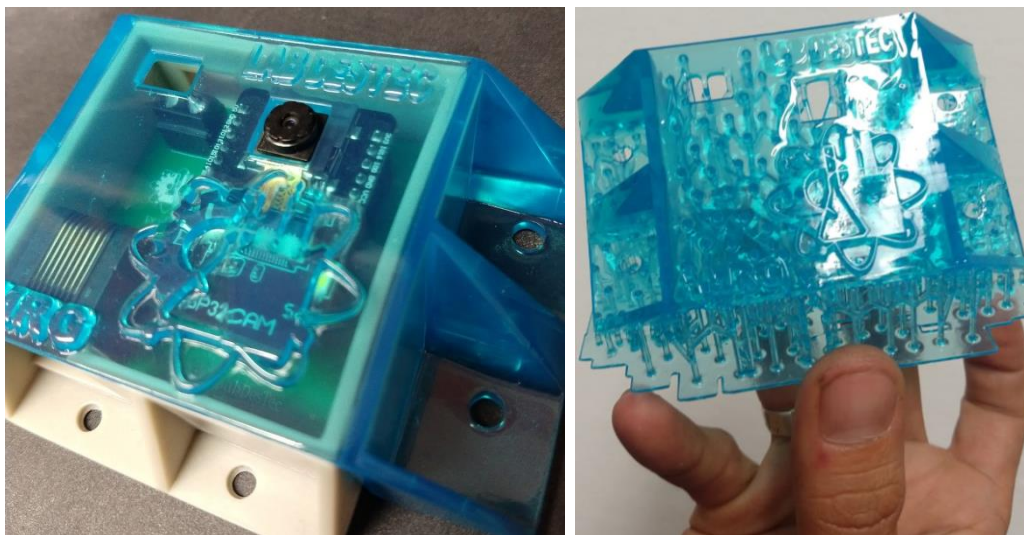


Figura 165: Tapa y Soporte Principal para la Inserción del Microcontrolador Esp32 Cam, "Impresión".

Fuente: Elaboración Propia 2024.

Cabe mencionar que para el caso del diseño del soporte principal para el microcontrolador Esp32 Cam se obtuvieron las medidas del microcontrolador ya colocado en la placa de circuito impreso previamente armada, tomando medidas de la distancia que hay entre los rieles de la segunda plancha, tomando en cuenta el método de sujeción a la misma, la cual sería a través de tornillos con contra tuerca ancha la cual entraría sobre los rieles de la placa, no obstante para la elaboración del estuche, se decidió que este se separaría en dos piezas.

- La primera es el Soporte Principal del Microcontrolador

Es en la cual se monta la placa de circuito impreso, tomando en cuenta la altura del microcontrolador ya colocado en la misma, así como huecos que permitieran la conexión del cable Rj45 y para poder introducir los cables de los interruptores de final de carrera.

- La segunda fue la Tapa de dicho Soporte

Diseñada para cubrir y sujetar la pieza que contiene la cámara teniendo en cuenta las mediciones de la ubicación donde se encuentra la cámara, con el objetivo que sobresalga y no sea obstruida, con la finalidad de obtener siempre la mejor calidad de imagen.

De forma adicional, se agregaron elementos de diseño estéticos como lo son el nombre de en donde se realizó, el logo de la Facultad para representación; la impresión se realizó en resina ABS color azul transparente, la cual nos permite visualizar el microcontrolador y lo conectores internos, a su vez permite la salida de la luz que emite el LED integrado.

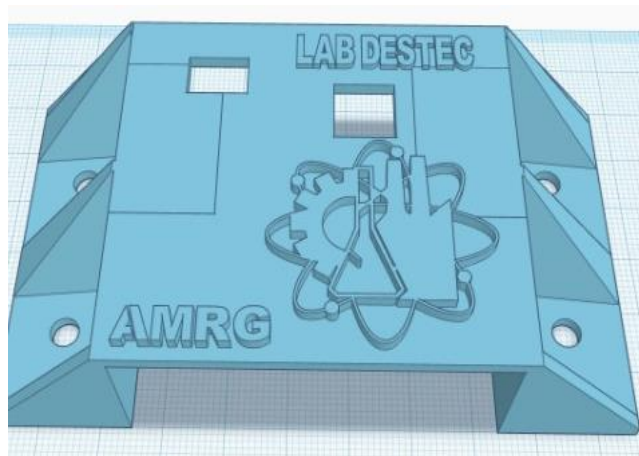


Figura 166: Tapa del Soporte Principal para la Inserción del Esp32 Cam, “Diseño Grabado”.

Fuente: Elaboración Propia 2024.

Con las piezas ya impresas, se procedió con la inserción de estas, a la estructura del Módulo de Monitoreo, esto con la finalidad de reforzar la estructura y brindarle mayor estética, así como proporcionar una mayor funcionalidad en torno al desplazamiento del Sistema de Monitoreo. A continuación, se presentan todas las piezas ya acopladas para el refuerzo de la estructura del Módulo.

3.6.3 Inserción de las Piezas Impresas en 3D, a la Estructura General del sistema de Monitoreo

Soporte-Grapa en forma de “U” para la Fijación del Cableado



Figura 167: Soporte-Grapa en forma de “U” para la Fijación del Cableado, “Acoplado”.
Fuente: Elaboración Propia 2024.

Soporte-Cople para la Inserción del Cable RJ45



Figura 168: Soporte-Cople en forma de Pirámide para interconectar el cable RJ45, “Acoplado”.
Fuente: Elaboración Propia 2024.

Soporte-Tope de Final de Carrera

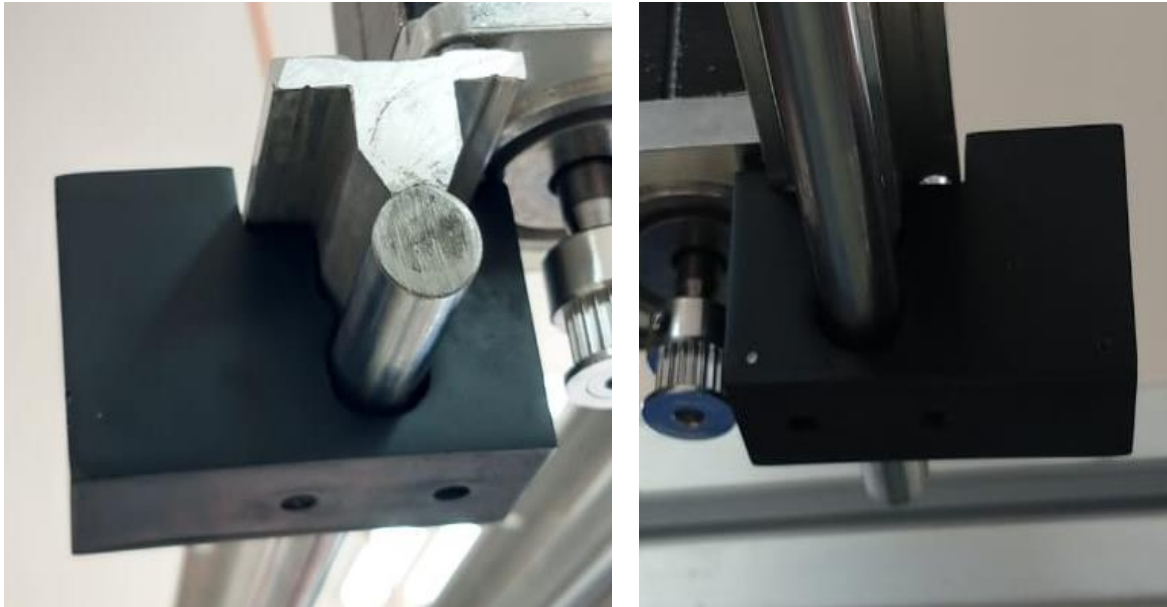


Figura 169: Soporte-Tope Final de Carrera, "Acoplado".
Fuente: Elaboración Propia 2024.

Soporte-Final de Carrera en forma de "L"



Figura 170: Soporte-Final de Carrera en forma de "U", "Acoplado".
Fuente: Elaboración Propia 2024.

Soporte-Tope de Eje de Banda



Figura 171: Soporte-Tope de Eje de Banda, "Acoplado".
Fuente: Elaboración Propia 2024.

Soporte Principal para la Inserción del Microcontrolador Esp32 Cam y Tapa

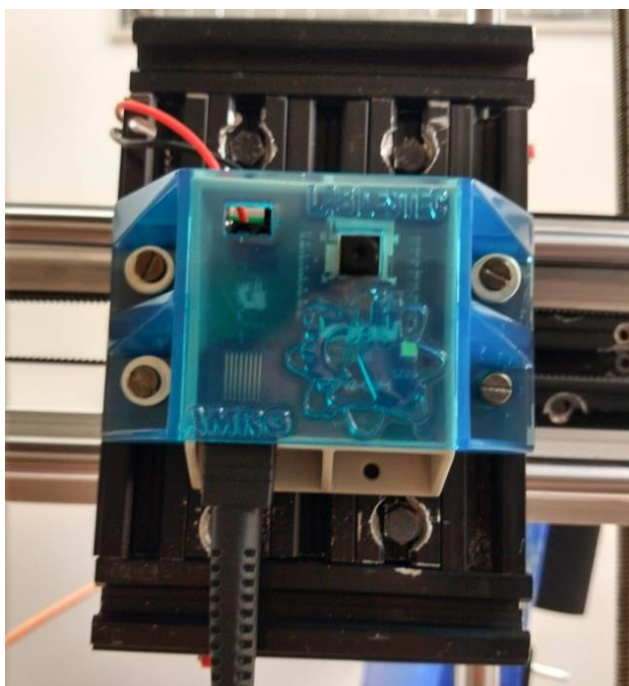


Figura 172: Soporte y Tapa Principal para la Inserción del Microcontrolador Esp32 Cam,
"Acoplado".
Fuente: Elaboración Propia 2024.

CAPITULO CUARTO

4.1 Resultados Finales

Durante la elaboración de este proyecto, se realizó una profunda investigación en diversos artículos, proyectos de investigación y fuentes científicas sobre los diferentes prototipos de módulo de monitoreo para un laboratorio remoto mediante dispositivos IoT que existen, así como su funcionamiento y los diversos avances tecnológicos que estos han tenido a lo largo del y que han permitido ampliar el área de uso de estas herramientas.

Adicional a esto, durante la gestión y desarrollo del “Sistema de Monitoreo”, se planteó el desarrollo de 2 cursos con la finalidad de tener los conocimientos necesarios para la elaboración del proyecto, estos fueron los siguientes:

- Se llevó a cabo un curso de capacitación en la plataforma Linux Professional Institute sobre los “Fundamentos de Programación en Linux LPI” con el fin de obtener los conocimientos básicos y esenciales para la programación en Linux Essentials, el cual tuvo una duración de 75 horas.
- De igual forma Se llevó a cabo un curso de capacitación en la plataforma ASOCIACIÓN IOT MÉXICO sobre los “Fundamentos de Internet de las Cosas de IOT” con el fin de obtener los conocimientos básicos y esenciales para la programación y desarrollo de proyecto en el Internet de las Cosas en IOT, el cual tuvo una duración de 75 horas.

Una vez modificada la estructura de todo el sistema de Monitoreo, integrando a este un considerable número de impresiones en 3D diseñadas para fungir, como refuerzos de los soportes de dicha estructura, como puntos de sujeción para el cableado entre los componentes eléctricos y electrónicos, y como topes para la delimitación del desplazamiento del Módulo de Monitoreo, habiendo fabricado y acoplado al sistema las nuevas placas de circuito impreso para un mejor control, una mayor interconexión, un aumento en la estabilidad y comunicación entre los dos microcontroladores “Arduino UNO R4 Wifi y Esp32 Cam”, teniendo elaborado el

centro o tablero de control manual mediante un Dashboard para el dominio del control y Monitoreo del desplazamiento del Módulo constante y continuo pudiendo ver la transmisión de la imagen que genera la Esp32Cam, y con el diseño y mejoramiento de los tres códigos de programación para el control del Mismo en su funcionamiento más rápido y eficiente, se obtuvo como resultado un correcto “Sistema de Monitoreo para un Laboratorio Remoto Mediante Dispositivos IoT”.

Este “Sistema de Monitoreo” cuenta principal mente con dos factores indispensables para la función de Monitoreo entorno al Laboratorio remoto, estos son los siguientes:

- El primer Factor es el Módulo de Monitoreo, integrado a un microcontrolador Esp32Cam, permitiendo a todo usuario tener una apreciación total de todos los procesos que se realicen dentro del rango de visibilidad proporcionado por el microcontrolador, en todo momento y desde cualquier lugar con acceso a Internet.
- El segundo Factor es el diseño e implementación de un tablero de control manual que le permite al usuario tener un control absoluto de las direcciones de desplazamiento del Módulo de monitoreo, mediante una comunicación entre el Dashboard y los dos microcontroladores.

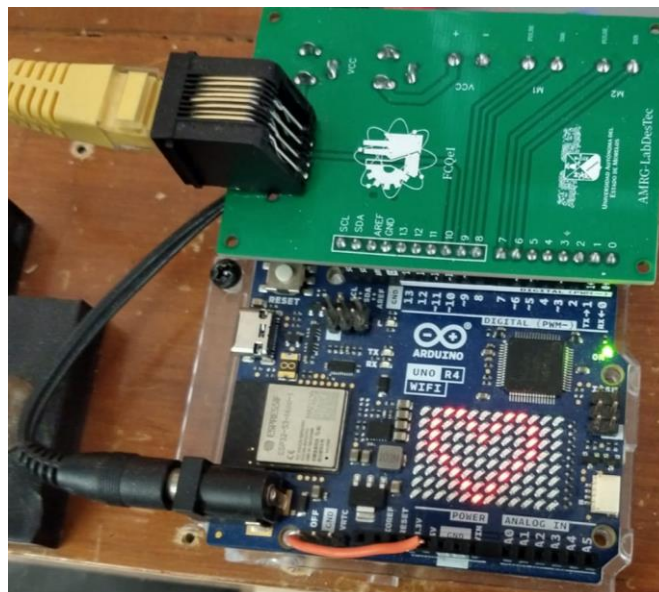


Figura 173: Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.
Fuente: Elaboración Propia 2024.

4.2 Discusión de los Resultados Finales

A continuación, se presenta los puntos claves a discutir respecto a los Resultados finales del “Sistema de Monitoreo”:

- Cada uno de los códigos diseñados e implementados para el control del “Sistema de Monitoreo” como parte de su programación le permiten al “Modulo de Monitoreo” desplazarse dentro de los rangos establecidos y visualizar todo el entorno dentro de los rangos de visibilidad del “Microcontrolador Esp32 Cam”. Sin embargo como el desplazamiento del módulo esta a expensas de un pulso emitido desde el Dashboard este si no recibe ninguno por ende no se desplazara, es decir que permanecerá estático permitiendo visualizar el entorno, pero sin ningún desplazamiento, sin embargo, si este si recibe un pulso, hacia cualquiera de las direcciones que el Dashboard permite “Adelante, Atrás, Izquierda, Derecha”, este se desplazara en torno a toda la superficie, teniendo entonces un control total de este proceso, de manera eficiente y precisa la visualización y el monitoreo en tiempo real del Laboratorio Remoto
- En el caso de presentarse un corte energético o una interrupción en el suministro de energía en todo el “Sistema de Monitoreo”, este se apagará inmediatamente y por consiguiente debido a su programación en el momento en el que se reestablezca la energía, el sistema inicializará nuevamente su programación, del mismo modo si se desea seguir con el proceso de monitoreo y poder controlar este mismo mediante el Dashboard, se debe volver a recargar o refrescar la página web.
- Durante las pruebas de funcionamiento de cada uno de los códigos ya mencionados siempre se respetó el límite de peso establecido por los motores paso a paso, implementados en la estructura para el desplazamiento del Módulo de Monitoreo, así como también el establecimiento de la velocidad mínima y máxima de desplazamiento de este para lograr así un

funcionamiento idóneo entorno al dispositivo, ya que esta velocidad puede ser variada en la programación.

- Las placas de circuito impreso empleadas para el control del Sistema de Monitoreo se pudo tener una interconexión estable entre ambos microcontroladores, con una diferencia aun mayor entorno a las primeras impresiones debido a un problema en la capacitancia, ya que el método de impresión fue erróneo y por la diferencia en la calidad de los materiales, reduciendo así el tamaño del prototipo.
- La comunicación entre los microcontroladores fue por medio de pulsos digitales, debido a que uno solo no cumple con las necesidades que el sistema demanda de ambos, en otras palabras se deduce que ninguno de los microcontroladores por sí solo cuenta con las especificaciones técnicas necesarias para lograr el control o la funcionalidad de todo el módulo, en el caso del Esp32 Cam no se cuenta con las salidas, el voltaje y corrientes para enviar la señal desde el módulo hasta los drivers y en el caso del Arduino UNO R4 Wifi no se cuenta con ninguna cámara, es decir que este cuenta con una tecnología que le permite se controlado a distancia, sin embargo no cuenta con un módulo de cámara que le permita la visualización de un entorno y el monitoreo del mismo, por esta razón es que se emplearon los dos microcontroladores, con el objetivo de cubrir las necesidades técnicas esenciales para el funcionamiento del Sistema de Monitoreo.
- Uno de los problemas más grandes entorno al diseño e implementación del Dashboard, fue generar la página en HTML “Lenguaje de Marcado de Hipertexto” y CSS “Lenguaje de diseño gráfico”, empleando estos lenguajes mediante Visual Estudio Code, sin embargo el problema surge al momento de realizar el acoplamiento de las funciones de la botonera, con las acciones a efectuar respecto al Arduino en lenguaje C, en otras palabras, juntar la página web con las funciones en C en el código del microcontrolador Esp32Cam.

Por último, se presentan algunas fotografías con el objetivo de demostrar el resultado final de la estructura terminada y el correcto funcionamiento del “Sistema de Monitoreo para un Laboratorio Remoto Mediante Dispositivos IoT”, así como todas las modificaciones estéticas realizadas, durante el proyecto.

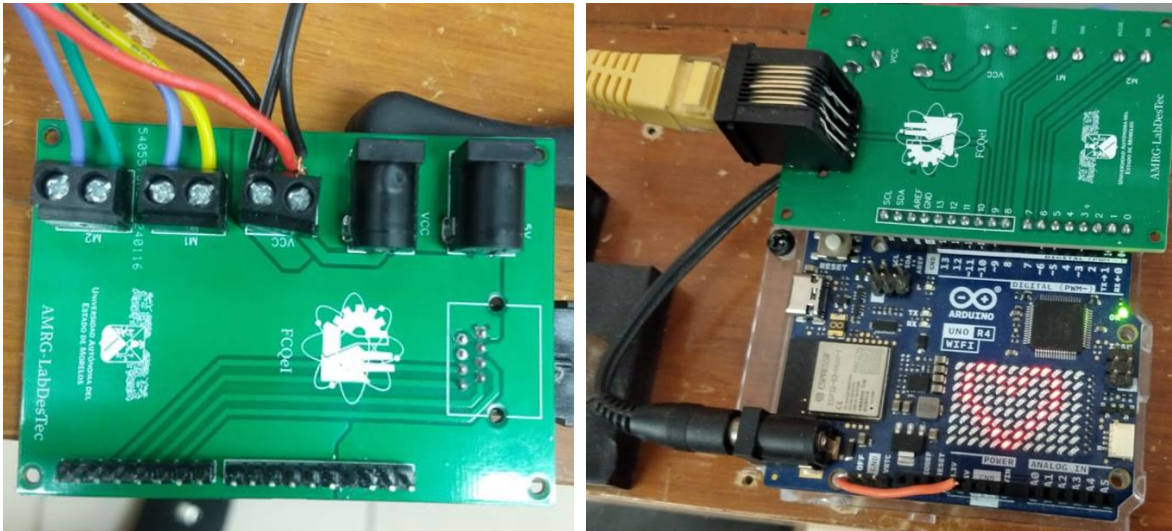


Figura 174: Microcontrolador Arduino UNO R4 Wifi “Encendido Recibiendo una Señal”.
Fuente: Elaboración Propia 2024.



Figura 175: Proceso de Impresión mediante la implementación una Impresora 3D.
Fuente: Elaboración Propia 2024.

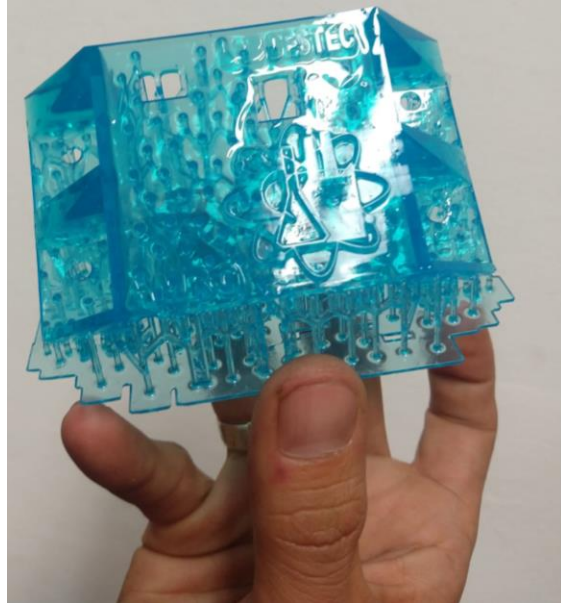


Figura 176: Tapa Principal para la Inserción del Microcontrolador Esp32 Cam.
Fuente: Elaboración Propia 2024.

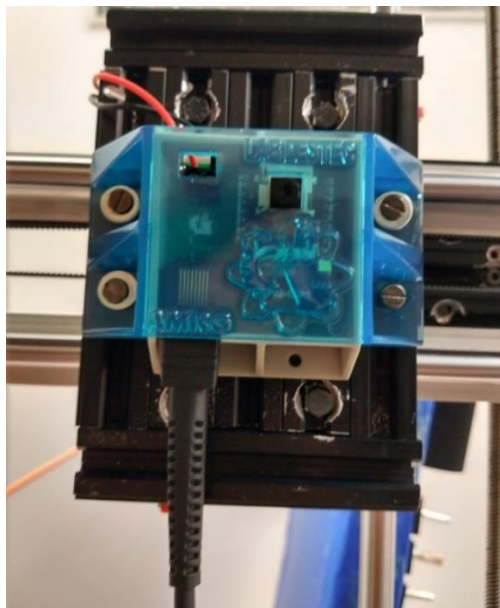
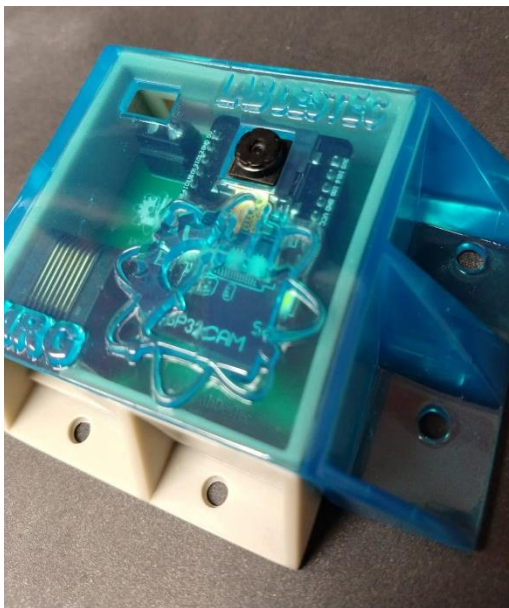


Figura 177: Soporte y Tapa Principal para la Inserción del Microcontrolador Esp32 Cam,
"Acoplado".
Fuente: Elaboración Propia 2024.

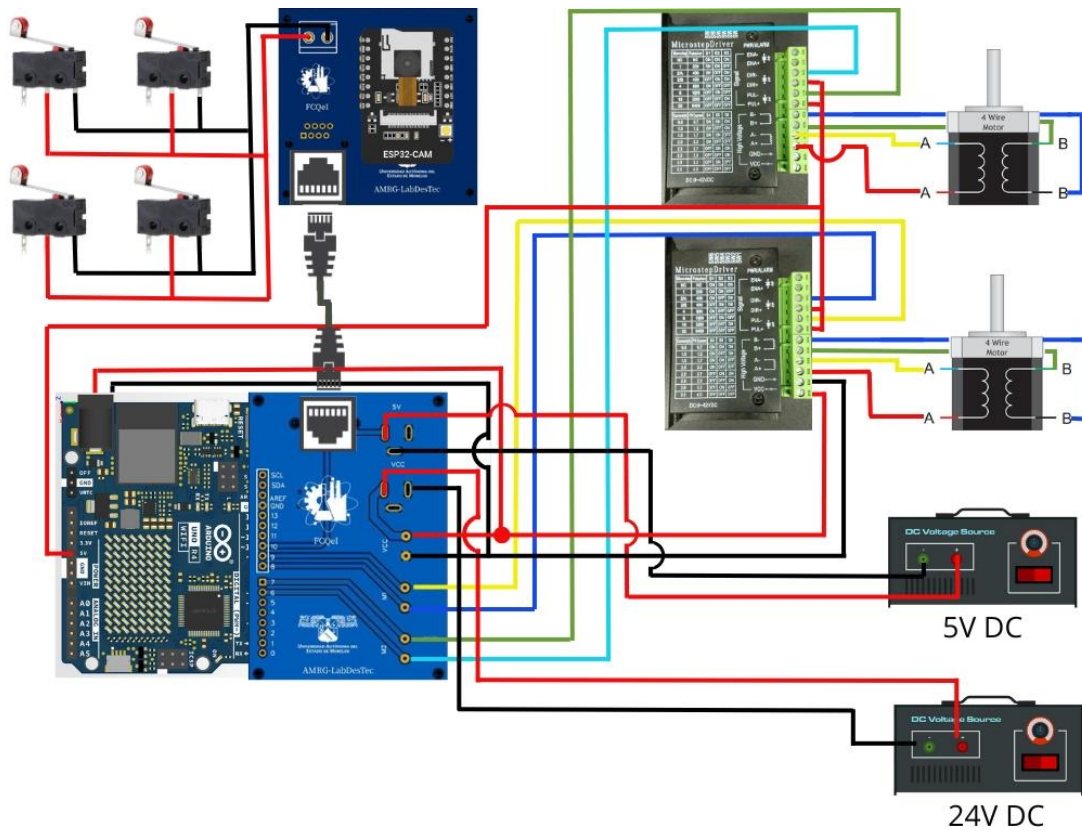


Figura 178: Diagrama de Conexión entre el Arduino UNO R4 Wifi, los Drives y los Motores a pasos
Fuente: Elaboración Propia 2024.

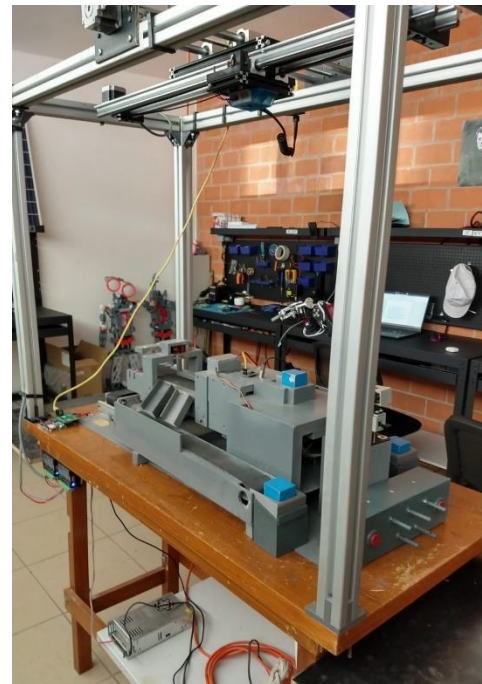
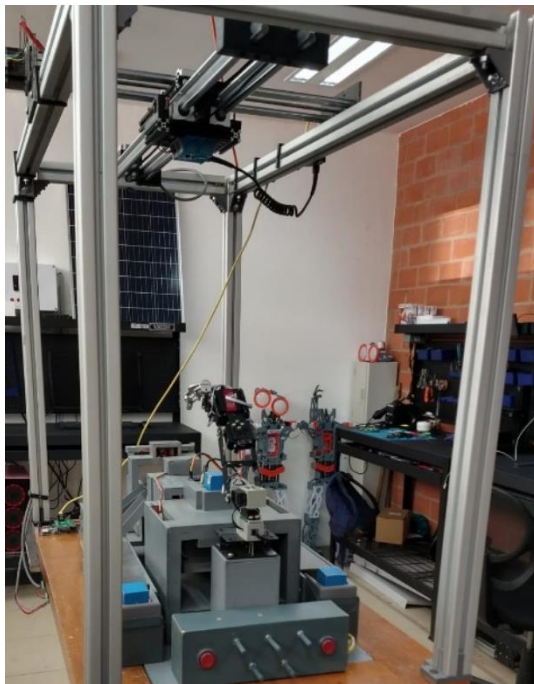


Figura 179: Laboratorio Remoto en función del Sistema de Monitoreo.
Fuente: Elaboración Propia 2024.

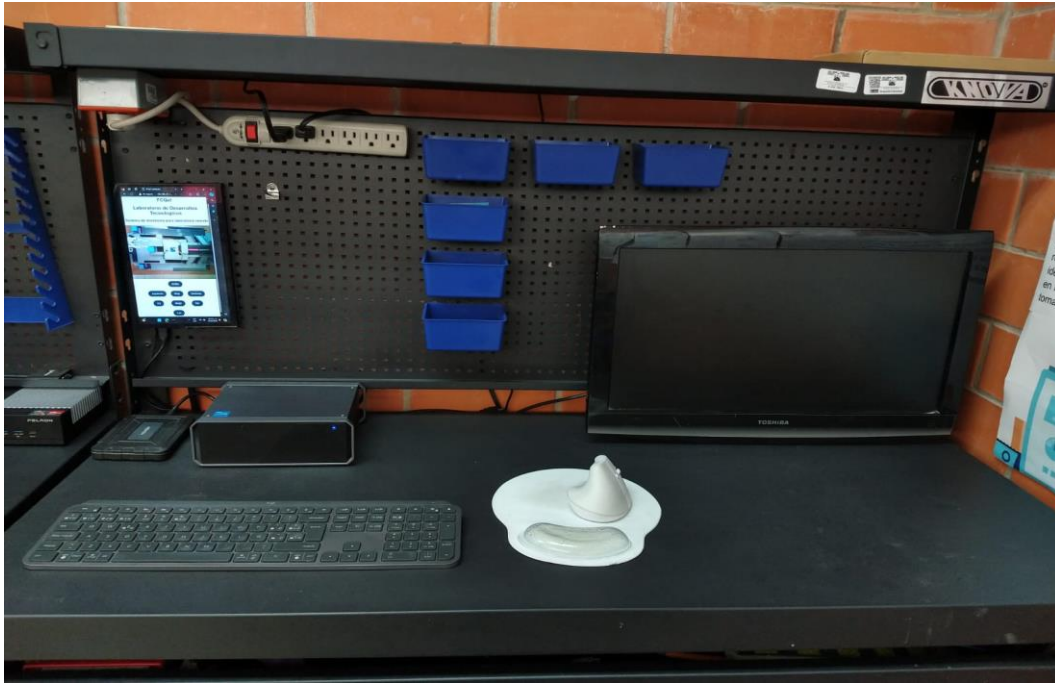


Figura 180: Modulo de control para el Control del Modulo de Monitoreo.
Fuente: Elaboración Propia 2024.

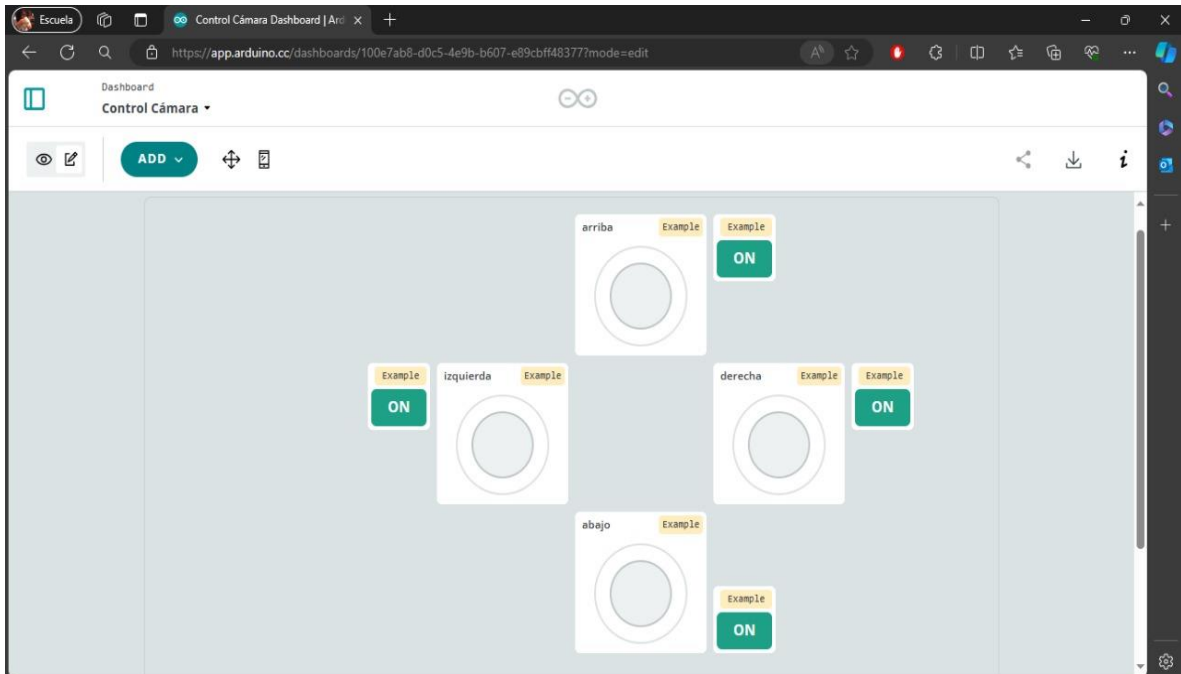


Figura 181: Dashboard Generado en Arduino Cloud para el control de los Motores Paso a Paso.
Fuente: Elaboración Propia 2024.

4.3 Conclusiones

Durante la elaboración de este proyecto se logró comprender que el uso de la tecnología en todos los sectores de la vida cotidiana ha tenido un fuerte impulso por parte de investigadores, sin embargo, desde el surgimiento del Internet de las Cosas (IoT) ha permitido ampliamente ser usado en la ingeniería como en el monitoreo ambiental, ciudades inteligentes, entre otras siendo usado en la conexión de dispositivos y la reducción de la información.

Este proyecto de nombre “Sistema de Monitoreo para un Laboratorio Remoto Mediante Dispositivos IoT”, ha sido pensado y desarrollado con el objetivo de ser una herramienta de aprendizaje a distancia, en la que por medio de un conjunto de dispositivos IoT sea posible la visualización y monitorización de los procesos realizados en un Laboratorio Remoto en tiempo real, dicho sistema tiene la característica principal de tener bajo costo y un bajo consumo energético. Este sistema permitir la manipulación de dicho monitoreo de manera remota, proporcionando la capacidad de desplazar el módulo de monitoreo “la cámara” mediante una serie de rieles, a cuatro diferentes posiciones “Al Frente, Atrás, Izquierda y Derecha, de todo su entorno, teniendo un control total de este proceso de manera eficiente y precisa para el monitoreo en tiempo real, sin embargo con este tipo de sistema se puede fortalecer la infraestructura de equipos de laboratorio en el área de ingeniería para poder ser manipulados de forma remota.

No obstante, este proyecto se pudo llevar a cabo a partir de 5 diversas etapas, fundamentales para su desarrollo dentro de las cuales se concluyó lo siguiente:

➤ **Diseño de la estructura del sistema de Monitoreo:**

En esta etapa se logró diseñar y desarrollar la estructura general sobre la cual se instalaría el Modulo de Monitoreo, dicha estructura se elaboro a base de perfiles de aluminio con 3mm de espesor, brindándole la fuerza, resistencia y estabilidad necesaria para su correcto funcionamiento, no obstante adicional a las tiras de aluminio acopladas a la base del “Laboratorio Remoto”, se inserto y diseño una

estructura adicional compuesta por 4 rieles de desplazamiento, 1 tornillo sin fin y 2 placas como bases acopladoras de los rieles, sobre las cuales se montó e instaló el Módulo de Monitoreo.

- Diseño de códigos de programación en el software Arduino para el control del Sistema de Monitoreo:

En esta segunda etapa se realizó, modificó y estableció una serie de 3 códigos de programación en el software de programación Arduino para el control de los motores a pasos mediante la comunicación entre los 2 Microcontroladores empleados en el sistema. Con el objetivo de que la señal proveniente del Dashboard se emitiera mediante un pulso, el cual sería enviado al Microcontrolador Esp32 Cam y este a su vez se comunicara con el Arduino UNO R4 Wifi para poder así indicarle a los drivers y estos a los motores a pasos el desplazamiento a realizar, con la capacidad de controlar la dirección y la velocidad del desplazamiento del Módulo de Monitoreo,

- Diseño de Placas de circuito Impreso, para el control de los Microcontroladores:

En esta etapa se logró diseñar y desarrollar 2 placas circuito impreso (PCB), estas fueron diseñadas en un software llamado Fritzing, no obstante, este no contaba con las herramientas necesarias para la realización de las placas, por lo cual se realizaron los diseños en el software “easyEDA”, ya elaboradas mediante este software se mandaron a imprimir las placas PCB. Estas placas permitieron poder obtener una interconexión estable entre ambos microcontroladores, con una diferencia aun mayor entorno a las primeras impresiones debido a un problema en la capacitancia, ya que el método de impresión fue el erróneo y por la diferencia en la calidad de los materiales, logrando así al implementar estas placas la reducción en el tamaño del prototipo, permitiendo una mayor estética y optimización de su espacio.

- Diseño de una plataforma para el control a distancia de todo el sistema de Monitoreo: centro de control manual del Brazo:

En esta etapa se logró generar un Dashboard, en el cual se puede ver la transmisión de la imagen capturada por el microcontrolador Esp32 Cam y a su vez puede

apreciarse el tablero de Control, el cual permite tener un dominio sobre el desplazamiento constante y continuo de los Motores paso a paso o Motores de pasos. Sin embargo uno de los problemas más grandes entorno al diseño e implementación del Dashboard, fue generar la página en HTML “Lenguaje de Marcado de Hipertexto” y CSS “Lenguaje de diseño gráfico”, empleando estos lenguajes mediante Visual Estudio Code, sin embargo el problema surge al momento de realizar el acoplamiento de las funciones de la botonera, con las acciones a efectuar respecto al Arduino en lenguaje C, en otras palabras, juntar la página web con las funciones en C en el código del microcontrolador Esp32Cam.

➤ Diseño, modelado e integración de impresiones 3D al Brazo:

En esta etapa, se lograron diseñar, desarrollar e imprimir 15 piezas de impresiones 3D diseñadas para fungir como refuerzos de los soportes de la estructura del Sistema de Monitoreo y como aditamento para mejora de los componentes que la integran, de las cuales 8 fueron diseñadas para servir como soportes de fijación del sistema de cableado, teniendo una forma de “U”, 1 pieza para servir como un soporte multifuncional del acoplamiento del cable “RJ45” que permite la comunicación entre los 2 microcontroladores, 1 pieza diseñada como tope del interruptor final de carrera con el fin de evitar que la banda se dañe al no permitir que la placa llegara al límite del eje del motor paso a paso, delimitando su desplazamiento. Insertado en un riel y se sujeta mediante un tornillo el mismo que sujeta el motor paso a paso, 2 pieza que sirven como soportes para la inserción de un interruptor final de carrera, con el objetivo de delimitar el espacio de giro de los motores para evitar daños a la banda y al acoplador de aluminio del tornillo sin fin, y final mente dos piezas en las cuales se inserta el microcontrolador Esp32 Cam siendo estas diseñadas para servir como estuche y tapa de tal forma que permitieran la sujeción del microcontrolador al estuche y a la placa misma, esto con la finalidad de proteger el dispositivo y evitar así entrar en contacto con una mayor acumulación de polvo.

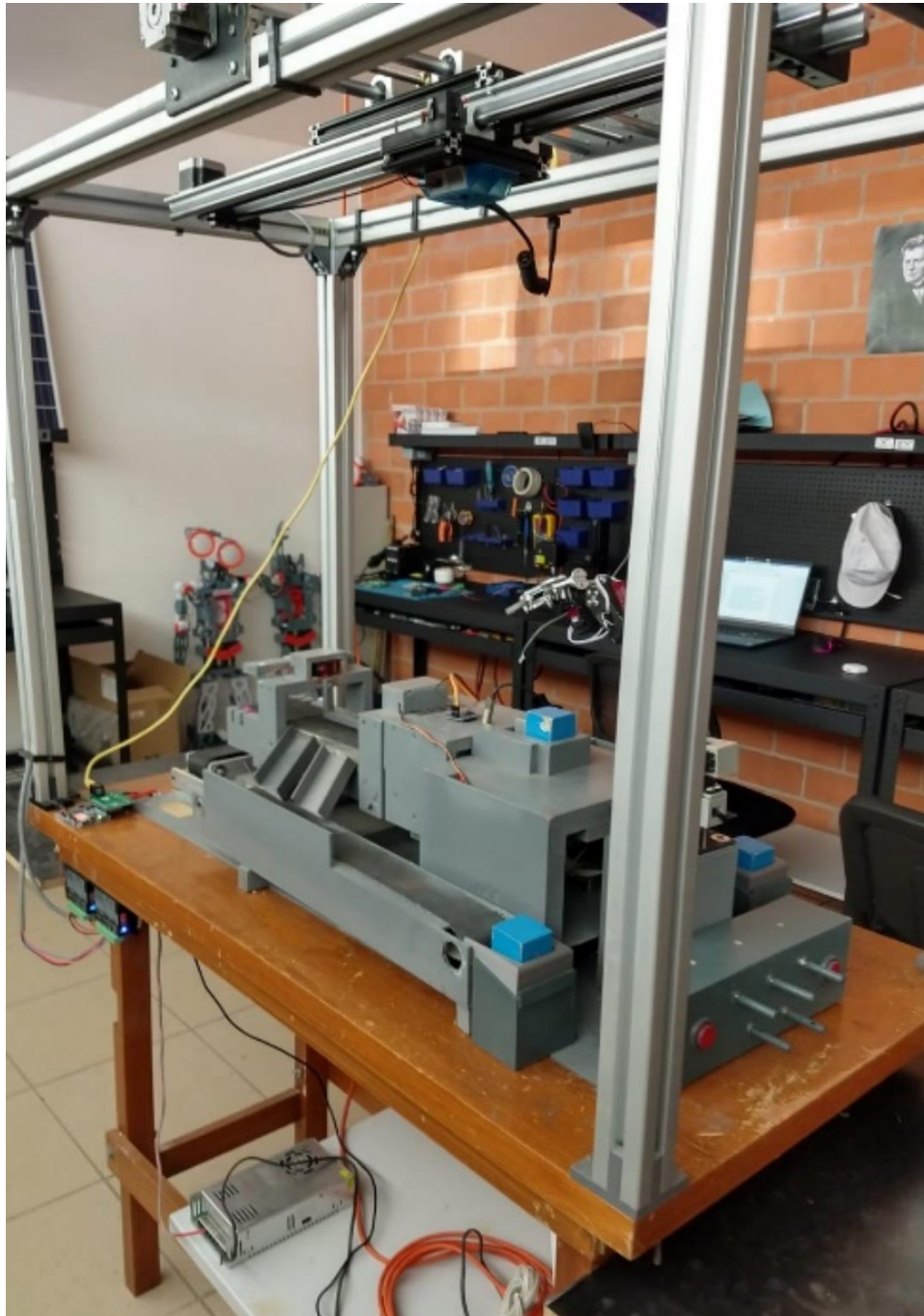


Figura 182: Sistema de Monitoreo Funcionando.
Fuente: Elaboración Propia 2024.

4.4 Bibliografía

1. Sotelo, J. L. R., Londoño, A. L., Botero, C. A. V., & Hurtado, R. D. F. (2017). Sistema de monitoreo y control remoto usando IOT para un regulador de presión. *Scientia et Technica*, 22(4), 391-397.
2. Yacelga Yacelga, E. A. (2022). *Implementación de un sistema de monitoreo y control basado en IOT para un sistema dinámico real* (Bachelor's thesis).
3. Londoño Alzate, J. A. Diseño e implementación de un laboratorio remoto IOT para una planta térmica didáctica.
4. Guerrero-Ibañez, J. A., Estrada-Gonzalez, F. P., Medina-Tejeda, M. A., Rivera-Gutierrez, M. G., Alcaraz-Aguirre, J. M., Maldonado-Mendoza, C. A., ... & Lopez-Gonzalez, V. I. (2017). SgreenH-IoT: Plataforma IoT para agricultura de precisión. *Sistemas, cibernética e informática*, 14(2).
5. Parra, M., Guillen, E., Le Mouël, F., & Carrillo, O. (2019, October). Sistema colaborativo de medición de parámetros ambientales basado en IoT. In *2nd Workshop CATAI-SmartData for Citizen Wellness*.
6. Salazar, J., & Silvestre, S. (2016). Internet de las cosas. *Techpedia. České vysoké učení technické v Praze Fakulta elektrotechnická*.
7. Muñoz, O. Q. (2019). *Internet de las cosas (IoT)*. Ibukku LLC.
8. Cisco IBSG, abril de 2011, Evans, D. (2011). Internet de las cosas. *Cómo la próxima evolución de Internet lo cambia todo. Cisco Internet Business Solutions Group-IBSG*, 11(1), 4-11.

9. Valencia, A., & Portilla, P. (2019). Internet Industrial de las Cosas (IIOT): Nueva Forma de Fabricación Inteligente. *Grade Thesis, Fundación Universitaria de Popayán, Colombia*.
10. Montoya Ponce, R. J. (2019). Método para la preservación de la privacidad en dispositivos iot vestibles extendiendo la Seguridad usando fog computing.
11. PEREIRA, R., DE SOUZA, C., PATINO, D., & LATA, J. PLATAFORMA DE ENSEÑANZA A DISTANCIA DE MICROCONTROLADORES E INTERNET DE LAS COSAS. *Ingenius [online]*. 2022, n. 28.
12. Escobar Gallardo, E., & Villazón, A. (2018). Sistema de monitoreo energético y control domótico basado en tecnología internet de las cosas. *Investigación & Desarrollo*, 18(1), 103-116.
13. Quiñones-Cuenca, M., González-Jaramillo, V., Torres, R., & Jumbo, M. (2017). Sistema de monitoreo de variables medioambientales usando una red de sensores inalámbricos y plataformas de Internet de las Cosas. *Enfoque UTE*, 8, 329-343.
14. Gómez, J. E., Castaño, S., Mercado, T., Fernandez, A., & Garcia, J. (2017). Sistema de internet de las cosas (IoT) para el monitoreo de cultivos protegidos. *Ingeniería e innovación*, 5(1).
15. Velásquez, S., & Gómez, C. (2013). Monitoreo de variables analogicas usando raspberry Pi. *Universidad, Ciencia y Tecnología*, 17(69), 170-175.
16. Ortiz, K. N. T., Muñoz, D. C. H., & Mendoza, W. N. M. (2020). Importancia de los laboratorios remotos y virtuales en la educación superior. *Documentos De Trabajo ECBTI*, 1(1).

4.5 Anexos

4.5.1 Certificados del Diplomado de IoT.



Se otorga el presente

CERTIFICADO DE TÉRMINO

a

Andres Michel Rodriguez Gordillo

por haber concluido el

Diplomado de Internet de las Cosas

atendiendo los requisitos de asistencia

entre el día 17 de ABRIL de 2023 y el 20 de JULIO de 2023.



FIRMA DIGITAL

Mtro. Diego Trujillo Pisanty

Responsable de Programa Académico

Código IoT - Orvon, S.A. de C.V.



CERTIFICADO DE CONOCIMIENTO

Por medio de la presente, certificamos que la evaluación de las competencias adquiridas, tanto en los aspectos teóricos como prácticos, necesarias para el desarrollo de proyectos basados en el

Internet de las Cosas

ha sido realizada a

Andres Michel Rodriguez Gordillo

quien ha demostrado de manera satisfactoria el conocimiento evaluado.

Verificación:



Código de certificado:
9477350952AR

Fecha de emisión:
8 de septiembre de 2023

Vigencia:
2 años

Mtro. Nahim de Anda Martín
Presidente

Lic. Antonio Guzmán Trejo
Dirección de Entidad de Evaluación

Asociación Mexicana del Internet de las Cosas A.C.

Enlace de verificación:

<https://edu.codigoiot.com/admin/tool/certificate/index.php?code=9477350952AR>



CONSTANCIA DE PARTICIPACIÓN

Por medio del presente hacemos constar que

Andres Michel Rodriguez Gordillo

participó en el programa

Diplomado de Internet de las Cosas

entre el día 17 de ABRIL de 2023 y el 20 de JULIO de 2023.

impartido por Código IoT.

Atentamente,

Mtro. Diego Trujillo Pisanty

Responsable de Programa Académico

Código IoT - Orvon SA de CV

4.5.2 Diploma de Certificación en Linux Essentials



Andrés Michel Rodríguez Gordillo

has fulfilled the qualifications and achieved all requirements of the Institute's program

Linux Essentials

and is entitled to all honours and privileges associated with this achievement.

Issued in Toronto, Ontario, Canada

25th of August, 2023


Chairman of the Board

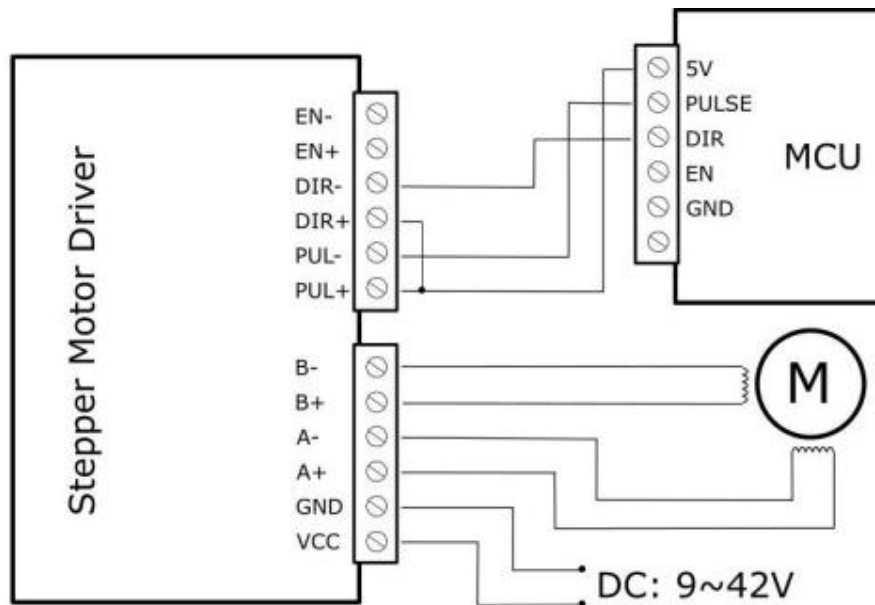

Executive Director

This certificate is not proof of certification, please visit:

lpi.org/v/LPI000575151/fm4a4n42ym

Linux Professional Institute and the circle-L logo are trademarks and registered trademarks (in Japan and the European Union) of Linux Professional Institute Inc. All Rights Reserved.

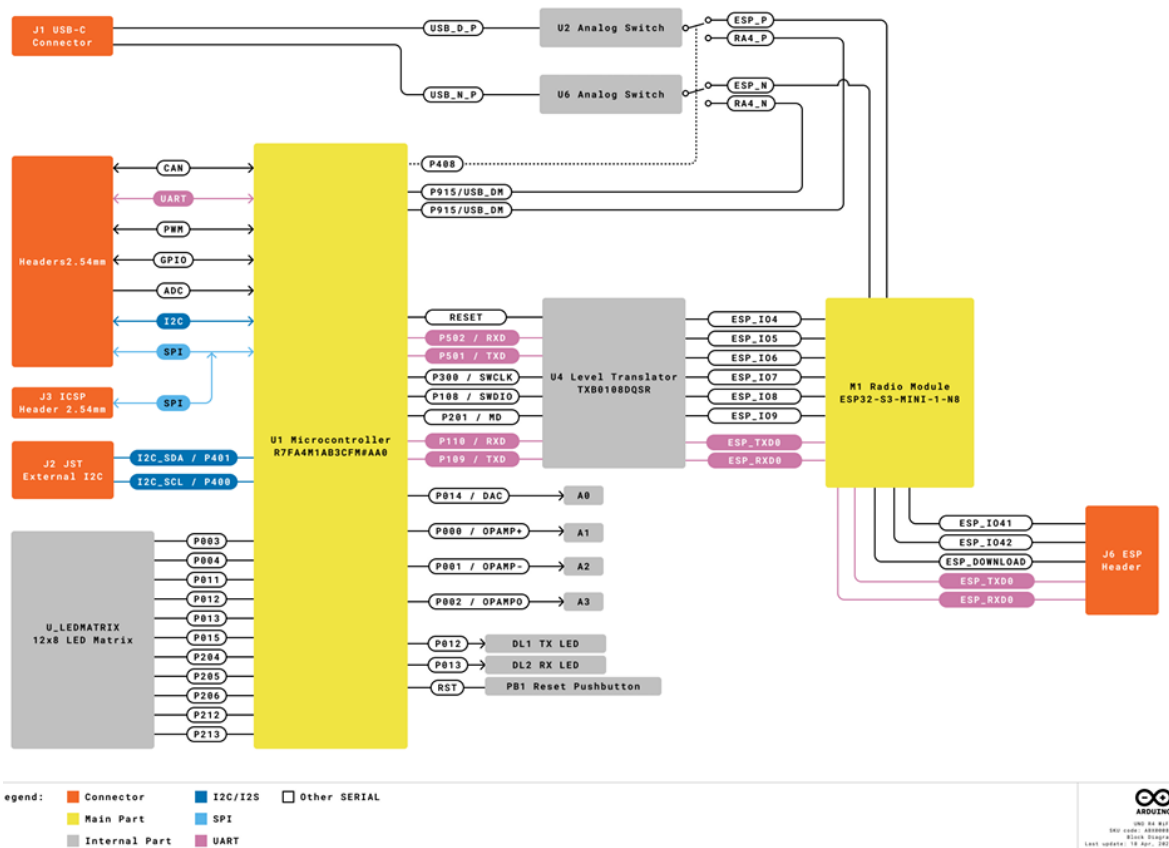
Diagrama de Conexiones de la Configuración de los Drivers y los Motores a Pasos, por pulsos negativos.



➤ **Representación Gráfica del Arduino UNO R4 Wifi**
[Arduino® UNO R4 WiFi — Arduino Online Shop](https://www.arduino.cc/en/Store/Arduino-UNO-R4-WiFi)

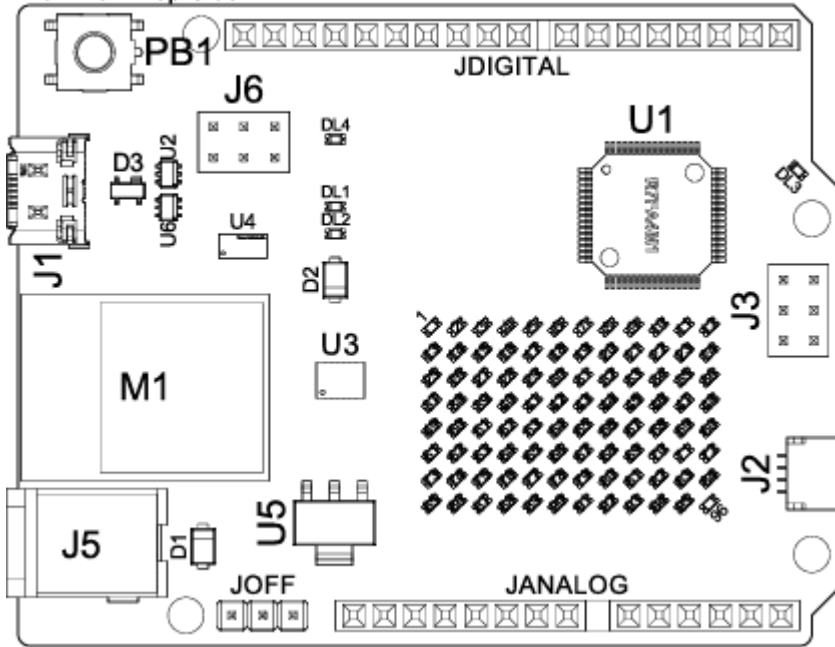


➤ **Diagrama a Bloques de la Composición del Arduino UNO R4 Wifi**

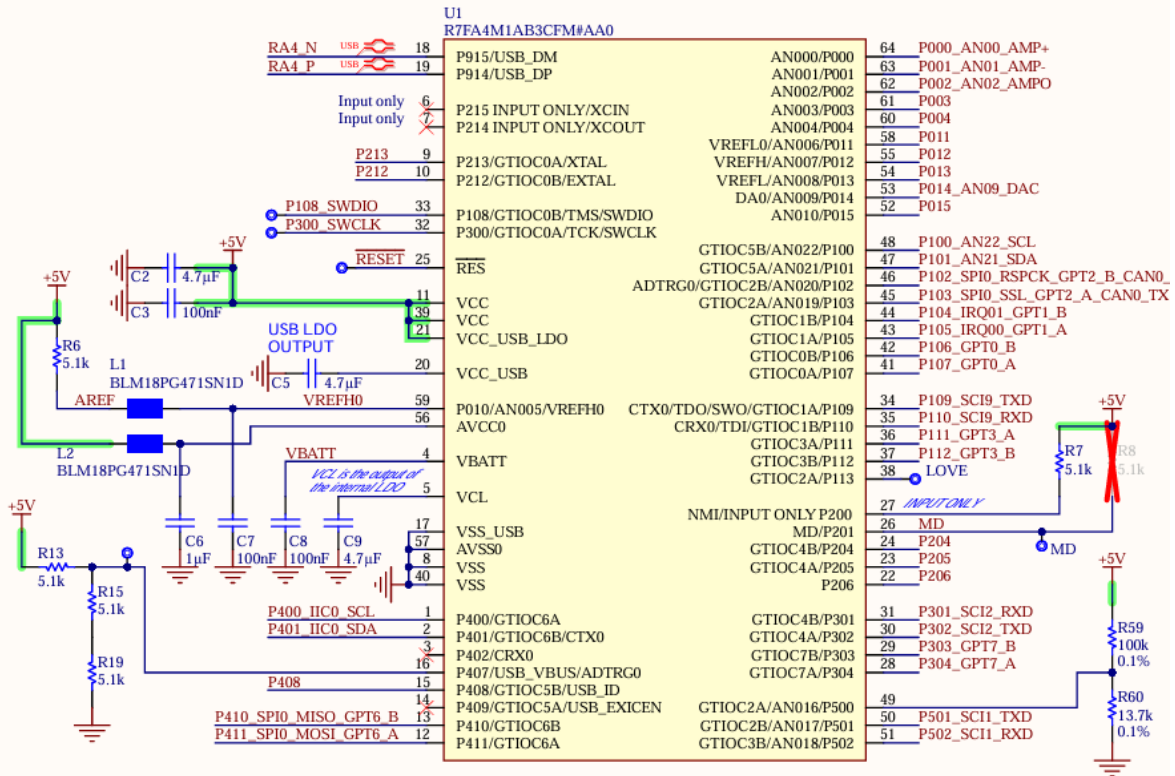


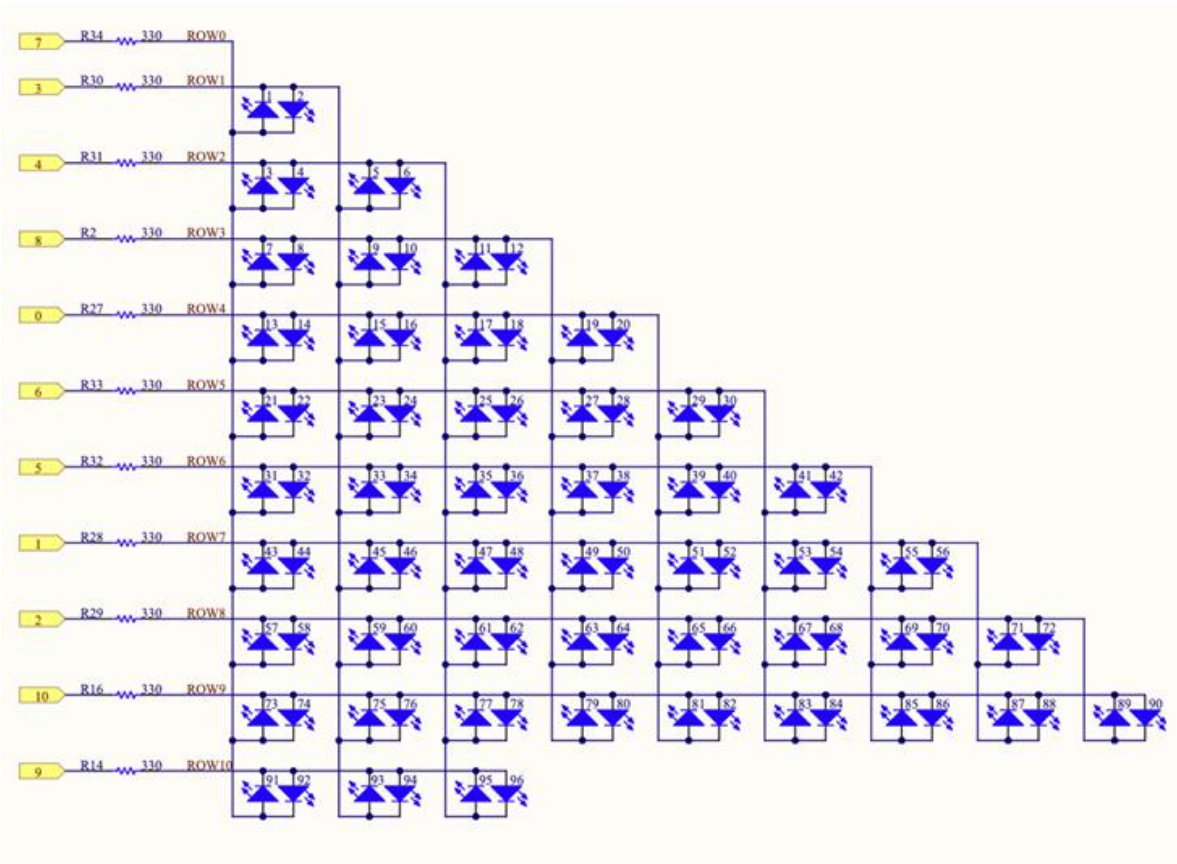
Arduino UNO R4 WiFi Block Diagram

View from Top Side



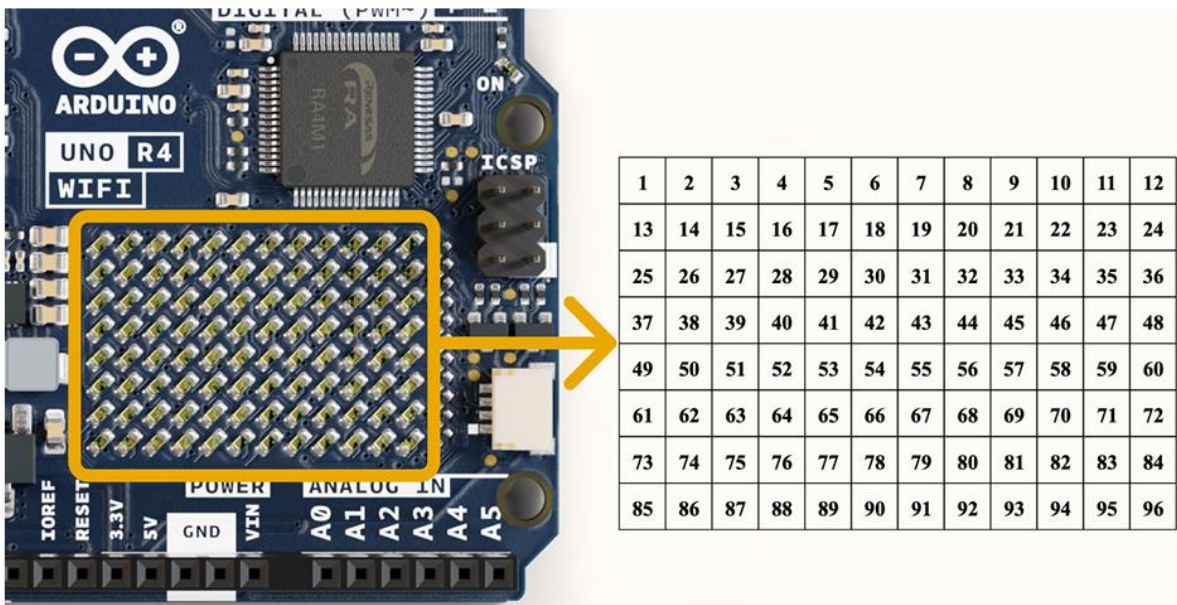
MICROCONTROLLER

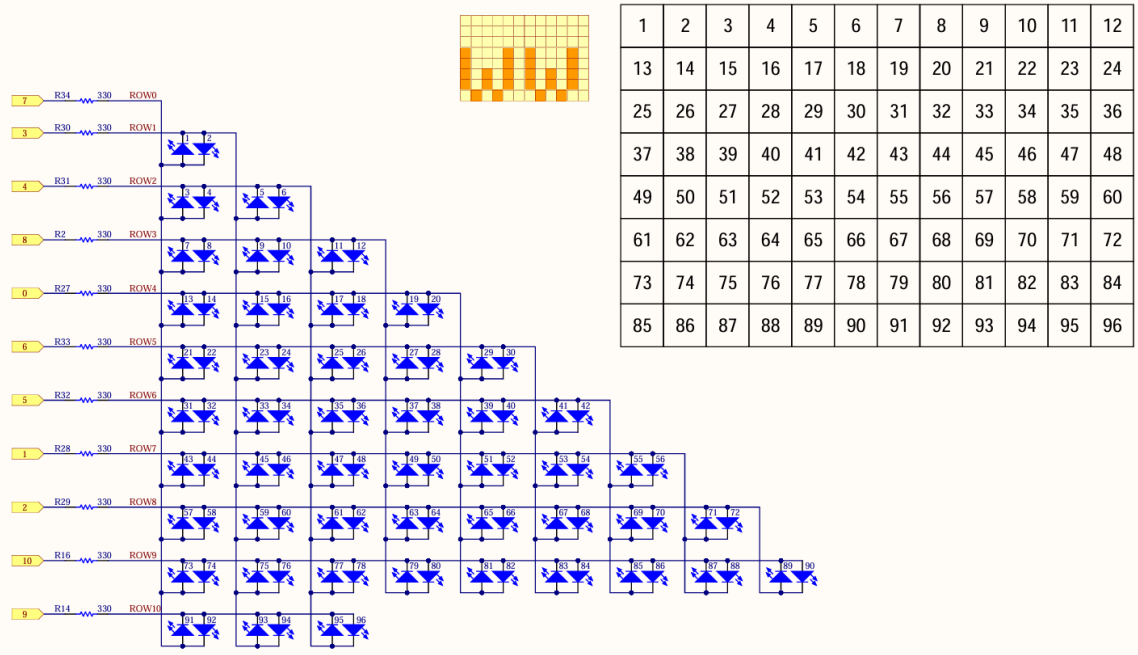




LED matrix schematics.

These LEDs can be accessed as an array, using a specific library. See the mapping below:

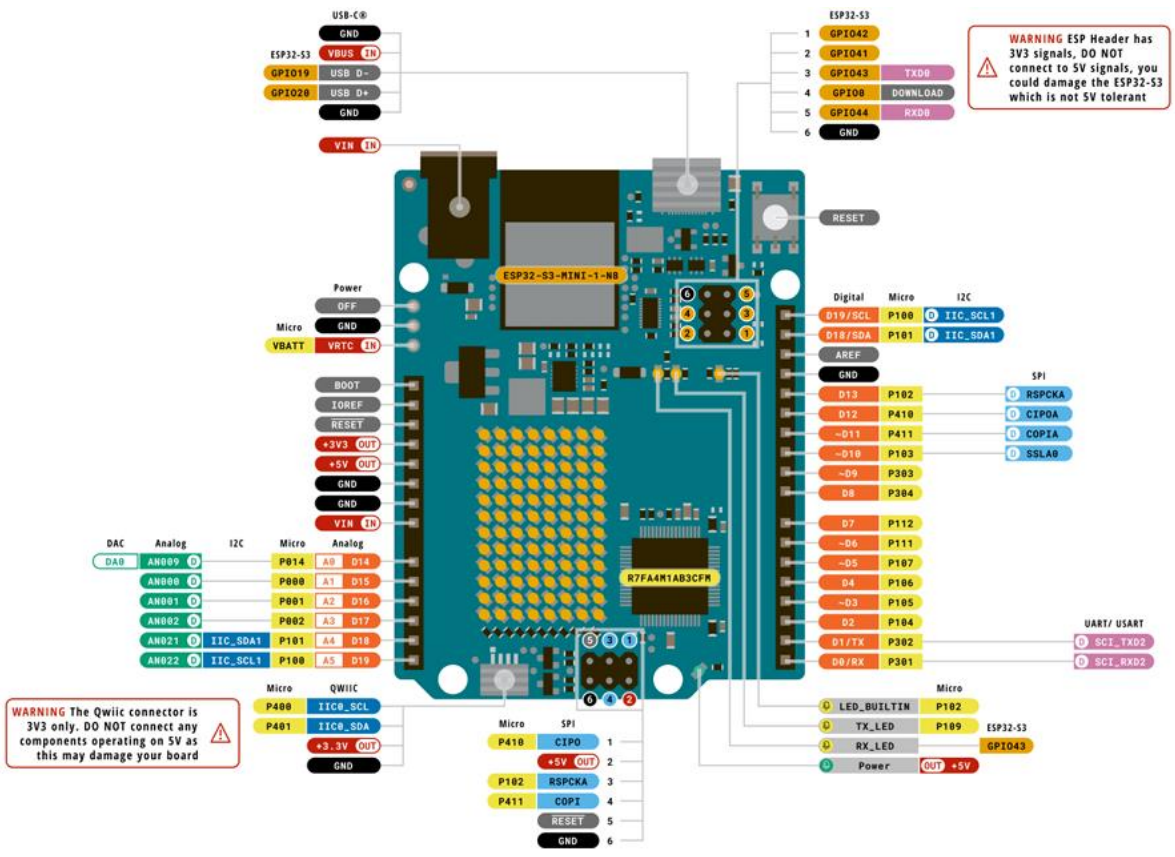




Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino SA DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
 Arduino SA may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Arduino SA reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice.
 Do not finalize a design with this site, ARDUINO and other Arduino brands and logos and Trademarks of Arduino SA. All Arduino SA Trademarks cannot be used without owner's formal permission.

Title: LED MATRIX
ID: ABX00087
Date: 23/08/2023 **Time:** 18:21:46
Version: Sheet 2 of 2
Author: Silvio Navaretti
RevAuthor: Silvio Navaretti





Pin	Function	Type	Description
1	BOOT	NC	Not Connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5 V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog	Analog input 0 / DAC
10	A1	Analog	Analog input 1 / OPAMP+
11	A2	Analog	Analog input 2 / OPAMP-
12	A3	Analog	Analog input 3 / OPAMPOut
13	A4	Analog	Analog input 4 / I2C Serial Datal (SDA)
14	A5	Analog	Analog input 5 / I2C Serial Clock (SCL)

Pin	Function	Type	Description
1	SCL	Digital	I2C Serial Clock (SCL)
2	SDA	Digital	I2C Serial Datal (SDA)
3	AREF	Digital	Analog Reference Voltage
4	GND	Power	Ground
5	D13/SCK/CANRX0	Digital	GPIO 13 / SPI Clock / CAN Receiver (RX)
6	D12/CIPO	Digital	GPIO 12 / SPI Controller In Peripheral Out
7	D11/COPI	Digital	GPIO 11 (PWM) / SPI Controller Out Peripheral In
8	D10/CS/CANTX0	Digital	GPIO 10 (PWM) / SPI Chip Select / CAN Transmitter (TX)
9	D9	Digital	GPIO 9 (PWM~)
10	D8	Digital	GPIO 8
11	D7	Digital	GPIO 7
12	D6	Digital	GPIO 6 (PWM~)
13	D5	Digital	GPIO 5 (PWM~)
14	D4	Digital	GPIO 4
15	D3	Digital	GPIO 3 (PWM~) / Interrupt Pin
16	D2	Digital	GPIO 2 / Interrupt Pin
17	D1/TX0	Digital	GPIO 1 / Serial 0 Transmitter (TX)
18	D0/TX0	Digital	GPIO 0 / Serial 0 Receiver (RX)

Pin	Function	Type	Description
1	OFF	Power	For controlling power supply
2	GND	Power	Ground
1	VRTC	Power	Battery connection to power RTC only

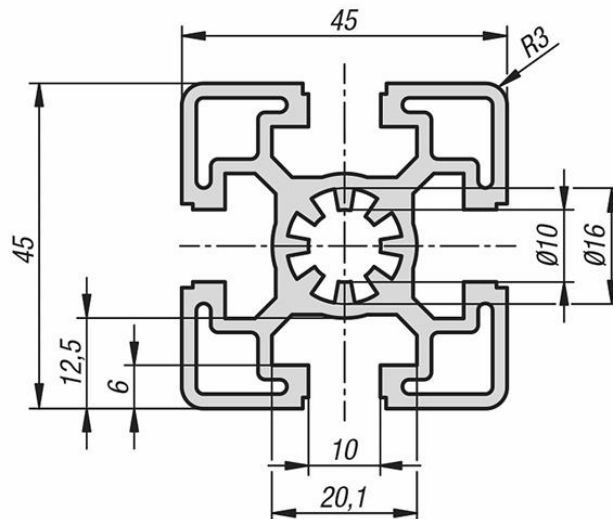
Pin	Function	Type	Description
1	CIPO	Internal	Controller In Peripheral Out
2	+5V	Internal	Power Supply of 5 V
3	SCK	Internal	Serial Clock
4	COPI	Internal	Controller Out Peripheral In
5	RESET	Internal	Reset
6	GND	Internal	Ground

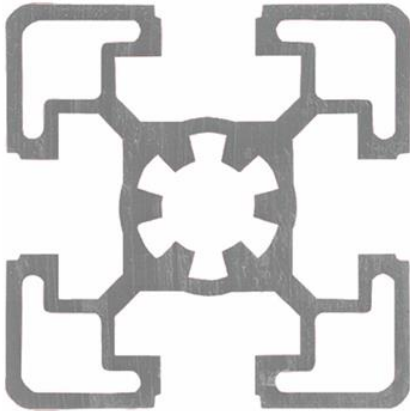
Tornillos Embalados para Máquina Herramienta de Alta Velocidad - Serie HMD

[Tornillos Embalados para Máquina Herramienta de Alta Velocidad - Serie HMD \(nskamericas.com\)](http://nskamericas.com)



[norem](http://norem.com) - Perfiles de aluminio 45x45 Tipo B





Descripción

Material:

Aluminio EN AW-6063 T66 (AlMgSi0,5 F25).

Versión:

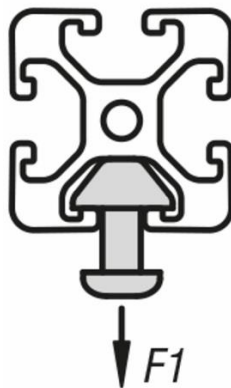
Endurecido en caliente, colores naturales anodizados.

Indicación:

Perfil de aluminio para construcciones en general.

A petición:

Cortes a inglete 15°, 30° y 45°.





UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Facultad de Ciencias
Químicas e Ingeniería

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

Programas educativos de calidad reconocidos por CIEES, CACEI y CONACYT

SGI certificado en la norma ISO 9001:2015 e ISO 21001:2018

Gestión Académica.

Jefaturas de Programa Educativo.

FORMA T-4A NOMBRAMIENTO COMITÉ REVISOR

Cuernavaca, Mor., a 17 de abril del 2024

DR. LUIS CISNEROS VILLALOBOS
DR. OUTMANE OUBRAM
DR. JOSÉ GERARDO VERA DIMAS
DRA. ALINA MARTÍNEZ OROPEZA
MTRO. FCO. JAVIER BECERRA GONZÁLEZ

P R E S E N T E

Me permito comunicarles que han sido designados integrantes del **COMITÉ REVISOR** del trabajo de **TESIS** titulada:

Diseño de un módulo de monitoreo para un laboratorio remoto mediante dispositivos IoT

Que presenta el **C. RODRIGUEZ GORDILLO ANDRES MICHEL**

Dirigido por: **DR. JOSÉ GERARDO VERA DIMAS**

Del programa educativo de: **INGENIERÍA ELÉCTRICA-ELECTRÓNICA** de la **FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA**

Para obtener el grado académico de: **LICENCIATURA EN INGENIERÍA ELÉCTRICA-ELECTRÓNICA.**

ATENTAMENTE

Por una humanidad culta

Se anexa firma electrónica

MTRA. ANGÉLICA GALINDO FLORES

DIRECTORA INTERINA DE LA FCQeI

D I C T A M E N

MTRA. ANGÉLICA GALINDO FLORES

DIRECTORA INTERINA DE LA FCQeI

P R E S E N T E

En respuesta a su amable solicitud para emitir DICTÁMEN sobre el trabajo que se menciona, me permito informarle que nuestro voto es:

VOTO	NOMBRE
APROBATORIO	DR. LUIS CISNEROS VILLALOBOS
APROBATORIO	DR. OUTMANE OUBRAM
APROBATORIO	DR. JOSÉ GERARDO VERA DIMAS
APROBATORIO	DRA. ALINA MARTÍNEZ OROPEZA
APROBATORIO	MTRO. FCO. JAVIER BECERRA GONZÁLEZ

Se anexan firmas electrónicas de cada revisor, en las cuáles se incluye la fecha y hora que se ha emitido el voto aprobatorio.

AGF/fjbg





UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

ANGELICA GALINDO FLORES | Fecha:2024-04-17 16:33:11 | FIRMANTE

Pob/wffQrrS4FwrizOsl4KmiLQv0bE/7xBJfzcwr7Q1m8D8OLqyH0d0ZHpfrefNclVcyEsjYdLc/JrykvVz+AsENFoHqjU05M92gzu1oh8Fw5D+HVndiRM5iB7GVVsY43mEbZH8QOEcdhbbZlkbNS12KbVHtr8UCPRyoZnxF6rYxOK6jgHMsQYadrR9fhyc1YfA1PeNsUtl+LPWvFiFBQ+L9Au7J4RNQSH/t5sXNPfAryt1GlicjXlgdcnZXvTMExqVEmNs2MfYcRsV9s70XL5wpFiDxKQ/7MVt013cyQowwl2Y6R01xz33as95yT63FjrYV9f2L+HDymvSTQEAg==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



[L3T1sGBay](#)

<https://efirma.uaem.mx/noRepudio/VYy20z97OLF34btmhhFOJsFnnBkfQPs0>



UAEM
RECTORÍA
2023-2029



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

ALINA MARTINEZ OROPEZA | Fecha:2024-04-18 23:44:49 | FIRMANTE

At3NIHKIPIOLgiDzZ86D/unD+8N9hfhUzCF/A0AFWgF5MQxi7/gtSxUCofeUR3P/tcqtAJWk5gXQZJPUYs7OPjImDb9hQ3aibmPt8XkFW/ILUbpNt9FJBnmXxXnNHeaPKdvCCPhCs
he/Qty81A7jL+NlenuKfyFGUW8Y2Y1TTtgXvQ6ja8dHV7sLAI5aYDW2Xfhs7Ji1sd9zDYWssGUuPCX+kQyvpchoVc12g7WtNArDCVkoSL4djdCikL+p96H7UpddJe3HsdAISBPB5O
ciEoMxvft29sMeWOWenVK2UuOwtG2Lw4jAo3duEsDCplmGdvh5/TWu7fvoR29F17L+g==

LUIS CISNEROS VILLALOBOS | Fecha:2024-04-21 21:01:20 | FIRMANTE

fnT7SG25WX+SRly9TXRvbcHL072KwCcuXGyY9PzR6b+dtDgeCW1wsJgHZTLL97u/AqhmePnA2HvUELZPgwiMY0W0fTTray/dmJTSrk/LJRGwiLzjPgX4RBtyl3bRrQ6km2ZWP
q9+K5/7U5EQyOwslqVHVzjN6f7iLuR3kWBcAT6kE/inDcgZFQxQ3HJIMX2+bSsvSSP1nUv24aJxiyw1VAku0is3OWWneVC2R2SzRKAG+Mo6CyXoHWQix447F7sajvcMtsdQk9pF
mWsqB30wCX/JqAQjPm8nf3thFeoBznLyffS5r6i+X/bmdyg9MB/dbH1WMa2PxcaBm2HP8mQBg==

FRANCISCO JAVIER BECERRA GONZALEZ | Fecha:2024-04-22 10:41:01 | FIRMANTE

UlcVmkLAX/MdbEDjyNXtqalqBV4QqE5wNLFoNvsWcmAdnOICvVUSQztXbdin2hP8GycE5iLct7ad5myo5FnhhLF54iKnNYDtjRHeGxfImBiYSOeH29ZkxINo1Hg76xBumi5/OFsKB
yP+V/w6J5MrZbDJSJ/vdss8+8AAAZgvnUXqWdUKbFMxfC4QDm78qOyCezNg6JNmaX5uom8AMhikhqe8XP1V75l6hvhvPfy7NmtRokyGdRfnKNrtKCgnUqCj28Z3s5EyXyHGo/b
OOCpns9oIh55SplFFs+GHtKD0Pwld9wt7rlsQ8eW+0e2Uk74GBgTqXELd+vkcoV53IT/4PQ==

OUTMANE OUBRAM | Fecha:2024-04-23 18:34:55 | FIRMANTE

dIK5N+V6pCN/IX9+SA54qg+pwIjtA623ZJbXZsgWFivgdgyUApVTVaX/3wNrZvhvXQrZ5d31HjK0XgOAb96jRz3ZgU3HKEWJNmlingHoA1/t1F/UbcDhVp+s3fqKI9STDIypN2tdeq
wqxMxLY+FLI/EOkhHi16U7cybkLFFJVPDcBRXvmSowNn60L0Mt+wzGTHHXGVAf1N5RdmBqmXr0qgTTPFoqhm81zxxDTvhyrITRj7VJ2uEoDA8nuQf4rgEEUEhu7APHXHb8KTU
Nrsqm9YJhu9BxPbA6XaPxZZTbvj1u271MSZy4eCdlE5M80RTIF6aEHxyDblfojUgMviJcQ==

JOSE GERARDO VERA DIMAS | Fecha:2024-04-23 18:48:39 | FIRMANTE

DJQ4ypCUp3roPi+/eTzwvfrhRyRwRc5GwK7edugeOWj6ll5hEzVyki4BGCmOxswZYWiYAJfxir4eGW/QUIWscj+6Pb4yaQgKpWY80gWejbtTnGtExOirC7DU87vZ4GzWIOvywByFc
24Elko/yx5mpzZ5VBsnZgYgp20jHej6PEiblZ9ghhsUxAmnqrkzafDiTSbDT/yVrIS5D+hEzyGCJvDb5sYku5Ygmll6Wp+5iKpvkSL7v4vAwaxyTTCcsylEwST0wkuKv0ylqjRRUB+b
e+vR0uFYO5QUHpfDwgJ1D1WglSxJG8uKkjhIF9+iJpQ/P5hQk6wDpvlk+pThIm2vQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o
escaneando el código QR ingresando la siguiente clave:



8YqwdZ70E

<https://efirma.uaem.mx/noRepudio/371niZA8cyG0KUuTMUzEntgDxlQRpiAn>



UAEM
RECTORÍA
2023-2029