



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**DISEÑO DE UN SISTEMA DE MONITOREO PARA PRODUCCIÓN LINEAL APLICADO A
PROTOTIPO DE RECHAZO POR COMPONENTES METÁLICOS**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTOR: Jaramillo Alcivar Boris Adrian
Montero Sánchez Victor Miguel
TUTOR: Ing. Gabriel Santiago García Vásquez MSc.

Guayaquil - Ecuador
2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Boris Adrián Jaramillo Alcivar** con documento de identificación N° **2300185721** y **Victor Miguel Montero Sánchez** con documento de identificación N° **0952558070**; manifestamos que:

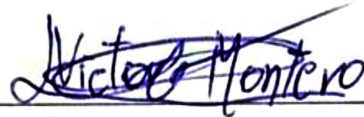
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 17 de marzo del año 2024

Atentamente,



Boris Adrián Jaramillo Alcivar
2300185721



Victor Miguel Montero Sánchez
0952558070

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A
LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Boris Adrian Jaramillo Alcivar** con documento de identificación N° 2300185721 y **Victor Miguel Montero Sánchez** con documento de identificación N° 0952558070, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **DISEÑO DE UN SISTEMA DE MONITOREO PARA PRODUCCIÓN LINEAL APLICADO A PROTOTIPO DE RECHAZO POR COMPONENTES METÁLICOS**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 17 de marzo del año 2024

Atentamente,



Boris Adrian Jaramillo Alcivar
2300185721




Victor Miguel Montero Sánchez
0952558070

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Gabriel Santiago García Vásquez**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DISEÑO DE UN SISTEMA DE MONITOREO PARA PRODUCCIÓN LINEAL APLICADO A PROTOTIPO DE RECHAZO POR COMPONENTES METÁLICOS**, realizado por **Boris Adrián Jaramillo Alcivar** con documento de identificación N° 2300185721 y **Victor Miguel Montero Sánchez** con documento de identificación N° 0952558070, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivos Tecnológicos** que cumple con todos los requisitos por la Universidad Politécnica Salesiana.

Guayaquil, 17 de marzo del año 2024

Atentamente,



Ing. Gabriel Santiago García Vásquez, Mg.
0920256500

ÍNDICE

I.	DEDICATORIA	8
II.	AGRADECIMIENTO	9
III.	RESUMEN	10
IV.	ABSTRACT	11
V.	INTRODUCCIÓN	12
VI.	PROBLEMÁTICA	13
VII.	JUSTIFICACIÓN	14
VIII.	OBJETIVOS	15
	VIII-A. Objetivo General	15
	VIII-B. Objetivos específicos	15
IX.	MATRIZ DE OBJETIVOS	16
X.	MARCO TEÓRICO REFERENCIAL	17
	X-A. Producción Continua.	17
	X-B. Sistema de monitoreo de producción.	17
	X-C. Sistema de rechazo por componentes metálicos.	17
	X-D. Control Remoto por Nodo Red.	18
	X-E. Raspberry Pi 3.	19
	X-F. Nodos en una red de comunicación.	19
	X-G. Banda transportadora.	20
	X-H. Motor eléctrico.	20
	X-I. Sensor de proximidad inductivo.	21
	X-J. Pistón neumático.	21
	X-K. Temporizador tipo relé.	22
	X-L. Relé de 5 Voltios.	22
	X-M. Relé de 24 V DC.	23
	X-N. Fuente de poder.	23
	X-Ñ. Guardamotor.	24
	X-O. Sensor infrarrojo.	24
	X-P. Encoder.	25
	X-Q. Patalla touch de 7 pulgadas.	25
	X-R. Válvula solenoide.	26
XI.	METODOLOGÍA	27
	XI-A. Sistema de monitoreo	28
	XI-A1. Proceso de instalación de Nodo-RED en Raspberry Pi	28
	XI-A2. Instalación de librería para DHT en Node-RED y Raspberry Pi	29
	XI-A3. Configuración de IP estática en Raspberry Pi 3	34
	XI-A4. Programación Node-RED - Script de nodos de función	36
	XI-A5. Script de nodos de función	45
	XI-A6. Elaboración de control remoto para la visualización de datos	47
	XI-A7. Recepción de datos en InfluxDB cloud	47

XI-A8. Comunicación InfluxDB cloud con Grafana cloud	55
XI-B. Sistema eléctrico	59
XI-C. Sistema electroneumático	61
XII. RESULTADOS	65
XII-A. Resultados diseño mecánico	65
XII-A1. Cálculos realizados	65
XIII. CRONOGRAMA Y ACTIVIDADES PARA DESARROLLAR	72
XIV. PROPUESTAS Y GASTOS	73
XV. CONCLUSIONES	74
XVI. RECOMENDACIONES	75
XVII. ANEXOS	76
XVII-A. Prototipo de banda transportadora en formato CAD	76
XVII-B. Fotografías del montaje de la banda transportadora	85
XVII-C. Fotografías del armado del panel eléctrico	90
Apéndice	99

ÍNDICE DE FIGURAS

1.	Producción continua en industria [1].	17
2.	Sistema de monitoreo. Fuente: Elaborado por autores.	17
3.	Sistema de detección de metales [43].	18
4.	Control por Nodo Red. Fuente: Elaborado por autores.	18
5.	Raspberry Pi3 [39].	19
6.	Topología de redes en estrella [31].	20
7.	Banda transportadora industrial [1].	20
8.	Motor eléctrico [17].	21
9.	Sensor inductivo [15].	21
10.	Pistón neumático [34].	22
11.	Temporizador tipo relé [22].	22
12.	Módulo relé de 5V [7].	23
13.	Relé de 24V DC [33].	23
14.	Fuente de 24V DC[28].	24
15.	Guardamotor [10].	24
16.	Sensor fotoeléctrico [24].	25
17.	Encoder [2].	25
18.	Patalla touch para Raspberry Pi [16].	26
19.	Válvula solenoide [11].	26
20.	Sistema de monitoreo de producción. Fuente: Elaborado por autores.	27
21.	Conexión de Raspberry Pi 3. Fuente: Elaborado por autores.	28
22.	Instalación de librería. Fuente: Elaborado por autores.	28
23.	Instalación de librería. Fuente: Elaborado por autores.	29
24.	Interfaz Node-RED. Fuente: Elaborado por autores.	29
25.	Node-RED Manage palette. Fuente: Elaborado por autores.	30
26.	User Settings. Fuente: Elaborado por autores.	30
27.	Instalación librería DHT. Fuente: Elaborado por autores.	31
28.	Librería DHT. Fuente: Elaborado por autores.	31
29.	Instalación librería BCM 2835. Fuente: Elaborado por autores.	31
30.	Link de librería BCM 2835. Fuente: Elaborado por autores.	32
31.	Carpeta con librería BCM 2835. Fuente: Elaborado por autores.	32
32.	Comando de configuración. Fuente: Elaborado por autores.	32
33.	Terminal de Raspberry Pi. Fuente: Elaborado por autores.	33
34.	Terminal de Raspberry Pi. Fuente: Elaborado por autores.	33
35.	Terminal de Raspberry Pi. Fuente: Elaborado por autores.	33
36.	Terminal de Raspberry Pi. Fuente: Elaborado por autores.	33
37.	Terminal de Raspberry Pi. Fuente: Elaborado por autores.	34
38.	User Settings. Fuente: Elaborado por autores.	34
39.	Interfaz Raspberry Pi 3. Fuente: Elaborado por autores.	34
40.	Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.	34
41.	Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.	35
42.	Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.	36
43.	Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.	36
44.	Inicialización de variables en Nodo-RED. Fuente: Elaborado por autores.	37
45.	Inicio y parada de proceso en Nodo-RED. Fuente: Elaborado por autores.	37
46.	Tiempos de trabajo en Nodo-RED. Fuente: Elaborado por autores.	38
47.	Función RPM en Nodo-RED. Fuente: Elaborado por autores.	39
48.	Función velocidad en Nodo-RED. Fuente: Elaborado por autores.	39
49.	Función entrada de producto en Nodo-RED. Fuente: Elaborado por autores.	40

50.	Función salida de producto en Nodo-RED. Fuente: Elaborado por autores.	41
51.	Función producto rechazado en Nodo-RED. Fuente: Elaborado por autores.	41
52.	Función calidad en Nodo-RED. Fuente: Elaborado por autores.	42
53.	Función eficiencia en Node-RED. Fuente: Elaborado por autores.	42
54.	Función rendimiento en Node-RED. Fuente: Elaborado por autores.	43
55.	Función temperatura en Nodo-RED. Fuente: Elaborado por autores.	43
56.	Función humedad en Nodo-RED. Fuente: Elaborado por autores.	43
57.	Función fecha y hora en Nodo-RED. Fuente: Elaborado por autores.	44
58.	Managge Palette. Fuente: Elaborado por autores.	48
59.	Instalación de InfluxDB cloud en nodo red. Fuente: Elaborado por autores.	48
60.	Interfaz de acceso. Fuente: Elaborado por autores.	49
61.	Requerimientos de cuenta InfluxDB cloud. Fuente: Elaborado por autores.	49
62.	Versión Influx 2.0. Fuente: Elaborado por autores.	50
63.	Condiciones. Fuente: Elaborado por autores.	50
64.	Bucket. Fuente: Elaborado por autores.	51
65.	Nombre de proyecto. Fuente: Elaborado por autores.	51
66.	API token. Fuente: Elaborado por autores.	51
67.	Acceso a Api token. Fuente: Elaborado por autores.	51
68.	Nombre de API. Fuente: Elaborado por autores.	52
69.	Dirección de token. Fuente: Elaborado por autores.	52
70.	Nodo Influx out. Fuente: Elaborado por autores.	52
71.	Datos para vinculo con InfluxDB. Fuente: Elaborado por autores.	53
72.	Acceso a coonfiguración de conexión. Fuente: Elaborado por autores.	53
73.	Versión2.0. Fuente: Elaborado por autores.	54
74.	Nombre del Bucket. Fuente: Elaborado por autores.	54
75.	Variables en InfluxDB cloud. Fuente: Elaborado por autores.	55
76.	Grafana Labs. Fuente: Elaborado por autores.	55
77.	Vinculación cuenta de Gmail. Fuente: Elaborado por autores.	55
78.	Región de implementación. Fuente: Elaborado por autores.	56
79.	Creación de dashboard. Fuente: Elaborado por autores.	56
80.	Selección de InfluxDB. Fuente: Elaborado por autores.	56
81.	Selección de visualización. Fuente: Elaborado por autores.	57
82.	Campos de vinculación con InfluxDB. Fuente: Elaborado por autores.	57
83.	Establecimiento de visualización. Fuente: Elaborado por autores.	58
84.	Selección de InfluxDB. Fuente: Elaborado por autores.	58
85.	Importado de código. Fuente: Elaborado por autores.	59
86.	Dashboard Grafana. Fuente: Elaborado por autores.	59
87.	Arranque directo de motor. Fuente: Elaborado por autores.	60
88.	Sistema electro neumático. Fuente: Elaborado por autores.	61
89.	Detección de metal. Fuente: Elaborado por autores.	62
90.	Enclavamiento de bobina -K1. Fuente: Elaborado por autores.	62
91.	Área de funcionamiento del sensor. Fuente: Elaborado por autores.	63
92.	Accionamiento de pistón. Fuente: Elaborado por autores.	63
93.	Sistema de rechazo por componentes metálicos. Fuente: Elaborado por autores.	64
94.	Banda transportadora Solid works. Fuente: Elaborado por autores.	76
95.	Plano vistas de banda transportadora. Fuente: Elaborado por autores.	77
96.	Plano larguero. Fuente: Elaborado por autores.	78
97.	Plano base. Fuente: Elaborado por autores.	79
98.	Plano soporte de guía. Fuente: Elaborado por autores.	80
99.	Plano soporte de banda. Fuente: Elaborado por autores.	81

100.	Plano rodillo de chumazera. Fuente: Elaborado por autores.	82
101.	Plano ensamble rodillo. Fuente: Elaborado por autores.	83
102.	Plano eje rodillo. Fuente: Elaborado por autores.	84
103.	Lijado de estructura. Fuente: Elaborado por autores.	85
104.	Soldadura de templadores de rodillo. Fuente: Elaborado por autores.	86
105.	Limpieza de guías. Fuente: Elaborado por autores.	86
106.	Pintado de estructura. Fuente: Elaborado por autores.	87
107.	Pintado y secado de guías para banda. Fuente: Elaborado por autores.	88
108.	Pintado y secado de soportes de guías. Fuente: Elaborado por autores.	89
109.	Conexión de componentes a la raspberry. Fuente: Elaborado por autores.	90
110.	Conexiones para prueba de arranque. Fuente: Elaborado por autores.	91
111.	Conexión de componentes en caja eléctrica. Fuente: Elaborado por autores.	92
112.	Perforación para botoneras de control. Fuente: Elaborado por autores.	93
113.	Conexión y Visualización de Pantalla touch. Fuente: Elaborado por autores.	93
114.	Implementación de pantalla y botonera en tablero. Fuente: Elaborado por autores.	94
115.	Panel de monitoreo en Node-RED. Fuente: Elaborado por autores.	94
116.	Panel de control con soporte. Fuente: Elaborado por autores.	95
117.	Instalación del panel de control. Fuente: Elaborado por autores.	96
118.	Panel de control. Fuente: Elaborado por autores.	97

ÍNDICE DE TABLAS

I.	Matriz de objetivos. Fuente: Elaborado por autores.	16
II.	Coefficientes de fricción aproximados. Fuente: http://tinyurl.com/3bbc4r39	65
III.	Factores C3. Fuente: http://tinyurl.com/3bbc4r39	66
IV.	Factores C1. Fuente: http://tinyurl.com/3bbc4r39	68
V.	Tabla de factores K. Fuente: http://tinyurl.com/3bbc4r39	70
VI.	Cronograma	72
VII.	Presupuesto de materiales y recursos. Fuente: Elaborado por autores.	73

I. DEDICATORIA

Este trabajo de titulación esta dedicado a mi madre Laura Araceli Alcivar Vera, por ser la persona que me impulso desde un principio a superarme y llegar a ser un profesional, a mi abuela Rosa Aurora Vera Rivas por su cuidado por su amor y por su paciencia en este proceso estudiantil, a mi enamorada Belén Loyda Rivera Prieto por ser la persona que siempre me impulso a seguir adelante y no rendirme en este extenuando y arduo proceso académico.

Boris Adrian Jaramillo Alcivar

Este trabajo de titulación está dedicado a mi madre María Sánchez Sacán, hermana Lady Montero Sánchez y padre Manuel Montero Buenaño que, aunque ya no nos acompañe hoy, sé que está orgulloso de verme convertido en profesional.

A mis amigos y seres queridos por su ánimo y comprensión en los momentos difíciles, de igual manera a mis profesores de la universidad por su guía y enseñanzas que han sido fundamentales en mi formación. Y finalmente, a todas aquellas personas que de alguna manera contribuyeron en este proceso. Sin su apoyo, este logro no hubiera sido posible.

Victor Miguel Montero Sánchez

II. AGRADECIMIENTO

Agradezco en primer lugar a Dios porque es quien me ha permitido llegar a este punto de mi carrera estudiantil a lo largo de estos años, a mi madre, a mi enamorada y a mi familia quienes han sido un pilar fundamental y un apoyo imprescindible en mi vida, a mis compañeros de trabajo quienes con su conocimiento me han guiado e instruido en este proceso.

Al ingeniero Gabriel García por su guía y tutelaje en este arduo proceso ya que con sus consejos y paciencia nos ha permitido terminar este trabajo de titulación.

Boris Adrian Jaramillo Alcivar

Agradezco a Dios por permitirme concluir mis estudios profesionales, a mi madre y a mi hermana que siempre han estado para apoyarme en lo que necesite, al Sr. Leonardo Quinde por su ayuda, a mi compañero de tesis por brindar buenas ideas al momento de elaborar y desarrollar nuestro prototipo.

Doy gracias a mis profesores y asesores académicos de la Universidad Politécnica Salesiana por siempre brindarme una solución oportuna en cuestiones pedagógicas.

Agradezco mis amigos Franz Baidal, Jean Pierre Vera, Boris Jaramillo y Leonardo Jara que compartieron conmigo este camino, aportando no solo conocimientos, sino también momentos de alegría, aquellos momentos que siempre se recuerdan, les deseo todo el éxito del mundo.

Victor Miguel Montero Sánchez

III. RESUMEN

La presente tesis se centra en el desarrollo y aplicación de un sistema de monitoreo de producción continua adaptado específicamente para una banda transportadora en entornos industriales. A través de la integración de tecnologías avanzadas como Raspberry Pi 3, se busca optimizar la eficiencia operativa y la calidad del producto en la cadena de producción. Se realiza un análisis exhaustivo del estado del arte en sistemas de monitoreo en bandas transportadoras, seguido de la implementación de un sistema prototipo que incluye sensores foto eléctricos e inductivos para detectar posibles fallos o anomalías en el proceso. Así como también la recopilación y almacenamiento de datos de producción en un entorno didáctico y de fácil acceso. La investigación se enfoca en la evaluación del rendimiento del sistema, tanto en términos de precisión de la monitorización como de su capacidad para identificar problemas en tiempo real. Se llevan a cabo pruebas experimentales en un entorno industrial simulado para validar la eficacia del sistema propuesto, y se proporcionan recomendaciones para su implementación y optimización en aplicaciones industriales reales. Este trabajo contribuye al avance en el campo de la automatización industrial, destacando el potencial de los sistemas de monitoreo de producción continua para mejorar la eficiencia y la calidad en la fabricación.

Palabras clave: Monitoreo de producción continua, Banda transportadora, Tecnologías avanzadas, Eficiencia operativa, Calidad del producto

IV. ABSTRACT

This thesis focuses on the development and application of a continuous production monitoring system specifically adapted for a conveyor belt in industrial environments. Through the integration of advanced technologies such as Raspberry Pi 3, the aim is to optimize operational efficiency and product quality in the production chain. An exhaustive analysis of the state of the art in conveyor belt monitoring systems is carried out, followed by the implementation of a prototype system that includes photoelectric and inductive sensors to detect possible failures or anomalies in the process. As well as the collection and storage of production data in a didactic and easily accessible environment. The research focuses on evaluating system performance, both in terms of monitoring accuracy and its ability to identify problems in real time. Experimental tests are carried out in a simulated industrial environment to validate the effectiveness of the proposed system, and recommendations are provided for its implementation and optimization in real industrial applications. This work contributes to advancement in the field of industrial automation, highlighting the potential of continuous production monitoring systems to improve efficiency and quality in manufacturing.

Keywords: Continuous production monitoring, Conveyor belt, Advanced technologies, Operational efficiency, Product quality

V. INTRODUCCIÓN

El diseño de un sistema de monitoreo para la producción lineal y su implementación en un prototipo de rechazo por componentes metálicos constituyen un aspecto crítico en la optimización de procesos industriales. En la manufactura, la eficiencia y la calidad son pilares esenciales para el éxito del producto final. Como señala [41], la integración de sistemas de monitoreo puede mejorar significativamente la calidad del producto y reducir los costos asociados con defectos y retrabajos. Detectar de manera oportuna y precisa componentes metálicos no deseados en la línea de producción es esencial para mantener estándares de calidad y seguridad [42].

La aplicación de tecnologías avanzadas de detección y seguimiento en el proceso de fabricación constituye un componente central en el diseño de estos sistemas. Según [42], el empleo de sensores y algoritmos especializados permite una vigilancia continua y en tiempo real, facilitando la identificación temprana de posibles fallos en la línea de producción. Esta afirmación sugiere que la aplicación de tecnologías innovadoras es fundamental para optimizar los procesos industriales y mejorar la competitividad en un mercado globalizado.

El prototipo de rechazo por componentes metálicos no solo garantiza la calidad del producto final, sino que también contribuye a optimizar recursos y reducir costos asociados con retrabajos y reclamaciones. De acuerdo con [42], la detección y eliminación eficientes de componentes metálicos no deseados pueden mejorar la eficacia del proceso de producción y minimizar el riesgo de daños en maquinaria o productos terminados. Por ende, estos sistemas representan una inversión estratégica con potencial para generar beneficios significativos en el ámbito industrial.

Se puede concluir que, el diseño y la implementación de un sistema de monitoreo para producción lineal, específicamente aplicado a la detección de componentes metálicos no deseados, son aspectos cruciales para garantizar la calidad, la eficiencia y la seguridad en los procesos industriales. Como sugieren las investigaciones previas, la adopción de tecnologías innovadoras y la atención a la detección temprana de fallos son fundamentales para mantener la competitividad y la excelencia en la industria manufacturera.

VI. PROBLEMÁTICA

La carencia de un espacio físico de monitoreo en las instalaciones de producción se ha convertido en un desafío crítico que impacta de forma negativa la eficiencia y la competitividad de las empresas [44]. En la era actual donde la automatización industrial ha experimentado tendencias en avances tecnológicos de manera exponencial, la ausencia de un sistema de monitoreo de los márgenes de eficiencia y productividad pueden limitar de manera significativa la capacidad de las industrias para supervisar y optimizar sus procesos en tiempo real (Smith, 2019).

En dichas industrias muchas veces los tiempos de producción planificados o establecidos quedan por debajo de los alcanzados, en varias ocasiones sin poder determinar realmente el causal de mencionados atrasos. Cuando se habla de áreas críticas como la elaboración de medicamentos puede conllevar a la pérdida de lotes completos motivados por estos problemas. Conocer las fallas y atrasos en los tiempos de producción es un requerimiento necesario para realizar los mantenimientos correctivos adecuados en los procesos industriales y lograr las metas propuestas de manera eficiente.

En las líneas de producción continua donde los aspectos de calidad son regidos por reglamentación local [29], la falta de implementación de un sistema de monitoreo representa una serie de interrupciones en la cadena de producción y disminución de la calidad del producto final. La carencia de planificación, errores de operación, imprevistos de inventario, la falta de mejora continua son las principales causantes de los tiempos de parada no programadas en las industrias [14]. El no conocer la cantidad de producto que se pierde o que la línea de producción deja de abastecer por el tiempo que se encuentre en modo inoperante genera retrasos de suministro al mercado. En consecuencia, esta serie de desafíos afecta la productividad y competitividad general de las empresas farmacéuticas en el entorno industrial actual.

Cuando en ciertas líneas de producción como en la elaboración de productos de grado alimenticio la parada no programada se extiende más de lo previsto, los riesgos de aparición de microorganismos que contaminen el producto aumentan significativamente [14]. La pérdida de producto tiene mayor impacto pues no solo se detiene la producción de este, sino que el producto semielaborado al ser muy delicado se ve afectado por agentes contaminantes, esto termina con la pérdida completa del lote actual, más el que se dejaría de producir por el tiempo inactivo de la línea.

VII. JUSTIFICACIÓN

Debido a que la optimización y control de los sistemas de producción se ha convertido en un bien desarrollado ampliamente, surge la necesidad de explorar soluciones innovadoras que permitan abordar la dificultad mencionada en la sección anterior y maximizar la eficiencia operativa en la producción [23]. La implementación de un sistema de monitoreo para obtener información en tiempo real se presenta como una oportunidad valiosa para resolver estos desafíos. El enfoque propuesto no solo permitirá la supervisión constante de los procesos de producción, sino que también facilitará la toma de decisiones basada en datos en tiempo real y la identificación temprana de problemas. Se podrá lograr una solución asequible y escalable que sea accesible para empresas de diferentes tamaños [38].

El objetivo final de este proyecto está orientado a líneas de producción continua donde los tiempos de trabajo son prolongados y los tiempos de paradas por planificación son mínimos. Mediante la implementación de un sistema de monitoreo para la producción se manejará de manera óptima y precisa la eficiencia de las máquinas, como también la velocidad real en las que ellas operan [5]. Se contabilizará por medio de datos reales la cantidad de producto que ingresa, la cantidad de producto óptimo y el producto rechazado que se tendrá en las líneas de producción. Se recibirán valores medibles de la cantidad de producto que se deja de producir en el lapso que la máquina se encuentre inactiva por problemas de parada no programada [4].

En ciertas líneas de producción continua como en la elaboración de líquidos, cuando las paradas no programadas se extienden más de lo previsto, produce pérdida de material semi elaborado, el cual podrá ser tabulado y medido mediante la información recolectada de dicha línea. Uno de los factores que determinan el rechazo de dicho producto son los contaminantes metálicos externos, debido a esto se simulará la aplicación del sistema de monitoreo en un prototipo a escala que cuente con una banda transportadora [13].

La verificación de todos los datos recolectados en las líneas de producción al que se aplique el sistema de monitoreo se visualizara por medio de un control local y un control remoto a través de dispositivos que cuenten con conexión a internet [8].

VIII. OBJETIVOS

VIII-A. *Objetivo General*

- Diseñar un sistema de monitoreo para producción lineal aplicado a un prototipo de rechazo por componentes metálicos.

VIII-B. *Objetivos específicos*

- Diseñar un sistema de monitoreo para producción lineal con un raspberry pi 3 que determine la cantidad de producto por tiempo de trabajo.
- Fabricar un prototipo a escala de una banda transportadora con un sistema para rechazo de contaminantes metálicos.
- Validar el funcionamiento del sistema de monitoreo mediante la aplicación al prototipo a escala, que permita la visualización de datos en tiempo real.

IX. MATRIZ DE OBJETIVOS

Tabla I
MATRIZ DE OBJETIVOS. FUENTE: ELABORADO POR AUTORES.

OBJETIVO	PLANTEAMIENTO	META	INDICADOR
1	Diseñar un sistema de monitoreo para producción lineal con un raspberry pi 3 que determine la cantidad de producto por tiempo de trabajo.	Un porcentaje de lectura de datos cercano al 100%. Obtener las RPM de la máquina.	Eficiencia y velocidad.
2	Fabricar un prototipo a escala de un sistema de rechazo por contaminantes metálicos en en una banda transportadora.	Sensar contaminantes externos con una frecuencia de conmutación de 4ms a una proximidad de 4mm.	Cantidad de producto optimo y rechazado con un margen de error menor al 5%
3	Validar el funcionamiento del sistema de monitoreo mediante la aplicación al prototipo a escala, el cual permitirá la visualización de datos en tiempo real.	Obtener datos en tiempo real con una latencia por debajo de los 10ms.	Tiempo de respuesta.

X. MARCO TEÓRICO REFERENCIAL

X-A. *Producción Continua.*

Es un enfoque industrial que implica la fabricación de productos de forma interrumpida, minimizando los tiempos de parada y generando un porcentaje exponencial en la mejora de la calidad y la eficiencia de los recursos utilizados (Bey, 2017). Estos protocolos permiten que la producción se lleve de forma mas fluida y se reduzca los gastos asociados con productos rechazados y detección de líneas de producción por paradas no programadas.



Figura 1. Producción continua en industria [1].

X-B. *Sistema de monitoreo de producción.*

Un sistema de monitoreo de producción se refiere a un conjunto de métodos, procesos y tecnologías utilizados para supervisar y regular eficientemente las operaciones de fabricación y flujo de trabajo en una organización [18]. El objetivo principal de un sistema de monitoreo de producción es optimizar la utilización de recursos, minimizar los tiempos de producción y maximizar la calidad del producto final, todo ello mientras se cumplen los objetivos y la demanda del mercado.



Figura 2. Sistema de monitoreo. Fuente: Elaborado por autores.

X-C. *Sistema de rechazo por componentes metálicos.*

El sistema de rechazo esta conformado por un sensor de proximidad de tipo inductivo el cual detectara a milímetros de distancia el campo electromagnético que generan los contaminantes externos de tipo metálico, enviará

una señal a un actuador posicionado en la línea de producción el cual activará el sistema electroneumático enviando el producto que el sensor inductivo calificó como deplorable a otro lugar, donde un sensor de presencia tipo infrarrojo lo contabilizará como producto rechazado y enviará esta señal para que el Raspberry Pi lo interprete como un dato y sea visualizado en el control remoto [3].

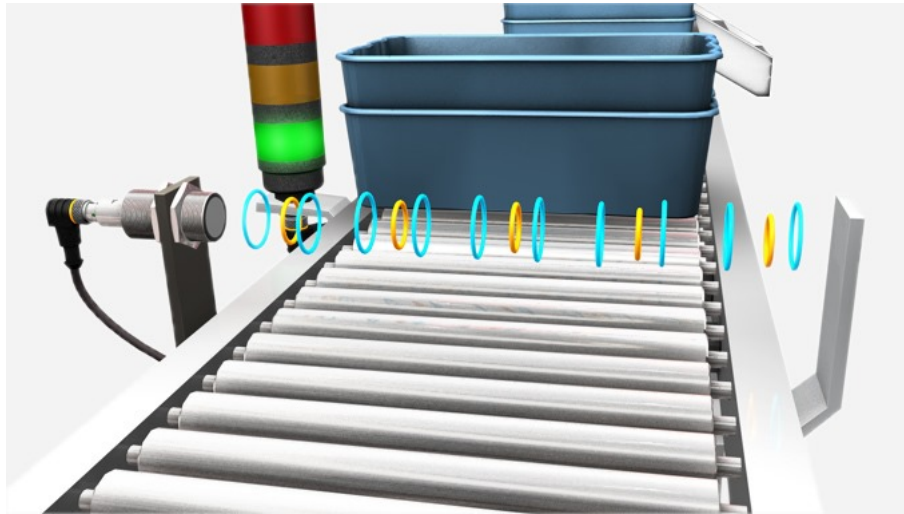


Figura 3. Sistema de detección de metales [43].

X-D. Control Remoto por Nodo Red.

Un sistema de monitoreo con nodo red es una infraestructura de seguimiento y control que utiliza nodos de red interconectados para recopilar, transmitir y analizar datos en tiempo real de dispositivos o sensores distribuidos en una red. Estos nodos, que pueden incluir sensores, controladores y otros dispositivos conectados, colaboran para recopilar información valiosa sobre el estado y el rendimiento de una red o sistema.

La información recopilada se transmite a través de la red a un punto central o un sistema de gestión, donde se procesa y analiza para tomar decisiones informadas y, en algunos casos, realizar ajustes automáticos. Este enfoque permite una supervisión constante y la toma de decisiones pro activas, lo que es esencial en aplicaciones que abarcan desde la gestión de la infraestructura urbana hasta la supervisión de sistemas industriales (Application, 2014).

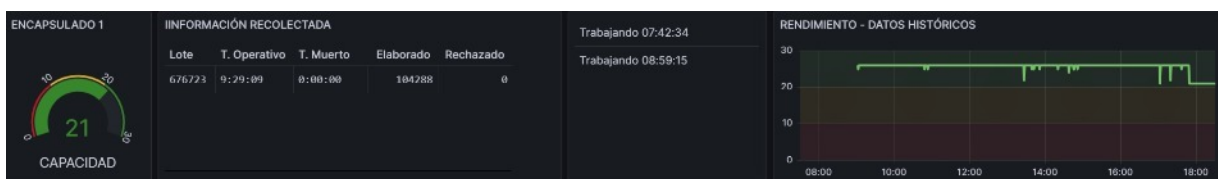


Figura 4. Control por Nodo Red. Fuente: Elaborado por autores.

X-E. Raspberry Pi 3.

Es un controlador pequeño semejante a una placa de tipo Arduino que se emplea regularmente para construir proyectos de tipo electrónico. Es una herramienta con mayor alcance en términos de potencia y su función es generalmente para realizar proyectos que requieren mayor eficiencia y mejor memoria de procesamiento, este dispositivo al ser un microcontrolador requiere de un sistema operativo que ayude a su respectiva funcionalidad y mediante el sistema operativo se puede utilizar al Raspberry Pi como un PC u ordenador de mesa al que se puede conectar periféricos para facilitar su manejo.[26].

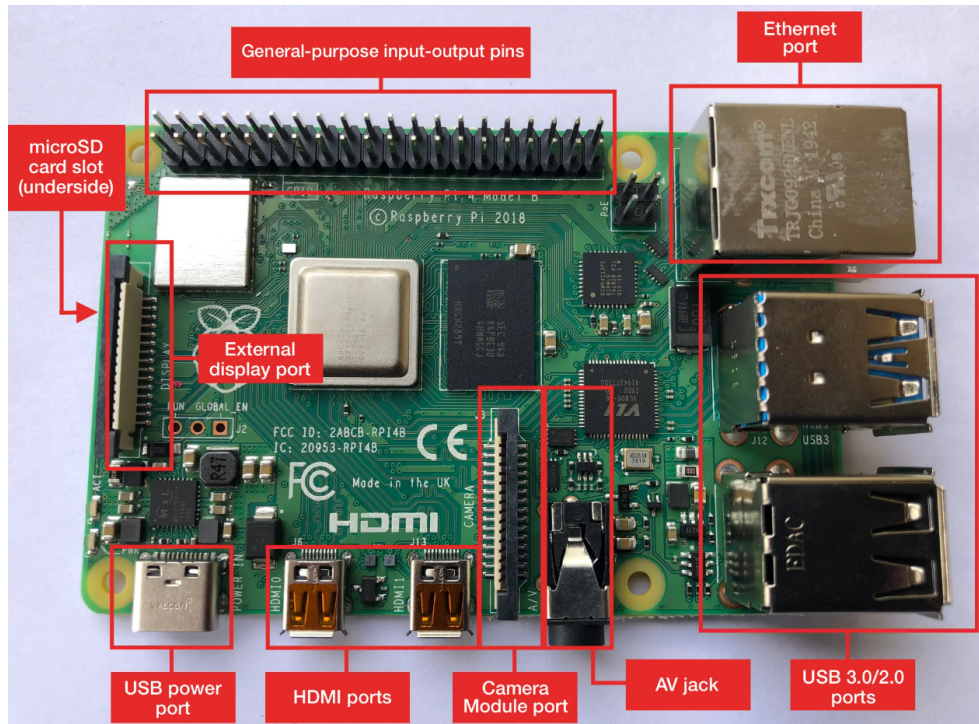


Figura 5. Raspberry Pi3 [39].

X-F. Nodos en una red de comunicación.

Un nodo red se refiere a cualquier dispositivo o entidad dentro de una red de comunicación que tiene la capacidad de recibir, enviar o retransmitir datos, así como realizar funciones de enrutamiento y procesamiento en el contexto de intercambio de información entre diversos puntos dentro de una red. Estos nodos pueden incluir computadoras, servidores, enrutadores, conmutadores y otros dispositivos de red interconectados que colaboran en la transferencia eficiente de datos. La noción de nodos de red es fundamental para la comprensión de la arquitectura y operación de sistemas de comunicación en red, ya que representan los puntos activos y funcionales que constituyen la infraestructura de interconexión. [21].

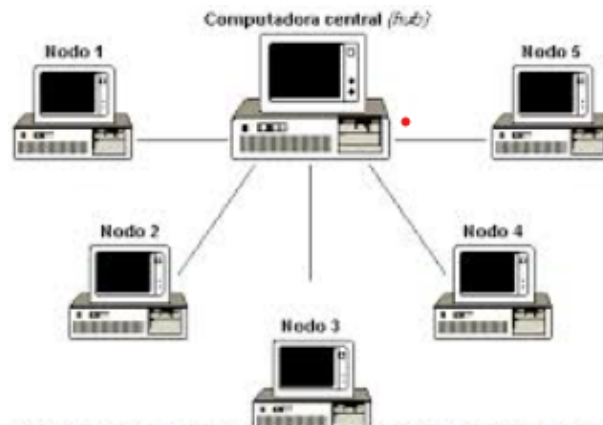


Figura 6. Topología de redes en estrella [31].

X-G. Banda transportadora.

Las bandas son elementos de transmisión motivados por un transmisor de potencia cuyo funcionamiento es movilizar de un punto de referencia a otro un sin número de materiales dependiendo de la capacidad y las especificaciones de los componentes de la banda.

Son aparatos que funcionan de forma autónoma, interconectados en las líneas de producción de forma autónoma, sin la necesidad que un operador que la manipule de forma constante. Su prioridad es operar de forma continua y sin interrupciones en aquellos procesos industriales que pueden concatenarse entre varios metros e incluso kilómetros. [36].



Figura 7. Banda transportadora industrial [1].

X-H. Motor eléctrico.

Es un componente que proporciona energía mecánica por medio de alimentación eléctrica a través de una bobina, por medio de movimientos rotatorios genera varios tipos de trabajo en las diferentes áreas en el cual son empleados. Los motores de tipo síncronos funcionan con corriente alterna y la rotación del eje está vinculada a la frecuencia de la corriente con la que se alimenta. [12].



Figura 8. Motor eléctrico [17].

X-I. Sensor de proximidad inductivo.

Son detectores de objetos metálicos en áreas de exploración, normalmente pequeñas. El tamaño del sensor determina la distancia de activación, la cual es comúnmente de dimensiones reducidas. Cuando un componente metálico se introduce en el área del sensor, se originan disturbios de histéresis en el objeto, generando un flujo oscilante en la bobina que emite un campo electromagnético que abarca el área activa del sensor. Un núcleo férnico conduce y configura este campo. Los desgastes energéticos causadas por flujo de corrientes parasitarias se incrementan hasta que el amplificador deja de suministrar suficiente energía para retener la oscilación, lo que lleva al colapso del campo electromagnético. El circuito del sensor detecta este cambio de amplitud, produciendo una señal conmutativa en la salida entre los estados.[9]



Figura 9. Sensor inductivo [15].

X-J. Pistón neumático.

Los pistones neumáticos son elementos neumáticos que tienen como principal tarea convertir la energía neumática en trabajo mecánico de movimiento lineal, dicho accionamiento abarca una carrera de avance y retroceso.

Generalmente están compuestos por un tubo circular cerrado en sus extremos con dos tapas, entre las que se desliza un émbolo separando dos cámaras.

Según el retroceso de su vástago, los pistones neumáticos se dividen en 2 grandes grupos: Pistones de simple efecto y pistones de doble efecto [37].



Figura 10. Pistón neumático [34]

X-K. Temporizador tipo relé.

Un temporizador es un tipo de relé con retardo, es decir, desde que se aplica el impulso al temporizador hasta que este cierra los contactos pasará un tiempo determinado [6]. Es un conmutador que censará una entrada y posterior a un tiempo establecido se cerrará y producirá una salida, ayuda a regular procesos en automatización industrial de forma que abre y cierra contactos por un periodo de tiempo determinado [27].



Figura 11. Temporizador tipo relé [22].

X-L. Relé de 5 Voltios.

Un relé de 5V es un componente clave en circuitos electrónicos que permite controlar el flujo de corriente eléctrica utilizando una tensión de alimentación de 5 Voltios, gracias a su funcionamiento electromecánico basado en la activación de una bobina mediante corriente eléctrica [25].



Figura 12. Módulo relé de 5V [7].

X-M. Relé de 24 V DC.

Un relé de 24V DC es un dispositivo electromecánico utilizado en circuitos eléctricos para controlar el flujo de corriente con una tensión de alimentación de 24 Voltios de corriente continua (DC) .

Son conmutadores que comúnmente son empleados en el sector industrial para la activación de objetos específicos a ese voltaje.[35].



Figura 13. Relé de 24V DC [33].

X-N. Fuente de poder.

Una fuente de poder de 24V DC es un dispositivo electrónico diseñado para transformar una entrada de corriente alterna (AC) o corriente continua (DC) en una salida de corriente continua con un voltaje constante de 24 voltios. Esta fuente de alimentación es comúnmente utilizada en una amplia gama de aplicaciones industriales, comerciales y residenciales donde se requiere un suministro estable y confiable de energía eléctrica de 24 voltios [30] .

24V 20A



Figura 14. Fuente de 24V DC[28].

X-Ñ. Guardamotor.

Un guardamotor es un dispositivo electromecánico de protección utilizado en sistemas eléctricos para detectar sobrecargas y cortocircuitos en motores eléctricos. Su función principal es interrumpir automáticamente el suministro de energía eléctrica al motor en caso de que se detecten condiciones anormales que puedan dañarlo o poner en peligro la integridad del sistema eléctrico en general [20].



Figura 15. Guardamotor [10].

X-O. Sensor infrarrojo.

Un sensor fotoeléctrico es un dispositivo electrónico utilizado para detectar la presencia, ausencia o características de objetos mediante la emisión y recepción de luz. Funciona emitiendo un haz de luz y detectando cualquier cambio en la cantidad de luz recibida, lo que permite determinar la presencia o movimiento de objetos en su campo de detección[40].

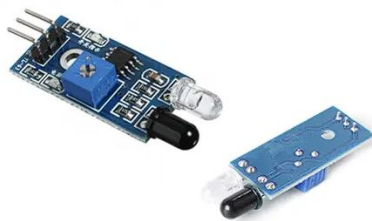


Figura 16. Sensor fotoeléctrico [24].

X-P. *Encoder.*

Un encoder es un dispositivo electromecánico utilizado para convertir el movimiento angular o lineal en señales eléctricas digitales que representan la posición, velocidad o dirección del movimiento. Funciona mediante la utilización de un disco codificado o una tira de código que, al girar o moverse, activa sensores para generar pulsos eléctricos, los cuales son interpretados por un sistema de control para determinar la posición o movimiento del objeto al que está acoplado.[32].

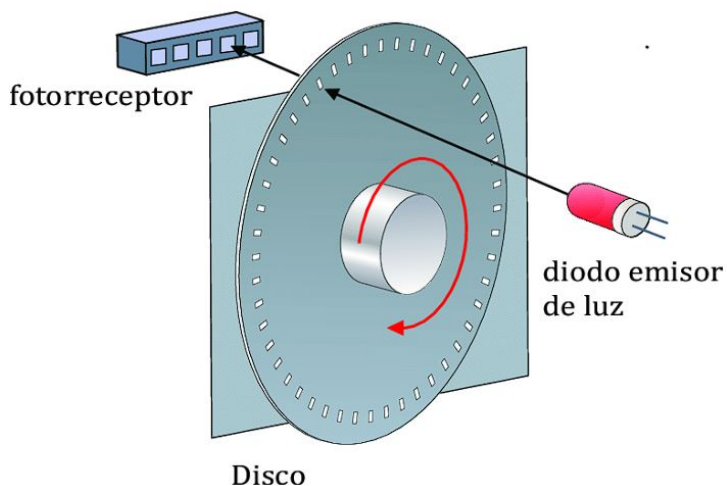


Figura 17. Encoder [2].

X-Q. *Patalla touch de 7 pulgadas.*

Una pantalla táctil de 7 pulgadas es un dispositivo de visualización electrónico que permite la interacción directa mediante el tacto con su superficie para controlar funciones o ejecutar acciones en un sistema electrónico. Estas pantallas utilizan tecnología capacitiva o resistiva para detectar la ubicación y la presión del contacto, permitiendo al usuario manipular objetos virtuales o seleccionar opciones mediante gestos táctiles.[19].

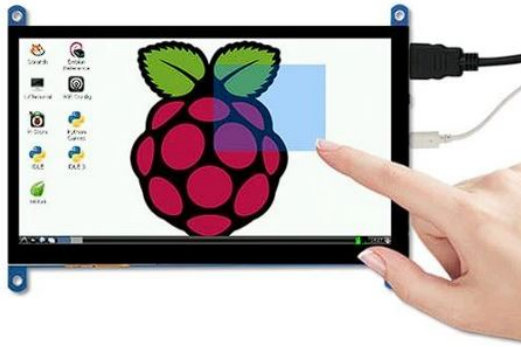


Figura 18. Patalla touch para Raspberry Pi [16].

X-R. *Válvula solenoide.*

Son componentes de distribución que cuentan con aberturas de escape separados por un tubo conductor en la entrada, se activan a través de diferentes componentes eléctricos o neumáticos y su curvatura ayuda a determinar la potencia que produce [13].



Figura 19. Válvula solenoide [11].

XI. METODOLOGÍA

SISTEMAS INVOLUCRADOS

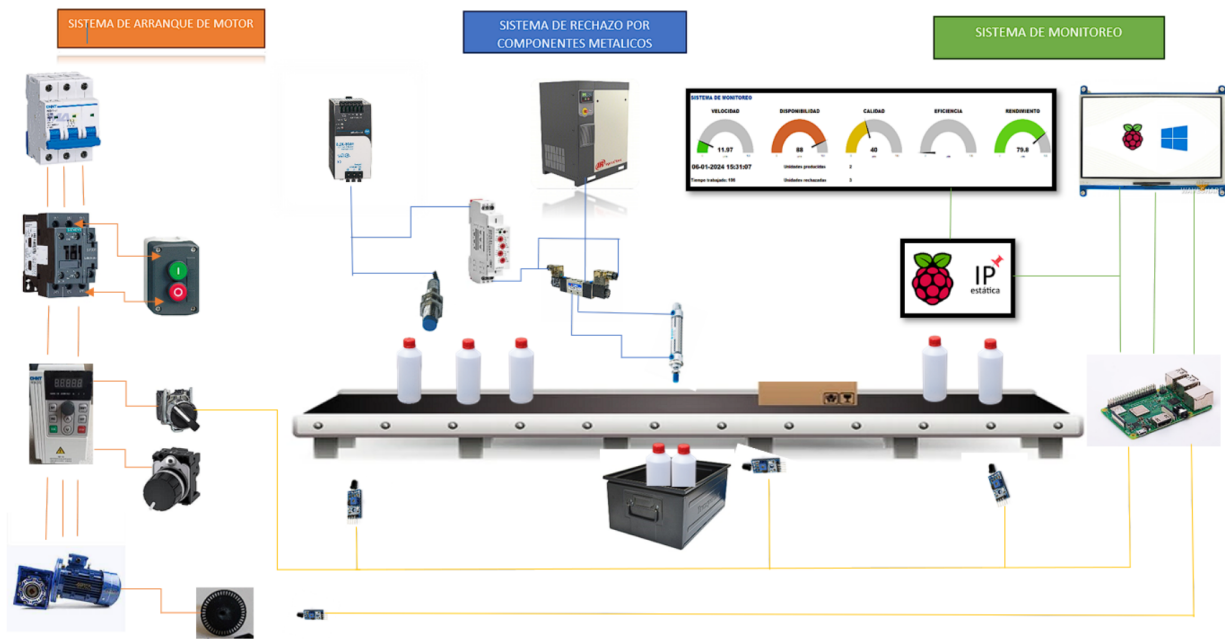


Figura 20. Sistema de monitoreo de producción. Fuente: Elaborado por autores.

XI-A. Sistema de monitoreo

EL sistema de monitoreo consta de un Raspberry Pi 3, el cual se encargará de ejecutar el programa Node-RED donde se pueden visualizar los datos de velocidad, rendimiento, eficiencia, disponibilidad y calidad del sistema en tiempo real. Para lograr observar estos valores a través de control local, se conectará una pantalla LCD de 7 pulgadas, para manejar el software se le colocará un mouse y un teclado al Raspberry Pi. Si se desea visualizar los datos del control remoto se deberá ejecutar el programa a través de una IP fija.

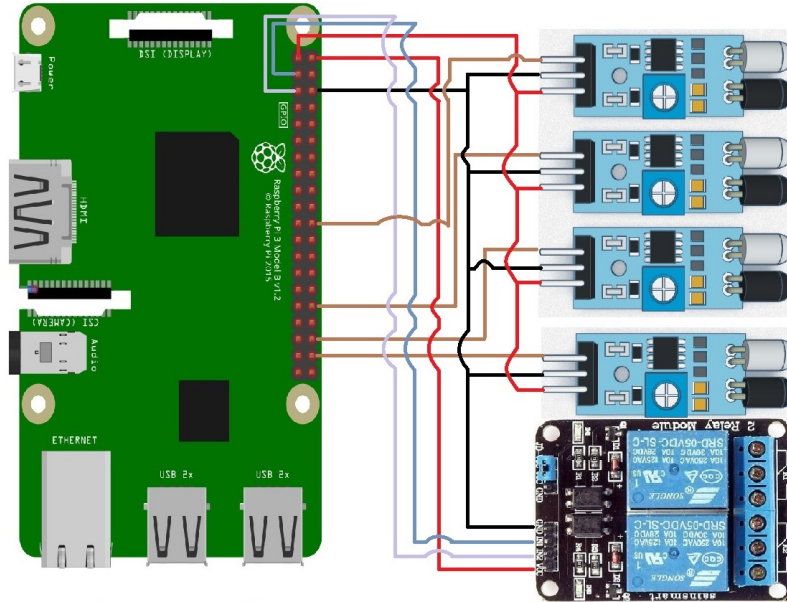


Figura 21. Conexión de Raspberry Pi 3. Fuente: Elaborado por autores.

Como se puede observar en la “figura 21”, se han conectado 6 pines hacia 4 sensores infrarrojo y 2 relés de 5V. Los pines GPIO 2, GPIO 3, son declarados como salidas digitales en Node-RED, las mismas serán utilizadas para controlar la marcha y paro de la banda transportadora a través de los relés de 5V.

Por otro lado los pines GPIO 17, GPIO 12, GPIO 16, GPIO 20, son declarados como entradas digitales para leer los valores de los sensores infrarrojo, dichos receptores nos entregaran valores de: “entrada de producto”, “producto rechazado”, “salida de producto” y “rpm del motor”.

XI-A1. Proceso de instalación de Nodo-RED en Raspberry Pi: Para la instalación de nodo red en el Raspberry Pi se utilizó la siguiente librería:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux  
-installers/master/deb/update-nodejs-and-nodered)
```

```
#!/bin/bash  
# Copyright 2016,2023 JS Foundation and other contributors, https://js.foundation/  
# Copyright 2015,2016 IBM Corp.  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at
```

Figura 22. Instalación de librería. Fuente: Elaborado por autores.

```

}
function HAS_NPM {
  if [ -x "$(command -v npm)" ]; then return 0; else return 1; fi
}
# check for apt and systemctl (set flags for later use and log if not found)
if [ -x "$(command -v apt)" ]; then
  APTOK=true;
else
  APTOK=false
  echo "apt not found. Node/npm install will be skipped" | $SUDO tee -a /var/log/nodered-install.log >>/dev/null
fi
if [ -x "$(command -v systemctl)" ]; then
  SYSTEMDOK=true;
else
  SYSTEMDOK=false
  echo "systemctl not found. shortcuts/services setup will be skipped" | $SUDO tee -a /var/log/nodered-install.log >>/dev/null
fi

```

Figura 23. Instalación de librería. Fuente: Elaborado por autores.

Después de la instalación de Node-RED en raspberry pi, se procede a copiar la IP del Raspberry Pi acompañado de su puerto de comunicación entre el Raspberry Pi y el internet. Para luego pegarlo en el navegador web de la siguiente manera:

`http://192.168.0.102:1880/`

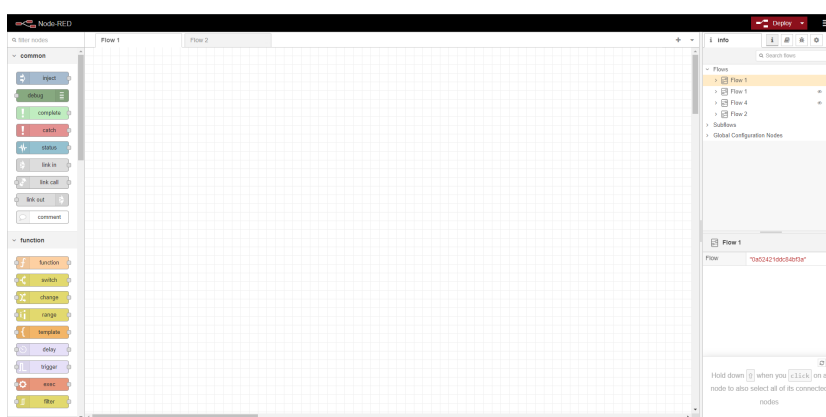


Figura 24. Interfaz Node-RED. Fuente: Elaborado por autores.

XI-A2. Instalación de librería para DHT en Node-RED y Raspberry Pi:

- 1.- Ingresar a Node-RED.
- 2.- Dirigirse a la ventana Manage palette and abrir la ventana Install.

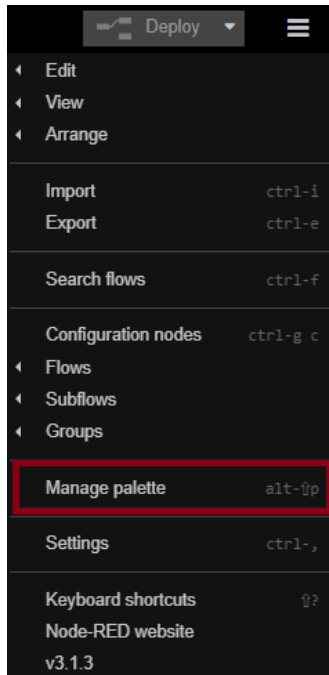


Figura 25. Node-RED Manage palette. Fuente: Elaborado por autores.

3.- Dentro de la ventana Install escribir “dht” y seleccionar la flecha como se muestra en la “figura 26”.

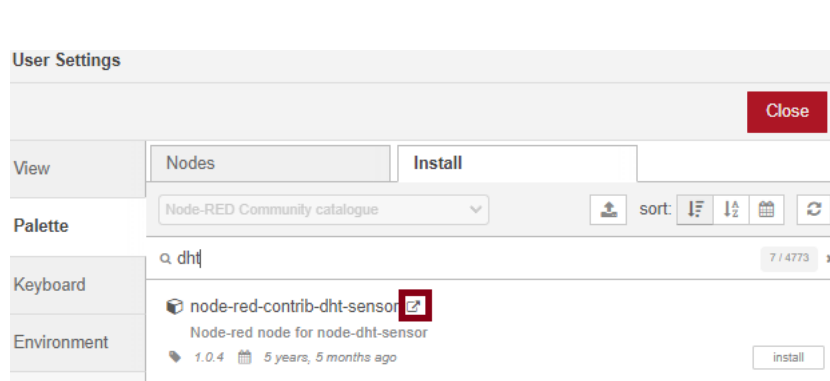


Figura 26. User Settings. Fuente: Elaborado por autores.

4.- Copiar el código que se muestra en la “figura 27” y pegarlo en un nuevo terminal de Raspberry Pi como se observa en la “figura 28”.

Installation

This node requires that `node-dht-sensor` is installed and accessible from Node Red. Since this requires access to low-level parts of the Raspberry Pi, you must run your flow with superuser access.

Installing this node requires three steps:

- Install the BCM2835 library from [here](#).
- Install the `node-dht-sensor` dependency

```
sudo npm install --unsafe-perm -g node-dht-sensor
```

Figura 27. Instalación librería DHT. Fuente: Elaborado por autores.

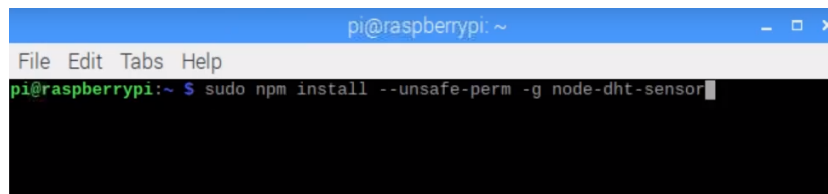


Figura 28. Librería DHT. Fuente: Elaborado por autores.

5.- Una vez instalada la librería, se hace clic en el botón [Here](#) como se ilustra en la “figura 29”.

Installation

This node requires that `node-dht-sensor` is installed and accessible from Node Red. Since this requires access to low-level parts of the Raspberry Pi, you must run your flow with superuser access.

Installing this node requires three steps:

- Install the BCM2835 library from [here](#).
- Install the `node-dht-sensor` dependency

```
sudo npm install --unsafe-perm -g node-dht-sensor
```

Figura 29. Instalación librería BCM 2835. Fuente: Elaborado por autores.

6.- Se descarga la carpeta comprimida haciendo clic en el enlace.

C library for Broadcom BCM 2835 as used in Raspberry Pi

This is a C library for Raspberry Pi (RPi). It provides access to GPIO and other IO functions on the Broadcom BCM 2835 chip, as used in the Raspberry 26 pin IDE plug on the RPi board so you can control and interface with various external devices.

It provides functions for reading digital inputs and setting digital outputs, using SPI and I2C, and for accessing the system timers. Pin event detection is supported).

It is C++ compatible, and installs as a header file and non-shared library on any Linux-based distro (but clearly is no use except on Raspberry Pi or any other Linux-based distro).

The version of the package that this documentation refers to can be downloaded from <http://www.airspayce.com/mikem/bcm2835/bcm2835-1.56.tar.gz> or <http://www.airspayce.com/mikem/bcm2835>

Figura 30. Link de librería BCM 2835. Fuente: Elaborado por autores.

7.- Una vez descomprimida la carpeta, se procede a entrar en la carpeta desde un nuevo terminal como se aprecia en la “figura 31”.

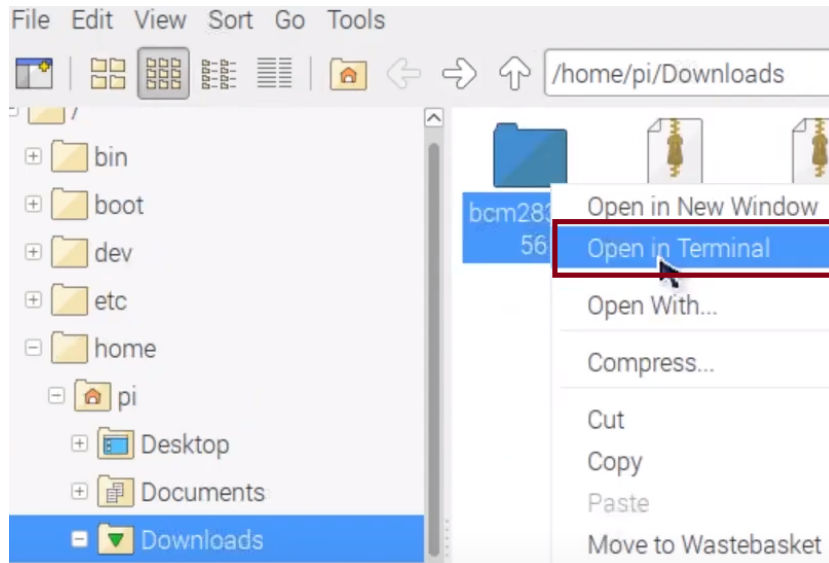


Figura 31. Carpeta con librería BCM 2835. Fuente: Elaborado por autores.

8.- Se copia y pega el comando en el terminal de la forma como se observa en las “figuras 32 y 33”.

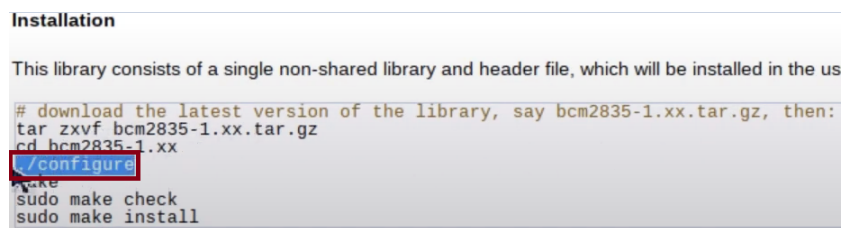


Figura 32. Comando de configuración. Fuente: Elaborado por autores.

```
pi@raspberrypi: ~/Downloads/bcm2835-1.56
File Edit Tabs Help
pi@raspberrypi:~/Downloads/bcm2835-1.56 $ ./configure
```

Figura 33. Terminal de Raspberry Pi. Fuente: Elaborado por autores.

9.- Luego se escribe “make” en el terminal de Raspberry Pi.

```
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating doc/Makefile
config.status: creating config.h
config.status: executing depfiles commands
pi@raspberrypi:~/Downloads/bcm2835-1.56 $ make
```

Figura 34. Terminal de Raspberry Pi. Fuente: Elaborado por autores.

10.- Después se ingresa “sudo make check” como se muestra en la “figura 35”.

```
make[2]: Leaving directory '/home/pi/Downloads/bcm2835-1.56'
make[1]: Leaving directory '/home/pi/Downloads/bcm2835-1.56'
pi@raspberrypi:~/Downloads/bcm2835-1.56 $ sudo make check
```

Figura 35. Terminal de Raspberry Pi. Fuente: Elaborado por autores.

11.- Para el siguiente paso se digita “sudo make install”.

```
make[1]: Entering directory '/home/pi/Downloads/bcm2835-1.56'
make[1]: Leaving directory '/home/pi/Downloads/bcm2835-1.56'
pi@raspberrypi:~/Downloads/bcm2835-1.56 $ sudo make install
```

Figura 36. Terminal de Raspberry Pi. Fuente: Elaborado por autores.

12.- Se vuelve a realizar el paso 4 para terminar con la instalación de la librería en la Raspberry Pi.

```
make[2]: Leaving directory '/home/pi/Downloads/bcm2835-1.56'
make[1]: Leaving directory '/home/pi/Downloads/bcm2835-1.56'
pi@raspberrypi:~/Downloads/bcm2835-1.56 $ sudo npm install --unsafe-perm -g node-dht-sensor
```

Figura 37. Terminal de Raspberry Pi. Fuente: Elaborado por autores.

13.- Finalmente hacer clic en Install de la ventana node-red-contrib-dht-sensor, como se observa en la “figura 38”, este paso es para instalar la librería dentro de Node-RED.

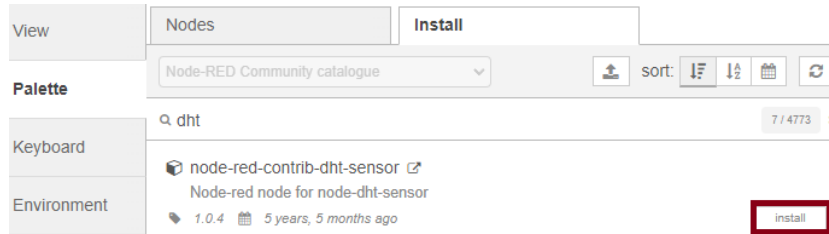


Figura 38. User Settings. Fuente: Elaborado por autores.

XI-A3. Configuración de IP estática en Raspberry Pi 3:

- Abrir el terminal de la Raspberry Pi.



Figura 39. Interfaz Raspberry Pi 3. Fuente: Elaborado por autores.

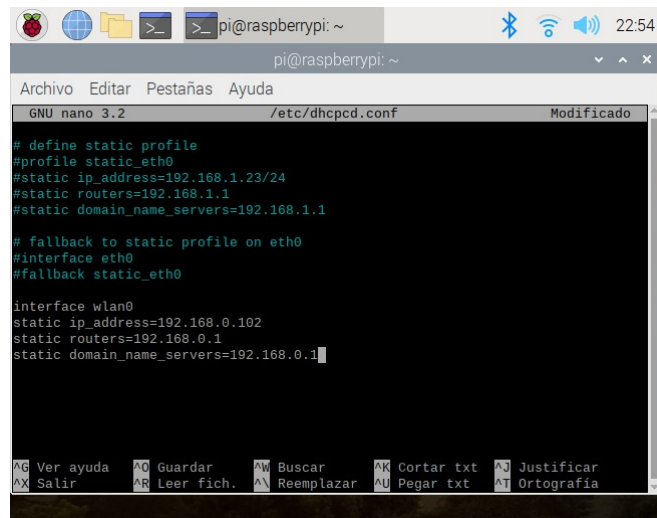
- Escribir el siguiente comando dentro del terminal para editar el archivo de configuración de red:
`sudo nano /etc/dhcpd.conf`

```
pi@raspberrypi:~ $ sudo nano /etc/dhcpd.conf
```

Figura 40. Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.

- Se debe desplazar hasta el final del archivo y agregar las siguientes líneas (reemplazar los valores con los de su propia red):

```
interface wlan0
static ip_address=192.168.0.102/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```



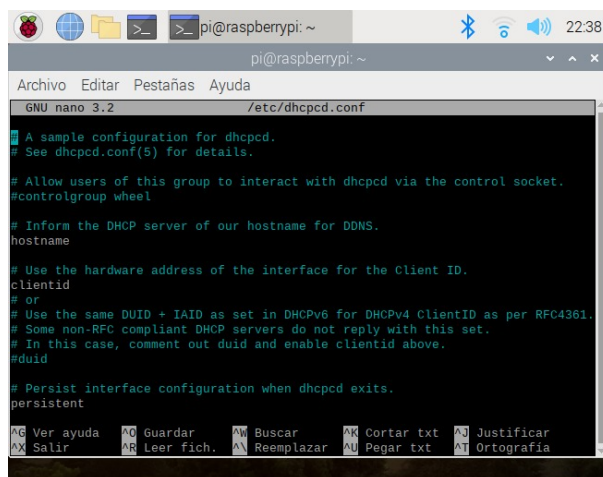
```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 3.2 /etc/dhcpd.conf Modificado
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

interface wlan0
static ip_address=192.168.0.102
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```

Figura 41. Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.

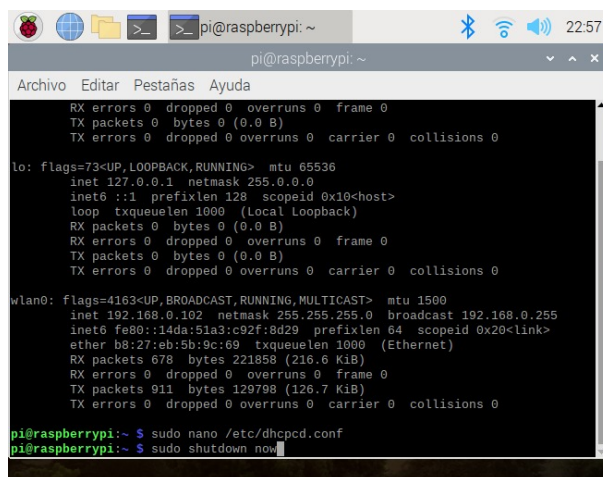
- `ip_address`: Este comando hace referencia a la dirección IP estática que se desea asignar a Raspberry Pi.
 - `routers`: Esta línea del código se refiere a la dirección IP de su enrutador (puerta de enlace predeterminada).
 - `domain_name_servers`: Código que alude a la dirección IP del servidor DNS que se desea usar (generalmente la misma del enrutador).
- Guardar los cambios:
- Presionar “Ctrl+o”.
 - Después “Enter”.
 - Cerrar el menú de configuración de red, ingresando el comando “Ctrl+x”.



```
pi@raspberrypi: ~
pi@raspberrypi: ~
GNU nano 3.2 /etc/dhcpd.conf
A sample configuration for dhcpd.
# See dhcpd.conf(5) for details.
# Allow users of this group to interact with dhcpd via the control socket.
#controlgroup wheel
# Inform the DHCP server of our hostname for DDNS.
hostname
# Use the hardware address of the interface for the Client ID.
clientid
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per RFC4361.
# Some non-RFC compliant DHCP servers do not reply with this set.
# In this case, comment out duid and enable clientid above.
#duid
# Persist interface configuration when dhcpd exits.
persistent
^G Ver ayuda  ^O Guardar    ^W Buscar     ^X Cortar txt ^J Justificar
^X Salir      ^R Leer fich. ^A Reemplazar ^E Pegar txt  ^T Ortografia
```

Figura 42. Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.

- Reiniciar la interfaz de red o reiniciar el Raspberry Pi para aplicar los cambios: `sudo shutdown now`.



```
pi@raspberrypi: ~
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.102 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::14da:51a3:c92f:8d29 prefixlen 64 scopeid 0x20<link>
ether b8:27:eb:5b:9c:69 txqueuelen 1000 (Ethernet)
RX packets 678 bytes 221858 (216.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 911 bytes 129798 (126.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$ sudo nano /etc/dhcpd.conf
pi@raspberrypi:~$ sudo shutdown now
```

Figura 43. Terminal de Raspberry Pi 3. Fuente: Elaborado por autores.

Una vez que se haya completado estos pasos, la Raspberry Pi queda configurada con una dirección IP estática. Se puede verificar la configuración ejecutando el comando “ifconfig” o “ip addr”.

XI-A4. Programación Node-RED - Script de nodos de función:

- **Inicialización de variables:** Para la siguiente función se utilizó un nodo inject, el cual se usa para iniciar un flujo de ejecución enviando un mensaje para un momento establecido. Luego se conecta un nodo inject a una función, estableciendo un punto de inicio para el procesamiento de datos mediante esa función.

A continuación se procede a declarar las variables que se van a utilizar mediante el nodo function, el cual brinda un entorno programable que permite personalizar y extender la funcionalidad de flujos mediante la ejecución de código JavaScript.

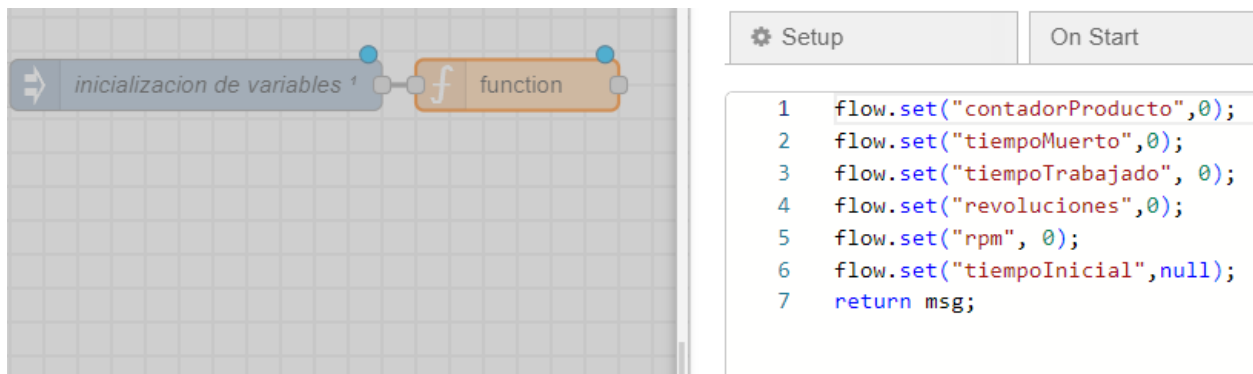


Figura 44. Inicialización de variables en Node-RED. Fuente: Elaborado por autores.

- Inicio y parada de proceso:** Se utiliza un nodo inject en la plataforma para indicar el comienzo y la finalización del proceso.

Dentro de la función se declara y obtiene el valor actual de la variable “maquinaTrabajando”. Se declara el loop If, dentro del cual se inserta la orden de verificar si el valor del campo “payload” en el mensaje es igual a 1, entonces se coloca la variable “maquinaTrabajando” en “true”.

Caso contrario se declara la variable “maquinaTrabajando” en “false”. Por último se devuelve el mensaje de “maquinaTrabajando”, ya sea en estado “true” o “false”.

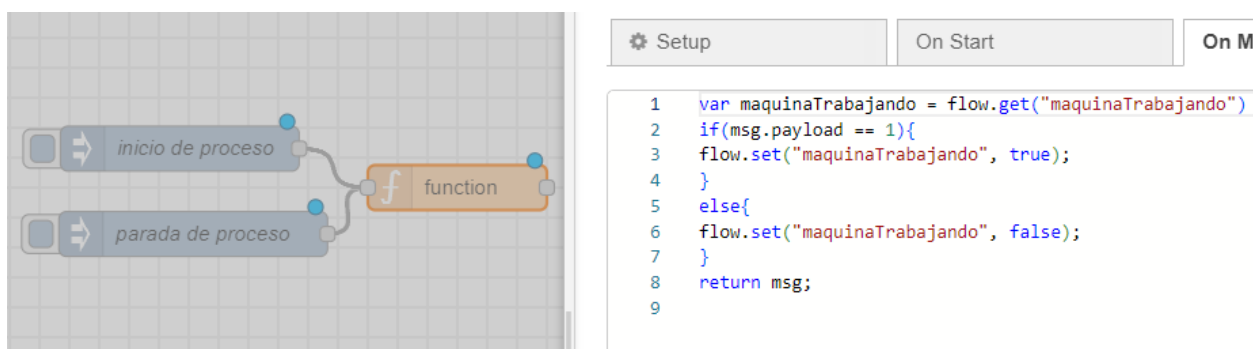


Figura 45. Inicio y parada de proceso en Node-RED. Fuente: Elaborado por autores.

- Tiempos de trabajo:** Se obtienen los valores actuales de las variables “maquinaTrabajando”, “tiempoTrabajado” y “tiempoMuerto” del contexto del flujo. Después se incrementa el tiempo trabajado si la máquina está trabajando. De forma similar, se incrementa el tiempo muerto si la máquina no está trabajando y se actualizan

las variables correspondientes.

Luego se calcula la disponibilidad en porcentaje, utilizando la fórmula: $(\text{tiempoTrabajado} / (\text{tiempoMuerto} + \text{tiempoTrabajado})) * 100$. A continuación se configura el mensaje de salida con información relevante, incluyendo el tiempo trabajado, el estado de la máquina y la disponibilidad. Seguidamente se actualiza la variable de disponibilidad en el contexto del flujo.

Para finalizar, la función devuelve el mensaje modificado. Además, el nodo function tiene asociados los nodos debug, text y gauge. El nodo debug permite imprimir mensajes en la consola de depuración y observar los datos que se encuentran a través del flujo. El nodo text permite imprimir mensajes exclusivamente de tipo string en el dashboard. El nodo gauge se usa para representar gráficamente valores numéricos en un panel de control o dashboard.

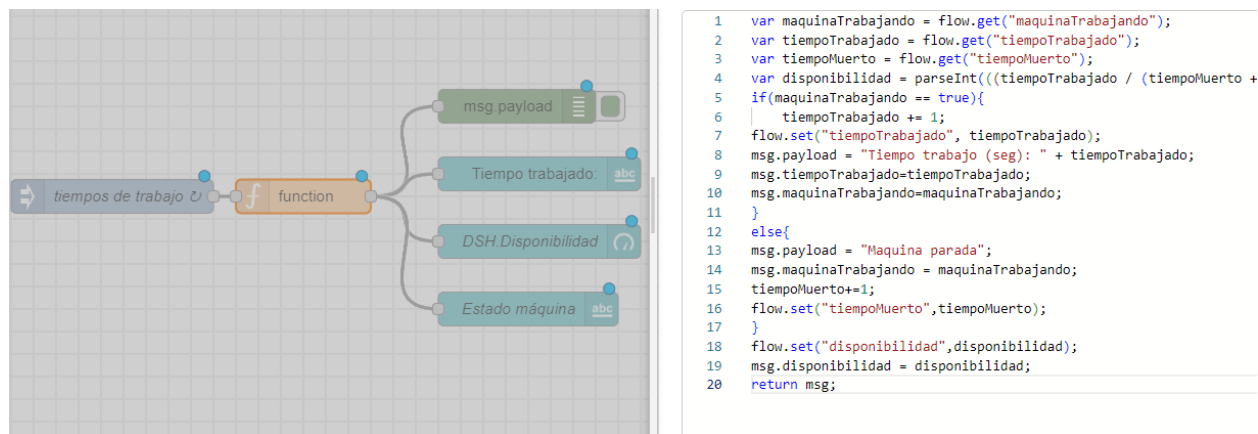


Figura 46. Tiempos de trabajo en Nodo-RED. Fuente: Elaborado por autores.

- **Función RPM:** Dentro del nodo function se obtiene el valor actual almacenado en la variable “revoluciones”, en la línea 2, se concatena la cadena “Velocidad de motor” con el valor de “revoluciones” y la cadena “rpm”, y este resultado se asigna al campo “payload” del objeto msg.

En la siguiente línea, se establece el valor de la variable “revoluciones” en el contexto de flujo de Node-RED a 0. En la línea 3 establece el valor de la variable “revoluciones” en el contexto de flujo de Node-RED a 0.

Finalmente, el código devuelve el objeto msg con su campo payload actualizado, y este objeto se envía al siguiente nodo en el flujo de Node-RED.

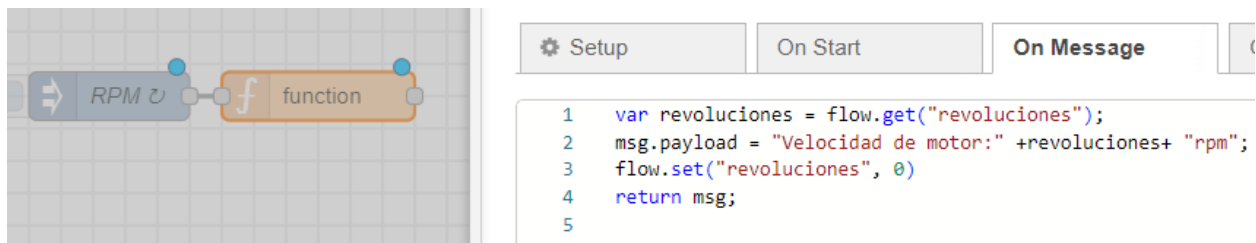


Figura 47. Función RPM en Node-RED. Fuente: Elaborado por autores.

- Función velocidad:** Se obtiene el valor actual de las variables: “tiempoTrabajado”, “tiempoInicial” y “revoluciones”, luego, se declaran 3 variables locales: “diferencia”, “tiempoActual” y “velocidad”.

En la línea 7, se verifica si “tiempoInicial” es nulo, si es así, significa que “tiempoInicial” aún no ha sido establecido. En este caso, se establece el valor de “tiempoInicial” en el tiempo actual usando “Date.now()”.

En las líneas 10 a 15, se comprueba si “tiempoInicial” no es nulo (lo que implica que este código ya se ha ejecutado anteriormente), se calcula la diferencia entre el tiempo actual y el tiempo inicial (“diferencia = tiempoActual - tiempoInicial”). En la siguiente línea, se calcula la velocidad del motor en revoluciones por minuto (rpm) y se asigna a la variable velocidad.

Se agrega la velocidad calculada como una propiedad adicional al objeto msg. Después, en la línea 16 se actualiza el valor de “tiempoInicial” en el contexto de flujo con el tiempo actual, de modo que esté listo para la siguiente ejecución. Finalmente, el código devuelve el objeto msg que contiene la información actualizada sobre la velocidad del motor.

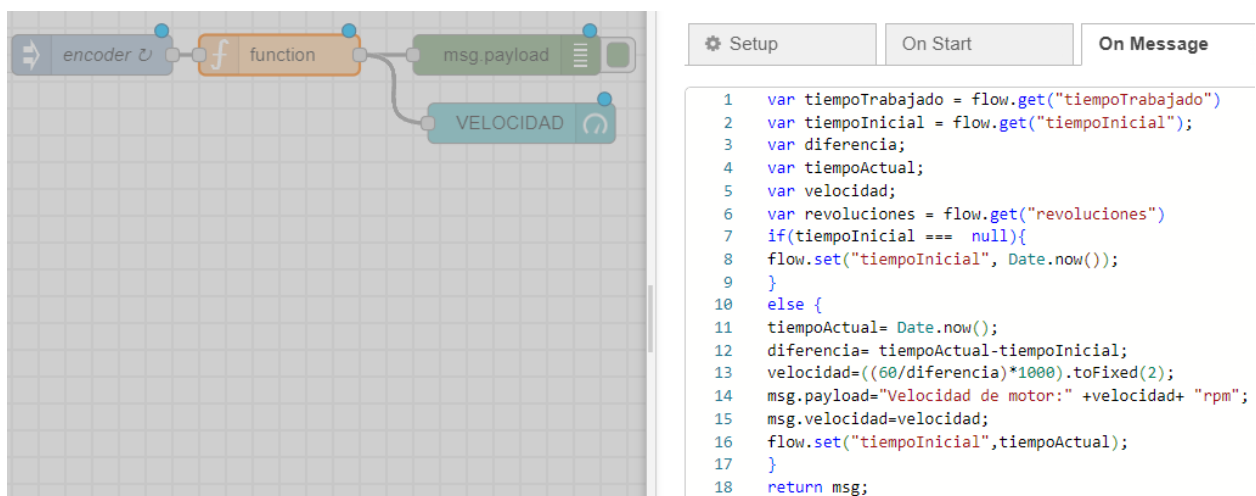


Figura 48. Función velocidad en Node-RED. Fuente: Elaborado por autores.

- Entrada de producto:** Se recoge el valor actual de una variable llamada “contadorProducto” desde el objeto “flow”, después se verifica si el valor de la propiedad “payload” del objeto msg es igual a cero. Si el valor de la propiedad “payload” es 0 en la línea 3, se incrementa el valor de la variable contadorProducto” en 1. En la siguiente línea, se actualiza el valor de la variable “contadorProducto” en el objeto “flow” con el

nuevo valor.

En las 2 últimas líneas, se actualiza el valor de la propiedad “payload” del objeto msg con el nuevo valor de “contadorProducto”, y con el comando “return msg” se devuelve el objeto msg modificado.

Además, esta función tiene como entradas 2 nodos inject que incrementan en 1 unidad la variable “contador-Producto”.

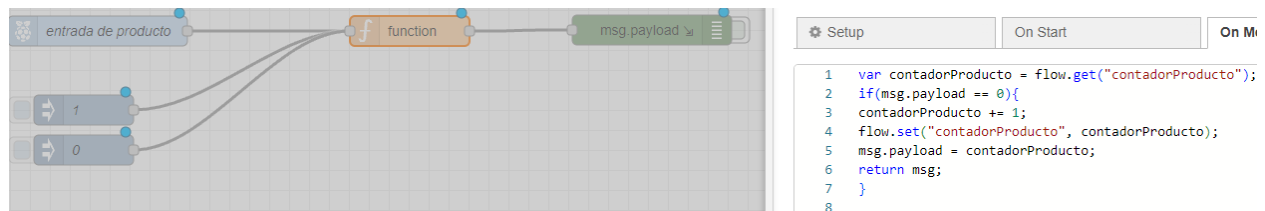


Figura 49. Función entrada de producto en Nodo-RED. Fuente: Elaborado por autores.

- **Salida de producto:** Esta función tiene asociados como entradas 2 nodos inject que se utilizan para aumentar en 1 unidad la variable “total_unidades_producidas”, luego se lee el valor actual de la variable “contador” desde el objeto “Flow”. Si “contador” no tiene un valor definido en el contexto de flujo, se asigna el valor 0. Después para la línea 2, se recupera el valor actual de “total_unidades_producidas” desde el objeto “flow”, asignando 0 si no tiene un valor definido.

En la línea 3, se verifica si el valor de la propiedad “payload” del objeto msg es igual a 1, seguidamente, dentro del bloque if, en la línea 4, si la condición de que la propiedad “payload” es igual a 1, se incrementa el valor de la variable “contador” en 1. En la línea 5, se actualiza el valor de “contador” en el contexto de flujo.

En la línea 6, se actualiza el valor de “total_unidades_producidas” con el nuevo valor de contador. En la línea 7, se actualiza “total_unidades_producidas” en el contexto de flujo. En la línea 8, el comando if (contador > 10) verifica si “contador” es mayor que 10. Si se cumple la condición anterior, se reinicia ‘contador’ a 1 con la orden “contador = 1”.

Para la línea 9, se actualiza “contador” con el comando “flow.set(‘contador’, contador)”, luego, en la línea 10 se refresca el elemento cambiante “total_unidades_producidas” con el nuevo valor de contador. Para la siguiente línea, se agrega una propiedad “counter” al objeto msg con el valor actual de contador.

Después, se agrega una propiedad “total_unidades_producidas” al objeto msg con el valor actual de “total_unidades_producidas”, finalmente se devuelve el objeto msg modificado.

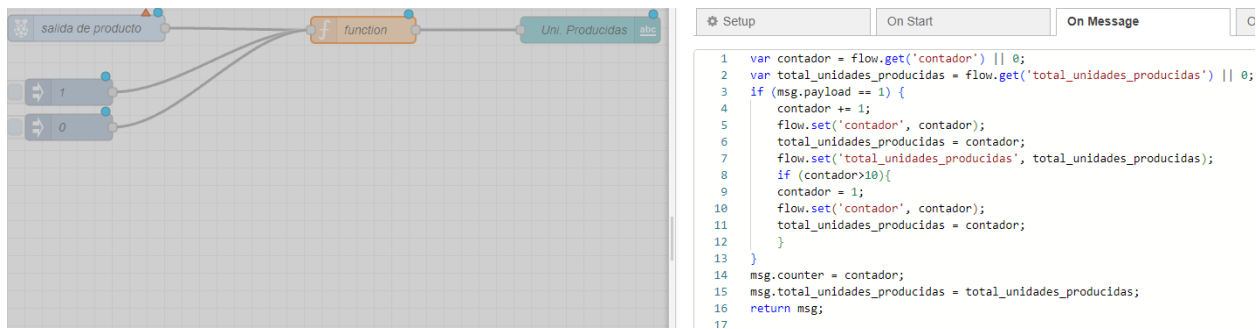


Figura 50. Función salida de producto en Node-RED. Fuente: Elaborado por autores.

- Producto rechazado:** La función tiene asociados como entradas 2 nodos inject que sirven para aumentar en 1 unidad la variable “total_unidades_rechazadas”. Después se lee el valor actual de la variable “contador” desde el objeto “Flow”. Si “contador” no tiene un valor definido en el contexto de flujo, se asigna el valor 0.

Para la línea 2, se recupera el valor actual de “total_unidades_rechazadas” desde el objeto “flow”, asignando 0 si no tiene un valor definido. En la línea 3, se verifica si el valor de la propiedad “payload” del objeto msg es igual a 1. Si la condición de que la propiedad “payload” es igual a 1 se cumple en la línea 4, se incrementa el valor de la variable “contador” en 1.

Se actualiza el valor de contador en el contexto de flujo en la línea 5. Se reajusta el valor de “total_unidades_producidas” con el nuevo valor de contador en la línea 6. Se renueva la variable “total_unidades_rechazadas” en el contexto de flujo en la línea 7.

En la línea 8, el comando “if (contador > 10)” se verifica si “contador” es mayor que 10. Si se cumple la condición anterior, se reinicia “contador” a 1 con la orden “contador = 1”, luego se actualiza “contador” en el contexto de flujo con el comando “flow.set(‘contador’, contador)” en la línea 9. A continuación, en la línea 10, se actualiza “total_unidades_rechazadas” con el nuevo valor de contador.

Para la siguiente línea, se agrega una propiedad “counter” al objeto msg con el valor actual de “contador”. Después, se agrega una propiedad “total_unidades_rechazadas” al objeto msg con el valor actual de “total_unidades_rechazadas”, por último se devuelve el objeto msg modificado.

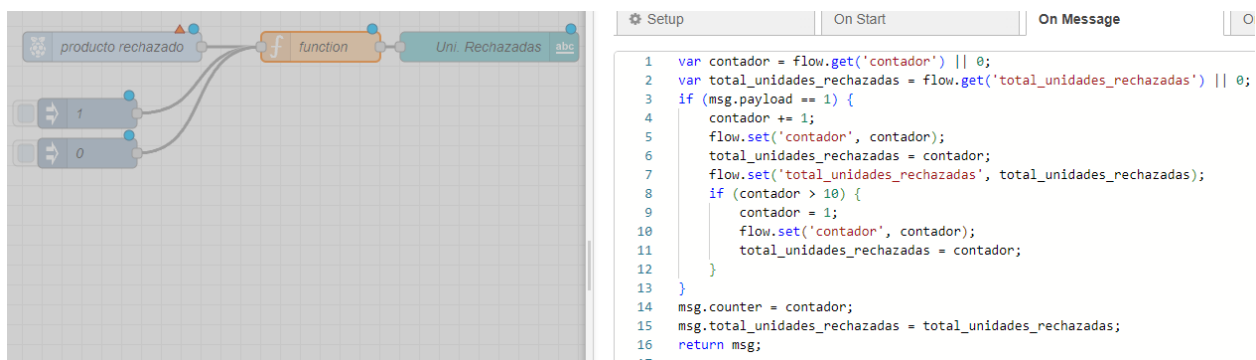


Figura 51. Función producto rechazado en Node-RED. Fuente: Elaborado por autores.

- Calidad:** La función comienza obteniendo el valor actual de las variables “total_unidades_producidas” y

“total_unidades_rechazadas” desde el objeto flow.

Después se calcula el porcentaje de calidad utilizando la fórmula: $\text{var calidad} = \text{parseFloat}(\frac{\text{total_unidades_producidas}}{\text{total_unidades_producidas} + \text{total_unidades_rechazadas}} * 100)$.toFixed(2)).

Luego se compara la proporción de unidades producidas con respecto a la fórmula: $\text{total de unidades (producción + rechazadas)} * 100$. Se convierte la proporción a porcentaje, con “.toFixed(2)” se redondea el resultado a dos decimales y se convierte a un número de punto flotante con “parseFloat()”.

En la línea 4, se almacena el valor de “calidad” en el contexto de flujo para su uso posterior. Para la siguiente línea, se agrega la propiedad “calidad” al objeto msg y se le asigna el valor calculado: “msg.calidad = calidad”, finalmente, “return msg” devuelve el objeto msg modificado.

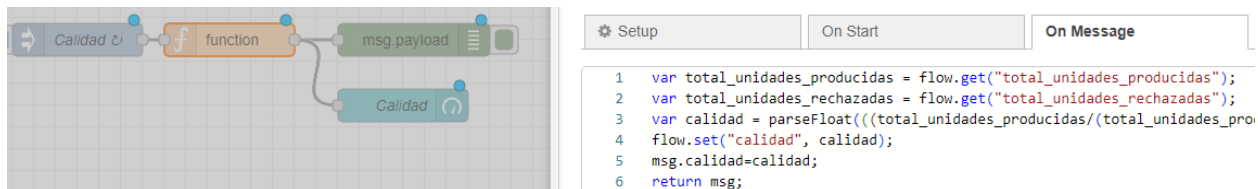


Figura 52. Función calidad en Node-RED. Fuente: Elaborado por autores.

- **Eficiencia:** La función inicia obteniendo el valor actual de las variables “calidad”, “disponibilidad” y “rendimiento” desde el objeto “flow”, luego se calcula la eficiencia con la fórmula, que se observa en la línea 3. En la línea 5, se almacena el valor de “eficiencia” en el contexto de flujo para su uso posterior, para la siguiente línea, se agrega una propiedad “eficiencia” al objeto msg y se le asigna el valor calculado. Finalmente, se devuelve el objeto msg modificado.

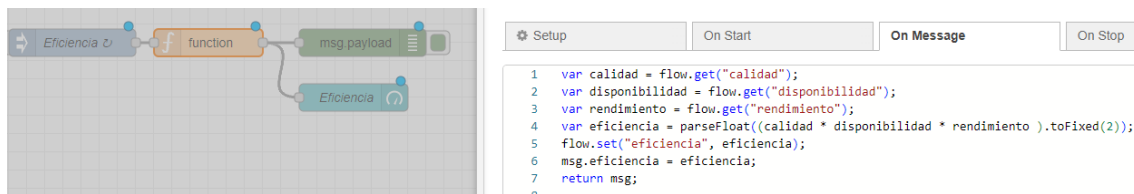


Figura 53. Función eficiencia en Node-RED. Fuente: Elaborado por autores.

- **Rendimiento:** Se obtiene la variable velocidad del flujo de Node-RED mediante “flow.get(“velocidad”)”.En la siguiente línea se define la velocidad máxima permitida como 15 (probablemente en alguna unidad específica, como km/h).

Después en la línea 3 se calcula el rendimiento utilizando la fórmula: $(\text{velocidad}/\text{velocidad_max}) * 100$. Esto dará un valor proporcional al rendimiento en términos porcentuales. En la misma línea con la función “parseFloat()” se convierte el resultado a un número decimal y con “toFixed(2)” se redondea el resultado a dos decimales.



Figura 54. Función rendimiento en Node-RED. Fuente: Elaborado por autores.

■ **Temperatura y humedad:**

En la línea 1 se está utilizando la función “set()” del objeto “flow” en Node-RED para almacenar el valor de msg.payload en una variable llamada ”temperatura”. El objeto “flow” es un mecanismo para almacenar datos entre las ejecuciones de los flujos en Node-RED, lo que significa que la información almacenada en “flow” persistirá mientras el flujo esté en ejecución. En este caso, se está almacenando el valor de “msg.payload” con la clave ”temperatura”.

Finalmente, en la línea 2 se devuelve el objeto msg. En Node-RED, msg es un objeto que se utiliza para pasar datos entre nodos del flujo. En este caso, el objeto msg se ha modificado al establecer msg.payload y almacenar su valor en flow, y luego se devuelve para que pueda continuar su procesamiento en el flujo de Node-RED.



Figura 55. Función temperatura en Nodo-RED. Fuente: Elaborado por autores.

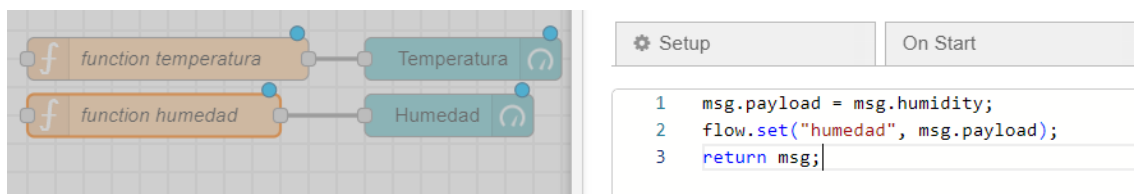


Figura 56. Función humedad en Nodo-RED. Fuente: Elaborado por autores.

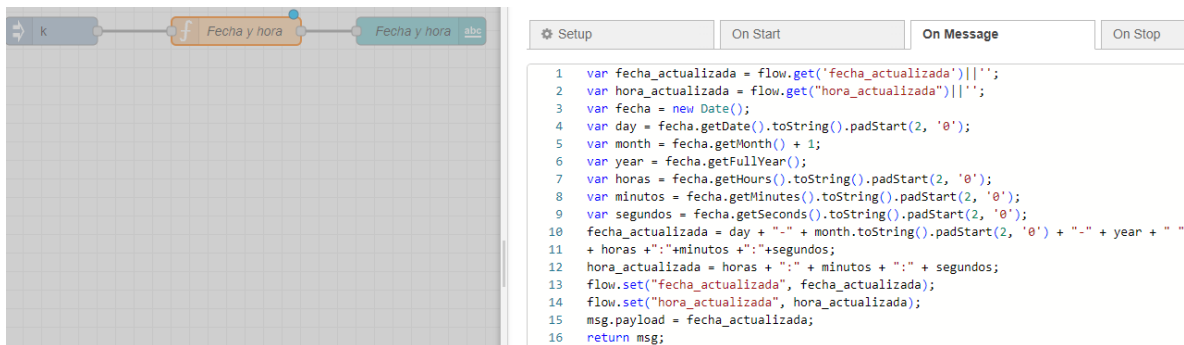
- **Fecha y hora:** Se inicia obteniendo las variables “fecha_actualizada” y “hora_actualizada”. Si no hay valores almacenados, se les asigna una cadena vacía, en la línea 3, se crea una nueva instancia de la clase Date en JavaScript para obtener la fecha y hora actuales, luego, en la línea 4, se obtienen el día, mes y año por separado y se formatean para tener dos dígitos utilizando el método padStart.

Después, en la línea 7, se obtienen las horas, minutos y segundos y se formatean también para tener dos dígitos, en la línea 10, se actualiza la variable “fecha_actualizada” con la fecha actual formateada como “dd-mm-yyyy hh:mm:ss”.

Para la línea 12, se actualiza la variable “hora_actualizada” con la hora actual formateada como “hh:mm:ss”, en la siguiente línea, se almacenan nuevamente en el flujo con las claves “fecha_actualizada” y “hora_actualizada”

utilizando el método “flow.set()”.

En la línea 15, se declara la propiedad msg.payload que se configura con el valor de “fecha_actualizada”. Por último, se devuelve el objeto de mensaje msg. En el contexto de Node-RED, esto probablemente enviará el mensaje a otro nodo en el flujo.



The image shows a Node-RED workspace with a flow containing two 'Fecha y hora' nodes. The second node is selected, and its code editor is open, displaying the following JavaScript code:

```
1 var fecha_actualizada = flow.get('fecha_actualizada')||'';
2 var hora_actualizada = flow.get("hora_actualizada")||'';
3 var fecha = new Date();
4 var day = fecha.getDate().toString().padStart(2, '0');
5 var month = fecha.getMonth() + 1;
6 var year = fecha.getFullYear();
7 var horas = fecha.getHours().toString().padStart(2, '0');
8 var minutos = fecha.getMinutes().toString().padStart(2, '0');
9 var segundos = fecha.getSeconds().toString().padStart(2, '0');
10 fecha_actualizada = day + "-" + month.toString().padStart(2, '0') + "-" + year + " "
11 + horas + ":" + minutos + ":" + segundos;
12 hora_actualizada = horas + ":" + minutos + ":" + segundos;
13 flow.set("fecha_actualizada", fecha_actualizada);
14 flow.set("hora_actualizada", hora_actualizada);
15 msg.payload = fecha_actualizada;
16 return msg;
```

Figura 57. Función fecha y hora en Node-RED. Fuente: Elaborado por autores.

XI-A5. Script de nodos de función:

```
var maquinaTrabajando = flow.get("maquinaTrabajando")
if(msg.payload == 1){
flow.set("maquinaTrabajando", true);
}
else{
flow.set("maquinaTrabajando", false);
}
msg.maquinaTrabajando = maquinaTrabajando;
return msg;
flow.set("contadorProducto", 0);
flow.set("tiempoMuerto", 0);
flow.set("tiempoTrabajado", 0);
flow.set("revoluciones",0);
flow.set("rpm", 0);
flow.set("velocidad", 0);
flow.set("calidad", 0);
flow.set("rendimiento", 0);
flow.set("eficiencia", 0);
flow.set("total_unidades_rechazadas", 0);
flow.set("total_unidades_producidas", 0);
flow.set("tiempoInicial",null);
return msg;
var maquinaTrabajando = flow.get("maquinaTrabajando");
var tiempoTrabajado = flow.get("tiempoTrabajado");
var tiempoMuerto = flow.get("tiempoMuerto");
var disponibilidad = parseInt(((tiempoTrabajado / (tiempoMuerto +
tiempoTrabajado)) * 100).toFixed(2));
if(maquinaTrabajando == true){
    tiempoTrabajado += 1;
flow.set("tiempoTrabajado", tiempoTrabajado);
msg.payload = "Tiempo trabajo (seg): " + tiempoTrabajado;
msg.tiempoTrabajado=tiempoTrabajado;
    msg.payload = "Maquina encendida";
msg.maquinaTrabajando=msg.payload;
}
else{
msg.payload = "Maquina parada";
msg.maquinaTrabajando = msg.payload;
tiempoMuerto+=1;
flow.set("tiempoMuerto",tiempoMuerto);
}
flow.set("disponibilidad",disponibilidad);
msg.disponibilidad = disponibilidad;
return msg;
var velocidad = flow.get("velocidad");
var revoluciones = flow.get("revoluciones");
var pi = Math.PI;
msg.payload = "Velocidad de motor:" +revoluciones+ "rpm";
msg.velocidad = revoluciones;
```



```

flow.set("revoluciones", 0);
flow.set("velocidad", revoluciones);
return msg;
var tiempoTrabajado = flow.get("tiempoTrabajado")
var tiempoInicial = flow.get("tiempoInicial");
var diferencia;
var tiempoActual;
var velocidad;
var revoluciones = flow.get("revoluciones")
if(msg.payload == 0){
  revoluciones += 1;
  flow.set("revoluciones", revoluciones);
}
return msg;
var total_unidades_producidas = flow.get("total_unidades_producidas");
var contadorProducto = flow.get("contadorProducto");
var calidad = parseFloat((((total_unidades_producidas) / (contadorProducto)) *
100).toFixed(2));
flow.set("calidad", calidad);
msg.calidad=calidad;
return msg;
var velocidad = flow.get("velocidad");
var rendimiento = flow.get("rendimiento");
var velocidad_max = 10;
rendimiento = parseFloat(((velocidad/velocidad_max)*100).toFixed(2));
flow.set("rendimiento",rendimiento);
msg.rendimiento = rendimiento;
return msg;
var calidad = flow.get("calidad");
var disponibilidad = flow.get("disponibilidad");
var rendimiento = flow.get("rendimiento");
var eficiencia = parseFloat((((calidad/100) * (disponibilidad/100)*
(rendimiento/100)*100).toFixed(2));
flow.set("eficiencia", eficiencia);
msg.eficiencia = eficiencia;
return msg;
msg.payload=msg.payload;
flow.set("humedad", msg.payload);
return msg;
msg.payload=msg.humidity;
return msg;
var contadorProducto = flow.get("contadorProducto")|| 0;
var tiempoMuerto = flow.get("tiempoMuerto")|| 0;
var tiempoTrabajado = flow.get("tiempoTrabajado")|| 0;
var revoluciones = flow.get("revoluciones")|| 0;
var rpm = flow.get("rpm")|| 0;
var tiempoInicial = flow.get("tiempoInicial")|| 0;
var maquinaTrabajando = flow.get("maquinaTrabajando")|| 0;
var contador = flow.get('contador') || 0;
var total_unidades_rechazadas = flow.get('total_unidades_rechazadas') || 0;

```

```

var total_unidades_producidas = flow.get('total_unidades_producidas') || 0;
var velocidadmotor = flow.get("velocidadmotor") || 0; //opcional
var disponibilidad = flow.get("disponibilidad") || 0;
var calidad = flow.get("calidad") || 0;
var eficiencia = flow.get("eficiencia") || 0;
var rendimiento = flow.get("rendimiento") || 0;
//Datos en topic
msg.payload = [{
  entrada_producto: contadorProducto,
  unidades_rechazadas: total_unidades_rechazadas,
  unidades_producidas: total_unidades_producidas,
  calidad_producto: calidad,
}, {
  tag1: "banda_1"
}];
return msg;

```

XI-A6. Elaboración de control remoto para la visualización de datos: Después de calcular los parámetros del prototipo con éxito, el siguiente paso es la transmisión y almacenamiento de estos datos. Esta fase implica la elección entre una base de datos local, una integrada directamente en el sistema, o una base de datos en la nube, siendo esta última la más apropiada porque proporciona ventajas de accesibilidad y escalabilidad.

Grafana, una plataforma de código abierto, se erige como una opción valiosa para facilitar la visualización y el formato de los datos.

Esta herramienta no solo proporciona una interfaz gráfica intuitiva para crear dashboards personalizados, sino que también ofrece la capacidad de conectarse directamente a la base de datos. La exploración de Grafana abre la posibilidad de crear dashboards dinámicos y en tiempo real, permitiendo una supervisión continua del rendimiento del prototipo.

En resumen, el proceso que sigue al cálculo de los parámetros del prototipo involucra la transmisión, almacenamiento y visualización de estos datos, con la posibilidad de aprovechar herramientas avanzadas como Grafana. Esta estrategia integral no solo garantiza una gestión eficiente de los datos métricos, sino que también proporciona una visualización potente y editable para mejorar la toma de decisiones y la comprensión del rendimiento del prototipo.

XI-A7. Recepción de datos en InfluxDB cloud:

- Abrir node red.

- Ingresar a Manage palette.



Figura 58. Managge Palette. Fuente: Elaborado por autores.

- Se instala el paquete de nodos node-red-contrib-InfluxDB.

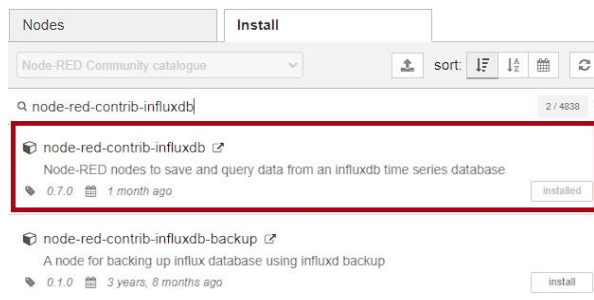


Figura 59. Instalación de InfluxDB cloud en nodo red. Fuente: Elaborado por autores.

- Se ingresa al siguiente link <https://cloud2.influxdata.com/signup> se registra en la página con una cuenta de Google.

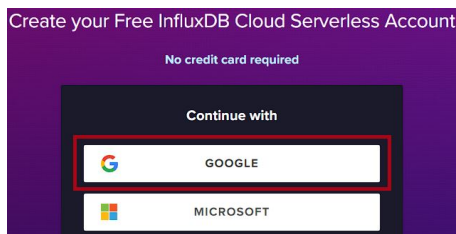


Figura 60. Interfaz de acceso. Fuente: Elaborado por autores.

- Se colocan los nombres de la cuenta y organización. Luego se presiona continuar.

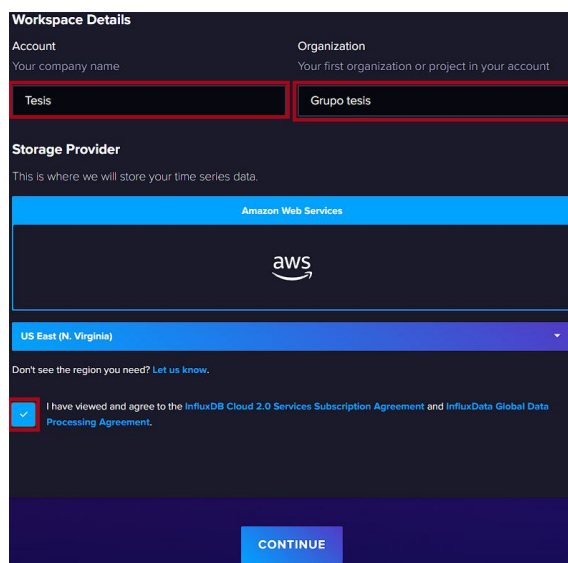


Figura 61. Requerimientos de cuenta InfluxDB cloud. Fuente: Elaborado por autores.

- Se mantiene el plan gratis en InfluxDB cloud.

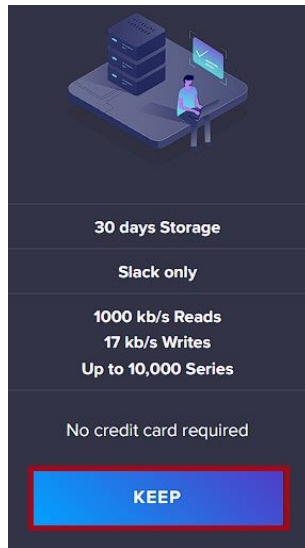


Figura 62. Versión Influx 2.0. Fuente: Elaborado por autores.

- Seleccionar los ítems como se muestran en la “figura 63” y presionar continuar.

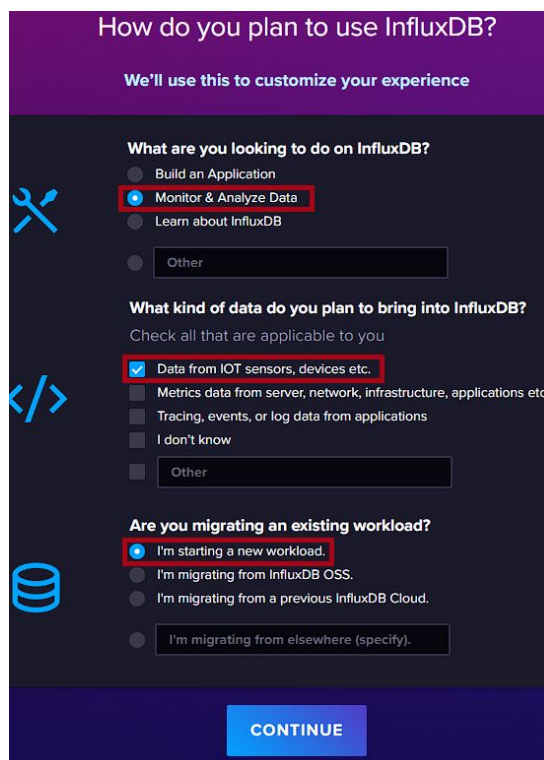


Figura 63. Condiciones. Fuente: Elaborado por autores.

- Crear un nuevo bucket

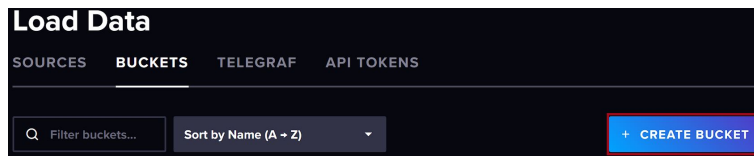


Figura 64. Bucket. Fuente: Elaborado por autores.

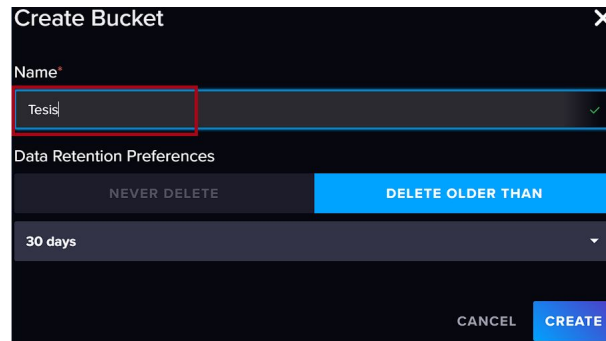


Figura 65. Nombre de proyecto. Fuente: Elaborado por autores.

- Generar una API token.

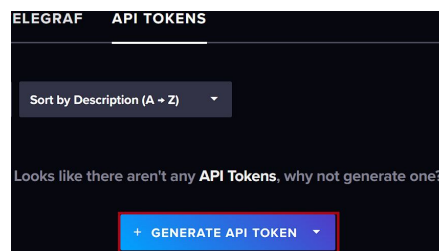


Figura 66. API token. Fuente: Elaborado por autores.

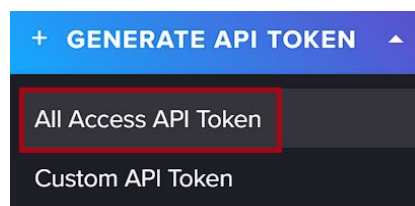


Figura 67. Acceso a Api token. Fuente: Elaborado por autores.

- Dar nombre a la API.

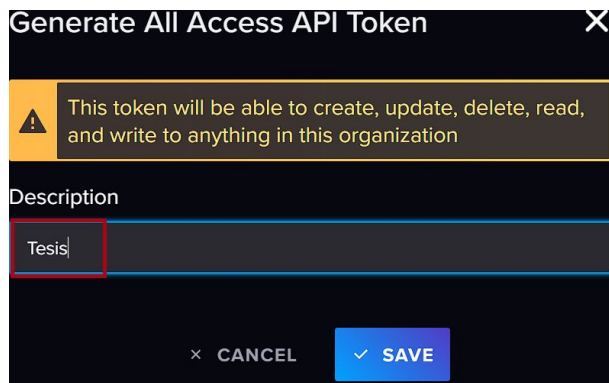


Figura 68. Nombre de API. Fuente: Elaborado por autores.

- Copiar el token generado.

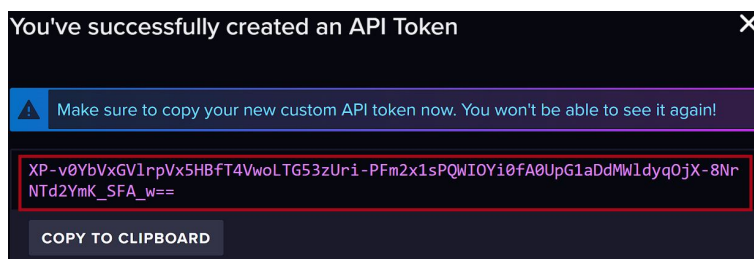


Figura 69. Dirección de token. Fuente: Elaborado por autores.

- Regresar a Node-RED y agregar un nodo influxdb out.

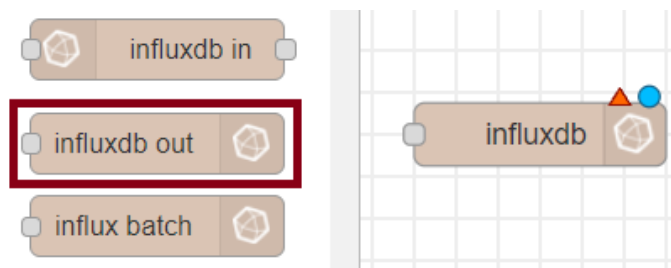


Figura 70. Nodo Influx out. Fuente: Elaborado por autores.

- Ingresar los datos de server Organization, Bucket y Measurement. De la misma manera que fueron ingresados en el servidor InfluxDB cloud.

The image shows a 'Properties' configuration form for InfluxDB. The form includes the following fields: 'Name' (text input), 'Server' (dropdown menu with '[v2.0] BASE DE DATOS NUBE' selected), 'Organization' (text input with 'Grupo tesis'), 'Bucket' (text input with 'Tesis'), 'Measurement' (text input with 'banda 1'), and 'Time Precision' (dropdown menu with 'Milliseconds (ms)'). Red rectangular boxes highlight the 'Server', 'Organization', 'Bucket', and 'Measurement' fields.

Figura 71. Datos para vinculo con InfluxDB. Fuente: Elaborado por autores.

- Hacer clic en el icono que se muestra en la “figura 72”.

This image is identical to Figure 71, showing the 'Properties' configuration form. However, a red rectangular box highlights the edit icon (a pencil) located to the right of the 'Server' dropdown menu.

Figura 72. Acceso a configuración de conexión. Fuente: Elaborado por autores.

- Ingresar la versión del InfluxDB cloud con que se esté trabajando, la URL de la página principal de InfluxDB cloud y el token previamente generado dentro de la base de datos.



Figura 73. Versión2.0. Fuente: Elaborado por autores.

- Para visualizar los datos de Node-RED en InfluxDB cloud se realizan los siguientes pasos:

- 1.- Se abre la ventana buckets.
- 2.- Luego “tesis”.

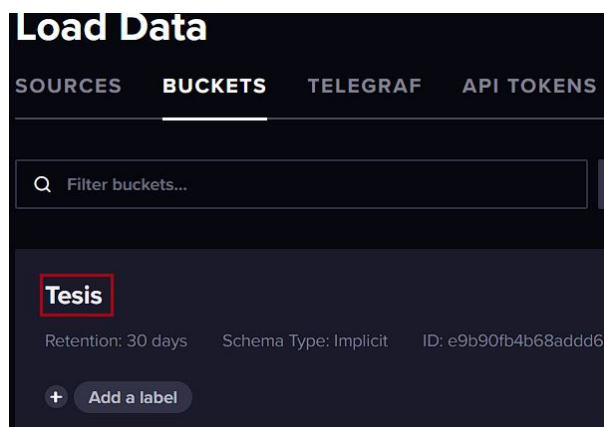


Figura 74. Nombre del Bucket. Fuente: Elaborado por autores.

- Seleccionar los campos que se muestran al desplegar los menús, como se muestra en la “figura 75” y hacer clic en “RUN” para visualizar los valores de las variables seleccionadas.

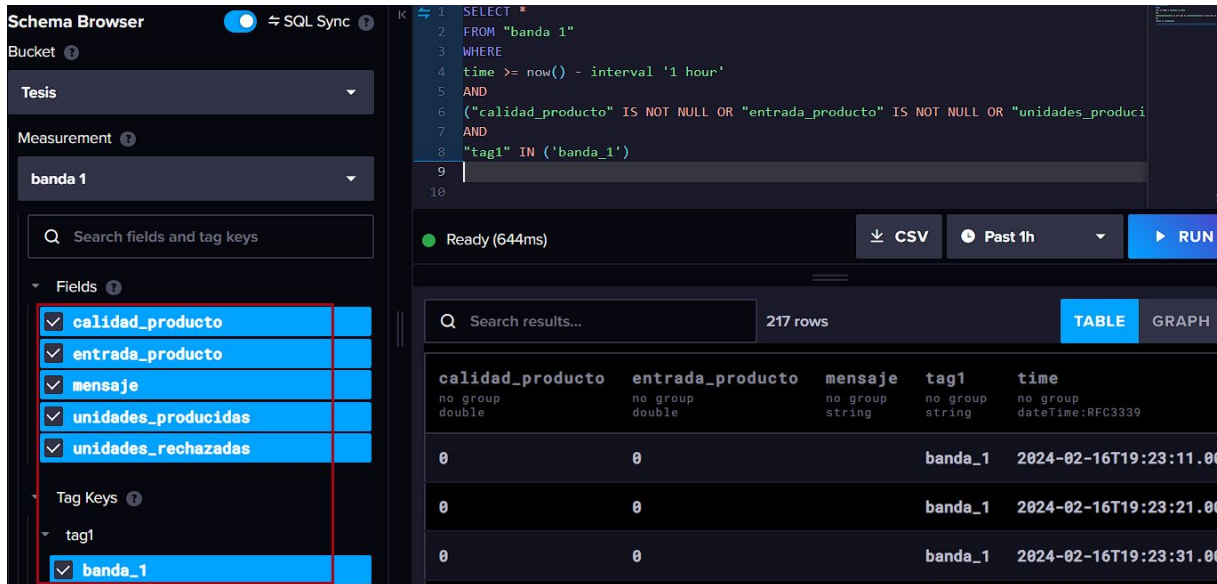


Figura 75. Variables en InfluxDB cloud. Fuente: Elaborado por autores.

XI-A8. Comunicación InfluxDB cloud con Grafana cloud:

- Ingresamos a Grafana a través del siguiente link:
<https://grafana.com/>
- Luego hacer clic en “create free account”.

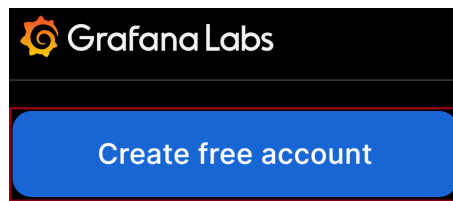


Figura 76. Grafana Labs. Fuente: Elaborado por autores.

- Se registra en la página con una cuenta de Google.

Create your Grafana Cloud account

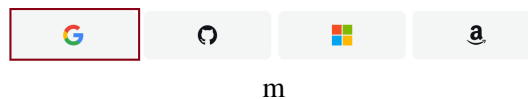


Figura 77. Vinculación cuenta de Gmail. Fuente: Elaborado por autores.

- Se elige la región de implementación para el software y luego se presiona “finish setup”.

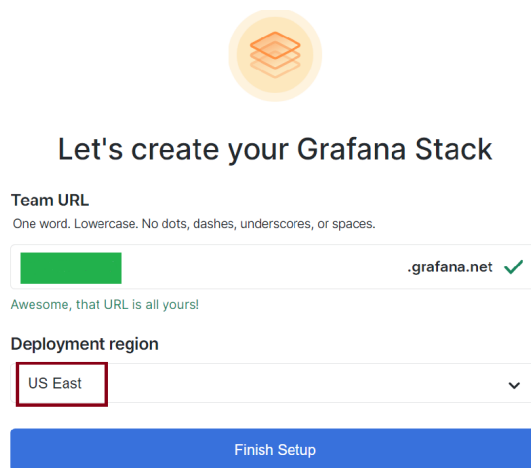


Figura 78. Región de implementación. Fuente: Elaborado por autores.

- Se escoge la opción “create a dashboard”.

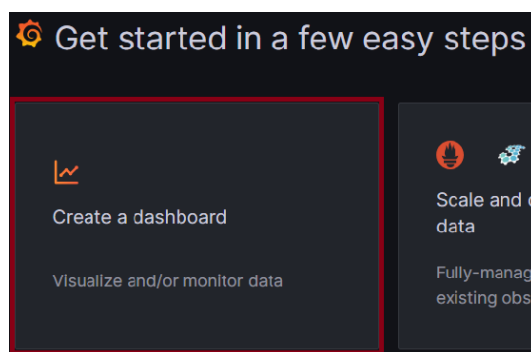


Figura 79. Creación de dashboard. Fuente: Elaborado por autores.

- Seleccionar el ícono de InfluxDB.

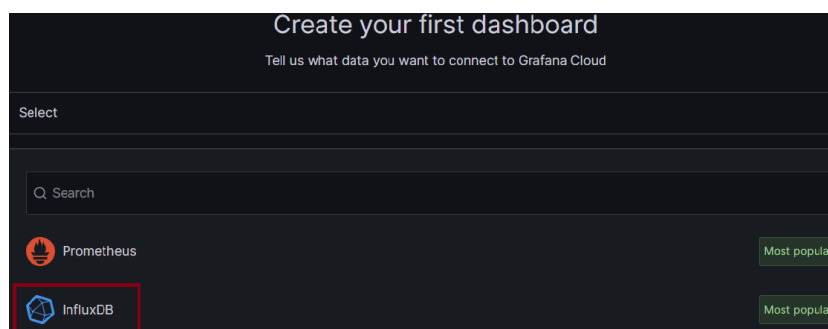


Figura 80. Selección de InfluxDB. Fuente: Elaborado por autores.

- Optar por la opción que se observa en la “figura 81”.

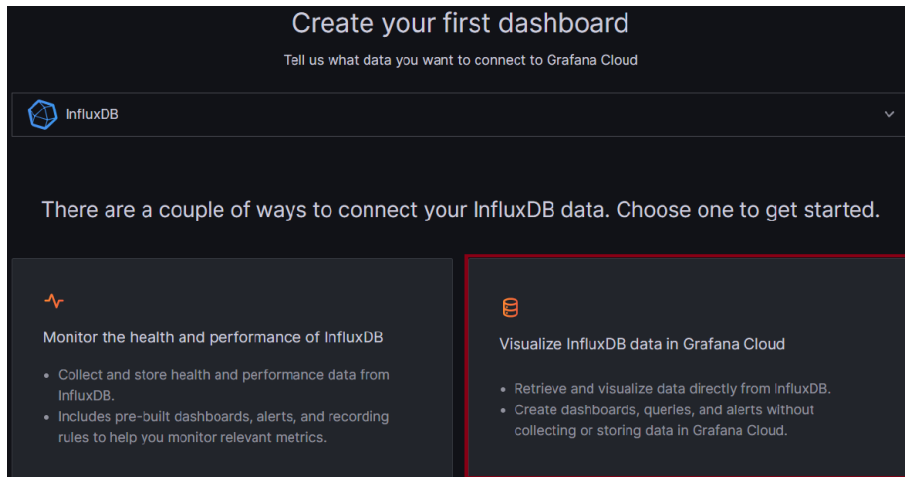


Figura 81. Selección de visualización. Fuente: Elaborado por autores.

- Rellenar los campos según se muestra en la “figura 82”, en la sección “Valor*”, colocar la API token generada en la base de datos InfluxDB cloud.

Figura 82. Campos de vinculación con InfluxDB. Fuente: Elaborado por autores.

- Elegir por la opción “+Add visualization”, para iniciar un nuevo dashboard.

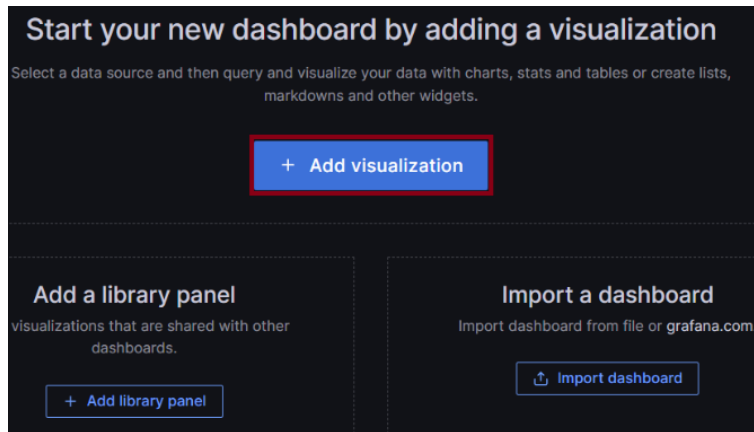


Figura 83. Establecimiento de visualización. Fuente: Elaborado por autores.

- Elegir InfluxDB dentro del menú de selección de fuente de datos.

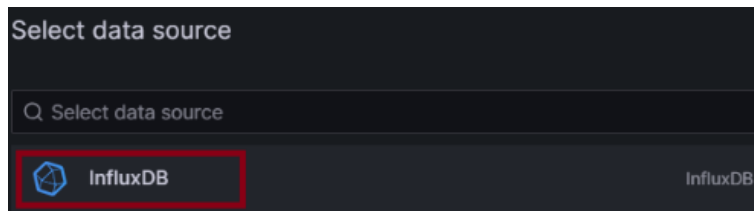


Figura 84. Selección de InfluxDB. Fuente: Elaborado por autores.

- Pegar las siguientes líneas de código para visualizar las variables declaradas dentro del mismo, como se puede ver en la “figura 85”.

```
from(bucket: "Tesis")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "banda 1")
  |> filter(fn: (r) => r["tag1"] == "banda_1")
  |> filter(fn: (r) => r["_field"] == "entrada_producto" or
    r["_field"] == "unidades_rechazadas" or r["_field"]
    == "unidades_producidas" or r["_field"] == "calidad_producto")
  |> last()
```

```

Query 1 Transform data 1
A (influxdb)
1 from(bucket: "Tesis")
2 |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3 |> filter(fn: (r) => r["_measurement"] == "banda 1")
4 |> filter(fn: (r) => r["tag1"] == "banda_1")
5 |> filter(fn: (r) => r["_field"] == "entrada_producto" or r["_field"] == "unidades_rechazadas"
6 or r["_field"] == "unidades_producidas" or r["_field"] == "calidad_producto")
7 |> last()

```

Figura 85. Importado de código. Fuente: Elaborado por autores.

- En la “figura 86” se observan los valores de las variables enviadas como salida hacia la base de datos InfluxDB cloud.

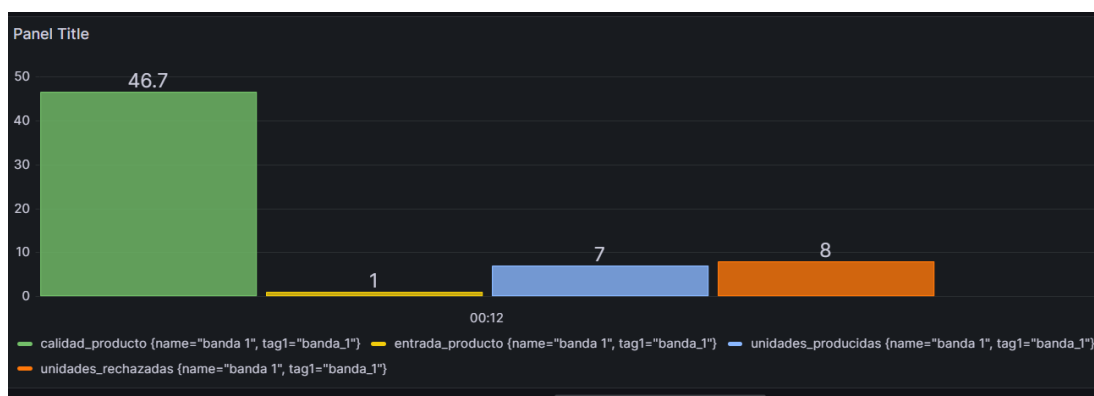


Figura 86. Dashboard Grafana. Fuente: Elaborado por autores.

XI-B. Sistema eléctrico

Arranque directo y sostenido

Para la elaboración del sistema de arranque del motor de la banda transportadora se usarán los siguientes componentes:

- * Disyuntor industrial de 3 polos a 25A para la conexión diagrama de fuerza.
- * Disyuntor industrial de 10 Amperios de 2 polos para conectar el control que alimentará la bobina del contactor.
- * Dos relés de 5V para el arranque sostenido, dichos relés recibirán las señales que se enviarán desde la raspberry para lograr el enclavamiento de la bobina.
- * Contactor de bobina 220V a 25A.
- * Motor reductor de 35 Watts a 10 rpm.

La alimentación del disyuntor principal estará dada por una entrada monofásica 220V que será el punto de partida para el arranque del motor, a continuación del diagrama se conectará un guardamotor de entrada y salida trifásica con un rango térmico de 2.5 a 4 Amperios para evitar picos de Voltaje o sobre consumo en la bobina.

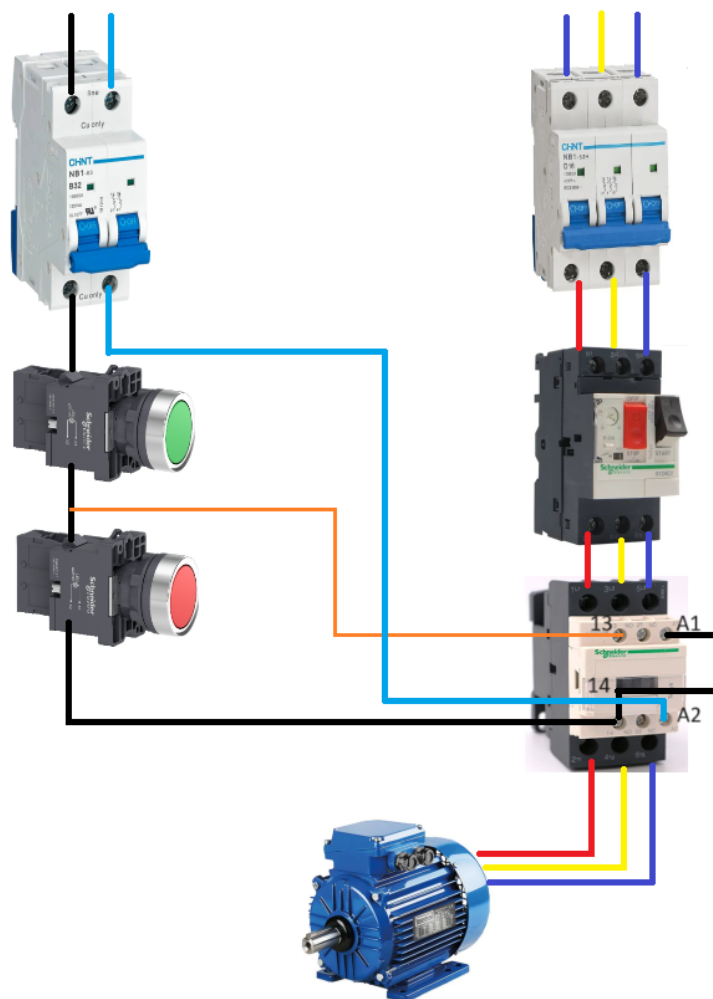


Figura 87. Arranque directo de motor. Fuente: Elaborado por autores.

Después, se activarán los contactos del actuador desde la salida del guardamotor, tal como se ilustra en la “figura 87”. Además, se conectará una salida del actuador al capacitor de $4\mu F$ del motor, mientras que la otra salida se dirigirá hacia la bobina directa del motor. Este proceso se llevará a cabo para facilitar el arranque sostenido del motor, proporcionando la fuerza necesaria para dicho propósito.

Para la alimentación y conexión del control del arranque se utilizará un disyuntor industrial de dos polos a 10A, en el cual L1 irá directamente al A1 de la bobina del actuador, mientras que L2 estará interceptada por un relé de 5V como primer punto de conexión; L2 se conectará al puerto común y al contacto “NC” del mismo, este actuará como una parada asistida evitando el enclavamiento de la bobina en el puerto 13 y 14 de los contactos del actuador, a continuación se acoplará L2 al contacto auxiliar “NC” puerto 1 de un pulsador tipo hongo que se empleará como parada de emergencia física, después se vinculará a otro contacto auxiliar “NC” entrada 1 y salida 2 de un pulsador normal que tendrá la función de parada de proceso y finalmente se conectará un contacto auxiliar “NO” entrada 3 y salida 4 el cual hará la función de inicio de proceso.

Para el enclavamiento de la bobina del contactor del puerto 3 del auxiliar “NO” que trabaja como inicio de proceso se conectará un cable al puerto 13 de los contactos del actuador y de la salida 4 del contacto auxiliar “NO” un cable al puerto 14 del mismo, estableciendo el auto enclavamiento de la bobina, evitando que el contactor quede desenergizado.

Como es un arranque sostenido se utilizará un relé de 5 Voltios al mismo puerto A2 de la bobina del contactor, pero esta vez de forma independiente conectando el puerto común a la entrada y el normalmente abierto a la salida del relé desde el puerto L2 y retornando al puerto A2 de la bobina del contactor.

XI-C. Sistema electroneumático

- Rechazo por detección de componentes metálicos

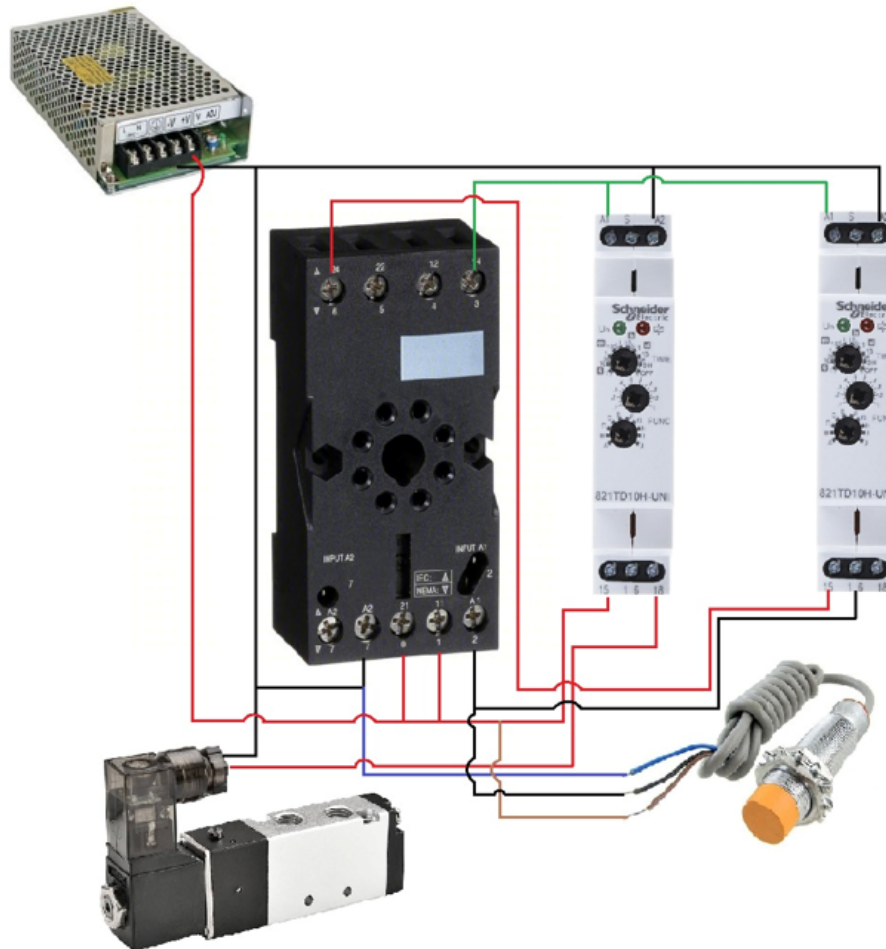


Figura 88. Sistema electro neumático. Fuente: Elaborado por autores.

Para el arranque del sistema de rechazo electroneumático se implementará una fuente de poder de 110V a 220V AC a 24V DC de 5 Amperios de capacidad, se utilizará un sensor inductivo PNP, un relé de 24V DC 8 pines, dos timer “ON” a la conexión y una electroválvula 5/2 a 24 V DC.

Se enlazará el cable azul del sensor inductivo al GND de la fuente y el cable marrón a los 24V. El cable negro es de señal y como se trata de un sensor PNP el pulso será enviado a través de los 24V del sensor. A continuación, se conectará al pin 2 (A1) de la bobina del relé el cable color negro de señal del sensor y al pin 7 de la bobina (A2) el cable GND desde la fuente cumpliendo la condición de activación del relé cada vez que el sensor inductivo detecte metal dentro de la superficie de censado.

Después de lograr enclavar la bobina del relé se utilizarán los contactos 1 y 8 para energizar las bobinas del timer 1 y el timer 2 conectadas al puerto A1 mientras que el puerto A2 será energizado desde el GND de la fuente de poder.

A la salida de los relés existen los puertos 15, 16 y 18. El puerto 15 es el contacto común, el puerto 16 es el “NC” y el puerto 18 es el “NO”, el puerto 15 se energizará con 24 v de la fuente de poder y el puerto 18 se conectará a la entrada de la electroválvula logrando enviar la señal después que el timer 1 se energice, prolongando la activación del puerto positivo de la válvula solenoide y activando la entrada del pistón neumático. Una vez se cumple la condición del timer 1 al timer 2 se conectará al puerto 15 una fase de 24V dada por el puerto 6 del relé y al puerto 16 una línea que energizará el pin 2 (A1) del mismo logrando un enclavamiento temporizado mientras el timer 2 retarda la señal y, abriéndolo para que el pistón neumático retorne a su posición original desenclavando el relé y repitiendo el ciclo.

El proceso inicia con la detección de un elemento metálico que pase al frente del sensor inductivo, dicha señal activa una bobina (-K2).

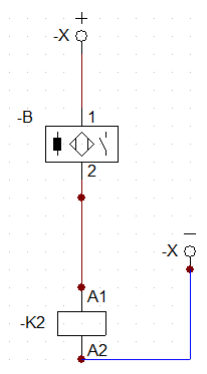


Figura 89. Detección de metal. Fuente: Elaborado por autores.

Luego la señal de la bobina (-K2) permite cerrar un contacto normalmente abierto permitiendo el enclavamiento de la bobina (-K1).

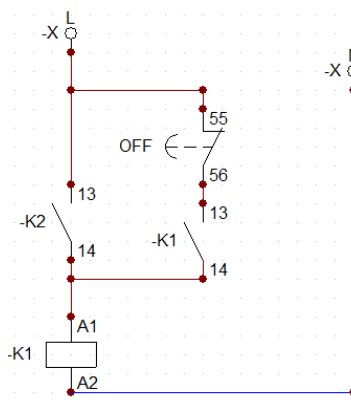


Figura 90. Enclavamiento de bobina -K1. Fuente: Elaborado por autores.

Después del accionamiento de la bobina (-K1) esta emite una señal que conmuta un contacto normalmente abierto (-K1) accionando un temporizador a la conexión (T1) durante 5 segundos.

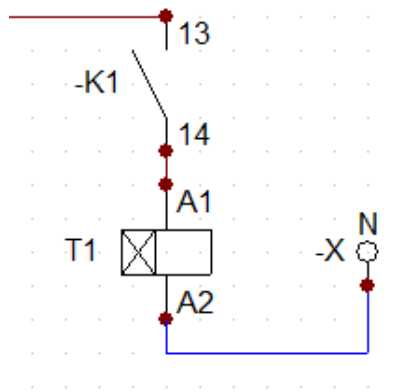


Figura 91. Área de funcionamiento del sensor. Fuente: Elaborado por autores.

Cuando han transcurrido 5 segundos después de activado el temporizador T1 se cierra el contacto normalmente abierto que posee la misma etiqueta del temporizador (T1), permitiendo accionar la electroválvula (A+) encargada de impulsar el pistón neumático y accionando a la vez otro temporizador a la conexión (OFF) el cual, pasados 2 segundos desactiva el enclavamiento de la bobina (-K1).

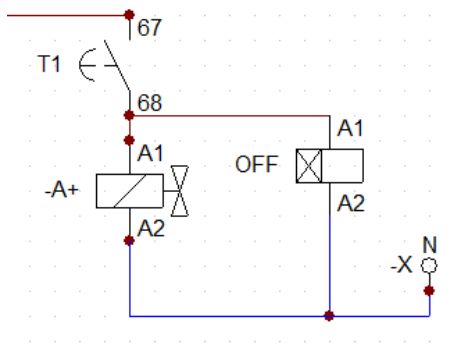


Figura 92. Accionamiento de pistón. Fuente: Elaborado por autores.

Para finalizar, el pistón debe regresar a la posición de reposo y así poder repetir el movimiento de accionamiento al momento de recibir la señal proveniente del sensor inductivo cuando este detecta un componente metálico dentro del producto.

Con el fin de darle tiempo a la banda transportadora para trasladar el producto contaminado hasta el pistón, se colocará un temporizador con retardo a la conexión.

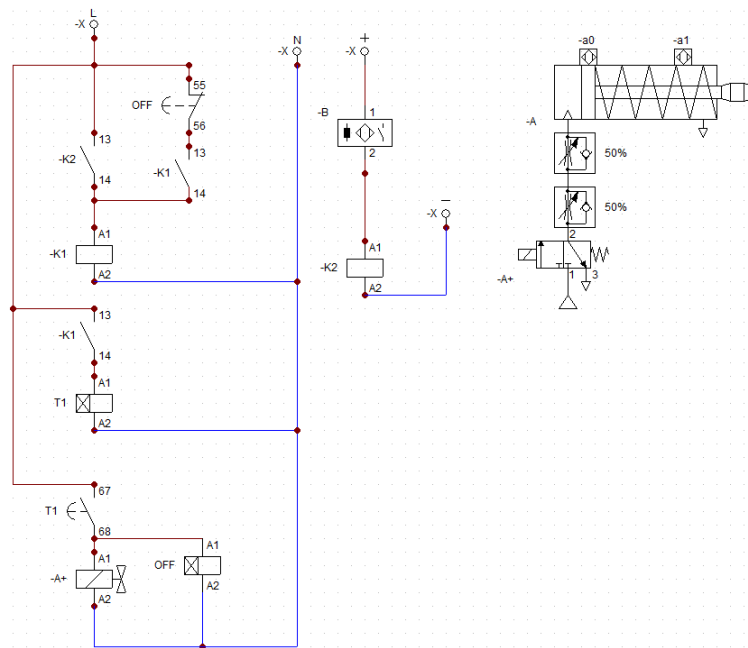


Figura 93. Sistema de rechazo por componentes metálicos. Fuente: Elaborado por autores.

XII. RESULTADOS

XII-A. Resultados diseño mecánico

XII-A1. Cálculos realizados:

1) Fuerza tangencial

g : Aceleración de la gravedad [$\frac{m}{s^2}$]

F_u : Fuerza tangencial [N]

μ_T : Coeficiente de fricción para marcha sobre mesa

μ_R : Coeficiente de fricción para marcha sobre rodillo

μ_{ST} : Coeficiente de fricción por acumulación

m : Masa de los elementos sobre la banda transportadora en toda su longitud (carga total) [kg]

m_B : Masa de la banda [kg]

m_R : Masa de todos los tambores sin el tambor motriz [kg]

En la figura 36 se presentan los valores aproximados para los coeficientes de fricción, para este diseño se escogieron los valores de $\mu_T = \mu_R = \mu_{ST} = 0,3$

Tabla II
COEFICIENTES DE FRICCIÓN APROXIMADOS. FUENTE: [HTTP://TINYURL.COM/3BBC4R39](http://tinyurl.com/3BBC4R39)

	0, A0, E0, T, U0, P	NOVO	U1, V1, VH	UH, V2H, U2H, V5H, V10H
μ_T (mesa)	0,33	0,33	0,5	0,5
μ_R (rodillo)	0,033	0,033	0,033	0,033
μ_{ST} (acumulación)	0,33	0,33	0,5	0,5

$$F_u = \mu_T * g(m + \frac{m_B}{2}) + \mu_R * g(\frac{m_B}{2} + m_R) + \mu_{ST} * g * m \quad (1)$$

$$F_u = 2,94 \frac{m}{s^2} * (1,43kg) + 2,94 \frac{m}{s^2} * (2,17kg) + 2,65 \frac{m}{s^2}$$

$$F_u = 2853,99 \frac{m}{s^2} * kg \rightarrow 3854N \quad (2)$$

2) Diámetro mínimo para tambor motriz

F_u : Fuerza tangencial [N]

d_a : Diámetro mínimo del tambor motriz [mm]

b : Ancho de la banda [mm]

β : Ángulo de contacto de tambor motriz [°]

C_3 : Factores de cálculo (tabla)

Para determinar el diámetro mínimo del tambor motriz, se considera el revestimiento en la parte inferior del tambor, que está especificado como liso y seco en el catálogo de Siegling Trasilon, de manera tal se puede observar en la “tabla III”

Tabla III
FACTORES C3. FUENTE: [HTTP://TINYURL.COM/3BBC4R39](http://tinyurl.com/3BBC4R39)

Recubrimiento de la cara inferior Siegling Trasilon	V3, V5, U2, A5, E3	V1, U1, UH	0, U0, NOVO, T, P	Factor C ₃ (válido para el tambor motriz)
Tambor de acero liso				
Seco	25	30	40	
Mojado	50	No recomendable	No recomendable	
Tambor con forro de fricción				
Seco	25	25	30	
Mojado	30	40	40	

$$d_a = \left(\frac{F_u * C_3 * 180}{b * \beta} \right) \quad (3)$$

$$d_a = \frac{13,23N * 25 * 180}{0,108m * 180^\circ}$$

$$d_a = \frac{59535}{19,44}$$

$$d_a = 3,062mm \rightarrow 0,003m \quad (4)$$

3) Distancia entre centros

L : Largo aparente de la banda (parte visible)[m]

C : Distancia entre centros [m]

r_1 : Radio del tambor menor

r_2 : Radio del tambor mayor

$$C = (L - (r_1 + r_2)) \quad (5)$$

$$C = (1,4m - (0,0335m + 0,0335m))$$

$$C = 1,33m \quad (6)$$

4) Longitud de la banda

L : Longitud de la banda [mm]

R : Radio mayor [mm]

r : Radio menor [mm]

C : Distancia entre centros [mm]

$$L = \pi(R + r) + \frac{(R - r)^2}{C} + 2C \quad (7)$$

$$L = 2870,5mm \quad (8)$$

5) Velocidad de la banda

V_L : Velocidad lineal de la banda [$\frac{m}{s}$]

N : rpm del motor [rpm]

D : Diámetro del tambor mayor [mm]

$$V_L = \frac{N(\pi * D)}{60seg} \quad (9)$$

$$V_L = \frac{10rpm(\pi * 0,067mm)}{60s}$$

$$V_L = 0,04 \frac{m}{s} \quad (10)$$

6) Velocidad angular

ω : Velocidad angular [$\frac{rad}{s}$]

V_L : Velocidad lineal de la banda [$\frac{m}{s}$]

r : Radio menor [mm]

$$\omega = \frac{V_L}{r} \quad (11)$$

$$\omega = 1,19 \frac{rad}{s} \quad (12)$$

7) RPM del tambor

η : RPM del tambor

$$\eta = \left(1,19 \frac{rad}{s}\right) * \left(\frac{1rev}{2 * \pi * rad}\right) * \left(\frac{60seg}{1min}\right) \quad (13)$$

$$\eta = 11,36rpm \quad (14)$$

8) Fuerza tangencial máxima

g : Aceleración de la gravedad [$\frac{m}{s^2}$]

F_u : Fuerza tangencial [N]

μ_T : Coeficiente de fricción para marcha sobre mesa

m : Masa de los elementos sobre la banda transportadora en toda su longitud (carga total) [kg]

m_B : Masa de la banda [kg]

$$F_u = \mu_T * g(m_n * 10 + m_B) \quad (15)$$

$$F_u = 0,8 * 9,81 \frac{m}{s^2} (0,09kg * 10 + 1,05kg)$$

$$F_u = 15,3 kg * \frac{m}{s^2} \rightarrow 15,3N \quad (16)$$

9) Fuerza de tracción máxima de la banda

$F1$: Fuerza de tracción máxima de la banda [N]

F_u : Fuerza tangencial [N]

$C1$: Factores de calculo (Para contacto de tambores, Tabla)

En la “tabla IV”, se muestra una tabla que describe los ángulos de contacto en función de los grados de contacto de la banda transportadora y su factor C1, basado en el recubrimiento del tambor motriz.

Tabla IV
FACTORES C1. FUENTE: [HTTP://TINYURL.COM/3BBC4R39](http://TINYURL.COM/3BBC4R39)

Factor C ₁ (válido para el tambor motriz)	Recubrimiento de la cara inferior Siegling Transilon			Recubrimiento de la cara inferior Siegling Transilon		
	V3, V5, U2, A5, E3			V1, U1, UH, U2H, V2H, V5H		
Ángulo de contacto β	180°	210°	240°	180°	210°	240°
Tambor de acero liso						
Seco	1,5	1,4	1,3	1,8	1,6	1,5
Mojado	3,7	3,2	2,9	5,0	4,0	3,0
Tambor con forro de fricción						
Seco	1,4	1,3	1,2	1,6	1,5	1,4
Mojado	1,8	1,6	1,5	3,7	3,2	2,9
Recubrimiento de la cara inferior Siegling Transilon						
	0, U0, NOVO, E0, A0, T, P			TX0 (AmpMiser)		
Ángulo de contacto β	180°	210°	240°	180°	210°	240°
Tambor de acero liso						
Seco	2,1	1,9	1,7	2,9	2,6	2,3
Mojado	No recomendable			No recomendable		
Tambor con forro de fricción						
Seco	1,5	1,4	1,3	1,8	1,6	1,5
Mojado	2,1	1,9	1,7	No recomendable		

$$F1 = F_u * C1 \quad (17)$$

$$F1 = 13,23N * 1,5 \quad (18)$$

$$F1 = 19,85N \quad (19)$$

10) Potencia calculada en el tambor motriz

F_u : Fuerza tangencial [N]

V_L : Velocidad lineal de la banda [$\frac{m}{s}$]

Pa : Potencia calculada en el tambor motriz [kW]

1,341[HP]=1[kW]

$$Pa = \frac{F_u * V_L}{1000} \quad (20)$$

$$Pa = \frac{13,23N * 0,04\frac{m}{s}}{1000}$$

$$Pa = 0,00053 N * \frac{m}{s} \rightarrow 0,00053 [kW] \quad (21)$$

$$Pa = 0,00053[kW] \rightarrow 0,00071[HP] \quad (22)$$

11) Potencia mecánica necesaria del motor

Pm : Potencia mecánica necesaria del motor [kW]

Pa : Potencia calculada en el tambor motriz [kW]

n : Eficiencia del accionamiento

$$Pm = \frac{Pa}{n} \quad (23)$$

$$Pm = \frac{0,00053kW}{0,75}$$

$$Pm = 0,0003975 [kW] \rightarrow 0,0005330 [HP] \quad (24)$$

12) Tensión necesaria para mover la banda

Te : Tensión efectiva para mover la banda [kg]

Pa : Potencia calculada en el tambor motriz [kW]

S : Velocidad de la banda en [$\frac{ft}{min}$] = 7,874 $\frac{ft}{min}$

$$Cte : 1HP = 33000lb * \frac{ft}{min}$$

$$Cte : 1kg = 2,2kg$$

$$Te = \frac{Pa * 33000 \frac{lb*ft}{min}}{S} \quad (25)$$

$$Te = \frac{0,00071HP * 33000 \frac{lb*ft}{min}}{7,874 \frac{ft}{min}}$$

$$Te = 0,002975 lb \rightarrow 0,001318 kg \quad (26)$$

13) Tensión del lado flojo de la banda

T_2 : Tensión del lado flojo [kg]

K : Gravedad de polea recubierta

T_e : Tensión efectiva para mover la banda [kg]

Para determinar el lado de la banda que está bajo mayor tensión, es esencial conocer el factor k de la polea, el cual depende del grado de inclinación de la banda, como se visualiza en la “tabla V”.

Tabla V
TABLA DE FACTORES K. FUENTE: [HTTP://TINYURL.COM/3BBC4R39](http://TINYURL.COM/3BBC4R39)

ANGULO DE CONTACTO EN LA BANDA DE TRANSMISION	TIPO DE TRANSMISION	TENSOR DE TORNILLO		TENSOR DE GRAVEDAD	
		POLEA LISA	POLEA RECUBIERTA	POLEA LISA	POLEA RECUBIERTA
150°	Recta	1.5	1	1.08	0.67
160°	Recta	1.4	0.9	0.9	0.60
170°	Recta	1.3	0.9	0.91	0.55
180°	Recta	1.2	0.8	0.84	0.50
190°	De contacto	1.1	0.7	0.77	0.45
200°	De contacto	1	0.7	0.72	0.42
210°	De contacto	1	0.7	0.67	0.38
220°	De contacto	0.9	0.6	0.62	0.35
230°	De contacto	0.9	0.6	0.58	0.32
240°	De contacto	0.8	0.6	0.54	0.30
340°	Tandem o Dual	0.5	0.4	0.29	0.143
360°	Tandem o Dual	0.5	0.4	0.26	0.125
380°	Tandem o Dual	0.5	0.3	0.23	0.108
400°	Tandem o Dual	0.5	0.3	0.21	0.095
420°	Tandem o Dual	0.4	0.3	0.19	0.084
440°	Tandem o Dual	-	-	0.17	0.074
460°	Tandem o Dual	-	-	0.15	0.064
480°	Tandem o Dual	-	-	0.14	0.056

$$T_2 = K * T_e \quad (27)$$

$$T2 = 0,84 * 0,002975lb$$

$$T20,0025 lb \rightarrow 0,00124 kg \quad (28)$$

14) Tensión del lado estirado de la banda

$T1$: Tensión del lado ajustado de la banda [kg]

$T2$: Tensión del lado flojo [kg]

T_e : Tensión efectiva para mover la banda [kg]

$$T1 = T2 + T_e \quad (29)$$

$$T1 = 0,0025lb + 0,002975lb$$

$$T1 = 0,005475 lb \rightarrow 0,002488 kg \quad (30)$$

15) Par o torque del motor

T : Torque del motor [$N * m$]

r : Radio de la polea motriz [m]

F : Fuerza a vencer [N]

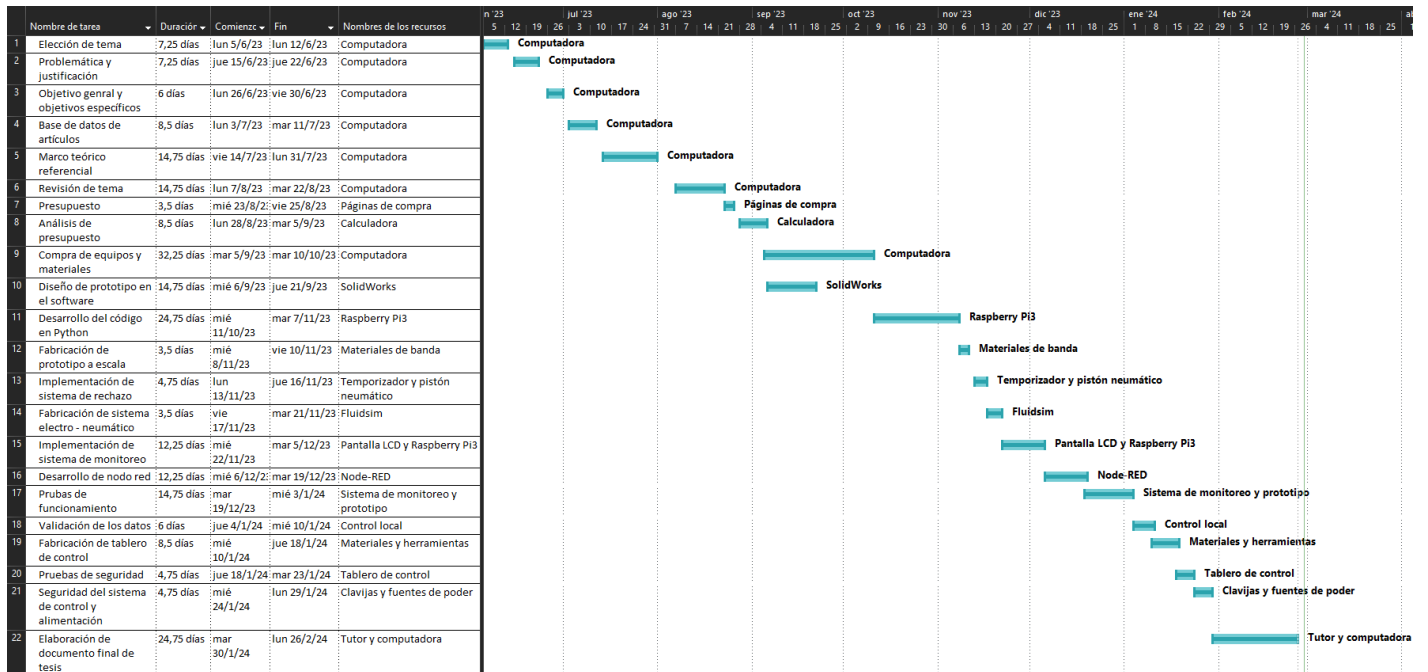
$$T = F * r \quad (31)$$

$$T = 13,23N * 0,0335m$$

$$T = 0,4432 N * m \quad (32)$$

XIII. CRONOGRAMA Y ACTIVIDADES PARA DESARROLLAR

Tabla VI
CRONOGRAMA



XIV. PROPUESTAS Y GASTOS

Tabla VII
PRESUPUESTO DE MATERIALES Y RECURSOS. FUENTE: ELABORADO POR AUTORES.

ELEMENTOS ELÉCTRONICOS/ELECTRÓNICOS	COSTO		COSTO
Motoreductor 110 a 220V a 10rpm de 125W	\$80	Cable #18 (Rollo)	\$16,96
Disyuntor de 3 polos a 20A	\$9,80	5 Presoestopas	\$4
Disyuntor de 2 polos a 10A	\$6,25	Teclado y mouse inalámbrico	\$14
5m de cable concéntrico 4x12	\$12	Sierracopa de 1/2 pulgada	\$20
Pulsador NO de 3A	\$2,23	Disco de pulir	\$4
Pulsador NC de 3A	\$2,23	Pernos autoperforantes	\$1,20
Pulsador tipo hongo para paro de emergencia	\$3,57	Contactador	\$40
Selector de 3 posiciones	\$3,12	Guardamotor	\$60
Selector de 2 posiciones	\$3	cable hdmi de 1m	\$5
Canaleta ranurada gris 15x25mm	\$3,72	Cable micro USB 1m	\$4,80
8m de cable termoencogible 5-2mm #18	\$11	Fuente de poder de 5V	\$9
2 Temporizadores ON-Delay	\$48	Case de raspberry	\$5
15 borneras eléctrica tipo Riel DIN	\$7,20	Protector de marco de pantalla	\$5
4m de cable termoencogible 5-2.5mm #12	\$8	ELEMENTOS NEUMÁTICOS	
10m cinta espiral #10 negra	\$5,80	Pistón neumático	\$32
Relé de 8 pines 24V DC	\$5,50	Electroválvula 5/2	\$20
Base relé 8 pines redondo	\$2	Válvula reguladora de caudal	\$0,80
2 módulos relé de 5V	\$4	3 Racores reductores hembra 12-10	\$4
4 sensores infrarojos	\$4,28	1m manguera plástica azul 10mm	\$1,90
5m de cable UTP	\$1,12	6m manguera plástica azul 8mm	\$9,60
LM35	\$1,60	ELEMENTOS MECÁNICOS	
Módulo MCP3008	\$6,29	Faja de PVC para banda transportadora de 2845x90x2mm	\$65
Baquelita perforada	\$0,75	Soporte para panel eléctrico	\$60
3 Paquetes de jumpers	\$9	30 Pernos de 1/4 de pulgada	\$6
Raspberry Pi 3B	\$70	2 chumaceras	\$14
Pantalla LCD de 7" para Raspberry Pi	\$40	2 Rodillos	\$60
Caja eléctrica dA28:B46e 40x30x16cm	\$70	4m cuadrado de plancha de hierro negro	\$133,32
Fuente de poder de 24V DC a 5A	\$12	4 soportes para sensores	\$17
Sensor inductivo de proximidad (PNP)	\$20	TRANSPORTE	\$250
Tomacorriente de 220V a 16A	\$2	mano de obra	\$1.500
3 Luces piloto	\$15	alimentacion	\$100
2m de canaleta de 25mm	\$8	IVA	\$15
		TOTAL	\$2.843,88

XV. CONCLUSIONES

- Los sistemas de monitoreo de producción cumplen un gran papel en las empresas que generan grandes lotes de producción, ya que permiten regularizar los procesos y tener un registro de datos de las máquinas que se encuentran operativas, así como también de las falencias que estas puedan presentar en el transcurso de la operación.
- Uno de los principales logros de este estudio fue la implementación exitosa de un sistema de monitoreo que demostró ser altamente sensible y confiable en la detección de componentes metálicos. Su integración con el prototipo de rechazo permitió una respuesta inmediata ante la detección de productos no conformes.
- Al momento de realizar las pruebas del sistema de monitoreo se implementó una Raspberry Pi 1 y se cargó el Node-RED versión 18.19.1 al sistema operativo, pero la capacidad de procesamiento impidió acceder a todos los apartados y paquetes de datos. Por lo que fue de suma importancia utilizar una versión más actualizada del raspberry para no tener ese tipo de problema.
- Además, se observó que el sistema de monitoreo desarrollado fue capaz de adaptarse a diferentes tipos de productos y entornos de producción, hecho que lo vuelve versátil y aplicable en una variedad de industrias. La flexibilidad del sistema también se reflejó en su capacidad para ajustarse a diferentes niveles de sensibilidad y configuraciones específicas de detección, lo que permite una personalización según las necesidades del cliente.
- Una vez realizada la implementación del sistema al prototipo a escala, se recibieron los datos de la cantidad de productos que se inyectaban a la banda transportadora, pero existía pérdida de información debido a la baja eficiencia de la conexión a internet del dispositivo, así que se optó por conectar la raspberry de forma directa a través de un cable de red y la eficiencia de lectura mejoró considerablemente.

XVI. RECOMENDACIONES

- Al momento de realizar la implementación del sistema sea a un prototipo o línea de producción verificar la conexión wifi disponible en el área ya que el tiempo de respuesta y la visualización de datos en el dashboard depende mucho de este factor y si es posible realizar la conexión del equipo en este caso las raspberry por medio de cable de red o ethernet.
- Realizar los cálculos de consumo correspondientes referentes a los componentes electrónicos que se van a conectar a la raspberry, ya sean módulos de relé, sensores u otro tipo de componente que requieran alimentación del controlador puesto que no llevar un seguimiento correspondiente de los datos asignados puede generar problemas de corriente afectando la potencia y rendimiento del controlador reflejándose a los demás circuitos.
- Para evitar problemas de corriente que se vean reflejados como pérdida de eficiencia y rendimiento, es recomendable utilizar fuentes alternas de 3,3 y 5 Volteos, librando energizar los componentes que sobrepasen el consumo energético capaz de suministrar la raspberry pi.
- Usar un controlador con mayor capacidad de procesamiento para tener un manejo completo y eficiente de todos los sistemas involucrados e implementar versiones de nodo red con mayores prestaciones que las admitidas por la raspberry pi 3.
- Implementar un perfil de usuario y contraseña al Node-RED para evitar la filtración de información y ataques a la base de datos.
- Implementar visión artificial en el sistema de monitoreo para calcular otro tipo de parámetros que no están especificados en el protocolo del prototipo de rechazo al que se lo implemento.
- Utilizar un túnel detector de metales para tener mayor capacidad de lectura y por ende someter el sistema a otro tipo de productos.

XVII. ANEXOS

XVII-A. *Prototipo de banda transportadora en formato CAD*



Figura 94. Banda transportadora Solid works. Fuente: Elaborado por autores.

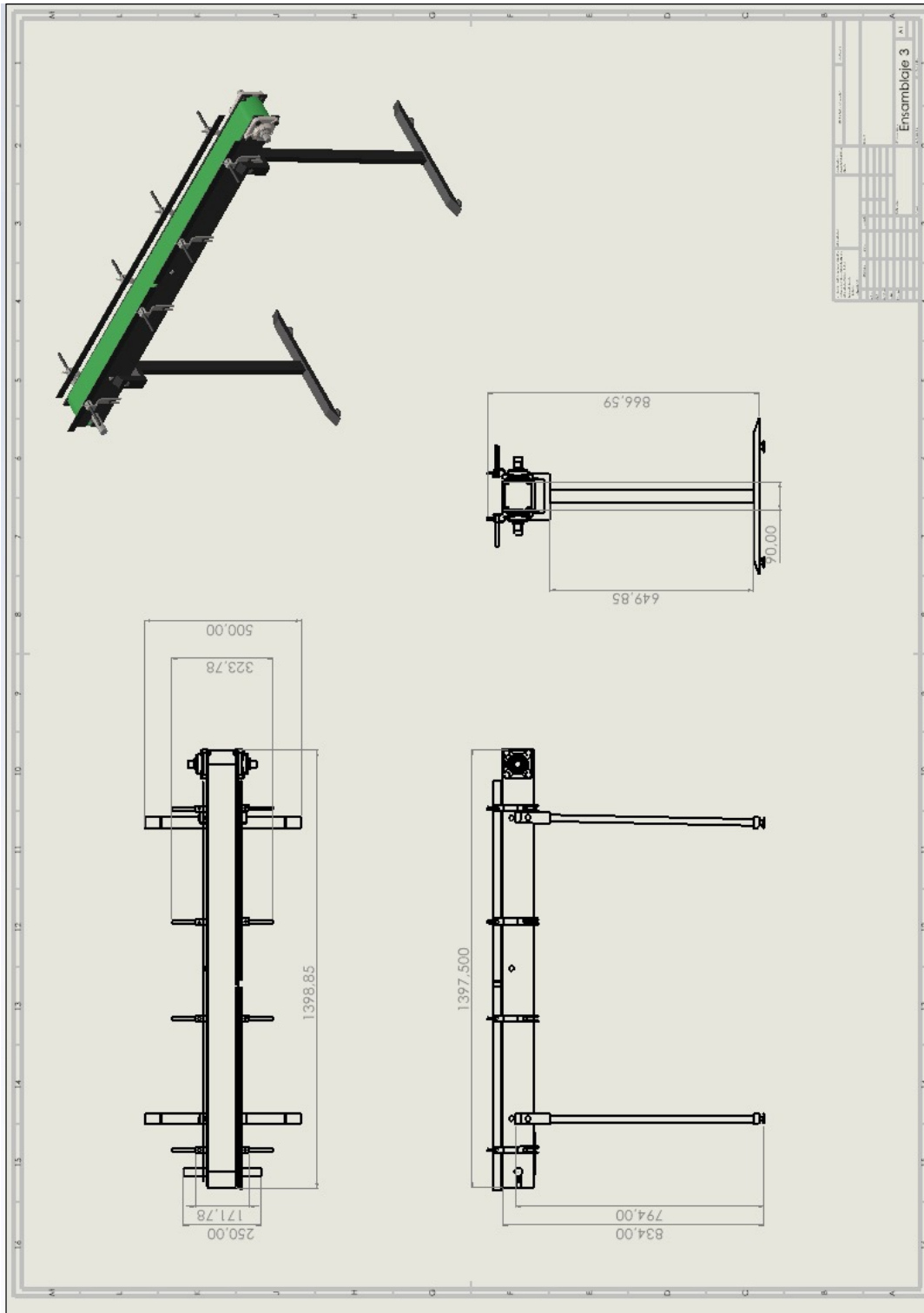


Figura 95. Plano vistas de banda transportadora. Fuente: Elaborado por autores.

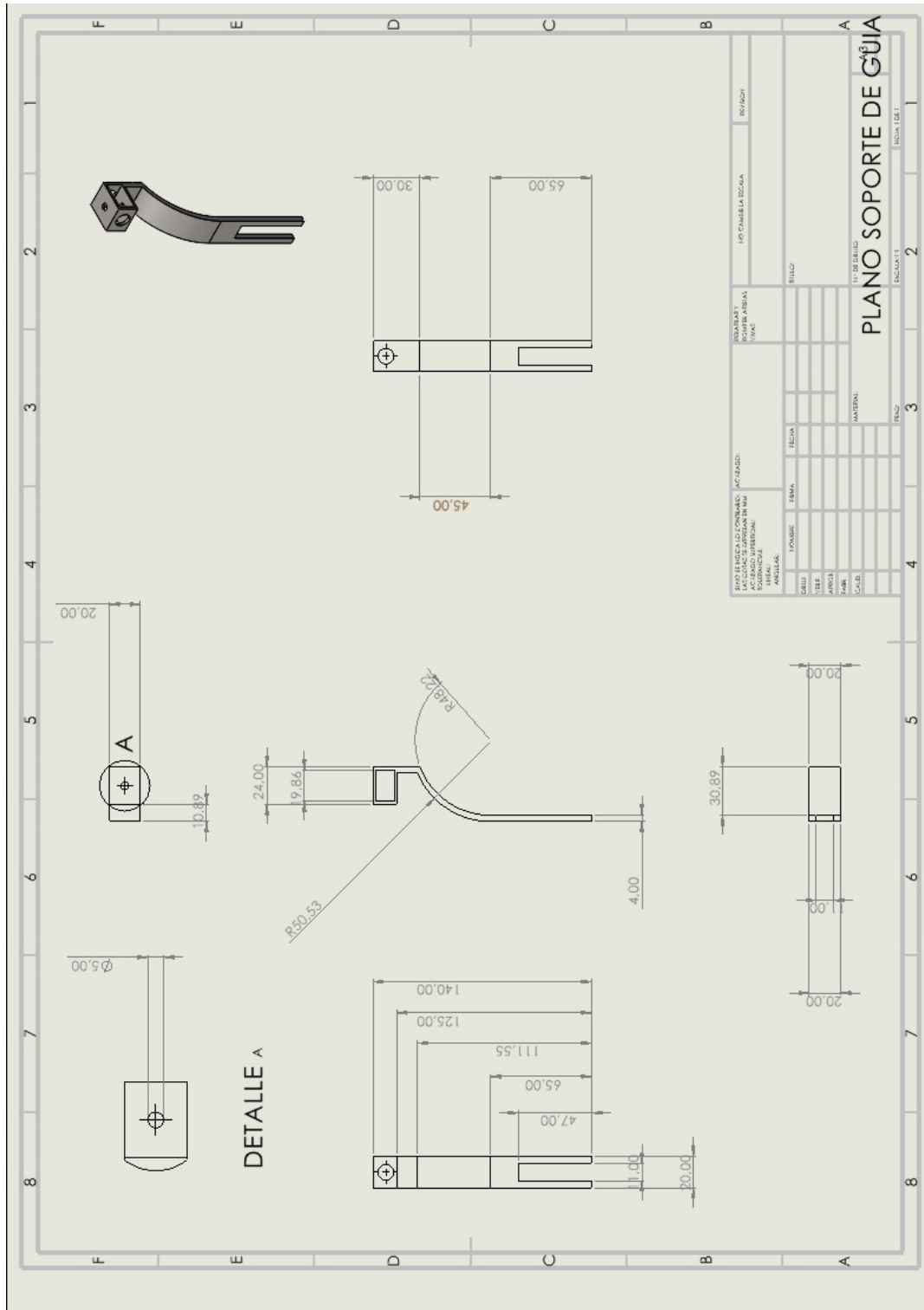


Figura 98. Plano soporte de guía. Fuente: Elaborado por autores.

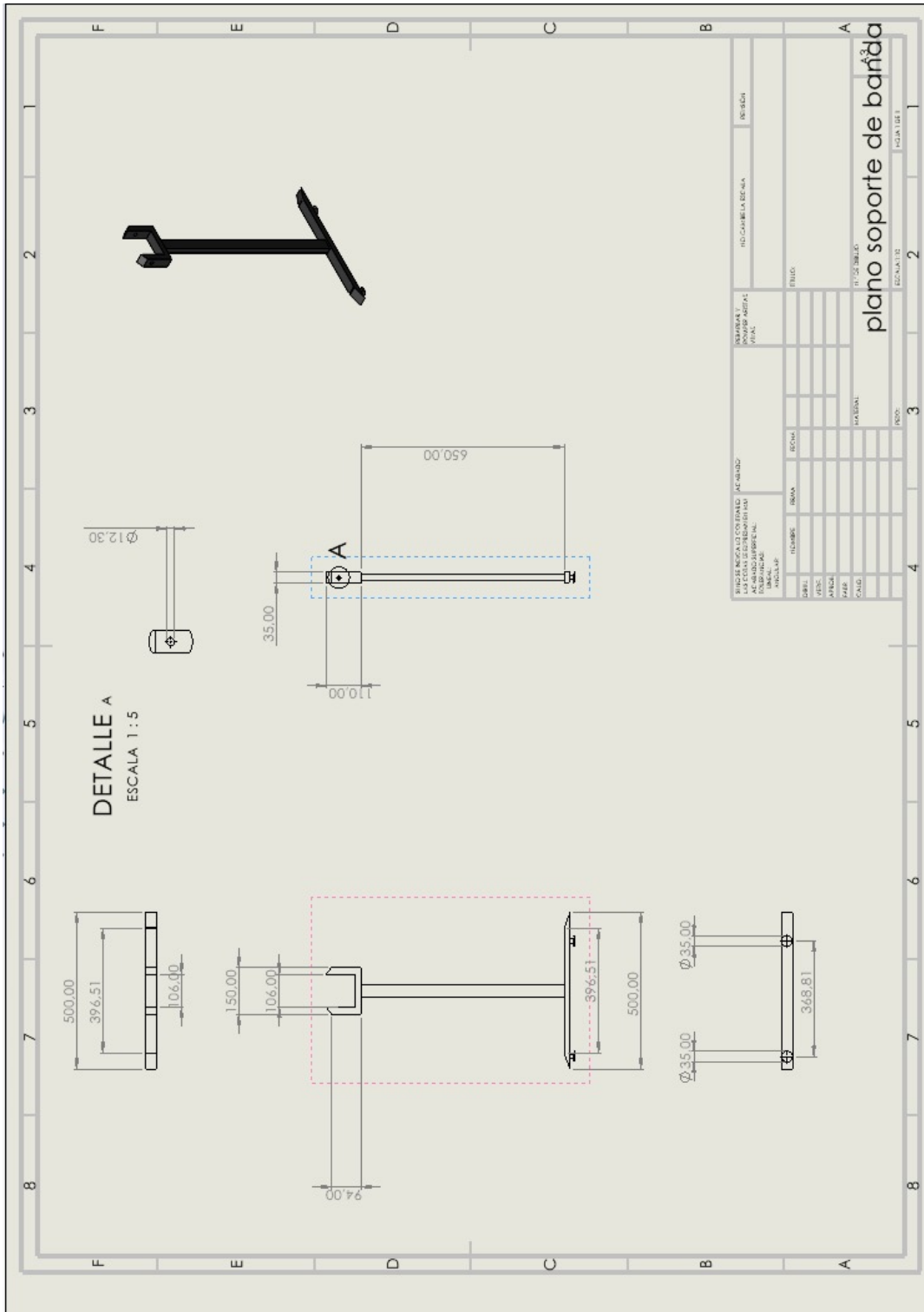


Figura 99. Plano soporte de banda. Fuente: Elaborado por autores.

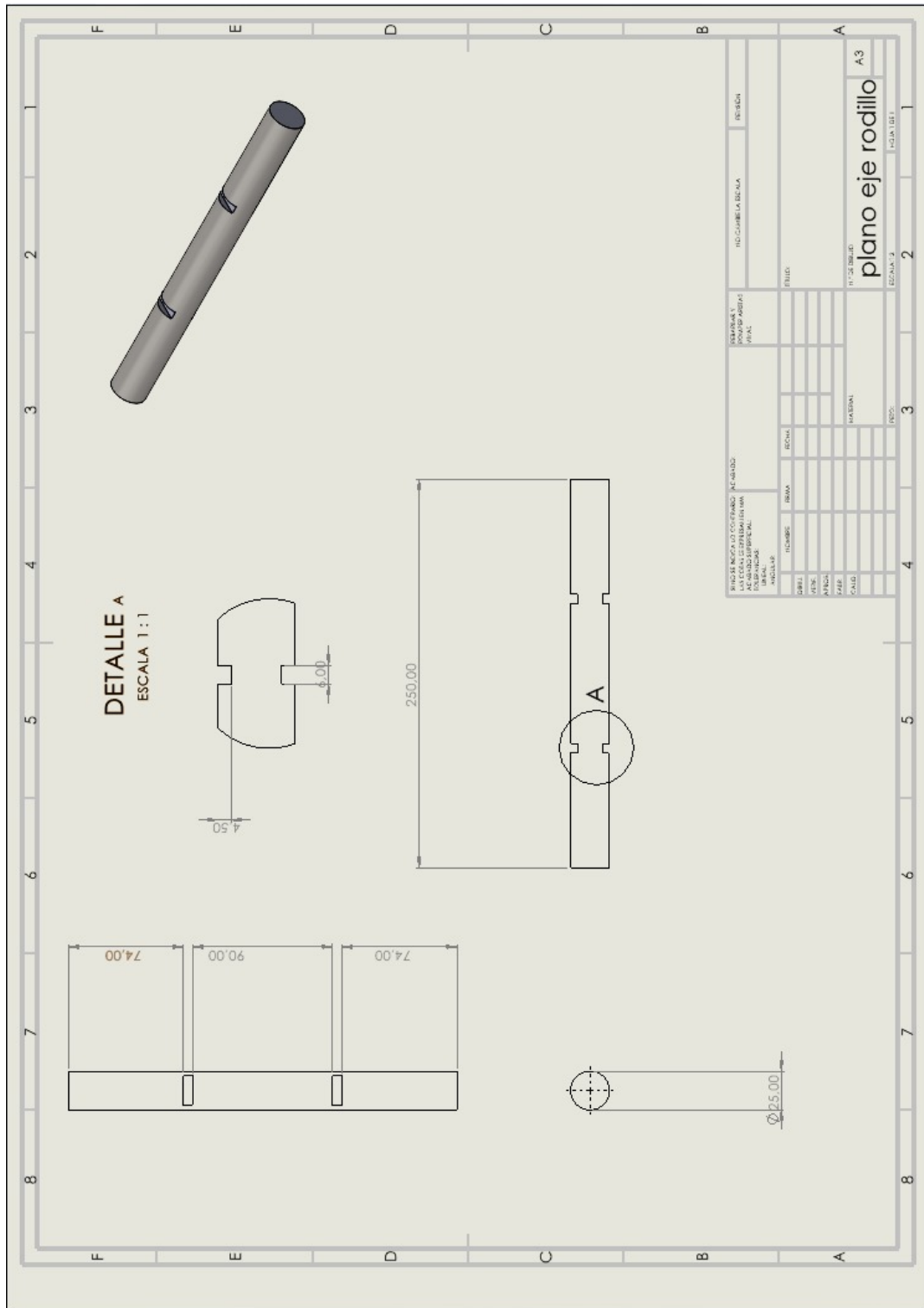


Figura 102. Plano eje rodillo. Fuente: Elaborado por autores.

XVII-B. Fotografías del montaje de la banda transportadora

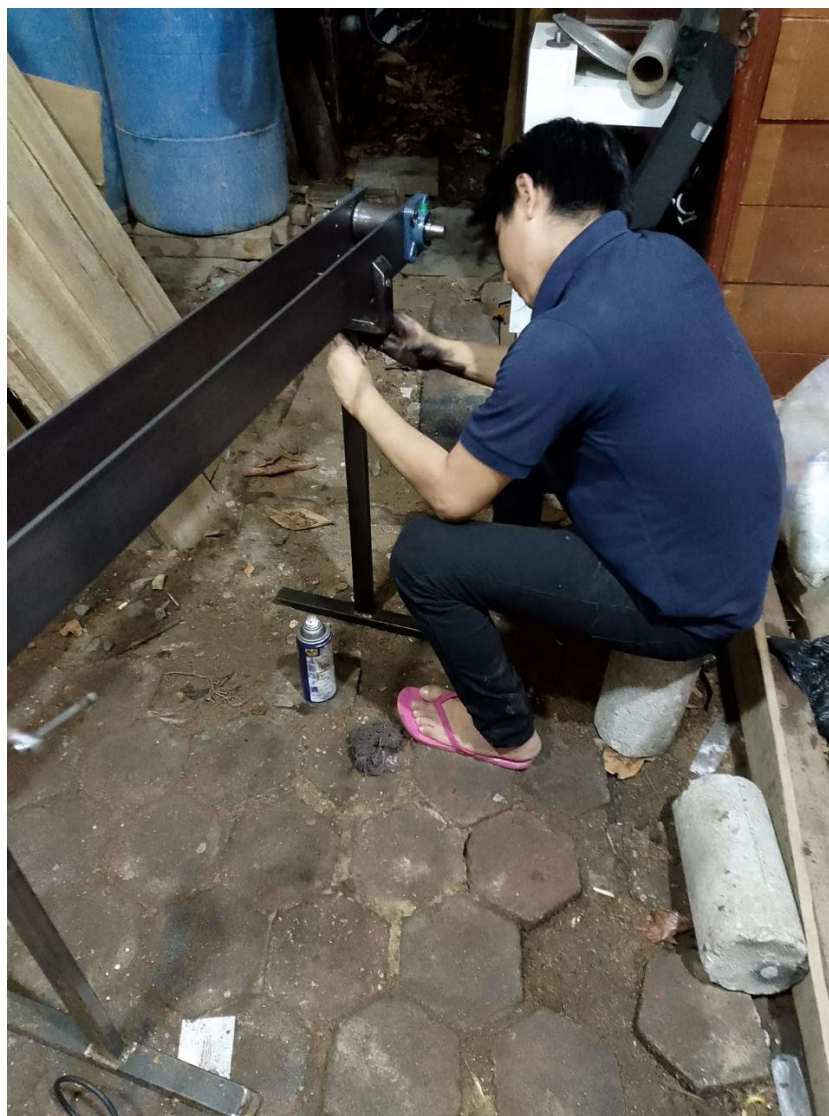


Figura 103. Lijado de estructura. Fuente: Elaborado por autores.



Figura 104. Soldadura de templadores de rodillo. Fuente: Elaborado por autores.



Figura 105. Limpieza de guías. Fuente: Elaborado por autores.



Figura 106. Pintado de estructura. Fuente: Elaborado por autores.



Figura 107. Pintado y secado de guías para banda. Fuente: Elaborado por autores.



Figura 108. Pintado y secado de soportes de guías. Fuente: Elaborado por autores.

XVII-C. *Fotografías del armado del panel eléctrico*



Figura 109. Conexión de componentes a la raspberry. Fuente: Elaborado por autores.



Figura 110. Conexiones para prueba de arranque. Fuente: Elaborado por autores.



Figura 111. Conexión de componentes en caja eléctrica. Fuente: Elaborado por autores.



Figura 112. Perforación para botoneras de control. Fuente: Elaborado por autores.



Figura 113. Conexión y Visualización de Pantalla touch. Fuente: Elaborado por autores.



Figura 114. Implementación de pantalla y botonera en tablero. Fuente: Elaborado por autores.

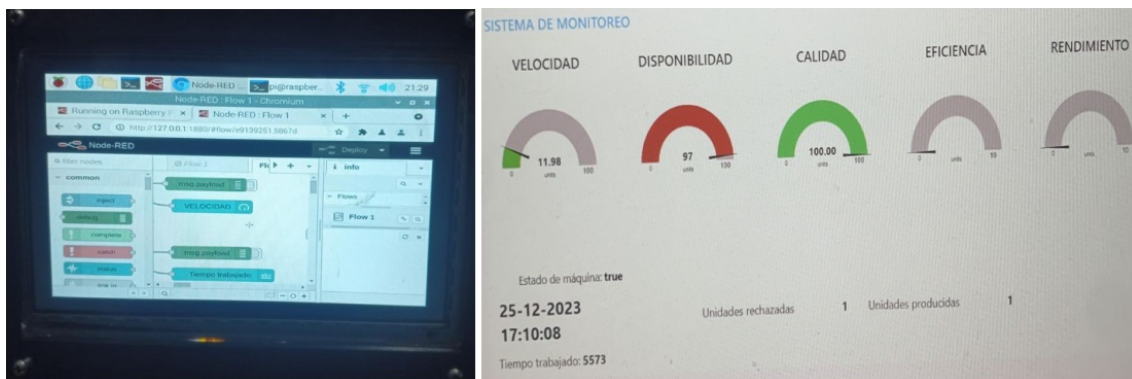


Figura 115. Panel de monitoreo en Node-RED. Fuente: Elaborado por autores.



Figura 116. Panel de control con soporte. Fuente: Elaborado por autores.



Figura 117. Instalación del panel de control. Fuente: Elaborado por autores.

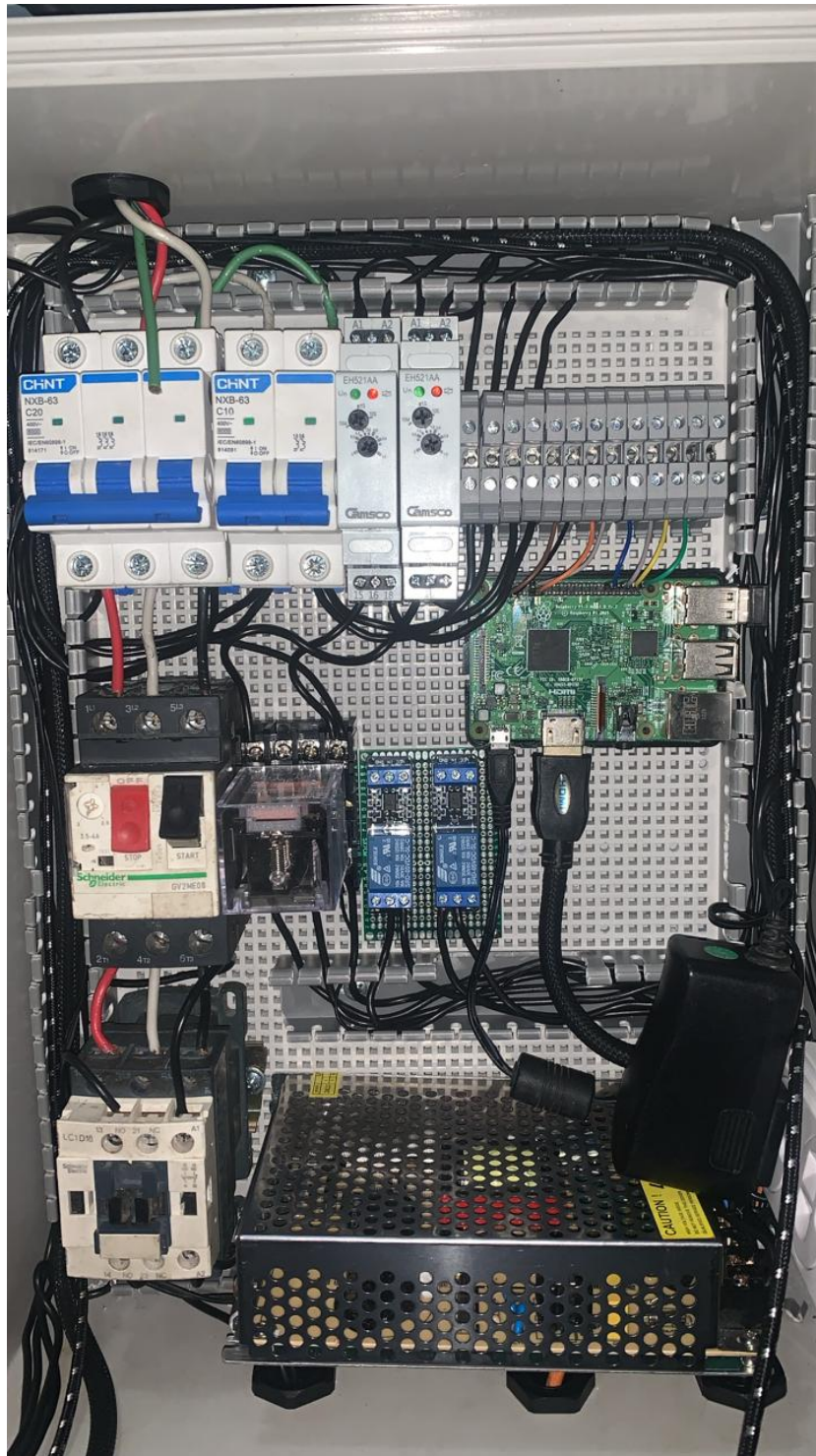


Figura 118. Panel de control. Fuente: Elaborado por autores.

REFERENCIAS

- [1] Alamy, «Coca Cola, la fábrica, la producción,» [*Disponible en: <http://tinyurl.com/2d427bzu>; Accedido el: 20 de diciembre del 2023*], 2023.
- [2] J. A. C. Álvarez, «Encoder óptico incremental,» [*Disponible en: <http://tinyurl.com/3z282ysn>; Accedido el: 22 de agosto del 2023*], 2016.
- [3] J. A. Arias Cadena, «Automatización de la parte final del subproceso de refinación y del subproceso de empaquetado de la línea de producción de azúcar refinada del Ingenio Risaralda,» 2019.
- [4] W. M. Calderón Mendoza, «Implementación de la gestión del mantenimiento de las talladoras para disminuir las paradas no programadas en la Empresa Topsa Productos Ópticos SA,» 2014.
- [5] A. Cardona Ospina, W. A. Mojica Coy et al., «Diseño y construcción de un área para la fabricación de líquidos no estériles en un laboratorio farmacéutico,» B.S. thesis, Universidad Piloto de Colombia, 2013.
- [6] F. CEMBRANOS NISTAL, *Automatismos eléctricos, neumáticos e hidráulicos* (Electricidad - Electrónica). Ediciones Paraninfo, S.A, 2008, ISBN: 9788497326582. dirección: <https://books.google.com.ec/books?id=TMa-xuhAUiIC>.
- [7] A. Chile, «Módulo Relé de Un Canal 5Vdc con Led,» [*Disponible en: <http://tinyurl.com/4nd8bj4p>; Accedido el: 27 de agosto del 2023*], 2021.
- [8] D. A. Córdor Chuqui, «Automatización de un sistema de refrigeración industrial mediante un PLC, interface HMI y control remoto desde dispositivos móviles,» B.S. thesis, Quito, 2019.
- [9] M. Dávila, J. Marcos, F. Rodríguez y C. Vivas, «Maqueta para la realización de prácticas de laboratorio con sensores de proximidad inductivos y capacitivos,» 1970.
- [10] DIELCO, «GUARDAMOTORES STECK,» [*Disponible en: <http://tinyurl.com/4bu2nx2e>; Accedido el: 28 de septiembre del 2023*], 2019.
- [11] D. Electrónicas, «Válvula solenoide neumática de 12V,» [*Disponible en: <https://tinyurl.com/34b3t5e4>; Accedido el: 2 de enero del 2024*], 2020.
- [12] A. E. Fitzgerald, C. Kingsley y S. D. Umans, «Electric machinery,» 2003.
- [13] L. S. Flórez Escobar y O. d. J. Cadavid Ramírez, «Diseño y cálculo de banda transportadora de residuos sólidos para la empresa del sector alimenticio inversiones Trocifres SA,» 2017.
- [14] D. A. Gutiérrez Díaz et al., «Construcción de un banco de automatización basado en una cinta transportadora horizontal,» Tesis doct., Universidad de Talca. Facultad de Ingeniería, 2019.
- [15] HETPRO, «Sensor inductivo de proximidad PNP,» [*Disponible en: <http://tinyurl.com/45ahx3vm>; Accedido el: 22 de septiembre del 2023*], 2022.
- [16] HUBOT, «Pantalla 7" LCD Raspberry Táctil,» [*Disponible en: <http://tinyurl.com/3x9u5kbd>; Accedido el: 28 de septiembre del 2023*], 2022.
- [17] I. suministros industriales, «La gran importancia que conlleva revisar correctamente un motor eléctrico,» [*Disponible en: <http://tinyurl.com/3f6hpu4>; Accedido el: 28 de noviembre del 2023*], 2023.
- [18] F. R. Jacobs, W. L. Berry, D. C. Whybark y T. E. Vollmann, *Manufacturing planning and control for supply chain management: APICS/CPIM Certification Edition*. McGraw-Hill Education, 2011.
- [19] N. K. Kamila, *Handbook of research on wireless sensor network trends, technologies, and applications*. IGI global, 2016.
- [20] I. L. Kosow, «Electric machinery and transformers,» 1991.
- [21] J. Kurose y K. W. Ross, *Redes de computadoras*. Pearson educación, 2010, vol. 5.
- [22] LEPSAC, «ETR2-11 262684 ELECTRIC Relé temporizador 1,» [*Disponible en: <http://tinyurl.com/mryb9sh2>; Accedido el: 14 de agosto del 2023*], 2021.
- [23] W. Liao y T. Gu, «Optimization and control of production systems based on interval speed continuous Petri nets,» en *2005 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, vol. 2, 2005, págs. 1212-1217.
- [24] L. Llamas, «Detector de obstáculos con sensor infrarrojo y Arduino,» [*Disponible en: <http://tinyurl.com/5n89udt8>; Accedido el: 5 de agosto del 2023*], 2018.
- [25] A. P. Malvino, D. J. Bates et al., «Principios de electrónica,» vol. 6, 1991.

- [26] D. Martínez Martínez, «Diseño e implementación de un sistema de control y monitorización remoto de bajo coste mediante Raspberry Pi y gestión en la nube con Firebase,» Tesis doct., Universitat Politècnica de València, 2022.
- [27] D. Morales Navarro y D. Vila Morales, «Relé temporizador multifunción estomatología,» *Revista Cubana de Estomatología*, vol. 53, n.º 1, págs. 67-83, 2016.
- [28] Novatronic, «Fuente de poder 24V 20A 480W,» [Disponible en: <http://tinyurl.com/bdfdw9b2>; Accedido el: 28 de septiembre del 2023], 2019.
- [29] C. J. Ramírez Sarmiento, «Diseño de un sistema de producción y operaciones para la fabricación de anti-bióticos sólidos en la línea de genéricos e institucionales, en una planta farmacéutica nacional.,» 2018.
- [30] M. H. Rashid, «Power electronics: circuits, devices, and applications,» 2009.
- [31] T. S. REDES, «ELEMENTOS DE UNA RED,» [Disponible en: <https://tinyurl.com/mubmamfj>; Accedido el: 15 de noviembre del 2023], 2020.
- [32] P. Roberts y M. Baker, «Electrical Installation Work: Level 2: EAL Edition,» 2015.
- [33] G. Rodriguez, «Relay Encapsulados CAMSCO 8 Pines,» [Disponible en: <http://tinyurl.com/we46b7k7>; Accedido el: 11 de agosto del 2023], 2020.
- [34] RS, «Cilindro neumático con vástago RS PRO,» [Disponible en: <http://tinyurl.com/24exc4ee>; Accedido el: 14 de agosto del 2023], 2020.
- [35] M. A. Salam, «Electromagnetic field theories for engineering,» 2014.
- [36] M. Salinero Gervaso, «Diseño de una banda transportadora mediante guide de Matlab,» Tesis de mtría., 2013.
- [37] A. G. Salvador, «Introducción a la neumática,» 1988.
- [38] K. W. Saylor y M. C. Roberts, «Implementation Science Can Do Even More for Translational Ethics,» *The American Journal of Bioethics*, vol. 20, n.º 4, págs. 83-85, 2020.
- [39] J. V. Schooneveld, «Build Physical Projects With Python on the Raspberry Pi,» [Disponible en: <http://tinyurl.com/5n7as4da> ;Accedido el: 10 de noviembre del 2023], 2023.
- [40] A. Serna, F. Ros y J. Rico, «Guía práctica de sensores,» 2010.
- [41] R. S. Srivastava, A. Kumar, S. Sharma, H. Thakur, S. Patel y R. Vaish, «Development and applications of thermoelectric based dehumidifiers,» *Energy and Buildings*, vol. 252, pág. 111 446, 2021.
- [42] J.-T. Teng, H.-L. Yang y M.-S. Chern, «Economic order quantity models for deteriorating items and partial backlogging when demand is quadratic in time,» *European Journal of Industrial Engineering*, vol. 5, n.º 2, págs. 198-214, 2011.
- [43] Truck, «Mantenimiento preventivo en bandas transportadoras,» [Disponible en: <http://tinyurl.com/4ypwjm2s>; Accedido el: 10 de noviembre del 2023], 2023.
- [44] B. M. Vallejo y S. B. Vallejo, «Aspectos generales de la automatización industrial del sector farmacéutico,» *Revista Colombiana de Ciencias Químico-Farmacéuticas*, vol. 35, n.º 1, págs. 47-63, 2006.

APÉNDICE