



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIA Y TECNOLOGÍA
ESCUELA DE INGENIERÍA ELECTRÓNICA

**Diseño e Implementación de la Arquitectura del Internet
Industrial de las Cosas para la Recolección de Datos en un
Proceso de Manufactura**

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

Autores:

IVÁN FELIPE VELASTEGUÍ JARAMILLO

DAVID FERNÁNDEZ DE CÓRDOVA

Director:

HUGO MARCELO TORRES SALAMEA

CUENCA-ECUADOR

2020

DEDICATORIA

Agradezco a Dios por todos y cada una de las personas que forma parte de esta dedicatoria, forman parte de mi vida y que me han apoyado en este camino.

A mis padres Iván y Cesibel por su dedicación, entrega, enseñanza; fortaleza y persistencia; amor y cariño incondicional. Este esfuerzo es fruto de todo su trabajo y educación que me han brindado.

A mis hermanos Andrea, Cesibel y Fernando por permanecer conmigo en todos los momentos de mi vida, ser mi ejemplo a seguir y enseñarme a luchar para conseguir mis objetivos y sueños.

A mi abuelita Mami Vicky por ser pilar familiar y verdadero ejemplo a seguir para nuestras generaciones, desde el cielo sé que estarás orgullosa de todos nuestros logros.

A mis abuelitos Papi Alcides y Mami Yoli por su cariño, crianza y por tener siempre las puertas abiertas para mí y mi familia en todos los momentos de nuestra vida.

A todos mis familiares que siempre están presentes para los buenos y malos momentos, en quienes puedo confiar con absoluta certeza y forman parte importante en mi vida.

Finalmente, a todos mis amigos por su verdadera amistad que prevalece ante cualquier situación.

IVÁN VELASTEGUÍ

DEDICATORIA

En primer lugar, dedico este proyecto de tesis a ti, Mi Mamá Pepita, por estar en cada instante de mi vida junto a mí, desde el momento en el que me tuviste en tus brazos, persiguiendo conmigo cada una de mis metas propuestas, y ahora desde el cielo sé que podrás ver que se ha cumplido uno de tus grandes sueños.

A mi querido padre Patricio, por permitirme tener el estudio necesario para convertirme en profesional y ser un apoyo incondicional con tu sabiduría para tomar de una manera correcta mis caminos.

A mi madre Viviana, y mis hermanas Daniela y Carolina, por ser un ejemplo a seguir de lucha ante las situaciones más adversas de la vida y siempre alimentar esa parte de mí que me da alegría y felicidad.

A mis tíos Fernanda y Carlos, por enseñarme como la perseverancia te llevar cumplir los planes propuestos a futuro, y que así todos los arduos esfuerzos tienen su recompensa.

A mis primos Carlitos y Ariana, los más pequeños de la familia, esperando que algún día puedan enseñar todo su potencial y de igual forma convertirse en profesionales y personas de bien.

Finalmente, a mis amigos de toda la vida, por llegar a convertirse en familia más que amigos, y ser mi compañía en los momentos de felicidad y tristeza.

DAVID FERNÁNDEZ DE CÓRDOVA

AGRADECIMIENTO

En primer lugar, agradecemos a nuestros padres por el esfuerzo y sacrificio que han realizado para poder facilitarnos el estudio en una institución universitaria.

A la Universidad del Azuay por permitirnos estudiar dentro de sus instalaciones y poder aprender más y más cada día con sus docentes y estudiantes.

Agradecemos a nuestro director de tesis por habernos brindado el conocimiento, tiempo y su ayuda oportuna durante la realización de este proyecto. Así como también a los consejos y sugerencias de nuestro tribunal.

De igual forma a nuestros demás familiares y amigos por ser apoyo incondicional a lo largo de toda la carrera.

IVÁN VELASTEGUÍ

DAVID FERNÁNDEZ DE CÓRDOVA

ÍNDICE DE CONTENIDOS

CONTENIDOS	Pág.
DEDICATORIA	ii
ÍNDICE DE CONTENIDOS	v
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xv
RESUMEN	16
ABSTRACT	17
INTRODUCCIÓN	18
CAPÍTULO 1: ANTECEDENTES DE INVESTIGACION	21
1.1 Terminología y Definiciones	21
1.1.1 Internet de las cosas	21
1.1.2 Industria 4.0	21
1.1.3 Internet Industrial de las Cosas	22
1.2 Estado del arte	23
1.2.1 Aplicaciones e investigaciones acerca de IIoT e Industria 4.0	23
1.2.2 Fábricas que utilizan el concepto Industria 4.0	30
1.3 Protocolo MQTT	30
1.3.1 Funcionamiento del protocolo MQTT	31
1.3.2 Seguridad en el protocolo MQTT	32
1.3.3 Calidad de servicio (QoS) en los mensajes MQTT	33
1.3.4 Estructura de los topics en los mensajes MQTT	33
1.3.5 Sugerencias en la creación de topics, según HiveMQ:	36
1.3.6 Ejemplo de mensajería MQTT	38

1.4	Hardware	41
1.4.1	PLC Siemens S7-1200 1212C.....	41
1.4.2	Módulo de expansión para PLC S7-1200	42
1.4.3	Módulo Siemens IoT 2040.....	43
1.5	Descripción del banco didáctico.....	44
1.5.1	Componentes del banco didáctico.....	45
1.5.2	Esquema del banco didáctico	49
1.5.3	Proceso industrial simulado en el banco didáctico	50
CAPÍTULO 2: ARQUITECTURA, BASE DE DATOS Y VISUALIZACIÓN		52
2.1	Arquitectura del proyecto	52
2.1.1	Implementación del sistema.....	52
2.1.2	Funcionamiento del sistema.....	52
2.2	Plataforma IoT.....	53
2.2.1	ThingsBoard como plataforma IoT.....	54
2.3	Servidor en la nube.....	56
2.4	Arquitectura de la plataforma ThingsBoard	58
2.4.1	Sistemas externos.....	59
2.4.2	Base de datos de la plataforma	59
2.4.3	Arquitectura de despliegue: microservicios o monolítico.....	59
2.4.3.1	Implementación monolítica	59
2.4.3.2	Implementación en microservicios	60
2.4.4	Características de la arquitectura de ThingsBoard:.....	61
2.5	Características de ThingsBoard.....	61
2.6	Comparativa entre la versión de pago y la versión gratuita open-source.....	63

2.6.1	Características de ThingsBoard Community Edition	64
2.6.2	Características de ThingsBoard Professional Edition	65
2.7	Documentación de ThingsBoard	66
2.7.1	Seguridad.....	66
2.7.2	Roles de usuario	68
2.7.3	Activos	70
2.7.4	Dispositivos.....	71
2.7.5	Alias de las Entidades	72
2.7.6	Motor de Reglas	74
2.7.7	Comandos RPC	75
2.7.8	Widgets Disponibles	76
2.7.8.1	Tipos de widgets	77
2.7.9	Paquetes de widgets	83
2.7.9.1	Paquete Digital Gauges.....	84
2.7.9.2	Paquete Analogue Gauges	84
2.7.9.3	Paquete Cards	85
2.7.9.4	Paquete Charts	85
2.7.9.5	Paquete Alarm Widgets	86
2.7.9.6	Paquete Control Widgets	86
2.7.9.7	Paquete Date	87
2.7.9.8	Paquete Entity Admin Widgets.....	87
2.7.9.9	Paquete Gateway Widgets	88
2.7.9.10	Paquete Input Widgets	88
2.7.9.11	Paquete GPIO Widgets	89

2.7.9.12	Paquete Maps.....	89
2.7.10	Importar/Exportar Paneles, Widgets (Tipos Y Paquetes)	90
2.8	ThingsBoard API.....	91
2.8.1	API MQTT	91
2.8.2	API RPC.....	94
CAPÍTULO 3: CONFIGURACIÓN Y PROGRAMACIÓN DE LOS EQUIPOS DE CONTROL.....		96
3.1	Programación del PLC 1212C AC/DC/RLY Siemens.....	96
3.1.1	Secuencia inicio [FB4].....	96
3.1.2	Proceso banda [FB1].....	97
3.1.3	Robot tanque FB3 y Robot deposito [FB2].....	97
3.1.4	Envío de datos de IoT [FC1].....	97
3.1.5	Envío de tiempos para IoT [FC2].....	98
3.1.6	Control manual [FC3]	98
3.1.7	COM_IOT [DB1].....	98
3.2	Configuración de las propiedades del PLC.	99
3.3	Configuración del módulo Siemens IoT 2040.....	100
3.3.1	Configuración inicial del módulo IoT.....	100
3.3.2	Instalación de Node-Red en el módulo IoT	111
3.3.3	Nodos de programación en Node-Red	123
3.3.4	Conexión de nodos y envío de correo electrónico	128
3.3.5	Inicio Automático de Node-Red	130
CAPÍTULO 4: RESULTADOS		132
4.1	Conexión del banco didáctico.	132

4.2	Programación del PLC.	132
4.3	Programación del módulo IoT 2040.....	133
4.4	Instalación y configuración de los servidores en la nube	134
4.4.1	Configuración de IP Estática para las Máquinas Virtuales en Google Cloud 136	
4.4.2	Configuración de Firewall en Google Cloud	136
4.4.3	Instalación de la Plataforma ThingsBoard Community Edition	137
4.4.4	Instalación de la Plataforma ThingsBoard Professional Edition.....	137
4.5	Plataforma ThingsBoard.....	138
4.5.1	Pestaña OVERVIEW.	141
4.5.2	Pestaña ALARMAS.....	142
4.5.3	Pestaña GRÁFICOS.....	143
4.5.4	Pestaña CONTROL.....	144
4.6	Pruebas integrales.....	146
4.7	Visualización	151
	CONCLUSIONES Y RECOMENDACIONES.....	152
	Conclusiones:	152
	Recomendaciones:.....	155
	REFERENCIAS	157
6.	ANEXOS	161
6.1	Anexo 1: Brokers MQTT	161
6.2	Anexo 2: Configuraciones de los controladores PID externos.....	164
6.3	Anexo 3: Diagrama de conexión del banco didáctico	166
6.4	Anexo 4: Código implementado en el PLC s7-1200 1212c AC/DC/RLY.....	167

ÍNDICE DE FIGURAS

Figura 1.1: Diagrama de radar de la influencia de la Industria 4.0.	23
Figura 1.2: Arquitectura de una plataforma IIoT.	24
Figura 1.3: Layout de la planta en Factory I/O.	25
Figura 1.4: Panel de control del usuario Operador y usuario Ingeniero.	25
Figura 1.5: Diagrama de bloques de la aplicación.	26
Figura 1.6: Implementación física del proyecto y funcionamiento de la página web.	27
Figura 1.7: Ciclo de interacción de actores y elementos de la arquitectura propuesta. ...	28
Figura 1.8: Arquitectura propuesta para Industria 4.0.	28
Figura 1.9: Interfaz de usuario basada en la arquitectura planteada.	30
Figura 1.10: Protocolo MQTT, modelo publicador/suscriptor.	31
Figura 1.11: Estructura de un mensaje MQTT.	32
Figura 1.12: Separación de niveles de topics.	33
Figura 1.13: Utilización de wildcard +.	34
Figura 1.14: Resultado de búsqueda mediante Wildcard +.	35
Figura 1.15: Utilización de Wildcard #.	35
Figura 1.16: PLC Siemens S7-1200 1212 AC/DC/RLY.	41
Figura 1.17: Modulo de expansión utilizado en el banco didáctico.	42
Figura 1.18: Módulo IoT 2040 utilizado en el banco didáctico.	43
Figura 1.19: Banco didáctico a utilizar.	44
Figura 1.20: Sensor capacitivo.	45
Figura 1.21: Sensor inductivo.	45
Figura 1.22: Termocupla.	46
Figura 1.23: Sensor óptico reflectivo.	46
Figura 1.24: Sensor magnético.	47
Figura 1.25: Sensor de ultrasonido.	47
Figura 1.26: Controlador PID Bielco 9300.	48
Figura 1.27: Conversor de termocupla a 0-10 V.	48
Figura 1.28: Esquema gráfico del banco didáctico.	49
Figura 1.29: Esquema real del banco didáctico.	49
Figura 1.30: Borneras de conexión del banco didáctico.	50
Figura 2.1: Esquema de la arquitectura del sistema para el proyecto de tesis.	52

Figura 2.2: Logo de la plataforma ThingsBoard	54
Figura 2.3: Arquitectura de la plataforma ThingsBoard.	58
Figura 2.4: Agregar una nueva organización.	69
Figura 2.5: Ventana de Organizaciones.	69
Figura 2.6: Ventana de gestión de administradores de la organización.	69
Figura 2.7: Esquema jerárquico de los Alias de entidad.	72
Figura 2.8: Configuraciones de alias.	73
Figura 2.9: Ventana de configuración de alias.	73
Figura 2.10: Ingreso de nuevo alias.	73
Figura 2.11: Motor de reglas.	75
Figura 2.12: Función RPC cliente y servidor.	76
Figura 2.13: Tipos de Widgets.	77
Figura 2.14: Fuente de datos del tipo de widget últimos valores.	78
Figura 2.15: Widget digital gauge con último valor de humedad.	78
Figura 2.16: Fuente de datos del tipo de widget series de tiempo.	79
Figura 2.17: Widget “Timeseries - Flot” con valores de series temporales.	80
Figura 2.18: Dispositivo objetivo en widget tipo RPC.	80
Figura 2.19: Widgets del tipo RPC.	80
Figura 2.20: Configuraciones widget de alarma.	81
Figura 2.21: Widget tabla de alarmas.	82
Figura 2.22: Configuración de widget estático.	82
Figura 2.23: Widgets HTML card con contenido html estático.	83
Figura 2.24: Paquetes de Widgets en ThingsBoard.	83
Figura 2.25: Widgets del paquete Digital Gauges.	84
Figura 2.26: Widgets del paquete Analogue Gauges.	84
Figura 2.27: Widgets del paquete Cards.	85
Figura 2.28: Widgets del paquete Charts.	85
Figura 2.29: Widgets del paquete Alarmas.	86
Figura 2.30: Widgets del paquete Control Widgets.	86
Figura 2.31: Widget de fecha.	87
Figura 2.32: Widgets del paquete Entity Admin Widgets.	87
Figura 2.33: Widget de extensiones.	88

Figura 2.34: Widgets del paquete Input Widgets.....	88
Figura 2.35: Widgets del paquete GPIO Widgets.....	89
Figura 2.36: Paquete de widgets Maps.	89
Figura 2.37: Exportar Widget y Exportar Panel.....	90
Figura 2.38: Exportar paquete de widgets.....	90
Figura 2.39: Opciones de importación.....	91
Figura 2.40: Ventana para importar Widget en formato JSON.	91
Figura 3.1: Bloques de programación en el PLC.	96
Figura 3.2: Desactivación del acceso optimizado al bloque DB.....	99
Figura 3.3: Habilitación del acceso PUT/GET.	99
Figura 3.4: Desactivación de protección del dispositivo.	100
Figura 3.5: Adaptador WiFi a USB utilizado.....	101
Figura 3.6: Software Win 32 Disk Imager.	101
Figura 3.7: Proceso de escritura de imagen.	102
Figura 3.8: Particiones creadas en la tarjeta microSD.	102
Figura 3.9: Desbloqueo de la ranura de la tarjeta microSD.	103
Figura 3.10: Apertura de la ranura de la tarjeta microSD.	103
Figura 3.11: Colocación de la tarjeta microSD en el sitio designado.	104
Figura 3.12: Cierre y bloqueo de la ranura de la tarjeta microSD.	104
Figura 3.13 Conexión del módulo IoT.....	105
Figura 3.14: Configuración de dirección IP del computador.....	106
Figura 3.15: Conexión SSH con el módulo IoT utilizando PuTTY.....	106
Figura 3.16: Aviso de seguridad PuTTY.	107
Figura 3.17: Pantalla de inicio en PuTTY.....	107
Figura 3.18: Ventana de configuración de red WiFi.....	108
Figura 3.19: Ventana de conexión WiFi.	109
Figura 3.20: Ingreso a una red WiFi de acceso libre.....	110
Figura 3.21: Ventana de reinicio de los puertos de red.....	110
Figura 3.22: Inicio de Node-Red.....	111
Figura 3.23: Página principal Node-Red.....	112
Figura 3.24: Instalación de paquetes.....	113
Figura 3.25: Selección de menú en Node-Red.....	113

Figura 3.26: Instalación de nodo S7.....	114
Figura 3.27: Nodo S7 en Node-Red.....	114
Figura 3.28: Configuración de Nodo S7.	115
Figura 3.29: Configuración del nodo de conexión al PLC.....	115
Figura 3.30: Configuración de red del PLC.....	116
Figura 3.31: Creación de bloque de datos.....	116
Figura 3.32: Variables dentro del bloque DB.	117
Figura 3.33: Tabla de variables en Node-Red.....	118
Figura 3.34: Editores de texto para ingreso de variables en Node-Red.....	118
Figura 3.35: Modelo de introducción de variables.....	119
Figura 3.36: Modelo de introducción de variables.....	119
Figura 3.37: Variables ingresadas en Node-Red.....	122
Figura 3.38: Configuración de nodo s7.....	122
Figura 3.39: Nodo Function.	123
Figura 3.40: Nodo switch.....	124
Figura 3.41: Configuración nodo Switch.....	125
Figura 3.42: Nodo debug.....	125
Figura 3.43: Configuración nodo debug.	126
Figura 3.44: Nodo Email.....	126
Figura 3.45: Configuración del nodo Email.....	127
Figura 3.46: Configuración de correo cuenta Gmail.....	127
Figura 3.47: Conexión de nodos en el programa node-red.	128
Figura 3.48: Mensajes de prueba en la pestaña debug.	129
Figura 3.49: Correo recibido en el correo del destinatario.....	129
Figura 3.50: Nodos Dashboard.	130
Figura 3.51: Ventana IOT2000 Setup.	130
Figura 3.52: Ventana IOT2000 Setup > Software.	131
Figura 3.53: Ventana IOT2000 Setup > Software > Manage Autostart Options.....	131
Figura 4.1 Conexiones físicas del banco didáctico.	132
Figura 4.2: Flujo 1, envío de datos a la nube.	133
Figura 4.3: Inicio del flujo 2, control del banco desde la nube.....	133
Figura 4.4: Flujo 2, control del banco desde la nube, ramificaciones.....	134

Figura 4.5: Máquinas virtuales implementadas.	136
Figura 4.6: Configuración de direcciones IP externas estáticas.....	136
Figura 4.7: Configuración de puertos en el firewall de Google Cloud.	137
Figura 4.8: Ingreso a la plataforma IoT utilizando las credenciales predeterminadas...	138
Figura 4.9: Página principal ThingsBoard.	139
Figura 4.10: Ventana de paneles.	139
Figura 4.11: Panel Banco Didáctico Final.	140
Figura 4.12: Pestañas visibles en el panel del banco didáctico.....	140
Figura 4.13: Pestaña Overview.	141
Figura 4.14: Elementos de la pestaña Overview.	142
Figura 4.15: Pestaña Alarmas.	142
Figura 4.16: Elementos de la pestaña Alarmas.	143
Figura 4.17: Pestaña de Gráficos.	143
Figura 4.18: Elementos de la pestaña Gráficos.	144
Figura 4.19: Pestaña de control.....	145
Figura 4.20: Elementos de la pestaña Control.	145
Figura 4.21: Resultado de la prueba ejecutada.....	147
Figura 4.22: Tiempos de distribución de una caja en un lote.....	148
Figura 4.23: Valores totales de las cajas distribuidas desde el primer lote.....	148
Figura 4.24: Exportación de datos en formato CSV, visualizado en Excel.	150
Figura 4.25: ThingsBoard visto desde el ordenador.	151
Figura 4.26: Plataforma IoT desde un dispositivo móvil.....	151
Figura 4.27: Configuración de controlador de temperatura.	164
Figura 4.28: Configuración de controlador de nivel.	165

ÍNDICE DE TABLAS

Tabla 1.1: Tipos de mensajes MQTT.....	38
Tabla 1.2: Parámetros del mensaje CONNECT.....	38
Tabla 1.3: Parámetros del mensaje CONNACK.....	39
Tabla 1.4: Parámetros del mensaje SUBSCRIBE.....	40
Tabla 1.5: Parámetros del mensaje SUBACK.	40
Tabla 2.1: Comparativa de precios para servicios de cómputo en la nube.	57
Tabla 2.2: Características de la versión gratuita y versión de pago de la plataforma ThingsBoard.....	63
Tabla 2.3: Funciones de dispositivos para cada rol de usuario.	71
Tabla 3.1: Tabla de direccionamiento para Node-Red.....	119
Tabla 6.1: Comparación de servidores de mensajería MQTT.	161

RESUMEN

DISEÑO E IMPLEMENTACIÓN DE LA ARQUITECTURA DEL INTERNET INDUSTRIAL DE LAS COSAS PARA LA RECOLECCIÓN DE DATOS EN UN PROCESO DE MANUFACTURA

En este trabajo de titulación se desarrolla una comunicación entre un proceso industrial de manufactura simulado por un banco didáctico perteneciente a la Escuela de Ingeniería Electrónica de la Universidad del Azuay y una plataforma IoT, mediante una interfaz web que identifica gráficas de datos interactivos acerca de las métricas del proceso en complemento con sistemas y equipos del banco didáctico. Se logra visualizar datos de sensores en tiempo real y controlar remotamente a los actuadores existentes mediante la plataforma desarrollada; este trabajo aporta de manera significativa a la necesidad de la inteligencia empresarial de las fábricas del futuro.

Palabras clave: Industria 4.0, Internet de las cosas (IoT), Internet Industrial de las cosas (IIoT), proceso de manufactura

Hugo Marcelo Torres Salamea

Director de Tesis

Daniel Esteban Iturralde Piedra

Director de Escuela

Iván Felipe Velasteguí Jaramillo

Autor

David Alejandro Fernández de Córdova
Bernal

Autor

ABSTRACT

DESIGN AND IMPLEMENTATION OF INDUSTRIAL INTERNET OF THINGS ARCHITECTURE TO COLLECT DATA IN A MANUFACTURING PROCESS

This degree work details the development of a system of communication between an industrial manufacturing process, simulated by a didactic workbench, which belongs to the School of Electronic Engineering at *Universidad del Azuay*, and an IoT platform. The procedure uses a web interface that identifies interactive data graphs about the metrics of the process in complement with systems and equipment of the didactic workbench. It was evidenced the presence of sensors data in real time and remotely control existing actuators through the developed platform. This work contributes significantly to the business intelligence in the factories of the future.

Keywords: Industry 4.0, Internet of Things (IoT), Industrial Internet of Things (IIoT), manufacturing process.

Hugo Marcelo Torres
Salamea

Thesis director

Daniel Esteban Iturralde
Piedra

School Director

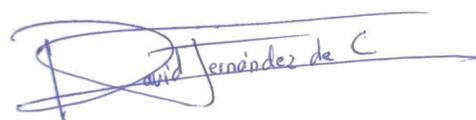
Iván Felipe Velasteguí
Jaramillo

Author

David Alejandro Fernández
de Córdoba Bernal

Author

Translated by



David Fernández de Córdoba

INTRODUCCIÓN

La industria de la producción busca optimizar los procesos para reducir costos, mejorar la calidad de sus productos y la atención al cliente. Para toda empresa, la inversión en la mejora de procesos industriales genera un gran cambio beneficioso en la economía de la misma. El control y monitoreo de procesos es esencial para mantener un trabajo ininterrumpido, sin embargo, este control requiere de personal operador que se encuentre pendiente de la maquinaria en todo momento.

En los sistemas industriales actuales existen varias máquinas y equipos de trabajo instalados en distintas cadenas de proceso, para conocer el estado de la maquinaria y tener un control del proceso es necesario monitorear los distintos parámetros que se recopilan por sensores instalados en los equipos. Los datos recopilados por los sensores de los equipos solo son datos instantáneos si no se los convierte en información, para que estos datos se puedan convertir en información se necesita almacenarlos a lo largo del tiempo y visualizarlos en interfaces gráficas entendibles.

El sistema diseñado y desarrollado en este proyecto tiene como objetivo ser utilizado como una herramienta de sustancial mejora en procesos industriales de manufactura, facilitando a la gerencia y al personal técnico de una empresa las actividades de monitorización y control. Para esto se utiliza una plataforma IoT desplegada en la nube, que genera gráficas, datos interactivos en tiempo real y reportes para el análisis de la información, los cuales se almacenan generando un histórico de datos que permite tomar decisiones empresariales en función del tiempo, esto beneficia a la empresa en el área de *Business Intelligence* (inteligencia empresarial).

Al finalizar este proyecto se proporciona un sistema estructurado de fácil implementación para el manejo de datos interactivos en tiempo real que se almacenen generando un histórico, que brinde la posibilidad de visualizar los parámetros físicos del banco didáctico del laboratorio de PLC de la Universidad del Azuay.

Durante el desarrollo de este proyecto se emplearán varias metodologías de estudio: la investigación científica y de laboratorio, las cuales brindarán la información para involucrarse con los conceptos de Internet Industrial de las Cosas e Industria 4.0, además de llevar a cabo los objetivos planteados. Adicionalmente con la investigación formativa se buscará aplicar el conocimiento adquirido para generar una guía de prácticas, útil como herramienta de estudio.

El proyecto se organiza en 5 capítulos, explicados a continuación:

En el primer capítulo se presenta el tema en base a la problemática y motivación que lo sustentan, además se tratan las bases teóricas que corresponden al concepto de IIoT e Industria 4.0, importancia, funcionalidad y aplicaciones, por otra parte, se incluye brevemente el estado del arte, con el fin de proporcionar al lector una idea de la situación en la que se encuentran los procesos industriales actualmente. Como parte final, dentro de este capítulo se explican los componentes como los sensores y actuadores, la funcionalidad, el proceso que cumple y la conexión esquemática del banco didáctico a utilizar.

En el segundo capítulo se describen las tecnologías que se utilizan para la arquitectura de software planteada, se describe la plataforma IoT utilizada para la representación de los datos, la administración y manejo de la misma. Se desarrollan paneles con las funcionalidades de la plataforma, en los cuales se visualizan los datos y variables del banco didáctico, que se almacenan en la nube y que posteriormente podrán ser recuperadas y monitorizadas en tiempo real.

En el tercer capítulo se expone la configuración y programación de los equipos de control. La programación del equipo PLC (Controlador Lógico Programable) Siemens S7-1200 utilizando el software de automatización TIA PORTAL; la configuración inicial del módulo Siemens IoT 2040, y el uso de Node-Red para comunicarse con el PLC Siemens S7-1200.

En el cuarto capítulo se describe la instalación de la plataforma IoT ThingsBoard en la nube, se exponen los resultados obtenidos en paneles interactivos, se evidencian gráficas y elementos de control en la interfaz de la plataforma. Se visualizan los parámetros relevantes de todo el proceso de control en el banco didáctico utilizado. Adicionalmente, se valida la interconexión entre elementos y la arquitectura del sistema, evidenciando los datos en tiempo real con pruebas de integración.

Por último, en conclusiones se contextualiza la importancia de los resultados obtenidos y como se aplicaría a un sistema de control de escala industrial, además, como se solventarían posibles problemas u obstáculos, tomando como referencia la realización de este proyecto.

CAPÍTULO 1: ANTECEDENTES DE INVESTIGACION

1.1 Terminología y Definiciones

Se describen a continuación varios términos que serán aplicados a lo largo del presente proyecto, incluyen definiciones de IoT, Industria 4.0 e IIoT.

1.1.1 Internet de las cosas

El Internet de las Cosas (IoT Internet of Things) fue propuesto por el empresario e ingeniero Kevin Ashton en 1999, y dado a conocer públicamente en 2009 en la conferencia LG CNS, acotando que, si los ordenadores pudieran hacer un seguimiento de cualquier objeto, serían capaces de tener un seguimiento detallado absoluto, teniendo el potencial de cambiar el mundo.

IoT lleva a la interconexión entre las personas y los dispositivos mediante aplicaciones, es la interconexión de los objetos del mundo físico a través de internet, los cuales están equipados con actuadores, sensores e implementos de comunicación, siendo posible encaminarlo a varios ámbitos como industria, salud, etc.. (Tavizon Salazar, Guajardo, & Laines, 2016).

Los beneficios que ofrece esta tecnología es la cantidad de datos que pueden ser generados, donde cada objeto virtual o físico podría ser conectado a una nube digital que genere actualizaciones periódicas (Tavizon Salazar, Guajardo, & Laines, 2016).

1.1.2 Industria 4.0

BITKOM (2016) & Del Val Román (2016) en sus trabajos afirman que:

El término "Industria 4.0" representa la cuarta revolución industrial y se refiere a un nuevo modelo de organización y de control de la cadena de valor a través del ciclo de vida del producto y a lo largo de los sistemas de fabricación. Los procesos de fabricación se encuentran en un proceso de transformación digital producida por el avance de las tecnologías de la información y, particularmente, de la informática y el software.

En definitiva, se trata de la aplicación a la industria del modelo IoT lo que conduce a la creación de redes de valor dinámicas, autoorganizadas, interorganizativas, que pueden optimizarse de acuerdo a una gama de criterios tales como costos, disponibilidad y consumo de recursos. Lo fundamental es la disponibilidad de toda la información relevante en tiempo real, así como la capacidad de obtener el mejor flujo de datos en todo momento.

1.1.3 Internet Industrial de las Cosas

IoT, que surgió hace algunos años, ha sido adoptada por la industria, dando como resultado lo que se conoce como el Internet Industrial de las Cosas (IIoT Industrial Internet of Things), este permite que los procesos y entidades industriales formen parte de Internet. Al restringir la definición de IIoT a la fabricación, se obtiene otro subconjunto de IoT, conocido como Industria 4.0 (J. Wan, 2016).

IIoT incorpora el aprendizaje de máquina (*machine learning*) y la tecnología de grandes volúmenes de datos (*big data*), aprovechando los datos de actuadores, sensores, comunicación M2M¹ (máquina a máquina) y las tecnologías de la automatización ya existentes en diversas configuraciones industriales (Rouse, 2017).

IIoT facilita la obtención de datos, con la suficiente precisión y coherencia que se requiera, de manera que las empresas puedan captar los puntos de ineficiencia mediante un monitoreo efectivo, ahorrando tiempo, dinero y apoyando los esfuerzos de inteligencia empresarial. En lo que respecta a la fabricación, IIoT tiene un gran potencial para el control de calidad y la eficiencia en general de toda una cadena de suministro (Rouse, 2017).

¹ Comunicación en formato de datos entre dos máquinas (MQTT Community, 2019).

1.2 Estado del arte

1.2.1 Aplicaciones e investigaciones acerca de IIoT e Industria 4.0

Dentro del proceso de investigación que se ha llevado a cabo en la realización de este proyecto de tesis se han encontrado trabajos relacionados con IIoT y la Industria 4.0. Entre las fuentes encontradas existen trabajos realizados en el Ecuador, así como también trabajos internacionales. Se pueden apreciar los resultados de las distintas investigaciones, la arquitectura implementada en su sistema y la visualización para los usuarios finales, resultados que serán contrastados con los resultados del presente proyecto.

Industria 4.0 y sus aplicaciones en la optimización de procesos y eficiencia energética

Gavira (2018), en su trabajo indica que las tecnologías más influyentes en la Industria 4.0, (ver Figura 1.1), son: IoT, robótica colaborativa, Big Data, Cloud Computing, y los sistemas de integración. Dentro de una cadena de montaje se aprovechan las ventajas de IoT con la robótica colaborativa mediante la comunicación M2M en cada uno de los procesos que se lleven a cabo. Los datos generados por toda la maquinaria dentro de un proceso de manufactura se almacenan y manejan bajo el concepto de Big Data y Cloud Computing.

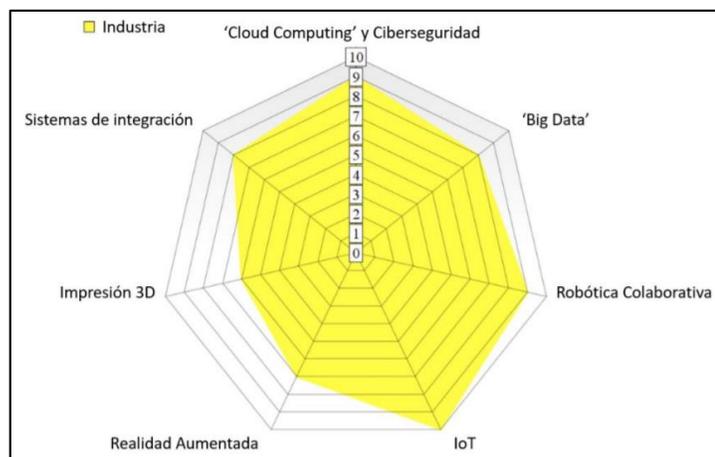


Figura 1.1: Diagrama de radar de la influencia de la Industria 4.0.

Fuente: (Gavira, 2018).

Industria 4.0 para la monitorización de un proceso industrial

López Flores (2019), desarrolló un proyecto utilizando la arquitectura IIoT para controlar y monitorizar un proceso de una planta simulado en el software Factory I/O, el cual utiliza dos tarjetas Raspberry Pi, una con un broker central instalado como servidor en la tarjeta Raspberry Pi A, y un cliente MQTT instalado en la tarjeta Raspberry Pi B.

Para el desarrollo de la plataforma se utilizaron dos ordenadores, uno para la creación y simulación de la fábrica, mediante el software Factory I/O, y otro como cliente MQTT que monitoriza el proceso desde un explorador Web. Utiliza un PLC marca Siemens para el control del proceso programado mediante TIA Portal V13, (ver Figura 1.2).

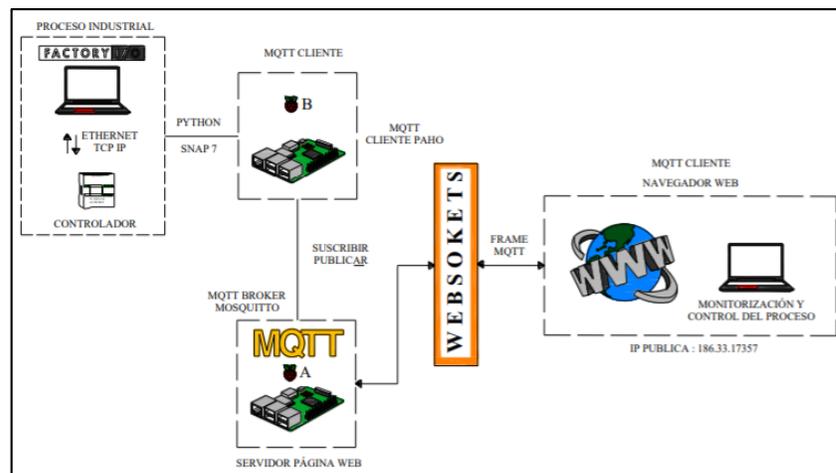


Figura 1.2: Arquitectura de una plataforma IIoT.

Fuente: (López Flores, 2019)

Posterior a la implementación de todos los recursos utilizados en base a la arquitectura planteada, se evidencian los resultados. En la Figura 1.3 se muestra el proceso industrial simulado en el software Factory I/O, mientras que, en la Figura 1.4 se muestran los paneles de control del usuario Operador y del usuario Ingeniero, a los cuales se accede utilizando el navegador web.

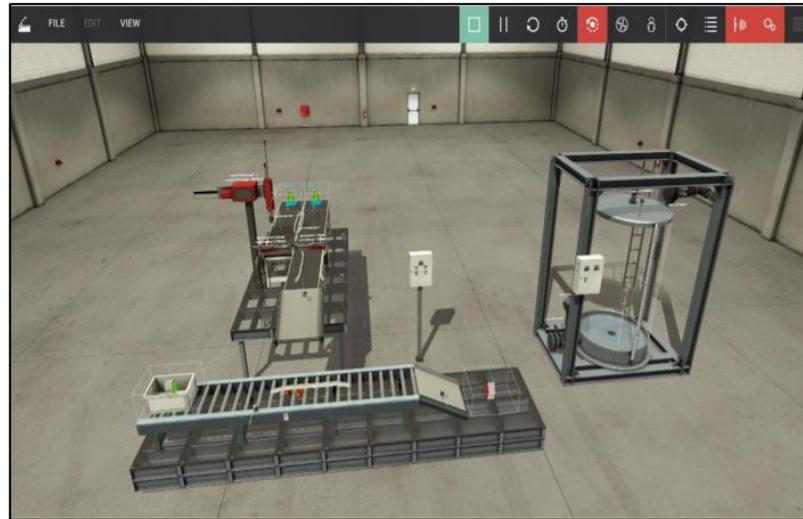


Figura 1.3: Layout de la planta en Factory I/O.

Fuente: (López Flores, 2019)



Figura 1.4: Panel de control del usuario Operador y usuario Ingeniero.

Fuente: (López Flores, 2019)

Desarrollo e implementación de internet industrial de las cosas aplicado al laboratorio de PLC's de la facultad de ingeniería industrial en procesos de automatización de la universidad técnica de Ambato

Gallo Castillo (2018), desarrolló dentro de la Universidad Técnica de Ambato una aplicación de IIoT para el laboratorio de PLC's. Mediante la implementación de software para un dispositivo Raspberry Pi se logró la comunicación con un PLC de marca Siemens, Modelo S7-1200, los datos generados por el controlador son almacenados en una base de datos y mostrados en un portal web para la visualización de la información extraída.

Para realizar la comunicación entre la tarjeta Raspberry y el PLC se usó el software Python, implementado la librería SNAP7 que permite la adquisición de los datos que son generados por el PLC. Se utilizó la biblioteca PAHO, como cliente MQTT, mediante el protocolo de mensajería MQTT los datos son enviados hacia la base de datos desarrollada en PhpMyAdmin y estos son visualizados en una página web desarrollada con el lenguaje de programación PHP. La arquitectura de la plataforma se observa en la Figura 1.5 (Gallo Castillo, 2018).

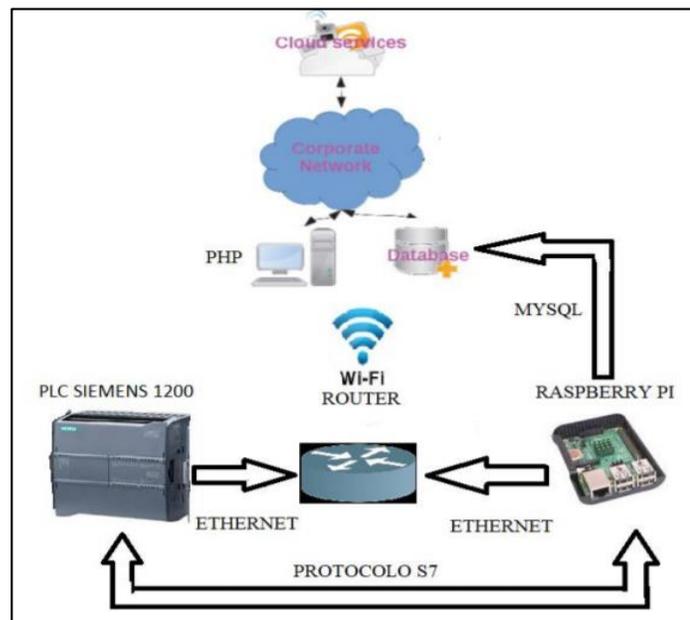


Figura 1.5: Diagrama de bloques de la aplicación.

Fuente: (Gallo Castillo, 2018)

Se utilizó el broker MOSQUITTO, siendo un broker de código abierto es totalmente compatible con el resto de componentes utilizados en esta plataforma. Para la administración de la base de datos de la plataforma se utilizó PhpMyAdmin, programa que tiene una interfaz web de fácil manejo (Gallo Castillo, 2018). El resultado del acoplamiento de todos los recursos utilizados se muestra en la Figura 1.6.

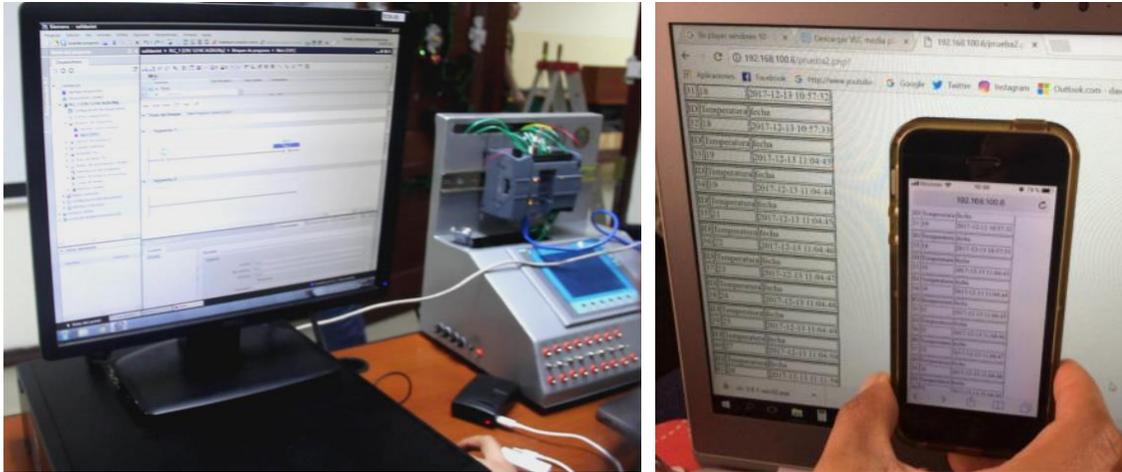


Figura 1.6: Implementación física del proyecto y funcionamiento de la página web.

Fuente: (Gallo Castillo, 2018)

Propuesta de una arquitectura de la Industria 4.0 en la cadena de suministro desde la perspectiva de la ingeniería industrial

Blanco Rojas, Gonzáles Rojas & Rodríguez Molano (2017), desarrollaron dentro de la Universidad Distrital Francisco José de Caldas, en Bogotá, una arquitectura con base en la Industria 4.0 desde el punto de vista de la ingeniería industrial, la cual se basa en la complementación de cuatro pilares principales, los cuales son: interconexión, la transparencia de la información, asistencia técnica y decisiones descentralizadas.

Con estos pilares se busca que los procesos de fabricación sean inteligentes, o en otras palabras, auto optimizados para la generación de beneficios económicos para la empresa. En la Figura 1.7 se muestra la interacción de los actores en Industria 4.0, mientras que, en la Figura 1.8 se muestra la arquitectura propuesta en su proyecto.

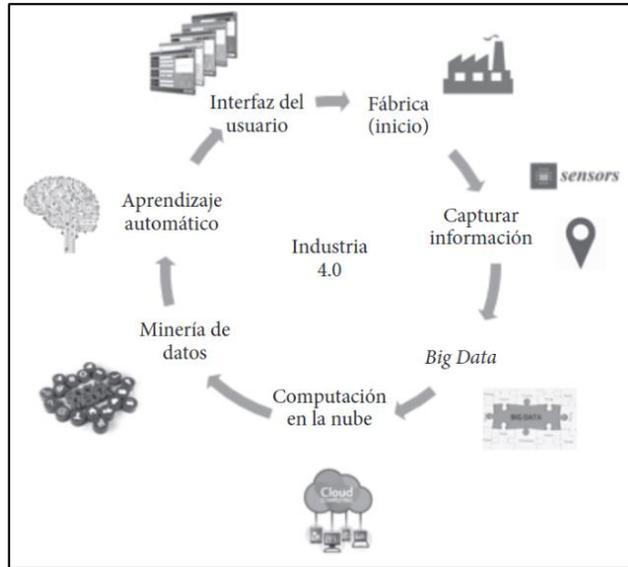


Figura 1.7: Ciclo de interacción de actores y elementos de la arquitectura propuesta.
 Fuente: (Blanco Rojas, González Rojas, & Rodríguez Molano, 2017)

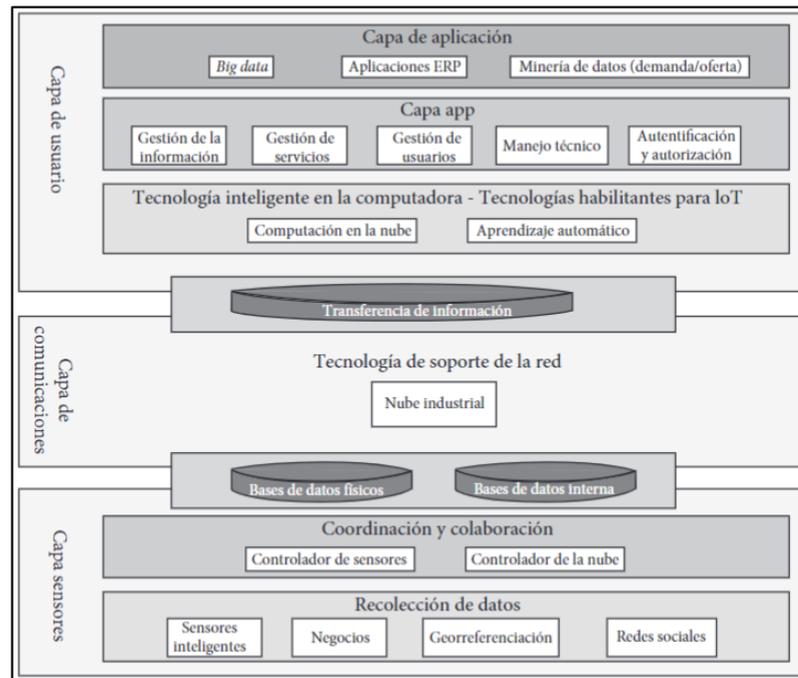


Figura 1.8: Arquitectura propuesta para Industria 4.0.
 Fuente: (Blanco Rojas, González Rojas, & Rodríguez Molano, 2017)

La arquitectura de la Figura 1.8 se divide en los siguientes componentes:

- **Capa de sensores**

Es la capa que se encarga de la recolección de datos en la cadena de suministro inteligente, contiene la subdivisión de recolección de datos, conformada por los sensores de las maquinarias, georreferenciación y las redes sociales como minería de datos. Por otra parte, en la subdivisión de coordinación y colaboración se encuentra el controlador de la nube que se encarga del envío de los datos de forma automática, y controlador de sensores que almacena los programas de control de sensores y actuadores (Blanco Rojas, González Rojas, & Rodríguez Molano, 2017).

- **Capa de comunicaciones**

En esta capa se encuentra la infraestructura que contiene la nube industrial, adicionalmente es la capa de integración entre los datos generados y las aplicaciones de visualización externas. Es necesario poder tener un acceso a internet desde esta capa (Blanco Rojas, González Rojas, & Rodríguez Molano, 2017).

- **Capa de usuario**

En esta capa se facilita la visualización de diversos puntos de la línea de producción como materias primas, falla de maquinaria o equipos, control de calidad, entre otras, mediante computación en la nube referente a contratación de servicios y aprendizaje automático. En la subcapa de aplicación se desarrolla una app que mediante autenticación de usuario y contraseña autoriza el ingreso a los datos generados (Blanco Rojas, González Rojas, & Rodríguez Molano, 2017). Las pantallas de la aplicación creada se pueden apreciar en la Figura 1.9.



Figura 1.9: Interfaz de usuario basada en la arquitectura planteada.

Fuente: (Blanco Rojas, González Rojas, & Rodríguez Molano, 2017).

1.2.2 Fábricas que utilizan el concepto Industria 4.0

Empresas de vehículos como Ford, que utilizan robots colaborativos en sus líneas de montaje (CENTER), y Maserati que ocupa soluciones digitales de Siemens para mejorar los procesos de producción y su automatización (SIEMENS), ya cuentan con implementaciones de Industria 4.0 en sus fábricas.

La plataforma avanzada de EcoStruxure, de Schneider Electric, conecta sus capas tecnológicas a través de una columna vertebral flexible en la nube, permitiendo implementar soluciones IoT de manera transparente, rentable y a escala. De esta manera, EcoStruxure acelera la convergencia entre las tecnologías de la información (TI) y las tecnologías operativas (OT) (Interempresas).

1.3 Protocolo MQTT

MQTT (Message Queue Telemetry Transport) se ha convertido en uno de los principales protocolos de comunicación M2M en las aplicaciones IoT por su ligereza y sencillez. Ambas características son condicionantes importantes para los dispositivos IoT, que a menudo tienen limitaciones en aspectos como potencia, consumo, y ancho de banda (Llamas, 2019).

El protocolo MQTT fue desarrollado por el Dr. Andy Stanford-Clark de IBM y Arlen Nipper de Arcom en el año 1999 como un mecanismo de conexión para dispositivos utilizados en la industria del petróleo. Inicialmente era un formato no abierto, en 2010 fue liberado y en 2014 pasó a ser un estándar según la OASIS (Organization for the Advancement of Structured Information Standards) (Llamas, 2019).

1.3.1 Funcionamiento del protocolo MQTT

El funcionamiento de MQTT se basa en un servicio de mensajería publicador/suscriptor (ver Figura 1.10), se utiliza un broker, servidor central en donde todos los clientes se conectan y los mensajes que son enviados se distribuyen en temas o “*topics*” organizados, de esta forma un cliente puede publicar un mensaje en un *topic* específico y el broker hará llegar este mensaje al resto de clientes suscritos a este *topic* (Llamas, 2019). En la sección de Anexos como Anexo 1: Brokers MQTT se encuentra una tabla comparativa de los servidores Brokers MQTT disponibles en el mercado.

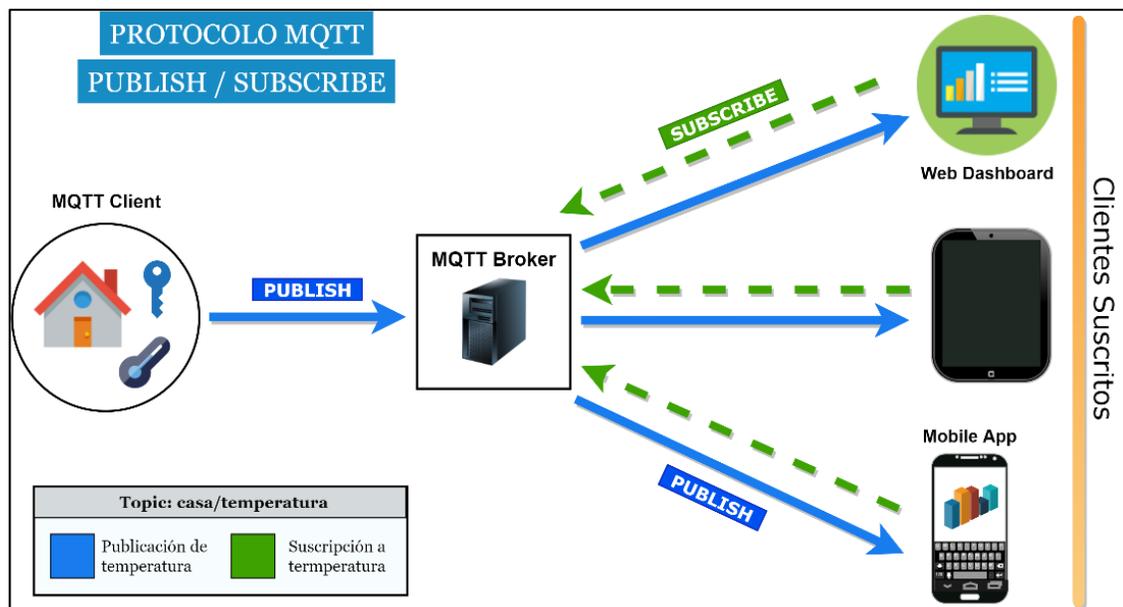


Figura 1.10: Protocolo MQTT, modelo publicador/suscriptor.

Fuente: Los autores

Para que un cliente se conecte con un broker se inicia una conexión TCP/IP, la conexión establecida se mantiene abierta hasta que el cliente la finalice. Por defecto el protocolo utiliza el puerto 1883, y el puerto 8883 cuando funciona con conexión TLS² para mensajes que requieran mayor seguridad (Yuan, 2017).

La estructura de un mensaje MQTT es muy sencilla y es en donde radica su fortaleza, según la Figura 1.11 se compone de:

- Cabecera fija: ocupa de 2 a 5 bytes, siempre es obligatorio y dentro de este encabezado se encuentra un código de control como identificación del mensaje enviado.
- Cabecera opcional: contiene información adicional que es necesaria para ciertos mensajes de diferente carácter.
- Payload (contenido): es el contenido real del mensaje, contiene un máximo de 256 Mb.

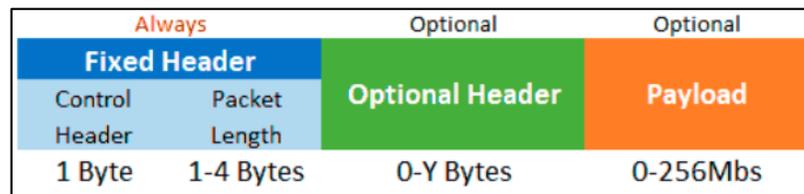


Figura 1.11: Estructura de un mensaje MQTT

Fuente: (Llamas, 2019)

1.3.2 Seguridad en el protocolo MQTT

El protocolo utiliza varias medias de seguridad que pueden adoptar las comunicaciones, incluye:

- Transporte SSL/TLS
- Autenticación por usuario y contraseña

² El protocolo TLS (Transport Layer Security) funciona utilizando cifrado que protege la transferencia de datos e información.

1.3.3 Calidad de servicio (QoS) en los mensajes MQTT

El protocolo consta de un sistema de calidad de servicio en los mensajes, el cual tiene tres niveles (MQTT Community, 2019):

- QoS 0 (unacknowledged, at most one): no confiable, se envía el mensaje una sola vez, el mensaje no se entrega en caso de falla. Si el cliente no está conectado pierde el mensaje
- QoS 1 (acknowledged, at least one): se envía el mensaje hasta que se garantiza la entrega, el mensaje se duplica en caso de fallas.
- QoS 2 (assured, exactly one): se envía el mensaje y se garantiza que llegue al suscriptor una sola vez.

1.3.4 Estructura de los topics en los mensajes MQTT

Suscripción y publicación

Cuando un cliente se suscribe a un *topic*, este permanece a la escucha de mensajes que se publiquen en ese *topic*. El MQTT Broker se encarga de enviar todos los mensajes que se publiquen en un *topic* a todos los clientes que se suscriben al mismo.

En comparación con otros protocolos que manejan cola de mensajes, los *topics* MQTT son muy livianos. El cliente no necesita crear el *topic* deseado antes que se publique o que otros clientes se suscriban al mismo. El MQTT Broker acepta cada *topic* válido sin ninguna prioridad de inicialización (HiveMQ, 2019). Un *topic* se estructura en forma de niveles, (ver Figura 1.12).



Figura 1.12: Separación de niveles de topics.

Fuente: (HiveMQ, 2019)

Algunos ejemplos de *topics*:

- casa/planta_baja/cocina/temperatura
- USA/California/San Francisco/Sillicon Valley
- 7c27e707-7cb3-4258-9e5d-7cc1fbf3203c/status
- Alemania/Bavaria/car/23156498123156/latitude
- fabrica/proceso 1/robot/posicion_x

Wildcards

Cuando un cliente se suscribe a un *topic*, este puede hacerlo en un *topic* específico o puede usar wildcards para suscribirse a múltiples *topics* simultáneamente.

Importante: Un wildcard solo se puede usar para suscribirse a *topics*, no para publicar un mensaje.

Existen dos tipos diferentes de wildcards: nivel único y multinivel

Nivel único: Como el nombre lo indica, el wildcard (+) reemplaza un único nivel de *topic*. El símbolo de suma representa un “comodín de único nivel” en un *topic*. Un ejemplo del uso del wildcard (+) se muestra en la Figura 1.13.

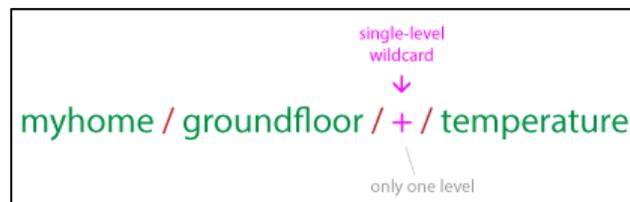


Figura 1.13: Utilización de wildcard +.

Fuente: (HiveMQ, 2019)

El nivel en donde se ubique el wildcard coincide si contiene una cadena arbitraria en lugar del comodín. Por ejemplo, una suscripción a **myhome/groundfloor/+/temperature** puede producir los siguientes resultados, (ver Figura 1.14):



Figura 1.14: Resultado de búsqueda mediante Wildcard +.

Fuente: (HiveMQ, 2019)

Multinivel: El wildcard (#) cubre muchos niveles de *topic*. El símbolo numeral representa un “comodín multinivel” en un *topic*. Para que el Broker MQTT determine qué *topics* coinciden, el comodín multinivel siempre debe colocarse como el último nivel, siendo el último carácter del *topic* y debe ir precedido de una barra diagonal, como se aprecia en la Figura 1.15.



Figura 1.15: Utilización de Wildcard #.

Fuente: (HiveMQ, 2019)

Cuando un cliente se suscribe a un *topic* con un wildcard multinivel, este recibe todos los mensajes de los *topics* que empiezan con el patrón antes del carácter (#). No importa que tan extenso sea el *topic*. Si el wildcard se especifica como un *topic* “#”, el cliente recibe todos los mensajes que se envían al MQTT Broker en cualquier *topic*. No se recomienda utilizar una suscripción al *topic* “#” debido a la gran ingesta de información (HiveMQ, 2019).

1.3.5 Sugerencias en la creación de topics, según HiveMQ:

Cada *topic* debe contener al menos 1 carácter y pueden agregarse espacios en blanco. Los *topics* diferencian entre mayúsculas y minúsculas, por ejemplo: **_micasa/temperatura** y **_MiCasa/temperatura** son dos *topics* distintos. El *topic* “/” es válido (HiveMQ, 2019).

(HiveMQ, 2019) plantea las siguientes sugerencias en la creación de topics:

a) Nunca utilizar un slash por delante del *topic*

Un slash como primer carácter es permitido en MQTT. Por ejemplo, **/casa/planta_baja/sala**. Sin embargo, el slash como primer carácter introduce un nivel de *topic* innecesario con un carácter vacío al frente. El carácter vacío como primer nivel de *topic* no provee ningún beneficio y usualmente genera confusión

b) No utilizar espacios en un *topic*

Los espacios en un *topic* generan mayor dificultad al leer o realizar debug de los *topics*. Esto debido a que UTF-8 cuenta con varios tipos de espacios en blanco, por lo que caracteres que no son comunes o estándares para todos los teclados deberían excluirse en los *topics*.

c) Crear *topics* cortos y concisos

Cada *topic* se incluye en cada mensaje enviado. Cuando se trata de dispositivos pequeños, cada byte cuenta y el tamaño del *topic* tiene gran impacto.

d) Utilizar solo caracteres ASCII

Esto debido a que los caracteres que no pertenecen a ASCII UTF-8 comúnmente no se imprimen correctamente. Es muy complicado encontrar errores de escritura o problemas cuando se trata de caracteres. A menos que sea estrictamente necesario, se recomienda evitar utilizar caracteres que no pertenecen al código ASCII en un *topic*.

e) Incluir un identificador único del dispositivo en el *topic*

Puede ser muy útil incluir un identificador único del cliente que publica los mensajes en el *topic*. El identificador único en el *topic* ayuda a identificar quien envió el mensaje. Este también puede ser utilizado para fortalecer los sistemas de autorización. Con esto se pueden crear reglas para que solo el mismo cliente pueda publicar mensajes en los *topics* con su ID. Por ejemplo, un cliente con el **id _client1**, solo puede publicar en el *topic* **_client1/status**, pero no está permitido a publicar en el *topic* **_client2/status**.

f) Nunca suscribirse al *topic* “#”

Si un cliente MQTT se suscribe al *topic* “#”, este recibirá todos los mensajes de todos los *topics* que llegan al Broker MQTT. Es por esto que no se recomienda suscribirse a todos los *topics*, debido a la gran carga de información que recibiría el cliente MQTT.

g) Generar estructuras de nivel en los *topics* que sean escalables

Si se requiere agregar nuevos dispositivos IoT a futuro, los *topics* deben basarse en una estructura flexible. Por ejemplo, si en una casa inteligente se agregan nuevos sensores, debe ser posible agregar *topics* para los mismos sin cambiar toda la jerarquía de niveles de *topic*.

h) Mejor utilizar *topics* específicos que generales

Cuando se nombren a los *topics*, deben ser los más específicos como sea posible. Por ejemplo, si se tienen tres sensores en una sala de estar, los *topics* podrían ser: **casa/sala/temperatura**, **casa/sala/brillo** y **casa/sala/humedad**. No enviar todos los valores de los sensores sobre un solo *topic* **casa/sala**. El uso de un solo *topic* para todos los mensajes es un patrón contraproducente. La nomenclatura específica también permite utilizar otras funciones MQTT, como en retained messages (mensajes retenidos).

1.3.6 Ejemplo de mensajería MQTT

Dentro de la comunicación entre un cliente y un broker, el protocolo MQTT maneja varios comandos, véase Tabla 1.1, el cliente envía estos comandos como solicitudes al broker y este responde de acuerdo a la solicitud.

Tabla 1.1: Tipos de mensajes MQTT.

MENSAJE	CODIGO	MENSAJE	CODIGO
CONNECT	0X10	SUSCRIBE	0X80
CONNAC	0X20	SUBACK	0X90
PUBLISH	0X30	UNSUBSCRIBE	0XA0
PUBACK	0X40	UNSUBACK	0XB0
PUBREC	0X50	PINGREQ	0XC0
PUBREL	0X60	PINGRESP	0XD0
PUBCOMP	0X70	DISCONNECT	0XE0

Fuente: (Llamas, 2019)

Un ejemplo de mensajería MQTT entre un suscriptor y el broker sería el siguiente:

Un cliente se conecta al broker enviando un mensaje CONNECT pidiendo una conexión entre ambos, los parámetros del mensaje CONNECT se muestran en la Tabla 1.2 (Yuan, 2017):

Tabla 1.2: Parámetros del mensaje CONNECT.

PARAMETRO	DESCRIPCIÓN
cleanSession	Indica si la conexión es persistente o no. Una sesión persistente almacena todas las suscripciones y los mensajes posiblemente perdidos en el broker.
Nombre de usuario	Las credenciales de autenticación y autorización del broker
contraseña	Las credenciales de autenticación y autorización del broker

lastWillTopic	Cuando la conexión se cae inesperadamente, el broker automáticamente publicará en un <i>topic</i> un mensaje de “último deseo”.
lastWillQoS	QoS (calidad de servicio) del mensaje “último deseo”
lastWillMessage	El cuerpo del mensaje (payload) de “último deseo”
keepAlive	Este es el intervalo en el que el cliente necesita realizar un ping en el broker para mantener activa la conexión.

Fuente: (MQTT Community, 2019)

El broker responderá un con mensaje CONNACK hacia el suscriptor, los parámetros del mensaje CONNACK se muestran en la Tabla 1.3; **Error! No se encuentra el origen de la referencia.** (Yuan, 2017):

Tabla 1.3: Parámetros del mensaje CONNACK.

PARAMETRO	DESCRIPCIÓN
sessionPresent	Indica si la conexión ya contiene una sesión persistente, es decir la conexión ya contiene <i>topics</i> suscritos y recibirá la entrega de mensajes ausentes.
returnCode	0 indica éxito, otros valores identifican una causa de falla.

Fuente: (MQTT Community, 2019)

Una vez establecida la conexión entre el suscriptor y el broker el cliente puede enviar un mensaje SUSBCRIBE para recibir diferentes mensajes de determinados *topics* del broker, los parámetros del mensaje SUSBCRIBE se muestran en la Tabla 1.4 (Yuan, 2017):

Tabla 1.4: Parámetros del mensaje SUBSCRIBE.

PARAMETRO	DESCRIPCIÓN
QoS	Indica con que calidad de servicio necesitan ser entregados los mensajes a los clientes.
tema	Las credenciales de autenticación y autorización del broker

Fuente: (MQTT Community, 2019)

Una vez el cliente se haya suscrito al broker, este responde con un mensaje SUBACK, los parámetros de este mensaje se muestran en la Tabla 1.5 (Yuan, 2017). Eventualmente, cuando un cliente desee cancelar la suscripción utilizara el mensaje UNSUBSCRIBE.

Tabla 1.5: Parámetros del mensaje SUBACK.

PARAMETRO	DESCRIPCIÓN
returnCode	<p>Existe un código de retorno para cada uno de los <i>topics</i> del comando SUBSCRIBE.</p> <ul style="list-style-type: none"> • Valores de 0 a 2: éxito, con nivel de QoS correspondiente. • Valor 128: falla.

Fuente: (MQTT Community, 2019)

1.4 Hardware

1.4.1 PLC Siemens S7-1200 1212C



Figura 1.16: PLC Siemens S7-1200 1212 AC/DC/RLY.

Fuente: (SIEMENS, 1996-2020)

El PLC que se utiliza para el desarrollo de esta tesis (ver Figura 1.16), pertenece a la marca Siemens, modelo S7-1200 1212C AC/DC/RLY (modelo en catálogo 6ES72121BE400XB0), el cual tiene las siguientes características principales (SIEMENS, 1996-2020):

- CPU compacta AC/DC/RLY.
- 8 entradas digitales 24 VDC.
- 6 salidas digitales relé 2 A.
- 2 entradas analógicas 0-10V.
- Alimentación AC 85 – 264 VAC.
- Memoria interna de 75 kb.

- Comunicación PROFINET³.
- Protocolo de comunicación TCP/IP.
- Lenguaje de comunicación KOP, FUP, SCL.

El PLC se programa en el software TIA portal, de Siemens, este modelo de PLC consta de una ranura para la colocación de un módulo de expansión de entradas y salidas.

1.4.2 Módulo de expansión para PLC S7-1200



Figura 1.17: Modulo de expansión utilizado en el banco didáctico.

Fuente: (SIEMENS, 1996-2020)

El módulo de expansión de la Figura 1.17 (modelo en catálogo: 6ES72233BD300XB0) permite agrandar el número de entradas ya sean digitales o analógicas, este modelo tiene las siguientes características principales (SIEMENS, 1996-2020):

- Alimentación 24 VDC.
- 2 entradas digitales.
- 2 salidas digitales.

³ PROFINET es el estándar de Ethernet para la automatización, permitiendo un intercambio de datos seguro y de alta velocidad entre dispositivos de campo de diferentes fabricantes (PROFINET, 2014).

1.4.3 Módulo Siemens IoT 2040



Figura 1.18: Módulo IoT 2040 utilizado en el banco didáctico.

Fuente: (SIEMENS, 1996-2020)

Un módulo IoT es un dispositivo electrónico que se puede integrar en objetos y máquinas brindando la capacidad de conectarse a redes alámbricas y/o inalámbricas para transportar datos bidireccionalmente a cualquier otro dispositivo inteligente que se encuentre conectado a internet, estos datos se pueden utilizar para fines de visualización o como órdenes para sistemas de control. Posee las capacidades y funcionalidades de los protocolos de comunicación utilizados en IoT y proporciona conectividad siempre activa, esto se debe a que las aplicaciones de IoT necesitan enviar datos automáticamente, en tiempo real, sin que alguien presione un botón de envío.

El módulo Siemens IoT 2040 de la Figura 1.18 (modelo en catálogo: 6ES76470AA001YA2) posee las siguientes características principales (SIEMENS, 1996-2020):

- Fuente de alimentación industrial de 24 VDC.
- 2 puertos Ethernet RJ45 10/100 Mbits/s.
- 1 puerto USB 2.0.
- 1 cliente USB.
- Slot para tarjeta SD.
- Procesador Intel Quark x1020.
- Memoria DDR3 SDRAM 1GB.

El módulo Siemens IoT 2040 utiliza el sistema operativo Yocto Linux. Se utiliza este sistema por ser un proyecto de colaboración de código abierto, este ayuda a desarrolladores a crear sus propios sistemas basados en Linux, personalizar múltiples plataformas, independientemente de cómo se presente el hardware de la aplicación. (YOCTO PROJECT, 2018)

En el módulo IoT se pueden instalar diferentes programas y paquetes disponibles para la imagen de Linux creada con el proyecto YOCTO, la imagen se encuentra desarrollada por Siemens. Se instala el programa Node-Red, del cual se hablará más adelante en la sección 3.3.2 y que se utilizará para esta implementación.

1.5 Descripción del banco didáctico a utilizar

El banco didáctico, ver Figura 1.19, es proporcionado por el laboratorio de control de la Escuela de Ingeniería Electrónica, el mismo cuenta con varios elementos entre los cuales se destacan sensores: inductivo, capacitivo, ultrasónico, óptico reflectivo y tres sensores magnéticos; así como también actuadores: dos bombas de agua, una banda transportadora, un robot y un horno; y elementos estáticos: tanque y reservorio (Patiño Solís & Patiño Calle, 2015).



Figura 1.19: Banco didáctico a utilizar.

Fuente: (Patiño Solís & Patiño Calle, 2015)

Este banco de pruebas resulta ideal para la práctica de tecnologías IIoT aplicadas en procesos industriales, ya que cuenta con varios sensores industriales, actuadores y dispositivos de control de alta calidad y fiabilidad en la industria.

1.5.1 Componentes del banco didáctico

En el banco didáctico se han instalado previamente componentes de tipo industrial de diversas marcas como Siemens, Festo y Sick, los mismos serán detallados a continuación (Patiño Solís & Patiño Calle, 2015):

- Sensor capacitivo SICK modelo CM30-25NPP-KW1: permite la detección de objetos a cierta proximidad, en el caso del proceso simulado (sección 0) detectará la presencia de cualquier tipo de elemento, ya sea metálico o no metálico (ver Figura 1.20).



Figura 1.20: Sensor capacitivo.

Fuente: (SICK, Sensor Intelligence, 2020)

- Sensor inductivo SICK modelo IME18-05BPSZC0S: la detección de objetos metálicos, en el caso del proceso simulado (sección 0) detectará la presencia de un elemento metálico (ver Figura 1.21).



Figura 1.21: Sensor inductivo.

Fuente: (SICK, Sensor Intelligence, 2020)

- Termocupla: La termocupla genera una diferencia de potencial por la diferencia de los materiales de dos conductores que están colocados en el terminal. En el banco didáctico se utiliza una termocupla tipo J para medir el rango de temperatura de 0 a 100 °C (ver Figura 1.22).

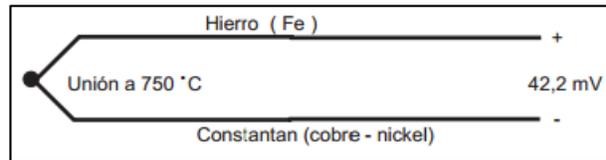


Figura 1.22: Termocupla.

Fuente: (Patiño Solís & Patiño Calle, 2015)

- Sensor óptico reflectivo SICK modelo WL170-P132: es un sensor que puede llegar a funcionar como interruptor basado en la detección de luz, envía una señal eléctrica de detección en cuanto el rayo de luz infrarrojo es bloqueado por algún objeto, en este caso, el sensor óptico enviará una señal eléctrica al PLC para informar que hay un elemento listo para la distribución al final de la banda transportadora (ver Figura 1.23).

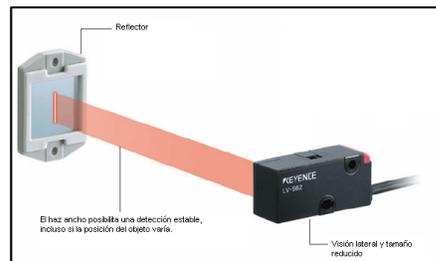


Figura 1.23: Sensor óptico reflectivo.

Fuente: (Patiño Solís & Patiño Calle, 2015)

- Sensor magnético FESTO modelo SMT/SME-8: Permite la detección de elementos metálicos, sin existir un contacto físico por lo que se conserva contra el desgaste (ver Figura 1.24).



Figura 1.24: Sensor magnético.

Fuente: (Grupo Festo, 2020)

- Sensor ultrasónico SICK modelo W30-13113: basa su funcionamiento en la reflexión de ondas acústicas en los objetos presentes en su rango, obtiene la distancia mediante el cálculo de la velocidad y el tiempo de rebote de las ondas acústicas (ver Figura 1.25).

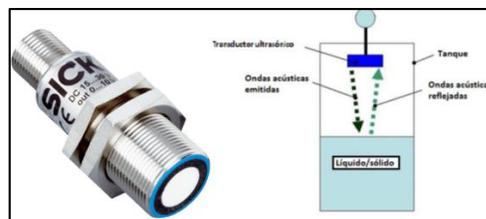


Figura 1.25: Sensor de ultrasonido.

Fuente: (SICK, Sensor Intelligence, 2020)

- Controlador PID Bielco BTC-9300:

Para el control de temperatura y de nivel, se utilizarán dos controladores PID independientes al PLC (ver Figura 1.26), los cuales poseen las siguientes características.

- Control PID y lógica difusa.
- Voltaje de alimentación: 90-264VCA.
- Sensores de ingreso: Termocuplas (tipos J, K, T, E B, R, S, N, L), variación analógica de -8mV a 70mV, -3mA a 27mA, o -1.3VCC a 11.5VCC
- Salidas: tipo relé (2A/240VCA), pulsos (5VCC).
- Alarmas: tipo relé.
- Comunicación: RS-232, RS-485, Modbus RTU.

Las configuraciones de cada controlador PID se encuentran en la sección de Anexos como Anexo 2: Configuraciones de los controladores PID externos



Figura 1.26: Controlador PID Bielco 9300.

Fuente: (Patiño Solís & Patiño Calle, 2015)

- Transmisor de temperatura Siemens SITRANS-TH200:

Modelo en catálogo Siemens: 6ES72121BE400XB0

Este transmisor convierte la señal de voltaje de la termocupla en una señal de estándar industrial 4-20mA, que puede ser fácilmente convertida a voltaje con una resistencia de carga. Se utiliza este transmisor para convertir el valor de temperatura en un rango de voltaje de 0-10V y enviarlo al PLC (ver Figura 1.27).

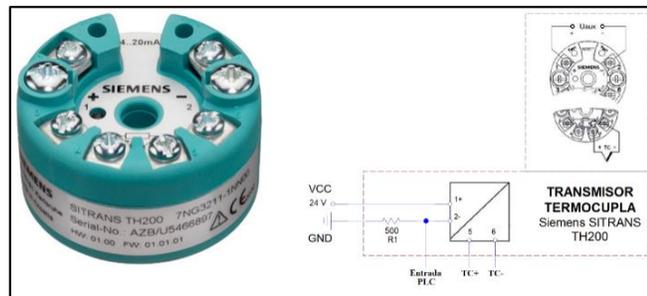


Figura 1.27: Conversor de termocupla a 0-10 V.

Fuente: (SIEMENS, 1996-2020).

1.5.2 Esquema del banco didáctico

En las Figuras Figura 1.28 y Figura 1.29 se muestra en vista superior el área de trabajo del banco didáctico utilizado en este proyecto, se evidencian los elementos que componen el proceso simulado que se explica en la siguiente sección. La estructura y conexión interna del cableado de los elementos se encuentra descrita en el trabajo de titulación de (Patiño Solís & Patiño Calle, 2015). El banco didáctico, en la parte frontal, cuenta con borneras libres para cada terminal de conexión de los distintos elementos, (ver Figura 1.30). El esquema de conexión de estos terminales se encuentra en la sección de Anexos como Anexo 3: Diagrama de conexión del banco didáctico.

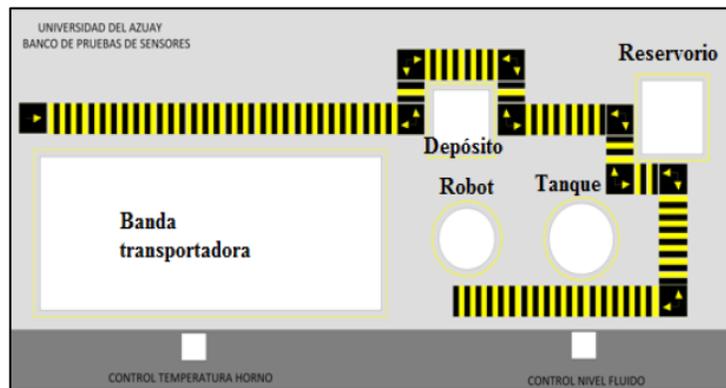


Figura 1.28: Esquema gráfico del banco didáctico.

Fuente: (Patiño Solís & Patiño Calle, 2015)



Figura 1.29: Esquema real del banco didáctico.

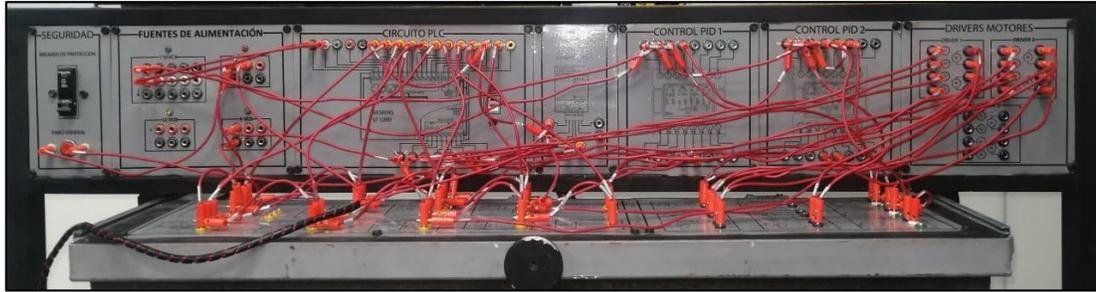


Figura 1.30: Borneras de conexión del banco didáctico.

1.5.3 Proceso industrial simulado en el banco didáctico

El banco didáctico simula un proceso industrial para la distribución de elementos metálicos o no metálicos en lugares separados, este es controlador por el PLC Siemens S7-1200 y establece conexión con la nube mediante la implementación del módulo Siemens IoT 2040. Cuenta con un proceso de calentamiento regulado por un controlador PID industrial, distribución de los elementos con un robot tipo pinza y un control de nivel de líquido con dos tanques, también regulado por otro controlador PID industrial. Se describe a continuación la secuencia del proceso (Patiño Solís & Patiño Calle, 2015):

1. Ingreso de elemento metálico o no metálico:

Al ingreso de la banda transportadora se coloca un elemento metálico o no metálico, este objeto es detectado por el sensor capacitivo. Una vez el sensor capacitivo detecta el objeto, el PLC activa el movimiento de la banda transportadora, paso seguido el elemento es detectado por el sensor inductivo para determinar si el elemento corresponde a un objeto metálico o no metálico.

2. Calentamiento del elemento ingresado:

Continuando por la guía de la banda transportadora el objeto se detiene en el horno un tiempo determinado. La temperatura del horno es regulada por el controlador PID Bielco BTC-9300 que se encuentra al frente del mismo; su funcionamiento es independiente del PLC. Una vez transcurre el tiempo de calentamiento, la banda transportadora vuelve a

energizarse para trasladar el elemento hasta el fin de guía de la banda transportadora, en donde se encuentra el sensor óptico reflectivo, dicho sensor detecta la presencia del elemento y el PLC detiene la banda transportadora para que el robot actúe.

3. Distribución del elemento ingresado:

Con el elemento detenido el robot se desplaza para entregar el mismo hasta su lugar de destino. Si el elemento es metálico, el robot lo distribuirá en el tanque; si el elemento no es metálico, el robot lo distribuirá en el depósito. Una vez el robot termine esta tarea regresará a la posición inicial, junto a la banda transportadora, para continuar con la distribución de los siguientes elementos.

El nivel de líquido del tanque es controlado por otro controlador PID Bielco BTC-9300, que se encuentra al frente del mismo, el funcionamiento de este es independiente del PLC y regula el nivel mediante la acción de dos pequeñas bombas, que envían líquido del reservorio al tanque o viceversa según sea necesario.

CAPÍTULO 2: ARQUITECTURA, BASE DE DATOS Y VISUALIZACIÓN

2.1 Arquitectura del proyecto

2.1.1 Implementación del sistema

La arquitectura implementada en el presente proyecto de tesis (ver Figura 2.1), es la siguiente:

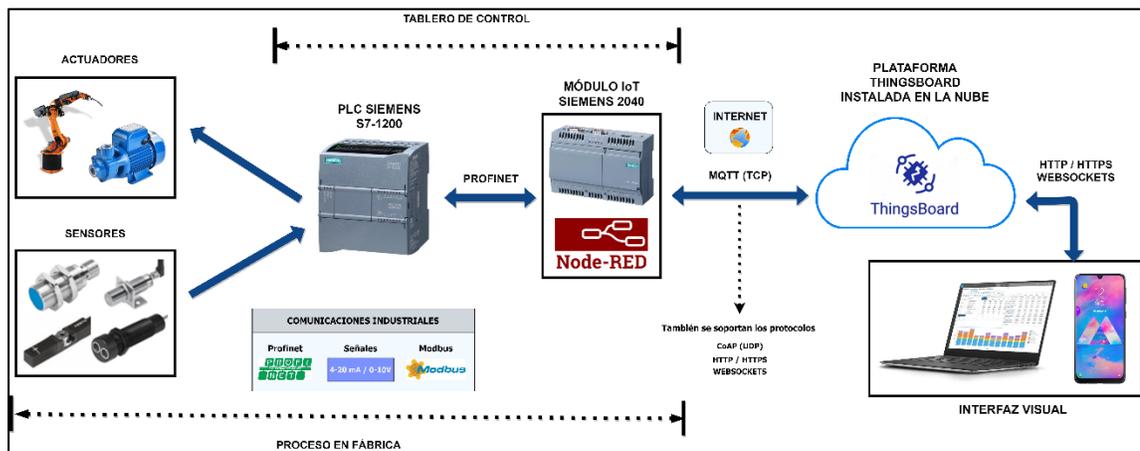


Figura 2.1: Esquema de la arquitectura del sistema para el proyecto de tesis.

Fuente: Los Autores

Se compone de 5 partes fundamentales:

- Los sensores y actuadores en el proceso industrial (banco didáctico).
- El PLC Siemens S7-1200.
- El módulo Siemens IoT 2040.
- La plataforma IoT ThingsBoard hospedada en la nube.
- Dispositivos finales con interfaz visual: computadores, smartphones.

2.1.2 Funcionamiento del sistema.

Tanto la plataforma ThingsBoard y la descripción e implementación de Node-Red serán descritas en las secciones 2.4, 3.3, respectivamente.

Los datos generados por los sensores y actuadores que se conectan en el banco didáctico son recolectados por el PLC y son almacenados en un bloque de base de datos DB, el PLC se comunica por Ethernet (PROFINET) con el módulo IoT. El módulo IoT cuenta con el sistema operativo Yocto Linux, proporcionado por Siemens, en donde se instala la interfaz visual Node-Red que brinda funcionalidades IoT extendidas.

El módulo IoT establece conexión con el PLC utilizando la programación implementada en Node-Red con el módulo `node-red-contrib-s7`, y solicita los datos en tiempo real del bloque de base de datos DB programado en el PLC. A continuación, el módulo IoT envía los valores de las variables a la plataforma IoT ThingsBoard a través de internet, utilizando el protocolo MQTT. Se utiliza el cliente MQTT de Node-Red para comunicarse con el servidor broker MQTT de la plataforma ThingsBoard. Tanto Node-Red como la plataforma ThingsBoard tienen la posibilidad de enviar y recibir datos utilizando distintos protocolos: HTTP, MQTT, entre otros.

La plataforma ThingsBoard se encuentra instalada en un servidor en la nube, que brinda muchas funcionalidades y servicios útiles para escalar la infraestructura si fuese necesario.

Finalmente, la información almacenada en el servidor se convierte en gráficas interactivas que pueden ser configuradas en paneles (*dashboards*) en la plataforma IoT ThingsBoard. ThingsBoard cuenta con una página web a la cual se accede mediante interfaces visuales como ordenadores y smartphones.

2.2 Plataforma IoT

Una plataforma IoT es el software de soporte que conecta hardware, puntos de acceso y redes de datos a otras partes de la cadena de valor (generalmente aplicaciones de usuario final). Las plataformas IoT manejan tareas de administración continua y visualización de datos, lo que permite a los usuarios automatizar su entorno. Estas plataformas son el

intermediario entre los datos recopilados y la aplicación móvil o SaaS⁴ orientada al usuario. (Ray, 2016)

Ray (2016) afirma que: Las plataformas de IoT a menudo se denominan soluciones de *middleware*⁵, una solución IoT es una combinación de funciones de varios proveedores, que incluyen:

- Sensores o controladores.
- Un dispositivo de puerta de enlace (*gateway*) para agregar y transmitir datos de ida y vuelta a la red de datos.
- Una red de comunicaciones para enviar datos.
- Software para analizar y traducir datos.
- El servicio de aplicación final (por ejemplo, una página web), que crea gran parte del valor.

Estas soluciones se pueden denominar colectivamente como la cadena de valor de IoT.

2.2.1 ThingsBoard como plataforma IoT



Figura 2.2: Logo de la plataforma ThingsBoard

Fuente: (ThingsBoard, 2020).

ThingsBoard (ver Figura 2.2), es una plataforma IoT de código abierto para recolección de información, procesamiento, visualización y manejo de dispositivos, que mantiene una

⁴ SaaS (*Software as a Service*) son servicios ofrecidos dentro de un entorno de Cloud Computing, en referencia al hardware y software de grandes centros de datos. Cloud Computing comprende los servicios que se venden desde estos centros de datos: tanto (SaaS) como utilidades (PaaS). Por tanto, SaaS ofrece aplicaciones de software que se pueden usar a través de Internet u otras redes (Schütz, Kude, & Popp, 2013).

⁵ Middleware es software que se sitúa entre un sistema operativo y las aplicaciones que se ejecutan en él. Funciona como una capa de traducción oculta para permitir la comunicación y la administración de datos en aplicaciones distribuidas. Conecta dos aplicaciones para que se puedan pasar fácilmente datos y bases de datos por una “canalización”. (Microsoft Azure, 2020)

versión gratuita y una versión de pago. Facilita un rápido desarrollo, gestión y ampliación de proyectos IoT. Permite la conexión de dispositivos a través de protocolos IoT industriales estándar como: MQTT, CoAP⁶ y HTTP. Soporta despliegue en la nube (*cloud deploy*) e implementaciones locales listas para usarse. La arquitectura del lado del servidor de ThingsBoard combina escalabilidad, tolerancia a fallos y alto rendimiento, ofreciendo fiabilidad para nunca perder la información almacenada.

Toda la información recolectada acerca de la plataforma IoT ThingsBoard y su documentación, explicada en la sección 2.7, ha sido obtenida de la fuente oficial y resumida por nosotros los autores.

Elección de la plataforma IoT

Se implementó esta arquitectura IIoT utilizando la plataforma ThingsBoard, que, a diferencia de varios servicios IoT que ofrecen distintos proveedores en la nube, como AWS IoT y Google Cloud IoT, tiene un interfaz web de fácil manejo y entendimiento. ThingsBoard incluye varios módulos y servicios que se instalan en un solo paquete en el servidor, desde los módulos para recepción de información, servidor MQTT y servidor Web, hasta los módulos que se encargan de organizar la información y ofrecer las distintas funcionalidades en el interfaz web, como la visualización y control de los datos.

Por otro lado, para implementar un sistema IoT con servicios como AWS IoT y Google Cloud IoT se necesita de un conocimiento más específico de cada módulo, que se ofrece como servicio independiente, además de que se necesita de mayor configuración para prestaciones específicas. Por estos motivos ThingsBoard cuenta con mayor facilidad y menor tiempo en la implementación de un sistema IoT, lo que facilita el estudio y práctica para el estudiante o investigador, sin dejar de lado sus extendidas funcionalidades que se explican en este capítulo.

⁶ El protocolo de aplicación restringida (CoAP - Constrained Application Protocol) es un protocolo de transferencia web especializado para usar con nodos restringidos y redes restringidas (por ejemplo, de baja potencia, con pérdida). (Shelby, Hartke, & Bormann, 2014)

Para más información de los servicios de AWS IoT y Google Cloud IoT consultar en las páginas oficiales respectivamente:

- <https://aws.amazon.com/es/>
- <https://cloud.google.com/solutions/iot>

2.3 Servidor en la nube

Se necesita de un servidor para instalar la plataforma ThingsBoard, esta plataforma cuenta con un servidor web y un broker MQTT para procesar las peticiones de los dispositivos IoT y de las interfaces visuales conectadas. En el servidor se instalará y configurará la base de datos PostgreSQL en donde se almacenarán todos los datos de telemetría de los dispositivos IoT conectados. Para abastecer las necesidades de varios cientos, miles o millones de mensajes provenientes de varios dispositivos se utilizará un servidor en la nube para aprovechar las funcionalidades de escalabilidad y otros servicios sin limitar las capacidades del sistema. La configuración del servidor e instalación de la plataforma ThingsBoard se describe en la sección 4.4.

Posterior a una investigación de servidores en la nube, se han elegido tres opciones principales, se realizó una comparación entre las mismas para poder optar por la más adecuada. Entre las empresas que brindan estos servicios se han escogido las opciones de Google Cloud, Amazon Web Services (AWS) y DigitalOcean.

Los precios varían en función de la localidad de la instancia, es distinto en cada país y región. Se muestran en la Tabla 2.1 los costos de las instancias de máquinas virtuales más comunes en la nube disponibles en la región por defecto de cada empresa, estos servicios se cobran en función del tiempo de actividad del servidor, también llamado bajo demanda, es por esto que se brindan los precios por hora para mayor facilidad en el cálculo del costo.

Tabla 2.1: Comparativa de precios para servicios de cómputo en la nube.

CPU Virtuales	Memoria (GB)			Google Cloud		Digital Ocean		AWS (Amazon)	
	Google Cloud	Digital Ocean	Amazon	Precio Mensual (\$)	Precio por hora (\$)	Precio Mensual (\$)	Precio por hora (\$)	Precio Mensual (\$)	Precio por hora (\$)
1	3,75	3	3,8	24,27	0,0475	15	0,0220	49,05	0,0670
2	7,5	8	8	48,55	0,0950	60	0,0890	55,05	0,0752
4	15	16	16	97,09	0,1900	120	0,1790	110,10	0,1504
8	30	32	32	194,18	0,3800	160	0,2380	220,19	0,3008
16	60	64	64	388,36	0,7600	320	0,4760	503,62	0,6880
32	120	64	128	776,72	1,5200	640	0,9520	1007,24	1,3760
64	240	---	256	1553,44	3,0400	---	---	2014,47	2,7520
96	360	---	384	2330,16	4,5600	---	---	3021,70	4,1280

Fuente: (Google, 2020) (Amazon Web Services, Inc. , 2020) (DigitalOcean, LLC., 2020).

Google Cloud

Para el desarrollo del proyecto de tesis se optó por elegir los servicios de la nube de Google. Al crear una cuenta, Google Cloud otorga un crédito gratuito de 300 dólares para consumo de cualquier servicio de la nube, para crear cualquier tipo de instancia que se necesite durante un período de 1 año; el usuario puede adquirir esta prueba gratuita de los servicios validando su identidad. Aunque AWS también otorga servicios gratuitos por un breve periodo de tiempo, no permite el uso de instancias de gran capacidad sin realizar el pago correspondiente. Con esta facilidad el estudiante o investigador puede crear tantos servidores como desee con distintas características si fuese necesario, para cualquier tipo de proyecto de tecnología.

2.4 Arquitectura de la plataforma ThingsBoard

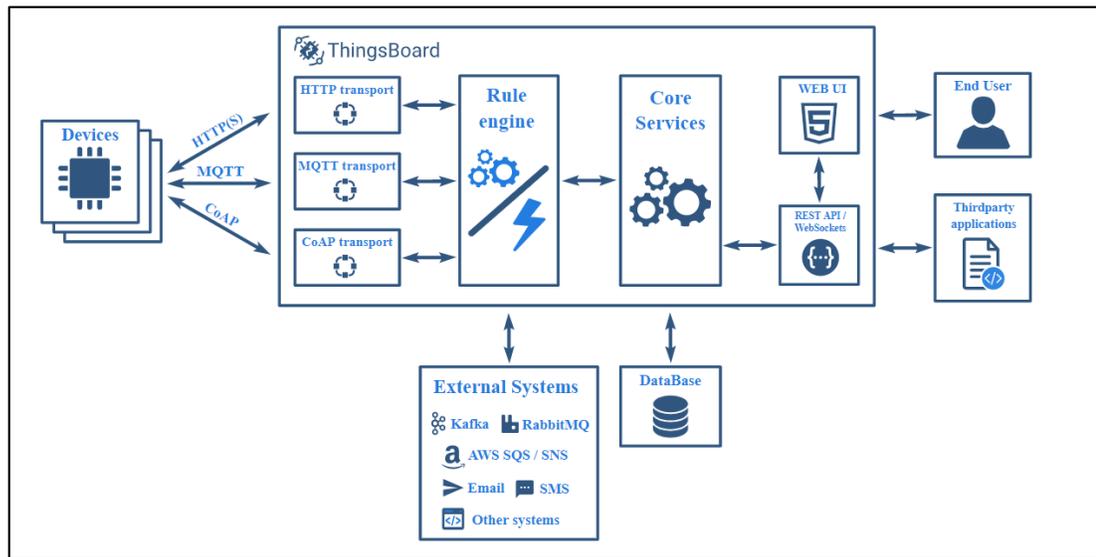


Figura 2.3: Arquitectura de la plataforma ThingsBoard.

Fuente: (ThingsBoard, 2020).

En la Figura 2.3 se menciona de izquierda a derecha: Los dispositivos IoT se pueden conectar a ThingsBoard a través de internet, utilizando protocolos de comunicación estándar como: HTTP, HTTPS, MQTT, CoAP, entre otros. ThingsBoard internamente cuenta con módulos para manejar las peticiones al servidor en los distintos puertos de los protocolos, interactuando con la capa de transporte. Posterior a esto los mensajes son enviados al motor de reglas, el cual interactúa con los servicios de la plataforma para brindar muchas funcionalidades al usuario.

De aquí en adelante la plataforma se encarga de manejar y almacenar la información, se distribuye la información en tiempo real a todos los interfaces visuales conectados, tales como: exploradores web, dispositivos móviles o aplicaciones de terceros.

La arquitectura de ThingsBoard cuenta con muchas herramientas y funcionalidades para interactuar con diversos sistemas externos, otras plataformas IoT, servidores MQTT y sistemas de análisis de datos.

2.4.1 Sistemas externos

Es posible enviar/publicar mensajes desde ThingsBoard a sistemas externos, utilizando el motor de reglas. Se pueden enviar datos al sistema externo, procesarlos y reenviarlos de vuelta a ThingsBoard para su visualización. Para mayor funcionalidad de la plataforma se puede implementar ThingsBoard Gateway que se describe en la sección 2.5.

2.4.2 Base de datos de la plataforma

Existen tres opciones disponibles, en cuanto a bases de datos, para desplegar la plataforma:

- SQL
 - PostgreSQL
- NoSQL
 - Cassandra
- Base de datos híbrida
 - PostgreSQL + Cassandra

La plataforma se puede desplegar para abastecer las necesidades de varios escenarios, conforme la capacidad de solicitudes y dispositivos, ThingsBoard admite varias opciones brindando la capacidad de elegir dónde almacenar las entidades principales y dónde almacenar los datos de telemetría. ThingsBoard cuenta con un servicio de asesoría en su página oficial para el despliegue de la plataforma en distintos escenarios.

2.4.3 Arquitectura de despliegue: microservicios o monolítico.

ThingsBoard admite la implementación monolítica para comenzar en entornos pequeños, así como también proporciona la capacidad de actualizar a microservicios para alta disponibilidad y escalabilidad horizontal.

2.4.3.1 Implementación monolítica

Un monolito o aplicación con arquitectura monolítica es una aplicación de software cuyos módulos no se pueden ejecutar de forma independiente. Esto hace que los monolitos sean difíciles de usar en sistemas distribuidos (Dragoni et al., 2017).

Una aplicación monolítica cuenta con un repositorio/código base único y extenso que ofrece decenas o cientos de servicios utilizando diferentes interfaces, como páginas HTML, servicios web y/o servicios REST (Villamizar et al., 2015).

Además, una implementación monolítica representa un “*single-point-of-failure*”⁷; si la aplicación falla, todos los servicios de la misma caen (Villamizar et al., 2015).

2.4.3.2 Implementación en microservicios

Un microservicio es un proceso coherente e independiente que interactúa a través de mensajes. Desde un punto de vista técnico, los microservicios deben ser componentes autónomos e independientes implementados conceptualmente de forma aislada de otras instancias y servicios, y equipados con herramientas dedicadas de persistencia de memoria, por ejemplo, bases de datos (Group, 2020).

Una arquitectura de microservicios es una aplicación distribuida donde todos sus módulos son microservicios. Su comportamiento distintivo se deriva de la composición y coordinación de sus componentes a través de mensajes (Dragoni et al., 2017).

ThingsBoard admite la arquitectura de microservicios (MSA)⁸, para realizar implementaciones escalables para millones de dispositivos. Esta arquitectura permite desplegar la plataforma en clúster, donde clúster o enjambre es un conjunto de servidores que trabajan conjuntamente como si fuesen uno solo. El administrador del sistema puede ajustar de manera flexible los microservicios para optimizar el clúster de acuerdo con la carga actual.

⁷ Un *single-point-of-failure* (SPOF) o único punto de falla es una parte de un sistema o cualquier elemento de red que, si falla, detendrá el funcionamiento de todo el sistema. Además, para que haya un solo punto de falla, no debe haber un sistema de respaldo. (Dooley, 2002)

⁸ MSA es un estilo de arquitectura que define y crea sistemas mediante el uso de pequeños servicios independientes y autónomos que se alinean estrechamente con las actividades comerciales. Los microservicios son los bloques de construcción arquitectónicos principales de un MSA. (Group, 2020)

2.4.4 Características de la arquitectura de ThingsBoard:

- **Escalable:** Plataforma escalable horizontalmente, construida sobre tecnologías líderes de código abierto. La cantidad de solicitudes y dispositivos del lado del servidor admitidos aumenta linealmente a medida que se agregan nuevos servidores ThingsBoard en modo de agrupación (clúster). Sin tiempo de inactividad, reinicios del servidor o errores de aplicación.
- **Tolerante a fallas:** En la arquitectura de microservicios todos los nodos ThingsBoard son idénticos. Arquitectura tolerante a fallas, inmune a “*single-point-of-failure*” (SPOF) cada nodo en el clúster es idéntico. Si algún nodo falla, esto se detecta automáticamente. Los nodos que fallen se pueden reemplazar sin tiempo de inactividad. Los datos persistentes se replican utilizando una base de datos NoSQL confiable.
- **Robusto y eficiente:** Un solo nodo de servidor puede manejar decenas o incluso cientos de miles de dispositivos dependiendo del caso de uso. El clúster de ThingsBoard puede manejar millones de dispositivos.
- **Personalizable:** Agregar una nueva funcionalidad es fácil con widgets personalizables y nodos de motor de reglas.
- **Fiable y duradero:** Nunca se perderán los datos.

2.5 Características de ThingsBoard

ThingsBoard es 100% de código abierto, tiene licencia de Apache License 2.0, por lo que se puede usar en productos comerciales de forma gratuita, incluso se puede alojar como una solución SaaS o PaaS⁹.

⁹ PaaS (*Platform as a Service*) es la provisión de una plataforma completa, es decir, hardware y software, como servicio, para dar a los proveedores de software independientes la oportunidad de desarrollar y proporcionar soluciones SaaS o integrarlas con aplicaciones de software tradicionales (Beimborn, Miletzki, & Wenzel, 2011).

ThingsBoard IoT Gateway

ThingsBoard IoT Gateway es una solución de código abierto que permite integrar dispositivos conectados a sistemas heredados y de terceros con ThingsBoard. Algunos tipos de dispositivos que se pueden conectar a ThingsBoard a través del Gateway IoT de la plataforma son:

- Sigfox
- LoRa
- ZigBee
- Bluetooth

Multi Alquiler

Admite instalaciones de múltiples inquilinos listas para usarse. Un inquilino u organización puede tener múltiples administradores inquilinos y millones de dispositivos y clientes.

Gestión de dispositivos

Proporciona la capacidad de registrar y administrar dispositivos. Permite monitorear los atributos del dispositivo del lado del cliente y del lado del servidor. Proporciona API's para aplicaciones del lado del servidor, para enviar comandos RPC (Remote Procedure Calls) a dispositivos y viceversa. Los comandos RPC se explican más a detalle en la sección 2.7.7.

Gestión de activos

Brinda la capacidad de registrar y administrar activos. Permite aprovisionar atributos de activos del lado del servidor y monitorear alarmas relacionadas con los mismos. Capacidad para construir jerarquía de entidades utilizando relaciones.

Seguridad

Admite el cifrado en la capa de transporte (TLS – Transport Layer Security) para los protocolos MQTT y HTTP. Admite autenticación de dispositivos y gestión de credenciales de dispositivos. Se explica en detalle en la sección 2.7.1.

Gestión de alarmas

Brinda la capacidad de crear y administrar alarmas relacionadas con sus entidades: dispositivos, activos, clientes, etc. Permite el monitoreo de alarmas en tiempo real y la propagación de alarmas a la jerarquía de entidades relacionadas. Se pueden generar alarmas en eventos de desconexión o inactividad del dispositivo.

2.6 Comparativa entre la versión de pago y la versión gratuita open-source

La plataforma ThingsBoard cuenta con dos versiones: la versión gratuita de código abierto (Community Edition) y la versión de pago (Professional Edition) que cuenta con funcionalidades extra, como se aprecia en la Tabla 2.2.

Tabla 2.2: Características de la versión gratuita y versión de pago de la plataforma ThingsBoard

Característica	Community Edition	Professional Edition
Atributos	SI	SI
Telemetría	SI	SI
Entidades y relaciones	SI	SI
Visualización de la información	SI	SI
Motor de reglas	SI	SI
Comandos RPC	SI	SI
Registro de auditoría	SI	SI
Arquitectura de microservicios	SI	SI
API Limits	NO	SI
White labeling	NO	SI
Platform Integrations	NO	SI

Grupos de Dispositivos y Activos	NO	SI
Planificador	NO	SI
Informes y reportes	NO	SI
Exportar información en formato CSV/XLS	NO	SI
Almacenamiento de archivos	NO	SI

Fuente: (ThingsBoard, 2020).

2.6.1 Características de ThingsBoard Community Edition

Atributos. - La plataforma tiene la habilidad de asignar atributos personalizables llave-valor (*key-value*) a las entidades (por ejemplo: datos de configuración, datos de fábrica del dispositivo, etc.).

Telemetría. - Parar recopilar y almacenar datos de telemetría de manera confiable, escalable y tolerante a fallas de red y hardware. Tiene una API para la recolección de series de datos en función del tiempo.

Entidades y relaciones. - Permite crear modelos de objetos del mundo físico (por ejemplo: dispositivos y activos) así como también relaciones entre ellos, con clientes o con cualquier otra entidad. Funcionalidad de administrar, supervisar y controlar las entidades IoT de forma segura utilizando API's enriquecidas del lado del servidor.

Visualización de datos. - Visualización de datos utilizando widgets integrados o personalizados. Creación de paneles (*dashboards*) IoT flexibles para visualización de datos y control remoto de dispositivos en tiempo real. Se incluyen más de 30 widgets personalizables para la mayoría de los casos de usos de IoT. Se pueden compartir los paneles creados con varios clientes.

Motor de reglas. - Permite crear cadenas de reglas complejas para procesar los datos de dispositivos de acuerdo a los casos de uso específicos de la aplicación. Cada cadena de

reglas puede manejar el procesamiento de los datos y generar acciones en base a telemetría entrante y eventos. Se puede incluir flujos lógicos personalizables para reenviar datos a sistemas externos o activar alarmas, las cuales pueden configurarse como cadenas de notificación. El motor de reglas cuenta con una interfaz amigable con la funcionalidad de arrastrar y soltar componentes.

Entre otras acciones del motor de reglas se puede:

- Transformar y normalizar los datos provenientes de dispositivos.
- Actualizar atributos de dispositivos o activos.
- Manejar y visualizar el estado del dispositivo, por ejemplo, si está activo o inactivo.

RPC. - API y widgets para enviar comandos RPC desde sus aplicaciones y paneles a dispositivos y viceversa. Se explican más a detalle en la sección 2.7.7.

Registro de auditoría. - Seguimiento de la actividad del usuario y el uso de llamadas API.

2.6.2 Características de ThingsBoard Professional Edition

ThingsBoard Professional Edition cuenta con las características de la versión gratuita e incluye varias características nuevas como:

API Limits. - Permite controlar el uso de la API, limitando el número de solicitudes de cada host individual durante una sola unidad de tiempo.

White-labeling – Permite configurar la plataforma con estilos personalizados, insertar el logo de la empresa y crear plantillas de correo.

Platform Integrations – Para conectar dispositivos utilizando soluciones de conectividad como NB IoT, LoRaWAN y Sigfox, manejar la información entrante e integrar a otras plataformas IoT, incluye:

- **Protocolos:** HTTP, HTTPS, MQTT
- **OPC-UA**
- **Sigfox**
- **ThingPark**
- **TheThingsNetwork**
- **Azure Event Hub**
- **IBM Watson IoT**
- **AWS IoT**
- **AWS Kinesis**

Grupos de dispositivos y activos. - Para configurar múltiples grupos de dispositivos y activos personalizados.

Planificador. - Para programar varios tipos de eventos (es decir, actualizaciones de configuración, generación de informes, comandos RPC) con opciones de configuración flexibles.

Informes y reportes. - Generar informes utilizando los paneles existentes y distribuirlos a los usuarios finales por correo electrónico. Genera archivos PDF.

Exportación de datos CSV / XLS. - Exporta datos de widgets a CSV o XLS.

Almacenamiento de archivos. - Capacidad de almacenar contenido binario (archivos) en la base de datos.

2.7 Documentación de ThingsBoard

2.7.1 Seguridad

La plataforma ThingsBoard cuenta con varias opciones de seguridad para el envío de datos y telemetría. El usuario accede por medio de distintas API's para la comunicación de los dispositivos IoT a la plataforma. Las API's disponibles se mencionan en la sección 2.8.

Opciones de autenticación del dispositivo

Las credenciales del dispositivo se utilizan para conectarse al servidor de ThingsBoard por medio de las aplicaciones que se ejecutan en el dispositivo IoT. ThingsBoard está diseñado para admitir diferentes credenciales de dispositivo. Hay dos tipos de credenciales compatibles:

Tokens de acceso: credenciales de uso general adecuadas para una amplia gama de dispositivos. La autenticación basada en tokens de acceso se puede usar en modo SSL no encriptado o unidireccional.

Ventajas: Respaldado por dispositivos con recursos limitados. Baja sobrecarga de red. Fácil de aprovisionar y usar.

Desventajas: Puede ser fácilmente interceptado mientras se usa una conexión de red sin cifrar (HTTP en lugar de HTTPS, MQTT sin TLS / SSL, etc.).

Certificados X.509: estándar PKI y TLS. La autenticación basada en certificados X.509¹⁰ se utiliza en modo SSL bidireccional.

Ventajas: Alto nivel de seguridad utilizando la conexión de red cifrada y la infraestructura de clave pública.

Desventajas: No es compatible con algunos dispositivos con recursos limitados. Afecta el uso de la batería y la CPU.

Las credenciales del dispositivo deben suministrarse a la entidad del dispositivo correspondiente en el servidor. Hay varias formas de hacer esto:

¹⁰ X.509 es un estándar UIT-T de criptografía para infraestructuras de claves públicas (Public Key Infrastructure o PKI). Los certificados X.509 se utilizan en muchos protocolos de Internet, incluidos TLS/SSL, que es la base de HTTPS (International Telecommunication Union (ITU), s.f.).

- **Automáticamente:** usando la API REST de ThingsBoard. Por ejemplo, durante la fabricación, el control de calidad o el cumplimiento de la orden de compra, se realiza una llamada al API REST de ThingsBoard y se asigna una credencial a la entidad del dispositivo.
- **Manualmente:** usando la interfaz de usuario web de ThingsBoard. Por ejemplo, para fines de desarrollo, o por el administrador del sistema al crear un dispositivo.

2.7.2 Roles de usuario

Existen 3 tipos de usuario en la plataforma: Administrador, Inquilino y Cliente.

Administrador. - Es el administrador del sistema, el rol de usuario más alto. Puede crear organizaciones y usuarios administradores de las organizaciones. Puede cambiar configuraciones generales del sistema, de correo y de seguridad.

Inquilino. – También llamado administrador de la organización, es el alquiler de la plataforma, por tanto, puede administrar los widgets, paneles, dispositivos, activos, así como también crear usuarios clientes de la organización y asignar los permisos respectivos a cada uno.

Cliente. - Este usuario puede visualizar los dispositivos y paneles a los cuales tiene acceso, este tipo de usuario no puede modificar ninguna configuración. Su rol es estrictamente visualizar los paneles con la información en los mismos.

Se describen a continuación las funciones y permisos de cada usuario en la plataforma:

Usuario Administrador del Sistema

ThingsBoard admite la multipropiedad (*Multitenancy*). La plataforma ThingsBoard instalada en un servidor puede tener a varios inquilinos, en donde cada inquilino del sistema es una entidad comercial separada: individuo u organización que posee o produce dispositivos.

El usuario administrador del sistema puede crear entidades de inquilinos (organizaciones), como se aprecia en la Figura 2.4. Para esto ingresa en la pestaña de Organizaciones y las agrega con el botón de acción.

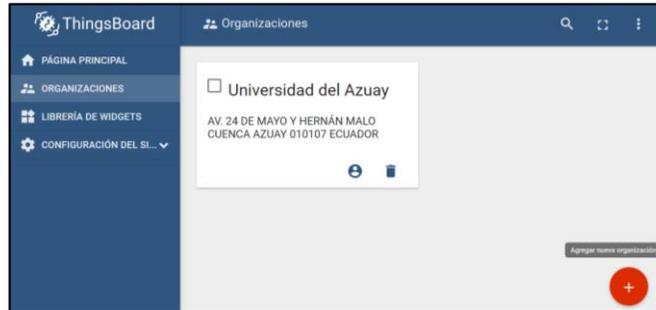


Figura 2.4: Agregar una nueva organización.

El administrador del sistema también puede crear múltiples usuarios con el rol de inquilino/administrador de la organización (*Tenant Administrator*) para cada organización, presionando el botón “Gestionar administradores de la organización” en los detalles de la organización, como se muestra en las Figuras Figura 2.5 y Figura 2.6.

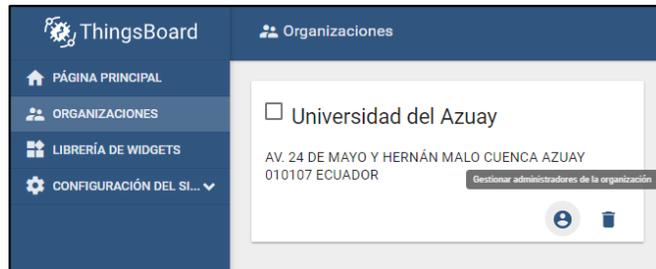


Figura 2.5: Ventana de Organizaciones.

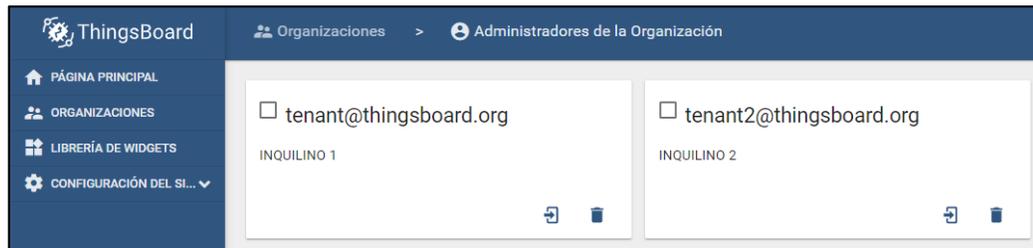


Figura 2.6: Ventana de gestión de administradores de la organización.

Usuario Inquilino

El inquilino/administrador de la organización (*Tenant Administrator*) puede realizar las siguientes acciones:

- Aprovisionar y administrar dispositivos.
- Manejar credenciales del dispositivo.
- Provisión y gestión de activos.
- Crear y gestionar clientes.
- Crear y administrar paneles.
- Configurar motor de reglas.
- Agregar o modificar widgets predeterminados usando la Biblioteca de widgets.

Usuario Cliente

El usuario cliente forma parte de la organización, puede realizar las siguientes acciones:

- Visualizar atributos de sus dispositivos asignados.
- Visualizar paneles asignados.
- Visualizar atributos de activos asignados.

2.7.3 Activos

Para las empresas, los activos son cosas de valor que sostienen la producción y el crecimiento. Para una empresa, los activos pueden incluir máquinas, propiedades, materias primas e inventario, así como intangibles como patentes, regalías y otra propiedad intelectual. (Arthur O'Sullivan, 2007)

En ThingsBoard los activos (*Assets*) son grupos jerárquicos, que se pueden relacionar con dispositivos, clientes y otros activos. Por lo general se crean activos en base a lugares operativos como: edificios, casas, laboratorios, procesos, fábricas; los cuales albergan varios dispositivos.

ThingsBoard admite las siguientes funciones de gestión de activos:

- Agregar y eliminar activos.
- Obtener ID de un activo.
- Asignar activos a los clientes.
- Administrar atributos de activos.
- Examinar alarmas de activos.
- Examinar eventos de activos.
- Gestionar relaciones de activos.

2.7.4 Dispositivos

ThingsBoard admite las siguientes funciones de administración de dispositivos mediante la interfaz de usuario web y la API REST. Se muestran a continuación en la Tabla 2.3 los usuarios y las funciones que pueden efectuar en los dispositivos.

Tabla 2.3: Funciones de dispositivos para cada rol de usuario.

Acción/Función	Roles de usuario		
	Administrador del Sistema	Inquilino (<i>Tenant</i>) /Administrador de la organización	Cliente *
Agregar y eliminar dispositivos.	NO	SI	NO
Administrar credenciales de dispositivos.	NO	SI	NO
Obtener ID de un dispositivo.	NO	SI	SI
Asignar dispositivos a los clientes.	NO	SI	NO
Examinar atributos del dispositivo.	NO	SI	SI
Explorar telemetría del dispositivo.	NO	SI	SI
Examinar alarmas del dispositivo.	NO	SI	SI
Examinar eventos del dispositivo.	NO	SI	SI
Administrar relaciones de dispositivo.	NO	SI	NO

* Nota: Las funciones de administración de dispositivos que tiene el usuario cliente solo están disponibles para los dispositivos que tiene asignado. Estos permisos los otorga un usuario administrador de la organización.

2.7.5 Alias de las Entidades

Para realizar un panel o dashboard, como primer paso se debe crear un Alias de entidad, con estos se definen las entidades a utilizar. Estos Alias pueden ser muy sencillos, como referirse a un único dispositivo, o caso contrario, muy complicados como para crear una consulta de búsqueda compleja, dependiendo de lo que se necesite en la creación del panel.

El esquema de la Figura 2.7 indica una estructura de activos y dispositivos usual. Un Alias de entidad puede ubicar un solo dispositivo o activo, así como también varios dispositivos o activos dependiendo de varios criterios de búsqueda.

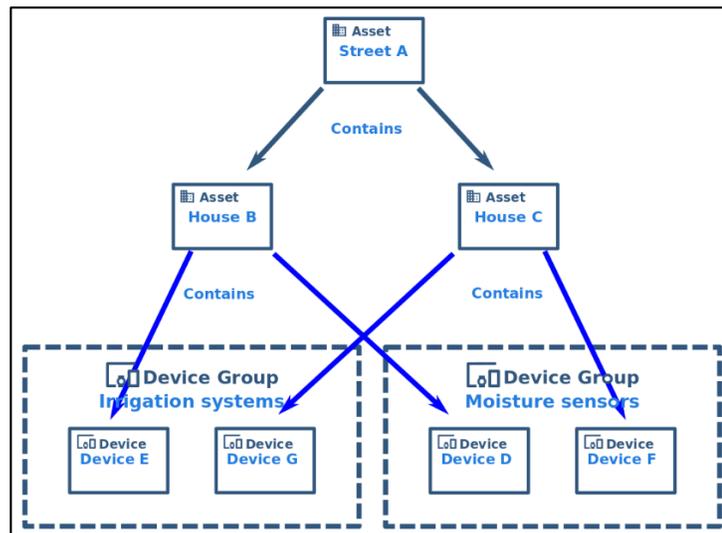


Figura 2.7: Esquema jerárquico de los Alias de entidad.

Fuente: (ThingsBoard, 2020)

Cada panel tiene sus Alias de entidad definidos, para crear un Alias de entidad se debe entrar en modo de edición del panel y posterior a esto hacer clic en el ícono superior de la barra del panel como se muestra en la Figura 2.8.

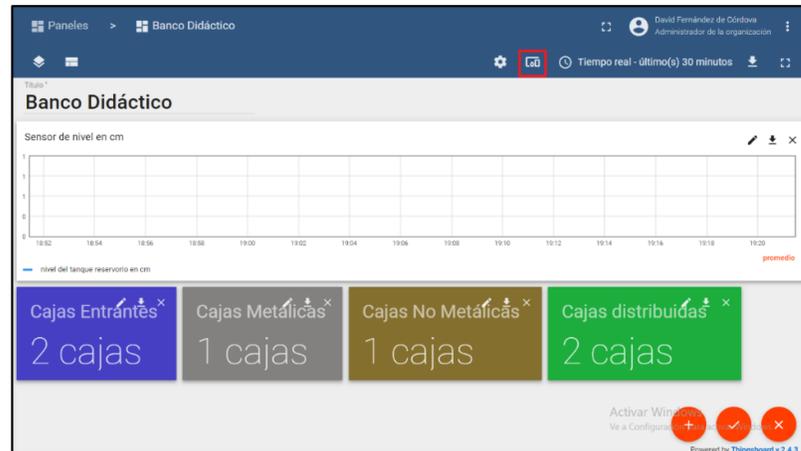


Figura 2.8: Configuraciones de alias.

Posterior a esto se puede crear el Alias de entidad llenando los campos solicitados en las ventanas de configuración, como se muestra en las Figuras Figura 2.9 y Figura 2.10.



Figura 2.9: Ventana de configuración de alias.

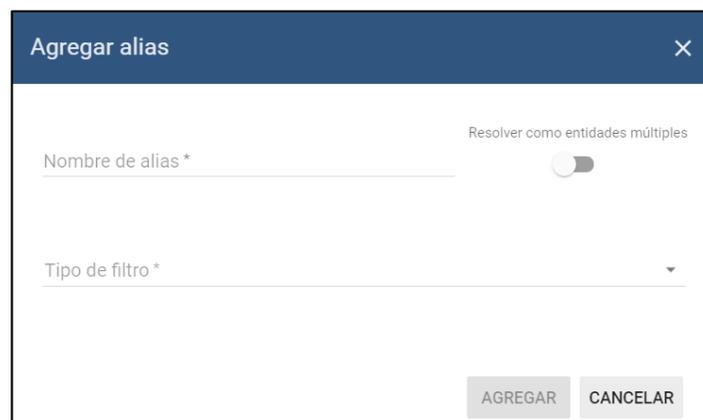


Figura 2.10: Ingreso de nuevo alias.

ThingsBoard (2020), en su documentación oficial indica que cuenta con varios tipos de Alias de Entidad como:

- Entidad simple
- Grupo de entidades
- Lista de entidades
- Nombre de identidad
- Lista de grupo de identidades
- Nombre de grupo de entidades
- Entidad del estado del tablero
- Tipo de activo
- Tipo de dispositivo
- Tipo de vista de entidad
- Consulta de relaciones
- Consulta de búsqueda de activos
- Consulta de búsqueda de dispositivos
- Consulta de búsqueda de vista de entidad

2.7.6 Motor de Reglas

El motor de reglas de la plataforma IoT es un entorno de trabajo fácil de usar, sirve para la creación de flujos de trabajo basado en eventos, el mismo tiene 3 componentes principales:

- Mensaje: corresponde a cualquier evento entrante, el mismo puede ser datos de dispositivos, eventos de ciclo de vida, entre otros.
- Nodo de regla: es una función que se ejecuta al entrar un mensaje, este nodo puede realizar varias acciones como filtrar, transformar o ejecutar alguna acción con el mensaje entrante.
- Cadena de reglas: es la conexión de nodos de reglas, un mensaje que sale de un nodo de reglas se envía hacia los siguientes nodos conectados.

Casos de uso de motor de reglas

El motor de reglas de ThingsBoard (ver Figura 2.11) es personalizable pensado para el procesamiento de eventos complejos, entre algunas de sus funcionalidades permite:

- Validación o modificación de datos entrantes previo al almacenamiento en la base de datos.
- Creación o eliminación de alarmas definidas según las condiciones configuradas.
- Envío de datos adicionales para su procesamiento.
- Envío de correos electrónicos cuando ocurra un evento o alarma configurada.
- Realizar llamadas RPC dependiendo de la función definida.

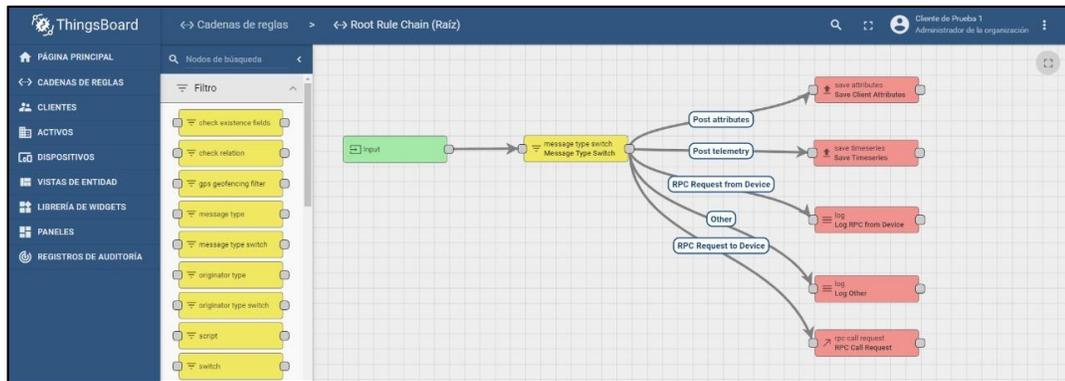


Figura 2.11: Motor de reglas.

2.7.7 Comandos RPC

ThingsBoard permite enviar llamadas de procedimiento remoto (RPC Remote Procedure Calls) desde aplicaciones del lado del servidor a dispositivos y viceversa. Básicamente, esta característica permite enviar comandos a dispositivos y recibir resultados de la ejecución de comandos. De manera similar, puede ejecutar la solicitud desde el dispositivo, aplicar algunos cálculos u otra lógica del lado del servidor en el back-end y enviar la respuesta al dispositivo.

La función RPC de ThingsBoard se puede dividir en dos tipos según el creador del mensaje: llamadas RPC originadas en el dispositivo y originadas en el servidor. Las llamadas RPC originadas en el dispositivo son llamadas RPC del lado del cliente, y las llamadas RPC originadas en el servidor son llamadas RPC del lado del servidor, apreciado en la Figura 2.12.

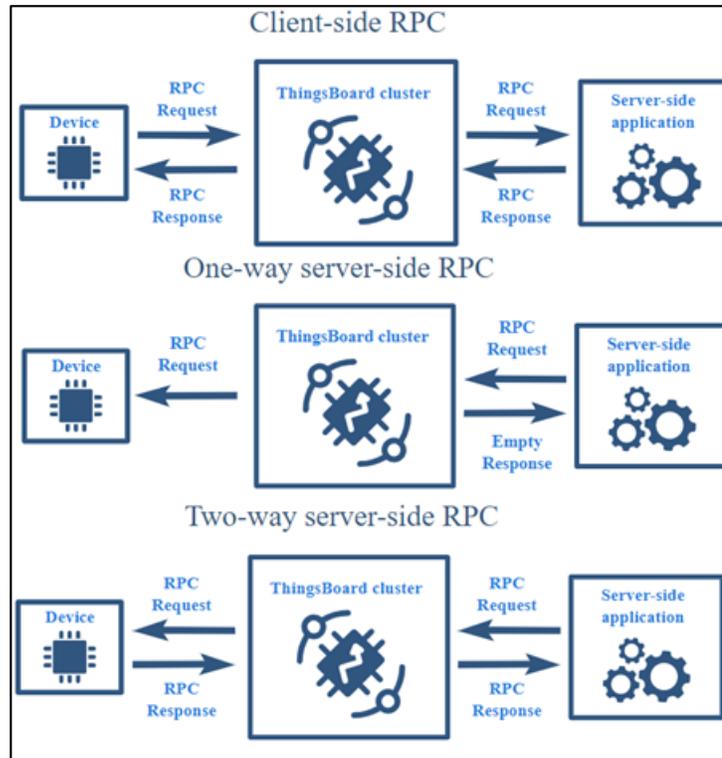


Figura 2.12: Función RPC cliente y servidor.

Fuente: (ThingsBoard, 2020)

ThingsBoard proporciona una API conveniente para enviar y recibir comandos RPC de aplicaciones que se ejecutan en el dispositivo. Esta API es específica para cada protocolo de red compatible: HTTP, CoAP y MQTT. El API para manejar comandos RPC utilizando el protocolo MQTT se explica más adelante en la sección 2.8.

2.7.8 Widgets Disponibles

ThingsBoard cuenta con varios paquetes de widgets, que se definen en la biblioteca de widgets, los cuales proporcionan diferentes funciones para el usuario final, como: visualización de distintos tipos de datos a través del tiempo, control remoto de dispositivos, gestión de alarmas y visualización de contenido HTML estático personalizado. Todos los paneles (*dashboards*) de IoT se construyen utilizando widgets de ThingsBoard. Se describen a continuación los paquetes con sus widgets correspondientes:

2.7.8.1 Tipos de widgets

Según las características proporcionadas hay cinco tipos de widgets, (ver Figura 2.13):

- Últimos valores
- Series de tiempo
- RPC (widget de control)
- Widget de alarma
- Estático



Figura 2.13: Tipos de Widgets.

Cada tipo de widget tiene su propia configuración de fuente de datos específica y la API de widget correspondiente. Cada widget requiere una fuente de datos para la visualización de los datos. Los tipos de fuente de datos disponibles dependen del tipo de widget del widget:

- **Dispositivo de destino (*target device*):** esta fuente de datos se utiliza en RPC básicamente, debe especificar el dispositivo de destino para el widget RPC
- **Fuente de alarma (*alarm source*):** utilizado en widgets de alarma. Esta fuente de datos requiere que la entidad fuente muestre las alarmas relacionadas y los campos de alarma correspondientes.
- **Entidad (*entity*):** se utiliza tanto en las series de tiempo como en los widgets del tipo últimos valores. Básicamente, se debe especificar la entidad de destino y la clave de la serie de tiempo o el nombre del atributo.

- **Función (*function*):** utilizado tanto en las series de tiempo como en los widgets del tipo últimos valores, con fines de depuración. Básicamente, se puede especificar una función de JavaScript que emulará los datos de un dispositivo para ajustar la visualización del widget.

Tipo de widget: Últimos valores

Muestra los últimos valores de un atributo de la entidad particular o un punto de datos de una serie temporal (por ejemplo, cualquier widget de indicador o widget de tabla de entidades). Este tipo de widget utiliza los valores de atributos definidos de la entidad o series de tiempo como fuente de datos, (ver Figura 2.14).

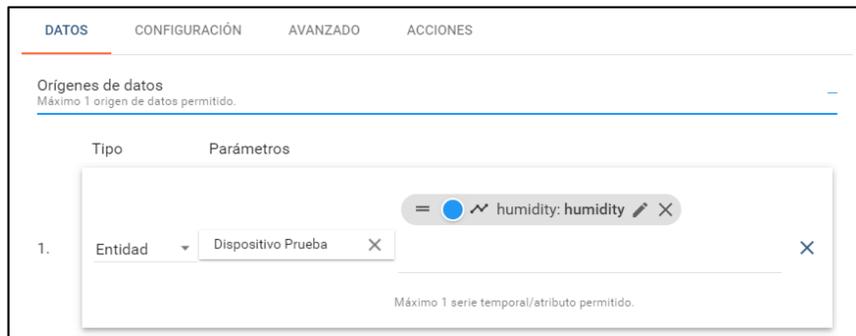


Figura 2.14: Fuente de datos del tipo de widget últimos valores.

En la Figura 2.15 se muestra un ejemplo de un widget del paquete Digital Gauges el cual pertenece al tipo de widget: últimos valores. Este widget muestra el valor actual de humedad de un dispositivo.



Figura 2.15: Widget digital gauge con último valor de humedad.

Tipo de widget: Series de tiempo

Muestra valores históricos para el período de tiempo seleccionado o los últimos valores de la fuente de datos. Puede ser en tiempo real, que actualiza los datos en base al intervalo de tiempo determinado; por ejemplo, cada 5 segundos. Este tipo de widgets (por ejemplo, el widget "*Timeseries - Flot*" del paquete de widgets Charts) usa solo valores de series de tiempo de la entidad seleccionada como fuente de datos. Para especificar el rango de tiempo de los valores mostrados, se utiliza la configuración de la ventana de tiempo. La ventana de tiempo se puede especificar a nivel del tablero o a nivel del widget.

En la Figura 2.16 se muestran los ajustes de la configuración de widgets de series temporales.



Figura 2.16: Fuente de datos del tipo de widget series de tiempo.

En la Figura 2.17 se muestra un ejemplo del widget "*Timeseries – Flot*" que pertenece al tipo de widget series de tiempo. En este ejemplo se muestra el amperaje de tres dispositivos distintos en tiempo real.

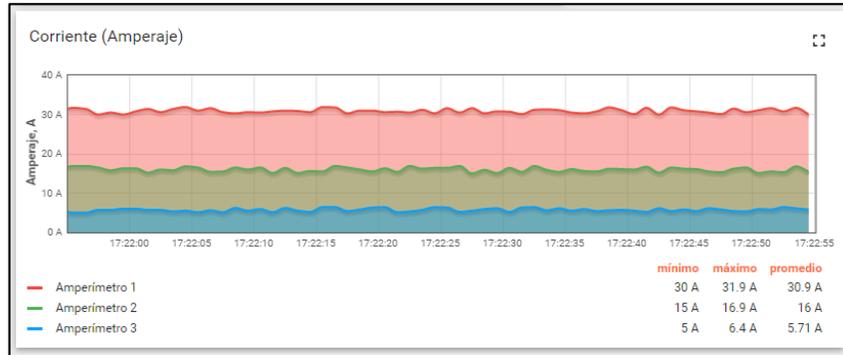


Figura 2.17: Widget “Timeseries - Flot” con valores de series temporales.

Tipo de widget: RPC (Control widget)

Se encuentran en el paquete Control widget, este tipo de widget permite enviar comandos RPC a dispositivos y puede manejar/visualizar la respuesta del dispositivo (por ejemplo, "Control de GPIO Raspberry Pi"). Los widgets RPC se configuran especificando el dispositivo como el objetivo para los comandos RPC, como se indica en la Figura 2.18.



Figura 2.18: Dispositivo objetivo en widget tipo RPC.

En la Figura 2.19 se muestran ejemplos de widgets del tipo RPC para controlar o visualizar el estado de un dispositivo.



Figura 2.19: Widgets del tipo RPC.

Tipo de widget: Widget de Alarma

Este widget muestra alarmas relacionadas con la entidad especificada en la ventana de tiempo determinado, en donde, la ventana de tiempo puede estar determinada por el panel o puede tener una ventana de tiempo independiente. Los widgets de alarma se configuran especificando la entidad como fuente de alarmas y los campos de alarma correspondientes.

Además, la configuración contiene los parámetros "Estado de la alarma" e "Intervalo de sondeo de alarmas". El parámetro "Estado de la alarma" determina cuales alarmas se mostrarán en el widget de acuerdo a su estado, mientras que el "Intervalo de sondeo de alarmas" controla la frecuencia de búsqueda de las alarmas en segundos, (ver Figura 2.20).

Figura 2.20: Configuraciones widget de alarma.

En la Figura 2.21 se muestra un ejemplo del widget "Tabla de alarmas" que muestra las últimas alarmas activadas en tiempo real.

Tiempo de creación	Origen	Tipo	Criticidad	Estado
2020-02-06 16:30:08	Dispositivo Prueba	Temperatura Alta	Alta	Activa ignorada
2020-02-06 16:30:06	Dispositivo Prueba	Humedad Alta	Alta	Activa ignorada

Figura 2.21: Widget tabla de alarmas.

Tipo de widget: Estático

En la Figura 2.22 se muestra contenido HTML estático personalizable. Los widgets estáticos no utilizan ninguna fuente de datos y generalmente se configuran especificando contenido HTML estático y, opcionalmente, estilos CSS.

```

CSS
1 .card {
2   font-weight: bold;
3   font-size: 32px;
4   color: #999;
5   width: 100%;
6   height: 100%;
7   display: flex;
8   align-items: center;

```

```

HTML
1 <h1 style="text-align: center;">Widget estático - Título</h1>
2 <div class='card'>Contenido</div>

```

Figura 2.22: Configuración de widget estático.

En la Figura 2.23 siguiente se muestran dos ejemplos del widget estático “HTML card”. El widget HTML card permite insertar contenido HTML en el mismo, por ejemplo, una imagen.



Figura 2.23: Widgets HTML card con contenido html estático.

2.7.9 Paquetes de widgets

Los widgets se agrupan en paquetes de widgets según su propósito. Hay paquetes de widgets a nivel de sistema (Usuario Administrador del sistema) y nivel de inquilino (Usuario Inquilino). La instalación inicial de ThingsBoard cuenta con el conjunto básico de paquetes de widgets a nivel de sistema.

Hay más de treinta widgets separados en siete paquetes de widgets disponibles, listos para usarse, Figura 2.24. Los paquetes a nivel del sistema pueden manejarse por un usuario administrador del sistema y están disponibles para su uso por cualquier usuario inquilino en el sistema. Los paquetes de nivel de inquilino pueden ser administrados por un usuario “Administrador de la Organización” y están disponibles para su uso solo por este inquilino y sus clientes correspondientes.

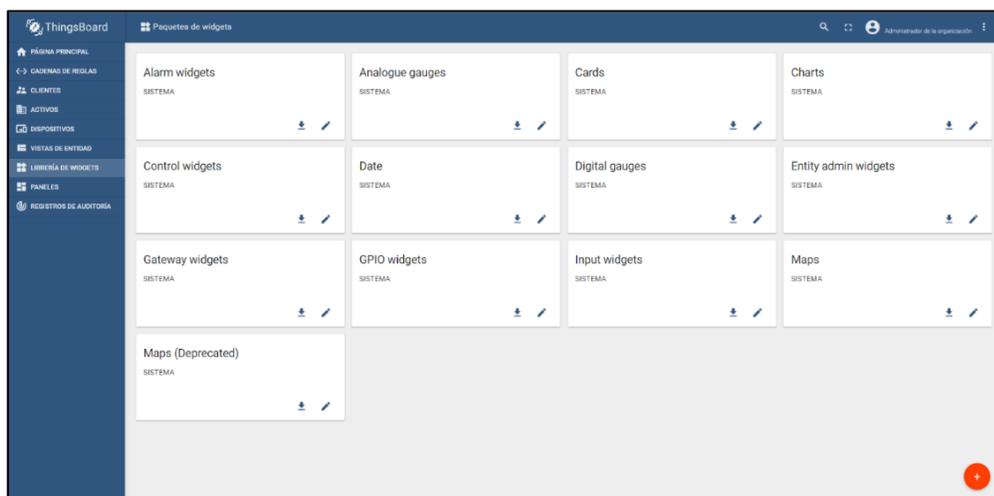


Figura 2.24: Paquetes de Widgets en ThingsBoard.

2.7.9.1 Paquete Digital Gauges

Indicadores digitales comúnmente utilizados para visualizar métricas de temperatura, humedad, velocidad o cualquier otro tipo de variable de tipo numérico entero o con punto flotante, como se aprecia en la Figura 2.25.

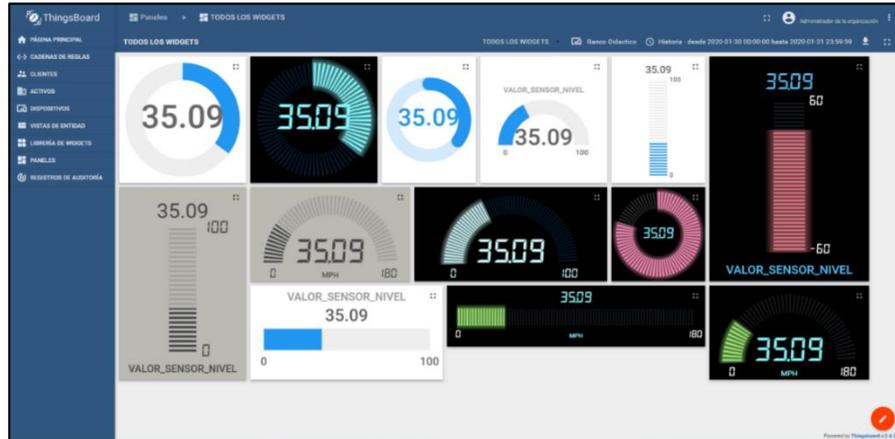


Figura 2.25: Widgets del paquete Digital Gauges.

2.7.9.2 Paquete Analogue Gauges

Indicadores analógicos similares a los indicadores digitales. Contiene widgets de tipo: últimos valores, (ver Figura 2.26).



Figura 2.26: Widgets del paquete Analogue Gauges.

2.7.9.3 Paquete Cards

Útil para visualizar datos de series temporales en una tabla o en un widget de tipo card. El widget card contiene información textual y relevante dentro de una tarjeta de relieve que contrasta con el fondo del panel, (ver Figura 2.27). Contiene widgets de tipo: series temporales, últimos valores y un widget estático.

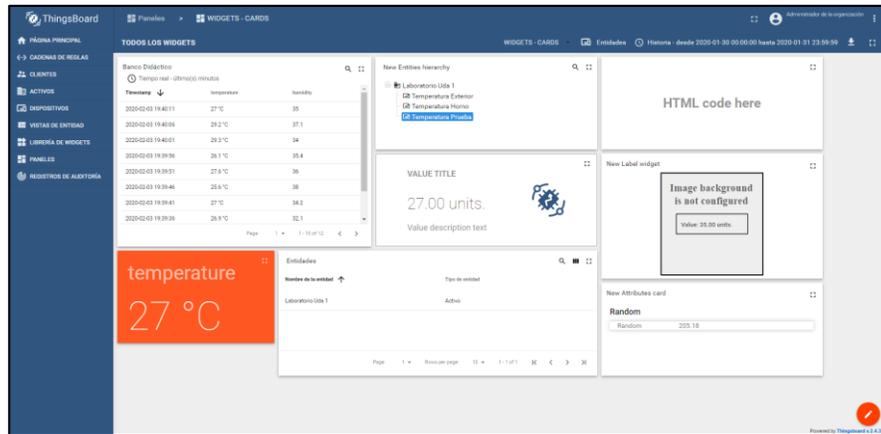


Figura 2.27: Widgets del paquete Cards.

2.7.9.4 Paquete Charts

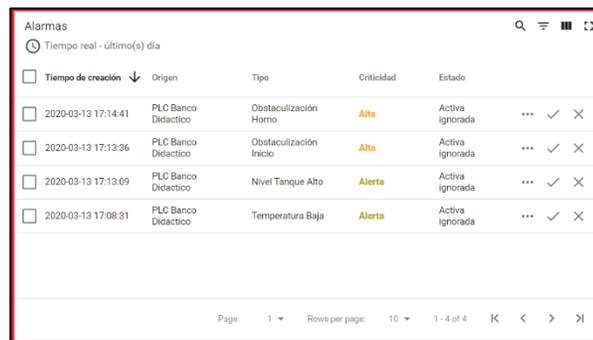
Útil para visualizar datos en tiempo real o en histórico en base a un rango de fechas seleccionado. Contiene widgets de tipo: series temporales y últimos valores, (ver Figura 2.28).



Figura 2.28: Widgets del paquete Charts.

2.7.9.5 Paquete Alarm Widgets

El paquete Alarm widgets cuenta con un solo widget. Este widget de Alarma permite notificar alarmas que se encuentren asignadas a un dispositivo o entidad en la lógica definida en el motor de reglas. Por ejemplo, cuando un valor de temperatura de un dispositivo supere cierto rango definido por un setpoint. Se pueden visualizar alarmas en tiempo real o las generadas en algún momento previo en el histórico, (ver Figura 2.29).



Timeo de creación	Origen	Tipo	Criticidad	Estado			
2020-03-13 17:14:41	PLC Banco Didactico	Obstaculización Horno	Alta	Activa ignorada	...	✓	✕
2020-03-13 17:13:36	PLC Banco Didactico	Obstaculización Inicio	Alta	Activa ignorada	...	✓	✕
2020-03-13 17:13:09	PLC Banco Didactico	Nivel Tanque Alto	Alerta	Activa ignorada	...	✓	✕
2020-03-13 17:08:31	PLC Banco Didactico	Temperatura Baja	Alerta	Activa ignorada	...	✓	✕

Figura 2.29: Widgets del paquete Alarmas.

2.7.9.6 Paquete Control Widgets

Permite visualizar el estado actual del dispositivo, así como también enviar comandos RPC al dispositivo objetivo. Con estos widgets se puede controlar el dispositivo objetivo, (ver Figura 2.30).

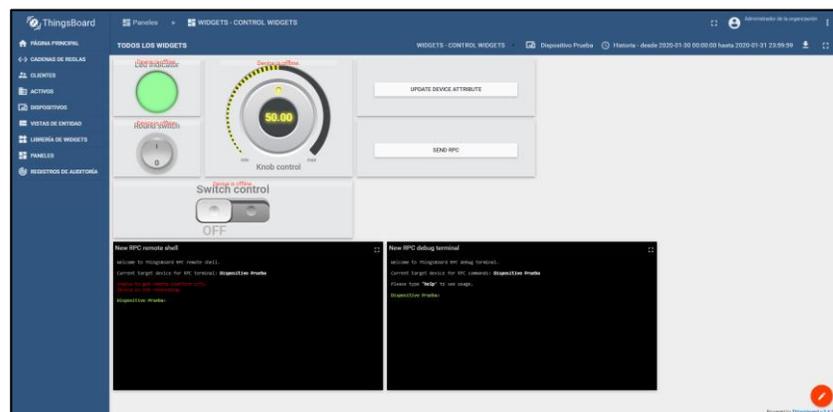


Figura 2.30: Widgets del paquete Control Widgets.

2.7.9.7 Paquete Date

Este widget permite seleccionar un rango de fechas e intervalo de tiempo, (ver Figura 2.31), para visualizar los datos en el historial del periodo seleccionado. Se agrega a un dashboard (panel) y actualiza todos los widgets que se encuentren en el mismo al tiempo correspondiente, de acuerdo al periodo seleccionado, facilitando el análisis de la información.

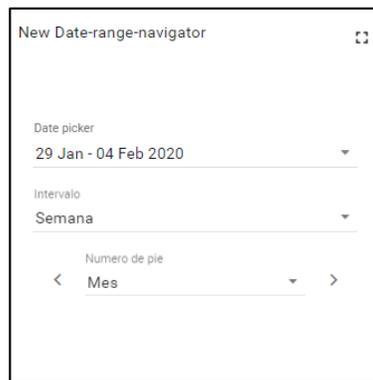


Figura 2.31: Widget de fecha.

2.7.9.8 Paquete Entity Admin Widgets

Este paquete contiene dos widgets del tipo últimos valores: Tabla de administración de activos y Tabla de administración de dispositivos. Estos widgets permiten editar información general de los activos o de los dispositivos que se encuentran en la tabla correspondiente, así como también la posibilidad de eliminar un activo o dispositivo, (ver Figura 2.32).

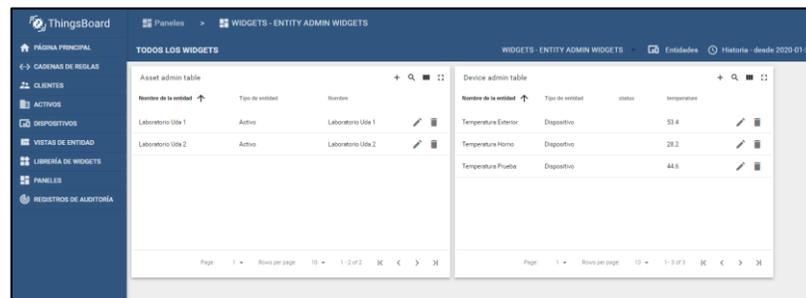


Figura 2.32: Widgets del paquete Entity Admin Widgets.

2.7.9.9 Paquete Gateway Widgets

Contiene un widget que permite manejar extensiones, se muestra en la Figura 2.33, permite exportar e importar configuraciones de extensiones. Este tipo de widget se utiliza conjuntamente con ThingsBoard Gateway.



Figura 2.33: Widget de extensiones.

2.7.9.10 Paquete Input Widgets

En este paquete se encuentran widgets que permiten cambiar los atributos de entidades definidas, como se muestra en la Figura 2.34. Existen 3 tipos de atributos que se pueden asignar a dispositivos: Atributos del cliente, Atributos del servidor y Atributos compartidos. Con estos widgets se pueden actualizar los valores de un atributo de cualquier tipo, que corresponda al dispositivo seleccionado.

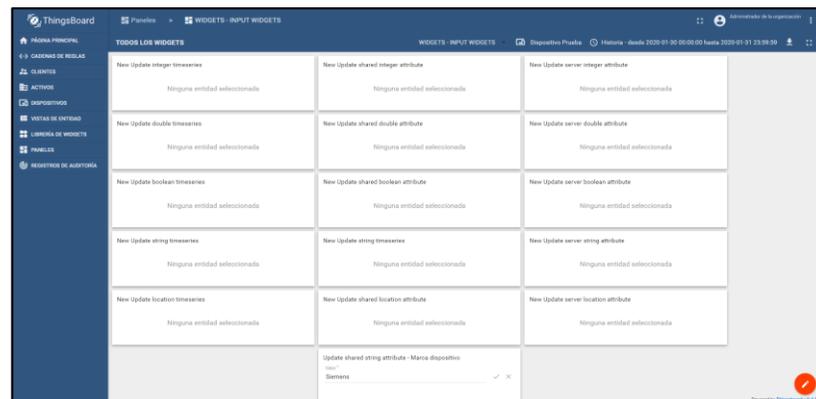


Figura 2.34: Widgets del paquete Input Widgets.

2.7.9.11 Paquete GPIO Widgets

Estos Widgets, como su nombre indica, permiten visualizar y controlar el estado de los pines GPIO de una tarjeta electrónica, se muestra en la Figura 2.35, como por ejemplo de la Raspberry Pi 3. La configuración personalizable de estos widgets admite agregar o quitar tantos pines como sea necesario.

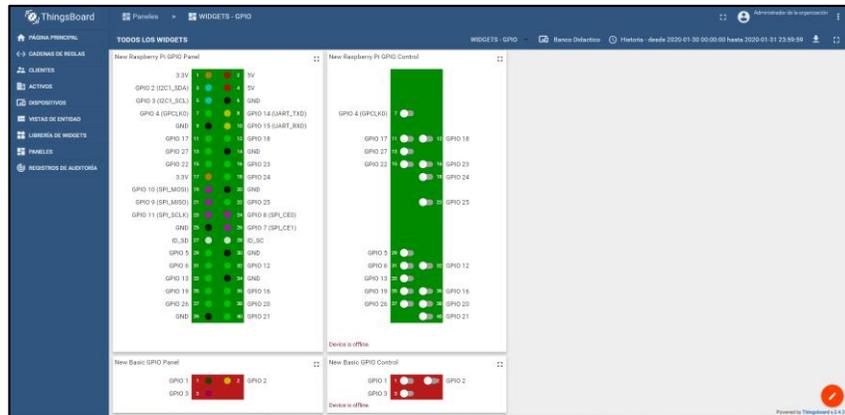


Figura 2.35: Widgets del paquete GPIO Widgets.

2.7.9.12 Paquete Maps

Contiene widgets útiles para la visualización de ubicaciones geográficas de dispositivos y seguimiento de ruta de dispositivos, como se muestra en la Figura 2.36, tanto en tiempo real como en modo historial.

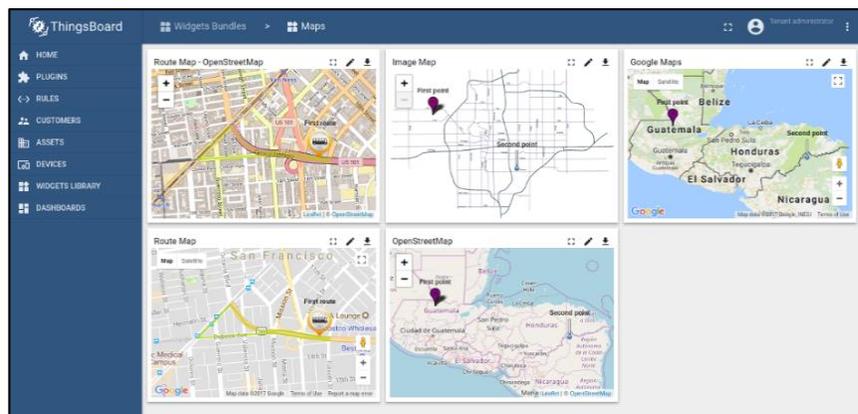


Figura 2.36: Paquete de widgets Maps.

2.7.10 Importar/Exportar Paneles, Widgets (Tipos Y Paquetes)

La plataforma ThingsBoard tiene la posibilidad de importar y exportar: paneles, widgets, paquete de widgets, como se muestra en las Figuras 2.37 y 2.38. Estos se exportan y descargan en un archivo en formato JSON.

Exportación de paquete de widgets



Figura 2.37: Exportar Widget y Exportar Panel.



Figura 2.38: Exportar paquete de widgets.

Importación de paquete de widgets

Se pueden importar widgets, paneles o paquete de widgets utilizando el botón de acción de importar, que se encuentran en la esquina inferior derecha de las pantallas correspondientes. Al hacer clic en el botón de importar, ver Figura 2.39, se despliega una ventana de carga que admite un archivo en formato JSON, tal como se descarga al momento de exportar un objeto. Para importar el objeto se arrastra el archivo JSON a la ventana, Figura 2.40, y se hace clic en Importar.

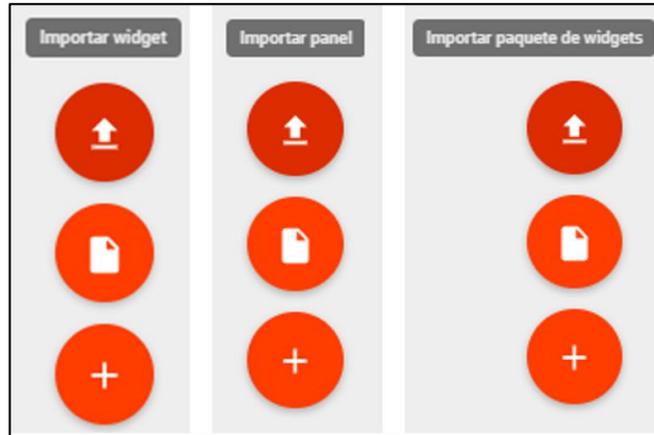


Figura 2.39: Opciones de importación.

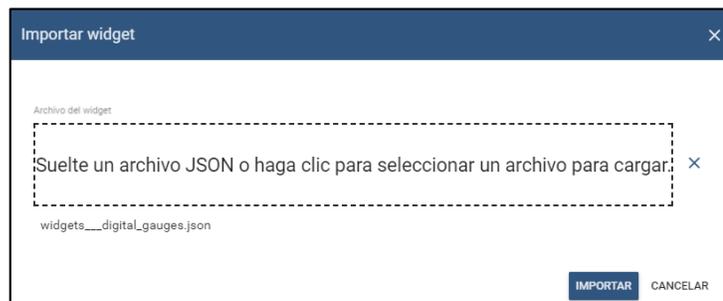


Figura 2.40: Ventana para importar Widget en formato JSON.

2.8 ThingsBoard API

2.8.1 API MQTT

ThingsBoard cuenta con un API que utiliza el protocolo MQTT para enviar telemetría, la cual se asigna a un dispositivo. Los nodos del servidor de ThingsBoard actúan como un agente de MQTT que admite los niveles de calidad de servicio: 0 (como máximo una vez) a 1 (al menos una vez), y un conjunto de *topics* predefinidos. Se utilizarán estos *topics* predefinidos para enviar datos a la plataforma.

API para carga de telemetría

Para publicar datos de telemetría en el nodo del servidor de ThingsBoard, se envía un mensaje con el *topic*:

v1/devices/me/telemetry

Los formatos de datos compatibles más simples, para el cuerpo del mensaje, son:

{"key1":"value1", "key2":"value2"}

[{"key1":"value1"}, {"key2":"value2"}]

Por defecto el servidor asignará la marca de tiempo del dato (*timestamp*), sin embargo, se puede enviar la marca de tiempo utilizando el siguiente formato en el cuerpo del mensaje:

{"ts":1451649600512, "values":{"key1":"value1", "key2":"value2"}}

API de atributos

Los dispositivos cuentan con 3 tipos de atributos: atributos del cliente, atributos del servidor y atributos compartidos.

La API de atributos de ThingsBoard permite que los dispositivos puedan:

- Subir atributos del dispositivo del lado del cliente al servidor.
- Solicitar atributos del lado del cliente y atributos compartidos por el cliente y el servidor.
- Suscribirse a los atributos compartidos del dispositivo desde el servidor.

Publicar actualización de atributos

Para publicar los atributos del dispositivo del lado del cliente en el servidor de ThingsBoard, se envía un mensaje con el *topic*:

v1/devices/me/attributes

Solicitar atributos del servidor

Para solicitar al servidor de ThingsBoard atributos del cliente o atributos compartidos:

El cliente debe suscribirse al *topic*:

v1/devices/me/attributes/response/+

Luego debe enviar un mensaje al *topic*:

v1/devices/me/attributes/request/\$request_id

en donde *\$request_id* es cualquier número entero positivo que servirá de identificador para la respuesta del servidor, por ejemplo 1. El formato del mensaje es el siguiente:

```
{"clientKeys":"attribute1,attribute2", "sharedKeys":"shared1,shared2"}
```

En donde *clientKeys* son los nombres de los atributos del lado del cliente a consultar y *sharedKeys* son los nombres de los atributos compartidos a consultar.

Seguido de esto el servidor responderá al cliente con los valores de los atributos en el *topic* que el cliente se suscribió: **v1/devices/me/attributes/response/+**.

Suscripción a las actualizaciones de atributos

Para suscribirse a los cambios de atributos compartidos de los dispositivos, el cliente debe suscribirse al *topic*:

v1/devices/me/attributes

2.8.2 API RPC

Esta API cuenta con una capa de seguridad que utiliza las credenciales del dispositivo, sean de token de acceso o de certificado X.509. Se utilizará el tipo de credencial de token de acceso. La aplicación que se conecte a ThingsBoard a través del protocolo MQTT deberá autenticarse con el token de acceso en el campo de usuario de las credenciales en la conexión MQTT.

Como se mencionó en la sección 3.6.8 existen dos tipos de mensajes RPC: generados desde el servidor y generados desde el cliente. A continuación, se describe cómo comunicarse por medio del protocolo MQTT para los dos casos.

RPC desde el servidor

En este caso el servidor enviará mensajes MQTT al dispositivo o a los dispositivos requeridos. Para suscribirse a los comandos RPC desde el servidor, el cliente MQTT, en este caso un dispositivo IoT, deberá suscribirse al siguiente *topic*:

v1/devices/me/rpc/request/+

Una vez suscrito, el cliente recibirá comandos de forma individual en cada mensaje enviado por el servidor Broker MQTT. Estos mensajes se enviarán por el servidor con el *topic*:

v1/devices/me/rpc/request/\$request_id

En donde *\$request_id* es un identificador entero positivo que asigna el servidor.

Seguido de esto el cliente deberá responder el mensaje al siguiente *topic*:

v1/devices/me/rpc/response/\$request_id

RPC desde el cliente

En este caso el cliente enviará mensajes MQTT al servidor. Para enviar comandos RPC al servidor, el cliente MQTT deberá publicar un mensaje al siguiente *topic*:

v1/devices/me/rpc/request/\$request_id

En donde *\$request_id* es un identificador entero positivo generado por el cliente. La respuesta del servidor se publicará en el siguiente *topic*:

v1/devices/me/rpc/response/\$request_id

CAPÍTULO 3: CONFIGURACIÓN Y PROGRAMACIÓN DE LOS EQUIPOS DE CONTROL

3.1 Programación del PLC 1212C AC/DC/RLY Siemens

Para comandar el proceso del banco didáctico se realiza la programación del PLC de manera organizada en bloques de programación. Se utiliza en total 8 bloques de programación, entre tipo FC, FB y DB. Todos los bloques nombrados se adjuntan al bloque principal Main para permitir su ejecución, los mismos se muestran en la Figura 3.1.

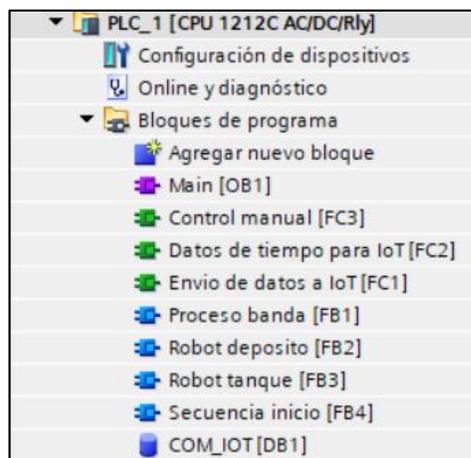


Figura 3.1: Bloques de programación en el PLC.

La descripción de cada uno de los bloques de programa se encuentra en la sección de Anexos como Anexo 4: Código implementado en el PLC s7-1200 1212c AC/DC/RLY.

3.1.1 Secuencia inicio [FB4]

Al iniciar el banco didáctico, el primer bloque que se ejecuta es “Secuencia inicio”, el mismo está a la espera de la señal proveniente del pulsante de inicio para mover el brazo robótico a su posición inicial listo para la distribución de las cajas metálicas o no metálicas.

3.1.2 Proceso banda [FB1]

En el momento en el que se ingresa una caja, ya sea metálica o no metálica se ejecuta la función “Proceso banda” gracias a la señal enviada por el sensor capacitivo ubicado al inicio de la banda, la cual activa la banda transportadora, detiene la caja en el horno durante el tiempo determinado en el programa y posteriormente lleva la caja hacia el sensor óptico para que el brazo robótico traslade la caja según corresponda su material.

3.1.3 Robot tanque FB3 y Robot deposito [FB2]

Si la caja ingresa es una caja metálica se ejecutará la función “Robot tanque”, esta función espera la señal del sensor inductivo al inicio de la banda, la misma que será enviada hacia el PLC para informar que se ingresó una caja metálica, esta señal se almacena para que cuando la caja esté lista para su traslado, el brazo robot tenga el conocimiento de que la caja metálica se transporta al tanque, caso contrario, si no recibe señal del sensor inductivo es una caja no metálica y se ejecutará la función “Robot deposito”, la cual indica que el transporte de la caja no metálica es solamente hacia el depósito.

3.1.4 Envío de datos de IoT [FC1]

El bloque de función “Envío de datos de IoT”, tiene toda la labor de enviar los datos que genera el banco didáctico en todo el proceso, todos los datos podrán ser visualizados en el servidor instalado en la nube:

- Cajas ingresadas.
- Cajas metálicas.
- Cajas no metálicas.
- Cajas distribuidas.
- Falla de robot antes de la distribución.
- Falla del robot durante la distribución.
- Obstaculización al inicio de la banda.
- Obstaculización en el horno.
- Valores de nivel de tanque y temperatura del horno en tiempo real.
- Alarmas de nivel y temperatura.

3.1.5 Envío de tiempos para IoT [FC2]

En el bloque de función “Envío de tiempos para IoT” se envían los tiempos que toma cada etapa del proceso, los mismo que servirán para su posterior análisis si es requerido.

Dentro del proceso se tienen 2 etapas:

- Etapa 1: recorrido de las cajas por la banda transportadora.
- Etapa 2: distribución de caja hacia depósito o tanque, según corresponda.

Estos tiempos se mostrarán en un informe el cual recopilará los datos de todas las cajas que ingresen en el proceso, de esta forma se podrá observar cómo trabaja el banco didáctico y los puntos a mejorar del mismo.

3.1.6 Control manual [FC3]

En el bloque “Control manual” se encuentran las ordenes provenientes de la nube para el control completo de la banda transportadora y del brazo robótico, la finalidad de este bloque es poder otorgar al usuario la capacidad de control de la maqueta de manera remota según sea conveniente.

3.1.7 COM_IOT [DB1]

En este bloque de base de datos, se almacenan las variables que serán enviadas y recibidas desde la nube a través del módulo IoT, para lo cual es necesario realizar las siguientes configuraciones:

- En las propiedades del bloque de base de datos COM_IOT, se debe desactivar la pestaña de acceso optimizado al bloque, (ver Figura 3.2).

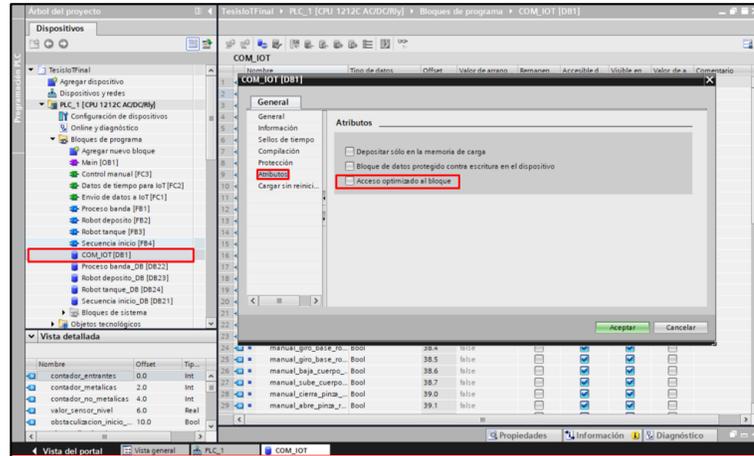


Figura 3.2: Desactivación del acceso optimizado al bloque DB.

3.2 Configuración de las propiedades del PLC.

En las propiedades del dispositivo PLC, en la pestaña de protección se selecciona la opción “Permitir el acceso vía comunicación PUT/GET”, para firmware de PLC 4.0 o superiores, como se aprecia en la Figura 3.3. En caso de tener una versión de firmware anterior, en la misma pestaña de protección se debe activar la pestaña “Sin protección”, como se aprecia en la Figura 3.4.

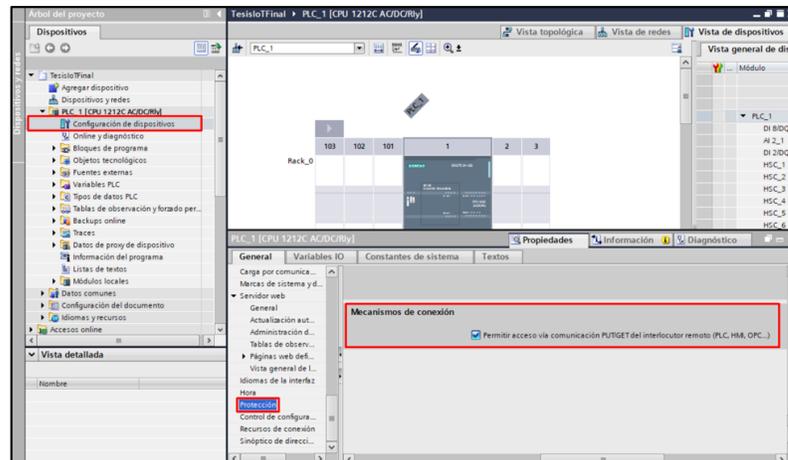


Figura 3.3: Habilitación del acceso PUT/GET.

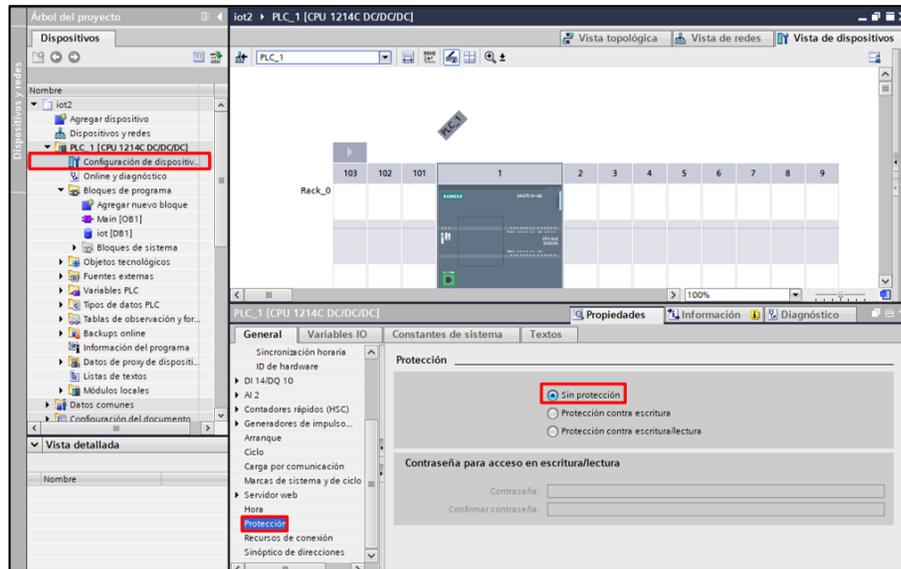


Figura 3.4: Desactivación de protección del dispositivo.

3.3 Configuración del módulo Siemens IoT 2040.

En esta sección se describen los pasos para instalar el sistema operativo Yocto proporcionado por Siemens que se encuentra adecuado para el módulo IoT 2040, luego se procede a configurar el programa Node-Red en el módulo IoT para la conexión con dispositivos PLC Siemens y se describe un ejemplo de programa en Node-Red.

3.3.1 Configuración inicial del módulo IoT

Para la instalación del sistema operativo y la configuración inicial en el módulo IoT se necesitan los siguientes recursos:

- Tarjeta microSD de entre 8 y 32 Gb.
- Adaptador de tarjetas microSD para lectura en computador.
- 2 cables Ethernet.
- Módulo Siemens SIMATIC IOT 2040.
- Antena adaptador WiFi a USB, Figura 3.5.
- PLC Siemens.
- Conexión estable a internet.



Figura 3.5: Adaptador WiFi a USB utilizado.

Se describen a continuación las configuraciones iniciales para el funcionamiento del módulo IoT:

1. Escritura de Firmware en el módulo IoT (SIEMENS, 2018).

Como primera intervención, en la tarjeta microSD, se debe escribir la imagen de firmware proporcionada por Siemens y creada con el proyecto Yocto, el cual posee herramientas que permiten la creación de distribuciones del sistema de Linux para el manejo de software embebido e IoT. La imagen a instalar ha sido proporcionada a la Universidad del Azuay, o la misma también puede ser descargada de la página oficial de Siemens:

https://support.industry.siemens.com/cs/attachments/109761191/IOT2000_RT-Image_V2.4.0.zip (SIEMENS, 2018).

Una vez descargada, se descomprime el archivo “.zip” para encontrar la imagen con nombre “example-V2.4.0_RC2.wic”. Para realizar la escritura de la imagen en la tarjeta SD, se requiere el Software Win32 Disk Imager (ver Figura 3.6), disponible en <https://sourceforge.net/projects/win32diskimager/>.



Figura 3.6: Software Win 32 Disk Imager.

Instalar el programa y abrirlo, luego se realizan los siguientes pasos, (ver Figura 3.7) (SIEMENS, 2018):

1. Se busca y selecciona la imagen a escribir.
2. Se selecciona la unidad donde se encuentra insertada la memoria SD.
3. Presionar “Write”.

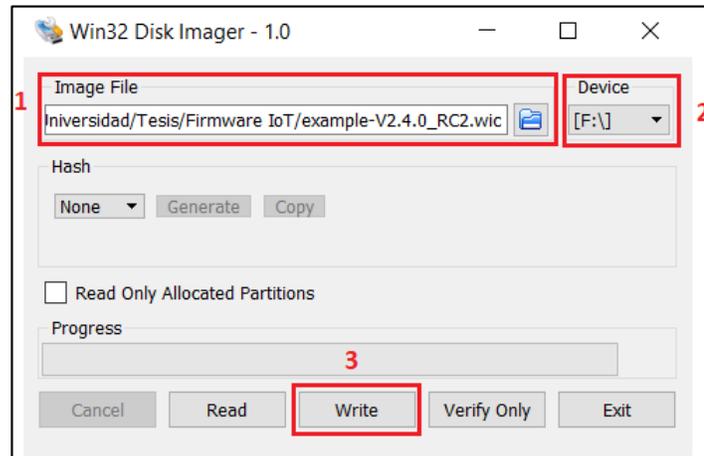


Figura 3.7: Proceso de escritura de imagen.

Se espera el tiempo necesario para que la escritura se complete, la tarjeta microSD se ha particionado en tres unidades, como se muestra en la Figura 3.8, dentro de esta se encuentran los archivos de arranque y el sistema de archivos del sistema operativo Yocto Linux.



Figura 3.8: Particiones creadas en la tarjeta microSD.

2. Colocación de la tarjeta dentro del módulo IoT (SIEMENS, 2018).

Cuando la tarjeta está lista con la imagen escrita, se la introduce en el módulo IoT de la siguiente manera:

Se debe desbloquear la ranura de la tarjeta microSD moviéndola hacia abajo como se muestra en la Figura 3.9.



Figura 3.9: Desbloqueo de la ranura de la tarjeta microSD.

Se abre la ranura nuevamente hacia abajo de manera suave para evitar daños de la misma, (ver Figura 3.10).



Figura 3.10: Apertura de la ranura de la tarjeta microSD.

Se coloca la tarjeta microSD de manera que calce correctamente en su espacio establecido, (ver Figura 3.11).



Figura 3.11: Colocación de la tarjeta microSD en el sitio designado.

Por último, se cierra la ranura y se la bloquea nuevamente, deslizando hacia adelante, para asegurar que la tarjeta se mantenga en su lugar en todo momento, (ver Figura 3.12).



Figura 3.12: Cierre y bloqueo de la ranura de la tarjeta microSD.

Conectar el módulo IoT como se muestra en la Figura 3.13.

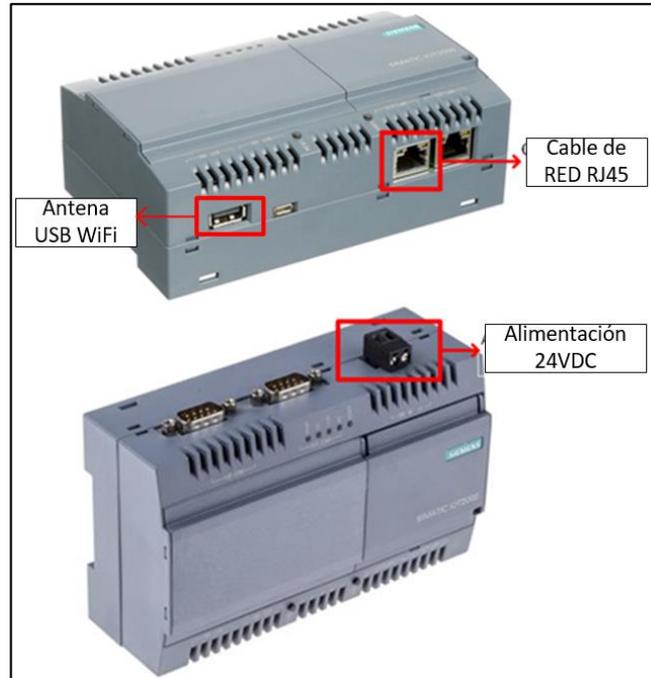


Figura 3.13 Conexión del módulo IoT.

3. Descarga del Software cliente SSH para el inicio de sesión en el módulo IoT (SIEMENS, 2018).

Se utilizará el software PuTTY como cliente SSH¹¹ para realizar las configuraciones pertinentes en el módulo IoT, el mismo puede ser descargado de <http://www.putty.org/>.

Para tener acceso al módulo se debe conectar un cable Ethernet al puerto X1 del mismo. El módulo tiene por defecto la dirección IP 192.168.200.1, y para que pueda existir comunicación con el computador este debe tener una dirección IP dentro del rango de red (192.168.200.2 – 192.168.200.254) y la máscara de subred 255.255.255.0, tal y como se muestra en la Figura 3.14.

¹¹ Cliente SSH: o Secure Shell, protocolo que realiza las comunicaciones de modo seguro utilizando arquitectura tipo cliente/servidor.

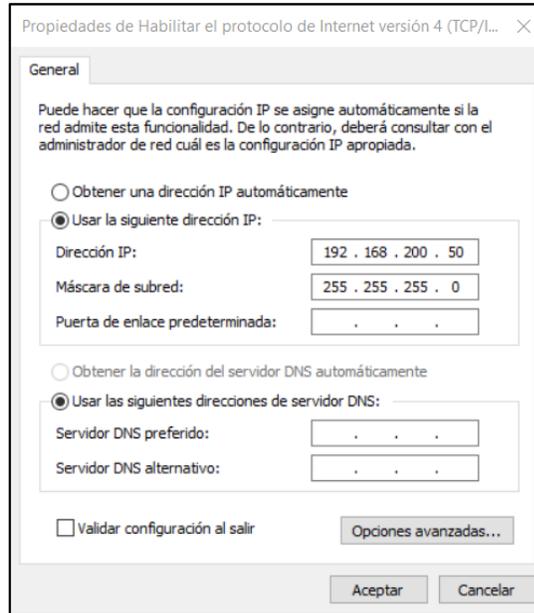


Figura 3.14: Configuración de dirección IP del computador.

Iniciar la herramienta PuTTY para realizar la comunicación SSH al módulo, Ingresar el IP 192.168.200.1 en el campo “Host Name”, puerto 22 (campo “Port”) y hacer clic en Open, como se aprecia en Figura 3.15.

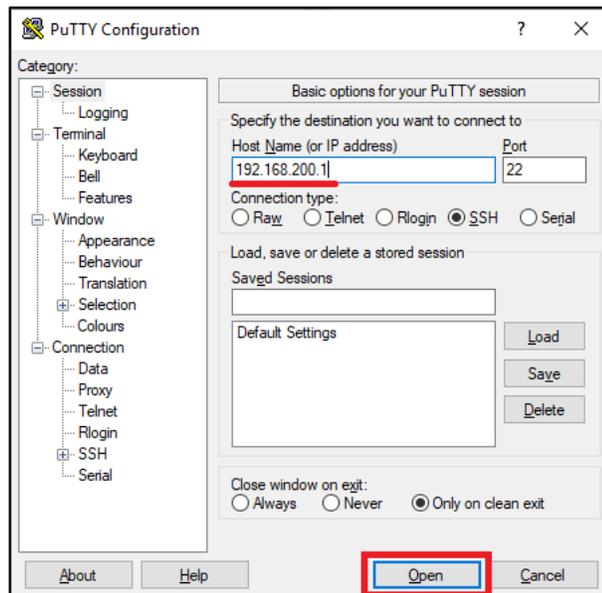


Figura 3.15: Conexión SSH con el módulo IoT utilizando PuTTY.

Luego de acceder a la conexión, PuTTY advertirá sobre una brecha de seguridad, hacer clic en SI/YES.



Figura 3.16: Aviso de seguridad PuTTY.

A continuación, se mostrará la pantalla de la Figura 3.17.



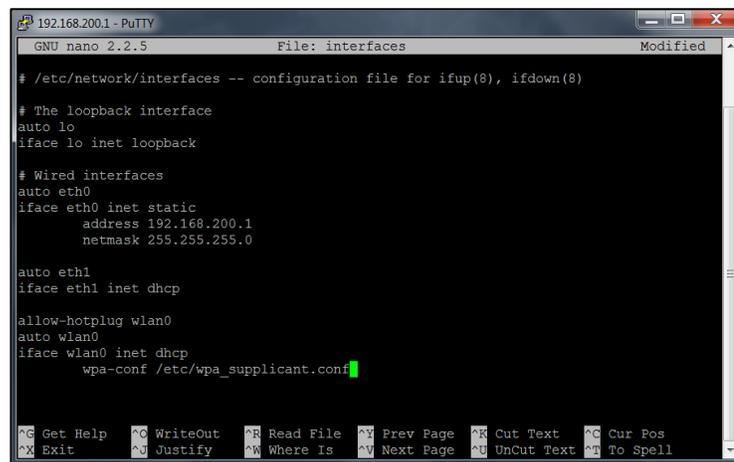
Figura 3.17: Pantalla de inicio en PuTTY.

Para iniciar sesión se utiliza el usuario root, por defecto este no tiene contraseña.

4. Configuración de conexión WiFi (SIEMENS, 2018).

Para realizar la conexión WiFi se debe definir la interfaz de red y su configuración, para esto iniciar sesión a través de PuTTY con el usuario root, por defecto no tiene contraseña, pero puede ser configurada posteriormente, ahora se empieza a modificar el archivo *interfaces* mediante los siguientes pasos, como se muestra en la Figura 3.18:

- Comando ***cd /etc/network/*** (acceso al directorio del archivo).
- Comando ***nano interfaces*** (modificación del archivo).
- Al final se añade el siguiente texto:
 - allow-hotplug wlan0*** (detecta conexión y desconexión de hardware en la interfaz)
 - auto wlan0*** (inicia automáticamente la interfaz cuando se enciende el dispositivo)
 - iface wlan0 inet dhcp*** (indica asignación de IP dinámica)
 - wpa-conf /etc/wpa_supplicant.conf*** (indica en que directorio se encuentra el archivo de configuración.)
- ***Ctrl + X*** y luego ***Y*** para aceptar, luego enter (guardar modificaciones).



```

192.168.200.1 - PuTTY
GNU nano 2.2.5 File: interfaces Modified
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# Wired interfaces
auto eth0
iface eth0 inet static
address 192.168.200.1
netmask 255.255.255.0

auto eth1
iface eth1 inet dhcp

allow-hotplug wlan0
auto wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant.conf
  
```

Figura 3.18: Ventana de configuración de red WiFi.

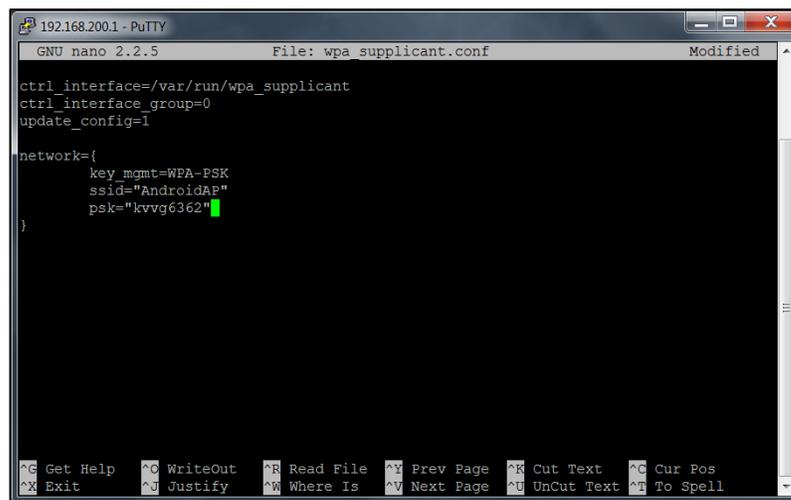
NOTA: La interfaz de red puerto X2 del módulo IoT se encuentra configurado en este archivo como la interfaz eth1, el mismo que tiene la configuración de DHCP para obtener una dirección IP automáticamente de la red a la que se conecte. Esto servirá a futuro para conectarse en una misma red con el PLC Siemens S7-1200.

Posteriormente se modifica el archivo *wpa_supplicant.conf* para determinar la red WiFi que vamos a utilizar:

- Comando *cd /etc*
- Comando *nano wpa_supplicant.conf* (modificación del archivo).
- Se modifica el apartado *network* con los datos de la conexión que se quiere realizar, (ver Figura 3.19).

```
key_mgmt=WPA-PSK
ssid="nombre_de_la_red"
psk="contraseña_de_la_red"
```

- *Ctrl + X* (guardar modificaciones), Luego “Y” y enter.



The screenshot shows a terminal window titled "192.168.200.1 - PuTTY" with the GNU nano 2.2.5 editor open to the file *wpa_supplicant.conf*. The configuration content is as follows:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

network={
    key_mgmt=WPA-PSK
    ssid="AndroidAP"
    psk="kvvg6362"
}
```

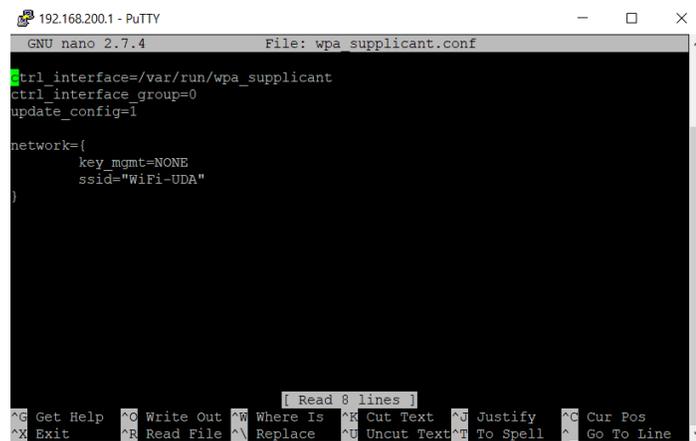
The terminal window also displays a status bar at the bottom with various keyboard shortcuts for nano editor operations.

Figura 3.19: Ventana de conexión WiFi.

Como se aprecia en el ejemplo, el módulo IoT se conectará a la red “Android AP” con contraseña “kvvg6362”.

NOTA: Si se desea ingresar a una red Wi-Fi de acceso libre, sin contraseña, entonces se accede al archivo de configuración con *nano /etc/wpa_supplicant.conf* y luego se inserta la siguiente configuración, como se muestra en la Figura 3.20:

```
key_mgmt=NONE
ssid="nombre_de_la_red"
```



```

192.168.200.1 - PuTTY
GNU nano 2.7.4 File: wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

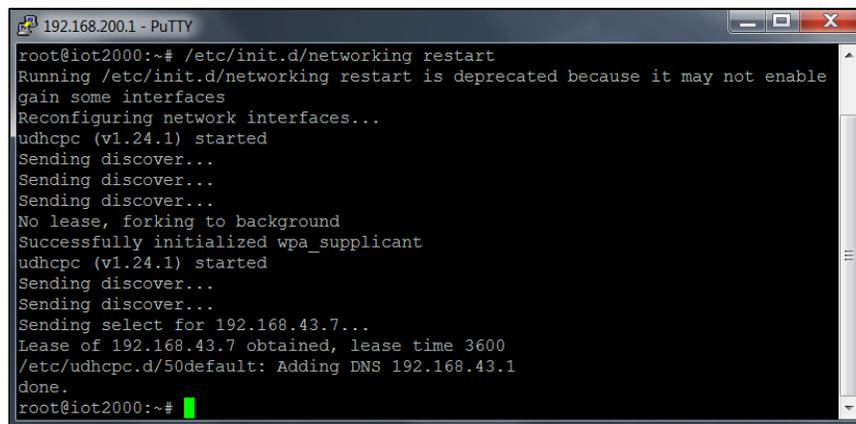
network={
    key_mgmt=NONE
    ssid="WiFi-UDA"
}

```

Figura 3.20: Ingreso a una red WiFi de acceso libre.

Finalmente se conecta el adaptador WiFi a USB en puerto USB del módulo IoT, adquirido anteriormente y se reinicia las interfaces para que el puerto wlan0 adquiera una dirección IP y se conecte al internet, (ver Figura 3.21).

- Comando `/etc/init.d/networking restart`



```

192.168.200.1 - PuTTY
root@iot2000:~# /etc/init.d/networking restart
Running /etc/init.d/networking restart is deprecated because it may not enable
gain some interfaces
Reconfiguring network interfaces...
udhcpc (v1.24.1) started
Sending discover...
Sending discover...
Sending discover...
No lease, forking to background
Successfully initialized wpa_supplicant
udhcpc (v1.24.1) started
Sending discover...
Sending discover...
Sending select for 192.168.43.7...
Lease of 192.168.43.7 obtained, lease time 3600
/etc/udhcpc.d/50default: Adding DNS 192.168.43.1
done.
root@iot2000:~#

```

Figura 3.21: Ventana de reinicio de los puertos de red.

Una vez conectado a la red se puede verificar la conexión realizando un ping a Google con el comando `ping www.google.com`, si existe respuesta la conexión a internet ha sido efectiva. Con la configuración de la conexión WiFi, ya es posible iniciar la aplicación Node-Red y descargar e instalar los paquetes de aplicación.

3.3.2 Instalación de Node-Red en el módulo IoT

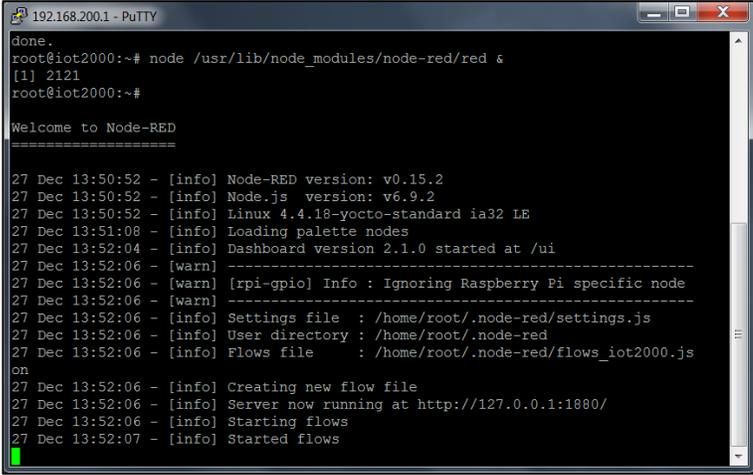
La forma en la que se programará las aplicaciones en el módulo IoT es a través de la herramienta de programación visual Node-Red, la misma se puede programar sin necesidad de conocimiento del lenguaje, ya que los programas se realizan en base a nodos, los cuales ya tienen sus funciones programadas.

Para iniciar Node-Red:

- Comando `node /usr/lib/node_modules/node-red/red &`

Esperar hasta que se muestre la imagen de la Figura 3.22

Cuando este se inicie, se procede a abrir un explorador de internet y se navega a la dirección 192.168.200.1:1880, esta dirección corresponde al servidor web que inicia Node-Red. como se muestra en la Figura 3.23.



```
192.168.200.1 - PuTTY
done.
root@iot2000:~# node /usr/lib/node_modules/node-red/red &
[1] 2121
root@iot2000:~#

Welcome to Node-RED
=====

27 Dec 13:50:52 - [info] Node-RED version: v0.15.2
27 Dec 13:50:52 - [info] Node.js version: v6.9.2
27 Dec 13:50:52 - [info] Linux 4.4.18-yocto-standard ia32 LE
27 Dec 13:51:08 - [info] Loading palette nodes
27 Dec 13:52:04 - [info] Dashboard version 2.1.0 started at /ui
27 Dec 13:52:06 - [warn] -----
27 Dec 13:52:06 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
27 Dec 13:52:06 - [warn] -----
27 Dec 13:52:06 - [info] Settings file : /home/root/.node-red/settings.js
27 Dec 13:52:06 - [info] User directory : /home/root/.node-red
27 Dec 13:52:06 - [info] Flows file : /home/root/.node-red/flows_iot2000.js
on
27 Dec 13:52:06 - [info] Creating new flow file
27 Dec 13:52:06 - [info] Server now running at http://127.0.0.1:1880/
27 Dec 13:52:06 - [info] Starting flows
27 Dec 13:52:07 - [info] Started flows
```

Figura 3.22: Inicio de Node-Red.

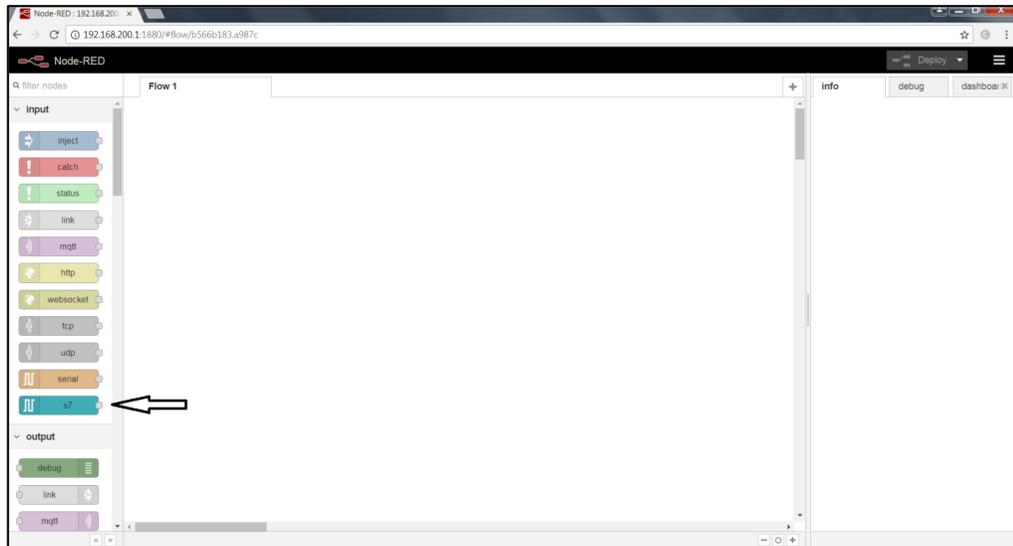


Figura 3.23: Página principal Node-Red.

Como se puede apreciar en la Figura 3.23, Node-Red ya posee varios nodos de programación, el nodo “S7”, que permite la conexión con dispositivos Siemens S7, se debe instalar ya que por defecto no se incluye. Para poder realizar esta instalación, se la puede hacer de dos formas, a través de la ventana de comandos de PuTTY o a través de la interfaz visual de Node-Red.

1. A través del terminal:

- Comando `cd /usr/lib/node_modules` (directorio donde se instalan los paquetes).
- Comando `npm install node-red-contrib-s7@2.0.2` (Instala los nodos input y output de S7, versión del paquete 2.0.2).

```

192.168.200.1 - PuTTY
[1] 2121
root@iot2000:~#

Welcome to Node-RED
=====

27 Dec 13:50:52 - [info] Node-RED version: v0.15.2
27 Dec 13:50:52 - [info] Node.js version: v6.9.2
27 Dec 13:50:52 - [info] Linux 4.4.18-yocto-standard ia32 LE
27 Dec 13:51:08 - [info] Loading palette nodes
27 Dec 13:52:04 - [info] Dashboard version 2.1.0 started at /ui
27 Dec 13:52:06 - [warn] -----
27 Dec 13:52:06 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
27 Dec 13:52:06 - [warn] -----
27 Dec 13:52:06 - [info] Settings file   : /home/root/.node-red/settings.js
27 Dec 13:52:06 - [info] User directory  : /home/root/.node-red
27 Dec 13:52:06 - [info] Flows file     : /home/root/.node-red/flows_iot2000.js
on
27 Dec 13:52:06 - [info] Creating new flow file
27 Dec 13:52:06 - [info] Server now running at http://127.0.0.1:1880/
27 Dec 13:52:06 - [info] Starting flows
27 Dec 13:52:07 - [info] Started flows
root@iot2000:~#
root@iot2000:~# cd /usr/lib/node_modules/
root@iot2000:/usr/lib/node_modules# npm install node-red-contrib-s7

```

Figura 3.24: Instalación de paquetes.

Esperar a que se instalen los paquetes, cuando se hayan instalado correctamente se procede a reiniciar el dispositivo utilizando el comando *sudo reboot*. Una vez reiniciado el dispositivo se inicia Node-Red nuevamente:

Comando *node /usr/lib/node_modules/node-red/red &* (inicio de Node-Red).

2. A través de Node-Red:

Dentro de la pantalla principal de Node-Red se realizan los siguientes pasos:

- Menú esquina superior derecha *Manage palette*, (ver Figura 3.25).

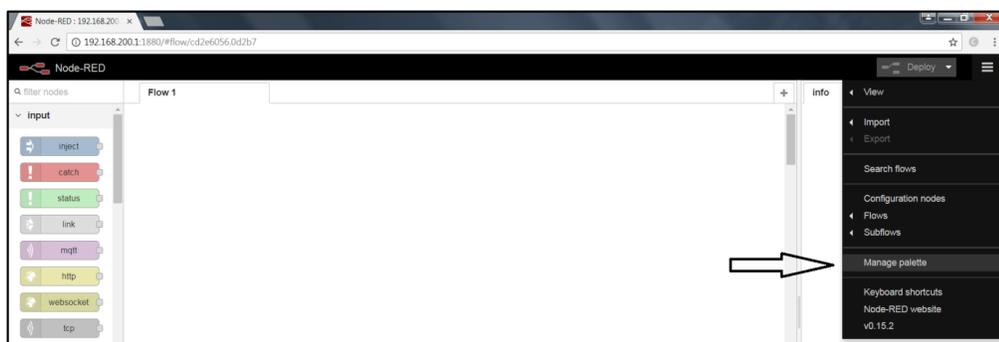


Figura 3.25: Selección de menú en Node-Red.

- Se busca `node-red-contrib-s7` y se presiona ***install***, (ver Figura 3.26).

En esta forma de instalación no es necesario reiniciar el dispositivo.



Figura 3.26: Instalación de nodo S7.

5. Programación en Node-Red (SIEMENS, 2018).

Dentro de la página principal de Node-Red se debe establecer la comunicación entre el módulo IoT con el PLC que esté conectado. Se utiliza el nodo S7 input para conectar y vincularse con los datos del PLC, como se muestra en la Figura 3.27.



Figura 3.27: Nodo S7 en Node-Red

Dentro del nodo S7 se configura la conexión con el PLC y la asignación de variables que se requieren comunicar. Al hacer doble clic se puede entrar a la configuración del nodo y editar configuración, (ver Figura 3.28):

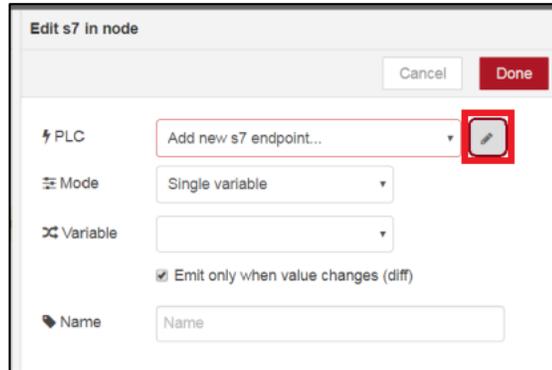


Figura 3.28: Configuración de Nodo S7.

Se despliega la ventana de configuración para la conexión del PLC, donde se encuentran las pestañas de Conexión y Variables.

En la pestaña de conexión (ver Figura 3.29), se introducen algunos parámetros: en el campo “Address” la dirección IP del PLC, el puerto se mantiene en el valor por defecto 102, El Rack y el Slot en donde se encuentra ubicado el PLC. Todos estos parámetros se observan en el Software TIA Portal, como se muestra en la Figura 3.30

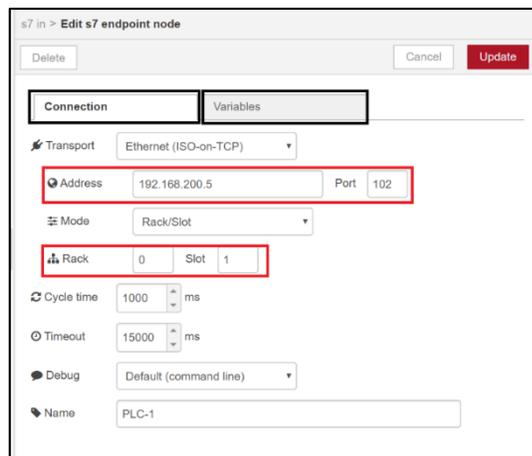


Figura 3.29: Configuración del nodo de conexión al PLC.

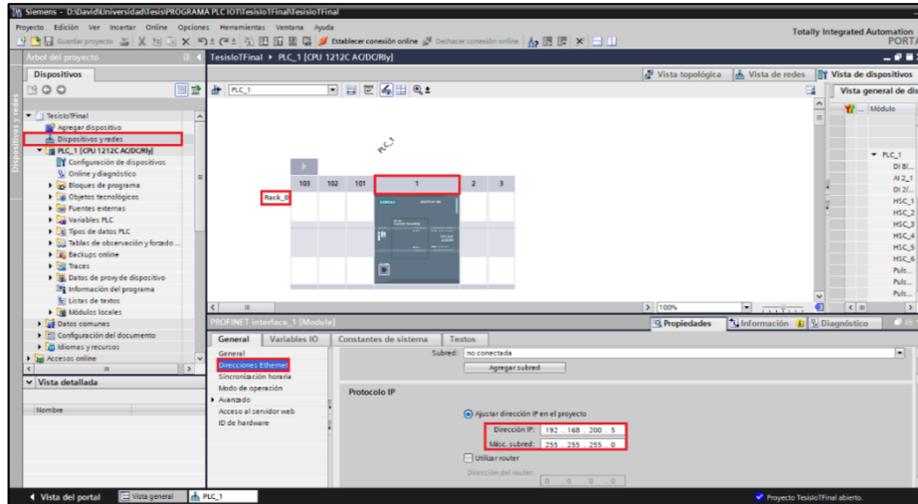


Figura 3.30: Configuración de red del PLC.

En la pestaña de variables se configuran las variables a leer.

Inicialmente en el programa del PLC se deben crear estas variables en un nuevo bloque de base de datos DB, (ver Figura 3.31).

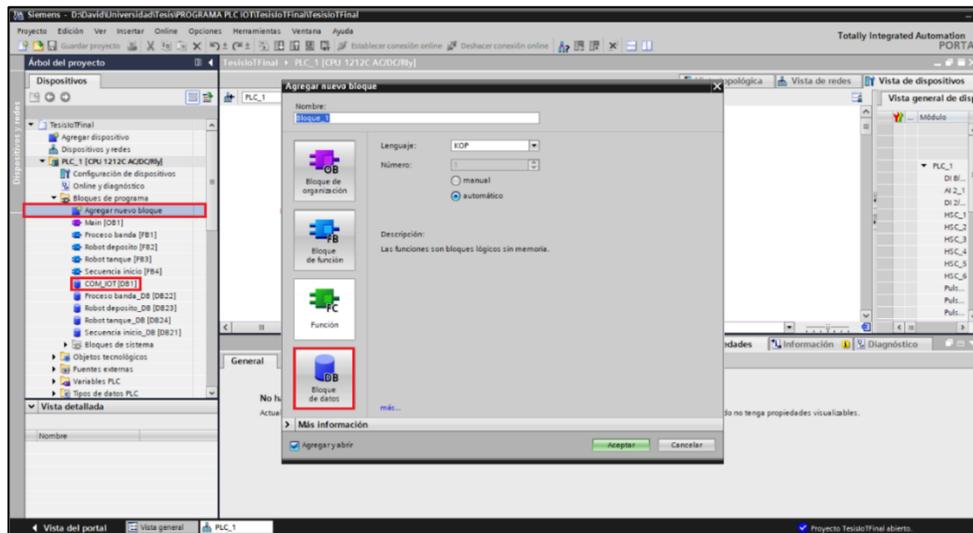


Figura 3.31: Creación de bloque de datos.

Dentro del bloque de base de datos creado en el programa del PLC (ver Figura 3.32), se guardan todas las variables que se necesiten leer o modificar desde un agente externo al PLC, en este caso desde la plataforma IoT y desde Node-Red.

Es necesario conocer el número de DB, el tipo de dato y la dirección asignada a cada variable, ya que esta información se utiliza en la página de Node-Red para la conexión de las variables.

Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..	Co...
Static								
contador_entrantes	Int	0.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
contador_metalicas	Int	2.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
contador_no_metalicas	Int	4.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
valor_sensor_nivel	Real	6.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
obstaculizacion_inicio_banda	Bool	10.0	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
obstaculizacion_horno	Bool	10.1	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
cajas_distribuidas	Int	12.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
falla_robot_pinza	Bool	14.0	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
nivel_tanque_alto	Bool	14.1	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
nivel_tanque_bajo	Bool	14.2	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
marcha	Bool	14.3	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
paro	Bool	14.4	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
valor_sensor_temperatura	Real	16.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
falla_robot_distribucion	Bool	20.0	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
tiempo_etapa_1	Real	22.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
tiempo_etapa_2_no_metalas	Real	26.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
tiempo_etapa_2_metalas	Real	30.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
nivel_tanque_porcentaje	Real	34.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
orden_marcha_remoto	Bool	38.0	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
orden_paro_remoto	Bool	38.1	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
activa_modos_manual	Bool	38.2	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
manual_encender_banda	Bool	38.3	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
manual_giro_base_robot_derecha	Bool	38.4	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
manual_giro_base_robot_izquierda	Bool	38.5	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
manual_baja_cuerpo_robot	Bool	38.6	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
manual_sube_cuerpo_robot	Bool	38.7	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
manual_cierra_pinza_robot	Bool	39.0	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
manual_abre_pinza_robot	Bool	39.1	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figura 3.32: Variables dentro del bloque DB.

Ahora, se selecciona la pestaña de variables en Node-Red y se añaden las variables que previamente se han implementado dentro del bloque de DB como se muestra en la Figura 3.33.

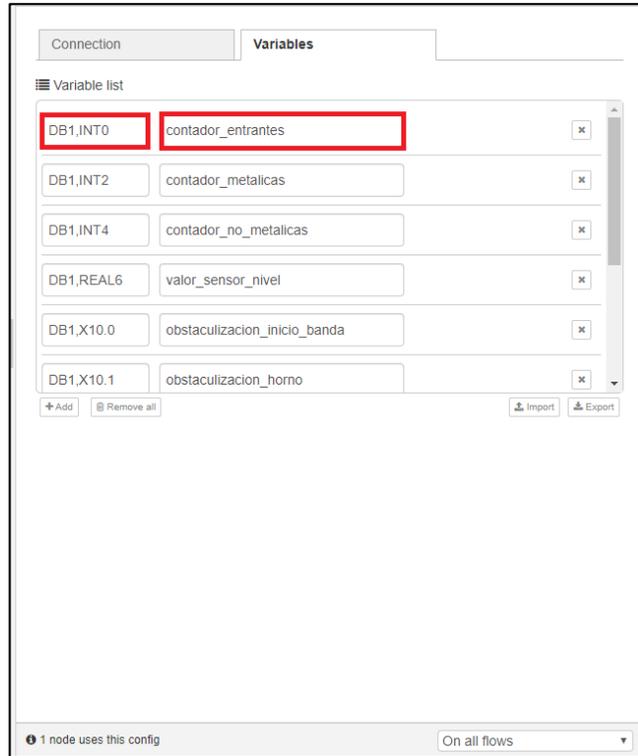


Figura 3.33: Tabla de variables en Node-Red.

Basándose en los datos del bloque de programación de la Figura 3.32, se introducen todas las variables, (ver Figura 3.34). En el campo “Name” ingresamos el nombre de la variable que va a manejar el programa Node-Red, preferiblemente asignamos las variables con el mismo nombre del bloque de programación del PLC. En el campo “Address” especificamos el número de DB, el tipo de dato y la dirección, en base a la Tabla 3.1, guiándose de la siguiente manera como se muestra en las Figuras Figura 3.35 y Figura 3.36.



Figura 3.34: Editores de texto para ingreso de variables en Node-Red.



Figura 3.35: Modelo de introducción de variables.

- DB1: número de bloque DB.
- INT: tipo de dato.
- 0: offset.



Figura 3.36: Modelo de introducción de variables.

- DB1: número de bloque DB.
- X: tipo de dato.
- 10.0: offset.

Tabla 3.1: Tabla de direccionamiento para Node-Red.

DIRECCION NODE RED	TIA PORTAL	TIPO DE DATO	DESCRIPCION
DB5,X0.1	DB5.DBX0.1	Booleano	Bit 1 del byte 0 de la DB 5
DB23,B1 or DB23,BYTE1	DB23.DBB1	Número	Byte 1 (0-255) de la DB 23
DB100,C2 or DB100,CHAR2	DB100.DBB2	String	Byte 2 de la DB 100 como Char
DB42,I3 or DB42,INT3	DB42.DBW3	Número	Número con signo de 16-bit en el byte 3 de la DB 42
DB57,WORD4	DB57.DBW4	Número	Número con signo de 16-bit en el byte 4 de la DB 57
DB13,DI5 or DB13,DINT5	DB13.DBD5	Número	Número con signo de 32-bit en el byte 5 de la DB 13

DB19,DW6 or DB19,DWORD6	DB19.DBD6	Número	Número sin signo de 32-bit en el byte 6 de la DB 19
DB21,DR7 or DB21,REAL7	DB19.DBD7	Número	Número de punto flotante de 32-bit en el byte 7 de la DB 21
DB2,S7.10*	-	String	String de longitud 10 comenzando en el byte 7 de la DB 2.
I1.0 or E1.0	I1.0 or E1.0	Booleano	Bit 0 del byte 1 del área de entrada.
Q2.1 or A2.1	Q2.1 or A2.1	Booleano	Bit 1 del byte 2 del área de salida.
M3.2	QM3.2	Booleano	Bit 2 del byte 3 del área de memoria.
IB4 or EB4	IB4 or EB4	Número	Byte 4 (0 -255) del área de entrada
QB5 or AB5	QB5 or AB5	Número	Byte 5 (0 -255) del área de salida
MB6	MB6	Número	Byte 6 (0 -255) del área de memoria.
IC7 or EC7	IB7 or EB7	String	Byte 7 del área de entrada del tipo Char
QC8 or AC8	QB8 or AB8	String	Byte 8 del área de salida del tipo Char
MC9	MB9	String	Byte 9 del área de memoria del tipo Char
I10 or EI10	IW10 or EW10	Número	Número con signo de 16-bit en byte 10 del área de entrada
QI12 or AI12	QW12 or AW12	Número	Número con signo de 16-bit en el byte 12 del área de salida
MI14	MW14	Número	Número con signo de 16-bit en el byte 14 del área de memoria
IW16 or EW16	IW16 or EW16	Número	Número sin signo de 16-bit en el byte 16 del área de entrada
QW18 or AW18	QW18 or AW18	Número	Número sin signo de 16-bit en el byte 18 del área de salida

MW20	MW20	Número	Número sin signo de 16-bit en el byte 20 del área de memoria
IDI22 or EDI22	ID22 or ED22	Número	Número con signo de 32-bit en el byte 22 del área de ingreso
QDI24 or ADI24	QD24 or AD24	Número	Número con signo de 32-bit en el byte 24 del área de salida
MDI26	MD26	Número	Número con signo de 32-bit en el byte 26 del área de memoria
ID28 or ED28	ID28 or ED28	Número	Número sin signo de 32-bit en el byte 28 del área de entrada
QD30 or AD30	QD30 or AD30	Número	Número sin signo 32-bit en el byte 30 de área de salida
MD32	MD32	Número	Número sin signo de 32-bit en el byte 32 del área de memoria
IR34 or ER34	IR34 or ER34	Número	Número de punto flotante de 32-bit en el byte 34 en el área de ingreso
QR36 or AR36	QR36 or AR36	Número	Número de punto flotante de 32-bit en el byte 36 en el área de salida.
MR38	MR38	Número	Número de punto flotante de 32-bit en el byte 38 del área de memoria

Fuente: (Cittolin, 2016-2019)

Una vez que se añadan todas las variables, clic en “Update”, (ver Figura 3.37).

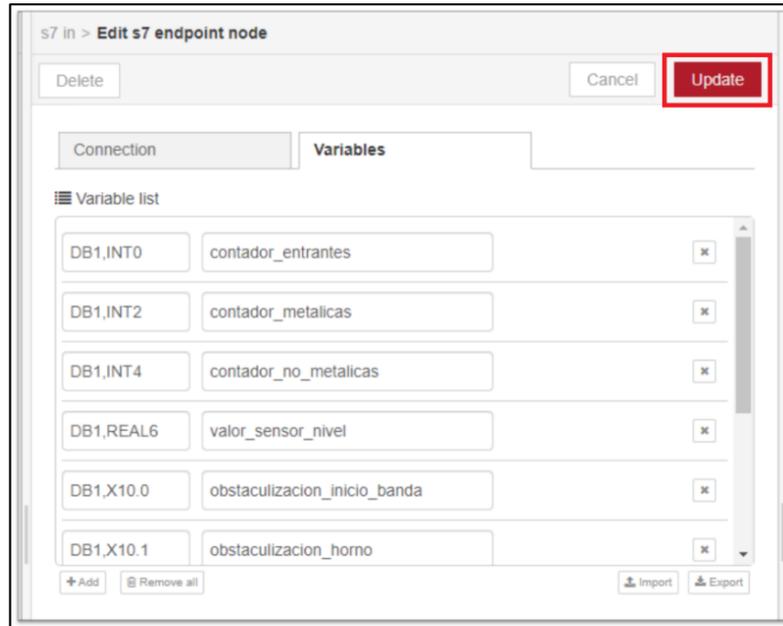


Figura 3.37: Variables ingresadas en Node-Red.

Ahora en el campo “Mode” seleccionar “All variables”, y finalmente clic en *Done*, (ver Figura 3.38).

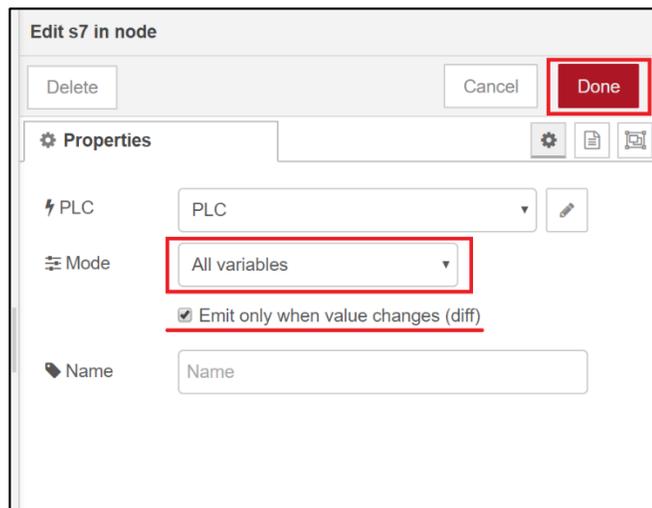


Figura 3.38: Configuración de nodo s7.

3.3.3 Nodos de programación en Node-Red

Se describe a continuación un ejemplo de programación en Node-Red para el envío de un correo electrónico que notifica el estado de las variables del PLC, cuando estas superan un límite establecido.

Nodo Function



Figura 3.39: Nodo Function.

Este nodo, ver Figura 3.39, se programa en JavaScript, en este caso se crea una función sencilla para comprobar cuantas cajas han pasado por la banda transportadora del proceso del banco didáctico (sección 0), en el momento que pasen 5 cajas se habilitará una notificación. También se definirá el texto de la notificación.

Arrastrar el nodo *function* a la mesa de trabajo (*workspace*) y hacer doble clic en el mismo. En el campo “Name” se asigna un nombre a la función para identificar la misma en el *workspace*, este nombre solo es informativo. En el cuerpo de la función copiar el siguiente texto:

```

let limiteContador = 5;
if (msg.payload.contador_entrantes == limiteContador){
  msg.topic = "Cajas entrantes";
  msg.payload = "Le informamos que ya se han contado " + limiteContador + " cajas.
y los valores de los sensores son los siguientes:\n\n Temperatura Horno: " +
msg.payload.valor_sensor_temperatura + " °C \n Nivel Tanque: " +
msg.payload.valor_sensor_nivel + " cm \n\n Un saludo, \n PLC.";
  if(global.notificar === null){
    global.notificar = 1;
  }else{
    global.notificar = 0;
  }
} else {
  global.notificar = null;
}
msg.notificar = global.notificar;
return msg;

```

La función comprueba el número de cajas entrantes en el proceso, cuando este sea igual a 5 (variable `limiteContador`), asigna la variable `global.notificar` a 1. En caso de que las cajas entrantes no sean 5, la variable `notificar` se asigna a nulo (`null`). Esto permitirá más adelante diferenciar en qué momento se debe notificar al usuario para, por ejemplo, enviar un email. El campo `topic` se agrega como variable al mensaje (variable `msg`) para luego utilizarlo como asunto del correo electrónico. El cuerpo del mensaje se escribe en la variable `payload`.

Se maneja una variable global, `global.notificar`, para guardar el estado de la misma sin importar cuantas veces se ejecute el flujo. Además, se le asignan 3 estados: `null`, `0` y `1`, para asegurar enviar un solo correo electrónico. Cuando el número de cajas sea igual a 5, el valor de la variable `global.notificar` será 1; en este momento el flujo enviará el correo electrónico. Luego el flujo se repetirá, pero a pesar de que el número de cajas sea igual a 5, la variable `global.notificar` cambiará de 1 a 0, validando que ya no se reenvíe el mismo correo otra vez.

Finalmente se asigna el valor de la variable global, `global.notificar`, al mensaje (`msg.notificar`), lo que permitirá en un futuro decidir el camino del flujo con el nodo `switch`.

Nodo Switch

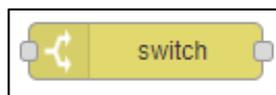


Figura 3.40: Nodo switch.

Este nodo, ver Figura 3.40, permite definir distintos caminos del flujo en función de alguna variable. Se utilizará en este caso para definir el flujo en función de la variable `msg.notificar`. En el caso que `notificar` sea 1 se redirigirá el flujo para que se envíe un correo con el nodo `mail`, y en el caso que sea 0 u otro valor distinto de 1, se imprimirá un mensaje con el nodo `debug`, el cual se indica más adelante.

Se realiza la configuración del nodo switch como se indica en la Figura 3.41, en el campo *property* se asigna el valor de *msg.notificar*. Ahora se agregan dos ramas con el botón *+add*. En la primera rama se cambia el tipo de dato de string a number y se asigna el valor de 1. En la segunda rama se selecciona el operador *otherwise*, que significa caso contrario. Finalmente, clic en Done.

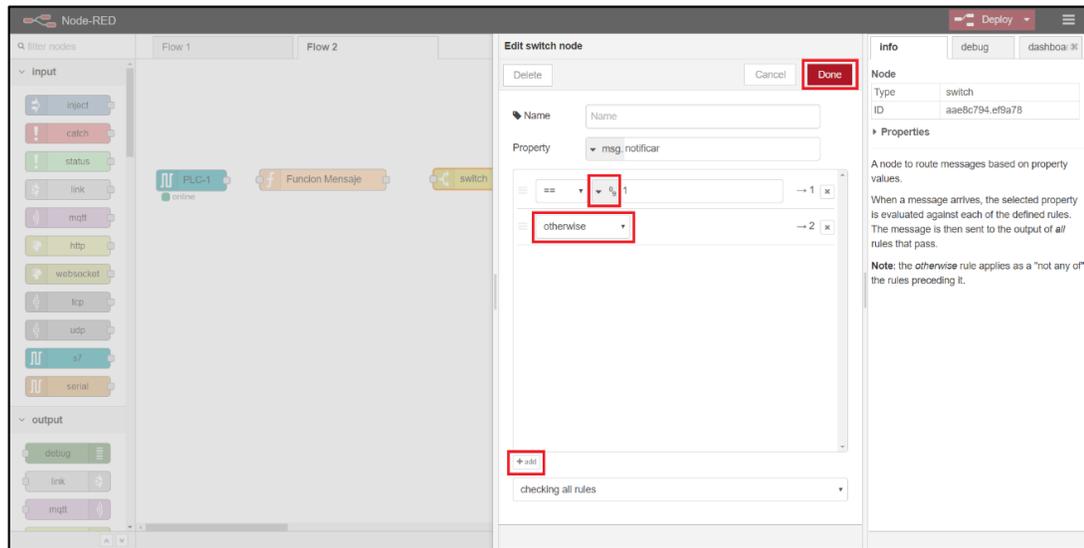


Figura 3.41: Configuración nodo Switch.

Nodo Debug



Figura 3.42: Nodo debug.

Este nodo, ver Figura 3.42, permite mostrar mensajes en la pestaña debug de la barra lateral o en la ventana de comandos. Se utilizará este nodo para mostrar todas las variables del flujo y monitorizar la producción en el caso que no haya que notificar por email al usuario.

Hacer doble clic en el nodo para acceder a su configuración, se muestra la ventana de la Figura 3.43, en *output* seleccionar *complete msg object*, y en *to* seleccionar *debug tab* ó *debug tab and console*, dependiendo si se requiere observar los mensajes en la columna *debug* o en la ventana de comandos respectivamente.

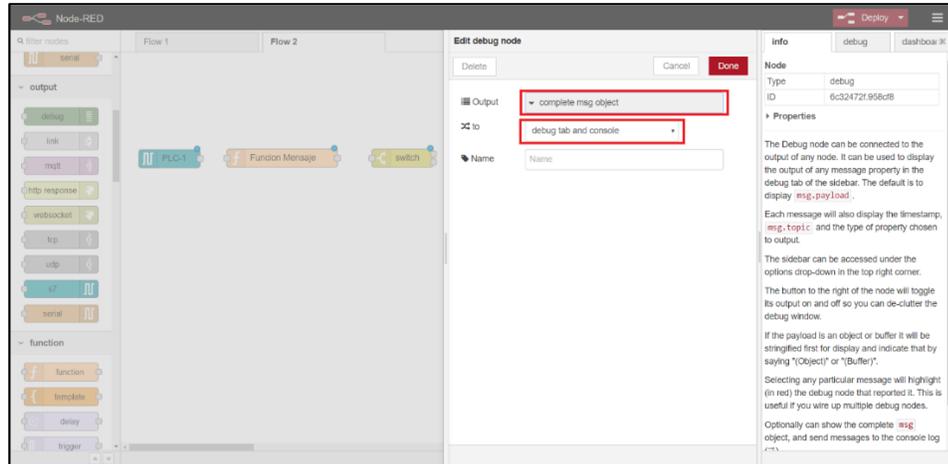


Figura 3.43: Configuración nodo debug.

Nodo Email



Figura 3.44: Nodo Email.

La configuración de este nodo, ver Figura 3.44, es bastante sencilla. Se deben rellenar los campos de destinatario y remitente. En el campo del destinatario se debe introducir una dirección de correo válida. En el campo remitente, node-red necesita conocer el servidor de correo electrónico que va a utilizar para enviar el correo y el puerto del mismo, en el caso de Gmail esta configuración ya se encuentra predeterminada. Finalmente se debe indicar el usuario y contraseña del correo electrónico del remitente de manera que el programa pueda acceder al mismo y pueda enviar el correo, como se muestra en la Figura 3.45.

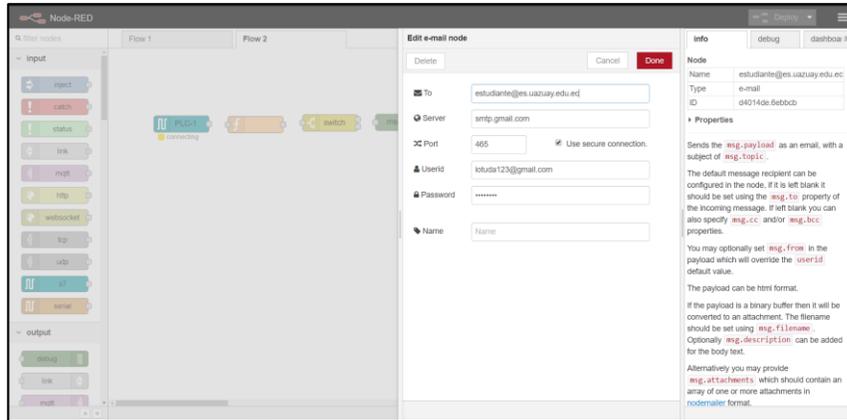


Figura 3.45: Configuración del nodo Email.

En el caso de utilizar Gmail, se requiere dar permiso en la cuenta del remitente para que esta aplicación pueda acceder automáticamente a enviar correos. Esta opción se encuentra en los ajustes de la cuenta de Gmail → *Aplicaciones con acceso a la cuenta* → *Permitir el acceso de aplicaciones menos seguras*, (ver Figura 3.46). Cuando se habilita este acceso a través de estas aplicaciones, Gmail informa a la cuenta del remitente que se ha habilitado.

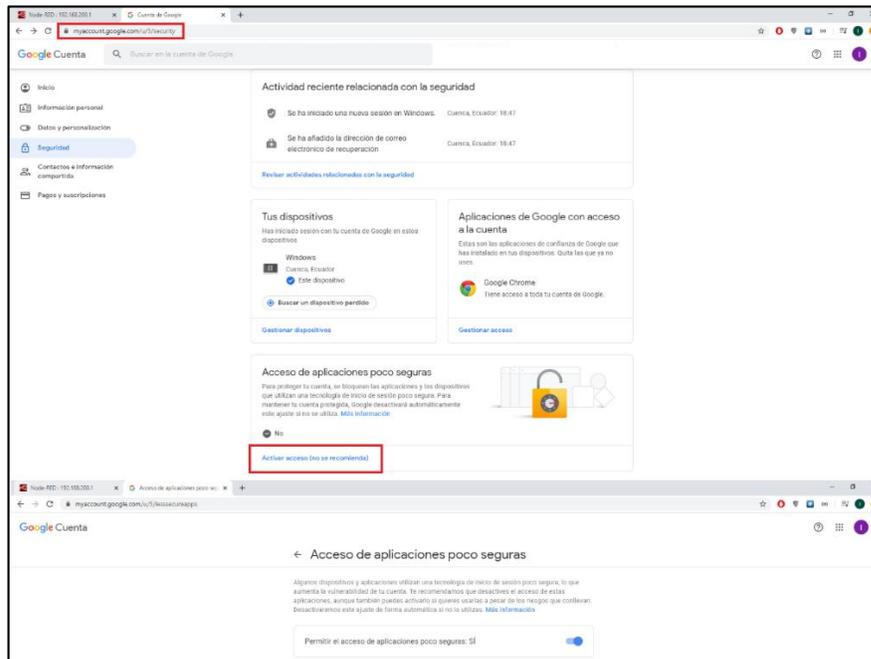


Figura 3.46: Configuración de correo cuenta Gmail.

3.3.4 Conexión de nodos y envío de correo electrónico

Finalmente, se conectan los nodos para finalizar el programa de Node-Red. Como se muestra en la Figura 3.47.

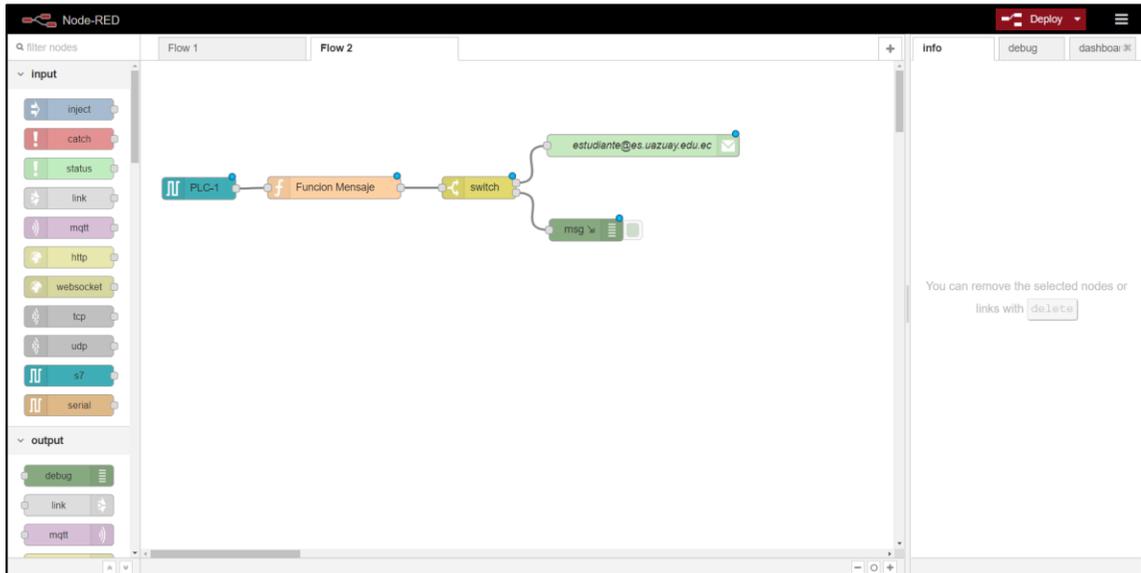


Figura 3.47: Conexión de nodos en el programa node-red.

Una vez conectados, se pulsa el botón “Deploy” en la esquina superior derecha de la pantalla para compilar el programa. Si todos los pasos se siguieron correctamente la compilación se debe ejecutar satisfactoriamente.

Importante: Es preferible conectar el módulo IoT a la misma red del PLC S7-1200 a través de un switch, lo cual permite establecer una conexión exitosa con el PLC utilizando el nodo s7.

Envío de e-mail

Finalmente, para verificar el funcionamiento correcto del programa, al ejecutarse se mostrarán los mensajes de prueba en la pestaña de debug de la barra lateral derecha. Como se muestra en la Figura 3.48.

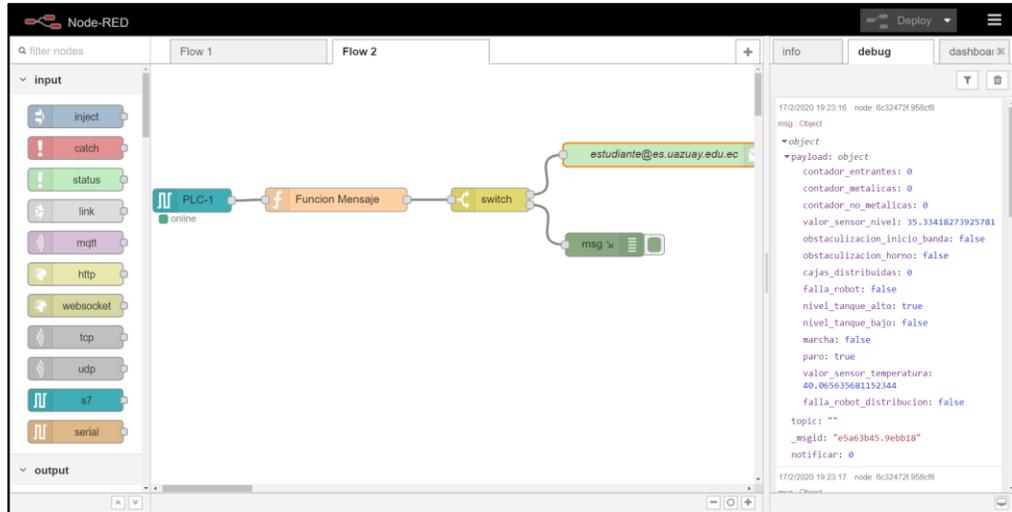


Figura 3.48: Mensajes de prueba en la pestaña debug.

Una vez pasen 5 cajas por la banda transportadora del banco didáctico, el programa Node-Red enviará el correo al destinatario como se había programado, (ver Figura 3.49).

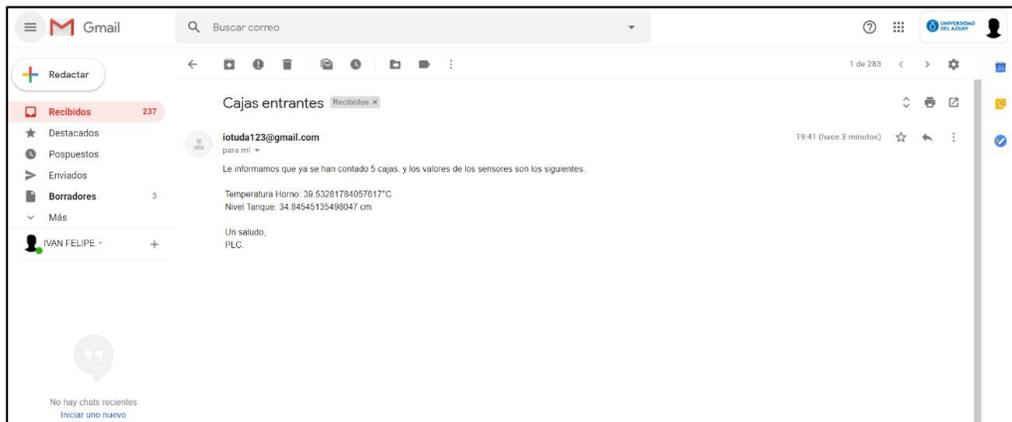


Figura 3.49: Correo recibido en el correo del destinatario.

Node-Red Dashboard.

Es una interfaz gráfica implementada en la librería Node-Red para visualizar diferentes tipos de variables en una variedad de widgets que pueden ser personalizados.

Node-Red Dashboard se encuentra como servidor web del módulo IoT en la dirección 192.168.200.1:1880/ui/

En la paleta se encuentran nodos widget de tipo input/output, así como también nodos widget de tipo output, (ver Figura 3.50).

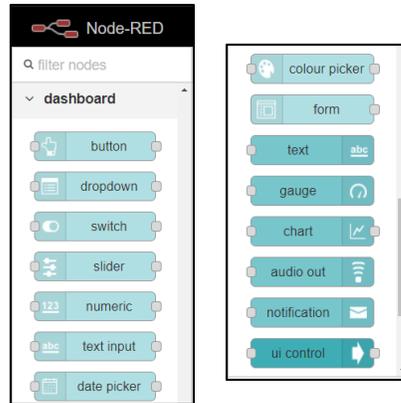


Figura 3.50: Nodos Dashboard.

3.3.5 Inicio Automático de Node-Red

Para que Node-Red se inicie automáticamente al encender o reiniciar el módulo IoT, ingresar sesión al módulo IoT con PuTTY y utilizar el comando:

- Comando *iot2000setup*

Aparecerá la ventana que se muestra en la Figura 3.51. Seleccionar la opción Software con las flechas y presionar enter.

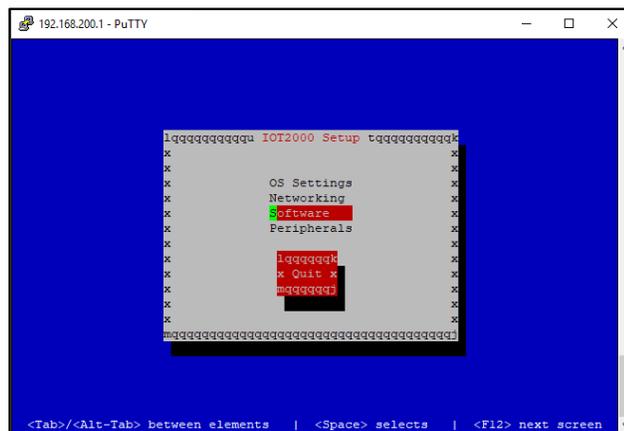


Figura 3.51: Ventana IOT2000 Setup.

Luego se selecciona la opción *Manage Autostart Options*, (ver Figura 3.52).

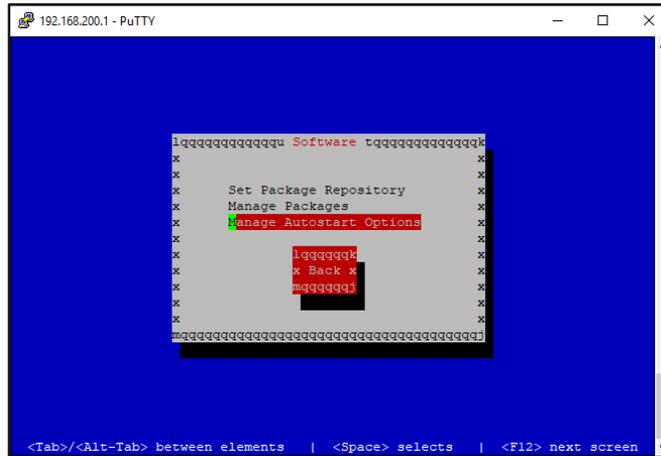


Figura 3.52: Ventana IOT2000 Setup > Software.

Se marca la opción *Auto Start node-red* pulsando enter y se presiona en *Done*, para esto utilizar la flecha derecha que marca el botón *Done* y presionar enter, (ver Figura 3.53). Luego se presiona el botón Escape hasta salir de la ventana de configuración a la terminal de comandos.

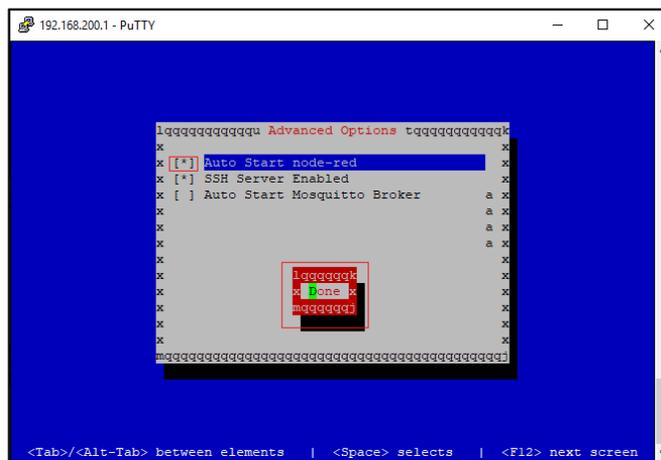


Figura 3.53: Ventana IOT2000 Setup > Software > Manage Autostart Options.

Ahora para reiniciar el módulo IoT desde la terminal de comando utilizar el comando **reboot**.

CAPÍTULO 4: RESULTADOS

4.1 Conexión del banco didáctico.

Para comprobar el funcionamiento del sistema del proceso de manufactura implementado en el banco didáctico (ver Figura 4.1), se realizaron las conexiones de todos los elementos de acuerdo al esquema que se encuentra en el Anexo 3: Diagrama de conexión del banco didáctico.

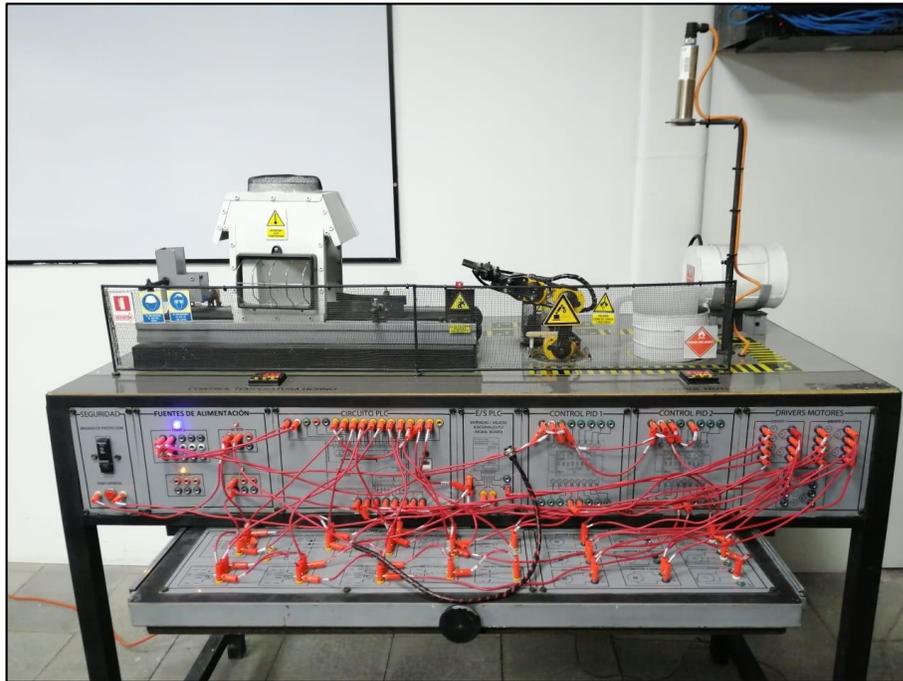


Figura 4.1 Conexiones físicas del banco didáctico.

4.2 Programación del PLC.

Se realizó la programación del PLC ubicado en el banco didáctico utilizando el software TIA PORTAL como se describe en la sección 3.1. Los bloques de programación se encuentran en la sección de Anexos como Anexo 4: Código implementado en el PLC s7-1200 1212c AC/DC/RLY.

4.3 Programación del módulo IoT 2040.

Como se indica en la sección 3.3, se realizó la instalación del sistema operativo en el módulo IoT, así como su configuración inicial, configuración de red e instalación de los paquetes de Node-Red para la comunicación con el PLC.

Una vez completada la configuración en el módulo IoT, se realizó la programación de dos flujos en Node-Red, en donde el flujo 1 se encarga del envío de los datos generados por el PLC hacia la nube (ver Figura 4.2), y el flujo 2 se encarga del control manual del banco desde la nube, (ver Figura 4.3 y Figura 4.4).

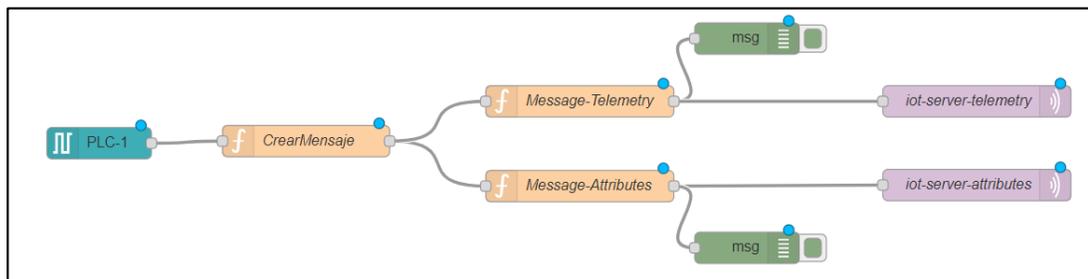


Figura 4.2: Flujo 1, envío de datos a la nube.

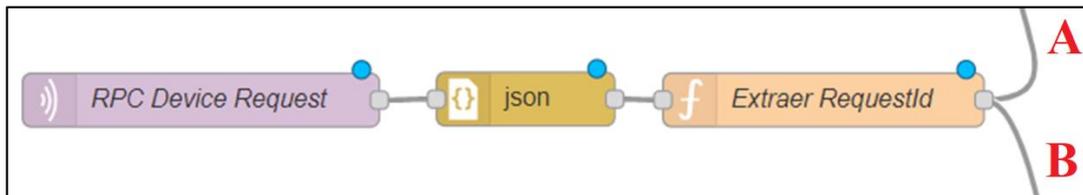


Figura 4.3: Inicio del flujo 2, control del banco desde la nube.

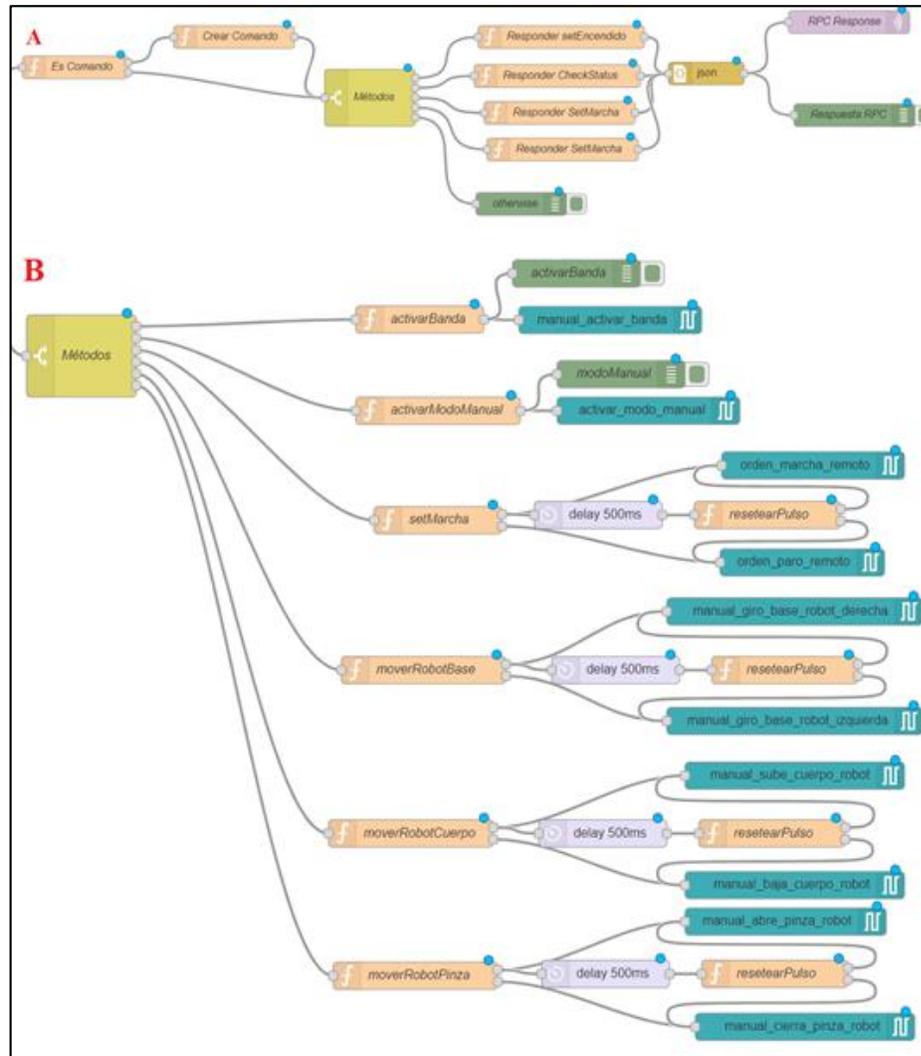


Figura 4.4: Flujo 2, control del banco desde la nube, ramificaciones.

4.4 Instalación y configuración de los servidores en la nube

Se realizaron las pruebas de la plataforma ThingsBoard en el servidor <https://demo.thingsboard.io/>, en esta página se puede utilizar la versión gratuita de la misma sin necesidad de instalarse en un servidor. Puede utilizarse este servidor para realizar pruebas de desarrollo, solo se necesita crear una cuenta para acceder al sistema.

Una vez desarrolladas las pruebas para evaluar las características del sistema se decide instalar la plataforma ThingsBoard en un servidor propio. Para esto se cotizaron los precios y funcionalidades de las distintas empresas que ofrecen productos y servicios en la nube, para finalmente elegir la nube de Google Cloud, descrito en la sección 2.3.

En Google Cloud se crea una cuenta de desarrollador con una cuenta de Gmail y se accede al crédito de 300\$ por un periodo de un año, a partir de este momento se desarrollaron las pruebas respectivas con la instalación y manejo de las máquinas virtuales. Se instalaron dos máquinas virtuales con el servicio de Compute Engine de Google Cloud con las siguientes características, (ver Figura 4.5):

Máquina 1:

- Nombre de la instancia: iot-uda-prd.
- Tipo de máquina: n1-standard-1.
- CPU: 1 vCPU.
- Memoria RAM: 3.75 GB.
- Sistema operativo: Ubuntu Server 18.04 LTS.
- Zona: us-east1-b.
- Disco Duro: 10 GB.

Máquina 2:

- Nombre de la instancia: iot-uda-qas.
- Tipo de máquina: n1-standard-1.
- CPU: 1 vCPU.
- Memoria RAM: 3.75 GB.
- Sistema operativo: Ubuntu Server 18.04 LTS.
- Zona: us-central1-a.
- Disco Duro: 10 GB.

La máquina 1 con nombre iot-uda-prd se utiliza para instalar la versión Premium de la plataforma ThingsBoard, siendo este el ambiente de producción del sistema. Mientras que la máquina 2 con nombre iot-uda-qas se utiliza para instalar la versión gratuita de código abierto de la plataforma ThingsBoard, también llamada Community Edition.

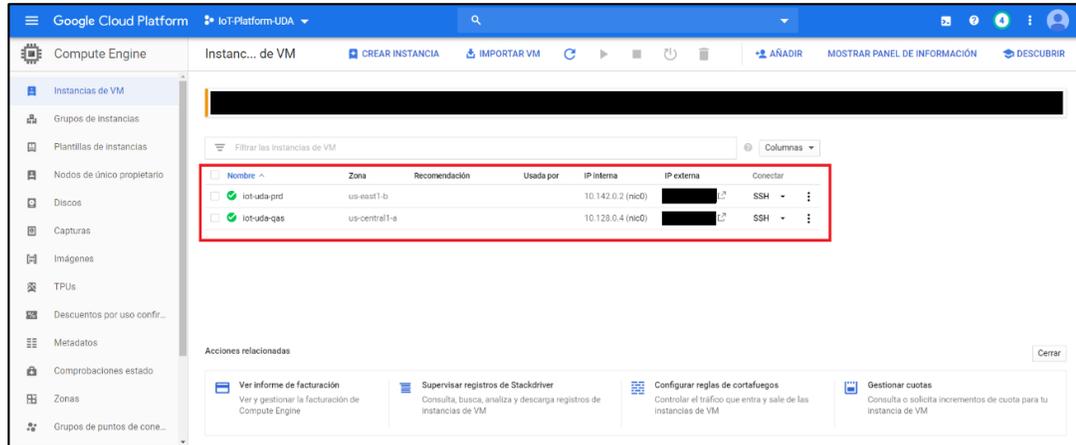


Figura 4.5: Máquinas virtuales implementadas.

4.4.1 Configuración de IP Estática para las Máquinas Virtuales en Google Cloud

Se crearon las máquinas virtuales en distintas zonas para tener la posibilidad de asignar una IP estática a cada una. En la consola de *Google Cloud* → *Red de VPC* → *Direcciones IP externas* se configuraron las IP's estáticas para cada instancia *iot-uda-prd* e *iot-uda-qas*, como se aprecia en la Figura 4.6. Configurar una IP estática permite asignar un dominio a la dirección de red.

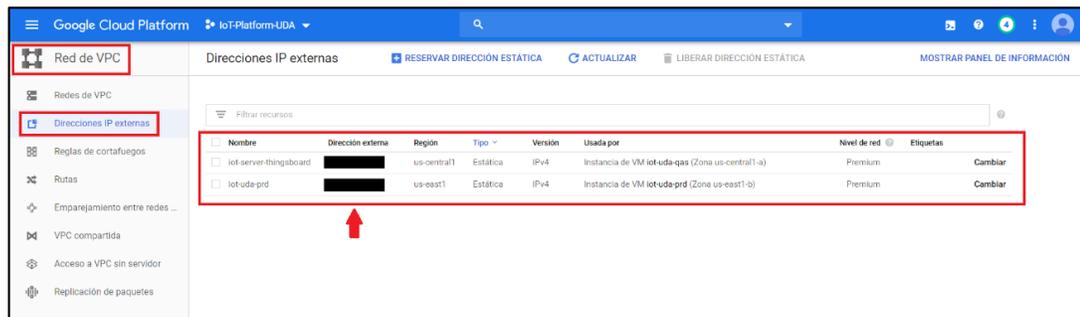


Figura 4.6: Configuración de direcciones IP externas estáticas.

4.4.2 Configuración de Firewall en Google Cloud

Se realiza la configuración del Firewall en la interfaz de red de las máquinas virtuales, esto se puede configurar en la consola de *Google Cloud* → *Red de VPC* → *Reglas de cortafuegos* en donde se agregan reglas personalizadas para permitir el tráfico en los puertos tal como se indica en la Figura 4.7.

- MQTT: TCP-1883
- ThingsBoard Web UI: TCP-8080
- CoAP: UDP-5683

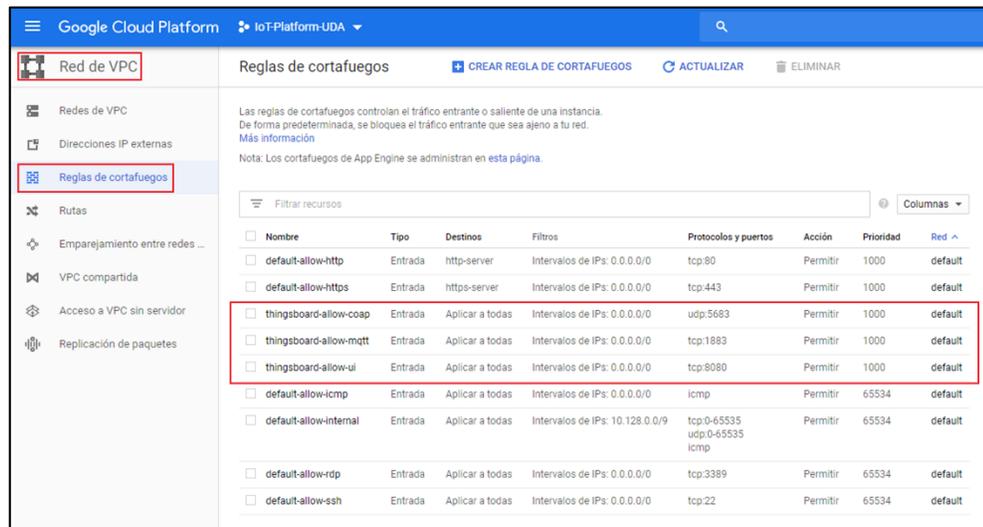


Figura 4.7: Configuración de puertos en el firewall de Google Cloud.

4.4.3 Instalación de la Plataforma ThingsBoard Community Edition

Se realiza la instalación de la plataforma ThingsBoard Community Edition en la máquina virtual 2 iot-uda-qas como implementación monolítica. El usuario debe conectarse a través de SSH a la máquina virtual e instalar la plataforma ThingsBoard para sistemas Ubuntu como indica su documentación <https://thingsboard.io/docs/user-guide/install/ubuntu/>.

4.4.4 Instalación de la Plataforma ThingsBoard Professional Edition

Se realiza la instalación de la plataforma ThingsBoard Professional Edition en la máquina virtual 1 iot-uda-prd como implementación monolítica. El usuario debe conectarse a través de SSH a la máquina virtual e instalar la plataforma ThingsBoard para sistemas Ubuntu como indica su documentación <https://thingsboard.io/docs/user-guide/install/pe/ubuntu/>. Es importante saber que para instalar la versión Professional Edition de la plataforma se debe comprar la licencia correspondiente. Una vez el usuario obtenga la licencia puede continuar con los pasos de instalación como indica la documentación oficial.

4.5 Plataforma ThingsBoard

Una vez instalada la plataforma ThingsBoard en el servidor, por defecto cuenta con varios ejemplos prácticos de uso a manera de demostración. Los usuarios predeterminados son los siguientes:

- **Administrador del sistema (System Administrator)**
 - **Usuario:** sysadmin@thingsboard.org
 - **Contraseña:** sysadmin
- **Usuario Inquilino (Tenant Administrator)**
 - **Usuario:** tenant@thingsboard.org
 - **Contraseña:** tenant
- **Usuario Cliente (Customer User)**
 - **Usuario:** customer@thingsboard.org
 - **Contraseña:** customer

Para el ingreso a la plataforma IoT se ingresa con el usuario predeterminado Inquilino, de la siguiente manera, como se aprecia en la Figura 4.8:

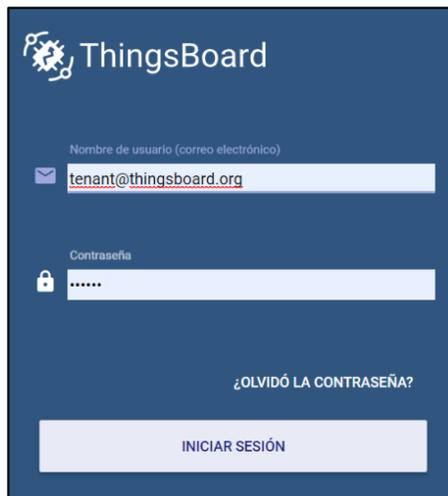


Figura 4.8: Ingreso a la plataforma IoT utilizando las credenciales predeterminadas.

Al iniciar sesión se aprecia la página principal de la Figura 4.9:



Figura 4.9: Página principal ThingsBoard.

Dentro de la página principal presentada anteriormente, para dirigirse a la ventana donde se encuentran los datos y graficas se busca la pestaña de paneles, (ver Figura 4.10).

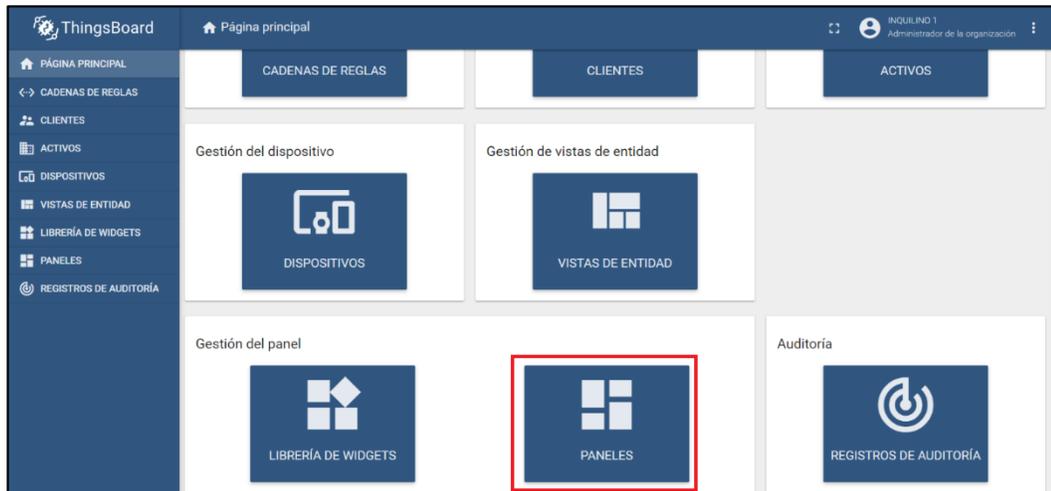


Figura 4.10: Ventana de paneles.

Al ingresar a la ventana de paneles, se encuentran los diversos paneles descritos en el proyecto de tesis, el panel dedicado al banco didáctico se lo encuentra como “Banco didáctico final”, (ver Figura 4.11).

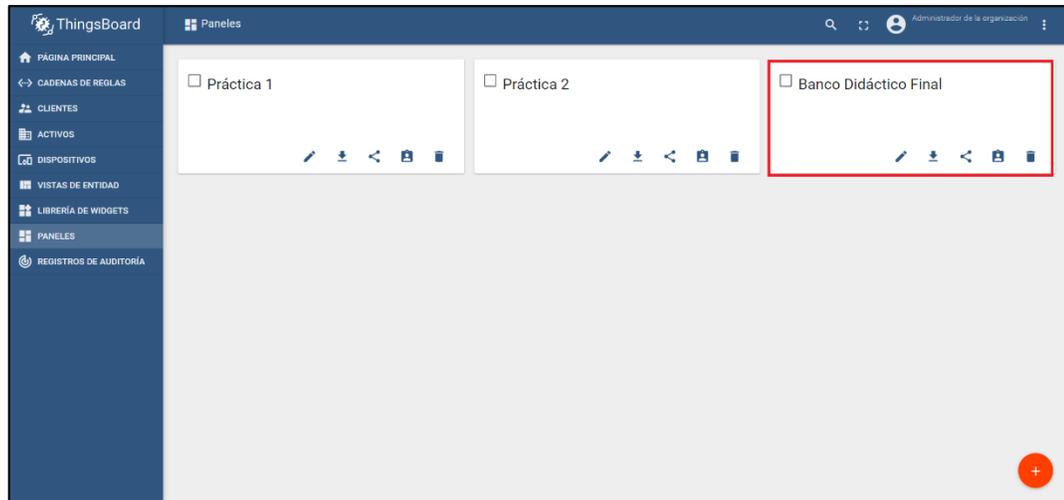


Figura 4.11: Panel Banco Didáctico Final.

Dentro del panel del banco didáctico se encuentran las pestañas de navegación, que se muestran en la Figura 4.12.



Figura 4.12: Pestañas visibles en el panel del banco didáctico.

En donde:

- Overview: muestra una pantalla general, en la que se observan los datos más importantes provenientes del banco.
- Alarmas: muestra una pantalla con el listado de alarmas que se han generado durante todo el proceso del banco didáctico.
- Gráficos: muestra datos visibles en graficas que se actualizan en tiempo real, con la opción de buscar el dato requerido en cualquier tiempo en el que el banco didáctico estuvo activo.
- Control: muestra los controles de los actuadores del banco en modo manual, adicionalmente se encuentran pulsantes de marcha y paro remoto para el control del banco en general.

4.5.1 Pestaña OVERVIEW.

Al hacer clic sobre la pestaña, se abre la siguiente página general, como se aprecia en la Figura 4.13.

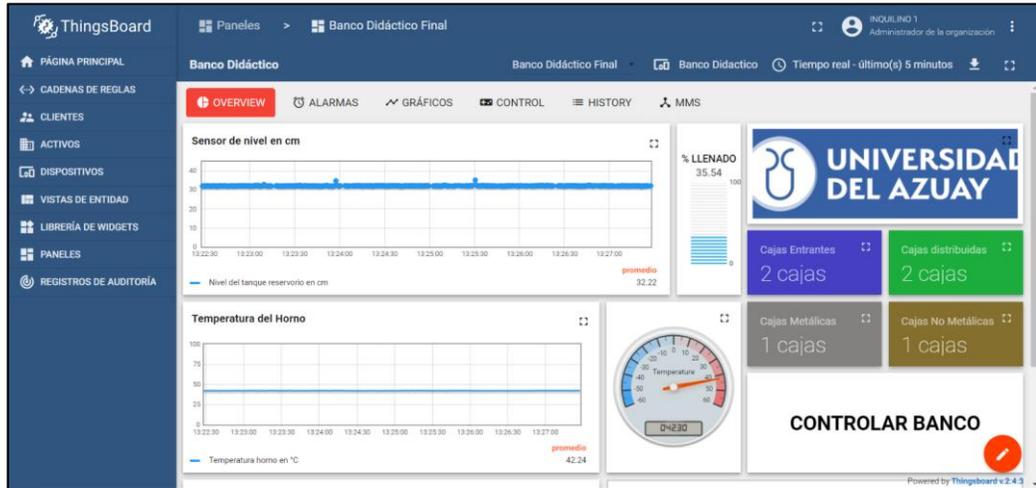


Figura 4.13: Pestaña Overview.

La pestaña Overview contiene los siguientes elementos especificados en la Figura 4.14.

- Valor del sensor de nivel en serie temporal.
- Valor de sensor de nivel en dato instantáneo.
- Valor del sensor de temperatura en serie temporal.
- Valor del sensor de la temperatura en dato instantáneo.
- Contadores de cajas ingresadas, metálicas, no metálicas y distribuidas.
- Distribución de cajas por lote en serie temporal.
- Control manual del banco.
- Tiempos de actividad.

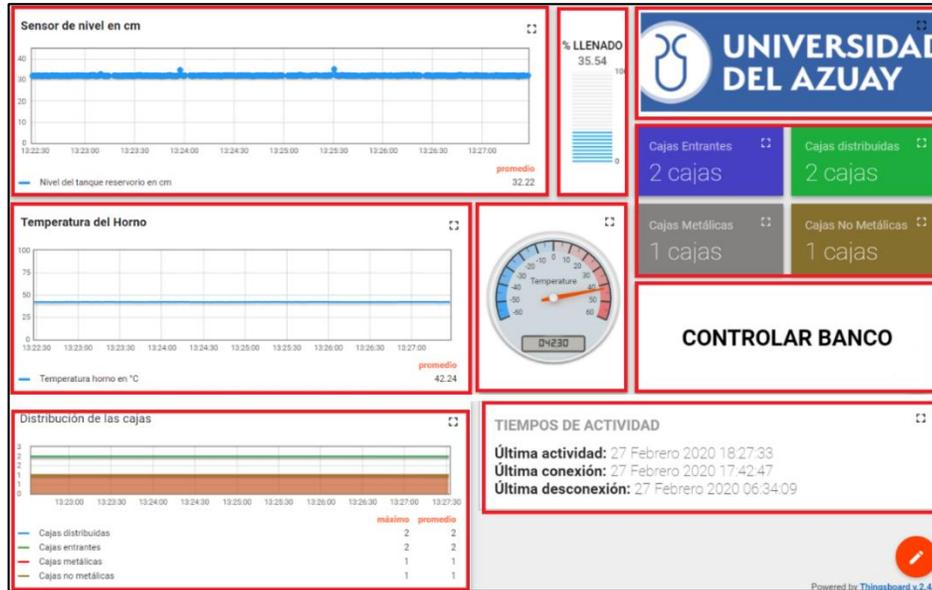


Figura 4.14: Elementos de la pestaña Overview.

4.5.2 Pestaña ALARMAS.

Al ingresar a la pestaña Alarmas, se abre la siguiente página principal, la misma que se observa en la Figura 4.15

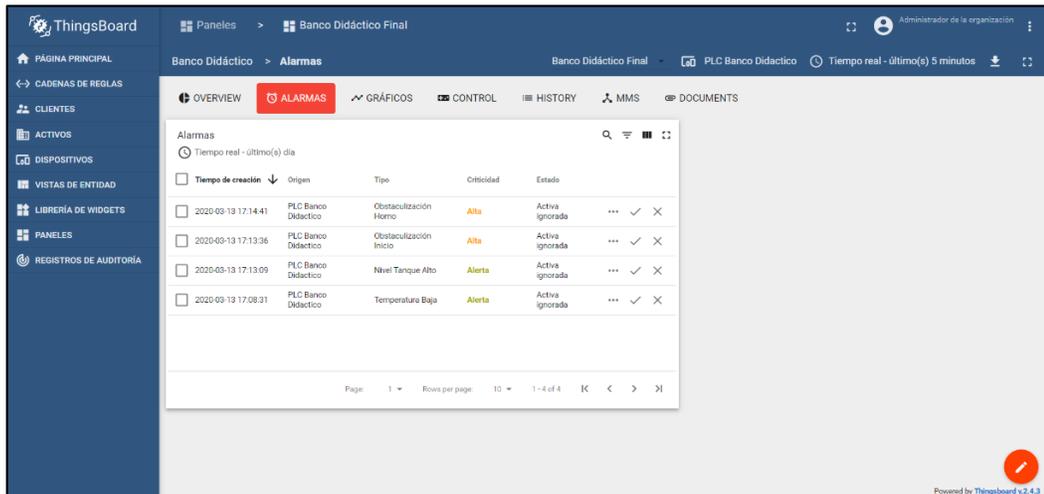


Figura 4.15: Pestaña Alarmas.

La pestaña de alarmas posee un visualizador en tipo listado para presentar las alarmas generadas a lo largo del proceso llevado por el banco didáctico, en esta lista se visualiza

la hora de ocurrencia de la alarma, el nodo de origen, el tipo de alarma, la criticidad y el estado actual de la alarma, (ver Figura 4.16).

<input type="checkbox"/>	Tiempo de creación	Origen	Tipo	Criticidad	Estado			
<input type="checkbox"/>	2020-03-13 17:14:41	PLC Banco Didactico	Obstaculización Horno	Alta	Activa ignorada	...	✓	✕
<input type="checkbox"/>	2020-03-13 17:13:36	PLC Banco Didactico	Obstaculización Inicio	Alta	Activa ignorada	...	✓	✕
<input type="checkbox"/>	2020-03-13 17:13:09	PLC Banco Didactico	Nivel Tanque Alto	Alerta	Activa ignorada	...	✓	✕
<input type="checkbox"/>	2020-03-13 17:08:31	PLC Banco Didactico	Temperatura Baja	Alerta	Activa ignorada	...	✓	✕

Figura 4.16: Elementos de la pestaña Alarmas.

El nivel de criticidad de las alarmas se especifica dentro de la programación en el motor de reglas según sea necesario.

4.5.3 Pestaña GRÁFICOS.

Al ingresar a la pestaña de gráficos, se muestra la siguiente pantalla general, que se aprecia en la Figura 4.17.

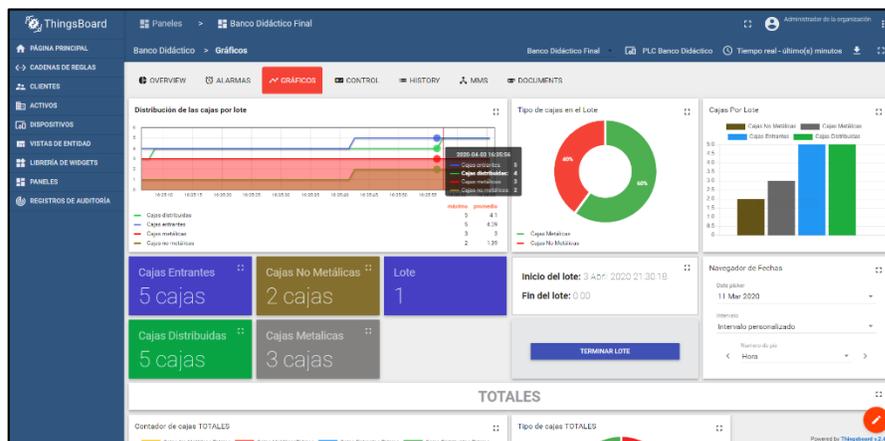


Figura 4.17: Pestaña de Gráficos.

Aquí se encuentran series temporales, gráficos de barras y pasteles, en los que se despliega la siguiente información, (ver Figura 4.18):

- Conteo de cajas por lote en serie temporal.
- Conteo de cajas entrantes, cajas metálicas, cajas no metálicas y cajas distribuidas en gráfico de barras.
- Tipo de cajas por lote en gráfico de pastel.
- Hora de inicio y hora de fin del lote.
- Navegador de fechas para visualizar histórico de series temporales.
- Conteo total de cajas en gráfico de barras.
- Gráfico de pastel para conteo total de cajas por tipo (metálicas, no metálicas).

Cada elemento de la ventana de gráficos se actualiza en tiempo real.

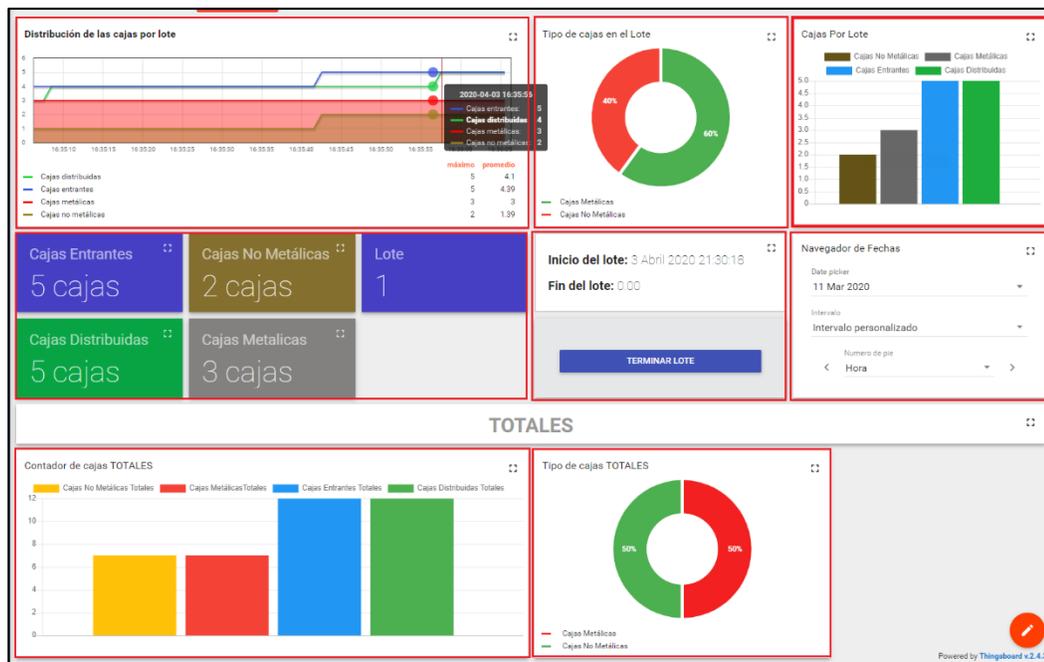


Figura 4.18: Elementos de la pestaña Gráficos.

4.5.4 Pestaña CONTROL.

Dentro de la pestaña de control, (ver Figura 4.19), se encuentran los controles para poner en estado de marcha o paro el banco, el movimiento de la banda transportadora, movimiento del brazo robot, y el estado de los sensores magnéticos que rodean el robot, tal como se muestra en la Figura 4.20

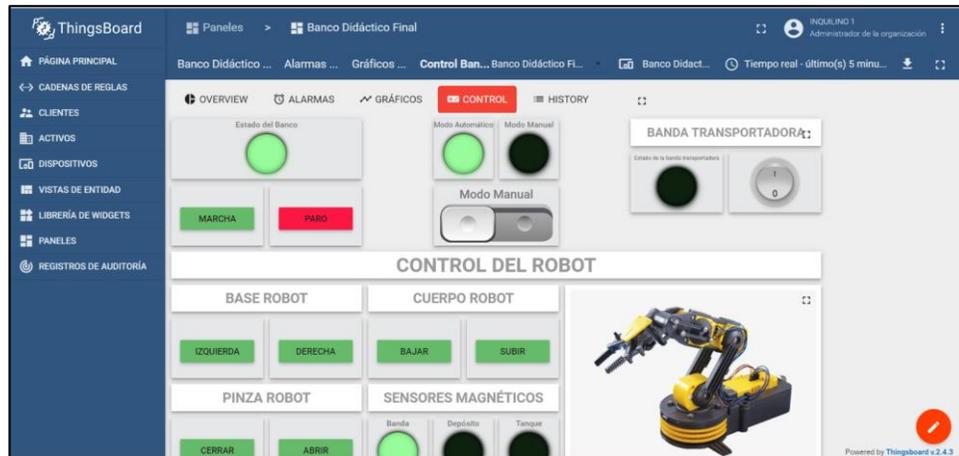


Figura 4.19: Pestaña de control.

Para poder tener el control del banco, lo primero que se debe hacer es presionar el pulsante de paro para desactivar el control automático actual, cuando el led indicador del banco se desactive se puede activar el switch de control manual.

Cuando el led indicador de control manual este activado, se pueden utilizar los pulsantes de control de los motores del brazo robótico y de la banda transportadora según sea necesario, (ver Figura 4.20).

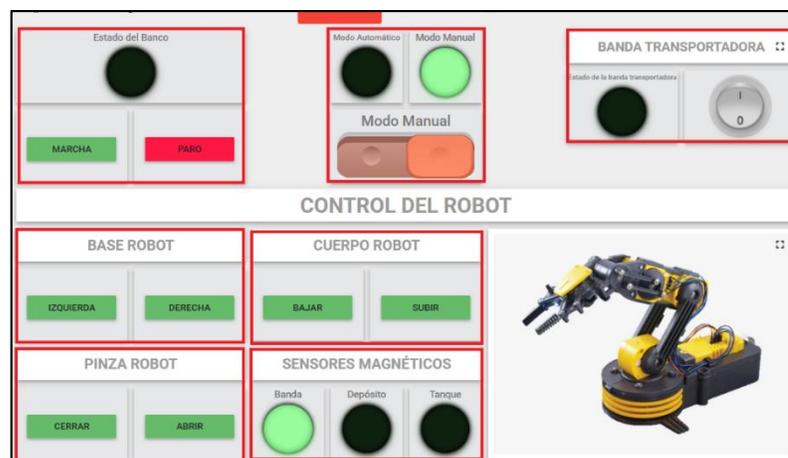


Figura 4.20: Elementos de la pestaña Control.

Los controles posibles son:

- Movimiento de la base del robot, hacia la derecha o izquierda.
- Movimiento del cuerpo del robot, hacia arriba o abajo.
- Movimiento de la pinza del robot, apertura o cierre.
- Activación o desactivación de la banda transportadora.

Cuando se requiera volver al control automático, simplemente se activa el pulsante de marcha del banco.

4.6 Pruebas integrales

Se realizaron pruebas integrales para comprobar el funcionamiento en conjunto de todos los elementos de la arquitectura planteada anteriormente, en la sección 2.1, para cual se planteó lo siguiente:

- Se comprobó que la pestaña Overview muestra la información general del proceso llevado por el banco didáctico de tal forma que el usuario puede obtener los datos de una manera simple. En la Figura 4.14 se aprecia el control estable para los sensores de nivel y temperatura, así como también el conteo de cajas por lote del proceso en tiempo real.
- Se ingresaron varias cajas entre metálicas y no metálicas, para que los contadores y las series temporales de las mismas puedan mostrar la información que se está generando.
- Se generaron las alarmas programadas para la comprobación de su funcionamiento, de esta forma en un ambiente de producción real se tendrá la certeza que saltarán las alarmas en el momento en el que ocurran.

Prueba por lotes

Se configuró el sistema de tal forma que cada 5 cajas entrantes se cierre un lote, de esta forma con el ingreso de varias cajas el número de lote se incrementará. Si el proceso se desconecta por un periodo mayor a 20 segundos (*timeout*), tiempo definido en ThingsBoard, el lote se cierra y empieza un nuevo lote para cuando el proceso se reanude.

Se realizó una prueba para conteo de cajas por lote, se ejecutaron 3 lotes con un total de 5 cajas por lote. Para esta prueba se energizó el banco didáctico y se comprobó el envío de datos a la plataforma ThingsBoard, luego se ingresaron cajas metálicas y cajas no metálicas de la siguiente manera:

1er Lote: Se enviaron 3 cajas no metálicas y 2 cajas metálicas.

2do Lote: Se introduce 1 caja metálica, luego se desconectaron momentáneamente los equipos de la red con un tiempo menor a 20 segundos, que fue el tiempo *timeout* programado, se conectaron los equipos a la red y finalmente se enviaron 2 cajas metálicas seguidas de 2 cajas no metálicas.

3er Lote: Se enviaron 5 cajas metálicas.

Con esta prueba verificamos el funcionamiento de todo el sistema, así como también validamos el envío de datos, correspondiente al lote, si existe un tiempo de desconexión en la red pequeño, menor al *timeout* programado. En la Figura 4.21 se muestran los resultados de la prueba en las series temporales de la pestaña “GRÁFICOS”, se seleccionó la ventana de tiempo en el histórico.

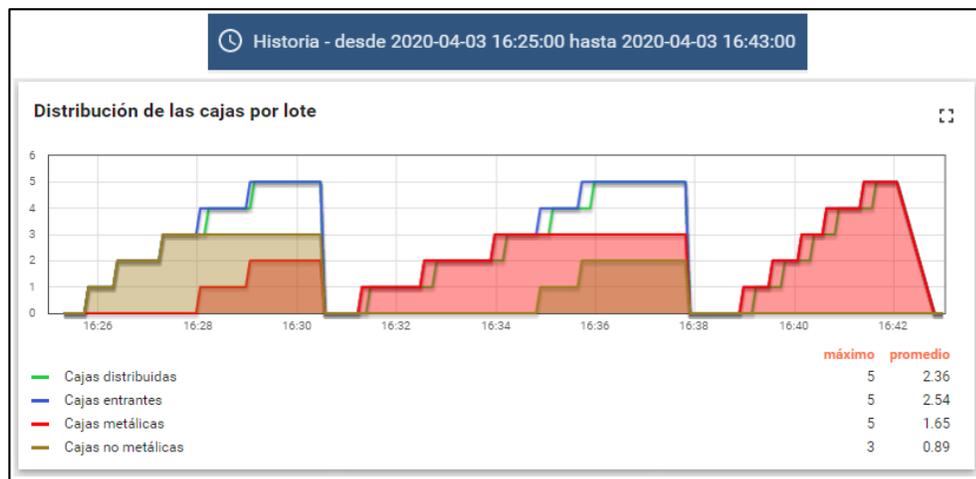


Figura 4.21: Resultado de la prueba ejecutada.

Para conocer los tiempos de distribución de cada caja se expande la ventana de la gráfica al área deseada y se puede visualizar el valor de cada métrica a lo largo del tiempo, se

selecciona el punto deseado y se muestran más detalles. Con estos valores es fácil determinar el tiempo de ingreso de cada caja, la demora hasta su distribución y el conteo de las mismas. En la Figura 4.22 se aprecia que una caja no metálica ingresó a las 16:34:52 y esta misma fue distribuida a las 16:35:08 siendo su tiempo de distribución un total de 16 segundos.

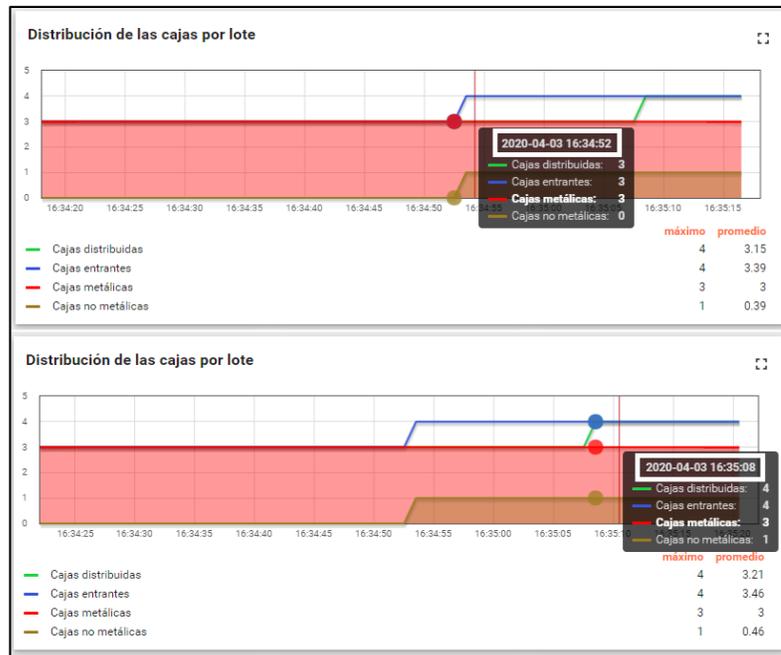


Figura 4.22: Tiempos de distribución de una caja en un lote.

Finalmente se verificó que los valores totales sean correctos, estos son la suma del conteo de cajas de todos los lotes, un total de 15 cajas, obteniendo como resultado las gráficas de la Figura 4.23.



Figura 4.23: Valores totales de las cajas distribuidas desde el primer lote.

Alarmas

Con el proceso del banco didáctico, descrito en la sección 1.5.3, se realizó otra prueba para verificar el funcionamiento de todas las alarmas programadas del proceso, un total de 8 alarmas, las cuales son:

- **Obstaculización Inicio:** ocurre cuando la caja se obstaculiza en el inicio de la banda transportadora.
- **Obstaculización Horno:** ocurre cuando la caja se obstaculiza en la guía de la banda transportadora que se encuentra en el horno.
- **Falla Robot Distribución:** ocurre cuando el robot no entrega la caja al depósito o al tanque, sucede si la caja llego al final de la banda transportadora y el robot no se mueve para desplazarla.
- **Falla Robot:** ocurre cuando el robot no regresa a su posición inicial.
- **Nivel Tanque Alto y Nivel Tanque Bajo:** límites alto y bajo para el valor del sensor de nivel en relación al porcentaje de llenado, 20% el límite bajo y 90% el límite alto.
- **Temperatura Baja y Temperatura Alta:** límites alto y bajo para el valor del sensor de temperatura, 25°C el límite bajo y 75°C el límite alto.

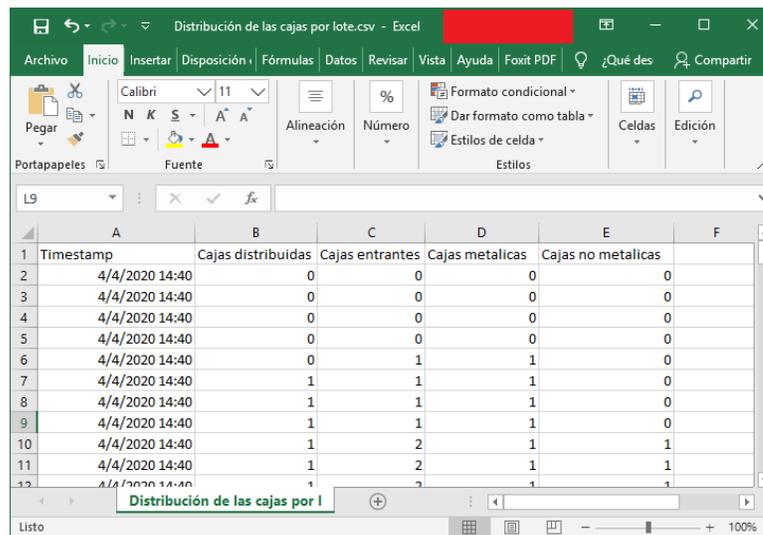
The screenshot shows the 'Alarms' page in the ThingsBoard interface. The page title is 'Banco Didáctico > Alarmas'. The left sidebar contains navigation options like 'PÁGINA PRINCIPAL', 'CADENAS DE REGLAS', 'CLIENTES', 'ACTIVOS', 'DISPOSITIVOS', 'VISTAS DE ENTIDAD', 'LIBRERÍA DE WIDGETS', 'PANELES', and 'REGISTROS DE AUDITORÍA'. The main content area displays a table of alarms with the following data:

Time of creation	Origin	Type	Criticality	Status	Actions
2020-04-01 17:29:29	PLC Banco Didáctico	Obstaculización Horno	Alta	Activa reconocida	... ✓ ✕
2020-04-01 17:29:25	PLC Banco Didáctico	Falla Robot Distribución	Crítica	Activa reconocida	... ✓ ✕
2020-04-01 17:29:17	PLC Banco Didáctico	Obstaculización Inicio	Alta	Activa reconocida	... ✓ ✕
2020-04-01 17:29:06	PLC Banco Didáctico	Falla Robot	Crítica	Activa reconocida	... ✓ ✕
2020-04-01 17:28:57	PLC Banco Didáctico	Nivel Tanque Alto	Alerta	Activa ignorada	... ✓ ✕
2020-04-01 17:28:45	PLC Banco Didáctico	Nivel Tanque Bajo	Alerta	Activa ignorada	... ✓ ✕
2020-04-01 17:28:06	PLC Banco Didáctico	Temperatura Bajal	Alerta	Activa ignorada	... ✓ ✕
2020-04-01 17:24:39	PLC Banco Didáctico	Temperatura Alta	Alerta	Activa ignorada	... ✓ ✕

At the bottom of the table, there is a pagination control showing 'Page: 1', 'Rows per page: 10', and '1 - 8 of 8'.

Exportación de la información

Finalmente, si se desea hacer un análisis de la información, la versión de pago ThingsBoard Premium Edition cuenta con la función de exportación de los datos y paneles en formato PDF y CSV. Para exportar la información de los widgets se hace click en la opción de exportación, como se indicó en la sección 2.7.10. Se realizó una prueba para exportar la distribución de las cajas por lote en formato CSV, (ver Figura 4.24).



	A	B	C	D	E	F
1	Timestamp	Cajas distribuidas	Cajas entrantes	Cajas metalicas	Cajas no metalicas	
2	4/4/2020 14:40	0	0	0	0	
3	4/4/2020 14:40	0	0	0	0	
4	4/4/2020 14:40	0	0	0	0	
5	4/4/2020 14:40	0	0	0	0	
6	4/4/2020 14:40	0	1	1	0	
7	4/4/2020 14:40	1	1	1	0	
8	4/4/2020 14:40	1	1	1	0	
9	4/4/2020 14:40	1	1	1	0	
10	4/4/2020 14:40	1	2	1	1	
11	4/4/2020 14:40	1	2	1	1	
12	4/4/2020 14:40	1	2	1	1	

Figura 4.24: Exportación de datos en formato CSV, visualizado en Excel.

Si solo se contase con la plataforma gratuita, al ser propietario del servidor en donde se hospeda, se puede exportar la información directamente desde la base de datos PostgreSQL configurada en el servidor. Otra opción sería la integración con una aplicación de terceros, por ejemplo, un sistema de análisis de datos; para más información consultar la sección “*Integrations*” de la documentación oficial de ThingsBoard.

Control del banco didáctico

Se verificó el control remoto de los actuadores del banco didáctico, los motores del robot y el motor de la banda transportadora. Para esto se utilizaron los widgets de control de la pestaña “CONTROL” que se muestra en la Figura 4.20. Se puede controlar el banco didáctico de forma manual solo si este modo se encuentra activado.

Funcionalidad de la plataforma ThingsBoard

Con estos puntos se ha comprobado la utilidad de la plataforma ThingsBoard y de la arquitectura IIoT planteada para la visualización de los datos, los cuales se obtienen en tiempo real de un proceso de manufactura. Se verificó también la funcionalidad de control remoto de los actuadores del proceso de manufactura.

4.7 Visualización

La implementación de todas las partes que componen la arquitectura puede ser visualizada tanto en ordenadores como en dispositivos móviles, (ver Figura 4.25 y Figura 4.26):

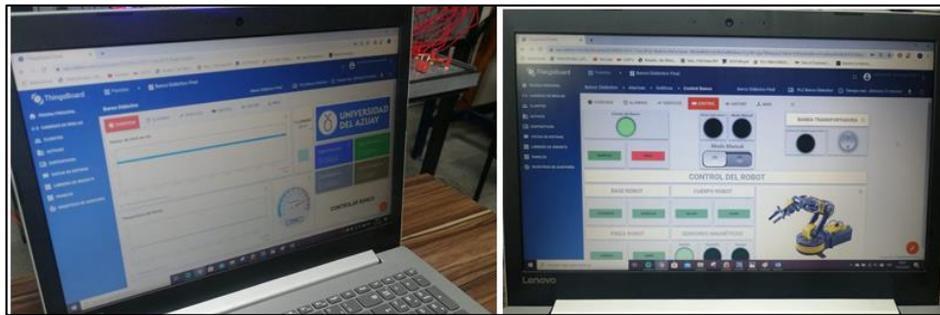


Figura 4.25: ThingsBoard visto desde el ordenador.

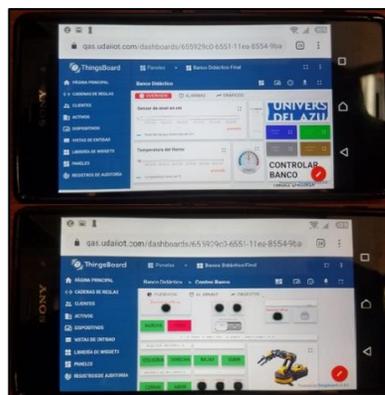


Figura 4.26: Plataforma IoT desde un dispositivo móvil.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

- El proceso de manufactura establecido en el banco didáctico permitió la simulación de un proceso industrial con el que se pudo implementar una arquitectura IIoT. Este contiene componentes como sensores industriales y actuadores conectados al PLC que permiten la recopilación de datos, además sus componentes cuentan con conexiones independientes para que se puedan realizar varios tipos de circuitos de control. Este banco es de gran utilidad para que los estudiantes realicen prácticas y proyectos relacionados con IIoT y la Industria 4.0.
- Se actualizó el PLC instalado en el banco didáctico por una versión nueva debido a que la versión de firmware del PLC antiguo no era compatible con el módulo IoT 2040. Se incorporaron nuevos componentes al banco didáctico como: el módulo IoT 2040 y un conversor de temperatura conectado a la termocupla tipo J. Con la implementación de la plataforma IoT hospedada en la nube se agregó la funcionalidad de control manual para los actuadores del banco didáctico.
- El módulo Siemens IoT 2040 permite la conexión entre el PLC Siemens S7-1200 e internet utilizando protocolos de comunicación IIoT como MQTT. Este contiene el sistema operativo Yocto Linux de gran versatilidad para incluir paquetes y componentes dentro del dispositivo, integrando el software con el hardware del mismo. Implementar físicamente un módulo IoT es muy fácil ya que solo se necesita la conexión a la red por Ethernet o Wi-Fi y la alimentación del mismo.
- El programa Node-Red es una interfaz visual de programación muy práctico para implementar la comunicación entre dispositivos IIoT y el internet, este se instaló en el módulo IoT brindando funcionales extendidas como puerta de salida a internet. La programación en Node-Red es amigable e intuitiva con el usuario ya que no requiere de gran conocimiento en programación, este cuenta con nodos ya programados de fácil entendimiento.

- La plataforma ThingsBoard contiene una gran cantidad de funcionalidades y características ideales para la implementación de sistemas IIoT, además, cuenta con una gran cantidad de documentación de fácil entendimiento, debido a que su software es de código abierto. Cuenta con una interfaz web fácil de usar y muy bien elaborado, que permite crear paneles personalizables con amplias librerías de widgets. Cuenta con varios API's para la comunicación con la plataforma utilizando diferentes protocolos y facilita la integración con dispositivos conectados a sistemas heredados y aplicaciones de terceros. Se puede instalar en varios escenarios, soporta el despliegue de la plataforma en la nube a modo de clúster y monolítico. Su versión Premium puede exportar información en formato de PDF y hojas de datos como CSV.
- Google Cloud cuenta con un crédito gratuito muy accesible y la facilidad de un tiempo prolongado de 1 año para utilizar todos sus servicios en la nube. Con este crédito el usuario puede implementar las máquinas virtuales que desee sin restricción alguna hasta que se consuma el crédito otorgado, siendo una herramienta ideal para el desarrollo de proyectos tecnológicos y ambientes de pruebas.
- Se configuraron dos máquinas virtuales con el sistema operativo Ubuntu Server 18.04.4 LTS en la nube de Google Cloud para la instalación de la versión gratuita y la versión de pago de la plataforma ThingsBoard. Si la cantidad de telemetría e información almacenada satura las capacidades de memoria y/o memoria en disco de la máquina virtual, los servicios en la nube alertan, sugieren y amplían recursos según requiera el usuario. En el desarrollo de este proyecto se simuló este inconveniente y se amplió el tamaño del disco de una máquina virtual en la nube.
- Se crearon paneles de monitorización y de control en la plataforma ThingsBoard para visualizar los datos generados por el banco didáctico en tiempo real, además estos datos se almacenan en la nube generando un histórico que se encuentra disponible para la extracción de información para un posterior análisis.

- Las tecnologías actuales que se manejan en la industria mejoran de manera sustancial la cadena de valor de la empresa, la extracción de datos sobre los procesos de manufactura es primordial para el mejoramiento de los mismos, de manera que se genera información útil para el personal administrativo y operativo.
- Todo el software utilizado en la arquitectura del proyecto cuenta con utilidades de importación y exportación de las configuraciones, código de programación y elementos de la interfaz. Estas funcionalidades permiten al usuario crear respaldos y migrar dichos datos a otras instancias o infraestructura de hardware.
- Se realizaron guías prácticas paso a paso que facilitarán a los estudiantes y docentes el aprendizaje de tecnologías IIoT, programación en PLC y creación de paneles con widgets personalizables para la visualización y control de procesos industriales utilizando la plataforma IoT ThingsBoard.
- El rango de temperatura de la termocupla tipo J instalada es de 0-100°C. El sensor de nivel mantiene una variación de voltaje en el rango de 0-10V, este se encuentra a una altura determinada que se encuentra marcada en la asta del mismo. Se realizaron medidas en función del porcentaje de llenado del tanque debido a que el sensor ultrasónico está dimensionado para aplicaciones a gran escala, que dificulta su funcionamiento en componentes de dimensiones reducidas. El esquema de conexión se realizó con las imágenes expuestas en los terminales de los componentes del banco didáctico
- El sistema IIoT implementado en el proceso de manufactura permite visualizar la distribución de las cajas en gráficas estadísticas conforme al tipo de elemento: metálico o no metálico, el tiempo de distribución y el número de lote al que corresponde. Se implementaron alarmas para notificar los posibles tipos de fallas en obstrucción a lo largo de todo el proceso, así como también se especificaron límites bajos y altos en la medición de los sensores. Este sistema de alarmas notifica alertas al usuario a través de correo electrónico.

Recomendaciones:

- Siempre tener en cuenta que las conexiones eléctricas se realicen correctamente. Todos los cables, borneras, entre otros no deben presentar fallas, de igual manera guiarse en el esquema de conexión para conectar los terminales de los componentes.
- Se debe tener presente que los dispositivos utilizados sean compatibles para su correcto funcionamiento, utilizar las versiones correctas de los programas evitará problemas de conexión entre dispositivos en el transcurso del proyecto.
- Verificar las direcciones IP de los dispositivos utilizados y comprobar la conexión de red entre los mismos realizando pruebas iniciales.
- Guiarse en la hoja de datos de cada sensor y actuador que entrega el fabricante para conectar y utilizar los componentes de manera correcta. Verificar los voltajes de alimentación y la referencia con tierra.
- Se debe elegir y configurar la máquina virtual de acuerdo a los requerimientos del sistema implementado, así como la proyección de su uso a lo largo del tiempo, para que los recursos utilizados por la misma ofrezcan el mayor rendimiento de la plataforma.
- Apoyarse en la documentación disponible para los dispositivos físicos y programas utilizados, de esta manera cualquier duda que surja en la realización del proyecto y de las guías prácticas podrá ser solventada.
- Seguir paso a paso las instrucciones de las guías didácticas creadas y verificar la asignación de variables en cada configuración, de forma que el estudiante entienda la funcionalidad de cada componente.

- Para analizar la información recuperada del banco didáctico, se recomienda visualizar el panel de “Overview” y “Gráficos” en la plataforma ThingsBoard, en donde se muestran los parámetros físicos y tiempos correspondientes a cada lote de producción.
- Si se desea realizar un análisis de los datos y de la información se recomienda exportar los paneles en formato PDF y CSV. El formato CSV se puede importar en una hoja de datos (por ejemplo, Excel) en donde se puede manipular, filtrar y visualizar la información de mejor manera. Se recomienda visualizar la documentación de la plataforma ThingsBoard, sección *Integrations* que facilita la exportación de toda la *data* a sistemas heredados y aplicaciones de terceros.
- Al finalizar el proyecto y guardar los cambios se debe verificar que tanto los programas del PLC, Node-Red y de la plataforma IoT se estén comunicando sin problemas. Verificar la conexión a internet del módulo IoT. Cuando se conecte un dispositivo IoT a la plataforma ThingsBoard verificar en Node-Red que las credenciales de acceso correspondan al dispositivo objetivo.

REFERENCIAS

- Alqinsi , P., Matheus Edward , I. J., Ismail, N., & Darmalaksana , W. (2018). IoT-Based UPS Monitoring System Using MQTT Protocols. *2018 4th International Conference on Wireless and Telematics (ICWT)*.
- Amazon Web Services, Inc. . (2020). AWS. Retrieved Febrero 27, 2020, from <https://aws.amazon.com/es/>
- Arthur O'Sullivan, S. M. (2007). *Economics Principles in Action*. Massachusetts: Prentice Hall.
- Beimborn, D., Miletzki, T., & Wenzel, S. (2011). Platform as a Service (PaaS). *Business & Information Systems Engineering*, 3(6), 381-384.
- BITKOM, V. a. (2016). Implementation Strategy Industrie 4.0, Report on the results of the Industrie 4.0 Platform.
- Blanco Rojas, M., González Rojas, K., & Rodríguez Molano, J. (2017). *Propuesta de una arquitectura de la Industria 4.0 en la cadena de suministro desde la perspectiva de la ingeniería industrial* . Bogotá: Universidad Distrital Francisco José de Caldas.
- CENTER, F. M. (n.d.). *FORD*. Retrieved Enero 31, 2019, from <https://media.ford.com>
- Cittolin, G. F. (2016-2019). *Smart-Tech*. (Smart-Tech) Retrieved Enero 22, 2020, from <https://www.npmjs.com/package/node-red-contrib-s7>
- del Val Román, J. L. (2016). *Industria 4.0: La transformación digital de la industria*. Bilbao: Facultad de Ingeniería de la Universidad de Deusto.
- DigitalOcean, LLC. (2020). *DigitalOcean*. Retrieved Febrero 27, 2020, from <https://www.digitalocean.com/pricing/>
- Dooley, K. (2002). *Designing Large-Scale LANs*. O'Reilly.
- Dragoni, N., Giallorenzo, S., Lafuente, A., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices : Yesterday , Today , and Tomorrow. In *Mazzara M., Meyer B. (eds) Present and Ulterior Software Engineering*. (pp. 195-216). Cham: Springer.
- Gallo Castillo, T. L. (2018). *Desarrollo e implementación de internet industrial de las cosas aplicado al laboratorio de plc's de la facultad de ingeniería industrial en procesos de automatización de la universidad técnica de Ambato*. Ambato: Escuela Superior Politécnica de Chimborazo.

- Gavira, R. M. (2018). *Industria 4.0 y sus aplicaciones a la optimización de procesos y eficiencia energética*. Sevilla: Universidad de Sevilla.
- Google. (2020). *Google Cloud*. Retrieved Febrero 27, 2020, from <https://cloud.google.com/?hl=es>
- Group, T. O. (2020). *Open Group*. (The Open Group) Retrieved Enero 10, 2020, from <https://www.opengroup.org/soa/source-book/msawp/p2.htm>
- Grupo Festo. (2020). *FESTO*. Retrieved Febrero 17, 2020, from <https://www.festo.com>
- Guth, J., Breitenbucher, U., & Falkenthal, M. (2016). Comparison of IoT Platform Architectures: A Field Study based on a Reference Architecture. *2016 Cloudification of the Internet of Things (CIoT)*.
- HiveMQ. (2019, Agosto 20). *HiveMQ*. (HiveMQ) Retrieved from <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/>
- Indiamart. (2016). *Indiamart*. Retrieved from Indiamart: <https://www.indiamart.com/proddetail/smart-home-installation-20426362255.html>
- Interempresas, R. (n.d.). *Interempresas*. Retrieved Enero 31, 2019, from <http://www.interempresas.net>
- International Telecommunication Union (ITU). (n.d.). *International Telecommunication Union (ITU)*. Retrieved Enero 10, 2020, from <https://www.itu.int/rec/T-REC-X.509>
- J. Wan, S. T. (2016). Software-Defined Industrial Internet of Things in the Context of Industry 4.0. *IEEE SENSORS JOURNAL*.
- Kutasi, D. N. (2019). Process Control with IIoT Capabilities of the Hollow Cathode Plasma Nitriding . *2019 20th International Carpathian Control Conference (ICCC)*.
- Llamas, L. (2019, Abril 17). *Ingeniería, informática y diseño*. Retrieved Enero 21, 2020, from <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>
- López Flores, M. X. (2019). *Industria 4.0 para la monitorización de un proceso industrial*. Ambato: Universidad técnica de Ambato.
- Mbohwa, C. &. (2018). Performance assessment of companies under IIoT architectures: Application of grey relational analysis technique. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*.

- Microsoft Azure. (2020). *Microsoft Azure*. Retrieved from ¿Qué es middleware?: <https://azure.microsoft.com/es-es/overview/what-is-middleware/>
- MQTT Community. (2019, Julio 4). *MQTT*. (mqtt.org) Retrieved Febrero 5, 2020, from <https://github.com/mqtt/mqtt.github.io/wiki/server-support>
- Patiño Solís, P. E., & Patiño Calle, A. X. (2015). *Diseño y elaboración de un banco didáctico para la manufactura integrada por computador*. Cuenca: Universidad del Azuay.
- Pozo, D. (2016). *La Empresa Digital: camino hacia Industria 4.0*. Siemens. Retrieved from https://www.upv.es/jornada/Industria/descargas/Presentacion_Siemens.pdf
- Priya, S. S. (2019). Self Configurations, Optimization and Protection Scenarios with wireless sensor networks in IIoT . *2019 International Conference on Communication and Signal Processing (ICCSP)*.
- PROFINET. (2014, Octubre). *PROFINET System Description Technology and Application*. Retrieved from http://us.profinet.com/wp-content/uploads/2012/11/PROFINET_SystemDescription_ENG_2014_web.pdf
- Ray, B. (2016, Agosto 03). *Link Labs*. Retrieved from IoT Platforms: What They Are & How To Select One: <https://www.link-labs.com/blog/what-is-an-iiot-platform>
- Rouse, M. (2017, Febrero). *SearchDataCenter*. Retrieved Enero 20, 2020, from <https://searchdatacenter.techtarget.com/es/definicion/Internet-Industrial-de-las-Cosas-IIoT>
- Rubell, B. (2020, Enero 28). *GitHub*. Retrieved from https://github.com/mqtt/mqtt.github.io/wiki/public_brokers
- Schütz, S. W., Kude, T., & Popp, K. M. (2013). The Impact of Software-as-a-Service on Software Ecosystems. *Software Business. From Physical Products to Software Services and Solutions*, 150, 130-140.
- Shelby, Z., Hartke, K., & Bormann, C. (2014, Junio). *RFC7252*. (I. Tools, Ed.) Retrieved from <https://tools.ietf.org/html/rfc7252>
- SICK, Sensor Intelligence. (2020). *SICK AG*. Retrieved Febrero 20, 2020, from <https://www.sick.com>
- SIEMENS. (1996-2020). *SIEMENS Ingenuity for life*. (siemens.com) Retrieved Enero 22, 2020, from <https://new.siemens.com/global/en.html>
- SIEMENS. (2018). *SIMATIC IOT2040: Primera instalacion y primer programa en node-*

red. SIMATIC IOT2040 V1.0.

- SIEMENS. (n.d.). *S. España*. Retrieved Enero 31, 2019, from <https://w5.siemens.com/spain>
- Su, W.-T. C.-C.-C. (2019). An Extensible and Transparent Thing-to-Thing Security Enhancement for MQTT Protocol in IoT Environment. *2019 Global IoT Summit (GIoTS)*.
- Tavizon Salazar, A., Guajardo, T., & Laines, C. (2016). *IOT, el internet de las cosas y la innovacion de sus aplicaciones*. Nuevo León, México: Autonomous University of Nuevo León.
- ThingsBoard. (2020). *ThingsBoard Documentation*. (ThingsBoard) Retrieved Enero 10, 2020, from <https://thingsboard.io/docs/>
- Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., & Gil, S. (2015). Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud. *10th Computing Colombian Conference (10CCC)*.
- X. Zhou, Y. H. (2017). An example for industry 4.0: Design and implementation of a mobile app for industrial surveillance based on cloud. *5th International Conference on Enterprise Systems*.
- YOCTO PROJECT. (2018). *Yocto Project*. Retrieved Noviembre 13, 2019, from <https://www.yoctoproject.org>
- Yuan, M. (2017, Octubre 4). *IBM*. (IBM Developer) Retrieved Enero 21, 2020, from <https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html>

6. ANEXOS

6.1 Anexo 1: Brokers MQTT

A continuación, en la Tabla 6.1 se realiza una comparativa de los diferentes servidores de mensajería MQTT disponibles en el mercado según los siguientes parámetros:

- QoS (Quality of Service): Referencia al nivel de calidad con el que se envían los mensajes dentro del protocolo, tipo 1, 2 y 3; se explica más adelante en la sección 1.3.3.
- Autenticación: Ofrece servicios de autenticación con nombre de usuario y contraseña.
- Bridge: función de reenvío de mensajes a otros brokers.
- \$SYS: Determina si se pueden acceder a los *topics* \$SYS que contienen información del broker MQTT.
- Topics dinámicos: creación de *topics* dentro del broker según el requerimiento de los clientes.
- Clúster: soporte de añadido dinámico (escalabilidad) de nuevos nodos.
- WebSockets: Permite conexiones en la red.
- Plugins e integración con otras plataformas.

Tabla 6.1: Comparación de servidores de mensajería MQTT.

Servidor	QoS 0	QoS 1	QoS 2	Autenticación	Bridge	\$SYS	SSL	Tópicos dinámicos	Clúster	WebSockets	Plugin e integración
2lemetry	✓	✓	✓	✓	✓	§	✓	✓	✓	✓	✗
Jmqtt	✓	✓	✓	✓	✓	§	§	✓	§	✓	✓
Apache ActiveMQ	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓

Apache ActiveMQ Artemis	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓
Bevywise IoT Platform	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	p
emitter	✓	§	✗	✓	✗	✗	✓	✓	✓	✓	✗
emqtt	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
flespi	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗
GnatMQ	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗
HBMQTT	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓
HiveMQ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IBM WIoT Message Gateway	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
JoramMQ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mongoose	✓	✓	?	?	?	?	?	?	?	?	?
moquette	✓	✓	✓	✓	?	?	✓	?	p	✓	✗
mosca	✓	✓	✗	✓	?	?	?	?	✗	✓	✗
mosquitto	✓	✓	✓	✓	✓	✓	✓	✓	§	✓	✓
MQTT.js	✓	✓	✓	§	✗	✗	✓	✓	✗	✓	✗
MqttWk	✓	✓	✓	✓	✓	?	✓	✓	✓	✓	✗
RabbitMQ	✓	✓	✗	✓	✗	✗	✓	✓	?	?	?
RSMB	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	?
Software AG Universal Messaging	✓	✓	✓	✓	✗	✗	✓	✓	✓	p	✗
Solace	✓	✓	✗	✓	§	✓	✓	✓	✓	✓	✗
SwiftMQ	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓

Trafero Tstack	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗
VerneMQ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
WebSphere MQ	✓	✓	✓	✓	✓	✓	✓	✓	?	?	?

Fuente: (MQTT Community, 2019)

En donde: ✓ permitido, ✗ no permitido, ? incierto, § ver limitaciones, p en planeación.

Limitaciones:

- 2lemetry: utiliza dominios en donde el primer segmento del tema es el propio nombre del dominio.
- Solace: entrega una solución de bridge entre brokers.
- MQTT.js: acepta las conexiones mediante nombre de usuario y contraseña, pero en realidad este no autentifica la conexión realizada.
- Mosquitto: la función de clustering se logra a un nivel de backend.
- Software AG: proporciona funciones de clustering y soluciones de bridge.

6.2 Anexo 2: Configuraciones de los controladores PID externos

Configuración del controlador de temperatura

Descripción	Parámetro	Valor
Nivel de complejidad de función	FUNC	BASC
Tipo de ingreso	In1	J_tc
Unidad de medida del ingreso	In1.u	Grados celcius
Cantidad de decimales	dP1	1-dp
Configuración de salida 1	Out 1	Revr
Tipo de salida 1	o1.ty	0-20
Tiempo de ciclo salida 1	CYC 1	2
Modo de transferencia en falla salida 1	o1.ft	0.0
Configuración de salida 2	Out2	-AL2
Función de alarma 1	A1.Fn	none
Función de alarma 2	A2.Fn	dE.Lo
Modo de operación alarma 2	A2.md	norm
Función de transferencia en falla alarma 2	A2.Ft	on
Selección de función de ajuste automático	SELF	yes
Valor objetivo mínimo	SP1.L	5.0
Valor objetivo máximo	SP1.H	80.0
Valor seleccionable por el usuario	SEL 1	none
Valor seleccionable por el usuario	SEL 2	none
Valor seleccionable por el usuario	SEL 3	A2.dv
Valor seleccionable por el usuario	SEL 4	none
Valor seleccionable por el usuario	SEL 5	none

Figura 4.27: Configuración de controlador de temperatura.

(Patiño Solís & Patiño Calle, 2015)

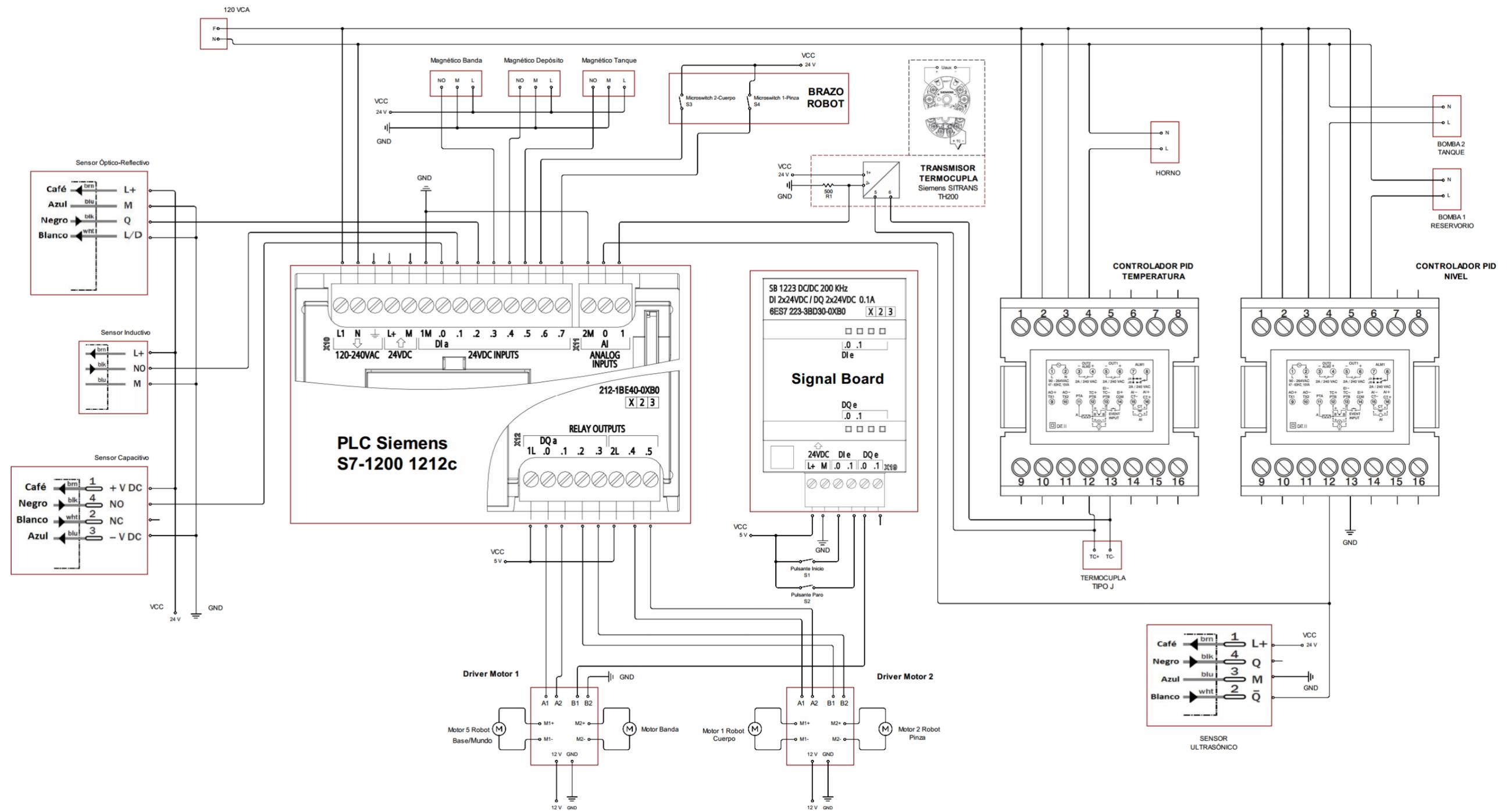
Configuración del controlador de nivel

Descripción	Parámetro	Valor
Nivel de complejidad de función	FUNC	BASC
Tipo de ingreso	In1	0-10
Unidad de medida del ingreso	In1.u	pu
Cantidad de decimales	dP1	2-dp
Valor mínimo ingreso 1	in1.L	0.00
Valor máximo ingreso 1	in1.H	50.00
Configuración de salida 1	Out 1	revr
Tipo de salida 1	o1.ty	rely
Tiempo de ciclo salida 1	CYC 1	1
Modo de transferencia en falla salida 1	o1.ft	bpls
Configuración de salida 2	Out2	Cool
Tipo de salida 2	O2.ty	rely
Tiempo de ciclo salida2	CyC2	1.0
Modo de transferencia en falla salida 2	O2.ft	Bpls
Función de alarma 1	A1.Fn	none
Función de alarma 2	A2.Fn	none
Selección de función de ajuste automático	SELF	yes
Valor objetivo mínimo	SP1.L	30.0
Valor objetivo máximo	SP1.H	45.28
Valor seleccionable por el usuario	SEL 1	none
Valor seleccionable por el usuario	SEL 2	none
Valor seleccionable por el usuario	SEL 3	none
Valor seleccionable por el usuario	SEL 4	None
Valor seleccionable por el usuario	SEL 5	None

Figura 4.28: Configuración de controlador de nivel.

(Patiño Solís & Patiño Calle, 2015)

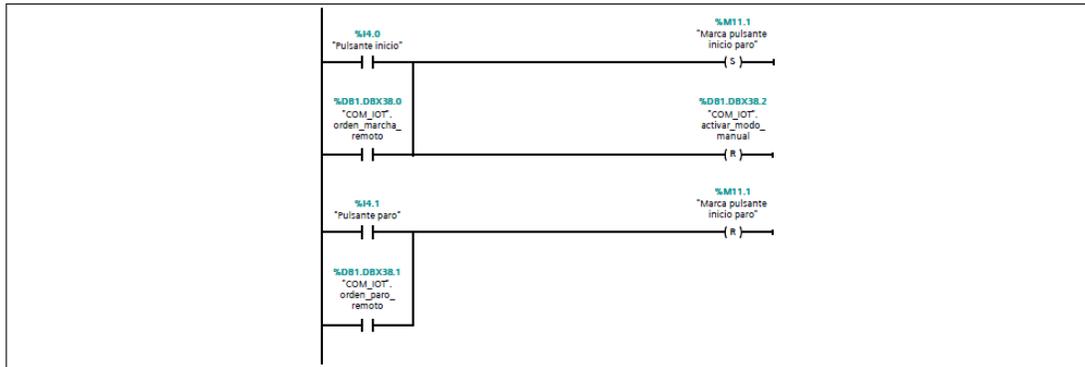
6.3 Anexo 3: Diagrama de conexión del banco didáctico



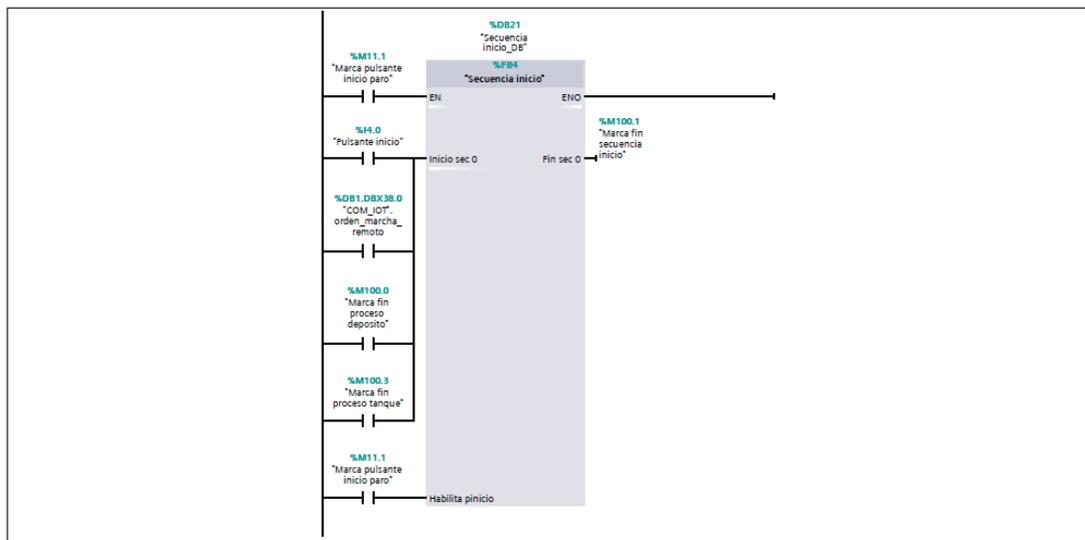
6.4 Anexo 4: Código implementado en el PLC s7-1200 1212c AC/DC/RLY

BLOQUE MAIN

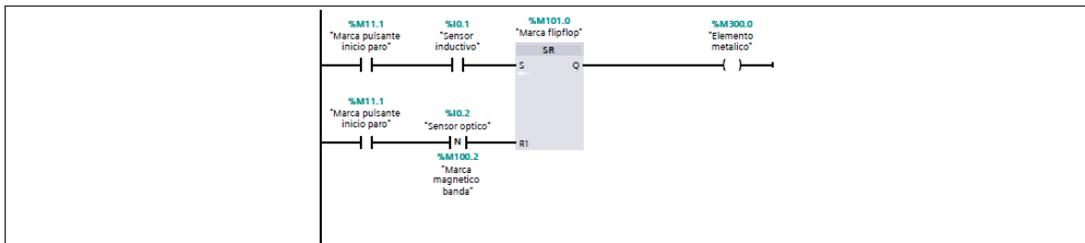
Segmento 1: Pulsantes marcha y paro



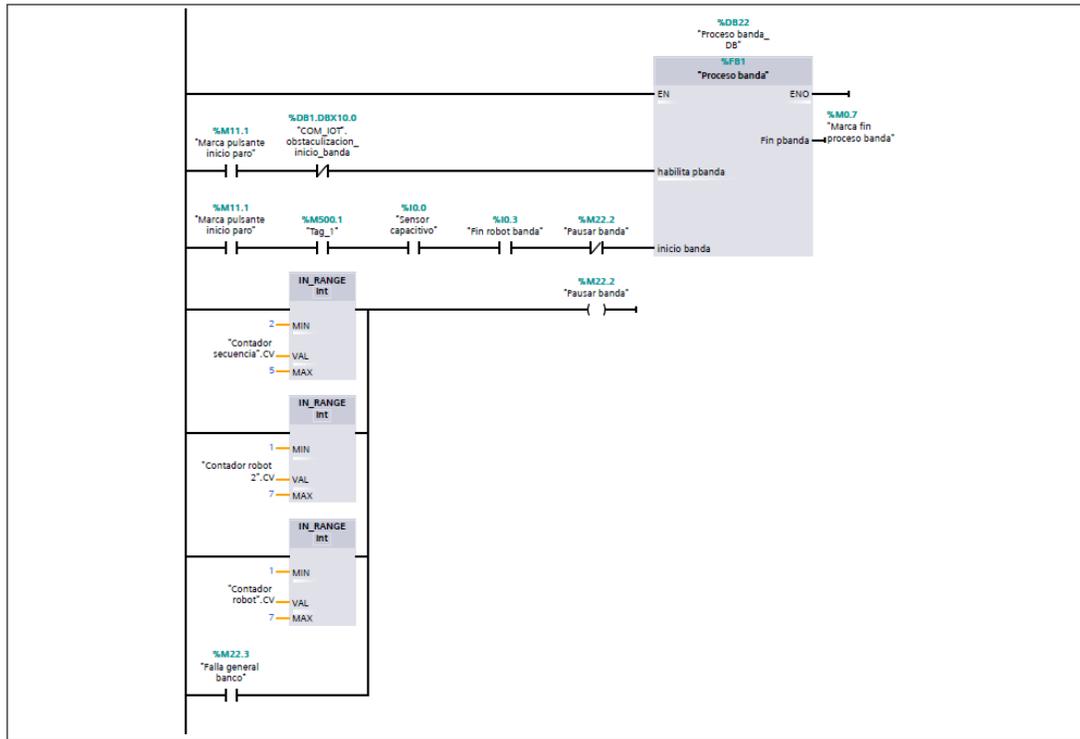
Segmento 2: Secuencia inicio



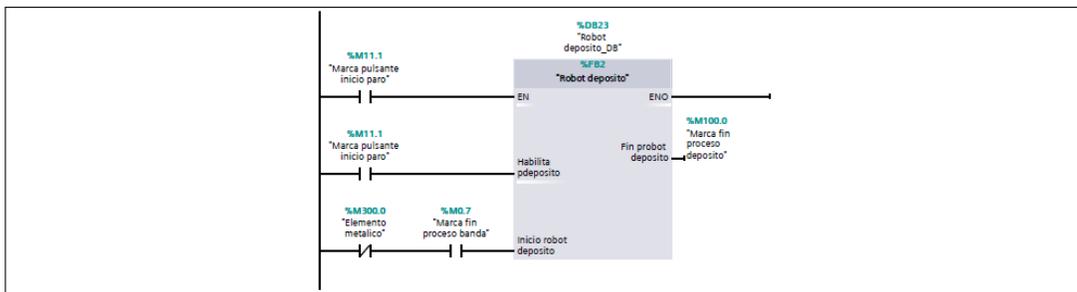
Segmento 3: Lectura de sensores en el ingreso



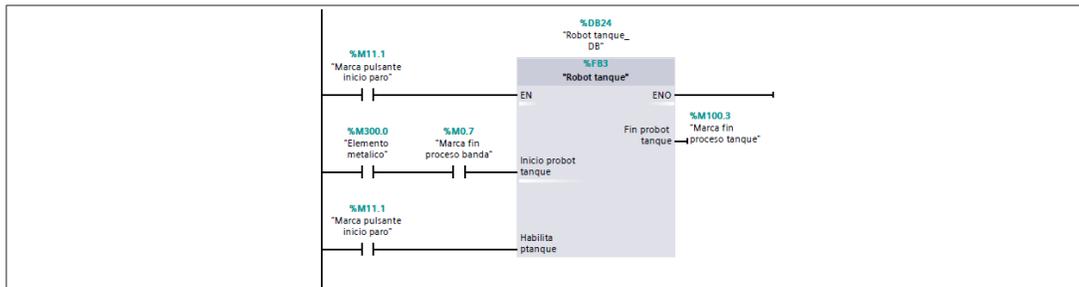
Segmento 4: Proceso banda transportadora



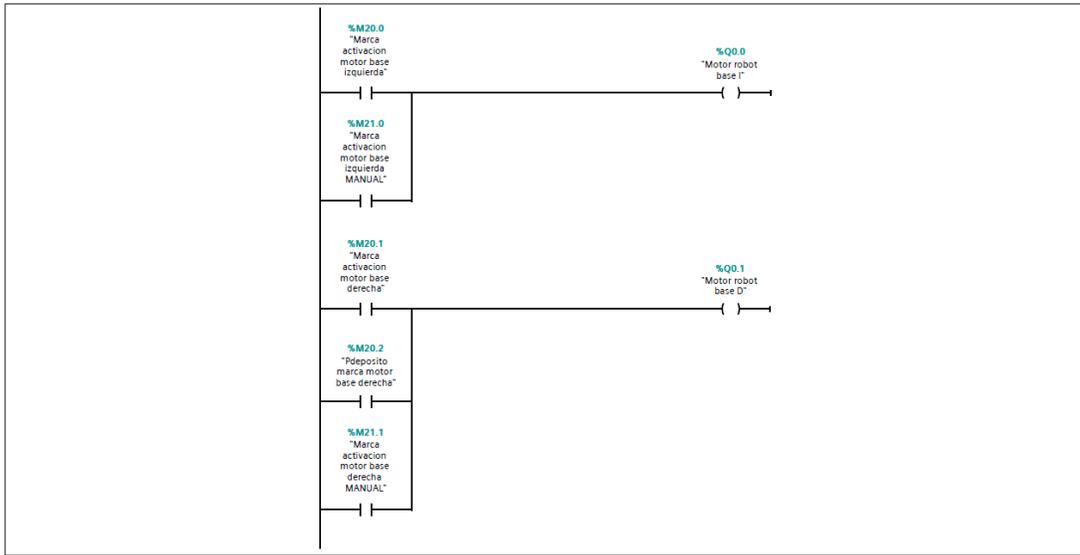
Segmento 5: Robot deposito



Segmento 6: Robot tanque



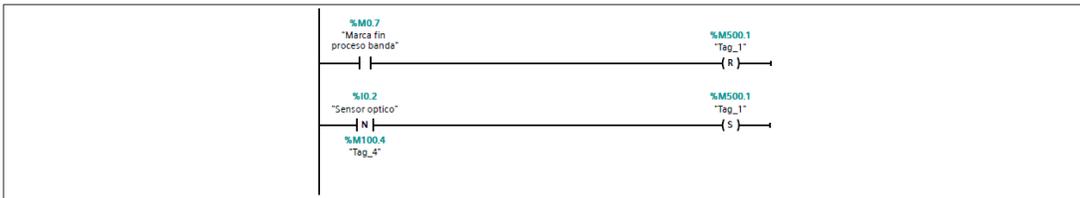
Segmento 7: Movimientos de la base brazo robot y banda



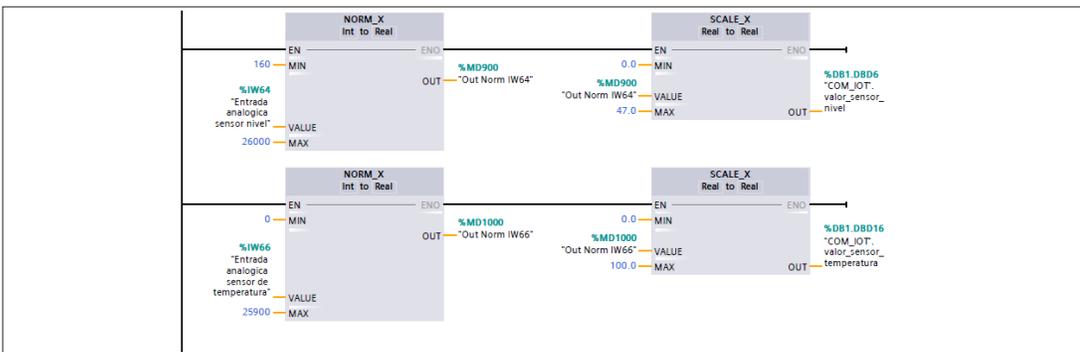
Segmento 8: Tag 500 pulsante de inicio



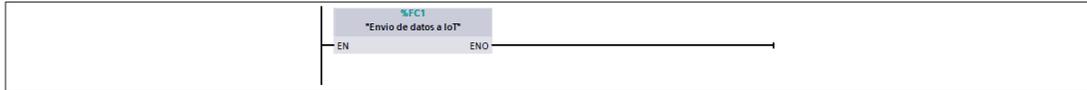
Segmento 9: Fin proceso banda, accionamiento brazo robot



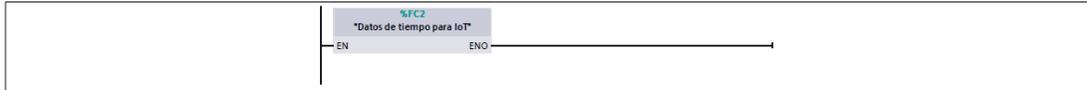
Segmento 10: Lectura de sensores nivel/temperatura



Segmento 11: Envío de datos a IoT



Segmento 12: Datos de tiempo para IoT

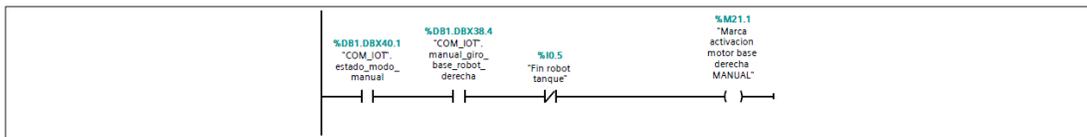


Segmento 13: Control manual

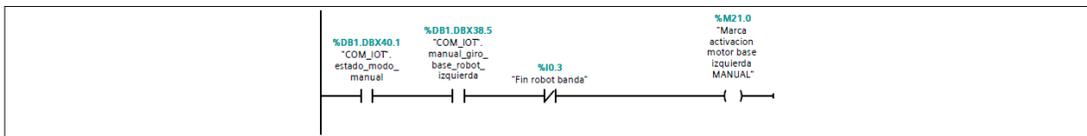


CONTROL MANUAL

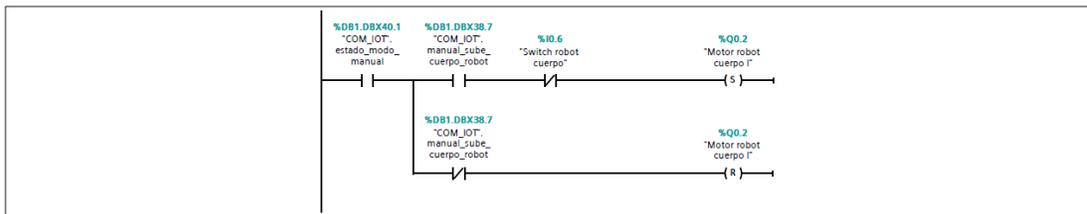
Segmento 1: Giro base derecha



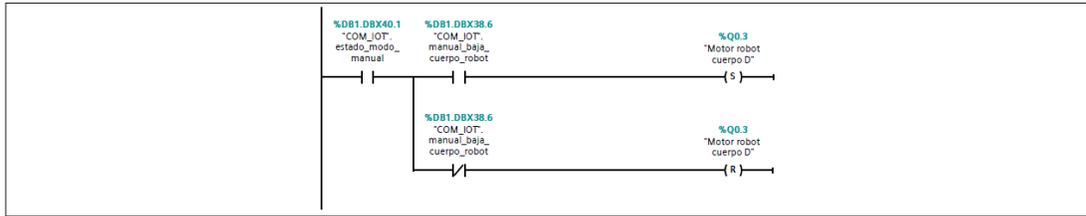
Segmento 2: Giro base izquierda



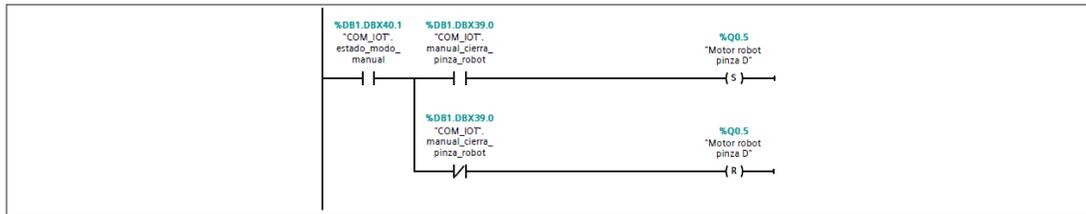
Segmento 3: Sube cuerpo



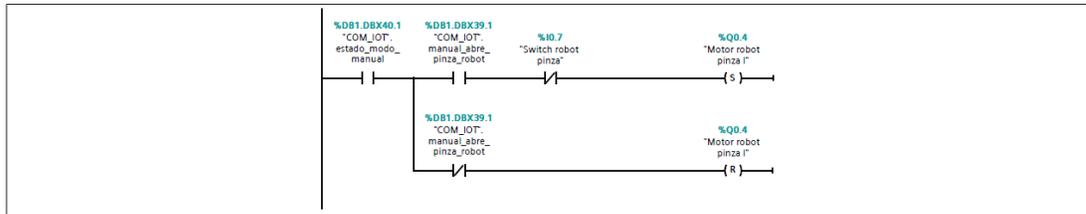
Segmento 4: Baja cuerpo



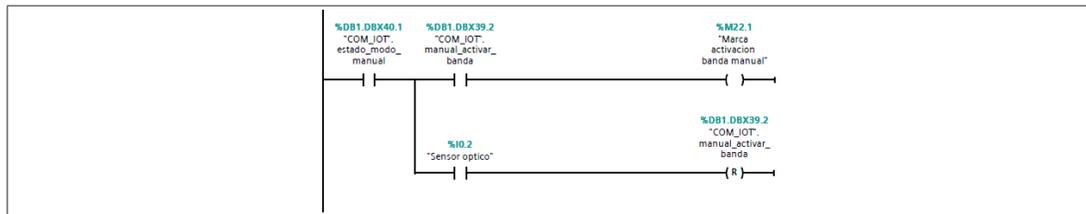
Segmento 5: Cierra pinza



Segmento 6: Abre pinza

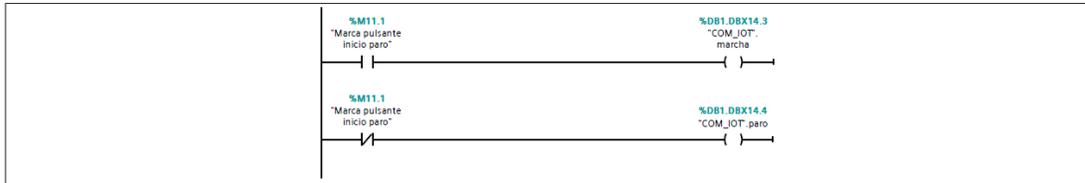


Segmento 7: Banda transportadora

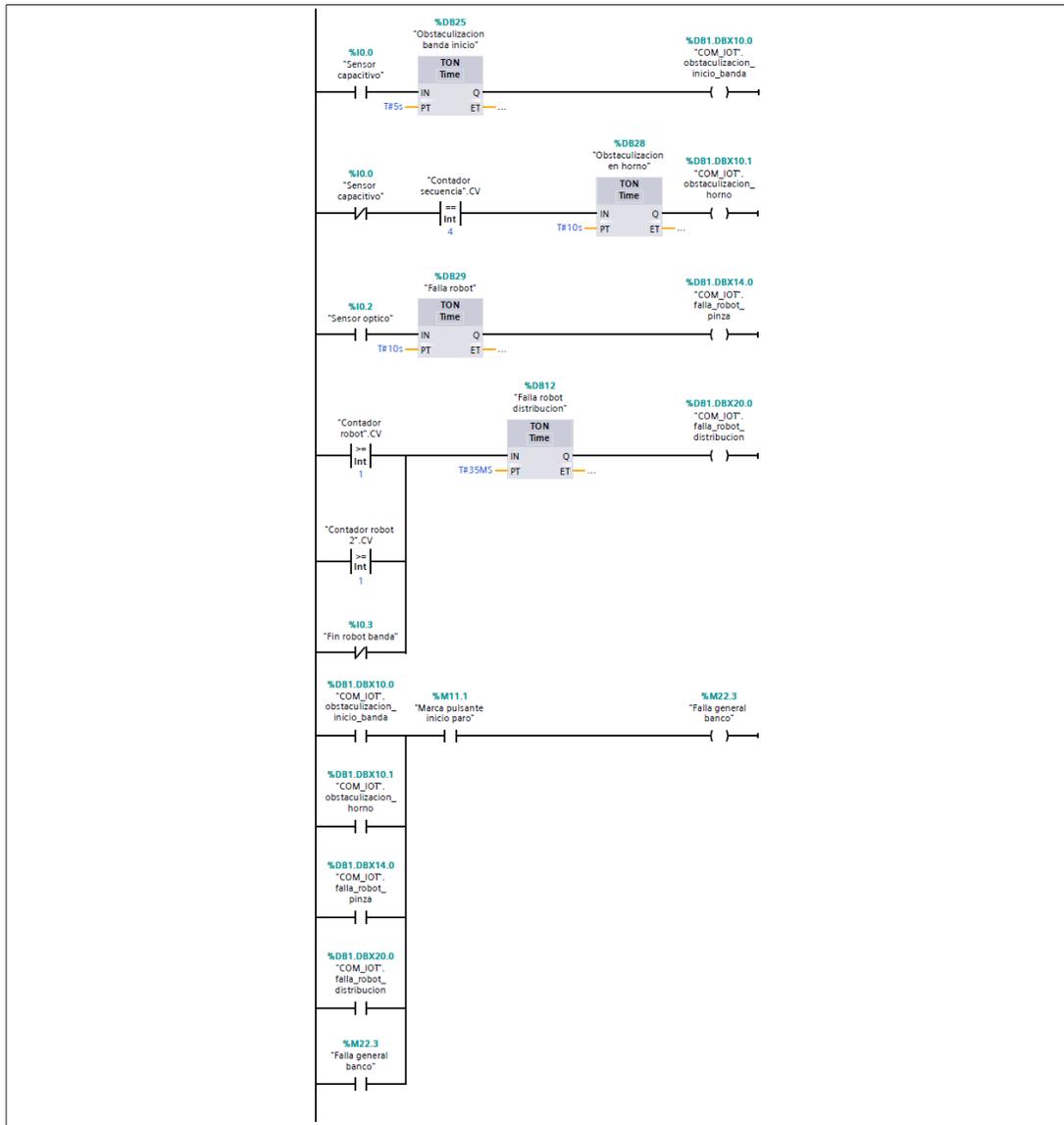


ENVÍO DE DATOS IOT

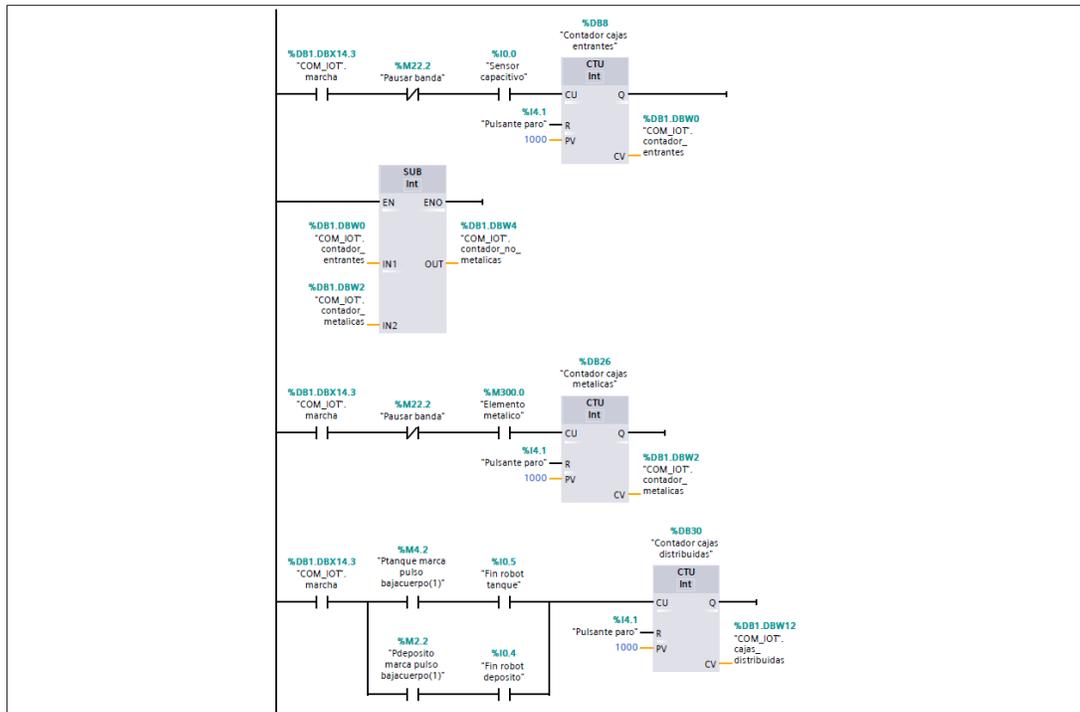
Segmento 1: Estado de marcha y paro



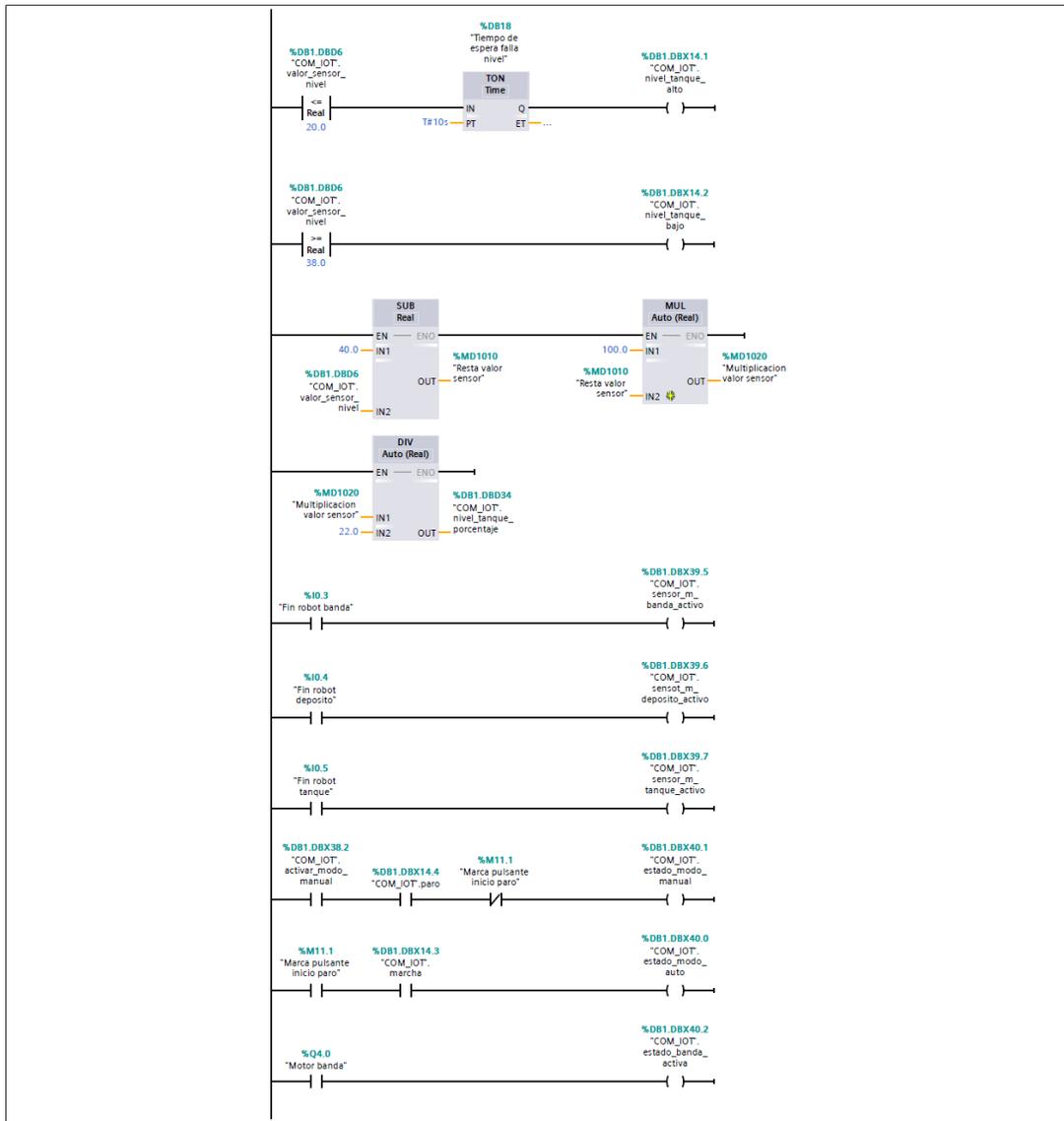
Segmento 2: Fallas

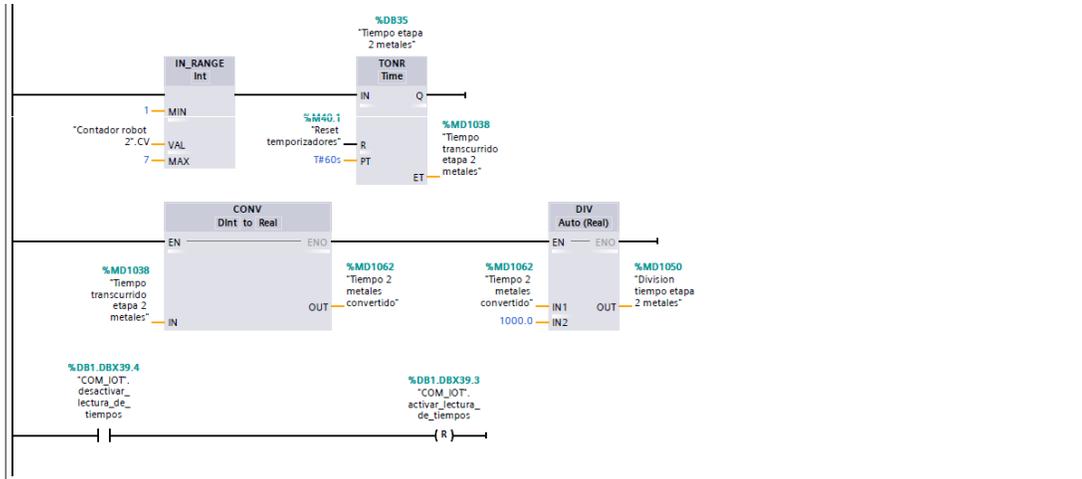


Segmento 3: Contadores



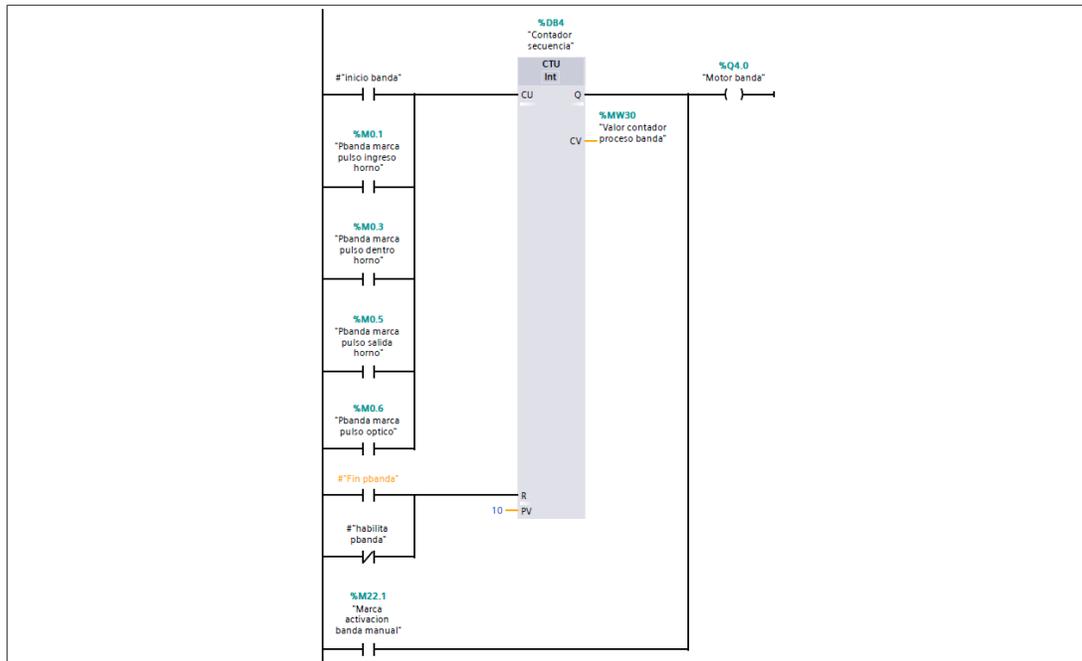
Segmento 4: Sensores



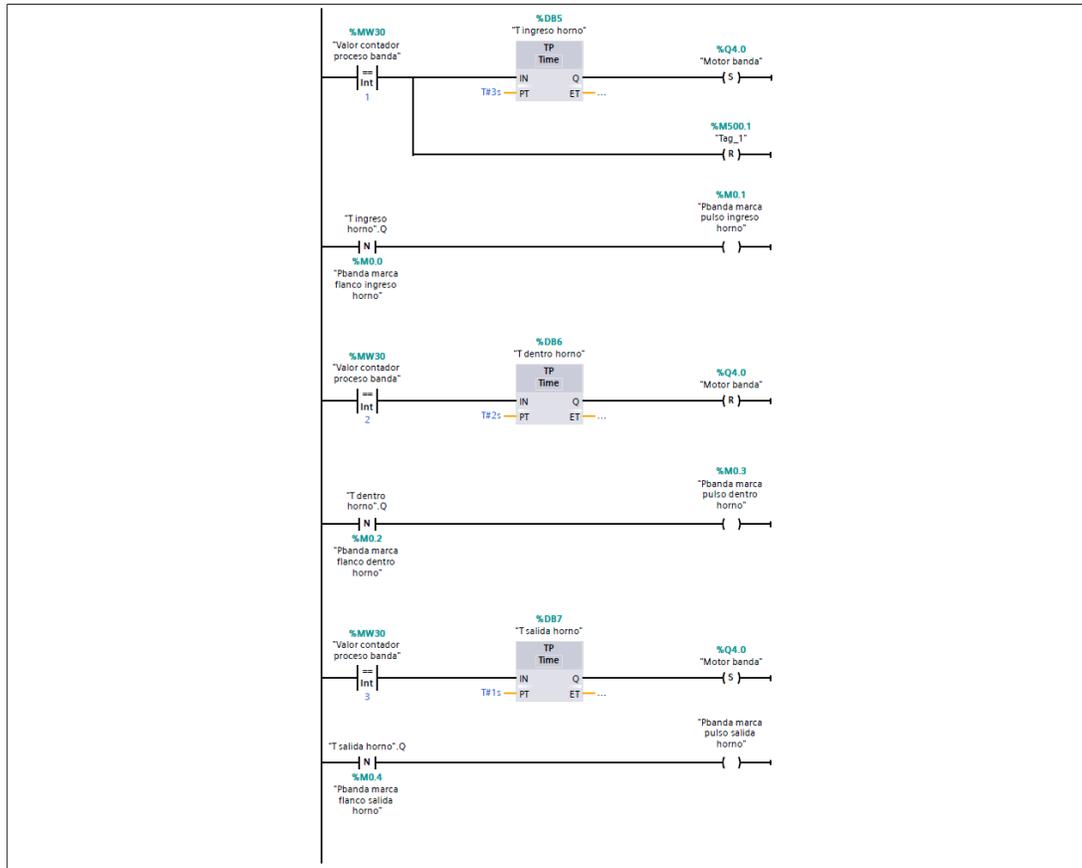


PROCESO BANDA

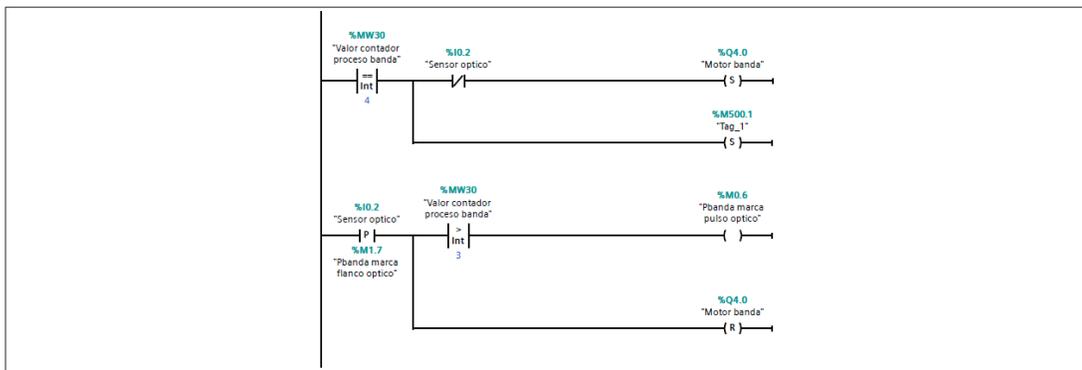
Segmento 1: Contador de secuencia proceso banda



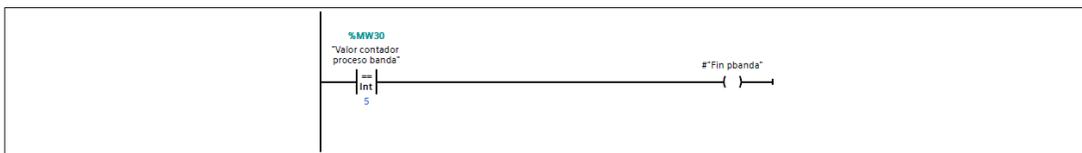
Segmento 2: Tiempo horno



Segmento 3:

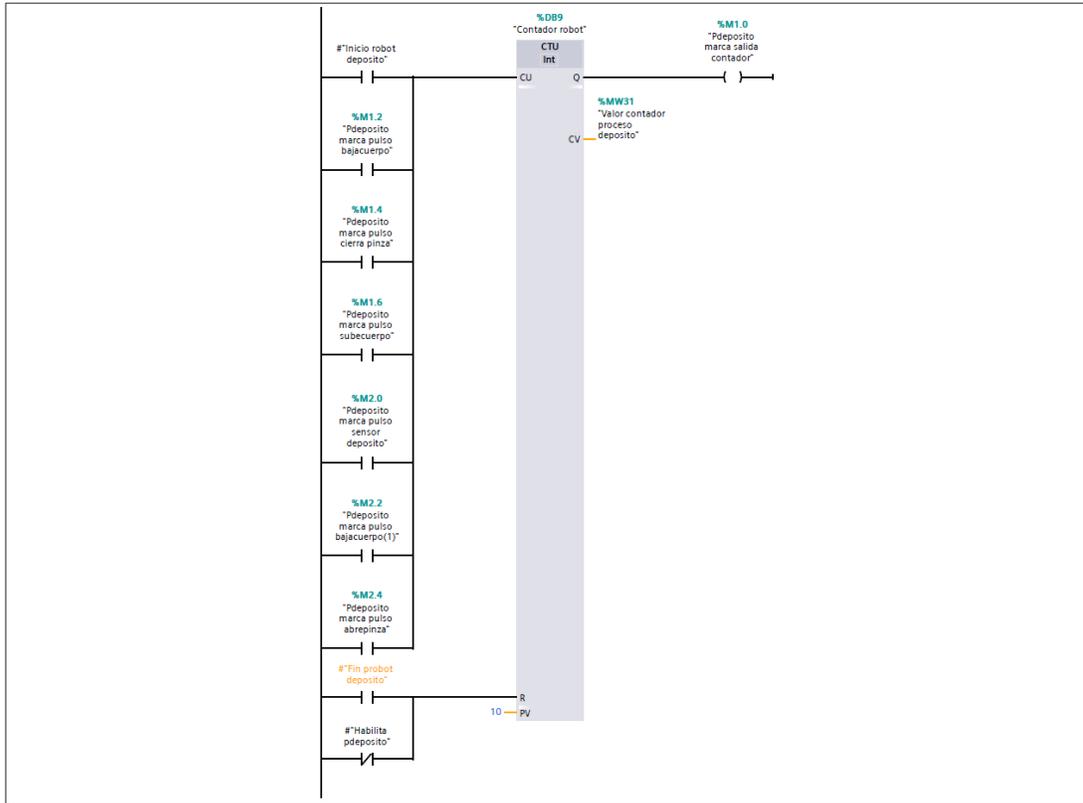


Segmento 4:

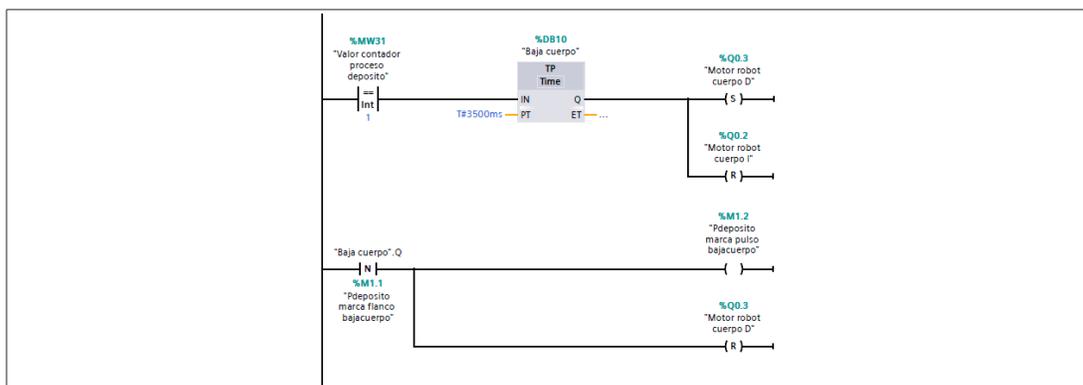


ROBOT DEPOSITO

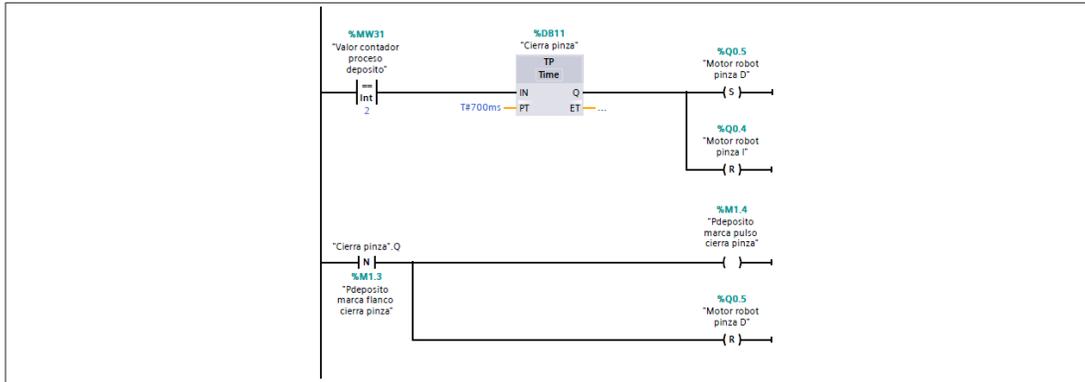
Segmento 1: Contador de secuencia robot deposito



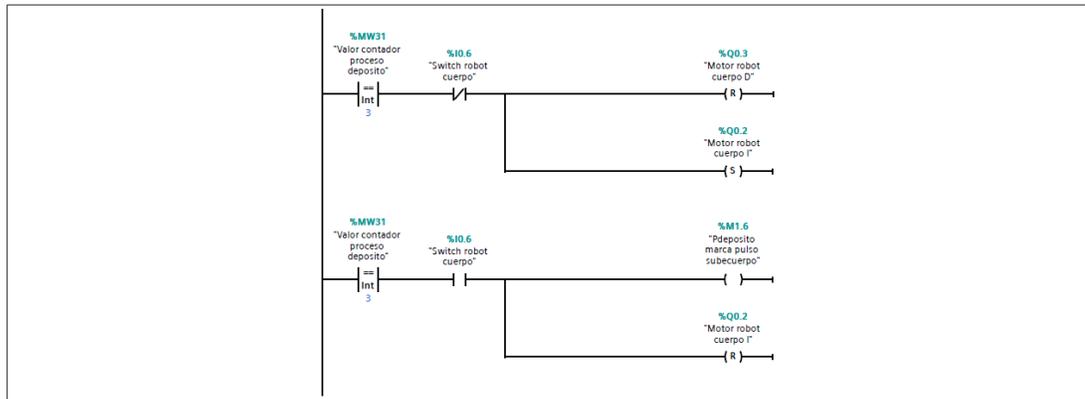
Segmento 2: Movimiento baja cuerpo



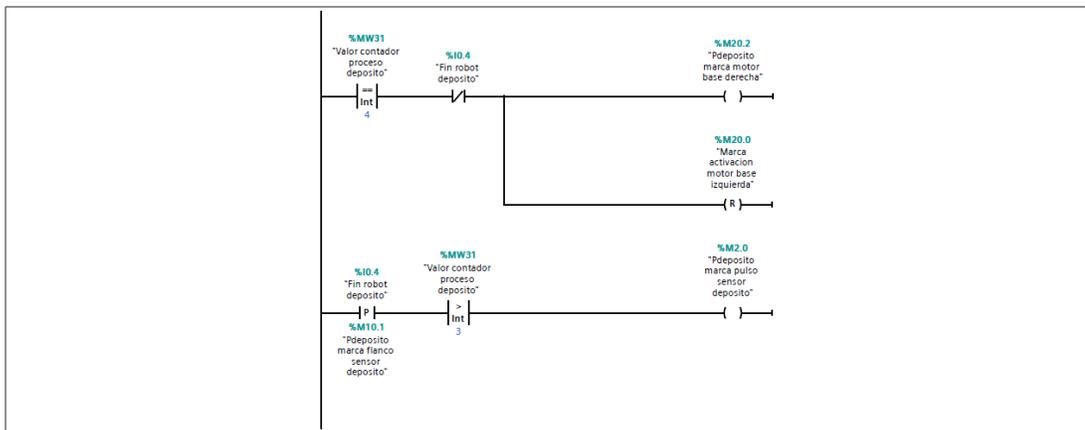
Segmento 3: Movimiento cierra pinza



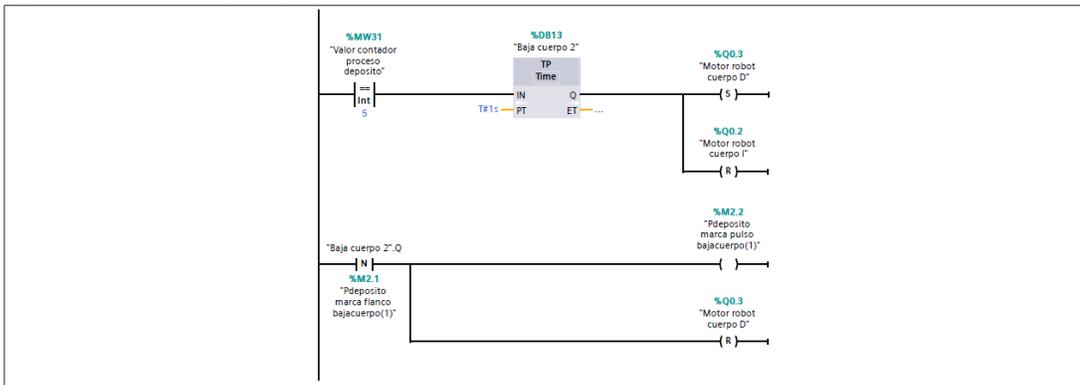
Segmento 4: Movimiento sube cuerpo



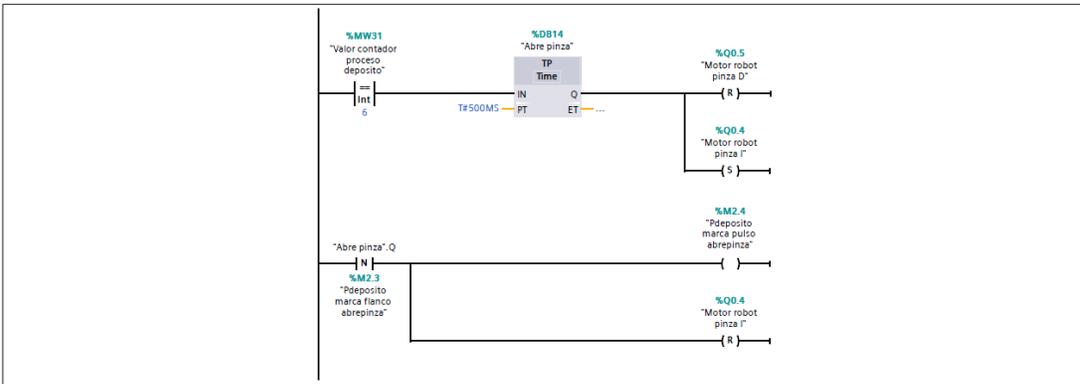
Segmento 5: Giro de robot derecha



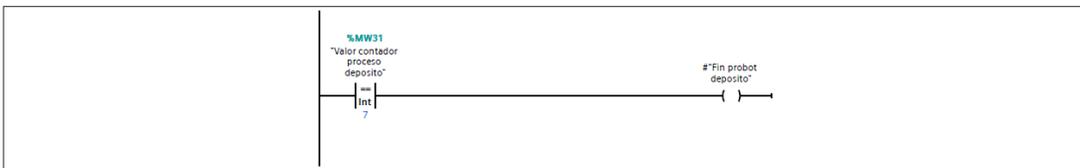
Segmento 6: Movimiento baja cuerpo



Segmento 7: Movimiento abre pinza

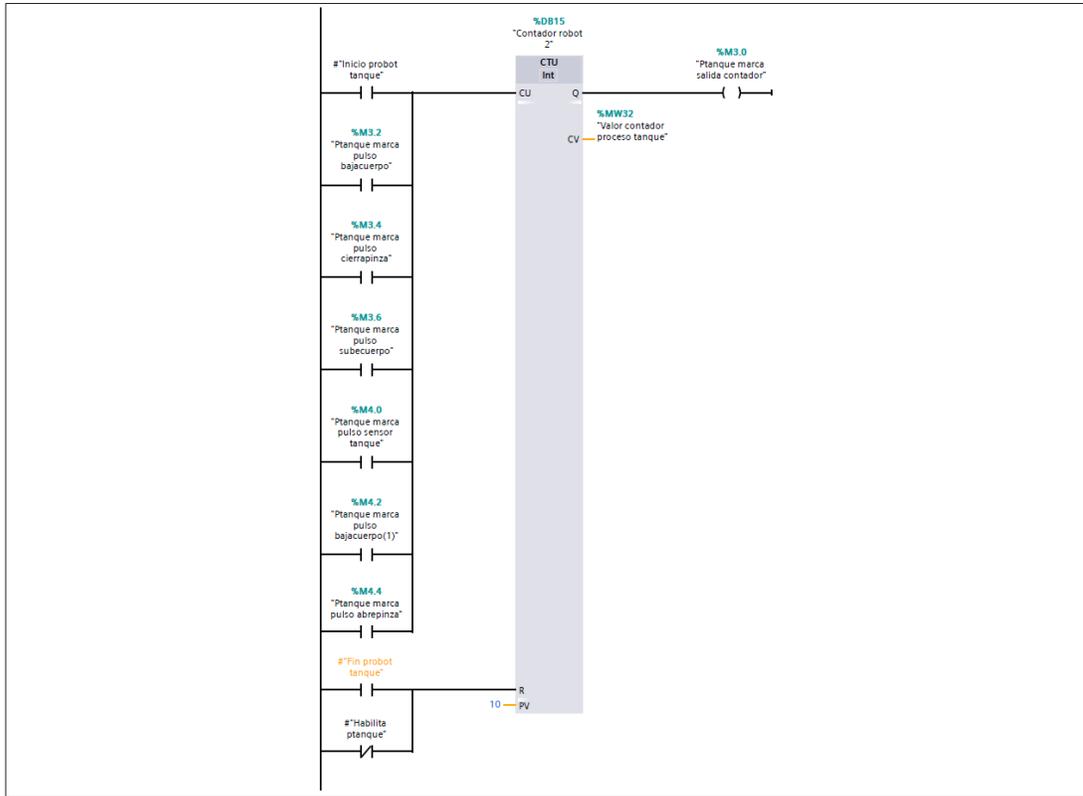


Segmento 8: Fin de secuencia

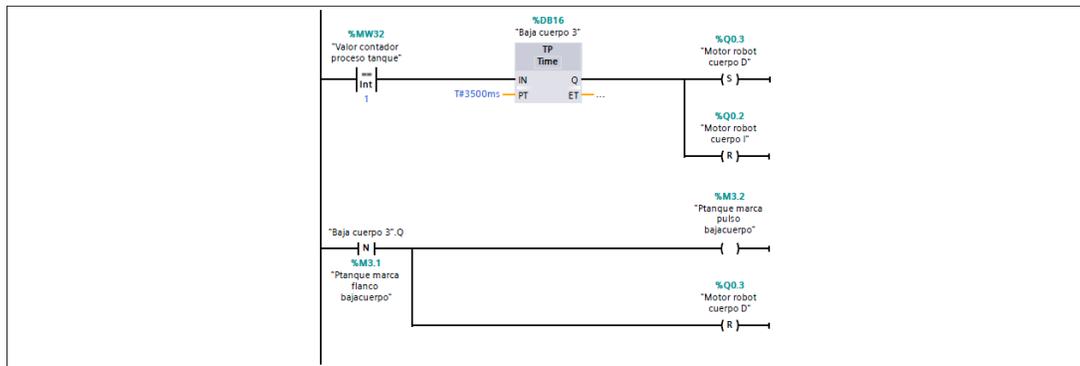


ROBOT TANQUE

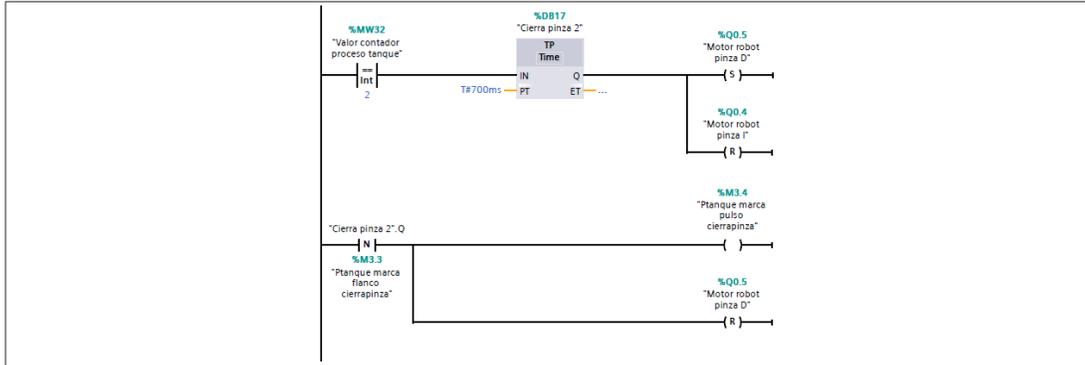
Segmento 1: Contador de secuencia robot tanque



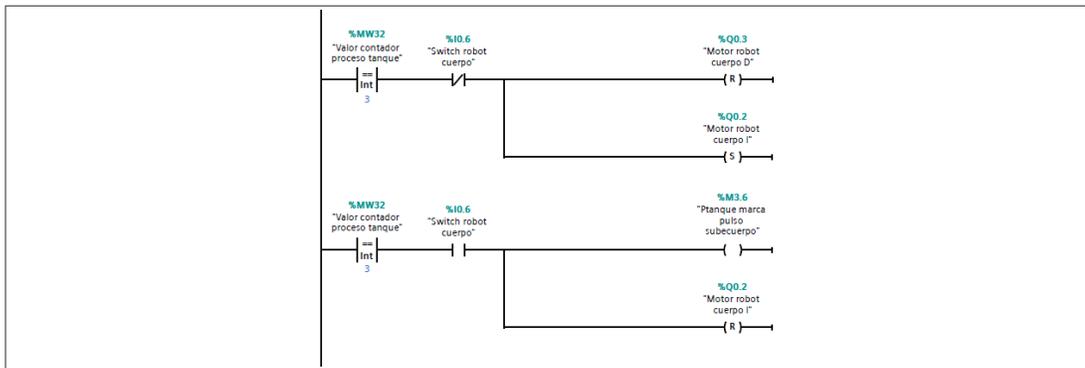
Segmento 2: Movimiento baja cuerpo



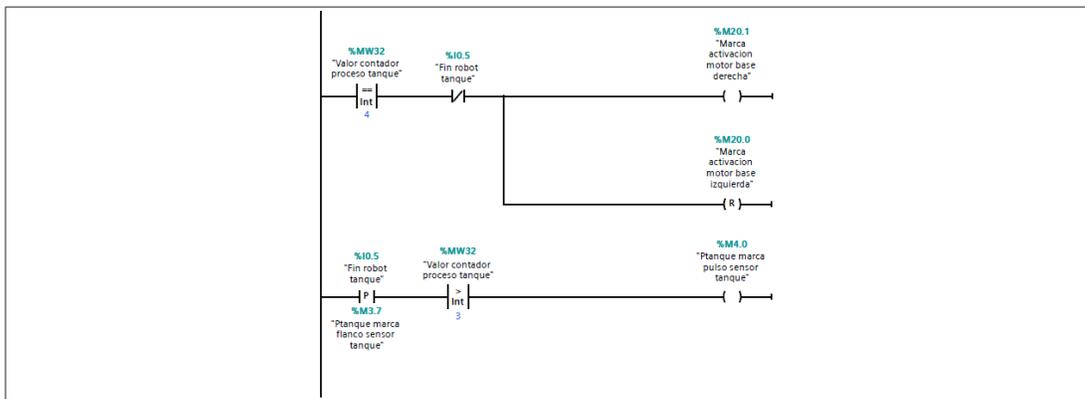
Segmento 3: Movimiento cierra pinza



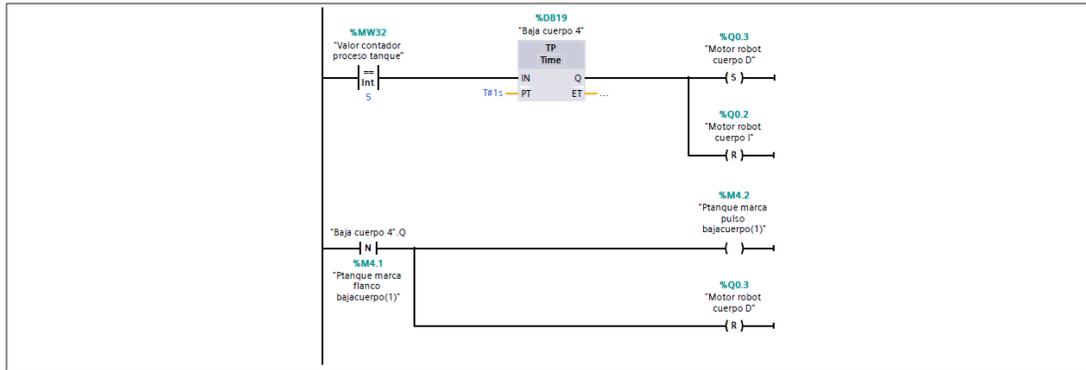
Segmento 4: Movimiento sube cuerpo



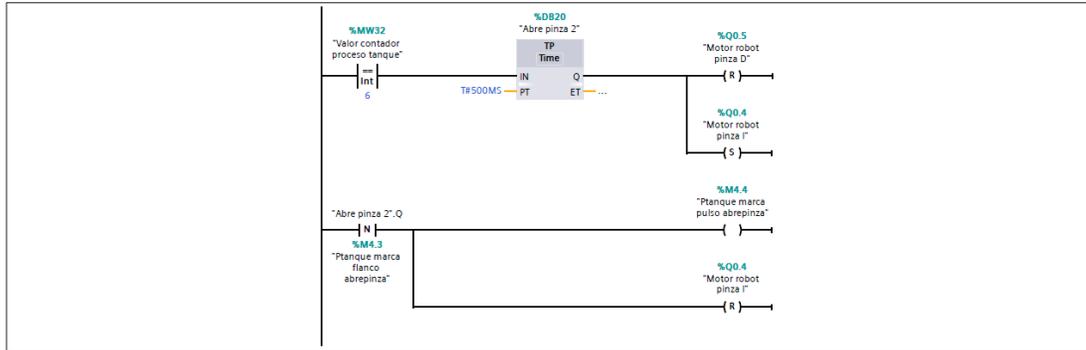
Segmento 5: Giro de robot derecha



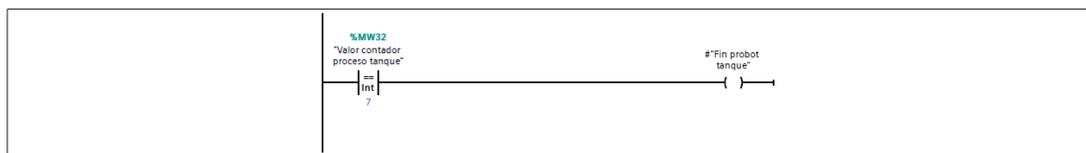
Segmento 6: Movimiento baja cuerpo



Segmento 7: Movimiento abre pinza

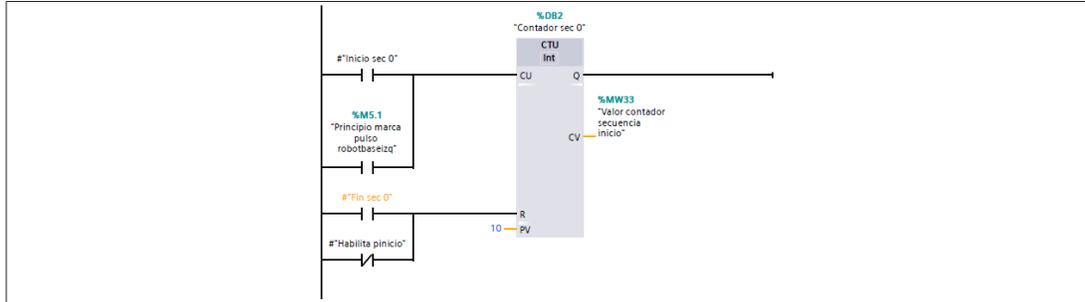


Segmento 8: Fin de secuencia

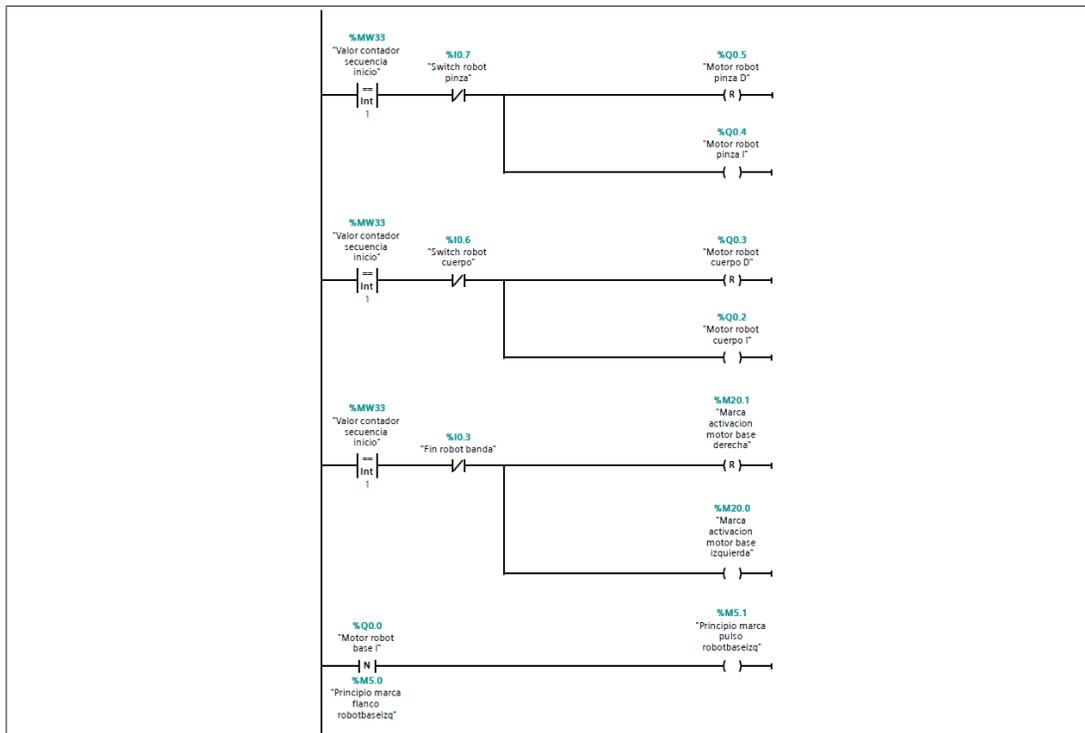


BLOQUE SECUENCIA INICIO

Segmento 1: Contador de secuencia de inicio



Segmento 2: Movimientos de robot iniciales



Segmento 3: Fin de secuencia inicio

