

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE CUENCA**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

*Trabajo de titulación previo  
a la obtención del título de  
Ingeniero Electrónico*

**PROYECTO TÉCNICO:**

**“DESARROLLO DE APLICACIONES DE MONITOREO Y  
CONTROL BASADAS EN IOT A TRAVÉS DE LA PLATAFORMA  
UBIDOTS. APLICACIONES A SISTEMAS DE  
AUTOMATIZACIÓN BAJO ENTORNOS DE  
SIMULACIÓN”**

**AUTORES:**

BRIAN ALEXANDER ESPINOSA APOLO  
MARCO ENRIQUE ORELLANA GUAYAS

**TUTOR:**

ING. JULIO CÉSAR ZAMBRANO ABAD, MSC.

CUENCA - ECUADOR

2021

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, Brian Alexander Espinosa Apolo con documento de identificación N° 0705756088 y Marco Enrique Orellana Guayas con documento de identificación N° 0104813720, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: **“DESARROLLO DE APLICACIONES DE MONITOREO Y CONTROL BASADAS EN IOT A TRAVÉS DE LA PLATAFORMA UBIDOTS. APLICACIONES A SISTEMAS DE AUTOMATIZACIÓN BAJO ENTORNOS DE SIMULACIÓN”**, mismo que ha sido desarrollado para optar por el título de: *Ingeniero Electrónico*, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, febrero del 2021



Brian Alexander Espinosa Apolo  
C.I. 0705756088



Marco Enrique Orellana Guayas  
C.I. 0104813720

## CERTIFICACIÓN

Yo, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **“DESARROLLO DE APLICACIONES DE MONITOREO Y CONTROL BASADAS EN IOT A TRAVÉS DE LA PLATAFORMA UBIDOTS. APLICACIONES A SISTEMAS DE AUTOMATIZACIÓN BAJO ENTORNOS DE SIMULACIÓN”**, realizado por Brian Alexander Espinosa Apolo y Marco Enrique Orellana Guayas, obteniendo el *Proyecto Técnico*, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, febrero del 2021



Ing. Julio César Zambrano Abad

C.I. 0301489696

## DECLARATORIA DE RESPONSABILIDAD

Nosotros, Brian Alexander Espinosa Apolo con documento de identificación N° 0705756088 y Marco Enrique Orellana Guayas con documento de identificación N° 0104813720, autores del trabajo de titulación **“DESARROLLO DE APLICACIONES DE MONITOREO Y CONTROL BASADAS EN IOT A TRAVÉS DE LA PLATAFORMA UBIDOTS. APLICACIONES A SISTEMAS DE AUTOMATIZACIÓN BAJO ENTORNOS DE SIMULACIÓN”**, certificamos que el total contenido del *Proyecto Técnico*, es de nuestra exclusiva responsabilidad y autoría.

Cuenca, febrero del 2021



Brian Alexander Espinosa Apolo  
C.I. 0705756088



Marco Enrique Orellana Guayas  
C.I. 0104813720

## **AGRADECIMIENTOS**

Agradezco en primer lugar a Dios que me ha permitido culminar mi formación académica.

A mi madre Bertha Marieta Apolo Encalada quien ha sido mi pilar fundamental en mi vida que gracias a sus consejos sus enseñanzas su apoyo incondicional me ha permitido culminar mis estudios Universitarios

A mi Padre Wilman Espinosa Puglla que me ha brindado su apoyo incondicional que sin la ayuda ni motivación me hubiese sido difícil culminar mi carrera de igual manera a y mis hermanos Ronald y Dayana, que han estado siempre presente en la buenas y en las malas

Agradezco a mi hija Briana Espinosa y a mi hijo Izan Espinosa que a pesar de la distancia con una llamada me daban más fuerzas para culminar mis estudios son mi vida hijos míos.

A nuestro tutor Ing. Julio Zambrano que nos brindó todo su apoyo, su conocimiento y paciencia durante la realización de todo nuestro proyecto técnico.

**Brian Alexander Espinosa Apolo.**

## **DEDICATORIAS**

Dedicó esta tesis con mucho cariño a mis padres Wilman Espinosa y Bertha Apolo, les pertenece a ustedes este triunfo por ser unos padres que nunca me dieron la espalda siempre me guiaron en el camino del bien me forjaron por ser un hombre con metas muy elevadas, no me alcanzara la vida por agradecerles el esfuerzo que hicieron para llegar a la meta tan anhelada. Este triunfo también le dedico a mis dos hijos Briana e Izan son mi motor de mi vida.

Este meta también se la dedico a una persona que fue muy especial me aconsejo en todo momento su anhelo era verme alcanzar mis metas, pero por cosas de la vida ya no te encuentras con nosotros ahora está disfrutando en las sendas de nuestro padre celestial, gracias por ser esa tía y a la vez una madre que estuvo conmigo en las buenas y en las malas gracias tía Hilda Bravo Apolo este logro también es suyo.

**Brian Alexander Espinosa Apolo**

## **AGRADECIMIENTOS**

Hoy estoy feliz y agradecido con Dios por cada día que me permite despertar y disfrutar de todas las bendiciones que me brinda, estoy agradecido por los aprendizajes que me envía cada día por que me ha enseñado que la vida a su lado y por su camino siempre es mejor.

Agradezco también a todas las personas que en algún momento formaron parte de mi vida en todo el transcurso de la universidad y me brindaron sus conocimientos y amistad, gracias a todos los profesores que con paciencia supieron duplicar su conocimiento para cada uno de los estudiantes que cursamos y siguen cursando hasta el día de hoy.

Agradezco por esos años increíbles que solo me los pudo dar la universidad, con todas las aventuras, amistades y sobre todo encontrar esos hermanos y hermanas que no son de sangre, pero son los que elegí.

Agradezco por la familia que tengo, ellos se encargaron de sembrar en mi esa semilla que libero a una persona que ama y sobre todo admira a cada ser humano. No tuve la infancia que muchos, pues no tenía lo que más anhelaba en ese momento, mis padres siempre me inculcaron, “si quieres algo trabaja por ello”; así que crecí y empecé a luchar por lo que quiero.

Agradezco al universo por enviarme tantas oportunidades y sobre todo enseñarme más caminos de brindarle a mi familia lo que se merece, gracias por enviarme las inversiones en estos años a mi vida.

A nuestro tutor Ing. Julio Zambrano quien supo brindarnos con mucha paciencia su apoyo y conocimientos, también nos enseñó que atrás de todo ingeniero existe una persona amable y con todas las ganas de ayudar a todos, ¡Gracias!

**Marco Orellana Guayas**

## **DEDICATORIAS**

Esta tesis va dedicada con todo el cariño para, mi yo de hace 10 años atrás, cuando empecé este camino existieron muchos momentos en donde pude dejarlo todo, pero seguí firme y luché por culminar este capítulo de mi vida.

También lo dedico a mis padres quienes con todo el amor me apoyaron desde un inicio, me brindaron la educación inicial, y una parte de la universidad. Seres humanos que no se equivocaron al decirme “Si pierdes una materia, te pagas tú la universidad”; recuerdo esto con tanto amor, porque me enseñaron algo más valioso, todo lo que no te cuesta no lo valoras y no lo cuidas, lo dedico a mi hermana quien me brindo sus mejores años de vida, sé que siempre le digo que me debe su infancia, pero es al contrario yo le debo mi infancia.

Va dedicado también para mis suegros, quienes confiaron en mí, al entregarme a su hija, sinceramente no recuerdo haberles prometido terminar la universidad, pero siempre me apoyaron y dieron ánimos de continuar.

Dedico este logro en mi vida a mi esposa e hijos, quienes estuvieron a mi lado luchando, dándome aliento desde el primer día en la universidad, formamos una familia que apenas va 8 años junta, pero tenemos algo importante que hemos logrado juntos, transmitir el amor a nuestros hijos y entre nosotros, te dedico a ti que jamás perdiste la fe en mí y siempre me recuerdas que tienes al mejor hombre a tu lado.

**Marco Orellana Guayas**

# ÍNDICE GENERAL

AGRADECIMIENTOS.....	I
DEDICATORIAS.....	II
AGRADECIMIENTOS.....	III
DEDICATORIAS.....	IV
ÍNDICE DE FIGURAS.....	IX
ÍNDICE DE TABLAS.....	XII
RESUMEN.....	XIII
INTRODUCCIÓN.....	XIV
ANTECEDENTES DEL PROBLEMA DE ESTUDIO.....	XVI
JUSTIFICACIÓN.....	XVIII
OBJETIVOS.....	XVIII
CAPÍTULO 1: INTERNET DE LAS COSAS “IoT”.....	1
1.1 ¿A qué nos referimos con el término “iot”?.....	1
1.2 Componentes de la Arquitectura IoT.....	2
1.3 Industria 4.0.....	3
1.3.1 Adaptación exitosa del sistema de la industria 4.0:.....	4
1.3.2 implementación exitosa del sistema de la industria 4.0.....	4
1.4 Objetivo y motivación del presente trabajo.....	5
CAPÍTULO 2: FACTORY I/O.....	6
2.1 DESCRIPCIÓN FACTORY I/O.....	6

2.2	COMPONENTES DE “FACTORY I/O” .....	6
2.2	NAVEGACIÓN .....	8
2.3	REQUISITOS DEL SISTEMA Y VERSIONES .....	11
2.4	ESCENAS .....	12
2.5	CONTROLADORES e/s .....	16
2.6	Etiquetas .....	18
2.7	CONTROL AUTOMÁTICO .....	18
CAPÍTULO 3: UBIDOTS.....		19
3.1	CREACIÓN DE UNA CUENTA.....	21
3.2	Dispositivos.....	21
3.3	Variables.....	22
3.4	Dashboard (Tablero).....	22
3.5	Widgets.....	22
3.6	Eventos.....	22
3.7	Creación de dispositivos.....	22
3.8	Creación de widgets.....	23
3.9	Creación de Variables.....	25
3.10	Compartir en Ubidots.....	26
3.11	API HTTP de Ubidots.....	26
3.12	Token.....	28
CAPÍTULO 4: NODE RED .....		29
4.1	Nodos.....	30
4.2	Flujos (Flows).....	31
4.3	Alambres (Wires) .....	33
4.4	Publicación .....	33
4.5	Administración de paleta.....	34

4.6	Instalación de Nodos .....	34
4.7	Paletas.....	37
CAPITULO 5: INTEGRACION DE HERRAMIENTAS INFORMATICAS Y DESARROLLO DE APLICACIONES .....		41
5.1	PLCSim Advance .....	41
5.2	Factory I/O. ....	43
5.2.1	Configuración.....	43
5.2.2	Conexión con el Automata.....	46
5.2.3	Definición de entradas y salidas.....	46
5.3	TIA PORTAL.....	47
5.3.1	Configuración INICIAL.....	47
5.3.2	ADMINISTRACIÓN DE VARIABLES. ....	51
5.3.3	Conexión CON plcsim .....	53
5.4	Node Red.....	54
5.4.1	Instalación. ....	54
5.4.2	Comunicación Node Red con PLC mediante nodos .....	60
5.4.3	Configuración nodos para comunicación PLC .....	60
5.4.4	Comunicación Node Red con Ubidots mediante nodos.....	62
5.4.7	Comunicación Ubidots con PLC mediante nodos .....	65
5.5	Ubidots. ....	68
5.5.1	Obtención Token.....	68
5.5.2	Creación de Dispositivos.....	69
5.5.3	Creación de Variables. ....	70
5.5.4	Agregar Tablero. ....	71
5.6	Videos Guía .....	73
CAPITULO 6: DESARROLLO DE APLICACIONES. ....		73
6.1	Banda Transportadora. ....	75
6.2	Clasificación de objetos.....	78
6.3	Clasificación mediante Pick Up .....	80
6.4	Traslado y Clasificación de objetos.....	81

6.5	Controlador PID .....	83
CAPITULO 7: CONCLUSIONES Y RECOMENDACIONES .....		85
7.1	Conclusiones. ....	85
7.2	Recomendaciones. ....	86
REFERENCIAS BIBLIOGRÁFICAS .....		88
APÉNDICE .....		91

## ÍNDICE DE FIGURAS

Figura 1.1 Arquitectura “IoT”.....	3
Figura 1.2 Marco de la industria 4.0 .....	5
Figura 1.3: Esquema de la Industria 4.0.....	6
Figura 2.2 Emisor de elementos en Factory I/O .....	8
Figura 2.3 Extractor de elementos en Factory I/O .....	8
Figura 2.4 Tipos de cámara.....	9
Figura 2.6 Control manual de las variables analógicas en Factory I/O. [2].....	18
Figura 2.7 Asignación de etiquetas en Factory I/O.....	19
Figura 3.1: Logo Ubidots.....	21
Figura 3.2: Interfaz de inscripción en Ubidots.....	21
Figura 3.3: Interfaz creación de dispositivos en Ubidots.....	23
Figura 3.4: interfaz de Ubidots .....	23
Figura 3.5: Creación de un variable dentro de Ubidots .....	24
Figura 3.6: Disposición variables tablero Ubidots.....	25
Figura 3.7: Cuadro de diálogo para selección de tipos de variables dentro de Ubidots .....	25
Figura 3.8: Envío y recepción de datos mediante una variable.....	25
Figura 3.9: Link para compartir iframe de la variable en Ubidots.....	26
Figura 3.10: Formato JSON.....	27
Figura 3.11 Obtención del Token.....	28
Figura 4.2: Flujo de programación en Node-Red.....	30
Figura 4.3: Clases de Nodos.....	31
Figura 4.4: Propiedades y activación de un flujo.....	32
Figura 4.5: Función de flujos entradas, intermedios y salidas.....	33
Figura 4.6 Conexión de Nodos mediante alambres (Wires).....	33
Figura 4.7 Publicación de flujo.....	33
Figura 4.8 Cuadro de diálogo que visualiza los nodos existentes.....	34
Figura 4.9 Cuadro de diálogo para la instalación de nuevos nodos.....	35
Figura 4.10: Cuadro de dialogo para configuración de la comunicación con un PLC Siemens.....	36

Figura 4.11: Cuadro de dialogo para configuración del bloque de comunicación de salida. ....	36
Figura 4.12: Cuadro de dialogo para configuración del bloque de comunicación de entrada. ....	37
Figura 5.1 Cuadro de Dialogo para creación de nuevo PLC virtual. ....	42
Figura 5.2 Asignación de nombre y dirección IP.....	42
Figura 5.3 Cuadro de dialogo, advertencia nueva IP detectada. ....	42
Figura 5.4 Nuevo PLC virtual creado. ....	43
Figura 5.5 Configuración Inicial Factory IO.....	44
Figura 5.6 Definición Autómata, direccion IP. ....	44
Figura 5.7 Configuración terminales. ....	45
Figura 5.8 Dispositivo Creado. ....	45
Figura 5.9 Botón Conectar. ....	46
Figura 5.10 Conexión exitosa. ....	46
Figura 5.11 Asignación de variables del proceso. ....	47
Figura 5.12 Creación nuevo dispositivo.....	47
Figura 5.13 Menú desplegable para acceder a las Propiedades de un proyecto. ....	48
Figura 5.15 Cambiar Propiedades. ....	49
Figura 5.18 Creación Bloque de Datos. ....	51
Figura 5.22 Conexión Exitosa.....	54
Figura 5.23 Página Oficial Node-Red.....	55
Figura 5.24 Pagina de Descarga Node.js. ....	55
Figura 5.25 Dialogo de Instalación Node.Js. ....	56
Figura 5.26 Instalación Node-Red. ....	56
Figura 5.27 Comando para Iniciar Node-Red.....	57
Figura 5.28 Dirección IP necesaria para iniciar aplicación.....	57
Figura 5.29 Interfaz Node-Red. ....	58
Figura 5.30 Manage Palette.....	58
Figura 5.31 Nodo a instalar. ....	59
Figura 5.32 Nodo para Ubidots.....	60
Figura 5.33 Nodos comunicación PLC. ....	60
Figura 5.34 Agregar nuevo PLC. ....	61
Figura 5.35 Configuración Nodo. ....	61
Figura 5.36 Asignación de Variables a obtener. ....	62

Figura 5.37 Bloque Function. ....	63
Figura 5.38 Programación dentro del nodo Function.....	63
Figura 5.39 Configuración nodo Ubidots Out. ....	64
Figura 5.40 Configuración Ubidots In. ....	65
Figura 5.41 Nodo Split.....	66
Figura 5.42 Obtención dato recortado.....	66
Figura 5.43 Envío de datos hacia el PLC.....	67
Figura 5.44 Creación cuenta Ubidots.....	68
Figura 5.45 Obtención Token. ....	68
Figura 5.46 Dispositivos. ....	69
Figura 5.47 Creación nuevo dispositivo.....	70
Figura 5.48 Creación de Variables.....	70
Figura 5.49 Nombre API label.....	71
Figura 5.50 Widget.....	72
Figura 5.51 Tablero.....	72
Figura 6 Escenas .....	74
Figura 6.1 Escenas de la herramienta grafica Factory IO .....	74
Figura 6.3 Entorno de trabajo en la herramienta Grafica Factory IO .....	75
Figura 6.4 Selección del autómeta a usar.....	76
Figura 6.5 Configuración para la comunicación con el autómeta.....	76
Figura 6.6 Asignación de Variables.....	77
Figura 6.7 Comunicación Exitosa.....	77
Figura 6.20 Sorting Station.....	78
Figura 6.21 Sorting by Weight.....	78
Figura 6.22 Diseño practica 4 página web.....	79
Figura 6.23 Diseño practica 6 página Web.....	79
Figura 6.30 Espacio de Clasificación.....	80
Figura 6.31 Control pick up página web.....	81
Figura 6.40 Clasificador de objetos por su color. ....	82
Figura 6.41 Control página Web.....	82
Figura 6.50 Level Control.....	83
Figura 6.51 Entorno para el control de nivel.....	83
Figura 6.52 Pagina Web para el control y monitoreo del sistema de flujo .....	84

## ÍNDICE DE TABLAS

Tabla 2.1 Detalle de comandos para el uso de la cámara de orbital [ .....	9
Tabla 2.2 Navegación con la cámara Fly .....	10
Tabla 2.3 Navegación con la cámara en primera persona.....	11
Tabla 2.4 Requerimientos del sistema para Instalación. ....	11
Tabla 2.5 Descripción de Escenas Factory IO. ....	12
Tabla 2.6 Controladores E/S .....	16
Tabla 4.1 Descripción de nodos dentro de la paleta Node-red, sección function.[33] .....	37
Tabla 4.2 Descripción de nodos dentro de la paleta Node Red sección common. [33] .....	39
Tabla 4.3 Descripción de nodos dentro de la paleta Node Red sección sequence.[33] .....	40
Tabla 4.4 Descripción de nodos dentro de la paleta Node Red sección input.[33]....	40
Tabla 5.1 Links videos guía. ....	73

## RESUMEN

En este trabajo de titulación se presenta “Desarrollo de aplicaciones de monitoreo y control basadas en “IoT” a través de la plataforma Ubidots. Aplicaciones a sistemas de automatización bajo entornos de simulación”, se lo elaboro con el objetivo de contribuir a los estudiantes de la Universidad Politécnica S. en la carrera de Ingeniería Electrónica, en el año que se desarrolló, la emergencia sanitaria a nivel mundial en el cual las personas en general fueron confinadas a aislamiento, se abrió el nuevo camino de la era digitalización mediante internet, esto nos permite incursionar en la tecnología “IoT” .Se dividió en varias etapas primero se realizó un estudio de las herramientas Factory I/O que ofrece para la creación de sistemas de automatización, en base a este estudio se ha logrado general varios procesos virtuales, de igual manera se realizó un análisis de las herramientas de programación de Tía Portal donde se ha desarrollado programas para automatizar, posteriormente se realizó una investigación sobre Node-Red y sus modos de operar, Node red permite instalar de forma gratuita, las herramientas que son basadas en JavaScrip, Este sistema permite la creación rápida de aplicaciones, especialmente aplicaciones que se activan en un evento como las de “IoT”, presenta tres componentes: Panel de nodo, Panel de flujo y Panel de información y depuración. Para finalmente analizar las herramientas de Ubidots, utilizada para visualizar las variables en tiempo real, también nos permite modificar las variables del proceso desde cualquier parte del mundo; desde controlar el proceso de una empresa de clasificación, hasta una cafetera que se mantengan siempre llena mientras el usuario lo indique, los cambios se pueden realizar mediante internet atreves de la página web Ubidots. La integración de esto, se ha realizado con un manual de 8 practicas las cuales servirán para fomentar el aprendizaje de la materia de Automatización Industrial o de igual a la materia de Redes Industriales estas forman parte de la malla curricular en la carrera de Electrónica y automatización.

# INTRODUCCIÓN

El internet de las cosas tiene un papel muy importante en muchas industrias, permite que varios procesos se ejecuten de manera exitosa y su control a través de internet.

El presente documento está destinado a el desarrollo de guías prácticas para los estudiantes de la carrera de Ingeniería Electrónica. En el año 2020 la pandemia mundial obligo a muchas personas a usar la tecnología para la comunicación, esto afecto directamente al entorno universitario, la carrera de ingeniería electrónica en el área de automatización, la importancia de contar con prácticas en donde se pueda observar resultados visibles es indispensable para el aprendizaje por tal motivo se creó la necesidad de poder desarrollar las practicas desde casa.

En el capítulo 1 se presenta la arquitectura general, formando un modelo de referencia de “IoT” que se encuentra compuesto de cuatros capas como la técnica de trabajar en función de administrar y brindar seguridad, describiendo los factores del “IoT”

En el capítulo 2 se expone la plataforma Factory IO los componentes básicos que permiten la simulación dentro de la plataforma; además, se presenta los controles de navegación con los cuales el usuario puede moverse dentro de la aplicación, cuenta con una explicación de las escenas pre diseñadas y los requerimientos del sistema necesario para su funcionamiento.

En el capítulo 3 se describe la página web Ubidots y sus diferentes subpáginas. Se detalla cómo crear una cuenta gratuita como también la manera de crear dispositivos y bloques de visualización dentro de tableros; la creación de nuevas variables y la obtención del “Token” que permite la comunicación con la página web esta descrita en este capítulo.

En el capítulo 4 se desarrolla la explicación sobre Node red sus componentes con sus diferentes bloques de programación, además se muestra los bloques con mayor uso durante el desarrollo de las guías prácticas.

En el capítulo 5 se presenta la forma de vinculación entre las herramientas Factory IO con Tia Portal, así como la comunicación entre Tia Portal y Node red además se muestra la manera de comunicar bidireccionalmente con la plataforma Ubidots.

En aras del objetivo mencionado se desarrolló ocho guías prácticas, donde el estudiante podrá trabajar desde casa en un ambiente virtual 3D brindando la sensación de realizar prácticas como en cualquier laboratorio. Es necesario la participación de un autómata, en aras de obtenerlo dentro de cada practica se usa la aplicación “PLC SIM Advance” el cual permite simular un autómata con todas sus características, el ambiente virtual 3D que seleccionamos es “Factory I/O” por su veracidad y cómoda conexión con la plataforma “TIA portal”. Para él envió de datos a la nube Node-Red es la aplicación que nos brinda la comunicación entre Factory I/O y la página Web Ubidots mediante programación de bloques.

Palabras Clave.

Ubidots, Node Red, Tia Portal, PLCSimAdvance, IoT.

# **ANTECEDENTES DEL PROBLEMA DE ESTUDIO**

El presente proyecto se propone con la premisa de investigar y desarrollar aplicaciones basadas en Internet Industrial de las Cosas “IoT” para la creación de sistemas de monitoreo remoto a través de la web. El producto de esta investigación permitirá a los estudiantes fortalecer los conocimientos teóricos y prácticos adquiridos en las materias de redes industriales y SCADA y automatización industrial. Mediante este conocimiento previo buscamos combinar la Tecnología de “IoT” con PLCSIM Advance y crear procesos industriales virtuales que puedan ser monitorizados y controlados en tiempo real mediante la herramienta Ubidots y el entorno de simulación Factory I/O.

Hoy por hoy, el nuevo camino determinado a las comunicaciones industriales se vuelve importante al incremento de una nueva era en los procesos de producción para las compañías. el avance tecnológico y la información encontrada forjan un papel importante en la nueva revolución industrial, también generalmente conocido como la Industria 4.0 ofrece una forma de organizar procesos de rendimiento esto permitirá que las empresas obtengan de manera eficaz de responder las solicitudes de los usuarios. Para elaborar el presente proyecto se tomará referencias de algunos informes técnicos relacionados a nuestro tema a desarrollar el Proyecto.[1]

La industria 4.0 toma decisiones y acciones, mientras monitorean y controlan procesos físicos. Cada sistema distribuye un determinado proceso comunicándose de manera bidireccional mediante las redes de comunicación, convirtiéndose en internet de las cosas.[2]

Los procesos industriales inteligentes se fijan como procesos adaptables debido a que todos sus sistemas de producción se encuentran conectados mediante redes localizadas con distintivos de comunicación, apto para manejar flujos constantes de datos de las operaciones además de la información en empresas gracias a la Tecnología de la Internet Industrial de las Cosas “IoT”.[3]

En conclusión, la investigación se destaca que en los últimos años la “IoT” continua ganando más fuerza al momento de observar que se tiene oportunidades optimistas para construir sistemas fuertes y aplicaciones con nuevos rendimientos para la industria. Ahora, los sistemas industriales se vuelven más complejos, por lo tanto, la seguridad de cada operación debe ir en aumento afianzando la fiabilidad y confiabilidad de las operaciones industriales.[4]

Con el presente proyecto nosotros implementaremos una guía para los estudiantes, para que puedan realizar monitoreo en tiempo real utilizando procesos virtualizados que serán desarrollados en contextos industriales utilizando la herramienta informática “Factory IO”. Hoy en día uno de los servicios indispensables de las personas es el internet, y pensar en todo lo que se podrá lograr con estos conocimientos permitiendo que el estudiante pueda introducirse en el campo del “IoT” y su conocimiento pueda expandirse a nuevos procesos, enfocados al mundo de hoy y su tecnología.

# **JUSTIFICACIÓN**

La tecnología IOT tiene una gama de aplicaciones en diferentes secciones que no se ha logrado explotar, por ejemplo el area de la industria, particularmente en la automatización y control de procesos, donde es indispensable saber nuevas formas de monitoreo y supervisión de los mismos, por tal motivo es necesario conocer más procedimientos, que empleen el internet industrial de las cosas “IoT” y la WEB para alcanzar a controlar dispositivos y al mismo tiempo generar datos por ellos mismo.

Dentro de los objetivos de la carrera de electrónica y automatización, se encuentra la formación de excelentes profesionales para tener la capacidad de solucionar problemas en el entorno que se desarrolle, por lo tanto, estamos llevando a cabo este proyecto con un enfoque investigativo y práctico para el monitoreo de variables en tiempo real en la web mediante el internet industrial de las cosas y visualizando su comportamiento en un entorno gráficos en 3D.

El motivo de este proyecto es mostrar las ventajas que hoy en día existen gracias a la tecnología “IoT” donde se puede desarrollar aplicaciones en diferentes ámbitos como el transporte, salud, domótica, robótica y en la automatización industrial. El proyecto se ejecutará con el propósito de ayudar a todos los estudiantes de la carrera de ingeniería electrónica para que puedan desarrollar desde su hogar todas las prácticas que tenga que ver con la automatización industrial por lo tanto se está proponiendo una guía de prácticas en la cual el estudiante se le facilite poner en práctica sus conocimientos y monitorear cada práctica ejecutada en tiempo real.

## **OBJETIVOS**

La formación del estudiante en habilidad digital y automatización industrial, se favorece enormemente si desde el inicio conocen el funcionamiento interno a la vez

que aprenden las bases de programación en nuevos y actuales lenguajes de software libres. El objetivo de este proyecto será para el beneficio de todo el alumnado de la carrera de Ingeniería Electrónica de la Universidad Politécnica S. Ya que les facilitara automatizar y monitorear todas las practicas a realizar.

Por la circunstancia dada por la emergencia sanitaria el último año se ha visto indispensable el uso prolongado de entornos virtuales, por lo que hemos investigado la manera en la cual todos los estudiantes puedan realizar las prácticas desde su hogar y no exponerse a riesgos de esta emergencia sanitaria, por lo tanto, los educadores podrán verse beneficiados junto a sus alumnos obtenido la facilidad de realizar y revisar prácticas en base al aprendizaje obtenido en clases.

### **Objetivo General**

- Desarrollar aplicaciones de monitoreo y control basadas en “IoT” y una plataforma de simulación de procesos, en aras de brindar a los estudiantes de la carrera de electrónica y automatización una guía de prácticas para entrenarse en el monitoreo remoto de variables a través de la web.

### **Objetivos Específico**

- Recopilar información sobre el internet de las cosas “IoT” y el manejo del programa informático UBIDOTS, en aras de sentar las bases de conocimiento para el desarrollo del proyecto.
- Investigar el manejo de la herramienta informática “Factory I/O” para crear procesos virtuales en 3D que serán monitorizados remotamente a través de la web.
- Simular y comprobar la correcta comunicación entre UBIDOTS, la herramienta informática “Factory I/O”y PlcSim Advance.
- Elaborar 8 guías prácticas incluidos videos explicativos para que el estudiante pueda desarrollar prácticas de laboratorio orientadas al monitoreo remoto de procesos industriales utilizando herramientas “IoT”.

### **Delimitaciones**

Iniciaremos en la investigación, donde logremos encontrar la información necesaria para llevar a cabo las guías de práctica, que están enfocadas a la interconexión de

dispositivos mediante la plataforma UBIDOTS, “Factory I/O” y las herramientas informáticas “PlcSim Advance” “TIA Portal”.

# CAPÍTULO 1: INTERNET DE LAS COSAS “IOT”

## 1.1 ¿A QUÉ NOS REFERIMOS CON EL TÉRMINO “IOT”?

“IoT” es la abreviatura de “Internet of Things” que en castellano significa internet de las cosas. Básicamente, “IoT” es la integración o comunicación de dispositivos tecnológicos los cuales pueden estar interconectados a través del internet o redes fijas. En el contexto industrial estos dispositivos tecnológicos principalmente pueden ser sensores, actuadores y controladores de proceso. En otras palabras, “IoT” también puede entenderse como una plataforma donde se distribuya información y datos específicos. [5].

Una arquitectura de “IoT” consta de tres capas principales:

- *Capa Física*

En la capa física, se trabajan conjuntamente con los sensores y los microcontroladores, a su vez desarrollan la parte más importante del internet de las cosas “IoT”. La detección de cambios en un objeto permite tomar capturas de los datos más importantes en tiempo real.

- *Capa de Conectividad*

La capa de conectividad, el objetivo primordial conectarse a otras cosas inteligentes, dispositivos de red y servidores. También una de las principales funciones es transmitir y procesar datos de sensores

- *Capa Digital*

La capa digital tiene la capacidad de almacenar y analizar grandes cantidades de datos que llegan directamente de la capa de conectividad. Empleando demasiadas tecnologías como bases de datos y módulos de procesamiento de big data.

## 1.2 COMPONENTES DE LA ARQUITECTURA IOT

La tecnología “IoT”, cada vez ofrece más opciones de soluciones, aplicaciones y plataformas en la nube, para seleccionar una u otra es necesario entender su estructura a continuación exponemos su estructura.[26]

- **Plataforma Cloud**

Está conformada por una serie de plataformas de servicio, estas son usadas para desarrollar soluciones en la nube por desarrolladores. Incluyen servicios de almacenamiento y seguridad y muchas plataformas son públicas como Microsoft Azure.[26]

- **Plataforma Internet de las Cosas.**

Esta es la base para que los dispositivos se conecten y comuniquen entre sí, así se construye el ecosistema IoT [26]

- **Capa de Dispositivos.**

Esta capa está conformada por varios dispositivos y estos dispositivos tienen comunicación, directa o indirecta con internet. Estos dispositivos tienen un identificador único grabado y un token.[28]

- **Capa de Comunicaciones.**

Esta capa permite la conectividad de dispositivos, estos dispositivos cuentan con protocolos como HTTP y MQTT.

HTTP es el protocolo más conocido y muchas librerías lo soportan, este protocolo es simple y se basa en texto, MQTT este protocolo se basa en mensajería publish y se basa en un modelo tipo bróker.

Los componentes de la arquitectura de figura 1.1, contienen en primer lugar actuadores para realizar cambios y sensores que recogen datos. Puertas de enlace para procesar y mover datos entre la nube y el usuario. Contienen una pasarela que permite almacenar los datos. Una Base de datos que permite almacenar datos y variables. Análisis de datos para el proceso de datos realizado por el usuario manualmente. Aplicaciones de usuario que permiten visualizar y controlar los dispositivos conectados.[28]

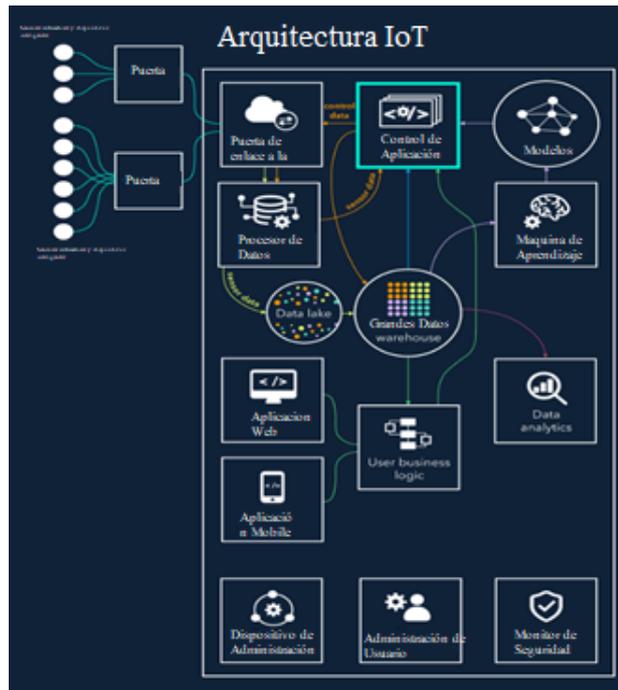


Figura 1.1 Arquitectura “IoT”. [28]

### 1.3 INDUSTRIA 4.0

La industria 4.0 esta designada como la cuarta revolución en la industria. Hoy en día en el mundo vivimos una era totalmente digitalizada, los desafíos tanto económicos, sociales pugnan a las empresas industriales a progresar su agilidad y capacidad de respuesta para ganar capacidad para gestionar toda la cadena de valor. Por lo cual las empresas necesitan tener asistencia de tecnología virtual y físicas brindando participación y adaptación acelerada para sus negocios y operaciones. [30]

Para implementar la Industria 4.0, se debe tener presente los principios de diseño. Proporcionando la adaptación integral de todo el sistema y permitiendo coordinación entre los componentes de la industria 4.0. Hay siete principios de diseño que aparecieron en la aplicación e implementación de la Industria 4.0: Agilidad, Interoperabilidad, Virtualización, Descentralización, Gestión de datos en tiempo real, Orientación al servicio y Procesos comerciales integrados.[30]

### **1.3.1 ADAPTACIÓN EXITOSA DEL SISTEMA DE LA INDUSTRIA 4.0:**

Se debe tomar en cuenta tres características fundamentales

1. Integración horizontal a través de cadenas de valor
2. Integración vertical y redes de sistemas de fabricación o servicios
3. Ingeniería de extremo a extremo de la totalidad cadena de valor

### **1.3.2 IMPLEMENTACIÓN EXITOSA DEL SISTEMA DE LA INDUSTRIA 4.0**

Se necesitan tener tres tecnologías centrales señaladas en el punto 1.3.1 y a su vez nueve fundamentales para que englobe todo el sistema:

1. Robótica adaptativa
2. Sistemas integrados
3. Fabricación aditiva
4. Tecnologías en la nube
5. Tecnologías de virtualización
6. Simulación
7. La seguridad cibernética
8. Sensores y actuadores
9. Tecnologías móviles

Estas tecnologías deben ser compatibles con tecnologías básicas como sensores, y actuadores.[29]. Este comienzo tecnológico este diseño que les permita a los profesionales prever el progreso de la adaptación de la Industria 4.0.

la figura 1.2 se visualiza el marco de la industria, las empresas deben centrarse en involucrar y redefinir el producto inteligente y los procesos inteligentes en sus funciones principales. [29]



Figura 1.2 Marco de la industria 4.0

#### **1.4 OBJETIVO Y MOTIVACIÓN DEL TRABAJO**

El objetivo de este proyecto llevar a cabo una serie de prácticas. Para ellos debemos de realizar la conexión de todos los elementos, así como la comunicación entre ellas, concluyendo con la programación que autoriza un óptimo funcionamiento de todas sus prácticas planteadas.

Hoy en día la tendencia mundial en la industria está dirigida a las aplicaciones de la Industria 4.0, es decir, apunta a evolucionar la producción y enfocada a una personalización al comprador.

Por este motivo, se necesitan tomar una serie cambios en el rendimiento que sean capaces de alcanzar a ejecutar una personalización idónea al cliente, esto aumenta la importancia del progreso y mejora de los elementos de la figura 1.3 ya que forman las columnas básicas para la creación de una producción basada en la Industria 4.0.



Figura 1.3: Esquema de la Industria 4.0.[9]

## CAPÍTULO 2: FACTORY I/O.

### 2.1 DESCRIPCIÓN FACTORY I/O

Factory I/O es un software de 3D orientado al aprendizaje de tecnologías de automatización. Su entorno es amigable y de fácil manejo, permite crear rápidamente una construcción virtual utilizando una gama de componentes industriales comunes. “Factory I/O” también está incluido aplicaciones industriales típicas, desde un nivel básico hasta un nivel avanzando. La aplicación más común consiste en usar “Factory I/O” como una plataforma de simulación para probar sistemas de automatización diseñados en un Controlador Lógico Programable.[1][10]

### 2.2 COMPONENTES DE “FACTORY I/O”

Factory I/O facilita una gran colección de componentes basados en equipos industriales. Estos componentes están organizados en ocho categorías:

- Sección.

- Carga de pieza pesada
- Carga de pieza ligera
- Detectores.
- Operadores.
- Estaciones.
- Dispositivos de alerta.
- Pasillos

En la figura 2.1 se muestra varios componentes disponibles en Factory I/O para su uso y simulación.[10]



Figura 2.1 Componentes disponibles en Factory I/O [10]

En adición a los componentes para la creación de procesos existen emisores y receptores. La función de estos elementos se detalla a continuación:

- Emisor

Permite introducir un objeto de prueba en cada escena ejecutada. Se puede configurar la base del emisor, el tiempo y la cantidad de elementos a emitir. El emisor de objetos en Factory I/O se muestra en la figura 2.2

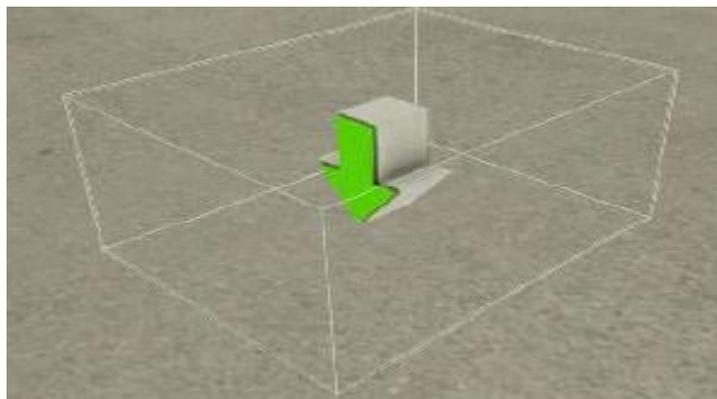


Figura 2.2 Emisor de elementos en Factory I/O

- Extractor

Permite retirar o extraer un objeto de una escena de simulación. El extractor de objetos en Factory I/O se muestra en la figura 2.3

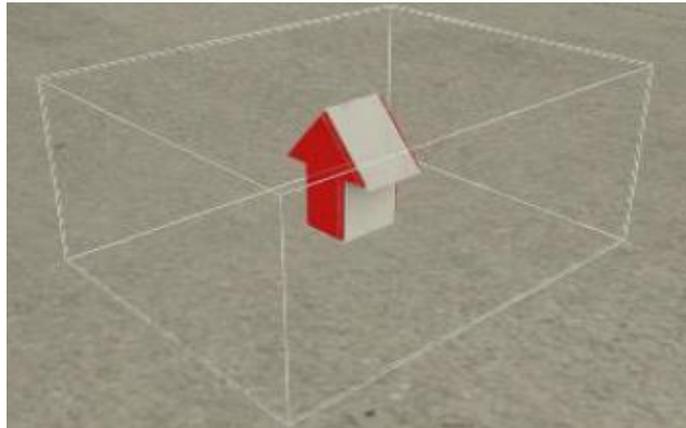


Figura 2.3 Extractor de elementos en Factory I/O

Imagen propia

## 2.2 NAVEGACIÓN

Factory I/O dispone de varias cámaras virtuales que permite navegar en el espacio tridimensional para la edición y la supervisión del proceso virtual (véase figura 2.4). A continuación, se detalla las características de estas cámaras:[11]

- Cámara Orbital:

Esta cámara facilita la edición de los entornos virtuales, por lo que resulta de gran utilidad al momento de crear escenas. Trabaja rotando alrededor de un punto de interés el cual se marca en color blanco al hacer doble clic en la pieza de la escena. Al momento de utilizar el método automático de la cámara, el punto de atención se coloca automáticamente al centro de la pieza una vez que avanza. Ahora, esto se puede mantener en el punto de atención en la postura original conservando presionado el botón de la izquierda. Ya definido el punto de monitoreo, gira la cámara a su alrededor

del mismo sosteniendo el RMB (*“Right Mouse Button”*) y corriendo el mouse. En la tabla 2.1 se indican las funciones principales de esta cámara.

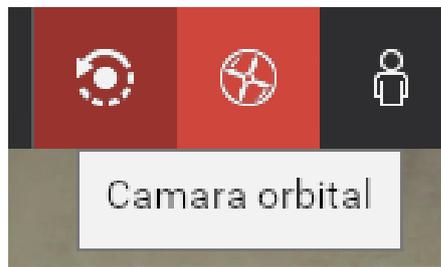


Figura 2.4 Tipos de cámara.

Imagen propia

Tabla 2.1 Detalle de comandos para el uso de la cámara de orbital [11]

<b>Control</b>	<b>Descripción</b>
Doble LMB ( <i>“Left Mouse Button”</i> )	Coloca el punto de beneficio de la cámara. La cámara rota alrededor de dicho punto y las piezas se ubicarán a esta altura.
RMB + arrastre ( <i>“Right Mouse Button”</i> )	Rotación de la cámara en la periferia del punto de interés.
MMB + arrastre ( <i>“Middle Mouse Button”</i> )	Traslada horizontalmente la cámara.
Rueda del mouse	Aproxima la cámara hacia afuera y hacia adentro.
Arriba “W”	La cámara se traslada hacia arriba.
Abajo “S”	La cámara se traslada hacia abajo.
Derecha “D”	La cámara se traslada a la derecha.

Izquierda "A"	Se traslada a la izquierda.
---------------	-----------------------------

- Cámara tipo Fly:

Se utiliza para trasladarse dentro del espacio tridimensional. Choca con cada parte de la escena, este no es detectada por los actuadores. En la tabla 2.2 se especifica el manejo de la cámara de sobrevolado.[11]

Tabla 2.2 Navegación con la cámara Fly.[11]

Control	Descripción
Doble LMB (" <i>Left Mouse Button</i> ")	visualiza la cámara hacia la dirección del cursor del mouse.
MMB + arrastre (" <i>Middle Mouse Button</i> ")	Rotación de la cámara.
Rueda del mouse	Mueve la cámara verticalmente.
LMB + MMB	Desliza la cámara hacia adelante.
Arriba "W"	Se desliza hacia adelante.
Atrás "S"	Se desliza hacia atrás.
Derecha "D"	Se desliza a la derecha.
Izquierda "A"	Se desliza a la izquierda.

- Cámara tipo Primera persona:

La cámara en primera persona se emplea para aparentar a una persona en una fábrica. Cabe recalcar que se cruza con partes de la escena y estos no son detectados por los sensores.[11]

Tabla 2.3 Navegación con la cámara en primera persona.[11].

Control	Descripción
Doble LMB (“ <i>Left Mouse Button</i> ”)	Ajusta la cámara a la dirección donde se encuentra el mouse.
RMB + arrastre (“ <i>Right Mouse Button</i> ”)	Rotación de la cámara.
LMB + RMB	Desliza la cámara hacia adelante.
“W”	Se desliza hacia adelante.
“S”	Se desliza hacia atrás.
“A”	Se desliza a la derecha.
“D”	desliza a la izquierda.
Tecla Espacio	Salta

### 2.3 REQUISITOS DEL SISTEMA Y VERSIONES

Las características principales que debe tener un ordenador para poder instalar Factory I/O y que el funcionamiento sea óptimo varían según su versión, su última versión requiere mayores características en video estos son detalladas en la tabla 2.4.[12]

Tabla 2.4 Requerimientos del sistema para Instalación. [11]

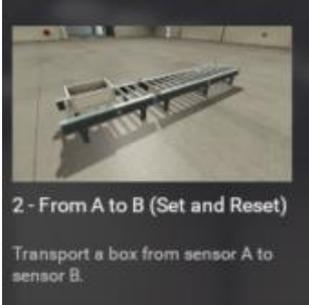
Sistema operativo	Windows Vista o superior.
Procesador	Intel Core 2 Duo a 2Ghz, o AMD Athlon 64 x2 2Ghz o superior.
Memoria	1 giga bite.
Memoria en disco duro	600 megabyte.
Video	Tarjeta gráfica NVIDIA desde 2007 (serie GeForce 8), tarjetas AMD desde

	2007 (serie Radeon 2xxx), tarjetas Intel desde 2008 (GMA 4500); soporte para shader modelo 2.0 o modelos superior.
Sonido	TS compatible con DirectX.
Otros	NET Framework 4.5.

## 2.4 ESCENAS

Factory I/O brinda al usuario una amplia gama de escenas las cuales están inspiradas en aplicaciones industriales reales y típicas. Dependiendo de la escena seleccionada el usuario se enfrentará a un proceso con menor o mayor complejidad. Las escenas son únicamente de lectura por lo que no se pueden sobre escribir, no obstante, estas pueden ser personalizadas luego de guardarlas y abrirlas dentro de la sección “**Mis escenas**”.[12]

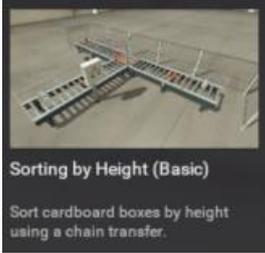
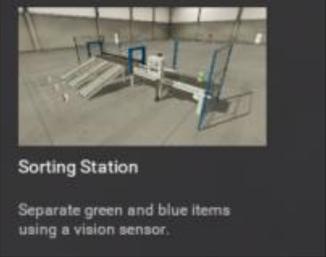
Tabla 2.5 Descripción de Escenas Factory IO.

<i>Escena</i>	<i>Descripción.</i>
	Transporte desde un punto A hacia un punto B.
	Transporte de una caja desde el sensor A hasta el sensor B.

 <p><b>3 - Filling Tank (Timers)</b> Fill and empty a tank using timers.</p>	<p>Llenado de un tanque usando Timers.</p>
 <p><b>4 - Queue of Items (Counters)</b> Load and unload three boxes onto a conveyor.</p>	<p>Carga y descarga de cajas desde una banda.</p>
 <p><b>Assembler</b> Assemble parts made of lids and bases using a two-axis pick and place.</p>	<p>Ensamblador de partes usando pick and place de dos ejes.</p>
 <p><b>Assembler (Analog)</b> Assemble parts made of lids and bases using a two-axis pick and place with analog values.</p>	<p>Ensamblador de partes usando pick and place de dos ejes y colocación con valores análogos.</p>
 <p><b>Automated Warehouse</b> Transport, store and retrieve boxes from a rack.</p>	<p>Transporte, almacenamiento de cajas desde un estante</p>
 <p><b>Buffer Station</b> Buffer and separate up to five boxes.</p>	<p>Clasificar hasta 5 cajas.</p>

 <p><b>Converge Station</b> Converge two paths into a single one.</p>	<p>Converge dos estaciones en una sola.</p>
 <p><b>Elevator (Advanced)</b> Collect up to five items into a stackable box using an elevator. Each item arrives randomly at a different floor.</p>	<p>Recolecta hasta 5 ítems y los almacena usando un elevador, cada ítem arriba de forma aleatoria en cada piso.</p>
 <p><b>Elevator (Basic)</b> Transport items through three different levels using two elevators.</p>	<p>Transporta ítems entre tres diferentes niveles usando dos elevadores.</p>
 <p><b>Level Control</b> Control the liquid level (or flow) of a tank using a Proportional-Integral-Derivative (PID) controller.</p>	<p>Control de nivel líquido de un tanque usando PID.</p>
 <p><b>Palletizer</b> Palletize cases in several layers.</p>	<p>Paletización de cajas</p>

 <p><b>Pick &amp; Place (Basic)</b> Pick and place items from one conveyor into another.</p>	<p>Usar pick and place para ítems de una banda a otra.</p>
 <p><b>Pick &amp; Place XYZ</b> Place boxes onto pallets using a three axes manipulator.</p>	<p>Colocar cajas en modo paletización usando tres ejes de manipulación.</p>
 <p><b>Production Line</b> Produce the same number of lids and bases by controlling two machining centers.</p>	<p>Producir la misma cantidad de tapas y bases controlando dos máquinas de aprendizaje</p>
 <p><b>Separating Station</b> Separate blue and green parts into two conveyors.</p>	<p>Estación de Separación material azul y verde en dos bandas.</p>
 <p><b>Sorting by Height (Advanced)</b> Sort cardboard boxes by height using a turntable. This scene is compatible with ITS PLC sorting system.</p>	<p>Sistema de clasificación por peso avanzada.</p>

 <p><b>Sorting by Height (Basic)</b> Sort cardboard boxes by height using a chain transfer.</p>	<p>Sistema de clasificación por peso básica.</p>
 <p><b>Sorting by Weight</b> Sort boxes by weight using a conveyor scale.</p>	<p>Clasificación de cajas usando una banda de peso.</p>
 <p><b>Sorting Station</b> Separate green and blue items using a vision sensor.</p>	<p>Separación de material verde y azul usando un sensor de visión.</p>

## 2.5 CONTROLADORES e/s

Un controlador de entrada/salida permite que Factory I/O interactúe con un controlador o programa informático externo. Dentro de las librerías de Factory I/O se pueden encontrar varios controladores, cada uno anexo a ser usado con una tecnología específica. La Tabla 2.6 resume las características de algunos de los controladores existentes en Factory I/O. [11]

Tabla 2.6 Controladores E/S [11]

<b>Control</b>	<b>Descripción</b>
Control I/O	Permite el desarrollo de programas utilizando diagramas de bloques. Utiliza las funciones más comunes de un PLC.

Servidor Modbus TCP / IP	Implementan un servidor Modbus TCP/IP (esclavo)
Ciente Modbus TCP /IP	Implementan un cliente Modbus TCP/IP (maestro)
Siemens S7-200 / 300/400	Conexión Ethernet a PLCs de Siemens: S7200, S7-200 SMART, S7-300 y S7-400.
Autómata Siemens S7-1200 /1500	Conexión Ethernet a los PLCs Siemens S7-1200 y S7-1500.
Siemens S7-PLCSIM	Interfaz de Siemens para S7-PLCSIM.
Advantech USB 4750	Esta interfaz es para tarjetas Advantech USB 4750 y 4704.
Allen-Bradley Logix5000	Conexión Ethernet a Allen-Bradley ControlLogix, CompactLogix.
Allen-Bradley Micr0800	Conexión Ethernet al PLC Allen-Bradley Micr0800.
Allen-Bradley Micro Logix	Conexión Ethernet al PLC MicroLogix de Allen-Bradley.
Servidor Automgen	Es una interfaz para Automgen por medio de un servidor TCP/IP.
MHJ	Interfaz para el software WinPLC-Engine y WinSPS-S7.
Allen-Bradley MicroLogix	Conexión Ethernet al PLC MicroLogix de Allen-Bradley.
Allen-Bradley SLC 5/05	Conexión Ethernet al PLC SLC-5/05 de Allen-Bradley.

## 2.6 ETIQUETAS

Antes de usar una escena con la ayuda de un controlador externo, como por ejemplo un PLC, se puede realizar una prueba manual del funcionamiento del proceso virtual. De esta manera es posible verificar que el diseño esté listo para ser automatizado [12]. Cada sensor o actuador por defecto tiene etiquetas:

- Etiquetas de actuadores.
- Etiquetas de sensores.

Estas etiquetas sirven para enlazar a los sensores y actuadores al controlador. Las etiquetas pueden ser de tres tipos diferentes dependiendo del tipo de sensor y actuador:

- Boolean: Para valores de tipo On/Off.
- Float: Para manejar variables de tipo analógico (números reales).
- Integer: Para manejar variables de tipo entero.

Para forzar las etiquetas, es necesario abrir el menú de tags (véase la figura 2.6); basta con dar clic encima del nombre de la etiqueta para poder modificar los valores. Si son variables de tipo Digital se podrá forzar su valor entre cero y uno. Si las variables son de tipo analógico, podrán ser modificadas desde 0 a 10. Finalmente, si son variables enteras se podrá modificar con el valor deseado. [2]

Type	Value ON	Value OFF	To force the value
Digital (Bool)			<b>Left-click</b> to switch on/off.
Analog (Float)			<b>Left-click</b> and <b>Drag</b> to set the desired value.
Numerical (Integer)	4	0	<b>Left-click</b> and <b>Insert</b> an integer value.

Figura 2.6 Control manual de las variables analógicas en Factory I/O. [2]

Imagen propia

## 2.7 CONTROL AUTOMÁTICO

El control automático permite conectar con un autómatas externo. Este modo debe ser configurado mediante la opción “drivers” dentro de la opción “archivo” en Factory IO.

Las etiquetas se arrastran hacia las entradas, salidas de los sensores y actuadores a usar en el autómatas como se muestran en la figura 2.7.

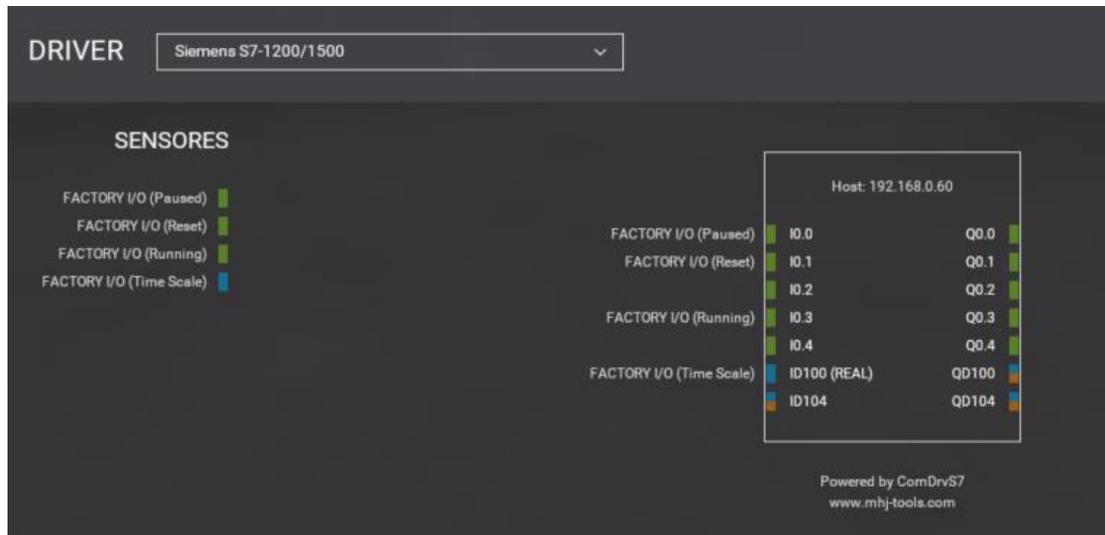


Figura 2.7 Asignación de etiquetas en Factory I/O. [11]

## CAPÍTULO 3: UBIDOTS.

Ubidots empodera a entusiastas de la tecnología y emprendedores a construir nuevos productos y aplicaciones a partir de objetos conectados a Internet (véase la Figura 3.1), donde se podrá realizar desde una aplicación que registre la cafeína que se ingiere durante el día y alerte con un mensaje de texto si se excede cierto nivel, hasta una nevera industrial que registre su temperatura en tiempo real desde la Nube. Ubidots permite crear aplicaciones para el “IoT” en poco tiempo y su interfaz es amigable con el usuario [12].

Es un servicio gratuito que permite a los desarrolladores crear un entorno de computación en la nube asequible, fiable y utilizable en un ecosistema de plataformas “IoT”. Ubidots está especializado en soluciones de hardware y software conectadas para monitorizar, controlar y automatizar procesos remotamente en el ámbito de la

salud, energía, industria, fabricación, servicios públicos y transporte, Con el fin de crear aplicaciones desde básicas hasta avanzadas. Los principales beneficios de la plataforma Ubidots son: [12]

- La conexión del hardware con la nube y diversas bibliotecas.
- Configuración de variables de forma automática, configurable sus propiedades y su apariencia en los dispositivos para duplicar el proceso en nuevos dispositivos.
- Modificación de la API (Application Programming Interface).
- Supervisión para el análisis de datos de las aplicaciones con integraciones API.
- Conversión de datos de origen nativos en información con variables sintéticas
- Diseño para cuadros de mando analizados en tiempo real para el análisis de datos y control en los dispositivos
- Mejora la función compartir datos con enlaces públicos o integrando cuadros de mando o widgets en aplicaciones web privadas y móviles.
- Los siguientes comandos "kill switch" o "restart" se activan cuando el hardware ha estado inactivo durante mucho tiempo.
- Asignación de permisos y restricciones para cualquier usuario que busque interactúe con los mandos, dispositivos y/o eventos.

Ubidots maneja 4 términos necesarios para el manejo de datos en la nube: Data Source, Variable, Value y Event. A continuación, se detalla cada uno de estos: [31]

- Data Source: maneja un conjunto de información haciendo referencia a un dispositivo, en donde cada data source utiliza una o más variables.
- Variable: maneja un conjunto de datos que varía con el tiempo.
- Value: es un valor actual de una variable en un momento determinado.
- Event: permite tomar una acción en un instante determinado. Permite la interacción con cada elemento de ubidots a través de su API, por lo tanto, cada elemento que presenta puede ser creado, modificado o eliminado.



Figura 3.1: Logo Ubidots. [13]

### 3.1 CREACIÓN DE UNA CUENTA.

Para ser un usuario en la nube de Ubidots se debe iniciar con un usuario en su servidor, con un email y una contraseña (véase la Figura 3.2).[13] Esto se realiza desde su página principal: “<https://ubidots.com>”.

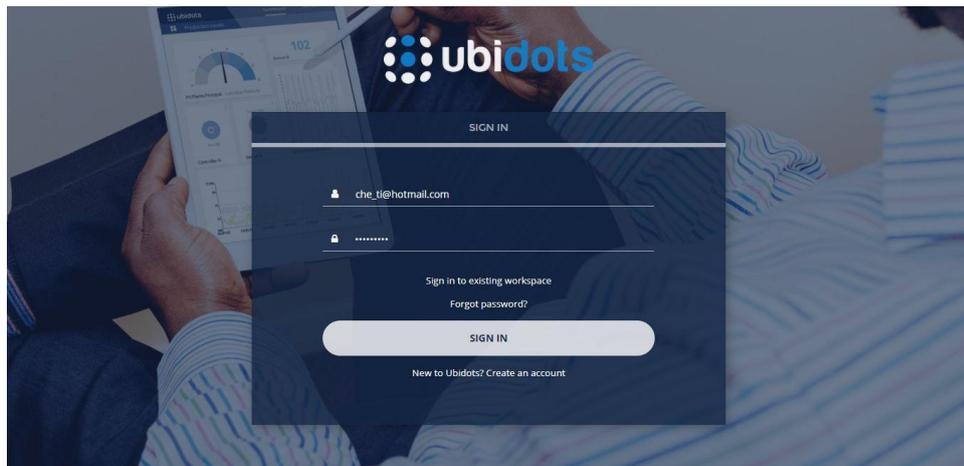


Figura 3.2: Interfaz de inscripción en Ubidots.

Una vez creado el usuario dentro de la página web Ubidots se procede a crear dispositivos y variables.

### 3.2 DISPOSITIVOS.

Los dispositivos contienen datos almacenados en variables, para acceder a un dispositivo es necesario contar con el token o key API; lo que permite la lectura y modificación de las variables almacenadas.[13]

### **3.3 VARIABLES.**

Ubidots almacena los datos en variables, estas variables pueden ser: sintéticas o cruda; [13]

Las variables crudas son variables extraídas desde un ambiente externo como Node red donde los datos llegan sin ser procesados, mientras que las variables sintéticas, son variables con las cuales se ha realizado cálculos dentro de Ubidots por lo tanto son variables que modifican su valor respecto a un cálculo.

### **3.4 DASHBOARD (TABLERO).**

De ahora en adelante nos referiremos a los Dashboard por su nombre en español tableros. Los tableros permiten visualizar y ordenar los datos recibidos mostrándolos en widgets. [13]

### **3.5 WIDGETS.**

Son herramientas de visualización que permiten obtener el dato de una variable almacenada en un dispositivo y mostrarlo. Existen varios widgets lo que permite crear un ambiente llamativo para el usuario al momento de operar. [13]

### **3.6 EVENTOS.**

Ubidots permite enviar notificaciones de mensaje a quien lo necesite y cuando lo necesite, las alertas que activan los eventos se disparan cuando los valores de las variables a observar cruzan el valor antes pre establecido.[13]

### **3.7 CREACIÓN DE DISPOSITIVOS.**

Los dispositivos son bloques para envío o recepción de datos a través de un protocolo de conexión a la nube de Ubidots. Existen librerías que permiten la comunicación con diferentes plataformas como Arduino, Siemens, Microchip, Raspberry pi, etc., (véase la Figura 3.3). En otras palabras, un dispositivo es una representación virtual de una fuente de datos.[31][32]

La creación de un dispositivo dentro de Ubidots es automática. Ubidots se maneja por datos y la cuenta gratuita tiene un máximo de 4000 datos de envío y recepción. Luego de esa cantidad se debe esperar 24 horas para continuar con un intercambio de

datos.[31][32]

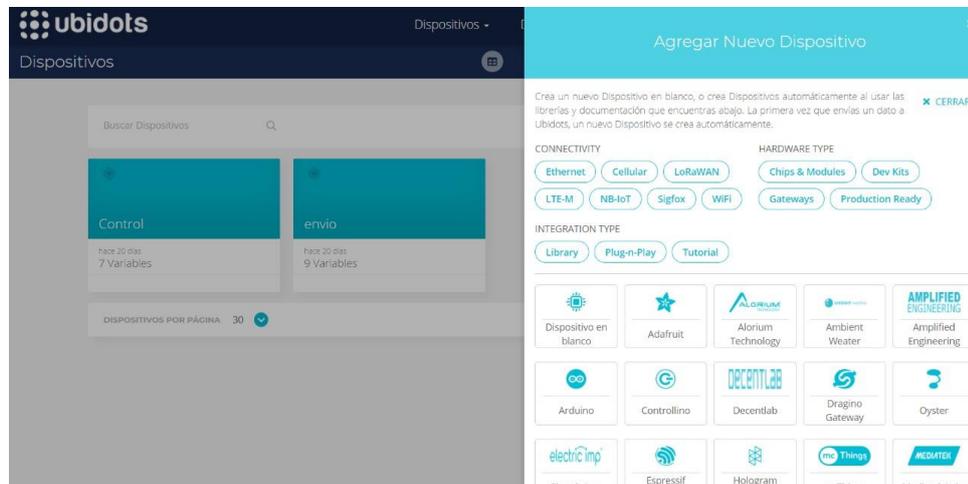


Figura 3.3: Interfaz creación de dispositivos en Ubidots.

### 3.8 CREACIÓN DE WIDGETS.

Después de haber creado un dispositivo, es necesario agregar widgets para interactuar con las variables del sistema (véase la figura 3.4). Al crear un widget se tendrá que dar un nombre y elegir el tipo. Después de que se ha creado se pueden colocar otros datos opcionales como la unidad de la variable y una descripción. Como se observa en la figura 3.5 las propiedades del widget Battery.[32]

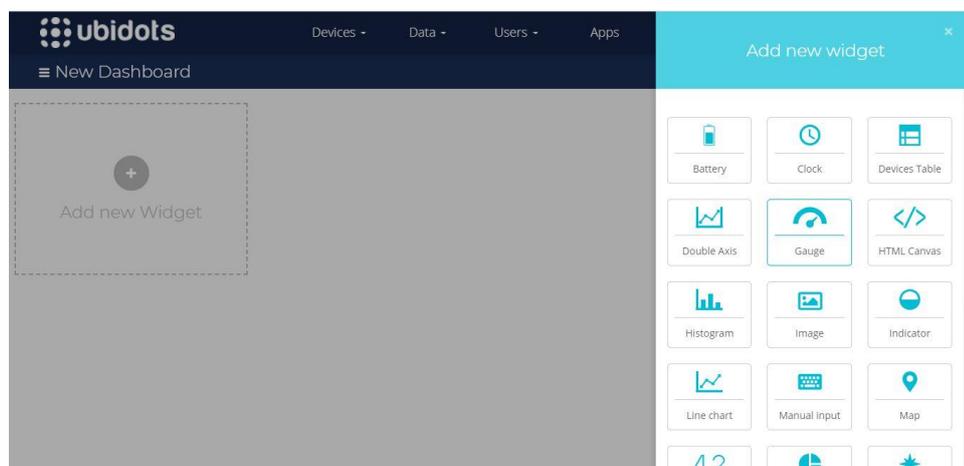


Figura 3.4: interfaz de Ubidots

Imagen propia

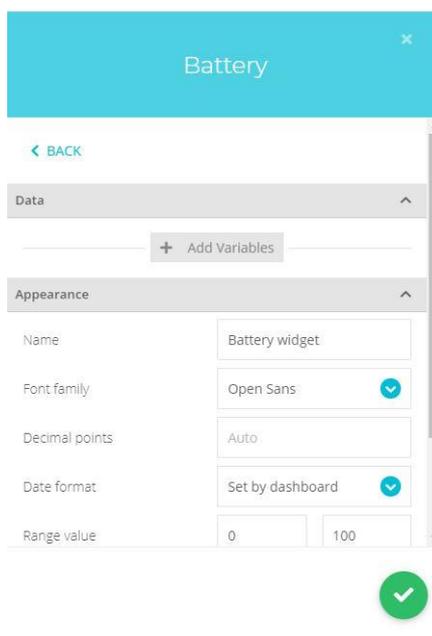


Figura 3.5: Creación de un variable dentro de Ubidots.

Imagen propia

Ubidots permite agregar a cada tablero un máximo de widgets. Dentro del tablero se modifica su tamaño, así como permite ubicar la posición de los widgets (véase la figura 3.6). [32]



Figura 3.6: Disposición variables tablero Ubidots.

### 3.9 CREACIÓN DE VARIABLES.

Las variables permiten administrar los datos dentro de Ubidots. Dentro de esta plataforma se pueden crear dos tipos de variables como lo habíamos mencionado anteriormente: “Cruda”, “Sintética” (Véase la figura 3.7). [32]

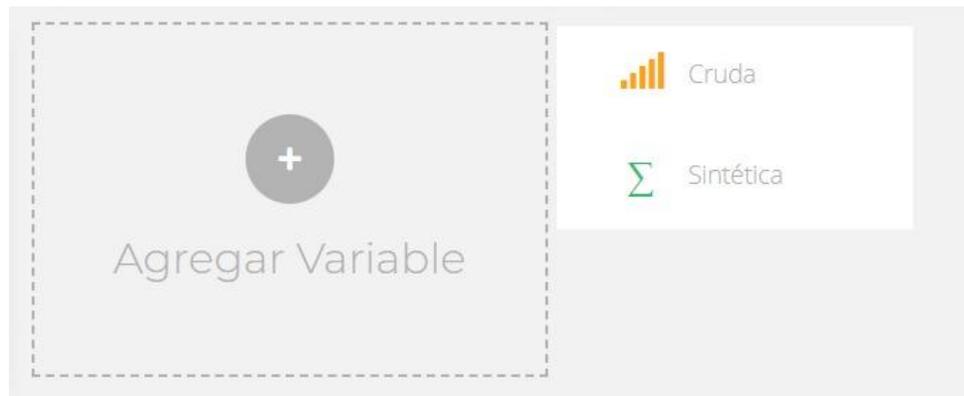


Figura 3.7: Cuadro de diálogo para selección de tipos de variables dentro de Ubidots

Para acceder a una variable dentro de la nube, el nombre de la variable debe coincidir con “API Label” como se muestra en la figura 3.9, esta variable guardara todos los datos que se envíe o reciba durante su uso.



Figura 3.8: Envío y recepción de datos mediante una variable.

Imagen propia

### 3.10 COMPARTIR EN UBIDOTS.

Para compartir los widgets en forma de iframe es necesario hacer clic en el botón compartir este está ubicado en cada widget. Además, puedes compartir esta venta con más usuarios a través de un enlace. Ver figura 3.9. [31]



envía hacia el servidor de Ubidots es decir solicita el valor de la última variable. Una vez que el servidor envía una respuesta con el estado de la solicitud. [31]

Una solicitud *HTTP* necesita tener los siguientes parámetros:

- **Host:** Prepara al servidor para ejecutar solicitudes de HTTP
- **Ruta:** Es denominada como la parte restante de la URL que formula el recurso que desea acabar, ya sea un dispositivo o una variable.
- **Encabezados:** se explica todos los factores operativos mediante la solicitud HTTP, definiendo a su vez los parámetros operativos de la solicitud HTTP, como autenticación, longitud de contenido y tipo de contenido.
- **Cuerpo:** en el caso de solicitudes POST y PATCH, estos son los datos enviados por su dispositivo al servidor. Las solicitudes GET generalmente no tienen cuerpo porque están destinadas a solicitar datos, no a enviar datos.

Ubidots acepta datos como notación de objetos “JavaScript o JSON” como se visualiza en la figura 3.10 mostrando un tipo de datos de HTTP típico, es una colección de pares de nombres o de valor. En diferentes lenguajes de programación se lo denomina como un objeto, registro, estructura, tabla hash, diccionario, lista con clave. También es legible por humanos e independiente del lenguaje [31]

```
JSON
-----
{
  "value": 1000, "context": { "estado":
    "encendido" }
}
```

Figura 3.10: Formato JSON. [14]

Imagen propia

Una solicitud generada de HTTP típica hacia ubidots se debe configurar de la siguiente manera. [31]

- POST {PATH} HTTP/1.1<CR><LN>Host:
- {HOST}<CR><LN>User-Agent:
- {USER\_AGENT}<CR><LN>X-Auth-Token:
- {TOKEN}<CR><LN>Content-Type:
- application/json<CR><LN>Content-Length:
- {PAYLOAD\_LENGTH}<CR><LN><CR><LN>{PAYLOAD}<CR><LN>

Donde:

- {RUTA}: Ruta al recurso a consumir
- {HOST}: URL de host.
- {USER\_AGENT}: una cadena opcional que se utiliza para identificar el tipo de cliente.
- {TOKEN}: clave única que autoriza a su dispositivo a ingerir datos dentro de su cuenta de Ubidots.
- {PAYLOAD\_LENGTH}: la cantidad de caracteres de su carga útil.  
 {PAYLOAD}: datos para enviar.

### 3.12 TOKEN

Para poder realizar el envío de datos a una variable es necesario el acceso a la nube, Ubidots proporciona un token por cada cuenta como se muestra en la figura 3.11. Este token caracteriza a la cuenta como única forma de encontrarla en la nube en otras palabras el token permite el acceso directo hacia la nube, el token se lo puede encontrar en la sección de credenciales y tiene la opción de crear más de un token si es necesario.[31][32]

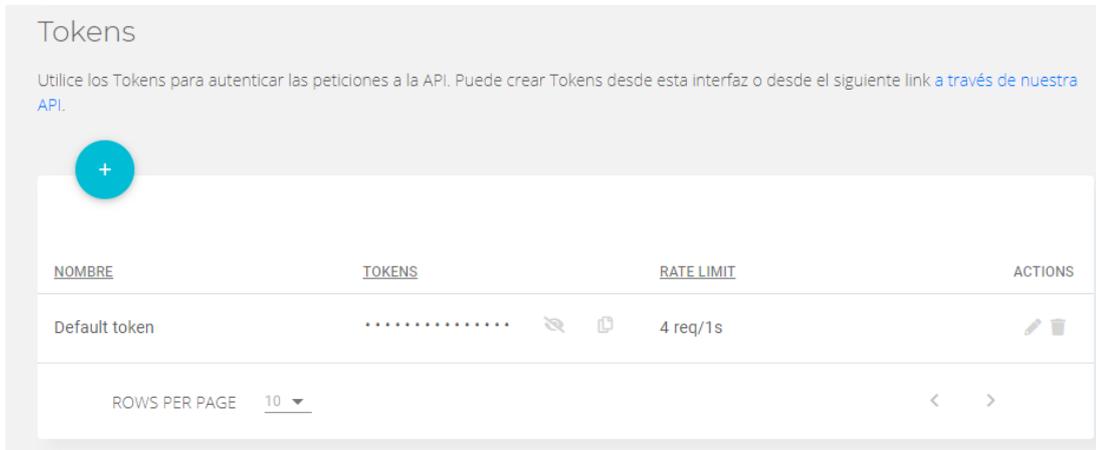


Figura 3.11 Obtención del Token.

El token esta conformado por una serie de 36 números, letras y símbolos especiales este es un ejemplo de un Token.

BBFF-Pi4613uPOxxxxx3BvPjgyUqSOiT6HD

## CAPÍTULO 4: NODE RED

Node-Red es un instrumento de programación visual basada en código abierto que permite integrar dispositivos de hardware, APIs y servicios online [16][17]. Su programación se basa en nodos y líneas de flujo que interconectan los nodos.

La figura 4.1 muestra una vista simplificada de los componentes de Node red. Las aplicaciones realizadas son equivalentes a flujos, que se encuentran compuestos por nodos; por lo tanto, un servidor puede ejecutar múltiples flujos estos son compuestos por un gráfico de nodos; además de los mensajes explícitos enviados, la información también se la puede enviar a través del flujo global y textos compartidos.

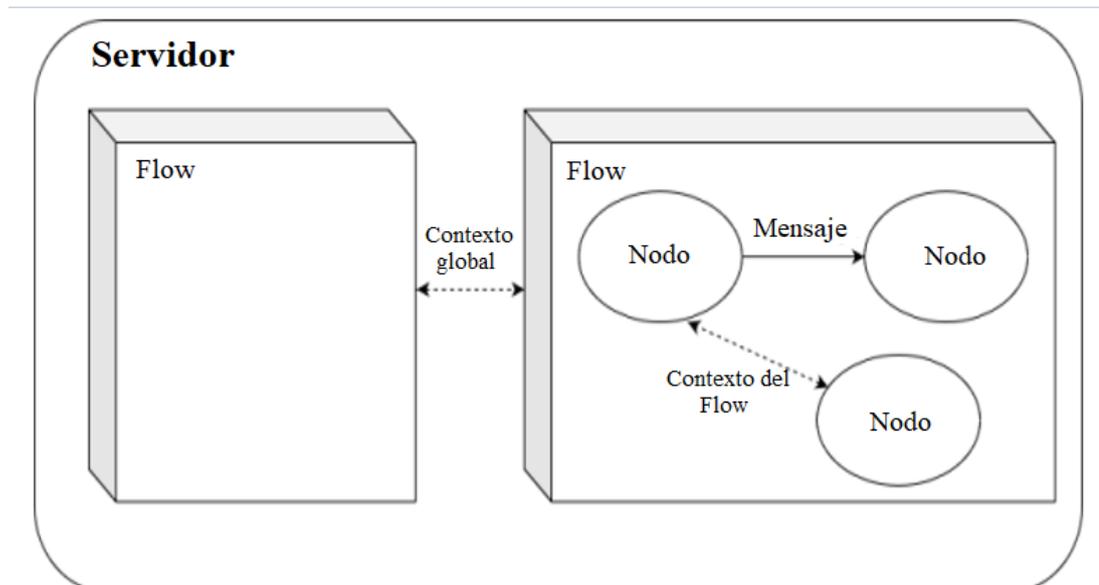


Figura 4.1: Arquitectura Node-Red. [17]

Node-Red es una plataforma de código abierto por lo que es gratuita. Las herramientas están basadas en JavaScript y diseñadas sobre la plataforma Node.js. Node-Red maneja un sistema que facilita a los nodos ser representados por iconos apropiados. Se opera de dos maneras diferentes: arrastrar, soltar y conectar nodos, o importar código JavaScript. Los desarrolladores pueden conectar entradas y salidas a sus respectivos nodos de procesamiento y facilitar el flujo para el procesamiento de datos y al mismo tiempo controlar letras, mensajes etc. [18][19].

Node-Red es una herramienta adaptable y al mismo tiempo poderosa, utilizada para crear prototipos. Este sistema permite la creación rápida de aplicaciones, especialmente aplicaciones que se activan en un evento como las aplicaciones de IoT. La esencia de esta herramienta es permitir a ingenieros y técnicos crear y configurar fácilmente en tiempo real aplicaciones en dispositivos finales [18][19]. En la Figura 4.2 se ilustra un flujo de programación dentro de Node-Red.

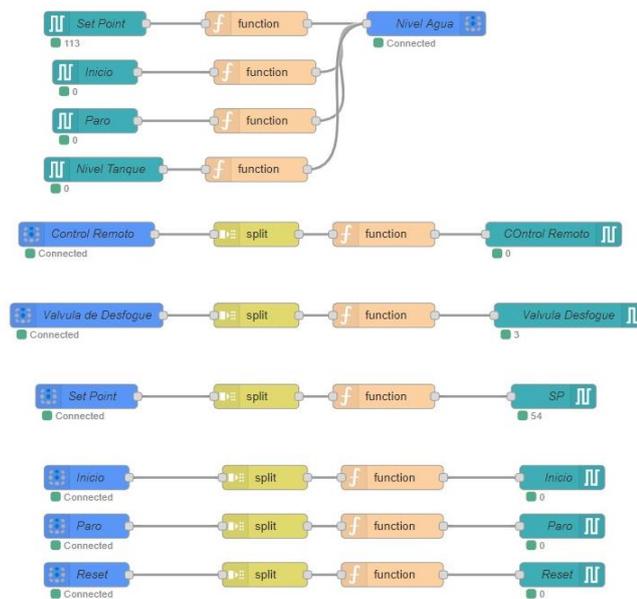


Figura 4.2: Flujo de programación en Node-Red

#### 4.1 NODOS.

Los nodos pertenecen a un segmento de código Node.js que envían y reciben eventos; cada evento se configura mediante el número de entradas y cualquier número de puertos de salida. Mediante cables se logra interconectar los nodos con los puertos seleccionados, cada evento contiene mensajes y al mismo tiempo toman forma de un

objeto .JS. Cada evento puede transmitir varios mensajes enviados por cada nodo y cada mensaje recibido en cada ejecución. Los nodos se clasifican en cuatro clases y dependen del número de puertos de entradas y salidas "E/S"

- **Nodos de Entradas:** Solo manejan puertos de salida, se activan mediante un disparado externo al momento que se inicialice la ejecución y transfiera un mensaje a sus respectivos puertos de salida.
- **Nodos de Salida:** Solo manejan puertos de entrada, Cada mensaje se realiza alguna acción externa con él, concluyendo en la rama de flujo.
- **Nodos Intermedios:** Son definidos ya que maneja un solo puerto de entrada y al menos en un puerto de salida. Abarcan la mayor parte del grafico de un flujo, todos los nodos se conectan entre los mismos nodos de entrada y salida "E/S" Ya que transforman el mensaje.
- **Nodos de configuración:** Normalmente carecen de puertos. la función es compartir la mayor cantidad de datos de configuración, como las credenciales de inicio de sesión.

En la figura 4.3 se muestra que cada nodo está bien definido, ejecutando la configuración necesaria para estos nodos que se interconecta entre sí y siendo responsables del mantenimiento del flujo de datos entre los nodos.

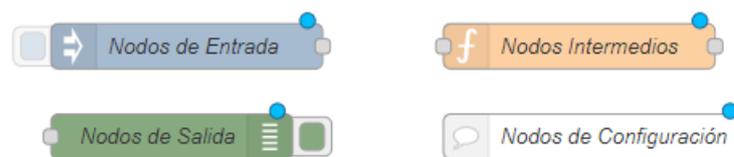


Figura 4.3: Clases de Nodos.

## 4.2 FLUJOS (FLOWS)

El flujo está representado como una pestaña que se encuentra dentro del espacio de trabajo y esta es la forma principal de organizar los nodos. Cada flujo puede tener una descripción y un nombre que se visualiza en la barra superior como una pestaña; además todos los nodos de un flujo pueden acceder al mismo contexto de ámbito del flujo.

Los flujos pueden ordenarse en el lugar de trabajo arrastrando su pestaña en la barra superior, para editar las propiedades de un flujo dar doble clic en la pestaña, lo que abrirá un cuadro de dialogo con las propiedades del flujo como se visualiza en la parte izquierda de la figura 4.4 [17]

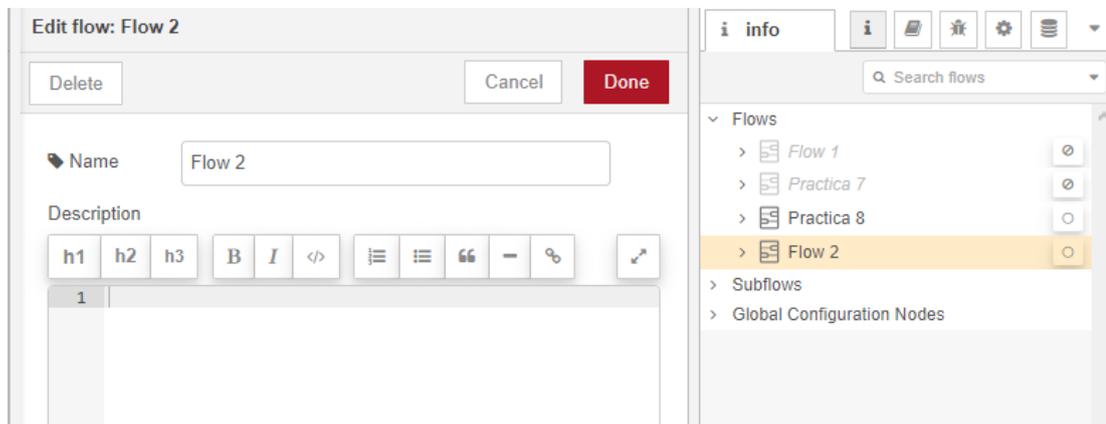


Figura 4.4: Propiedades y activación de un flujo.

Imagen propia

Un flujo se puede habilitar o deshabilitar al usar el botón  Enabled en la parte inferior del cuadro de dialogo, si un flujo esta deshabilitado, ningún nodo dentro de este se creara cuando se implemente el nodo, también se puede activar o desactivar el flujo desde el botón  en la barra lateral como se visualiza en la parte derecha de la figura 4.4 [17]

La implementación de cada flujo maneja un tiempo de ejecución en Node-Red, el mismo que se utilizará en cada evento para comprobar continuamente las entradas generadas y las condiciones de activación de cada nodo. Los nodos de ingreso se reflejan a través de un mensaje de Java Script de sus puertos de salida. Estos mensajes son recibidos por nodos subsiguientes en el flujo, lo que lleva a la ejecución de una parte del gráfico. Después de pasar por algunos nodos intermedios, la ejecución terminará en varios nodos de salida como se visualiza la figura 4.5. [20][22]

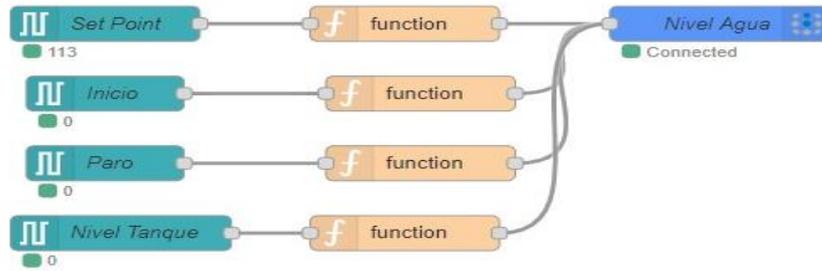


Figura 4.5: Función de flujos entradas, intermedios y salidas.

Imagen propia

### 4.3 ALAMBRES (WIRES)

Los nodos se conectan entre si mediante alambres presionando el botón izquierdo luego arrastrándolo al nodo de destino y soltando el botón, como se puede visualiza en la figura 4.6 [17]

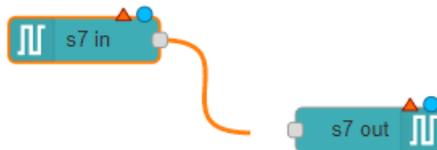


Figura 4.6 Conexión de Nodos mediante alambres (Wires).

### 4.4 PUBLICACIÓN

La publicación de los flujos se realiza mediante un botón en la parte superior derecha llamada “Deploy” (véase la figura 4.7). Este botón publica todo el flujo en la red para atrapar los datos o enviar los datos hacia una herramienta externa como Ubidots. Este botón se activa cada vez que exista cualquier cambio en el flujo [22].



Figura 4.7 Publicación de flujo.

## 4.5 ADMINISTRACIÓN DE PALETA

El administrador de paleta se puede utilizar para instalar nuevos nodos en la paleta, se accede a él desde la pestaña  en la parte superior derecha, luego en Manage palette (Administrador de paletas).

El administrador contiene dos pestañas, la primera enumera los nodos instalados mientras que la segunda pestaña enumera los nodos disponibles para instalación. Esto se ilustra de mejor manera en la figura 4.8.

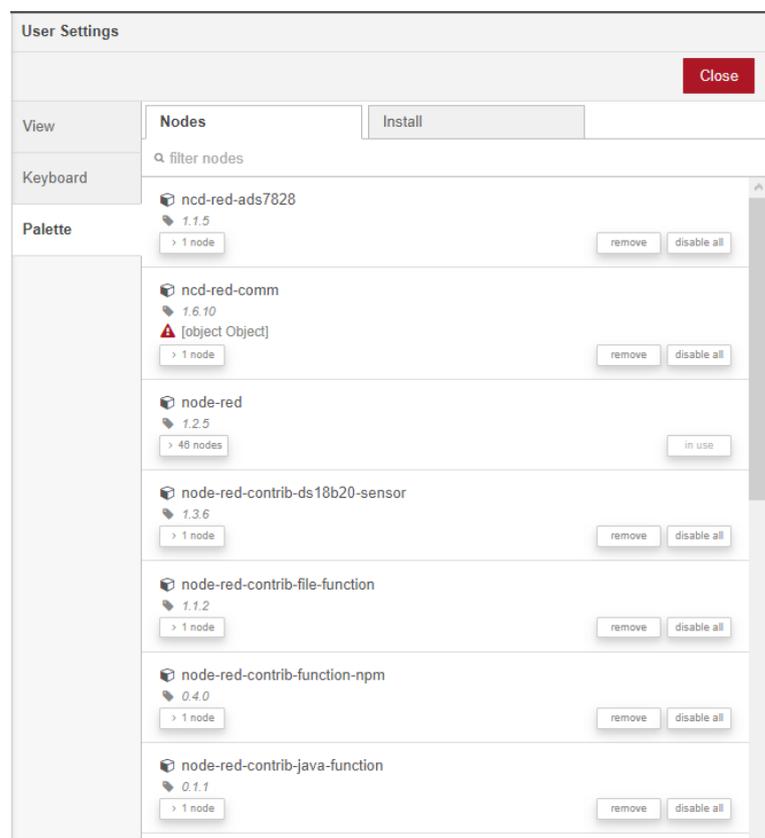


Figura 4.8 Ventana de comunicación que visualiza los nodos existentes.

## 4.6 INSTALACIÓN DE NODOS

La pestaña instalar se utiliza para buscar módulos disponibles para posteriormente instalarlos. Para buscar un nodo es necesario ingresar su nombre en la barra de búsqueda, los resultados de la búsqueda muestran los detalles de los nodos incluida la última vez que se actualizó (véase la figura 4.9) [17].

La comunicación de Node-Red se realiza mediante programación de nodos, los cuales se encuentran desarrollados por una comunidad y varios usuarios. Actualmente existen alrededor de 3063 módulos disponibles para instalarlos. El cuadro de diálogo para su instalación se muestra en la figura 4.9.[18]

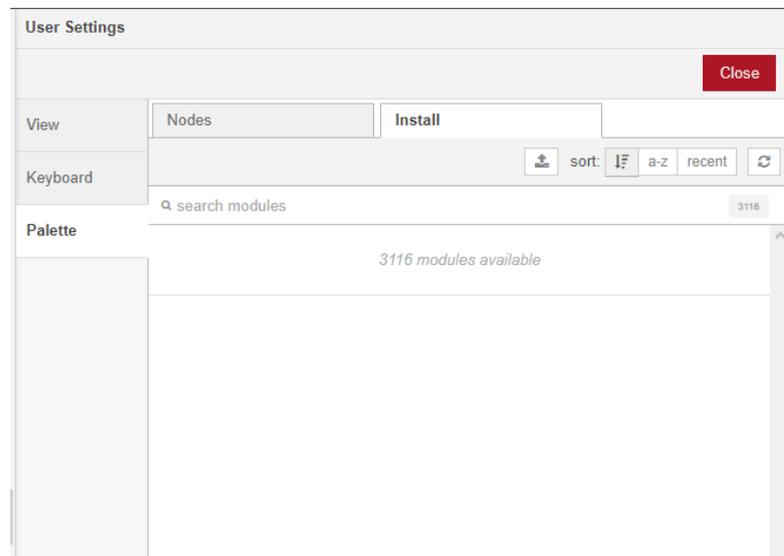


Figura 4.9 Ventana de comunicación para la instalación de nuevos nodos.

Node-Red cuenta con varios módulos desarrollados para la comunicación con varias plataformas como Siemens, Raspberry, etc. Estos módulos permiten acceder a las variables dentro de las plataformas.

A continuación, se observa el cuadro de diálogo del bloque para la comunicación con Siemens. Para ingresar a este cuadro de diálogo es necesario dar doble clic sobre el nodo, en el cual se debe configurar la IP con la cual se va a comunicar con el autómeta [17].

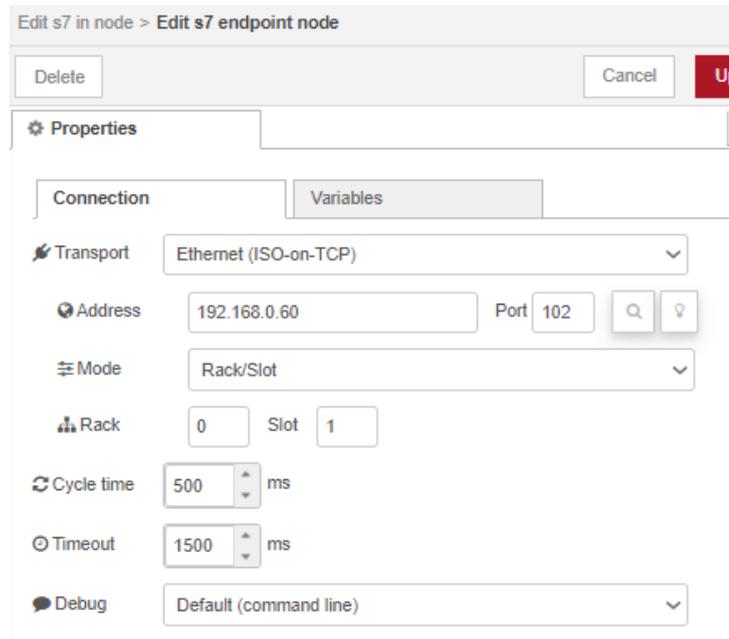


Figura 4.10: Ventana de comunicación para configuración de la comunicación con un PLC Siemens.

El módulo de Siemens cuenta con comunicación hacia Ubidots, que permite escribir variables dentro de Dispositivos. Este bloque se llama “Ubidots\_out”, el único requisito es contar con el Token brindado dentro de la página web Ubidots (véase las figuras 4.11 y 4.12). Para ingresar a los cuadros de dialogo basta con dar doble clic sobre los nodos.

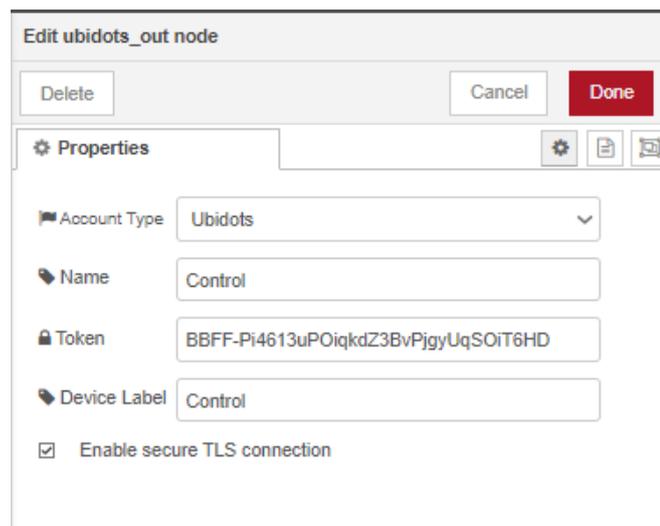


Figura 4.11: Ventana de comunicación para configuración del bloque de comunicación de salida.

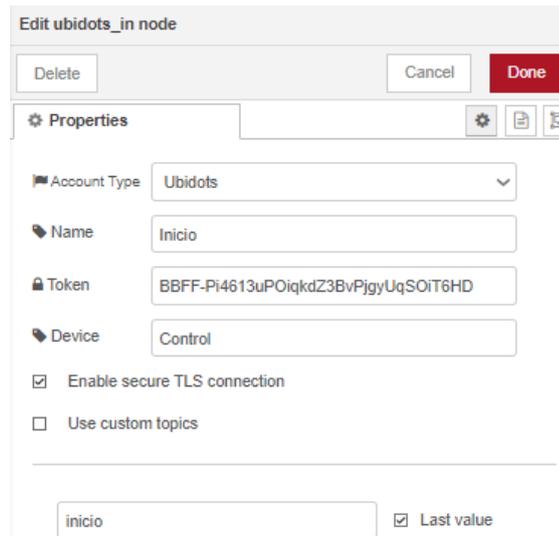
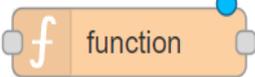


Figura 4.12: Ventana de comunicación para configuración del bloque de comunicación de entrada.

## 4.7 PALETAS

La Paleta se encuentra ubicada en la parte izquierda y contiene todos los nodos instalados y disponibles para su uso. Estos se encuentran organizados en varias categorías; además sobre la paleta se encuentra una sección donde se puede filtrar la lista de nodos [17]. A continuación, presentamos una lista con los nodos dentro de la paleta de Node-Red.

Tabla 4.1 Descripción de nodos dentro de la paleta Node-red, sección function.[33]

NOMBRE	GRAFICO	DESCRIPCION
<u>Nodo Function</u>		Trabaja con el código JavaScript, transporta cada mensaje que se pasa a través de él. El mensaje llega como un objeto denominado “msg”. Dentro de este bloque se puede trabajar con programación para usar el mensaje entrante.

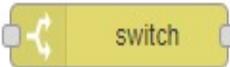
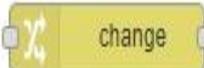
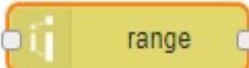
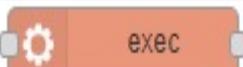
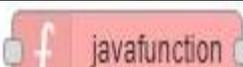
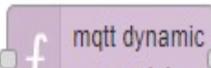
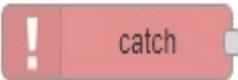
<u>Nodo Switch</u>		Autoriza que los mensajes se enruten a diferentes ramas de un flujo mediante la evaluación de un grupo de reglas para cada mensaje. La configuración del nodo esta con la propiedad de probar, que puede ser una propiedad de mensaje o de contexto.
<u>Nodo change</u>		Se utiliza para cambiar las propiedades de un mensaje y establecer propiedades de contexto sin tener que recurrir a un nodo Función. Cada nodo se puede configurar para múltiples operaciones que se aplican en orden.
<u>Nodo range</u>		Toma un número perteneciente a un rango de escala particular, y a su vez lo convierte en un nuevo número que pertenece a un rango diferente.
<u>Nodo template</u>		Se utiliza para generar texto utilizando las propiedades de un mensaje para completar una plantilla. El lenguaje que se utiliza para cada plantilla es denominado “Moustache”.
<u>Nodo trigger</u>		Se logra repetir un mensaje en un período arbitrario.
<u>Nodo exec</u>		Permite ejecutar sus funciones en un bloque javascript.
<u>Nodo file function</u>		Cumple la misma función del nodo principal, a diferencia que este nodo se carga el script para ser ejecutado desde un archivo real en su unidad.
<u>Nodo Javafunction</u>		En Node-RED pueden usar el lenguaje Java en lugar de JavaScript en el nodo de función.
<u>Nodo mqtt dynamic complete:</u>		Permite que el disparador continúe recibiendo nodos.

Tabla 4.2 Descripción de nodos dentro de la paleta Node Red sección common. [33]

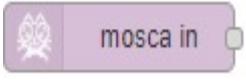
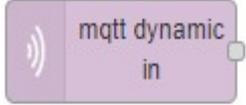
NOMBRE	GRAFICO	DESCRIPCION
<u>Nodo Inject</u>		Proporciona formas de inyectar cadenas, números, valores booleanos, marcas de tiempo y también variables de "flujo" y "global". Las variables globales y de flujo le permiten pasar valores de variable entre nodos, ya sea dentro de un flujo o en cualquier nodo de cualquier pestaña.
<u>Nodo Debug</u>		Se puede usar para mostrar mensajes en la barra lateral Debug dentro del editor. La barra lateral proporciona una vista estructurada de los mensajes que se envían, lo que facilita la exploración del mensaje.
<u>Nodo catch</u>		Se puede configurar para detectar errores de nodos específicos en el flujo o de cualquier nodo. Esto le permite acceder a crear diferentes flujos de manejo de errores para diferentes nodos
<u>Nodo status</u>		Mientras se ejecuta, un nodo puede compartir información de estado con la interfaz de usuario del editor.
<u>Nodo link in</u>		Puede conectarse e intercambiar datos entre nodos que pertenecen a diferentes flujos, este nodo sirve solo para datos de salida.
<u>Nodo link out</u>		Puede conectarse e intercambiar datos entre nodos que pertenecen a diferentes flujos, este nodo sirve solo para datos de entrada.

<u>Nodo comment</u>		Sirve para comentar para cada nodo y nombrar cada etiqueta
---------------------	---	--

Tabla 4.3 Descripción de nodos dentro de la paleta Node Red sección sequence.[33]

NOMBRE	GRAFICO	DESCRIPCION
<u>Nodo Split</u>		Permite separar el contenido del mensaje. Este nodo permite usar una sola parte del mensaje que llega, si se desea unir nuevamente el mensaje se debe combinar con un "Joinnode"
<u>Nodo join</u>		Controla el orden en que se unen los mensajes.
<u>Nodo sort</u>		Ordena la secuencia según el valor de una propiedad o el resultado de una expresión JSONata
<u>Nodo batch</u>		Crea secuencias de mensajes a partir de los mensajes recibidos.

Tabla 4.4 Descripción de nodos dentro de la paleta Node Red sección input.[33]

NOMBRE	GRAFICO	DESCRIPCION
<u>Nodo mosca in</u>		Es independiente de la plataforma, y se ejecuta donde se ejecuta Node-Js
<u>Nodo mqtt dynamic in</u>		Basado en el núcleo Node-red "mqtt in" que usa el nodo de configuración "mqtt-broker".
<u>Nodo dweetio in</u>		Este nodo no admite dweets privados o "bloqueados"

<u>Nodo ubidots_in</u>		Publica y se suscribe a una o múltiples variables. También brinda la posibilidad de suscribirse hasta 10 temas personalizados de Ubidots.
------------------------	---	---

Como se puede notar Node-Red ofrece una amplia gama de nodos para poder desarrollar aplicaciones. Para el presente proyecto de investigación únicamente se utilizarán algunos nodos. El detalle de cómo utilizarlos se puede encontrar en el capítulo número 5, en donde se proporciona una guía detallada para el desarrollo de aplicaciones.

## **CAPITULO 5: INTEGRACION DE HERRAMIENTAS INFORMATICAS Y DESARROLLO DE APLICACIONES**

### **5.1 PLCSIM ADVANCE**

PLC Sim Advanced es una herramienta de simulación para la creación de un PLC virtual con determinadas funciones de comunicación. Al iniciar el programa por defecto aparecerá seleccionada la opción “PLCSIM”. Con esta configuración solo es posible trabajar de forma local, por consiguiente, es necesario cambiar a la opción “PLCSIM Virtual Eth. Adapter”. Esta opción habilita las opciones de comunicación TCP/IP. Este proceso se ilustra de mejor manera en la figura 5.1.

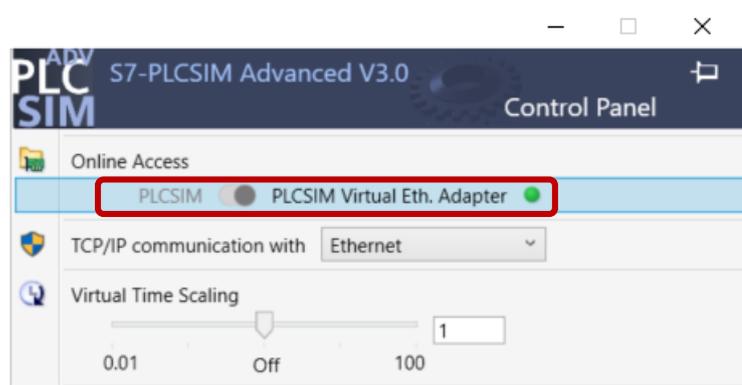


Figura 5.1 Ventana de comunicación para creación de nuevo PLC virtual.

Posterior a esto, se debe agregar un nombre al PLC virtual. En la sección de “IP Address” se debe especificar una dirección IP libre dentro de la red de trabajo. En adición debemos especificar una máscara de subred y una puerta de enlace. Al terminar se debe dar clic en el botón “Start”. Este proceso creará un nuevo PLC virtual con la dirección IP asignada. Los detalles de la configuración se ilustran en la figura 5.2.

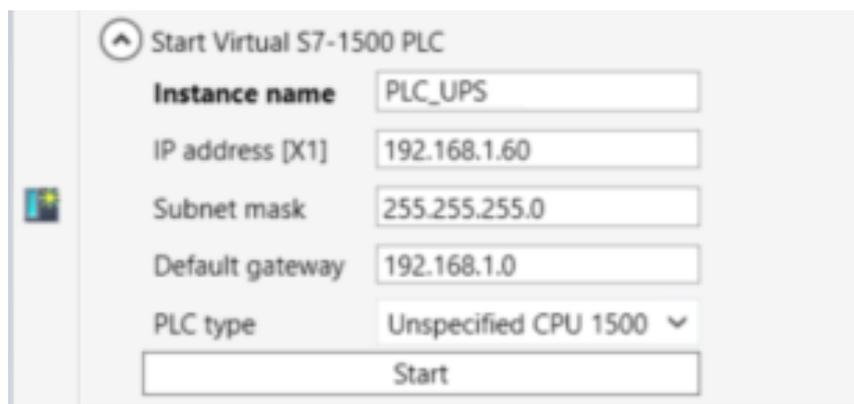


Figura 5.2 Asignación de nombre y dirección IP.

Luego de ejecutar todo este proceso se visualizará una alerta en el PC, anunciando la detección del nuevo PLC (véase la figura 5.3).



Figura 5.3 Ventana de comunicación, advertencia nueva IP detectada.

Para finalizar, en la parte inferior del programa se puede observar que el PLC virtual está activo y listo para comunicarse (véase la figura 5.4)

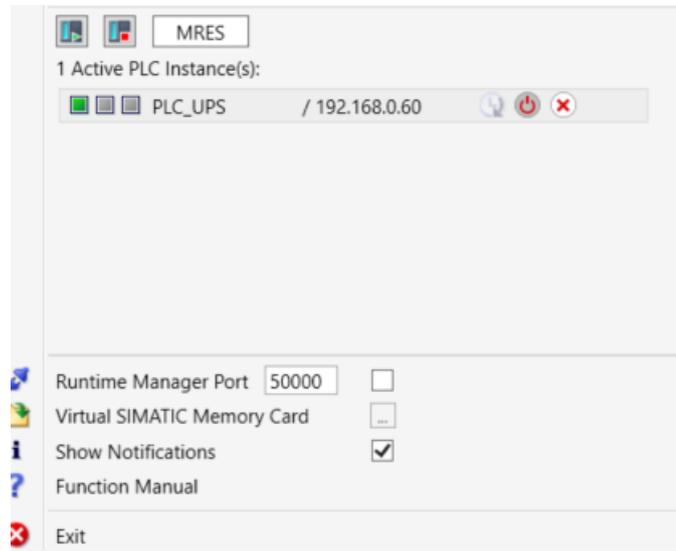


Figura 5.4 Nuevo PLC virtual creado.

## 5.2 FACTORY I/O.

### 5.2.1 CONFIGURACIÓN.

Factory IO tiene la capacidad de conectarse con un PLC virtual, para lo cual se usa una interfaz amigable y fácil. Para configurar el PLC que se va a conectar es necesario dirigirse a “Archivos”, luego a “Drivers” y posteriormente a “Configuración”. A continuación, se debe elegir el PLC. En la figura 5.5 se muestra el resultado de todo este proceso donde se ha seleccionado la opción para conexión con un PLC S7-1200 o S7-1500 de la firma SIEMENS. En la sección “Host” se debe colocar la dirección con la que se creó el PLC virtual con PLCsim Advance (véase la figura 5.6) En la configuración del autómeta podemos decidir el tipo de dato numérico, puede ser “DWORD” o “WORD”. En la sección de terminales se determina la cantidad de entradas y salidas tanto analógicas como digitales (véase la figura 5.7). Para guardar la configuración se debe regresar con la flecha de retroceso y los cambios se reflejarán. El resultado de este proceso se muestra en la figura 5.8

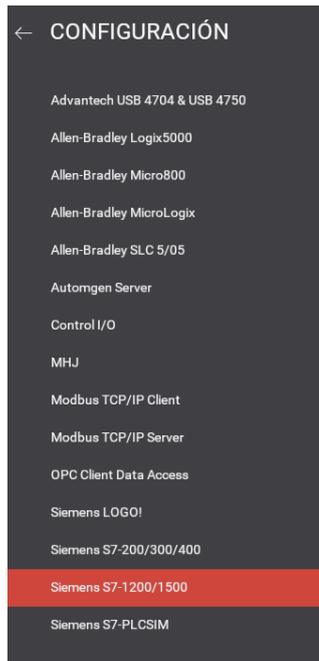


Figura 5.5 Configuración Inicial Factory IO.

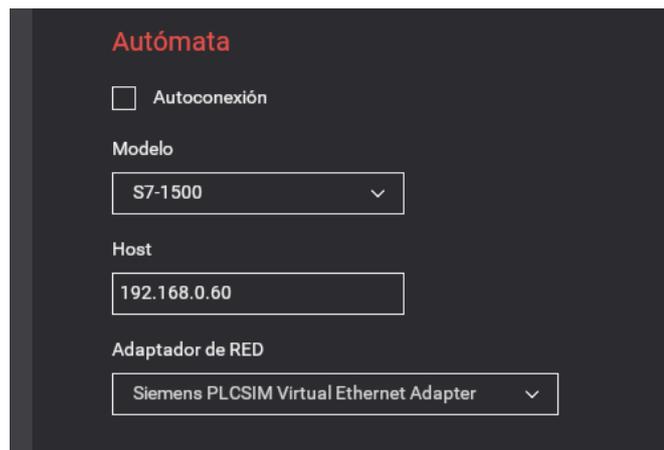


Figura 5.6 Definición Autómata, dirección IP.



Figura 5.7 Configuración terminales.

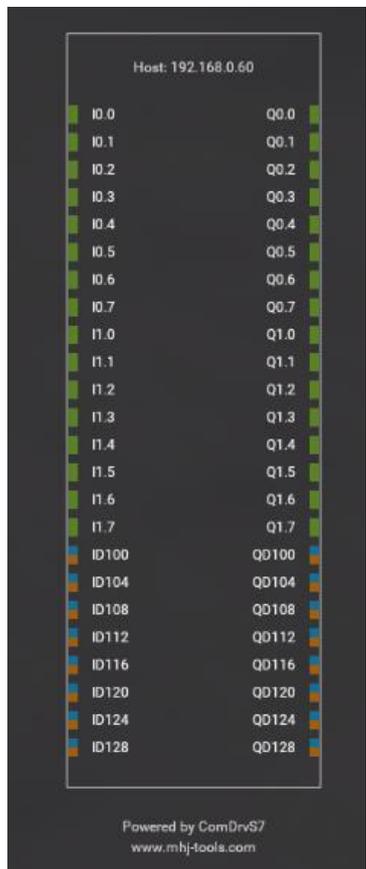


Figura 5.8 Dispositivo Creado.

### 5.2.2 CONEXIÓN CON EL AUTOMATA.

Para realizar la conexión entre la herramienta gráfica Factory IO y el PLC virtual, se debe dar clic en el botón “Conectar” (véase la figura 5.9). Si la herramienta encuentra el PLC en la dirección proporcionada la conexión se realizará y se visualizará un “check” verde como mecanismo de confirmación (véase la figura 5.10).

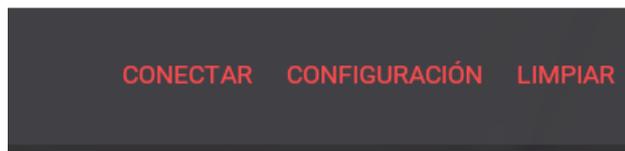


Figura 5.9 Botón Conectar.

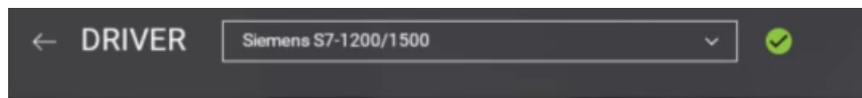


Figura 5.10 Conexión exitosa.

### 5.2.3 DEFINICIÓN DE ENTRADAS Y SALIDAS

La definición de las entradas y salidas dentro de la herramienta debe ser realizada por el usuario, arrastrando cada etiqueta hacia la entrada deseada. Es importante tener en cuenta que las etiquetas de las entradas y salidas deben coincidir en aquellas utilizadas en el programa TIA Portal. Existen entradas y salidas tanto digitales como analógicas y vienen diferenciados por colores. El color verde corresponde a una variable de tipo digital, mientras que el color azul corresponde a una variable analógica (véase la figura 5.11).

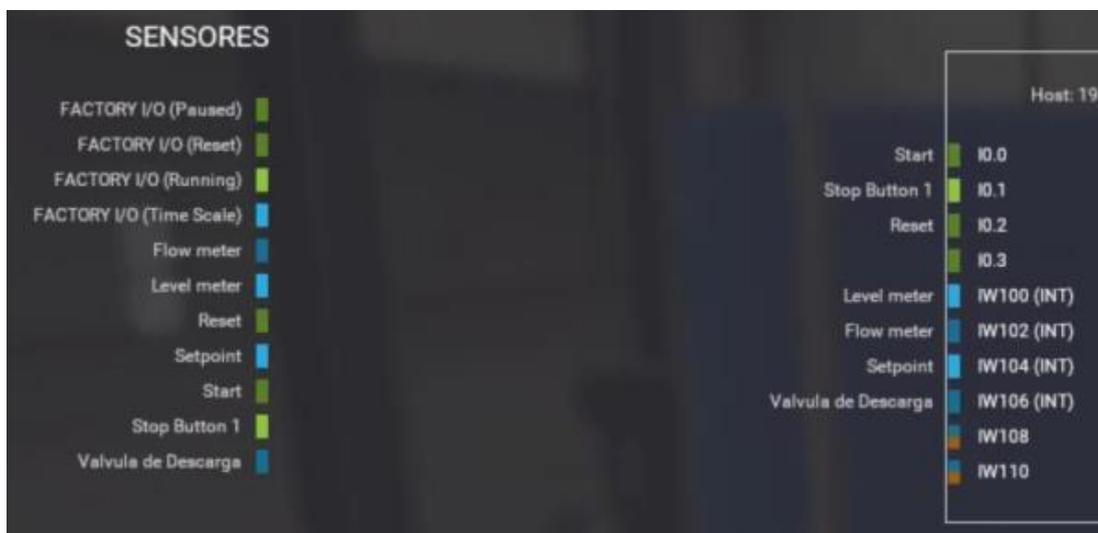


Figura 5.11 Asignación de variables del proceso.

## 5.3 TIA PORTAL

### 5.3.1 CONFIGURACIÓN INICIAL.

TIA Portal es una herramienta informática que permite la configuración, puesta en marcha y programación de dispositivos industriales de la marca Siemens. Principalmente permite la configuración y programación de PLCs y paneles HMI. Para el presente proyecto se utilizará un autómata de la familia s7-1500 para controlar los procesos virtualizados en Factory I/O. Para iniciar con TIA Portal, es necesario crear un proyecto y adicionar un autómata. Para esto nos dirigimos a la opción “agregar dispositivo” y desplegamos el menú de controladores. En este caso específico adicionaremos un CPU 1511C-1 PN, el cual tiene incorporado dos puertos de comunicación basados en Ethernet. En la figura 5.12 se ilustra el cuadro de diálogo de selección, así como la referencia y la versión de “firmware” especificada.

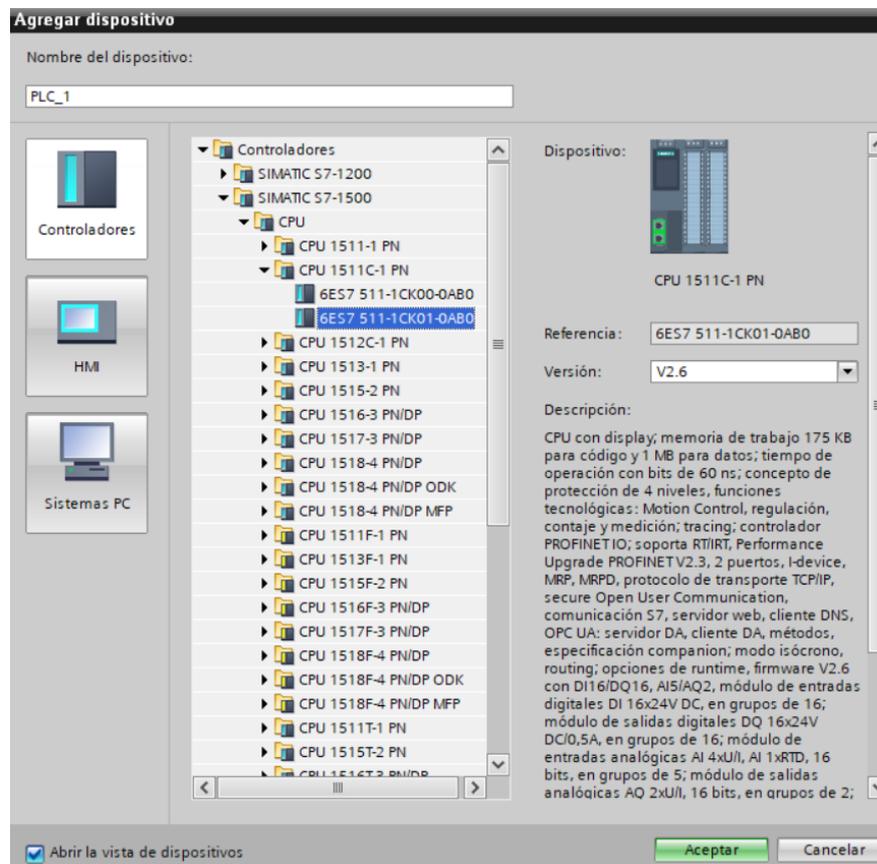


Figura 5.12 Creación nuevo dispositivo.

Una vez agregado el dispositivo, en la parte izquierda de la pantalla, dentro del Árbol del proyecto se debe acceder a las propiedades del proyecto (véase la figura 5.13) y se debe verificar que dentro de la sección de Protección se encuentre habilitada la opción “Permitir simulación al compilar por bloques” (véase la figura 5.14). Esta opción activa permitirá que el proyecto sea descargado y simulado con PLCSIM.

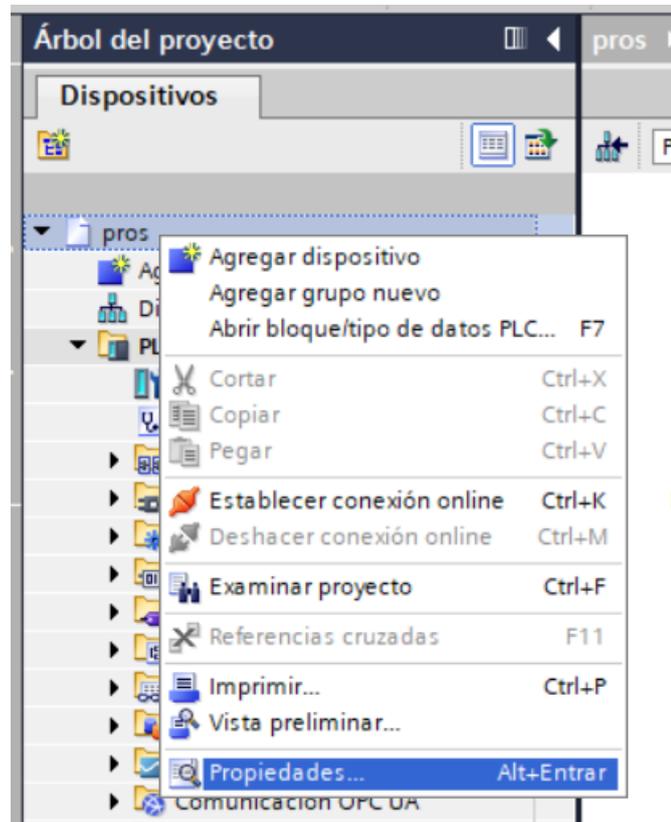


Figura 5.13 Menú desplegable para acceder a las Propiedades de un proyecto.

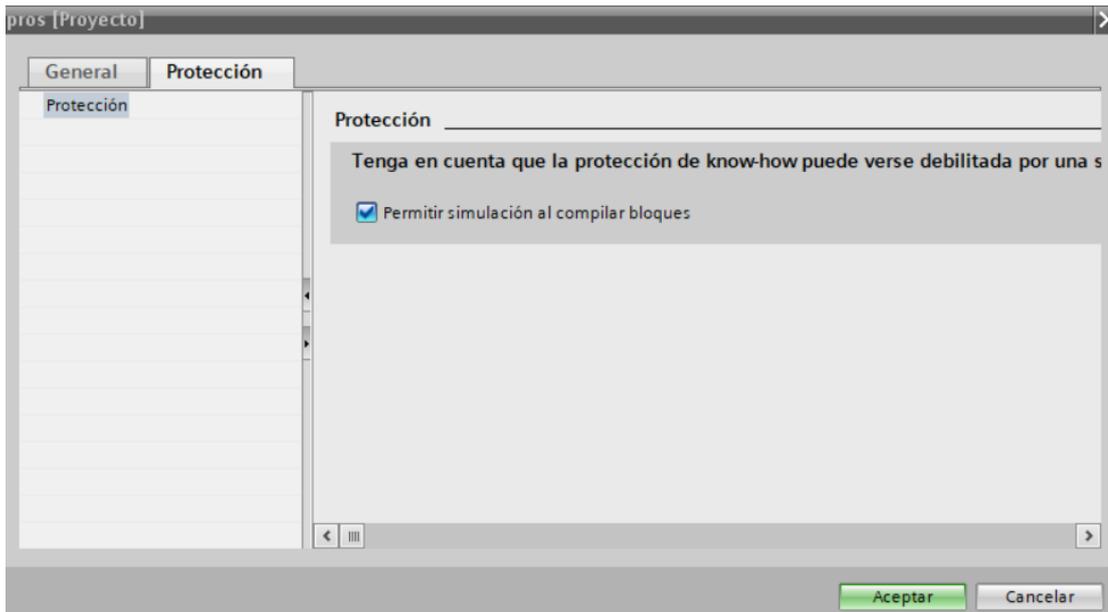


Figura 5.14 Cuadro de diálogo para permitir simulación al compilar bloques.

Para que la herramienta informática reconozca el autómata virtual es necesario modificar las propiedades del PLC como se realiza en la figura 5.15. La comunicación con el autómata virtual se realiza mediante una dirección IP, la cual se debe especificar en la pestaña General, dentro de la sección Direcciones Ethernet. Esto se ilustra de mejor manera en la figura 5.16.

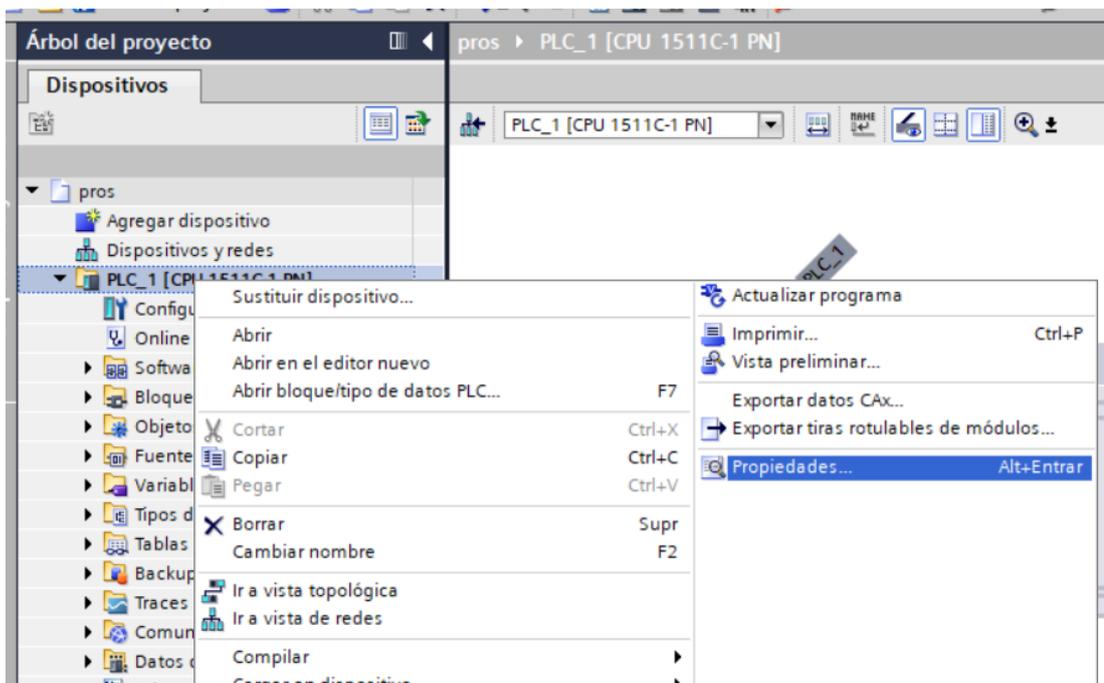


Figura 5.15 Cambiar Propiedades.

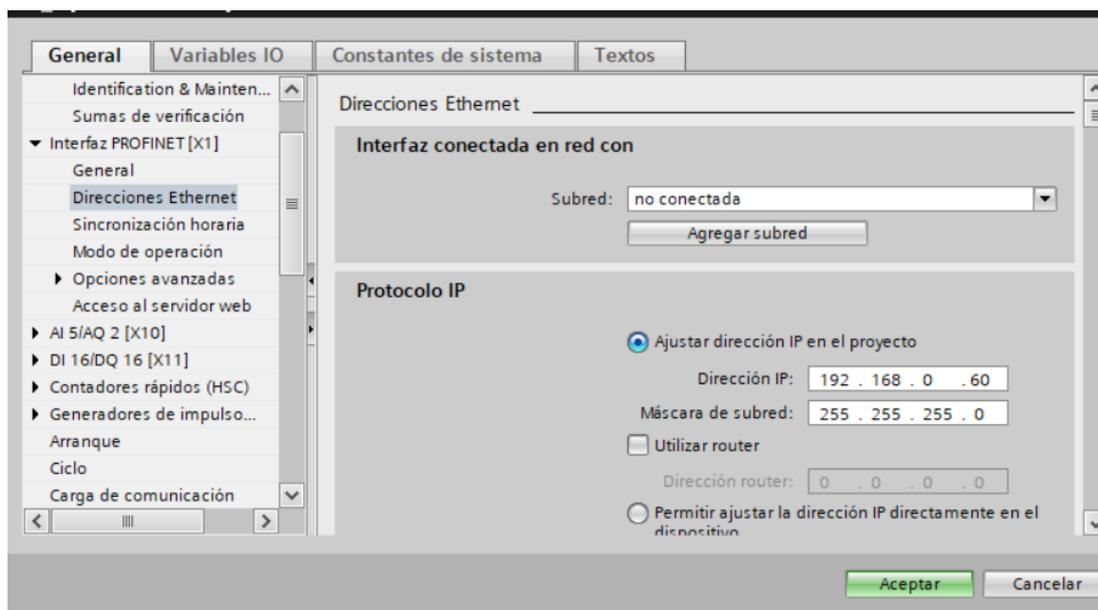


Figura 5.16 Modificación dirección IP.

En el caso de que se desee habilitar instrucciones de comunicación, dentro de la sección de Protección y Seguridad se debe habilitar la opción "Permitir acceso vía comunicación PUT/GET del interlocutor remoto". Este cuadro de diálogo se ilustra en la figura 5.17.

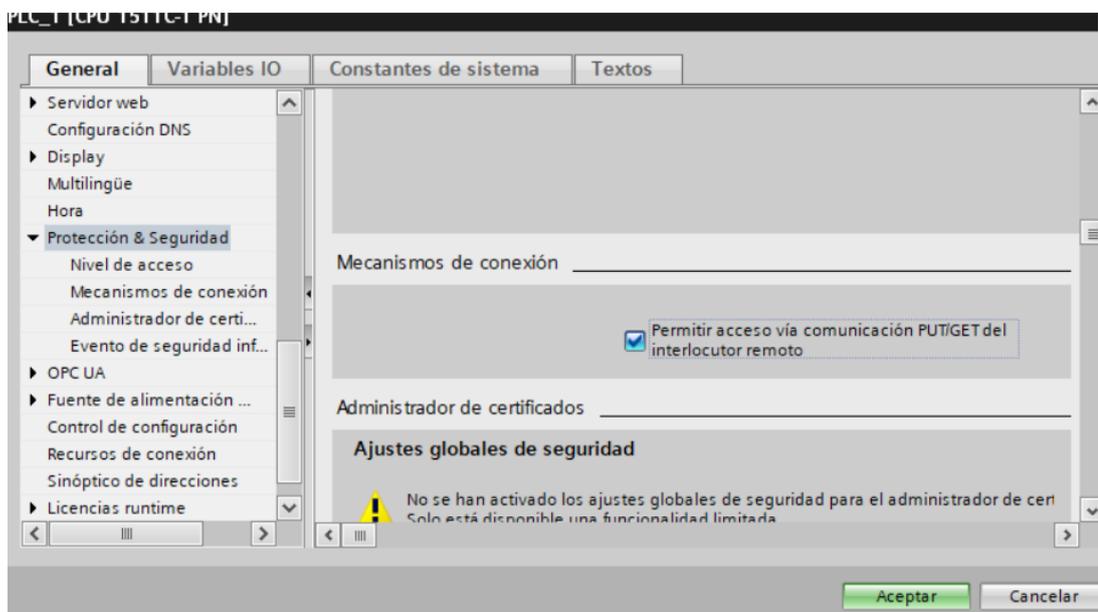


Figura 5.17 Activación opción PUT/GET.

### 5.3.2 ADMINISTRACIÓN DE VARIABLES.

Para una mejor administración de las variables del proyecto se puede establecer un bloque de datos desde la carpeta ubicada en el menú izquierdo con el nombre Bloques de Programa (véase la figura 5.18). TIA Portal almacena los datos de este nuevo bloque en lugares aleatorios de su memoria. Para poder obtener la dirección exacta de la memoria donde es guardada la variable es necesario cambiar las propiedades del nuevo bloque de datos. La figura 5.19 muestra el cuadro de diálogo dentro de las propiedades en la pestaña General en la sección Atributos. En esta parte se debe verificar que esté inactiva la opción acceso optimizado al bloque.

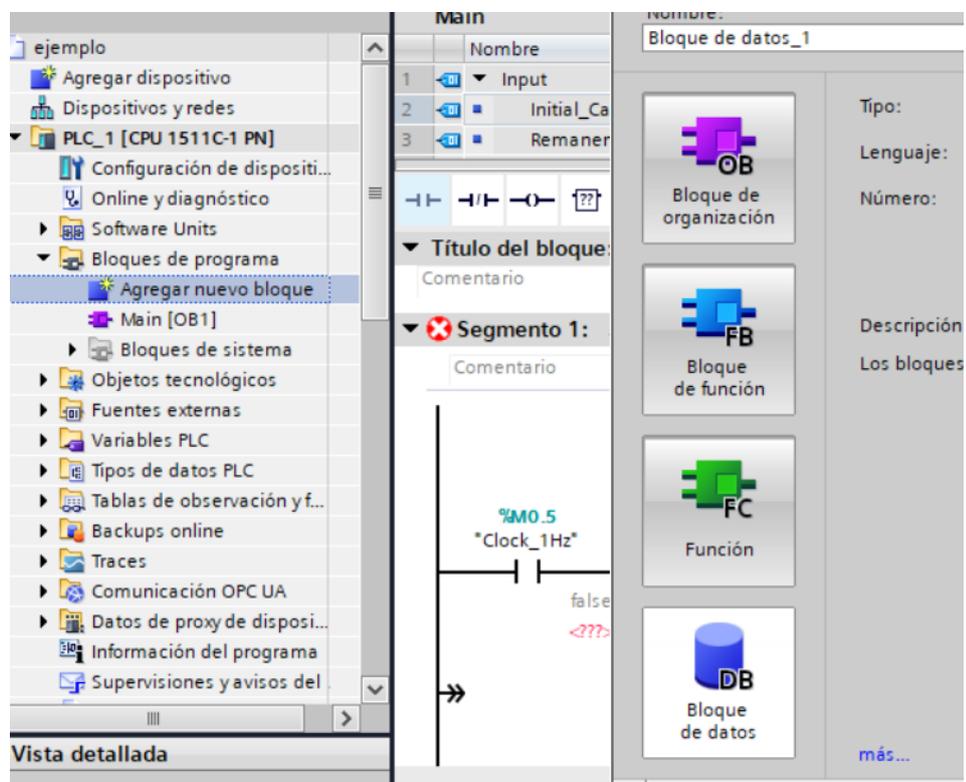


Figura 5.18 Creación Bloque de Datos.

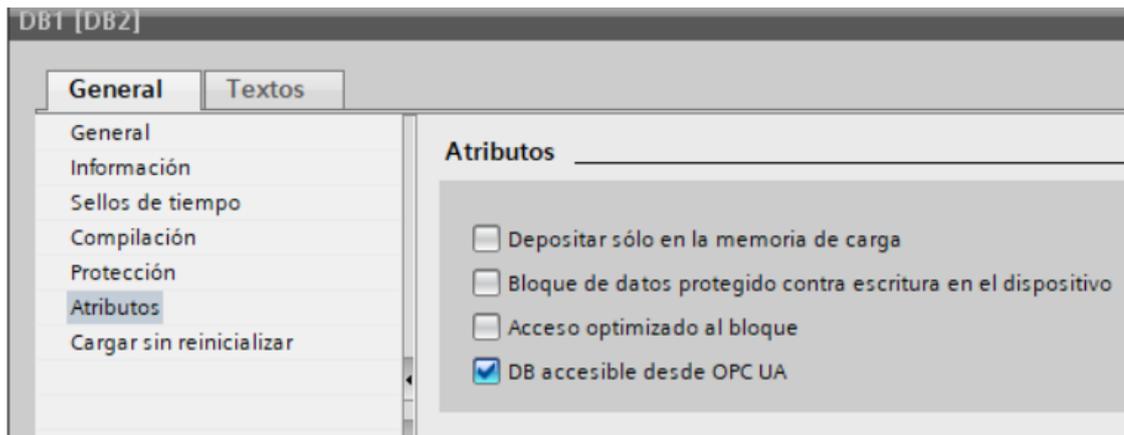


Figura 5.19 Configuración Bloque de Datos.

Dentro del nuevo bloque se debe crear las variables a utilizar. El lugar exacto de la memoria donde serán almacenadas se visualizará en la columna con el nombre de Offset, este dato va a ser usado para la conexión con Ubidots en la sección 5.4.2.1 (Configuración Nodos PLC).

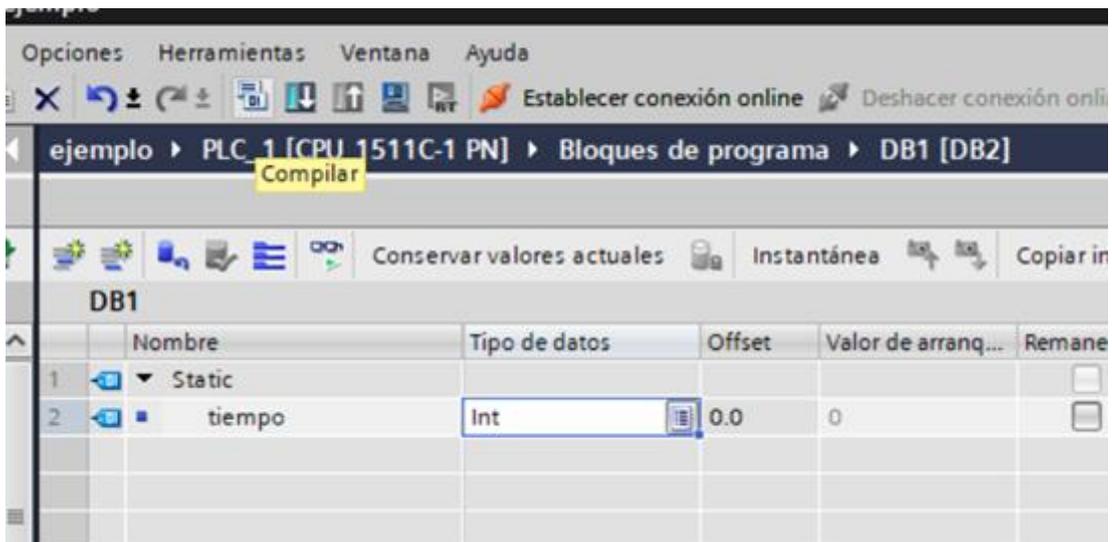
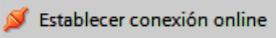


Figura 5.20 Obtención Offset.

Imagen propia

### 5.3.3 CONEXIÓN CON PLCSIM

Para establecer la conexión con el Automata es necesario dar clic sobre el icono en la parte superior del programa , esto abrirá la ventana de comunicación como se muestra en la figura 5.21, existen varios tipos de interfaz usar la interfaz PN/IE además seleccionar el adaptador de red propia de la computadora a usar.

Para realizar la conexión online con PLCSIM, en TIA Portal es necesario modificar la manera en la que se busca al autómata, usar la opción “Mostrar dispositivos con direcciones idénticas” (seleccionado en la imagen con un cuadrado rojo) esta modificación permitirá reconocer la IP del autómata virtual, esta IP es idéntica a la creada dentro del programa PLC SimAdvance, en la figura 5.22 se muestra la detección del autómata con la dirección IP 192.168.0.60.

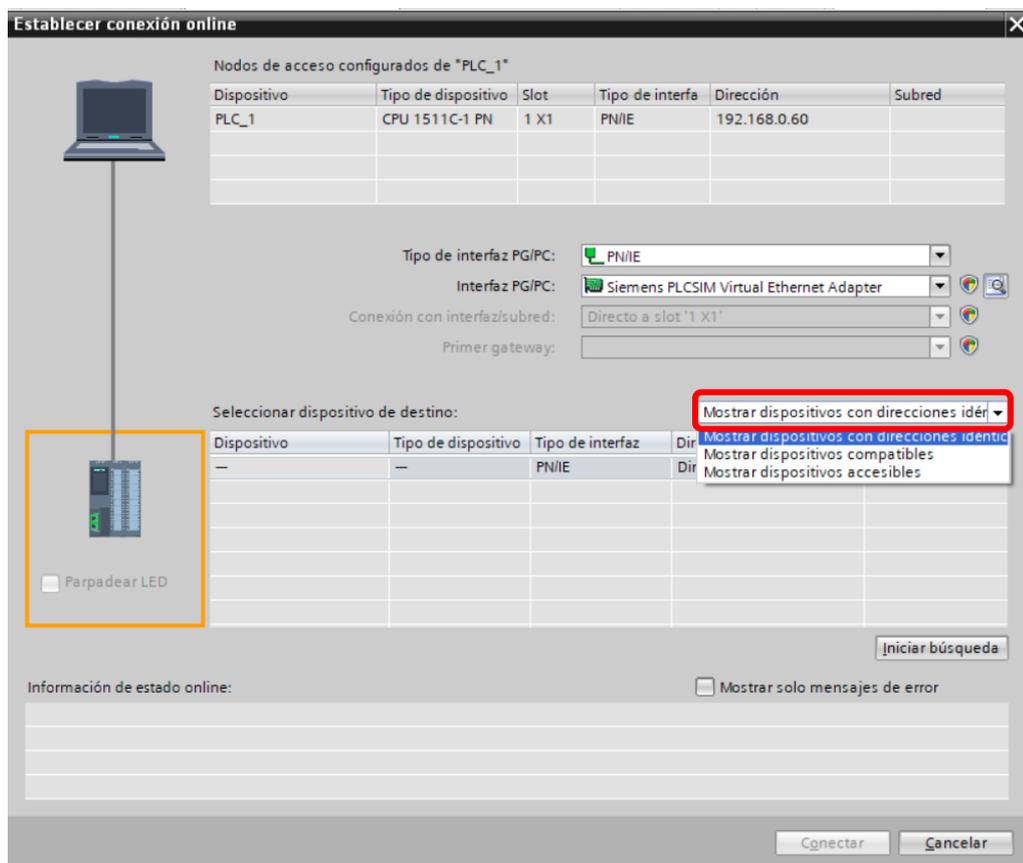


Figura 5.21 Conexión Automata.

Imagen propia

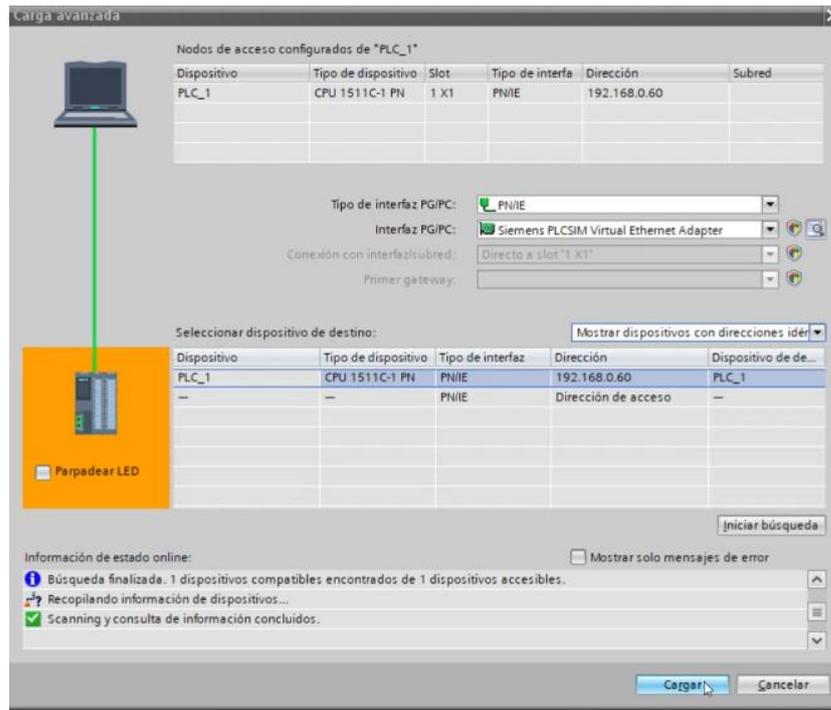


Figura 5.22 Conexión Exitosa.

Imagen propia

## 5.4 NODE RED.

### 5.4.1 INSTALACIÓN.

Node red es una herramienta informática, diseñada para crear flujos y nodos para la publicación de datos en la nube. La herramienta informática no tiene una instalación tradicional, es necesario ingresar a su página web oficial en la dirección “[nodered.org/docs/getting-started/Windows](http://nodered.org/docs/getting-started/Windows)” a continuación, seguir los pasos como los indica en su página, en la figura 5.23 muestra el punto 1; instalar en el equipo la aplicación Node.js (Install Node.js)

## Quick Start

### 1. Install Node.js

Download the latest 12.x LTS version of Node.js from the official [Node.js home page](#). It will offer you the best version for your system.

Run the downloaded MSI file. Installing Node.js requires local administrator rights; if you are not a local administrator, you will be prompted for an administrator password on install. Accept the defaults when installing. After installation completes, close any open command prompts and re-open to ensure new environment variables are picked up.

Once installed, open a command prompt and run the following command to ensure Node.js and npm are installed correctly.

Using Powershell: `node --version; npm --version`

Using cmd: `node --version && npm --version`

You should receive back output that looks similar to:

Figura 5.23 Página Oficial Node-Red.

Imagen propia

Dentro de la página es necesario descargar un programa para instalar Node.js usar la versión recomendada “14.15.4 LTS”, la única característica necesaria es tener Windows (x64) como se muestra en la figura 5.24. Terminada su descarga se procede a iniciar la instalación en la máquina, seguir todos los pasos mostrados en el cuadro de dialogo hasta finalizar la instalación véase figura 5.25.

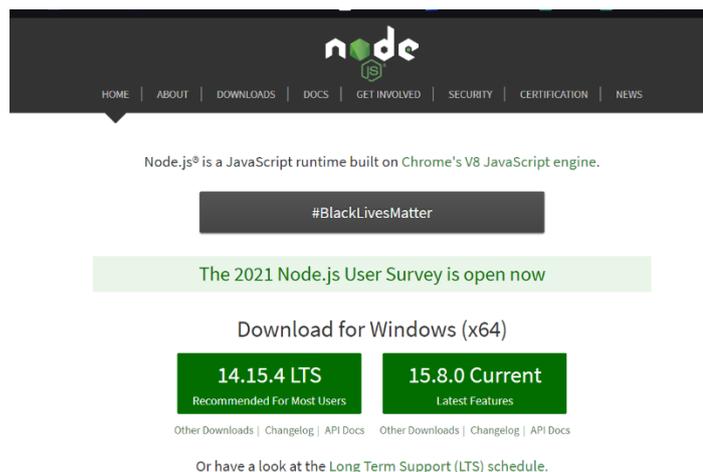


Figura 5.24 Pagina de Descarga Node.js.

Imagen propia

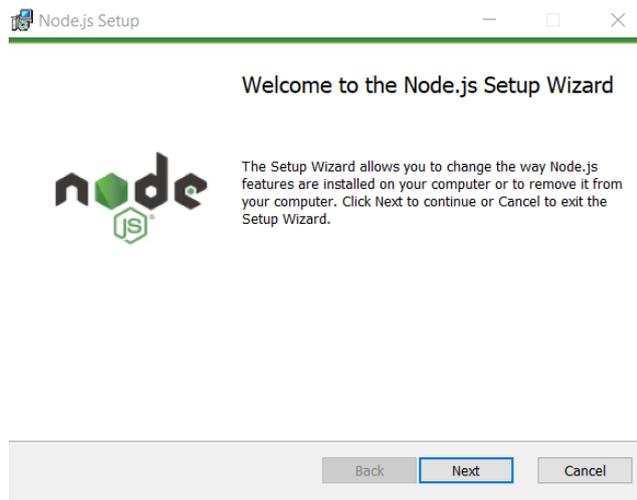


Figura 5.25 Dialogo de Instalación Node.Js.

Imagen propia

Continuar con el paso 2 dentro de la página oficial de Node Red, iniciar instalación mediante el símbolo de sistema *como administrador*, donde se usará el siguiente comando para instalar Node-Red “`npm install -g --unsafe-perm node-red`”, este línea de comando procederá con la instalación, la figura 5.26 muestra advertencias durante la instalación, estas advertencias son normales.

```
C:\WINDOWS\system32>npm install -g --unsafe-perm node-red
npm WARN deprecated bcrypt@3.0.6: versions < v5.0.0 do not handle NUL in passwords
npm WARN deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated bcrypt@3.0.8: versions < v5.0.0 do not handle NUL in passwords
C:\Users\Marko\AppData\Roaming\npm\node-red -> C:\Users\Marko\AppData\Roaming\npm\node-red
C:\Users\Marko\AppData\Roaming\npm\node-red -> C:\Users\Marko\AppData\Roaming\npm\node-red
+ node-red@1.1.3
updated 1 package in 32.194s
```

Figura 5.26 Instalación Node-Red.

Imagen propia

Al terminar la instalación, iniciar el programa mediante el símbolo de sistema con el siguiente comando: “`Node-red`” como muestra la figura 5.27. Este comando inicia una serie de procesos, una de las líneas iniciales muestra la dirección IP con la que

podremos usar Node red, mirar figura 5.28 ingresar esa dirección en un navegador, estos pueden ser Google Chrome, Firefox, Internet Explorer, Opera, etc.

```
C:\WINDOWS\system32>node-red
```

Figura 5.27 Comando para Iniciar Node-Red.

```
18 Aug 20:56:18 - [warn]
18 Aug 20:56:18 - [info] Settings file : \Users\Marko\.node-red\settings.js
18 Aug 20:56:18 - [info] Context store : 'default' [module=memory]
18 Aug 20:56:18 - [info] User directory : \Users\Marko\.node-red
18 Aug 20:56:18 - [warn] Projects disabled : editorTheme.projects.enabled=false
18 Aug 20:56:18 - [info] Flows file : \Users\Marko\.node-red\flows_MarcoOrellana.json
18 Aug 20:56:18 - [info] Server now running at http://127.0.0.1:1880/
18 Aug 20:56:18 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
```

Figura 5.28 Dirección IP necesaria para iniciar aplicación.

Imagen propia

La figura 5.29 muestra la interfaz principal de la herramienta informática Node Red. Para la comunicación con la herramienta informática Tia Portal y Ubidots es necesario nodos adicionales. En la figura 5.30 se visualiza la manera en la que se debe agregar nodos mediante la opción “Manage Palette” ubicada en el menú en la parte superior derecha.

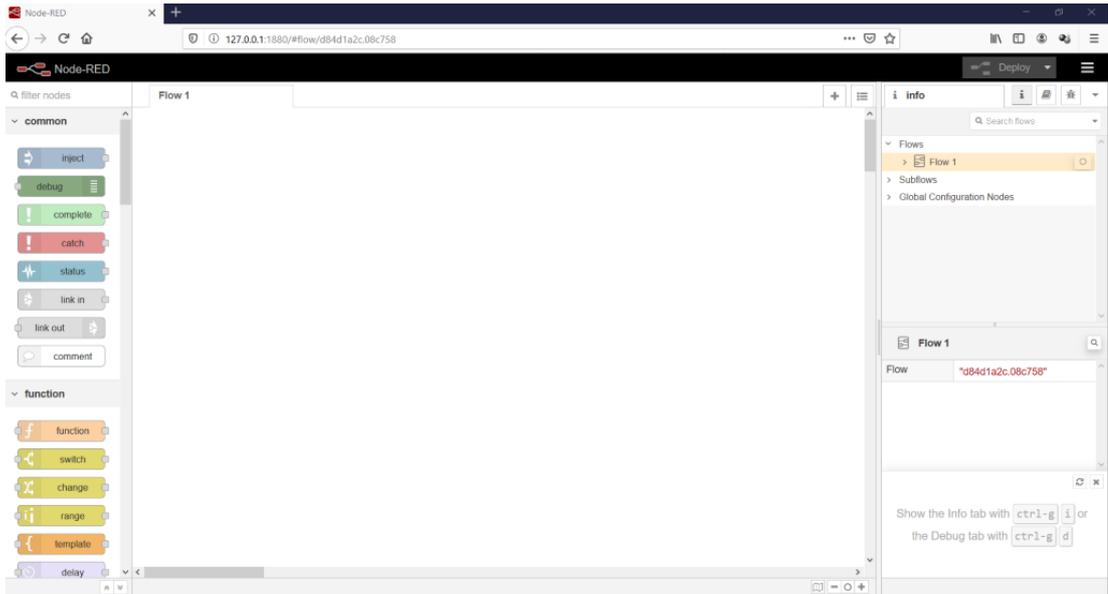


Figura 5.29 Interfaz Node-Red.

Imagen propia

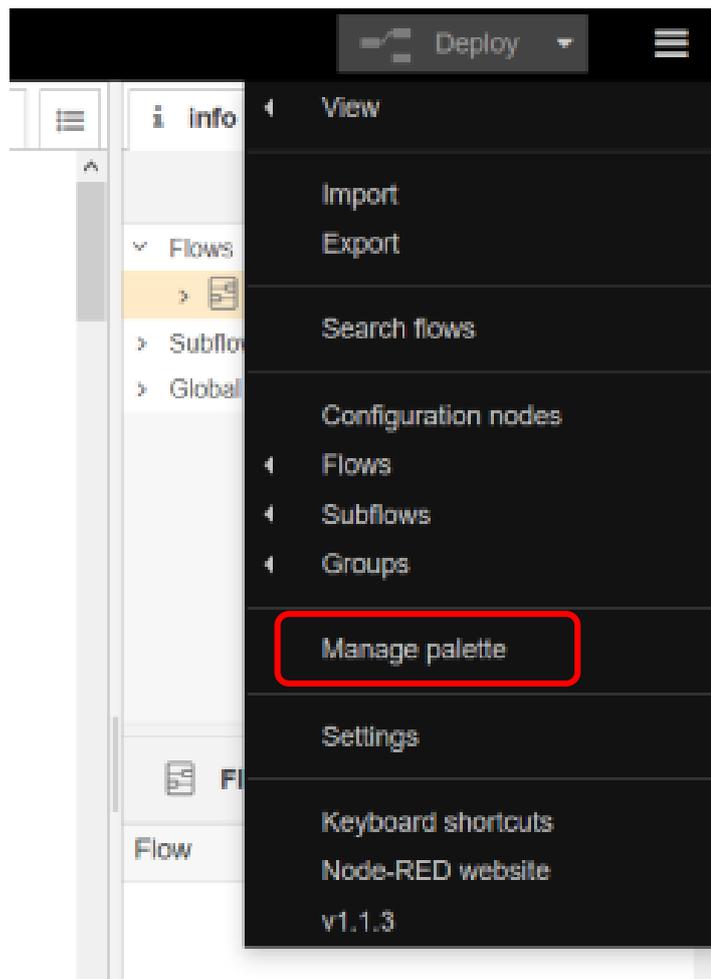


Figura 5.30 Manage Palette.

Imagen propia

Los nodos a instalar son desarrollados por una comunidad de la cual vamos a apalancarnos, luego de dar clic en el botón Manage palette tendremos una ventana de comunicación, dirigirse a la pestaña “Install” donde se puede buscar e instalar los paquetes mediante su nombre, para el presente proyecto buscar el nodo “node-red-contrib-s7” como muestra la figura 5.31, seleccionar instalar y en breves minutos se crea una sección en la página principal con los nuevos nodos, adicional también buscar el nodo “Ubidots-nodered” mirar figura 5.32

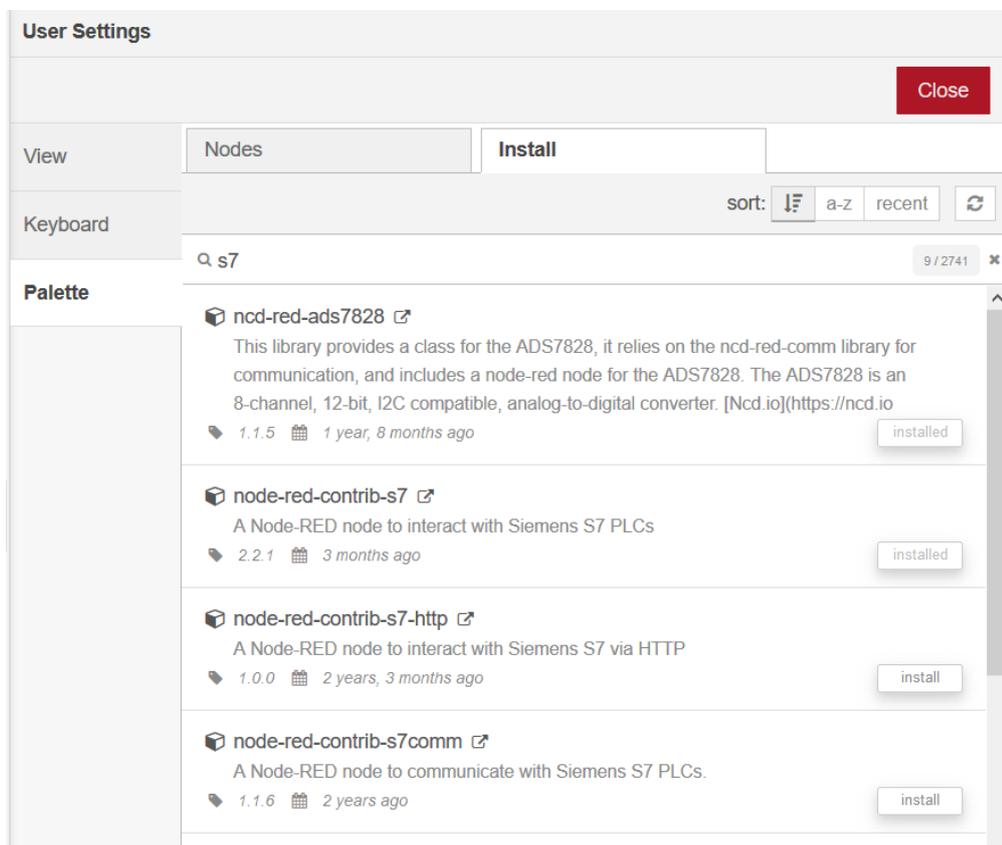


Figura 5.31 Nodo a instalar.

Imagen propia

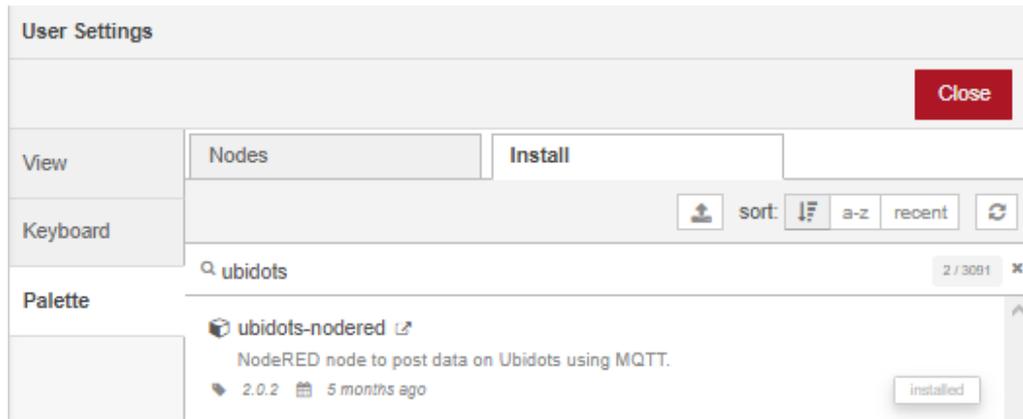


Figura 5.32 Nodo para Ubidots.

Imagen propia

## 5.4.2 COMUNICACIÓN NODE RED CON PLC MEDIANTE NODOS

La comunicación con el autómatas se realiza mediante los nodos instalados en el punto anterior, existen dos tipos como muestra la figura 5.33:

- S7 in: Permite la obtención de datos desde el PLC almacenándolo para posteriormente enviarlo.
- S7 out: Permite el envío de datos hacia el PLC.



Figura 5.33 Nodos comunicación PLC.

Imagen propia

## 5.4.3 CONFIGURACIÓN NODOS PARA COMUNICACIÓN PLC

Para usar los nodos basta con arrastrarlos hacia el espacio de diseño, para configurar los nodos entrar a sus propiedades con doble clic, primero agregar la IP del PLC mediante el símbolo de lápiz; mirar figura 5.34. Dentro de las propiedades colocar la

dirección del autómata (PLC) a comunicar, en la sección “Slot” modificar el valor a 1; como se observa en la figura 5.35.

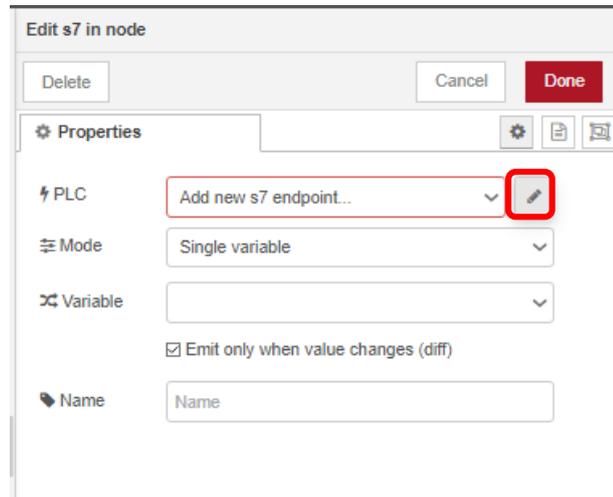


Figura 5.34 Agregar nuevo PLC.

Imagen propia

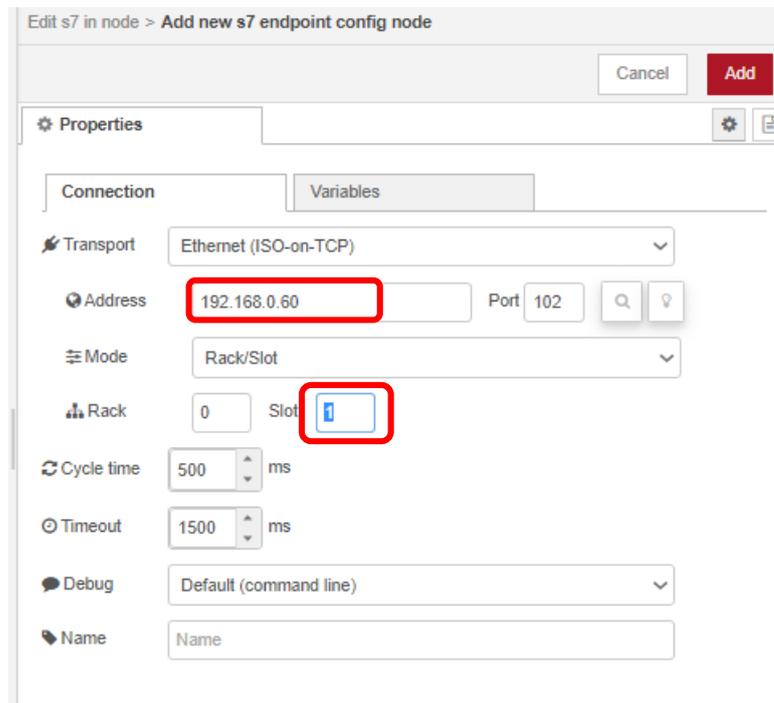


Figura 5.35 Configuración Nodo.

Imagen propia

Dentro de la configuración de propiedades dirigirse a la pestaña variables, en la sección lista de variables se coloca la dirección del dato a obtener con el dato “Offset” (separado por una coma), obtenido en la sección 5.2.2 (Variables a enviar); y el nombre asignado dentro de Tia portal; la estructura es la siguiente: Nombre del bloque de datos (DBx), tipo de dato (INT,REAL,BOOL,etc) Valor de Offset, mirar figura 5.36.

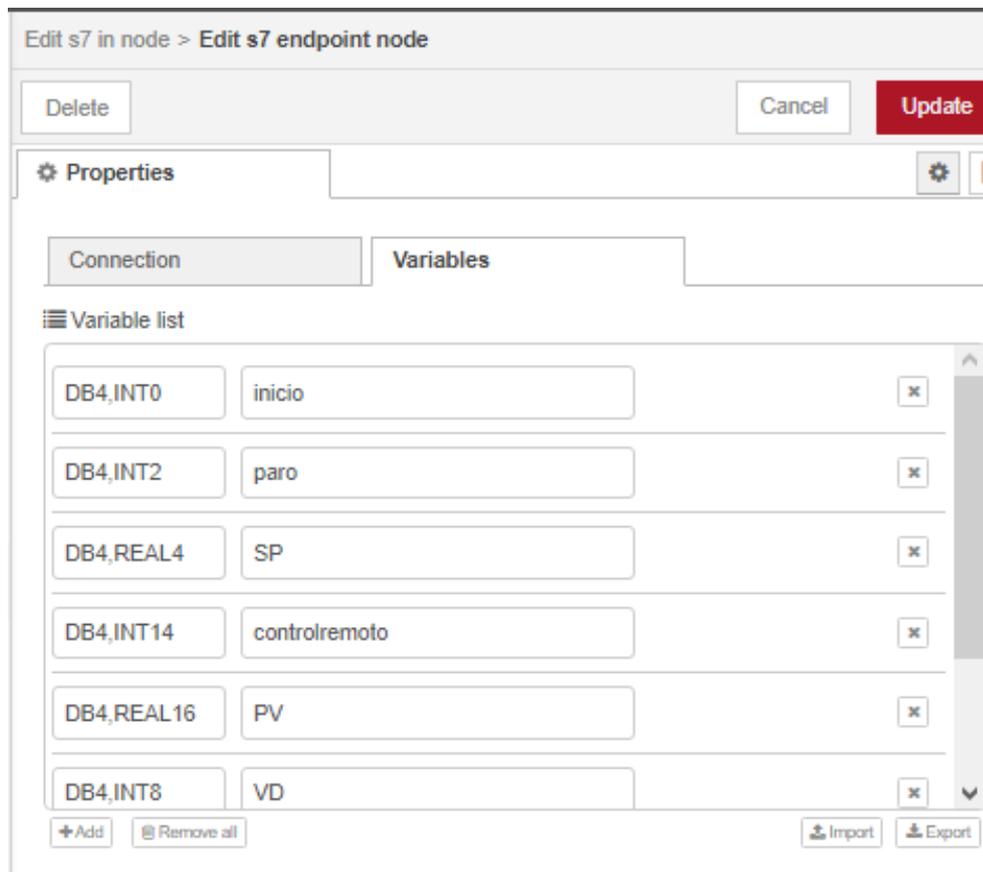


Figura 5.36 Asignación de Variables a obtener.

Imagen propia

#### 5.4.4 COMUNICACIÓN NODE RED CON UBIDOTS MEDIANTE NODOS

Los datos obtenidos hasta el momento deben ser enviados hacia la página web Ubidots, usar el nodo “Function” para enviar el dato hacia un dispositivo dentro de Ubidots como se muestra en la figura 5.37, dentro de este nodo programar con las siguientes líneas que permiten darle nombre a la variable dentro del dispositivo en Ubidots como se observa en la figura 5.38.

```

“var response = {};

response.payload = {"aquí_el_nombre": msg.payload};

response.topic = "/v1.6/devices/siemens";

return response;”

```

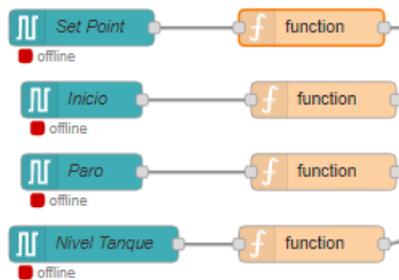


Figura 5.37 Bloque Function.

Imagen propia



Figura 5.38 Programación dentro del nodo Function.

Imagen propia

Usar el nodo de Ubidots instalado con anticipación, donde existen dos tipos:  
 Ubidots\_out: Permite enviar datos hacia Ubidots  
 Ubidots\_in: Permite obtener datos desde Ubidots

### 5.4.5 CONFIGURACION NODO UBIDOTS\_OUT.

El nodo para la comunicación con Ubidots pide seleccionar el tipo de cuenta que usa en la página web, seleccionar “Ubidots”, en la sección Name colocar el nombre a mostrar en Node red; es necesario obtener el token desde la página Ubidots véase la sección 5.4.1, en la opción “Device Label” se debe colocar el nombre del dispositivo a crear, este nodo crea automáticamente el dispositivo en la cuenta de Ubidots, mirar figura 5.39.

The image shows a configuration window titled "Edit ubidots\_out node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a "Properties" section with a gear icon and three sub-panels. The first sub-panel is "Account Type" with a dropdown menu set to "Ubidots". The second sub-panel is "Name" with a text input field containing "Nivel Agua". The third sub-panel is "Token" with a text input field containing "BBFF-hyrQWD29V1m28oqres6mplQAm7EV0X". Below these sub-panels is a "Device Label" section with a text input field containing "Entrada". At the bottom, there is a checkbox labeled "Enable secure TLS connection" which is checked.

Figura 5.39 Configuración nodo Ubidots Out.

Imagen propia

### 5.4.6 CONFIGURACION NODO UBIDOTS\_IN.

Este Nodo necesita información similar al nodo Ubidots\_out, obtiene los datos desde la página web por lo tanto en la sección “Device” colocar el nombre del dispositivo del que se requiere obtener el dato, como valor adicional añadir el nombre de la variable a obtener dentro del dispositivo de Ubidots en la sección remarcada en rojo en la figura 5.40.

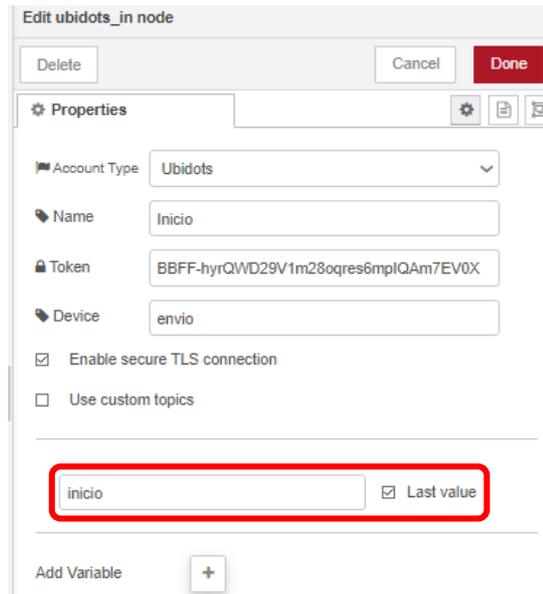


Figura 5.40 Configuración Ubidots In.

Imagen propia

### 5.4.7 COMUNICACIÓN UBIDOTS CON PLC MEDIANTE NODOS

La comunicación desde Ubidots, nos permite controlar el PLC desde la página web a través de la herramienta informática Node red, brindando la opción de tener el control sobre el PLC desde cualquier lugar del mundo.

Los datos que llegan desde Ubidots deben ser modificados para el reconocimiento del PLC, dentro de la paleta de nodos usar el nodo llamado Split, la función de este nodo es recortar el mensaje enviado por Ubidots en dos partes, no es necesario modificar el nodo para su funcionamiento como se muestra en la figura 5.41, posteriormente con un nodo function, obtener del mensaje recortado, identificar el dato a usar y enviarlo al PLC, la figura 5.42 muestra la configuración dentro de este nodo las líneas a programar son las siguientes:

***“var num = msg.payload.value;***

***msg.payload = num;***

***return msg;”***

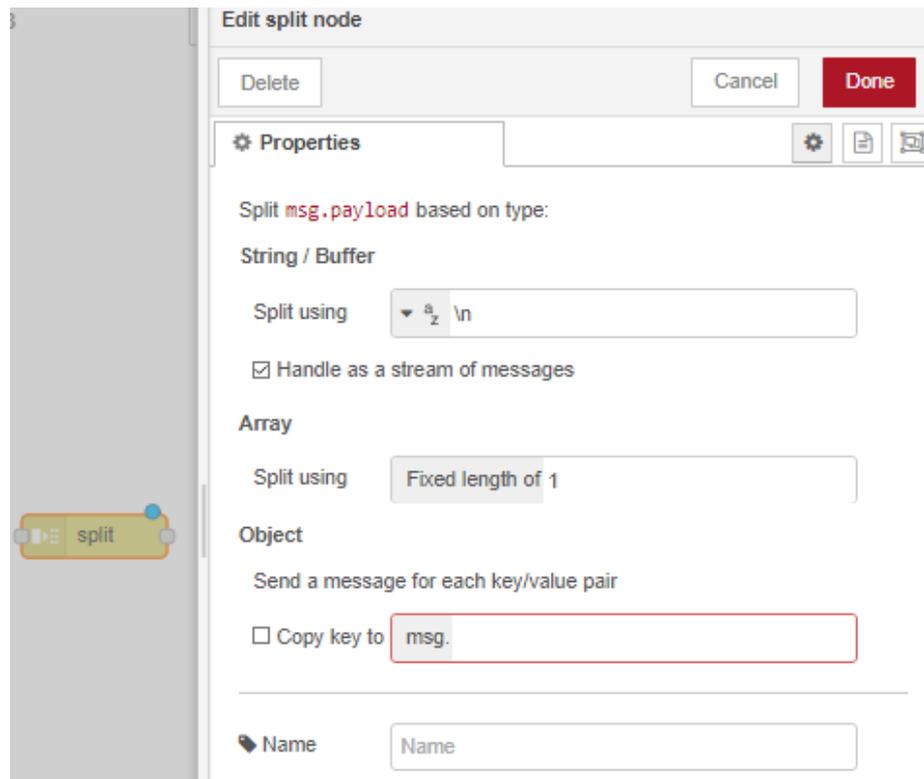


Figura 5.41 Nodo Split.

Imagen propia

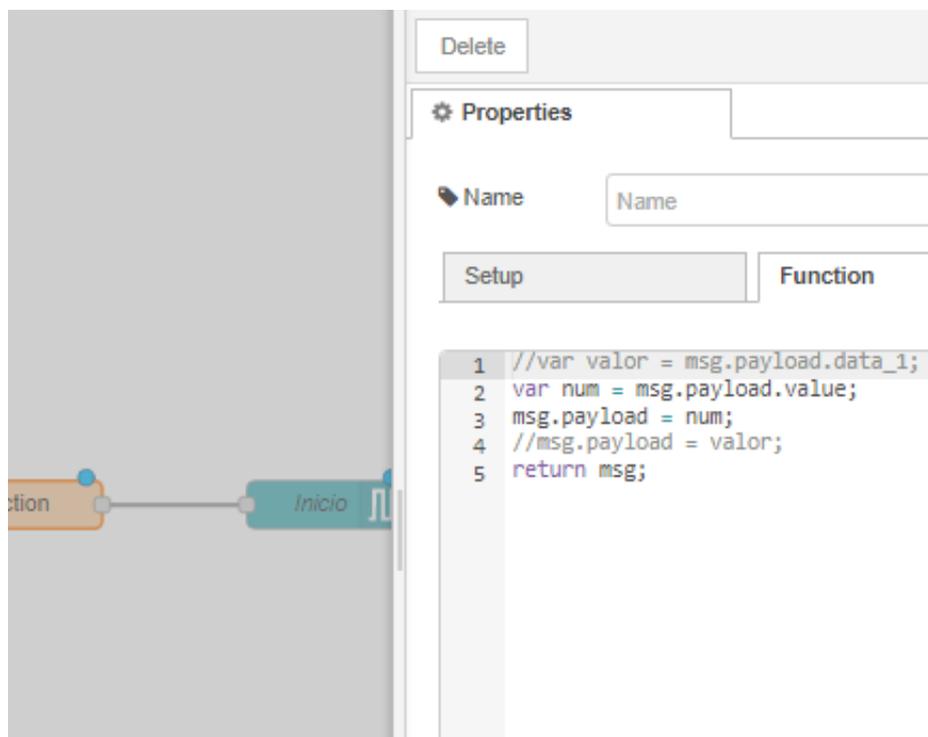


Figura 5.42 Obtención dato recortado.

Imagen propia

El nodo S7 in, es el bloque de comunicación hacia el PLC la configuración es similar al bloque “s7 out” donde se debe especificar a que variable dentro del autómata (PLC) se envía el dato obtenido; mirar figura 5.43.

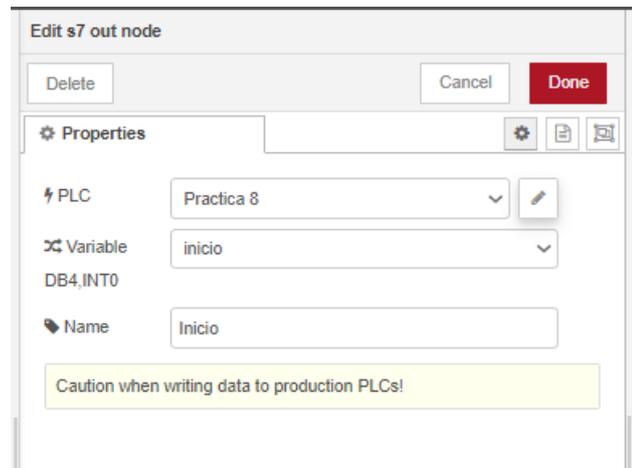


Figura 5.43 Envió de datos hacia el PLC.

Imagen propia

## 5.5 UBIDOTS.

Ubidots es la página web donde se podrá observar los datos obtenidos, y enviar datos hacia el PLC, la cuenta gratuita da acceso a las funciones principales. Acceder a su página oficial <https://ubidots.com> y crear una cuenta como en la figura 5.44.

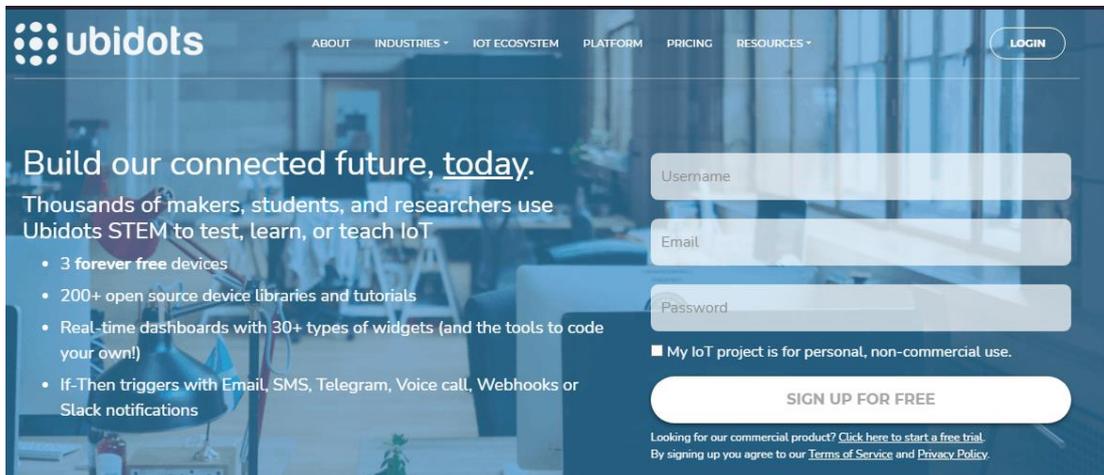


Figura 5.44 Creación cuenta Ubidots.

Imagen propia

### 5.5.1 OBTENCIÓN TOKEN.

La página web brinda un Token único para cada usuario, que permite el acceso a los datos almacenados en la plataforma, el token se localiza en la parte superior en el menú de usuario, “Credenciales API” donde se encuentra el API KEY y TOKENS, para el acceso desde Node red es necesario el Token, copiar y mantenerlo en un lugar seguro, mirar figura 5.45.

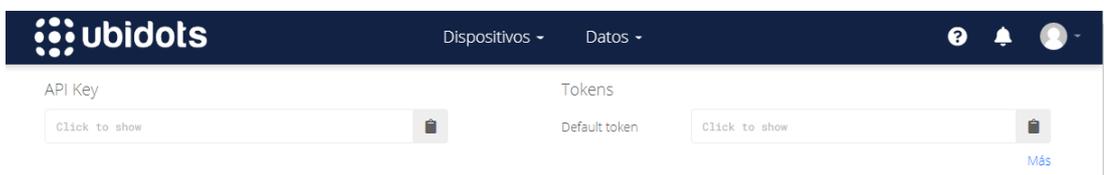


Figura 5.45 Obtención Token.

Imagen propia

## 5.5.2 CREACIÓN DE DISPOSITIVOS.

Los dispositivos son los encargados de almacenar las variables, la versión gratuita tiene un máximo de 3 dispositivos por cuenta; figura 5.46, para agregar un nuevo dispositivo realizar clic en el símbolo más, en la parte superior derecha. La figura 5.47 muestra las diferentes formas de agregar un nuevo dispositivo, la opción recomendada es seleccionar dispositivo en blanco y colocar un nombre.



Figura 5.46 Dispositivos.

Imagen propia



Figura 5.47 Creación nuevo dispositivo.

Imagen propia

### 5.5.3 CREACIÓN DE VARIABLES.

Existen dos formas de crear una variable, cruda y sintética; usar la variable cruda para lo cual dar clic en agregar variable y luego a cruda; figura 5.48.

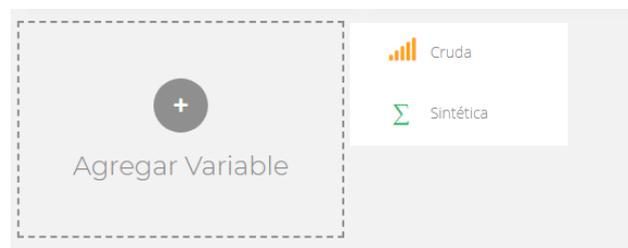


Figura 5.48 Creación de Variables.

Imagen propia

El nombre de la variable debe ser en minúsculas y sin espacios, caso contrario modificar el nombre de API label dentro de la variable, estas dos deben coincidir como lo indica la figura 5.49.

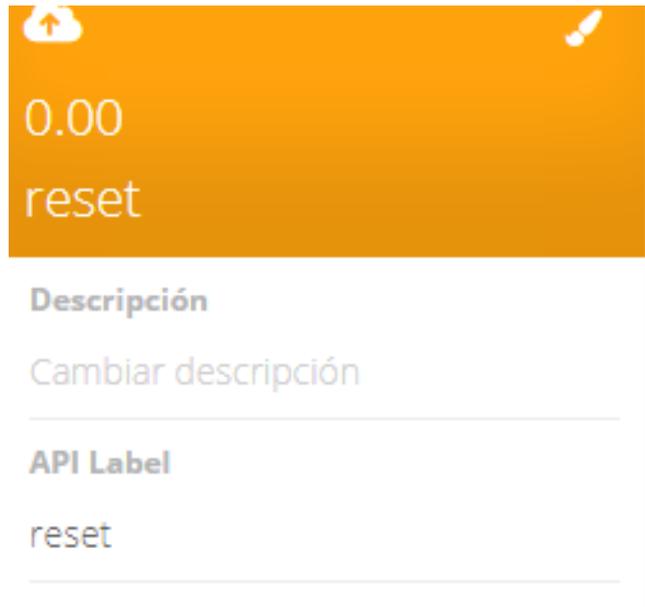


Figura 5.49 Nombre API label.

Imagen propia

#### 5.5.4 AGREGAR TABLERO.

Ubidots brinda un tablero que permite crear una interfaz y mostrar los datos almacenados, estos tienen un máximo de 3 por usuario, dentro del tablero crear Widget permite elegir la manera en la que se visualizara los datos obtenidos. Existen varios widgets, todos ellos necesitan una variable para mostrar mirar figura 5.50, estos se pueden agrupar en una sola ventana siempre que sean interruptor, deslizador, interruptores, etc.

Esto permite ahorrar espacio dentro del tablero y tener una mejor distribución del espacio como muestra la figura 5.51



Figura 5.50 Widget.

Imagen propia

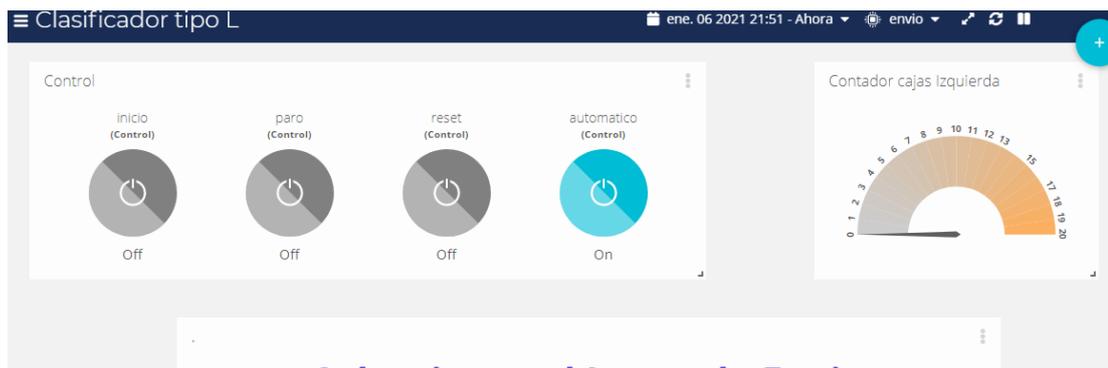


Figura 5.51 Tablero.

Imagen propia

Todos los Widget pueden modificar el rango de valores que muestra o envía, el medidor permite modificar el rango de valores a medir, el botón permite modificar los

valores enviados cuando se activa, como cuando se desactiva, estos valores se envían a través de Node-Red a el PLC.

## 5.6 VIDEOS GUÍA

Se desarrollo videos guía en aras del aprendizaje de los estudiantes, cada video explica de forma explícita la manera de desarrollar las practicas; para subir los videos tomamos en cuenta la plataforma vimeo, es una plataforma que permite almacenar los videos en la nube y generar un link específico para el video.

Tabla 5.1 Links videos guía.

<i>Practica #</i>	<i>Link de video.</i>
Practica 1	<a href="https://vimeo.com/514653138">https://vimeo.com/514653138</a>
Practica 2	<a href="https://vimeo.com/514653198">https://vimeo.com/514653198</a>
Practica 3	<a href="https://vimeo.com/516533560">https://vimeo.com/516533560</a>
Practica 4	<a href="https://vimeo.com/514653238">https://vimeo.com/514653238</a>
Practica 5	<a href="https://vimeo.com/516533288">https://vimeo.com/516533288</a>
Practica 6	<a href="https://vimeo.com/514653283">https://vimeo.com/514653283</a>
Practica 7	<a href="https://vimeo.com/514653337">https://vimeo.com/514653337</a>
Practica 8	<a href="https://vimeo.com/514653377">https://vimeo.com/514653377</a>

# CAPITULO 6: DESARROLLO DE APLICACIONES.

En este capítulo se explica el modo de funcionamiento de las aplicaciones que se han desarrollado para el presente trabajo de titulación. Se debe tener en cuenta que, en el apartado de anexos se puede encontrar una guía detallada de todas las prácticas.

Las aplicaciones de las guías prácticas se desarrollaron basados en las delimitaciones de la herramienta gráfica Factory I/O. La herramienta gráfica cuenta con escenas predeterminadas que simulan el ambiente y aplicación en distintos ámbitos. Para encontrar las escenas nos regimos a la sección “Escenas” como se visualiza en la Figura 6

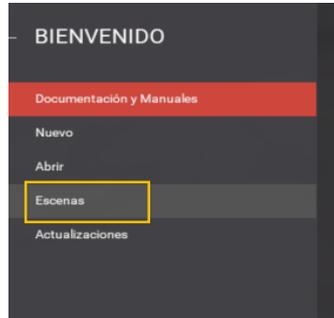


Figura 6 Escenas

Dentro de la cual encontraremos 17 escenas diseñadas con diferentes funciones mirar figura 6.1

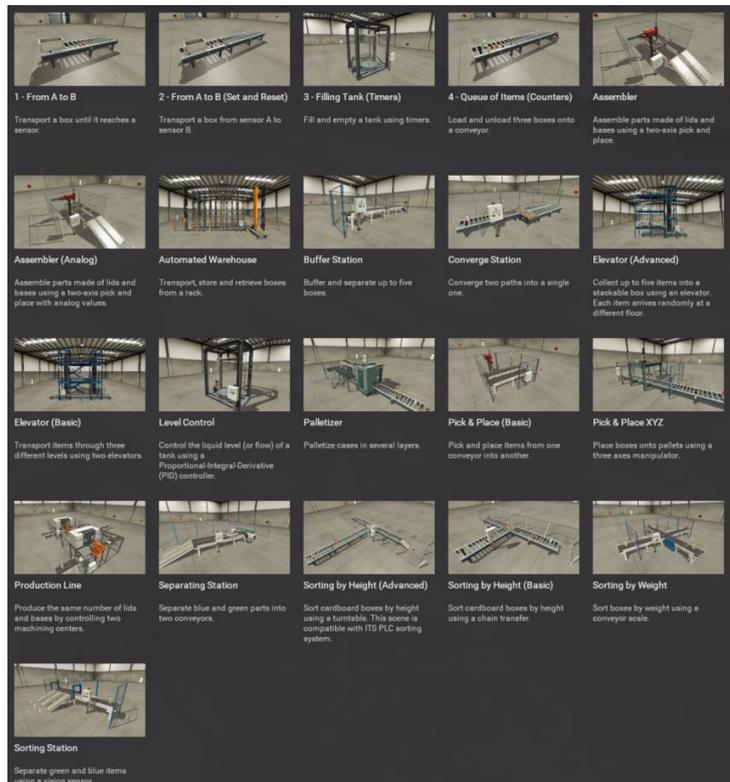


Figura 6.1 Escenas de la herramienta gráfica Factory IO

## 6.1 BANDA TRANSPORTADORA.

Para el desarrollo de la primera practica optamos por usar crear mediante las herramientas una banda transportadora que permite trasladar un objeto desde una posición A hacia una posición B. Además, se incluyo un sensor capacitivo para permitir contabilizar los objetos trasladados, la figura 6.3 muestra el entorno de trabajo en la herramienta Grafica.

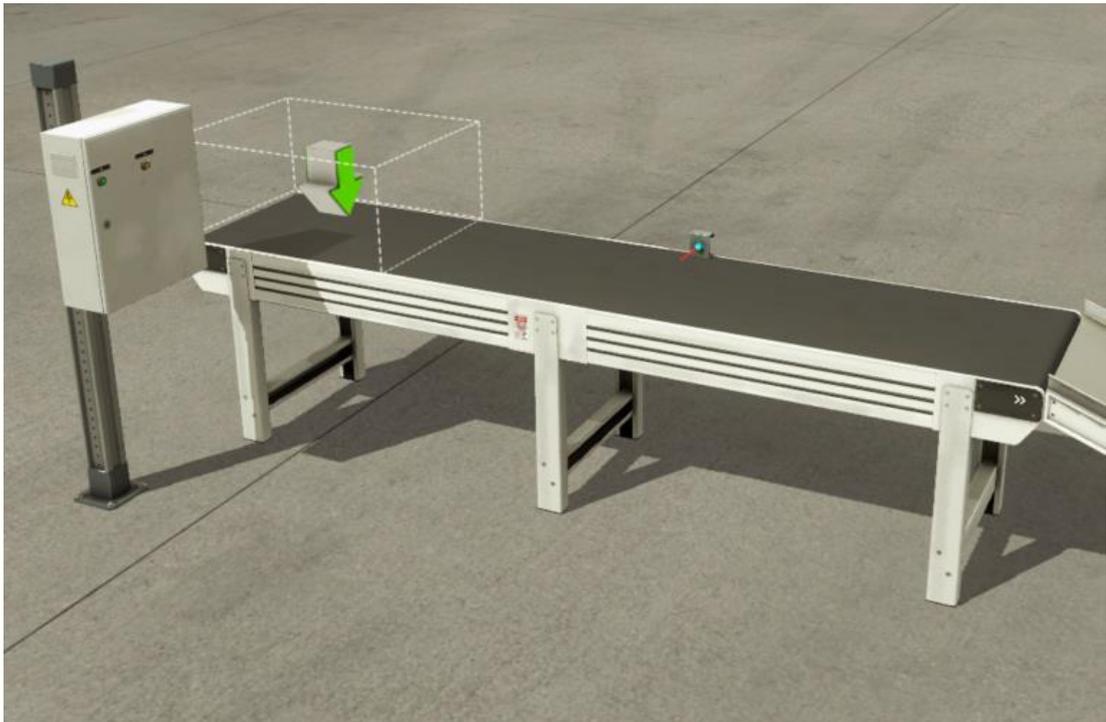


Figura 6.3 Entorno de trabajo en la herramienta Grafica Factory IO

La conexión con el autómata se realiza mediante la configuración dentro de la herramienta grafica en la sección **Drivers** ubicada en la pestaña archivo.

Un nuevo ambiente se visualiza, donde es necesario elegir el autómata a usar en la lista desplegable como se visualiza en la figura 6.4

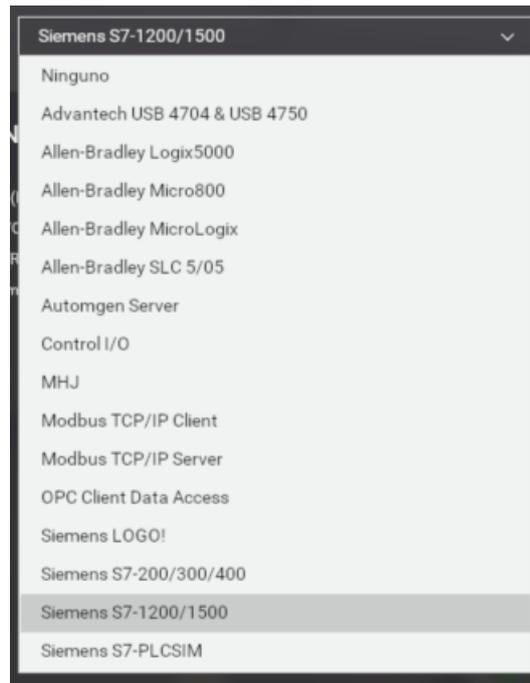


Figura 6.4 Selección del autómeta a usar.

En la sección “Configuración” llenar los datos necesarios para la comunicación como la dirección IP y el adaptador de red como se especifica en la figura 6.5



Figura 6.5 Configuración para la comunicación con el autómeta.

Para asignar las entradas y salidas basta con arrastrar las variables a sus respectivas entradas y salidas como se especifica en la Figura 6.6



Figura 6.6 Asignación de Variables.

Para comprobar la comunicación con el autómata se debe dar clic al botón **CONECTAR** este se encuentra ubicado en la parte sección superior derecha dentro de la configuración de la herramienta gráfica Factory I/O. Una vez generada la comunicación el programa nos enseñara con un símbolo de “Check“ de comunicación exitosa

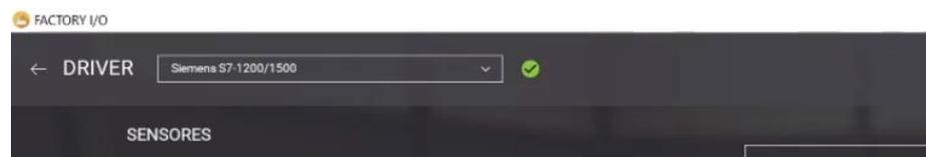


Figura 6.7 Comunicación Exitosa.

De esta aplicación se derivan la practica 1,2 y 3 en donde se busca ambientar al estudiante con las herramientas y la comunicación de estas, en las guías de prácticas 1,2 y 3 ubicada en los anexos se puede encontrar mayores detalles del procedimiento para desarrollar esta aplicación.

## 6.2 CLASIFICACIÓN DE OBJETOS

Las siguientes prácticas son enfocadas a la clasificación, en donde podrán encontrar la clasificación mediante color, peso, tamaño y control manual donde el usuario define a donde se clasifica cada objeto. Dentro de la herramienta gráfica Factory IO se encuentran varias escenas para la clasificación de objetos. Para las prácticas usaremos varias de esas escenas como “Sorting Station” que permite clasificar los objetos por color y contabilizar en cada estación cuantos objetos llegan. En las siguientes figuras se muestra las escenas a seleccionar en la herramienta gráfica.

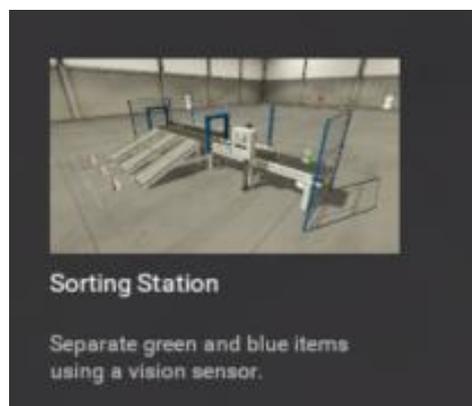


Figura 6.20 Sorting Station.

Sorting by Weight permite la clasificación del objeto por su peso.

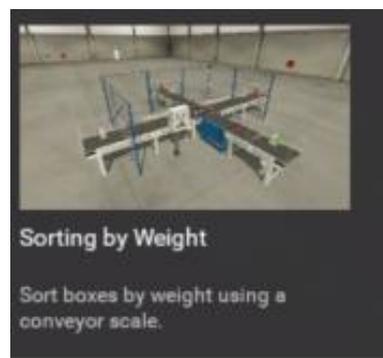


Figura 6.21 Sorting by Weight

De estas escenas se derivan las practicas 4 y 6 respectivamente.

En la figura 6.22 se muestra el diseño de la pagina web, en donde se tiene un botón para el reset en el contador.

En la figura 6.23 se visualiza el diseño de la pagina web en donde el usuario podrá definir y controlar el inicio, paro y reset de la aplicación, también controlar el destino de cada caja, invitamos a revisar las prácticas en la sección de Anexos. En las guías de prácticas 4 y 6 ubicada en los anexos se puede encontrar mayores detalles del procedimiento para desarrollar estas aplicaciones.

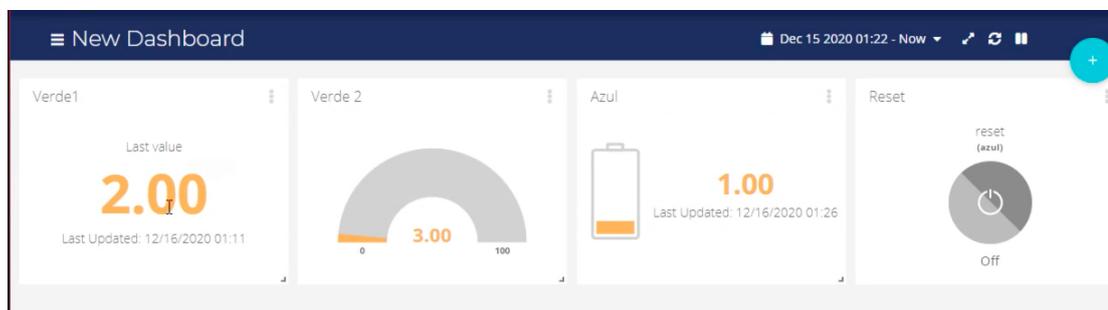


Figura 6.22 Diseño practica 4 página web.

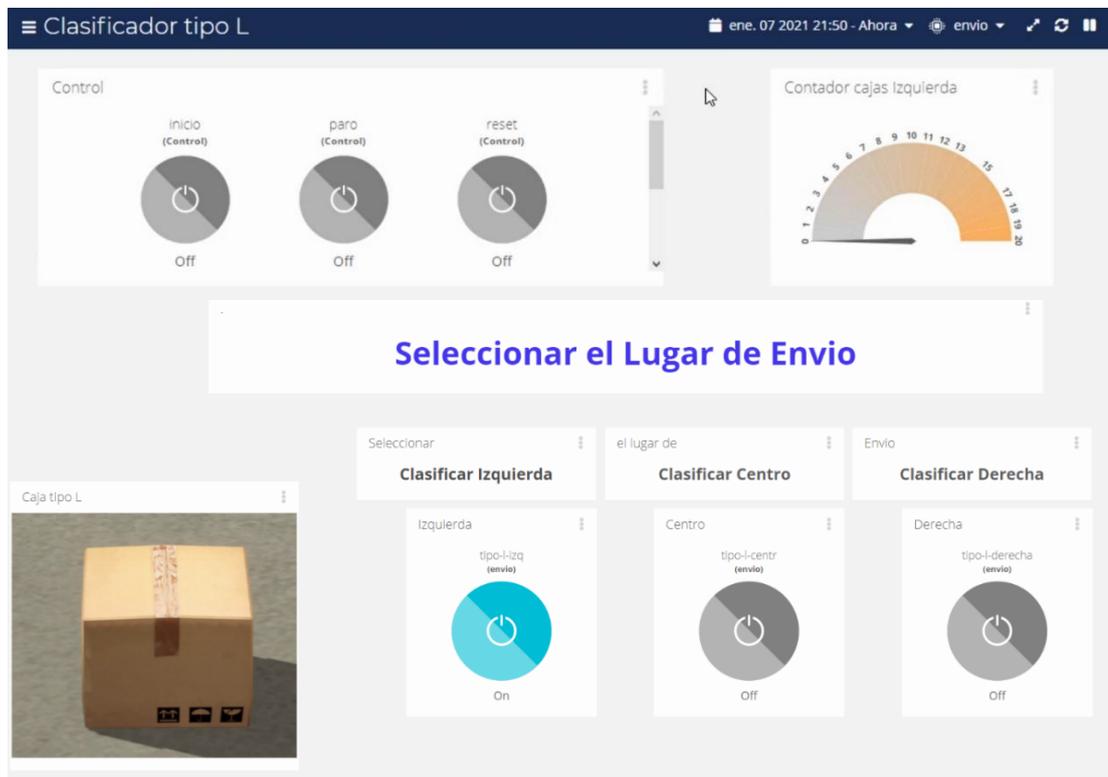


Figura 6.23 Diseño practica 6 página Web.

### 6.3 CLASIFICACIÓN MEDIANTE PICK UP

La práctica 5 fue diseñada para usar los conocimientos adquiridos y crear una escena desde cero lo que permitirá al estudiante introducirse más en la creación de espacios dentro de la herramienta gráfica. En la Figura 6.30 se muestra el espacio concluido para su funcionamiento.

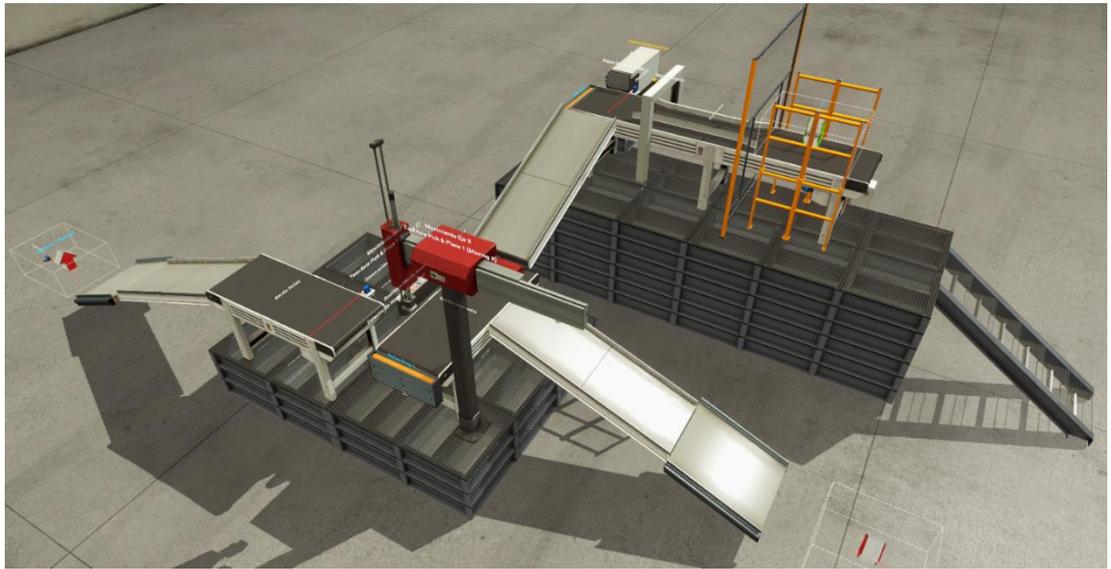


Figura 6.30 Espacio de Clasificación

Este proceso virtual está diseñado para clasificar según el color. El material ingresa por la banda transportadora principal, empuja al objeto para empezar su clasificación, donde lo espera un pick up para poder trasladar el objeto hacia una banda transportadora externa que permite su almacenamiento. En la figura 6.31 se observa el control mediante la página web.

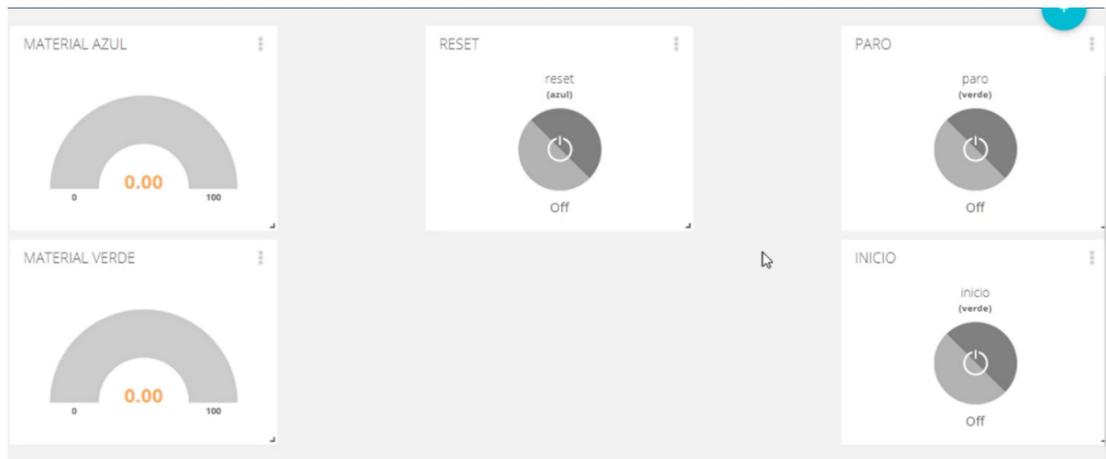


Figura 6.31 Control pick up página web.

En la figura 6.31 se muestra la página web en donde el usuario podrá visualizar el contador de cada caja, el inicio/paro y reset del programa. En la guía de práctica 5 ubicada en los anexos se puede encontrar mayores detalles del procedimiento para desarrollar esta aplicación.

#### **6.4 TRASLADO Y CLASIFICACIÓN DE OBJETOS**

Para la práctica 7 se diseñó un espacio de simulación para el traslado y clasificación de objetos según su color, además el usuario puede modificar la velocidad de la banda transportadora, esto permitirá al estudiante aprender a manejar valores análogos entre las herramientas de simulación. En la figura 6.40 se visualiza el espacio creado en la herramienta Factory IO

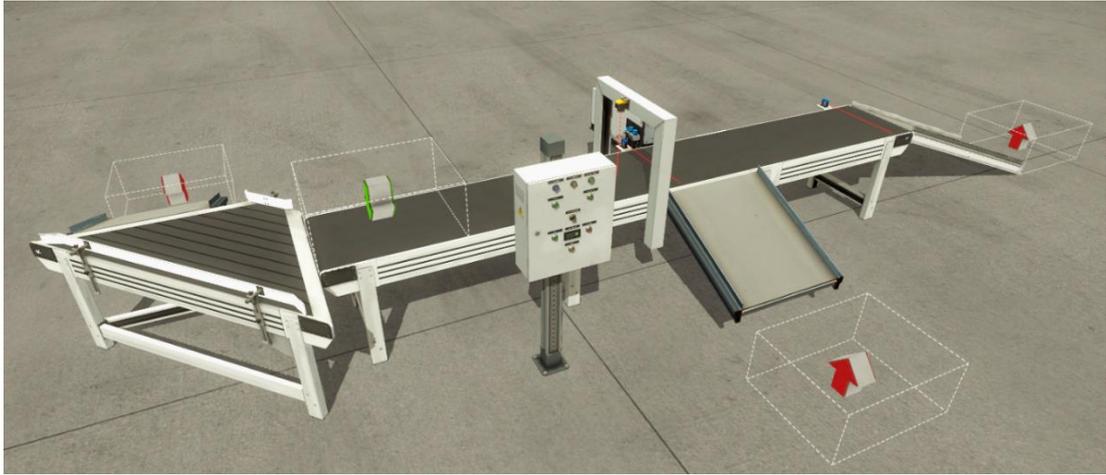


Figura 6.40 Clasificador de objetos por su color.

Esta aplicación permite al usuario modificar el objeto a clasificar desde Factory IO o desde la página web logrando el control y comunicación de estas. Véase la figura 6.41 para observar el control mediante la página web. En la figura 6.41 el control de la aplicación mediante la pagina web, este contiene el control para inicio, paro y la opción de definir el material a retirar, en la guía de práctica 7 ubicada en los anexos se puede encontrar mayores detalles del procedimiento para desarrollar esta aplicación.



Figura 6.41 Control página Web.

## 6.5 CONTROLADOR PID

En esta práctica se diseña un controlador PID usando una escena predeterminada. En la herramienta Factory IO se la encuentra como “Level Control” como se especifica en la figura 6.50



Figura 6.50 Level Control.

En la figura 6.51 se muestra el entorno creado, mediante el cual procedemos a controlar el nivel del agua en un tanque con PID. Este proceso está diseñado para poder controlar el flujo de salida en el agua a través de la herramienta gráfica o desde la página Web como se especifica en la figura 6.52



Figura 6.51 Entorno para el control de nivel.

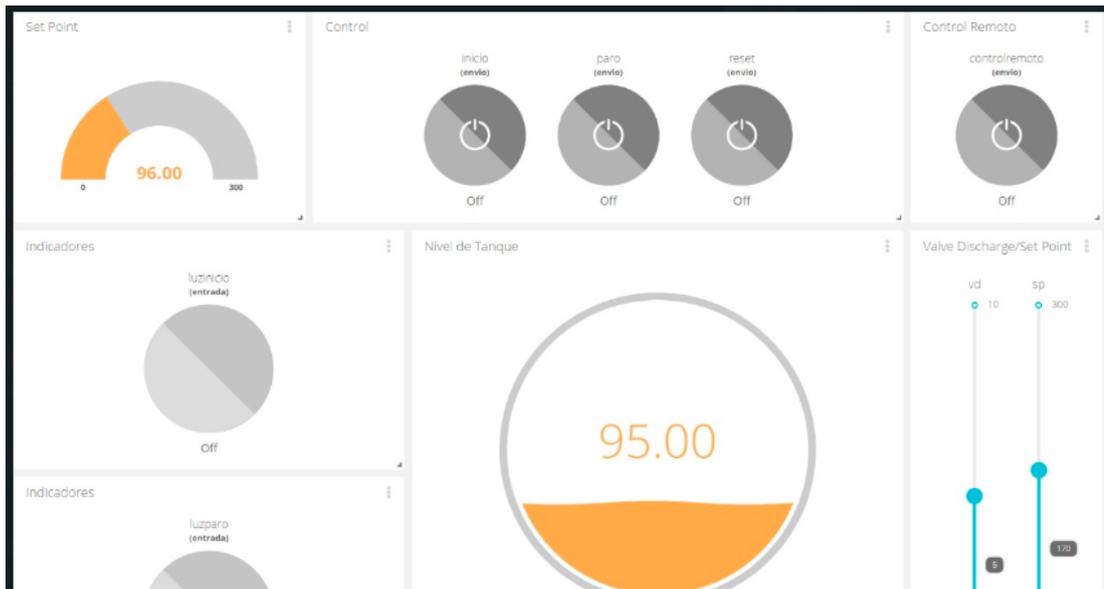


Figura 6.52 Pagina Web para el control y monitoreo del sistema de flujo

En la figura 6.52 se visualiza el diseño de la página web en donde el usuario podrá controlar el inicio, paro y reset. Así también podrá controlar el set point y la válvula de desfogue para crear perturbaciones dentro del sistema, en la guía de práctica 8 ubicada en los anexos se puede encontrar mayores detalles del procedimiento para desarrollar esta aplicación.

# **CAPITULO 7: CONCLUSIONES Y RECOMENDACIONES**

## **7.1 CONCLUSIONES.**

Para la elaboración de las guías prácticas fue impredecible entender el funcionamiento y su respectiva conectividad de las plataformas Factory I/O, Node-Red y la página web Ubidots permitiendo interactuar en tiempo real con el proceso y usuario.

La comunicación entre Factory I/O y el PLC SIM Advance, muestra una respuesta rápida, eficaz y cómoda al momento de llevar a cabo una simulación tomando en cuenta la configuración previa de ambas plataformas.

En este documento se encuentra plasmado, la manera en la cual los procesos pueden desarrollarse mediante “IoT”, fue diseñado para los estudiantes en aras de enseñar y brindar apoyo a la carrera de Ingeniería electrónica en automatización.

Los videos guía fueron desarrollados en aras de apoyar al estudiante con sus dudas, teniendo en cuenta que el internet tiene un gran impacto en la educación tradicional, muchos estudiantes buscan videos explicativos de temas que aún no comprenden, esa fue la razón principal del desarrollo de los videos guía.

Con este trabajo de titulación, podemos brindar la información y la motivación de continuar con la investigación sobre “IoT”, la idea de poder llevar el control de una industria ubicada al otro lado del mundo gracias al internet hoy es posible y cada día existen nuevos campos en los cuales se puede incursionar.

Con este documento esperamos que los estudiantes obtengan la suficiente información y práctica para poder usarlo en diferentes campos de la tecnología.

## 7.2 RECOMENDACIONES.

- 1 Factory I/O es una herramienta grafica que necesita configuración de IP, si se trabaja con un autómata, el botón de interacción “paro” dentro de la herramienta viene predeterminada en estado cerrado esto se debe tener en cuenta al momento de crear la programación para el autómata.
- 2 Factory I/O tiene la posibilidad de forzar los estados de cada sensor o actuador, brindando la posibilidad de corregir errores dentro de la programación.
- 3 Factory I/O tiene plantilla predeterminadas, donde el usuario podrá ahorrar tiempo al momento de diseñar el ambiente en 3D.
- 4 El emisor dentro de la herramienta grafica se puede modificar, brindando la posibilidad de modificar el material a trabajar, existen cajas, pallets, cajas de plástico, material de color azul y verde con diferentes formas; permite modificar el lugar y posición en la que aparece el material seleccionado.
- 5 Cada dispositivo dentro de Factory I/O se puede modificar, especialmente las bandas y sensores; los cuales permiten trabajar con valores Analógicos y Digitales.
- 6 Para poder trabajar con un autómata virtual, dentro de la herramienta Tia portal se debe modificar las propiedades del proyecto, lo que permitirá simular y reconocer al autómata virtual como uno real.
- 7 Tia portal tiene varios modelos de autómatas para trabajar, en este documento se optó por usar el s7-1500.
- 8 Se recomienda usar segmentos destinados para cada proceso para obtener mayor orden dentro de la programación.
- 9 Node-Red, tiene varios nodos que son desarrollados por una comunidad bastante amplia permitiendo a los usuarios usar Node red para diferentes fines.
- 10 Node red permite la programación mediante bloques conectados por líneas, se recomienda ordenar de la mejor manera y brindar un nombre distintivo a cada bloque.
- 11 Node-Red permite deshabilitar pestañas o también llamados “Flow” lo que permite poder trabajar en más de un proyecto a la vez, mientras que el momento que se publique la programación del Flow deshabilitado no se publica.

- 12 Ubidots, proporciona una cuenta gratuita de prueba durante 30 días, transcurrido ese tiempo elimina acciones para el usuario, si es necesario crear varias cuentas falsas para poder trabajar con varios proyectos.
  
- 13 Es importante conocer las limitaciones que tiene Ubidots como, por ejemplo: permite únicamente 3 tablero para crear interfaz, permite hasta 15 variables separadas dentro de cada tablero, por tal motivo se recomienda agrupar variables a mostrar; el nombre brindado a la variable debe coincidir en mayúsculas o minúsculas al API label.
  
- 14 Cada Widget tiene diferentes funciones que pueden ser modificadas por el usuario, como la escala a mostrar como los colores de cada widget, este último requiere un nivel avanzado de programación.
  
- 15 Ubidots, crea Token único para cada cuenta, recomendación guardar el token en un lugar seguro para evitar que otros usuarios puedan ingresar a modificar datos.
  
- 16 Los tableros de Ubidots, pueden compartirse permitiendo interactuar con la página web desde cualquier lado con el enlace que brinda Ubidots.

# REFERENCIAS BIBLIOGRÁFICAS

1. Quiñones-Cuenca, M., González-Jaramillo, V., Torres, R. and Jumbo, M., 2020. Sistema De Monitoreo De Variables Medioambientales Usando Una Red De Sensores Inalámbricos Y Plataformas De Internet De Las Cosas. [online] Available at: <[http://scielo.senescyt.gob.ec/scielo.php?script=sci\\_arttext&pid=S1390-65422017000100329](http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-65422017000100329)>.
2. Dol, S., & Bhing, R. (2018). SMART motor for industry 4.0. Artículo científico. doi:10.1109/ETECHNXT.2018.8385291.
3. Fernández, T., & Fraga Lamas, P. (2018). A Review on Human-Centered IoT Connected Smart Labels for the Industry 4.0. Artículo científico. doi:10.1109/ACCESS.2021.2833501.
4. Dol, S., & Bhing, R. (2018). SMART motor for industry 4.0. Artículo científico. doi:10.1109/ETECHNXT.2018.8385291.
5. In Lee and Kyoochun Lee (2015) "The Internet of Things (IoT): Applications, investments, and challenges for enterprises, Business Horizons, 58, 431-440.
6. E.Abasolo, A.Carrera, X.Gordillo, G.Romero, S., 2021. Evaluación del modelo de referencia de "Internet of Things" (IoT), mediante la implantación de arquitecturas basadas en plataformas comerciales, open hardware y conectividad IPv6. [online] Repositorio.espe.edu.ec. Available at: <<http://repositorio.espe.edu.ec/bitstream/21000/7648/1/AC-ET-ESPE-047621.pdf>> [Accessed 4 February 2021].
7. Rubio.Maria, Calderon, V., 2021. DISEÑO E IMPLEMENTACION PARA UN SERVIDOR IOT DE BAJO COSTO UTILIZANDO STM32 PARA EL MONITOREO DE CONSUMO ELÉCTRICO RESIDENCIAL. [online] Dspace.ups.edu.ec.Availableat<<https://dspace.ups.edu.ec/bitstream/123456789/19485/1/UPS-GT003036.pdf>> [Accessed 4 February 2021].
8. BANKINTER, F., 2021. [online] Boletines.prisadigital.com. Available at: <[http://boletines.prisadigital.com/El\\_internet\\_de\\_las\\_cosas.pdf](http://boletines.prisadigital.com/El_internet_de_las_cosas.pdf)> [Accessed 4 February 2021].
9. MICHILENA CALDERÓN, J., 2021. "DISEÑO DE UN SISTEMA DE MONITOREO DE INCENDIOS FORESTALES EN TIEMPO REAL BASADO EN INTERNET DE LAS COSAS Y PLATAFORMAS COMO SERVICIO EN LA NUBE". [online] Repositorio.puce.edu.ec. Available at: <<http://repositorio.puce.edu.ec/bitstream/handle/22000/11307/JAIME%20MICHILENA%20Caso%20de%20Estudio%20Unidad%20de%20Titulaci%C3%B3n.pdf?sequence=1&isAllowed=y>> [Accessed 5 February 2021].
10. Mobiliza Academy. 2021. ¿Qué es la Industria 4.0 y cómo va a afectar a las empresas[online]Availablal<[https://mobilizaacademy.com/industria40\\_como\\_afecta\\_a\\_las\\_empresas/](https://mobilizaacademy.com/industria40_como_afecta_a_las_empresas/)> [Accessed 5 February 2021].
11. Icl-didactica.com. 2021. Factory I/O – ICL DIDÁCTICA SAS. [online] Available at: <<https://www.icl-didactica.com/factory-i-o/>> [Accessed 5 February 2021].
12. FACTORY I / O. (2018). Recuperado 29 de agosto de 2018, de <https://factoryio.com/docs/>

13. *Introducción a Ubidots.* (s. f.). Ubidots página oficial. [https://ubidots.com/docs/es/get\\_started/introduccion](https://ubidots.com/docs/es/get_started/introduccion).
14. AGUILAR MACANCHÍ, PACALLA GUAMÁN, P., 2021. “DISEÑO DE UN SISTEMA DE ADQUISICIÓN, TRANSMISIÓN Y ANÁLISIS DE DATOS VEHICULARES, EN BASE A SENSORES, PARA LA GESTIÓN Y SEGURIDAD DEL TRÁFICO. [online] Dspace.espol.edu.ec. Available at: <<https://www.dspace.espol.edu.ec/retrieve/130579/D-CD108731.pdf>> [Accessed 5 February 2021].
15. Toquica Ramírez, Guzmán Ruiz, L., 2021. *Desarrollo de Sistema de Seguridad para Automóviles con IOT y Smartphone.* [online] Repository.udistrital.edu.co. Available at <[https:// repository .udistrital.edu.co /bitstream/handle /11349/2750 /Guzm%C3%A1nRuizMichaelLeandro2016.pdf?sequence=1&isAllowed=y](https://repository.udistrital.edu.co/bitstream/handle/11349/2750/Guzm%C3%A1nRuizMichaelLeandro2016.pdf?sequence=1&isAllowed=y)> [Accessed 5 February 2021].
16. Odr.chalmers.se. 2021. *Securing IoT Apps in Node-RED.* [online] Available at: <[https://odr.chalmers.se/bitstream/20.500.12380/300759/1/CSE%2020-06%20Olsson %20ODR.pdf](https://odr.chalmers.se/bitstream/20.500.12380/300759/1/CSE%2020-06%20Olsson%20ODR.pdf)> [Accessed 5 February 2021].
17. Flows.nodered.org. 2021. *btsimonh's node-opencv motion detection (2017-11-02) (flow) - Node-RED.* [online] Available at: <<https://flows.nodered.org/flow/33a93ac5418009993d38c00009ef453e>> [Accessed 5 February 2021].
18. Noderedguide.com. 2021. *Node RED Programming Guide – Programming the IoT.* [online] Available at: <<http://noderedguide.com/>> [Accessed 5 February 2021].
19. G. Gardasevic, H. Fotouhi, I. Tomasic, M. Vahabi, M. Bjorkman, M. Linden, “A Heterogeneous IoT-based Architecture for Remote Monitoring of Physiological and Environmental Parameters,” 4th EAI International Conference on IoT Technologies for HealthCare, Angers, France, Oct. 24-25, 2017.
20. Node.RED. *Design: Flow file format v2.* <https://github.com/node-red/node-red/wiki/Design:-Flow-file-format-v2>, 2015. [Online; accessed 7- January - 2020].
21. Node.RED. *btsimonh's node-opencv motion detection (2017-11-02).* <https://flows.nodered.org/flow/33a93ac5418009993d38c00009ef453e>, 2020. [Online; accessed 7-January-2020].
22. Node.RED. *Packaging.* [https://nodered.org/docs/creating-nodes/ packaging](https://nodered.org/docs/creating-nodes/packaging), 2020. [Online; accessed 7-January-2020].
23. Nodered.org. 2021. *Writing Functions: Node-RED.* [online] Available at: <<https://nodered.org/docs/user-guide/writing-functions>> [Accessed 5 February 2021].
24. Nodered.org. 2021. *The Core Nodes: Node-RED.* [online] Available at <<https://nodered.org/docs/user-guide/nodes>> [Accessed 5 February 2021].
25. *Aprendiendo Arduino.* 2021. *Funciones en Node-RED.* [online] Available at <<https://aprendiendoarduino.wordpress.com/2020/12/02/funciones-en-node-red>> [Accessed 5 February 2021].

26. Duc Truong Pham, University of Birmingham, Birmingham, UK. (2018). *Industry 4.0: Managing the Digital Transformation*. (Series ed.) [Libro electrónico]. Springer Publishing. <http://www.springer.com/series/7113>
27. Gilchrist, A. (2016). *Industry 4.0 (1.a ed.)* [Libro electrónico]. Apress. <https://www.apress.com/9781484220467>
28. J. (s. f.). *Arquitecturas IoT. Aprendiendo Arduino*. <https://aprendiendoarduino.wordpress.com/2018/11/11/arquitecturas-iot/>
29. Wang S, Wan J, Zhang D, Li D, Zhang C (2016) *Hacia una fábrica inteligente para la Industria 4.0: a Sistema autoorganizado de múltiples agentes con retroalimentación y coordinación basadas en big data*.
30. Jazdi N (2014) *Sistemas físicos cibernéticos en el contexto de la Industria 4.0. 2014 IEEE internacional conferencia sobre automatización, calidad y pruebas, robótica*. doi: 10.1109/AQTR.2014.6857843.
31. Ubidots Docs. (s. f.). *UBIDOTS*. Recuperado 18 de febrero de 2021, de <https://ubidots.com/docs/sw/>
32. *SISTEMA DE MONITOREO, TELEMETRIA Y TELECONTROL*". (2018, febrero). Iraam Antonio López Salas [https://infotec.repositorioinstitucional.mx/jspui/bitstream/1027/252/3/INFOTEC\\_MSE\\_IALS\\_26092019.pdf](https://infotec.repositorioinstitucional.mx/jspui/bitstream/1027/252/3/INFOTEC_MSE_IALS_26092019.pdf)

# APÉNDICE

Anexo 1