



Desarrollo de una prueba de concepto funcional aplicable al Internet Industrial de las Cosas (IIoT) utilizando Sistemas Embebidos de National Instruments, lenguaje de nivel de abstracción medio y una plataforma de servicios en la nube para proyección de aplicabilidad tanto académica como profesional en Datalights Distribuidor Autorizado de National Instruments.

Ortega Vintimilla, Hugo Gabriel y Vásquez del Hierro, Nelson André

Departamento de Eléctrica y Electrónica

Carrera de Ingeniería Electrónica e Instrumentación

Trabajo de titulación, previo a la obtención del Título de Ingeniero en Electrónica e Instrumentación

Ing. Naranjo Hidalgo, Cesar Alfredo

31 de agosto de 2020



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“Desarrollo de una prueba de concepto funcional aplicable al Internet Industrial de las Cosas (IIoT) utilizando Sistemas Embebidos de National Instruments, lenguaje de nivel de abstracción medio y una plataforma de servicios en la nube para proyección de aplicabilidad tanto académica como profesional en Datalights Distribuidor Autorizado de National Instruments.”** fue realizado por los señores **Ortega Vintimilla, Hugo Gabriel** y **Vásquez Del Hierro, Nelson André**, el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 31 de agosto de 2020

Naranjo Hidalgo, César Alfredo

C. C.: 0501498505

INFORME URKUND

**Document Information**

Analyzed document	ORTEGA_VINTIMILLA_HUGO_GABRIEL-VASQUEZ_DELHIERRO_NELSON_ANDRE-CORREGIDO (1).docx (D78414441)
Submitted	8/31/2020 6:06:00 PM
Submitted by	Eddie Galarza
Submitter email	eegalarza@espe.edu.ec
Similarity	1%
Analysis address	eegalarza.espe@analysis.arkund.com

Entire Document

CARATULA Desarrollo de una prueba de concepto funcional aplicable al Internet Industrial de las Cosas (IIoT) utilizando Sistemas Embebidos de National Instruments, lenguaje de nivel de abstracción medio y una plataforma de servicios en la nube para proyección de aplicabilidad tanto académica como profesional en Datalights Distribuidor Autorizado de National Instruments.

Ortega Vintimilla, Hugo Gabriel y Vásquez del Hierro, Nelson André

Departamento de Eléctrica y Electrónica

Carrera de Ingeniería Electrónica e Instrumentación

Trabajo de titulación, previo a la obtención del Título de Ingeniero en Electrónica e Instrumentación

Ing. Naranjo Hidalgo, Cesar Alfredo

31 de agosto de 2020

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN Certifico que el trabajo de titulación, "Desarrollo de una prueba de concepto funcional aplicable al

Naranjo Hidalgo, César Alfredo

C. C.: 0501498505



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Ortega Vintimilla, Hugo Gabriel**, con cédula ciudadanía n°1804656708, y **Vásquez Del Hierro, Nelson André**, con cédula ciudadanía n°0503498719, declaramos que el contenido, ideas y criterios del trabajo de titulación:

Desarrollo de una prueba de concepto funcional aplicable al Internet Industrial de las Cosas (IIoT) utilizando Sistemas Embebidos de National Instruments, lenguaje de nivel de abstracción medio y una plataforma de servicios en la nube para proyección de aplicabilidad tanto académica como profesional en Datalights Distribuidor Autorizado de National Instruments es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 31 de agosto de 2020

Ortega Vintimilla, Hugo Gabriel

C.C.: 1804656708

Vásquez Del Hierro, Nelson André

C.C.: 0503498719



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Ortega Vintimilla, Hugo Gabriel**, con cédula ciudadanía n°1804656708, y **Vasquez Del Hierro, Nelson Andre**, con cédula ciudadanía n°0503498719, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Desarrollo de una prueba de concepto funcional aplicable al Internet Industrial de las Cosas (IIoT) utilizando Sistemas Embebidos de National Instruments, lenguaje de nivel de abstracción medio y una plataforma de servicios en la nube para proyección de aplicabilidad tanto académica como profesional en Datalights Distribuidor Autorizado de National Instruments** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, 31 de agosto de 2020

Ortega Vintimilla, Hugo Gabriel

C.C.: 1804656708

Vásquez Del Hierro, Nelson André

C.C.: 0503498719

DEDICATORIA

Dedico este trabajo de finalización de mi etapa universitaria a toda mi familia en especial a mis padres Víctor Hugo Ortega y Bertha Vintimilla, quienes me apoyaron en todo momento durante esta grata etapa de mi vida, a mis hermanos Cinthya y Nicholas que siempre me alentaron a dar lo mejor de mí y que nunca dejaron que me rinda, esto es para ustedes y por ustedes.

Hugo Gabriel Ortega Vintimilla

DEDICATORIA

Dedico este trabajo a mi mami Grace por todo el apoyo, la confianza, los consejos y el amor incondicional que recibo de ella cada día de mi vida. A mi padre Eduardo quien es mi fortaleza, nunca ha permitido que me dé por vencido y me motiva a seguir adelante para cumplir con cada una de mis metas propuestas. A mi hermana Karen con quien he compartido mis alegrías y tristezas, y la persona que sin duda es mi ejemplo a seguir.

Les amo con todo mi corazón.

Nelson André Vásquez Del Hierro

AGRADECIMIENTO

Agradezco a mis amigos André y Christian, maestros y compañeros que de cierta manera aportaron para culminar mis estudios universitarios, también a mí novia Jessica que siempre estuvo apoyándome cuando lo necesite, de manera especial al Ingeniero César Naranjo que apoyo al desarrollo de este trabajo de investigación desde su inicio, a la empresa Dataligths y al Ing. Santiago Orellana que abrió las puertas de la empresa para que podamos desarrollar este trabajo prestando sus instalaciones y equipos.

Hugo Gabriel Ortega Vintimilla

AGRADECIMIENTO

Agradezco a Dios por todas sus bendiciones recibidas. A mis amados padres por el apoyo incondicional a lo largo de estos años, por su confianza y por todo lo que han hecho por mí, gracias a ellos he llegado hasta aquí y estoy agradecido infinitamente por tenerles en mi vida. A mis amigos Hugo y Christian que han compartido conmigo gran parte de mi vida universitaria y con los cuales he compartido triunfos y derrotas. A mi abuelita Silvia que siempre ha estado pendiente de mí brindándome todo su cariño. Al Ingeniero César Naranjo por su guía y orientación para culminar con este trabajo.

Nelson André Vásquez Del Hierro

ÍNDICE DE COTENIDOS

CARÁTULA	1
CERTIFICACIÓN	2
RESPONSABILIDAD DE AUTORÍA	4
AUTORIZACIÓN DE PUBLICACIÓN	5
DEDICATORIA	6
DEDICATORIA	7
AGRADECIMIENTO	8
AGRADECIMIENTO	9
ÍNDICE DE COTENIDOS	10
ÍNDICE DE FIGURAS	14
ÍNDICE DE TABLAS	17
RESUMEN	18
ABSTRACT	19
CAPÍTULO I	20
1. INTRODUCCIÓN	20
1.1 Planteamiento del problema	20
1.2 Antecedentes	20
1.3 Justificación e importancia	21
1.4 Objetivos	22
1.4.1 Objetivo general	22
1.4.2 Objetivos específicos	22
1.5 Variables de la investigación	23
1.5.1 Variable independiente	23
1.5.2 Variable dependiente	23
1.6 Hipótesis	23
CAPITULO II	24
2. MARCO TEÓRICO	24
2.1 Internet de las cosas (IoT)	24
2.1.1 Arquitectura IoT	27
2.2 Industria 4.0	30
2.3 Internet Industrial de las Cosas (IIoT)	32
2.3.1 Características de la IIoT	32

2.4	Diferencias entre IoT e IIoT	33
2.5	Paradigmas de la comunicación.....	35
2.5.1	Modelo Cliente / Servidor.....	35
2.5.2	Modelo Publicación / Suscripción.....	36
2.6	Protocolos de comunicación IIoT.....	39
2.6.1	Nivel de aplicación.....	39
2.6.2	Nivel de transporte.....	42
2.6.3	Nivel de red	43
2.6.4	Nivel físico	44
2.7	Labview.....	45
2.7.1	Instrumentos virtuales (VI)	46
2.7.2	Metodología de programación en <i>Labview</i>	47
2.7.3	NI MYRIO 1900	48
2.7.4	EPC – Entrenador de Planta de Control.....	49
2.8	Plataformas de desarrollo en la nube	51
2.8.1	Amazon AWS IoT	51
2.8.2	IBM Watson IoT para Bluemix	51
2.8.3	PTC ThingWorx	52
2.8.4	Microsoft Azure IoT Suite.....	52
2.8.5	Ubidots	53
2.9	Wireshark.....	55
2.9.1	Características.....	56
CAPÍTULO III.....		57
3.	DESARROLLO E IMPLEMENTACIÓN	57
3.1	Creación de una Cuenta en la Plataforma Ubidots.....	58
3.1.1	Visualización de Api Keys y Tokens de la cuenta	59
3.2	Creación de un Dispositivo.....	60
3.3	Creación de Variables en Ubidots	61
3.4	Creación de Dashboards y asignación de variables a los Widgets.....	65
3.4.1	Creación Dashboards	65
3.4.2	Asignación de Variables a Widgets.....	67
3.5	Desarrollo del código de conexión LabVIEW - Ubidots.....	71
3.5.1	Abrir y Cerrar conexión.	73
3.5.2	Credenciales de Escritura y Lectura.	74

3.5.3	Conversión de variables numéricas, booleanas y Strings.	75
3.5.4	Escritura de los datos (Write_Data.vi)	77
3.5.5	Lectura de variables booleanas, numéricas y strings.	77
3.6	Programa PING desde PC a la Plataforma Ubidots.....	78
3.7	Programa que Escribe Datos a la Plataforma de Ubidots	81
3.8	Programa que Recibe Datos desde la Plataforma de Ubidots	85
3.9	Eventos dentro de Ubidots	88
3.10	Creación de Reportes en Ubidots.....	94
3.11	Programa Control Motor DC con Lectura y Escritura de Variables desde NI myRIO en conjunto con planta EPC a Ubidots.	98
3.11.1	Configuración del Sistema Embebido NI myRIO.	98
3.11.2	Programa para comprobar la conexión a la plataforma Ubidots desde NI myRIO.....	102
3.11.3	Programa control Velocidad Motor DC de la planta EPC conexión con Ubidots.....	104
3.12	Interactuar con Dashboards desde un web browser y desde un Smartphone.....	114
3.12.1	Crear Organización.	114
3.12.2	Creación Usuario.....	119
3.12.3	Web Browser.....	123
3.12.4	Smartphone.....	124
CAPÍTULO IV	126
4. PRUEBAS Y RESULTADOS DEL PROYECTO	126
4.1	Pruebas de funcionamiento.....	126
4.1.1	Programa Enviar datos desde PC a Ubidots.....	126
4.1.2	Programa Recibir datos desde Plataforma Ubidots a la PC	127
4.1.3	Programa Control de Velocidad del Motor DC del EPC conectado a la Plataforma Ubidots	128
4.2	Pruebas de Latencia de PC y NI myRIO en relación a la conexión con la Plataforma Ubidots.....	129
4.3	Pruebas de Trafico de Red.....	130
4.3.1	Programa Enviar datos desde PC a Ubidots.	130
4.3.2	Programa Recibir datos desde Plataforma Ubidots a la PC	131
4.3.3	Programa Control de Velocidad del Motor DC del EPC conectado a la Plataforma Ubidots	132
4.4	Validación de la hipótesis	133

CAPÍTULO V	135
5. CONCLUSIONES Y RECOMENDACIONES	135
5.1 Conclusiones.....	135
5.2 Recomendaciones.....	136
REFERENCIAS BIBLIOGRÁFICAS	138
ANEXOS	141

ÍNDICE DE FIGURAS

Figura 1 El Internet de las cosas (IoT)	25
Figura 2 Arquitectura IoT	28
Figura 3 Arquitectura IoT de tres capas (IEEE).....	29
Figura 4 Modelo de referencia IoT (UIT).....	30
Figura 5 La industria 4.0 engloba varios campos dentro de la industria	31
Figura 6 Internet Industrial de las cosas	32
Figura 7 IoT vs IIoT.....	34
Figura 8 Modelo cliente/servidor.....	35
Figura 9 Patrón petición-respuesta de las comunicaciones entre procesos.....	36
Figura 10 Modelo publicación / suscripción	38
Figura 11 Componentes de un instrumento virtual (vi).....	47
Figura 12 Distribución de pines NI myRIO	49
Figura 13 Diagrama esquemático del EPC.....	50
Figura 14 Estructura Plataforma Ubidots	55
Figura 15 Esquema de integración de los componentes.....	58
Figura 16 Interfaz sing up Ubidots	59
Figura 17 Interfaz creación de cuenta Ubidots.....	59
Figura 18 Interfaz Api credentials	60
Figura 19 Creación de un dispositivo.....	61
Figura 20 Panel del dispositivo	61
Figura 21 Panel de dispositivos creados.....	62
Figura 22 Creación variable.....	63
Figura 23 Añadir variable.....	63
Figura 24 Modificar nombre de la variable	64
Figura 25 Panel de variables creadas.....	64
Figura 26 Ingreso sección dashboards	65
Figura 27 Panel demo dashboard.....	66
Figura 28 Agregar nuevo dashboard	66
Figura 29 Interfaz propiedades del nuevo dashboard	67
Figura 30 Interfaz dashboard creada	67
Figura 31 Panel de widgets	68
Figura 32 Interfaz widget gauge.....	69
Figura 33 Asignación de variables al widget.....	69
Figura 34 Selección de variables	70
Figura 35 Propiedades del gauge	70
Figura 36 Dashboard con diferentes widgets	71
Figura 37 Paleta de funciones http	72
Figura 38 Estructura post.....	72
Figura 39 Estructura get	73
Figura 40 Sub vi inicio de conexión	74
Figura 41 Sub vi cierre de conexión.....	74
Figura 42 Sub vi credenciales de escritura	75
Figura 43 Sub vi credenciales de lectura	75
Figura 44 Sub vi conversión booleano a string	76

Figura 45 Sub vi conversión dato numérico a string.....	76
Figura 46 Sub vi escribir datos a Ubidots.....	77
Figura 47 Sub vi lectura de dato booleano desde Ubidots	78
Figura 48 Sub vi lectura de datos numéricos desde Ubidots.....	78
Figura 49 Panel frontal programa ping desde pc	80
Figura 50 Diagrama de bloques programa ping pc a Ubidots	80
Figura 51 Diagrama de bloques programa enviar datos desde pc a Ubidots	82
Figura 52 Entramado de datos	83
Figura 53 Panel frontal del programa enviar datos desde pc a Ubidots	84
Figura 54 Dashboard plataforma Ubidots	84
Figura 55 Diagrama de bloques programa recibir datos	86
Figura 56 Panel frontal programa recibir datos	87
Figura 57 Dashboard Ubidots	87
Figura 58 Creación eventos.....	88
Figura 59 Interfaz crear evento.....	89
Figura 60 Interfaz de condición de evento	90
Figura 61 Selección de variable para el evento	90
Figura 62 Ingreso de condiciones para evento	91
Figura 63 Ingresar una nueva acción.....	91
Figura 64 Acciones para el evento.....	92
Figura 65 Configuración sms.....	92
Figura 66 Días y frecuencia que se produce la acción.....	93
Figura 67 Sms de texto que llega al celular	94
Figura 68 Creación reporte.....	95
Figura 69 Adición de reporte.....	95
Figura 70 Configuración de reporte	96
Figura 71 Configuración de tablas y gráficos.....	97
Figura 72 Reporte variable temperatura en excel	97
Figura 73 Interfaz NI Max	98
Figura 74 Network Settings.....	99
Figura 75 Software instalado en NI myRIO.....	99
Figura 76 Interfaz de Log in	100
Figura 77 Interfaz de instalación.....	101
Figura 78 Finalización de instalación	101
Figura 79 Panel frontal programa ping NI myRIO	103
Figura 80 Diagrama de bloques programa ping NI myRIO.....	104
Figura 81 Esquema de funcionamiento programa control de velocidad EPC.....	105
Figura 82 Dashboard programa control de velocidad de motor dc del EPC.....	106
Figura 83 Controles e indicadores programa control de velocidad EPC.....	107
Figura 84 Recepción de variable setpoint desde dashboard Ubidots.....	108
Figura 85 Escritura de velocidad rpm al dashboard de Ubidots	109
Figura 86 Lazo de comunicación programa control de velocidad motor dc del EPC ...	109
Figura 87 Conversión medición encoder a rpm.....	110
Figura 88 Diagrama del lazo de control de velocidad motor dc.....	111
Figura 89 Diagrama de bloques del lazo de control	111
Figura 90 Conexión de NI myRIO con el EPC	112
Figura 91 Panel frontal del programa control de velocidad motor dc en ejecución.....	113

Figura 92 Dashboard de Ubidots programa control de velocidad en ejecución	113
Figura 93 Crear organización	114
Figura 94 Nombre de la organización	115
Figura 95 Seleccionar la aplicación para la organización.....	115
Figura 96 Editar organización	116
Figura 97 Enlazar un dashboard a la organización	116
Figura 98 Selección del dashboard para la organización	117
Figura 99 Dashboard asignado a la organización	117
Figura 100 Selección de dispositivo para la organización	118
Figura 101 Organización creada con todos sus elementos.....	118
Figura 102 Creación de usuario.....	119
Figura 103 Ingreso de nombre de usuario y email	120
Figura 104 Asignar una organización a un usuario	120
Figura 105 Selección de rol para el usuario.....	121
Figura 106 Guardar el usuario	122
Figura 107 Establecer contraseña de usuario.....	122
Figura 108 Web browser para ingresar al dashboard de forma remota.....	123
Figura 109 Dashboard del programa visualizado de forma de remota	124
Figura 110 Acceso al dashboard desde un smartphone en forma remota	125
Figura 111 Resultados tráfico de red programa Wireshark al enviar datos	130
Figura 112 Resultados tráfico de red en el programa Wireshark al recibir datos.....	131
Figura 113 Código para determinar la veces que se enciende la escritura remota	132
Figura 114 Código para medir cantidad de datos recibidos	133

ÍNDICE DE TABLAS

Tabla 1	<i>Tabla comparativa sobre algunos de los protocolos.....</i>	41
Tabla 2	<i>Resultados de la prueba del programa enviar datos desde pc a Ubidots</i>	126
Tabla 3	<i>Resultados de funcionamiento programa recibir datos desde Ubidots a pc....</i>	127
Tabla 4	<i>Pruebas de funcionamiento programa control de velocidad del motor DC.....</i>	128
Tabla 5	<i>Prueba de latencia en pc y NI myRIO conexión Ubidots.....</i>	129
Tabla 6	<i>Prueba tráfico de red en envío y recepción de datos pc-Ubidots.....</i>	131
Tabla 7	<i>Resultados de tráfico del control de velocidad del motor dc</i>	133

RESUMEN

Se desarrolló un conjunto de Sub VI's en el Software LabVIEW que permiten la conexión de una PC y del Sistema Embebido NI myRIO con la Plataforma de Servicios en la Nube Ubidots, se logró realizar la conexión utilizando el protocolo de comunicación HTTP (Paleta de Funciones HTTP Client Software LabVIEW) y el formato de texto JSON, con estos Sub VI's fueron desarrollados 3 programas; el primero envía datos desde la PC hacia la Plataforma Ubidots, el segundo recibe datos desde la Plataforma Ubidots a la PC y por último se realizó el Control de un motor DC, de forma local que puede manipular el setpoint y visualizar la velocidad del motor en RPM, de igual manera esto se lo puede hacer de forma remota en la Plataforma de Ubidots. Además, se hicieron pruebas de concepto aplicables al IoT (Internet de las Cosas) e IIoT (Internet Industrial de las cosas). Los resultados encontrados después de las pruebas en torno a latencia fueron de 220 ms para la PC y de 261.65 ms para el Sistema Embebido NI myRIO, para tráfico de datos en una hora de ejecución del programa fueron de 2.375 kB en la PC y de en 42,93 kB el Sistema Embebido NI myRIO.

PALABRAS CLAVE:

- **IIOT**
- **PROTOCOLO DE COMUNICACIÓN HTTP**
- **PLATAFORMA DE SERVICIOS EN LA NUBE**
- **SISTEMAS EMBEBIDO NI MYRIO**

ABSTRACT

A set of SubVIs was developed using LabVIEW. They allow communication between a PC, an embedded system known as MyRio and the IoT platform Ubidots. To achieve this interaction, HTTP and the open standard file format JSON were used. Once the SubVIs were created, three applications were built. The first application collects and sends data from the PC to the platform Ubidots, the second one receives data from Ubidots to the PC and finally an application to control a DC motor speed was built, where parameters may be manipulated locally as well as remotely by means of Ubidots. In addition, a proof of concept applicable to the IoT and the IIoT was developed. After several latency tests the results were 220ms for PC and 261.65 ms for the embedded system NI MyRio. On the other hand traffic network results were 2375kb for PC and 42.93 kb for NI MyRio. To get these results, it was required to execute the applications during at least 60 minutes.

KEYWORDS:

- **IIOT**
- **COMMUNICATION PROTOCOL HTTP**
- **CLOUD SERVICES PLATFORM**
- **EMBEDDED SYSTEMS NI MYRIO**

CAPÍTULO I

1. INTRODUCCIÓN

1.1 Planteamiento del problema

Con la llegada de System Link Cloud al mundo de los servicios en la nube y como reemplazo del fallido toolkit IoT Education Beta, se comenzó a trabajar con el servicio en la Empresa Datalights, conjuntamente con la tarjeta NI myRIO, después varios meses de pruebas, comenzaron a existir problemas en el funcionamiento, problemas relacionados a pérdida de conexión, al realizar la consulta con el departamento encargado en National Instruments, y al tener esos problemas muchos de los usuarios del nuevo servicio, se llegó a la conclusión por parte de National Instruments, que el servicio System Link Cloud es recomendable usarlo en tarjetas con una memoria RAM superior a 550MB, dejando de lado tarjetas como NI myRIO y NI MyDAQ que tienen memoria RAM de 250MB.

1.2 Antecedentes

En la próxima década, con la implementación de nuevas aplicaciones habilitadas para Internet, la economía mundial cambiará considerablemente. Es posible que algunas de estas aplicaciones provoquen grandes cambios, sean innovadoras y tengan, también, un gran impacto económico. Esto afectará a muchas industrias y sectores, tal como ya se ha visto en los sectores de consumo, como hoteles (Airbnb) y taxis (Uber).

Esta tendencia también se puede observar en la industria; por ejemplo, la iniciativa Industria 4.0 de Alemania. Tanto en Industria 4.0 como en Internet Industrial de las Cosas (Industrial Internet of Things, IIoT), el enfoque se centra en la integración de las tecnologías existentes y en un mejor uso de ellas. En este sentido, la industria, las máquinas, los productos y las personas están todos conectados digitalmente. Los fabricantes están produciendo en masa productos individuales, hechos a medida.

Software inteligente en la nube conecta los sistemas de TI de las empresas con el mundo operativo, las máquinas, los dispositivos y los sensores para controlar y optimizar el flujo de producción. (National Instruments, 2020)

Siguiendo la tendencia mundial National Instruments lanzo el toolkit IoT Education Beta, un toolkit que permitía la conexión con la plataforma de servicios en la nube Thingworx de PTC, que en sus inicios funciono de manera rápida e intuitiva para sus usuarios, pero que después de cierto tiempo de pruebas se determinó que no cumplía con las expectativas que se tenían.

Posterior a este toolkit se lanzó el servicio de NI System Link Cloud, una plataforma de servicios en la nube propia de National Instruments, que permite monitorear e interactuar de manera segura con aplicaciones desde cualquier parte del mundo sin la carga de hospedar y administrar un servidor, servicio que actualmente se encuentra trabajando correctamente.

De acuerdo a los distintos oficios que se han realizado mediante las pruebas que se han efectuado para las diversas plataformas de servicios de internet en la nube varias de estas están orientadas al control y monitoreo de las mismas, muchas de ellas han adquirido cierta denotación por lo que se requiere hacer una orientación para las distintas acciones lo que permite acceder a ciertos privilegios muchos de estos antecedentes son descritos por algunas de las siguientes propuestas

1.3 Justificación e importancia

Al tener un servicio completamente levantado y probado como lo es System Link Cloud, pero limitado en el sentido de correcto funcionamiento a partir de requisitos como lo son RAM en los sistemas embebidos, nace la necesidad de poder tener una conexión que no presente inconvenientes y que se pueda usar en todo tipo de DAQs y sistemas embebidos sin limitaciones de requisitos en RAM, en el ámbito académico es importante

ya que muchas Universidades poseen tarjetas y sistemas embebidos de National Instruments cuya RAM es menor a 550MB (requisito para el correcto funcionamiento de System Link Cloud) y cuyo costo es menor a aquellas con RAM superior , así que tener un método de conexión sin fallas de comunicación es ideal.

En el ámbito profesional, como solución industrial, se torna importante en el sentido de libre elección de la Plataforma de Servicios en la Nube, permitiendo ajustarse al presupuesto y necesidades específicas del cliente, evitando así tener una única opción en Plataforma de Servicios en la Nube y permitiendo que la Empresa tenga mayor oferta al momento de vender sus servicios.

1.4 Objetivos

1.4.1 Objetivo general

- Desarrollar una prueba de concepto aplicable al Internet Industrial de las Cosas (IIoT) utilizando Sistemas Embebidos de National Instruments, lenguaje de nivel de abstracción medio y una plataforma de servicios en la nube para aplicabilidad tanto académica y profesional en Datalights Distribuidor Autorizado de National Instruments.

1.4.2 Objetivos específicos

- Realizar una investigación bibliográfica acerca de Protocolos de aplicaciones comunes para la conectividad de IoT.
- Realizar una investigación bibliográfica de Plataformas de desarrollo en la nube.
- Desarrollar el código, en ambiente de programación Labview, que permita probar conexión de un sistema embebido con una Plataforma de servicios en la Nube, usando lenguaje de abstracción medio.
- Realizar pruebas de concepto de la conexión de los sistemas embebidos con

diversas Plataformas de Servicios en la Nube.

- Determinar si es posible usar un lenguaje de abstracción medio para conectar un sistema embebido con una plataforma de servicios en la nube y de ser posible cuál de estos servicios funciona de la manera más adecuada.

1.5 Variables de la investigación

1.5.1 Variable independiente

Lenguaje Labview de Nivel de Abstracción Medio

1.5.2 Variable dependiente

Prueba de concepto aplicable al Internet Industrial de las Cosas (IIoT)

1.6 Hipótesis

Realizar la conexión de un Sistema Embebido de National Instruments con una plataforma de Servicios en la Nube, utilizando lenguaje LabVIEW de nivel de abstracción medio.

CAPITULO II

2. MARCO TEÓRICO

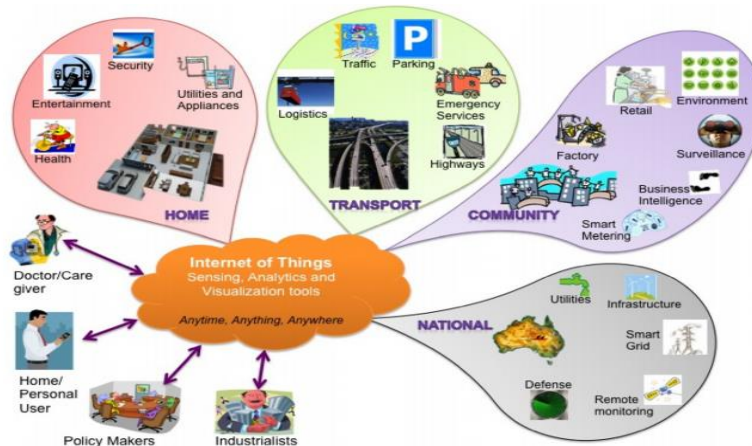
2.1 Internet de las cosas (IoT)

La recomendación de la UIT-T, Y.2060 (UIT, 2012), define IoT como “una infraestructura global de la sociedad de la información, que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras”.

Se puede definir IoT, como una red global encargada de interconectar objetos tanto físicos como virtuales capaces de interactuar entre sí y con el ser humano. Consiste en la integración de dispositivos y sensores que pueden establecer una interconexión mediante internet a través de redes fijas o inalámbricas de tal manera de que las cosas puedan ser monitorizadas en cualquier momento y lugar.

Figura 1

El Internet de las cosas (IoT)



Nota: El gráfico representa un esquema de Internet de las cosas que muestra los usuarios finales y las áreas de aplicación en función de los datos. Tomado de Internet of Things (IoT): A vision, architectural elements, and future directions (p.3), por J. Gubbi et al, 2013, Elsevier B.V.

Las características fundamentales de IoT, como se describen en (UIT, 2012) son las siguientes:

- **Interconectividad:** La característica primordial que provee al IoT de toda su importancia, que permite el acceso y compatibilidad con la infraestructura mundial de la información y la comunicación.
- **Heterogeneidad:** Los dispositivos en IoT se basan en diferentes plataformas hardware y redes por lo que podrán interactuar con otros dispositivos o plataformas de servicios a través de diferentes redes.
- **Cambios dinámicos:** El estado (conectado, desconectado, activo, inactivo), contexto (velocidad, ubicación, nivel, temperatura, etc) de los dispositivos pueden variar dinámicamente.

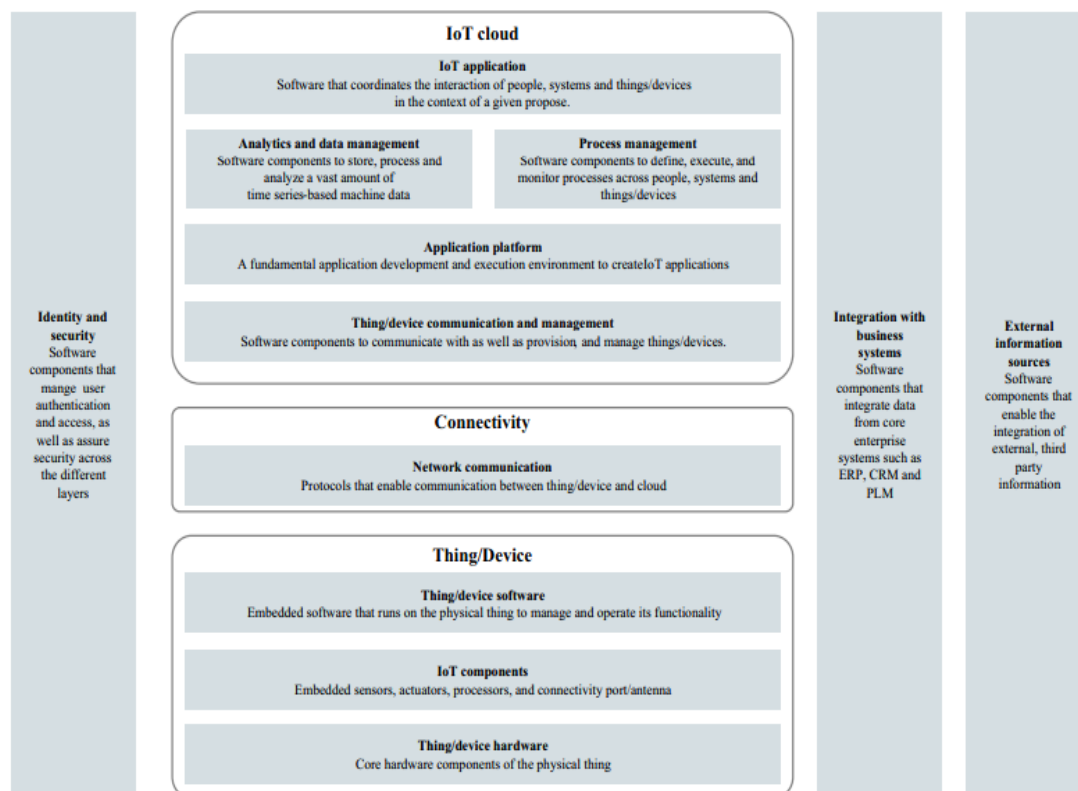
- Escalabilidad: Los dispositivos IoT interconectados pueden incrementarse de manera significativa durante los próximos años por lo que es esencial una mayor gestión, interpretación y manipulación eficiente de los datos generados.

Dentro de la amplia gama de aplicaciones de los sistemas IoT que se describen en (Serpanos & Wolf, 2018) se encuentran:

- En la industria hoy en día es muy común el uso de sensores para el monitoreo tanto de los procesos industriales en sí, la calidad del producto, así como el estado del equipo. Por ejemplo, actualmente se ha incrementado el número de motores eléctricos los cuales incluyen sensores que recopilan datos para predecir inminentes fallas del motor.
- Los edificios inteligentes cuentan con sensores para identificar el estado en el que éste se encuentra. Estos datos pueden utilizarse para controlar sistemas de calefacción, ventilación y sistemas de iluminación para disminuir costos operativos.
- Las ciudades inteligentes usan sensores que monitorean tanto el tráfico de peatones como de vehículos y su vez son capaces de integrar datos de edificios inteligentes.
- Los vehículos incorporan una red de sensores para monitorear el estado del vehículo, localización y selección de rutas, identificación de objetos y señales de tránsito, así como una alta potencia de cálculo para administrar estos datos. Los sistemas médicos implementan una amplia gama de sensores para la monitorización de pacientes que se encuentran en sus hogares, en vehículos de emergencia o en el hospital.

2.1.1 Arquitectura IoT

Desde una perspectiva tecnológica, la implementación de un producto conectado a IoT requiere la combinación de componentes tanto de software como de hardware en una pila de varias capas de tecnologías de IoT. Como se muestra en la *Figura 2*, dicha pila de tecnología de IoT, se compone de tres capas centrales, la capa de dispositivo, la capa de conectividad y la capa de nube de IoT. En la capa de dispositivo, se puede agregar hardware específico de IoT, como sensores, actuadores o procesadores y el software integrado se puede modificar o integrar nuevamente para administrar y operar la funcionalidad de los dispositivos. En la capa intermedia de conectividad, los protocolos de comunicación como TCP, HTTP, MQTT permiten la comunicación entre un dispositivo y la nube. Y en última capa de nube de IoT, el software de gestión y comunicación se utiliza para comunicar y administrar los dispositivos conectados a la nube, mientras que una plataforma de aplicaciones permite el desarrollo y ejecución de aplicaciones de IoT. (Wortmann & Flüchter, 2015)

Figura 2**Arquitectura IoT**

Nota: La gráfica muestra las tres capas centrales que frecuentemente se mencionan en la arquitectura IoT. Tomado de Internet of Things Technology and Value Added (p.3), por F. Wortmann & K. Flüchter, 2015, Springer Fachmedien Wiesbade.

Según (RS , 2018), existen tres elementos fundamentales que normalmente se indican en la arquitectura IoT.

- **Dispositivos IoT:** Los Dispositivos IoT son aquellos que se pueden conectar mediante cable o a través de manera inalámbrica a una red, pequeña o amplia; estos dispositivos pueden ser motores, elementos actuadores o sensores.

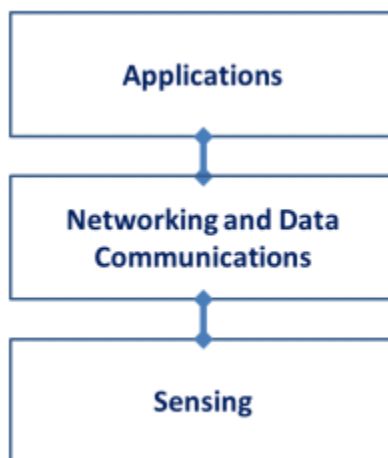
- **Redes:** Se encargan de ser las puertas de enlace a la red, o la nube con otros dispositivos conectados a la red.
- **Nube:** Son centros de datos que se alojan en servidores web remotos que almacenan y consolidan la información con seguridad.

Se presentan distintas propuestas de arquitectura dependiendo de la organización que trata de estandarizar IoT:

La IEEE propone una arquitectura basada en tres capas, como se muestra en la *Figura 3*. (IEEE, 2015)

Figura 3

Arquitectura IoT de tres capas (IEEE)



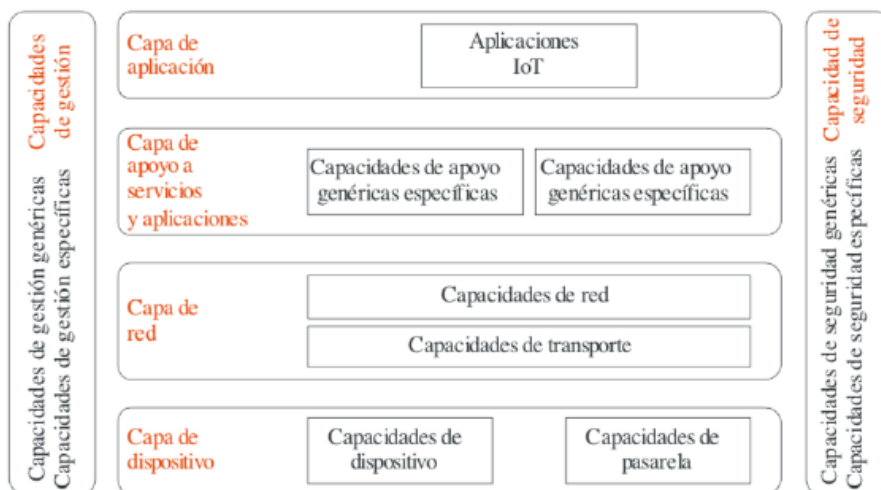
Nota: El gráfico muestra la arquitectura propuesta por la IEEE la cual consiste en tres capas, la primera capa de sensado, una capa intermedia de redes y comunicaciones de datos, y finalmente una capa de aplicación.

UIT propone un modelo de referencia que se muestra en la *Figura 4*. Dispone de cuatro capas: capa de dispositivo, capa de red, capa de apoyo a servicios y aplicaciones, y capa de aplicación; y dos capacidades: capacidad de seguridad y

capacidad de gestión. Estas dos capacidades se asocian con las cuatro capas antes mencionadas. (UIT, 2012).

Figura 4

Modelo de referencia IoT (UIT)



Y.2060(12)_F04

Nota: En la gráfica se observa un modelo referencial IoT que consta de cuatro capas y de capacidades de gestión y de seguridad relacionadas con estas cuatro capas.

2.2 Industria 4.0

La industria 4.0 es una iniciativa estratégica que nace en Alemania cuyo objetivo es llevar las tecnologías IoT a los sectores de fabricación y producción.

Este concepto "Industria 4.0" hace referencia a la cuarta revolución de la producción industrial. Es sumamente aceptado que hasta el día de hoy la producción industrial ha pasado por tres revoluciones. La primera revolución industrial, impulsada por la producción mecanizada en la cual su principal fuente de energía era aportada por agua y vapor. Durante la segunda revolución se introdujo la energía eléctrica, lo que llevó a la producción masiva, puesto que, gracias a la electricidad, la productividad se incrementó notablemente. Después de la Segunda Guerra Mundial, la inclusión de la

electrónica, a los componentes eléctricos y mecánicos condujo a la tercera revolución que permitió una producción automatizada. Actualmente, se cree que estamos al borde la cuarta revolución industrial, a través de la incorporación y uso de sistemas ciberfísicos que no solamente conducen a niveles de automatización superiores, sino que también permiten la fabricación y producción masiva, debido a la flexibilidad que ofrecen las líneas de fabricación fácilmente programables, configurables y controlables. (Serpanos & Wolf, 2018)

Figura 5

La industria 4.0 engloba varios campos dentro de la industria



Nota: El gráfico muestra algunos de los diferentes campos abarcados de la industria 4.0 tales como: edificios y productos inteligentes, Big Data, Cloud Computing, automatización robótica, realidad aumentada, etc. Tomada de Estandarización para la Industria 4.0 por Asociación Española de Normalización (p. 15).

2.3 Internet Industrial de las Cosas (IIoT)

La Internet Industrial de las Cosas (IIoT) está basada en la aplicación de tecnologías de Internet de Cosas (IoT) en los sectores industriales. Es una generalización de la industria 4.0 la cual se enfoca en la eficiencia de los procesos industriales. La visión de la IIoT abarca todos los aspectos dentro de las operaciones de una industria, no se enfoca solamente en la eficiencia del proceso, sino también en el mantenimiento, la gestión de activos, entre otros. (Serpanos & Wolf, 2018)

Figura 6

Internet Industrial de las cosas



Nota: En la figura se muestra algunas de las principales características del IIoT.

Tomado de What is Industrial Internet of Things: Overview por Altizon, 2020.

2.3.1 Características de la IIoT

- Una de las características esenciales es la inteligencia, ya que es posible interconectar múltiples dispositivos de un determinado proceso en entidades inteligentes con capacidad de auto organización, y actuar de forma

independiente de acuerdo a las circunstancias que se encuentren. (Optical Networks , 2020)

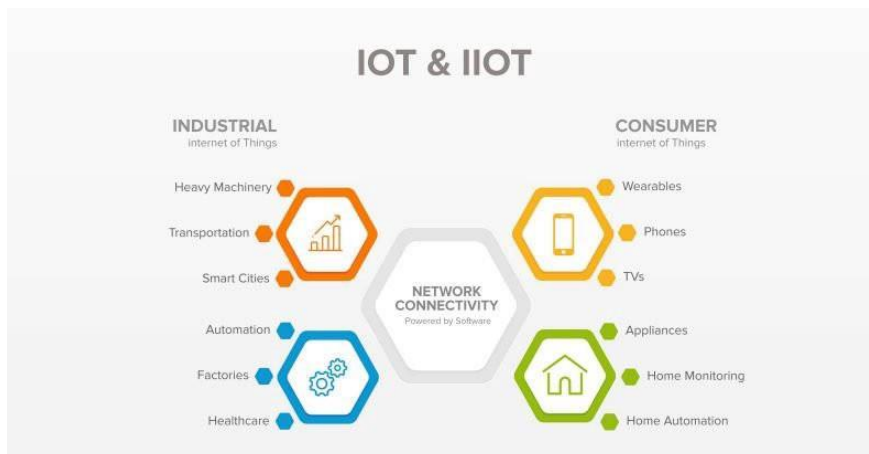
- La arquitectura, al ser un concepto nuevo, la tendencia es que sea una “arquitectura basada en eventos” la cual tenga la capacidad de tomar decisiones en función de los datos. (Optical Networks , 2020)
- Es un sistema complejo, esto se debe a que se produce una gran cantidad de enlaces e interacciones de un gran número de agentes autónomos, es decir se considera compleja su estructura la existir una gran cantidad de subredes dentro de una gran red que es internet. (Optical Networks , 2020)
- Los dispositivos IIoT están expuestos a condiciones extremas y deben funcionar bajo esas condiciones por lo que son construidos más resistentes justamente para trabajar a nivel industrial. (Llanos, 2020)
- La seguridad es un punto clave en estos dispositivos por lo que los sistemas de seguridad son más robustos que los de IoT, ya que un ciberataque puede resultar fatal debido a que IIoT conecta infraestructura y redes para el funcionamiento industrial. (Llanos, 2020)

2.4 Diferencias entre IoT e IIoT

Tecnológicamente, el IoT del consumidor y el IoT industrial son más diferentes de lo que son similares. La principal diferencia es que la IIoT hace referencia especialmente a dispositivos industriales, mientras que IoT se enfoca en dispositivos y aplicaciones como hogares inteligentes y dispositivos personales cotidianos, los cuales no crean una situación de emergencia en caso de que algo salga mal. (Schneider, 2016)

Figura 7

IoT vs IIoT



Nota: La figura muestra las competencias de la IoT e IIoT. Las aplicaciones de IoT son varias: hogares inteligentes, edificios inteligentes y ciudades inteligentes, así como dispositivos personales cotidianos. Mientras que IIoT cuenta con aplicaciones enfocadas en los sectores industriales y de fabricación. Tomada de IoT & IIoT in Real Life por L. Grasso, 2019.

Ambos dispositivos, tanto los IoT como los IIoT generan variables que pueden ser monitorizadas remotamente a través de su propia red o a través de una red pública. En el caso de los dispositivos IIoT, nos referimos a dispositivos más sofisticados que abarcan más datos, más conexiones y en general un funcionamiento mucho más complejo. A su vez, suelen ser mucho más robustos, y, por lo tanto, tiene un costo mayor, dado que habitualmente estarán sometidos a unas condiciones y ambientes diferentes que los dispositivos IoT. (Llanos, 2020)

2.5 Paradigmas de la comunicación

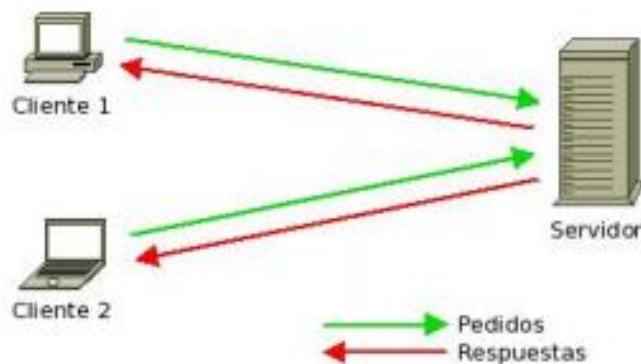
Existen varios protocolos para interconectar dispositivos y maquinas que trabajan bajo la filosofía del internet de las cosas. Cliente /Servidor y Publicador/ Suscriptor son los tipos de paradigmas de comunicación más utilizados.

2.5.1 Modelo Cliente / Servidor

Los protocolos cliente / servidor requieren que el cliente se conecte al servidor y realice las solicitudes. En este modelo, el servidor retiene los datos y responde a los pedidos o solicitudes que el cliente realice. Previamente es necesario que el cliente conozca al servidor para que pueda conectarse a la sesión. (Semle, 2016)

Figura 8

Modelo cliente/servidor



Nota: En la figura se muestra e modelo cliente/ servidor en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Tomado de El Modelo Cliente/Servidor por E. Marini (p.1), 2012.

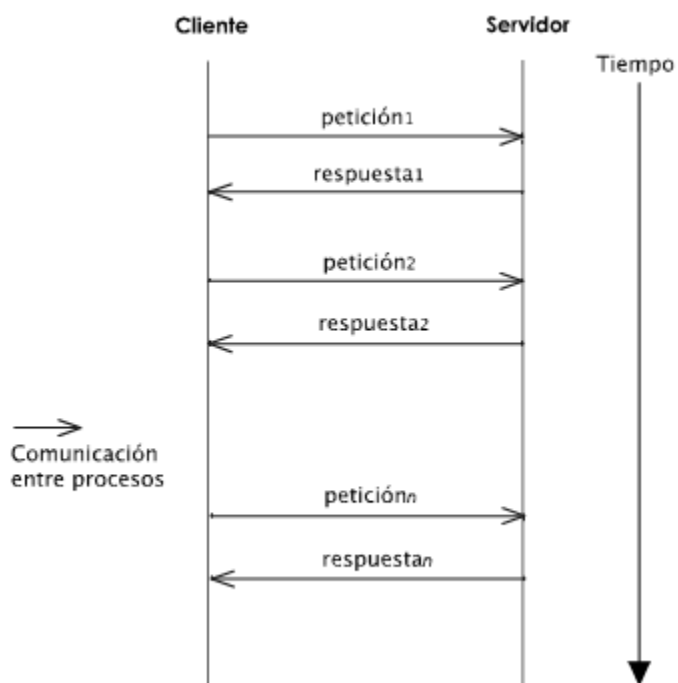
En el modelo cliente / servidor, la interacción entre los procesos sigue un patrón de petición - respuesta como se muestra en la *Figura 9*. En una sesión de servicio, un cliente realiza una petición al servidor, que contesta con una respuesta El cliente puede

hacer una petición subsiguiente, seguida por una respuesta del servidor. Este proceso puede repetirse de manera indefinida hasta que caduque la sesión. (Damian, 2020)

Por cada petición realizada, el cliente debe esperar por la respuesta del servidor antes de continuar.

Figura 9

Patrón petición-respuesta de las comunicaciones entre procesos



Nota: En la figura se muestra el patrón petición respuesta en la interacción de procesos del modelo cliente/ servidor. Tomado de El paradigma cliente-servidor de O. Damián (p.1),2020.

2.5.2 Modelo Publicación / Suscripción

Un protocolo de publicación / suscripción requiere menos acoplamiento entre el cliente y el servidor, como se ilustra en la figura 10. Este modelo requiere que los

dispositivos se conecten y publiquen datos en un topic. Los consumidores pueden conectarse al agente y suscribirse a los datos del topic. (Semle, 2016)

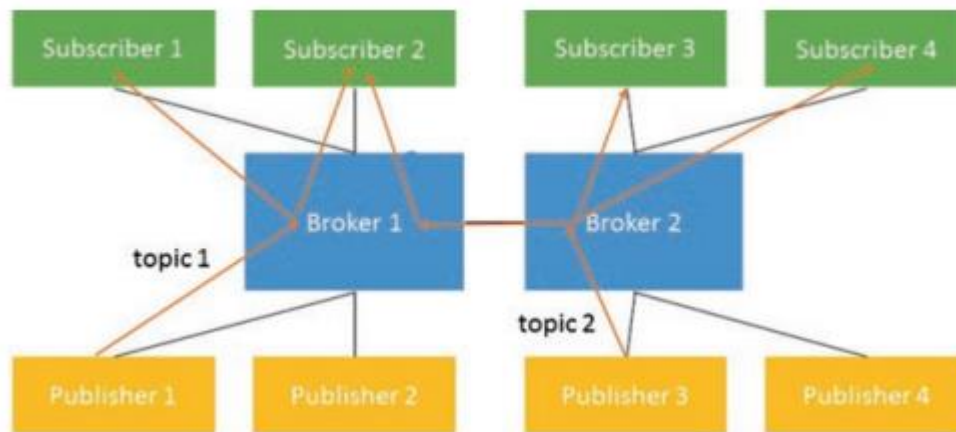
Las partes implicadas en la comunicación son:

- Un consumidor o suscriptor, recibe información de los *Topics* a los que está suscrito.
- Un publicador, puede publicar tantos *Topics* diferentes como crea preciso.
- Un *Broker* o mediador, es el encargado de recoger la información de los *Topics* y enviarla a los suscriptores.

Los clientes se suscriben a las categorías que les interesan. Los sistemas de publicación / suscripción suelen estar mediados por intermediarios que reciben mensajes publicados de los editores y los envían a los suscriptores. Los mensajes pueden estar organizados por tema o topics; todos los mensajes de un topic determinado son distribuidos por los intermediarios a los suscriptores de ese topic. El broker conoce las identidades de los suscriptores, pero el editor no. Los brokers pueden interactuar entre sí mediante un protocolo puente. Un puente permite la publicación indirecta de mensajes, con un mensaje que va del editor a un primer broker, luego a un segundo broker y finalmente a los suscriptores que no están conectados al primer broker. (Serpanos & Wolf, 2018)

Figura 10

Modelo publicación / suscripción



Nota: En el gráfico se muestra el modelo publicación/ suscripción en el cual el suscriptor puede estar suscrito a uno o varios Topics, de un mismo publicador o de varios. El publicador es el responsable de definir los diferentes tipos de Topics a los cuáles se pueden suscribir los suscriptores. Tomado de Internet of Things (IoT) Systems por Serpanos & Wolf (p.8), 2018, Springer International Publishing.

Los protocolos que manejan el modelo cliente/servidor son más compatibles y seguros porque se basan en conexiones punto a punto. Sin embargo, son menos escalables porque las conexiones punto a punto son más difíciles de manejar y mayores demandantes de recursos. Por otro lado, los protocolos que se manejan bajo el modelo publicación / suscripción son más escalables porque el separar a productores de consumidores permite que cada uno se agregue y se quite de forma independiente. También pueden aparecer cuestiones de compatibilidad dada la separación entre productor y consumidor. Por ejemplo, un cambio en el formato del mensaje enviado por el productor requiere que los consumidores se ajusten a dicho formato (Semle, 2016)

2.6 Protocolos de comunicación IIoT

El protocolo de IoT que se utilice dependerá del nivel de arquitectura del sistema en el que deban moverse los datos. El modelo de interconexión de sistemas abiertos (OSI) proporciona un mapa de los distintos niveles que envían y reciben datos. Cada protocolo de la arquitectura del sistema de IoT permite la comunicación de dispositivo a dispositivo, de dispositivo a puerta de enlace, de puerta de enlace a datos o de puerta de enlace a la nube, así como la comunicación entre centros de datos. (Microsoft Azure, 2020)

2.6.1 Nivel de aplicación

El nivel de aplicación actúa como interfaz entre el usuario y el dispositivo.

a) HTTP (REST/JSON)

El Protocolo de transferencia de hipertexto (HTTP) es un protocolo basado en el modelo cliente / servidor ubicuo en la web. Debido a que existen un sin número de herramientas de código abierto que usan HTTP y resulta muy accesible. (National Instruments, 2019)

El enfoque en HTTP y en IoT se basa en la Transferencia de Estado Representacional (REST), el cual es un modelo en el que los clientes pueden acceder a los recursos en el servidor a través de solicitudes. (National Instruments, 2019)

La industria tiene cierta experiencia en el uso de HTTP para la configuración de dispositivos y productos, pero no para el acceso a datos. Como tal, muchas plataformas de IoT y de TI admiten el consumo y el suministro de datos en forma HTTP, pero pocas plataformas industriales lo hacen. Esto está cambiando a medida que más puertas de

enlace y PLC comienzan a agregar soporte nativo de HTTP. (National Instruments, 2019)

b) MQTT

MQTT (Message Queuing Telemetry Transport, 'Cola de mensajes telemetría y transporte') es un protocolo publicación/suscripción diseñado para SCADA y redes remotas. (National Instruments, 2019)

Este protocolo de mensajería está diseñado para la comunicación ligera entre dispositivos el cual es usado comúnmente para las conexiones con ubicaciones remotas y poco ancho de banda. MQTT es ideal para dispositivos pequeños que requieren un uso eficiente del ancho de banda y de la batería. (Microsoft Azure, 2020)

c) CoAP

CoAP (Constrained Application Protocol, 'protocolo de aplicación restringida') fue creado para proveer la compatibilidad de HTTP con una mínima carga. Es un protocolo muy similar a HTTP, pero hace uso de UDP en lugar de TCP. (National Instruments, 2019)

Además, simplifica el encabezado HTTP y reduce el tamaño de cada requerimiento. Hoy en día es uno de los protocolos más utilizados por las plataformas de IoT, después de HTTP y MQTT. (Microsoft Azure, 2020)

d) DDS

DDS (Data Distribution Service, 'servicio de distribución de datos') es un protocolo basado en el modelo publicación/suscripción. DDS es un estándar abierto para tiempo real, confiable y de alto rendimiento operado por OMG (Object Management Group, 'Grupo de Gestión de Objetos'). A diferencia del protocolo MQTT,

que necesita de un agente centralizado, DDS está descentralizado. Esto hace que sea un protocolo de comunicación punto a punto muy versátil, veloz y con una resolución por debajo del milisegundo, que tiene la capacidad de ejecutar dispositivos pequeños hasta conectar redes de alto rendimiento. Este protocolo optimiza la implementación, cuenta con una arquitectura flexible y adaptable, aumenta la confiabilidad y reduce la complejidad. (Microsoft Azure, 2020)

e) AMQP

AMQP (Advanced Message Queuing Protocol) es otro protocolo basado en el modelo publicación/suscripción el cual proviene del sector financiero. Las principales características de este protocolo son a orientación a mensajes, encolamiento, enrutamiento, exactitud, seguridad una de las más importantes su transmisión sólida y robusta de datos. Y siendo su mayor desventaja, que se trata de un protocolo pesado. (National Instruments, 2019)

Nivel de software que crea interoperabilidad entre el middleware de mensajería. Ayuda a que una gran variedad de aplicaciones y sistemas funcionen juntos, lo que permite crear una mensajería normalizada a escala industrial. (Microsoft Azure, 2020)

Tabla 1

Tabla comparativa sobre algunos de los protocolos

	Transp orte	Modelo	Ámbito de aplicacion	Conocimie nto del contenido	Datos princip ales	Seguridad	Prioridad de los datos	Tolerancia a fallos
AMQP	TCP/I P	Intercam bio de mensaje s punto a punto	D2D D2C C2C	Ninguno	Codific ados	TLS	Ninguno	Especifica de la implementaci ón
CoAP	UDP/I P	Petición/ Respues	D2D	Ninguno	Codific ados	DTLS	Ninguno	Descentraliz ado

	Transp orte	Modelo	Ámbito de aplicacion	Conocimie nto del contenido	Datos princip ales	Seguridad	Prioridad de los datos	Tolerancia a fallos
		ta (REST)						
DDS	UDP/I P (unica st + mcast) TCP/I P	Publicac ión/Susc ripción Petición/ Respues ta	D2D D2C C2C	Enrutamie nto basado en el contenido, consultas	Declar ados codific ados	TLS, DTLS, DDS	Prioridade s de transporte	Descentraliz ado
MQTT	TCP/I P	Publicac ión/Susc ripción	D2D	Ninguno	No definid os	TSL	Ninguno	El nodo central (bróker) es el punto único de fallo (SPoF)

Nota: En esta tabla se resume algunas de las principales características de algunos protocolos. Recuperado de INCIBE.

2.6.2 Nivel de transporte

El nivel de transporte habilita y protege la comunicación de los datos a medida que viajan entre niveles.

a) Protocolo de control de transmisión (TCP)

Protocolo dominante y uno de los fundamentales en internet. Permite conexiones entre hosts, para lo que divide grandes agrupaciones de datos en paquetes individuales que reenvía y vuelve a ensamblar según sea necesario. Esto garantiza que los datos enviados lleguen a su destino en el orden que fueron transmitidos y sin errores.

(Microsoft Azure, 2020)

b) Protocolo de datagramas de usuario (UDP)

Protocolo de comunicaciones muy utilizado en internet que permite la comunicación entre procesos sin necesidad de una conexión previa como ocurre con TCP. Al no disponer de un control de flujo o de congestión puede existir pérdida de información lo que lo convierte en un protocolo no fiable. (Microsoft Azure, 2020)

2.6.3 Nivel de red

El nivel de red permite la comunicación entre los dispositivos individuales y el enrutador.

a) 6LoWPAN

Redes de área inalámbricas de bajo consumo en donde cada nodo tiene su propia dirección IPv6 que reduce los tiempos de transmisión.

b) IPv6

Esta actualización reciente de IP dirige el tráfico a través de Internet e identifica y localiza dispositivos en la red.

c) IEEE 802.15.4

Estándar para redes inalámbricas de bajo poder. Se usa con Zigbee, 6LoWPAN y otros estándares.

d) LPWAN

Redes de área amplia y de baja potencia, este tipo de red permite la comunicación de dispositivos separados por distancias en el rango de kilómetros. LoRaWAN es un ejemplo de este tipo de red optimizada para un consumo de energía muy bajo.

2.6.4 Nivel físico

El nivel físico establece un canal de comunicación que permite que los dispositivos se conecten dentro de un entorno especificado.

a) Bluetooth Low Energy (BLE)

Proporciona un reducido consumo de energía a un considerablemente bajo costo, manteniendo una distancia de conectividad muy similar a la del Bluetooth clásico. Los tiempos de conexión disminuyeron tan solo a unos pocos milisegundos, a diferencia de lo que se tardaba anteriormente. Gracias a su bajo costo y a la larga duración de la batería, BLE se ha convertido en el favorito de la electrónica de consumo. (Microsoft Azure, 2020)

b) Ethernet

Es un protocolo de conexión por cable lo que viene a ser una opción no muy costosa que proporciona una conectividad rápida para datos con una baja latencia. (Microsoft Azure, 2020)

c) Evolución a largo plazo (LTE)

Tecnología de comunicación inalámbrica de banda ancha con la que se pueden transmitir datos a dispositivos móviles y terminales de datos. LTE se destaca por aumentar la capacidad y la velocidad de las redes inalámbricas. (Microsoft Azure, 2020)

d) Transmisión de datos en proximidad (NFC)

Conjunto de protocolos de comunicación que utilizan campos electromagnéticos y permiten comunicaciones entre dispositivos a muy corta distancia, no mayor a los 5

centímetros. Suelen utilizarse comúnmente para realizar pagos móviles. (Microsoft Azure, 2020)

e) Identificación por radiofrecuencia (RFID)

Es una tecnología que permite la identificación y transmisión de información de un objeto a través de etiquetas RFID mediante ondas de radio frecuencia. El hardware compatible se comunica con estas etiquetas para leer su información. (Microsoft Azure, 2020)

f) Wi-Fi/802.11

Es una tecnología que permite acceder a internet de manera inalámbrica a varios dispositivos al conectarse a una red determinada, muy usada en hogares y oficinas. A pesar de ser una opción muy económica, puede que no se ajuste a todos los escenarios por su alcance limitado. (Microsoft Azure, 2020)

2.7 Labview

Labview es un lenguaje de programación gráfica y una poderosa herramienta que permite a los programadores desarrollar aplicaciones de software de manera rápida y sencilla. (Bitter , Mohiuddin, & Nawrocki, 2007)

Es un software desarrollado por National Instruments y lo definen como “LabVIEW es un software de ingeniería de sistemas que requiere pruebas, medidas y control con acceso rápido a hardware e información de datos.” (National Instruments, 2020)

Algunos campos de trabajo con *Labview* son:

- Comunicaciones
- Tiempo Real
- Adquisición y tratamiento de imágenes

- FPGAs
- Sincronización de dispositivos
- Etc.

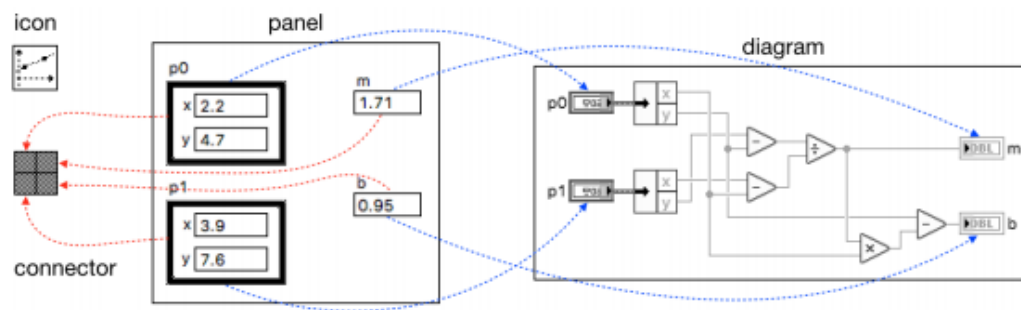
2.7.1 Instrumentos virtuales (VI)

Un instrumento virtual (VI) es un elemento de programación de Labview. Consta de un diagrama de bloques, un panel frontal y un icono. El diagrama de bloques contiene el código del VI. El panel frontal, muestra controles e indicadores para el usuario y el icono es una representación visual del VI que tiene los conectores para las entradas y salidas del programa. (Bitter , Mohiuddin, & Nawrocki, 2007)

Los controles e indicadores son las entradas y salidas de un VI, respectivamente, que aparecen en el panel frontal y que están asociados con una ubicación de terminal en el conector del VI. Cada control e indicador en el panel frontal tiene un terminal de origen y destino asociado en el diagrama de bloques del VI. Los controles e indicadores tienen nombres únicos y etiquetas visibles para facilitar la identificación del terminal de diagrama asociado. (Kodosky, 2020)

Figura 11

Componentes de un instrumento virtual (vi)



Nota: la figura muestra las relaciones entre el panel frontal, el diagrama de bloques y el icono de un VI. Tomado de LabVIEW por J. Kodosky (p.23), 2020.

2.7.2 Metodología de programación en Labview

La programación en Labview se realiza de manera secuencial, por lo tanto, es relevante que la ubicación de las entradas se lo haga a la izquierda y las salidas a la derecha, para conseguir posteriormente una correcta lectura y depuración. (National Instruments, 2020)

La metodología básica de programación de los VI's es la siguiente:

a) Adquisición

Se obtienen datos mediante los controladores que se encuentran en la página oficial de National Instruments. De esta forma se configura las entradas y el tipo de datos que se quieren adquirir para un hardware específico, ya sea de NI como de otros fabricantes. Los VI's Express son frecuentemente utilizados en este punto debido a que facilitan la tarea de configuración de un dispositivo.

b) Análisis

Para analizar los datos, existen múltiples controles en la biblioteca de *Labview*. También se pueden emplear VI Express.

c) Presentación

Para la presentación de los datos, se tiene varios tipos de indicadores como: gráficas, tablas, reportes, leds o secuencia de estados. En caso de que se quiera algo más específico como la creación de texto posteriormente, registrar datos en un fichero o generar una señal es recomendable usar VI Express.

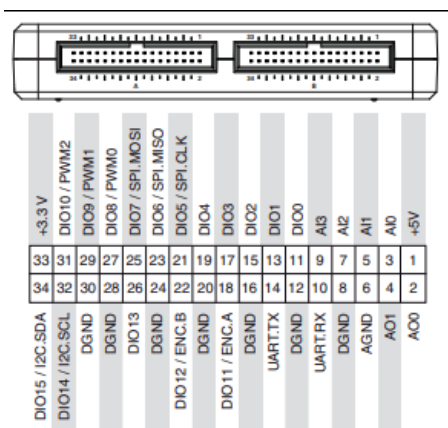
2.7.3 NI MYRIO 1900

El NI myRIO 1900 es un Sistema Embebido de National Instruments diseñado para el trabajo con estudiantes enfocados o involucrados en el diseño de sistemas de control, mecatrónicos y robóticos. (Instruments, USER GUIDE AND SPECIFICATIONS NI myRIO-1900, 2020)

Cuenta con entradas y salidas reconfigurables, “incluye 10 entradas analógicas, 6 salidas analógicas, 40 líneas de E/S digitales, WiFi, LEDs, un push-button, un acelerómetro interno, un FPGA Xilinx y un procesador dual-core ARM Cortex-A9”. (Instruments, NI, 2020)

Figura 12

Distribución de pines NI myRIO



Nota: la figura muestra la distribución de pines del NI myRIO. Tomado de USER GUIDE AND SPECIFICATIONS NI myRIO-1900 por National Instruments (p.6), 2020.

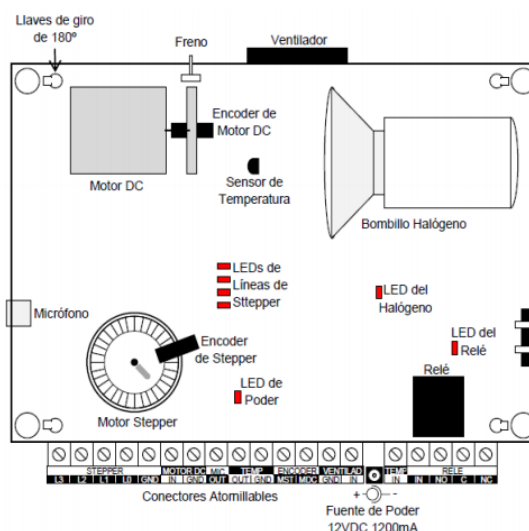
2.7.4 EPC – Entrenador de Planta de Control

La EPC es una planta construida por la empresa Datalights, con el fin de brindar un entorno que simule plantas reales de temperatura, velocidad y posición.

Esta planta está diseñada para conectarse con tarjetas DAQ o Sistemas Embebidos de National Instruments y de esta manera poder desarrollar aplicaciones en el Software LabVIEW.

Figura 13

Diagrama esquemático del EPC



Nota: la figura muestra los sensores y actuadores del EPC. Tomado de Manual EPC por Datalights, 2020.

El EPC posee los siguientes elementos:

- Entradas Digitales
 - Bombillo
 - Ventilador
 - Relé
 - Stepper
- Entradas Analógicas
 - Motor DC
- Salidas Analógicas
 - Temperatura
 - Micrófono (Datalights, 2020)

2.8 Plataformas de desarrollo en la nube

Las plataformas de IoT brindan varias capacidades en la industria, IoT industrial emergente y la cuarta revolución industrial (Industria 4.0) proporciona la flexibilidad para los planificadores e implementadores y conduce a una mejor toma de decisiones.

Además, la supervisión de la máquina y los servicios en la nube, además de las aplicaciones proporcionadas contribuyen al crecimiento y la producción. (Hejazi, Rajab, Cinkler, & Lengyel, 2018)

Algunas de las plataformas de desarrollo en la nube son:

2.8.1 Amazon AWS IoT

Amazon Web Services (AWS) ofrece infraestructura de tecnología de la información, como almacenamiento, base de datos, redes, potencia de cómputo, entre otras. Dentro de las principales ventajas que ofrece esta plataforma se encuentra su bajo costo, mientras más usuarios hagan uso del servicio, más económico es. Se factura solo por los servicios que se utiliza y por ultimo su escalabilidad, el usuario puede contratar y utilizar un servicio según como necesite. (National Instruments, 2019)

2.8.2 IBM Watson IoT para Bluemix

La plataforma de desarrollo en la nube de IBM, Bluemix, es una PaaS. Con Bluemix, puede acceder al software de IBM para servicios, como análisis de grandes volúmenes de datos, almacenamiento, bases de datos, seguridad, IoT, aplicaciones cognitivas y muchos más. (National Instruments, 2019)

Esta plataforma permite que una aplicación establezca comunicación con los dispositivos disponibles y haga uso de los datos recibidos desde esos dispositivos y los gateways de dispositivo. La forma más segura de conexión de los dispositivos con esta plataforma se lo realiza mediante el protocolo MQTT con TLS. (National Instruments, 2019)

2.8.3 PTC ThingWorx

ThingWorx es una plataforma de desarrollo de aplicaciones de IoT basada en un modelo que proporciona un diseño de aplicación completo, tiempo de ejecución y entorno. ThingWorx proporciona componentes para conectividad, análisis, colaboración y visualización, incluida la realidad aumentada. Principales características de desarrollo de la plataforma:

- **Composer:** Es un entorno para modelado de aplicaciones el cual permite el desarrollo de aplicaciones de manera rápida y fácil.
- **Mezclas sin código:** La plataforma utiliza una tecnología “arrastrar y soltar” que permite construir aplicaciones sin la necesidad de escribir código.
- **Motor de ejecución y almacenamiento:** Esta funcionalidad tiene el propósito de almacenar, relacionar y exponer grandes cantidades de datos para darles valor y que sean fácil de utilizar. (National Instruments, 2019)

2.8.4 Microsoft Azure IoT Suite

Azure IoT Suite permite recoger datos de los dispositivos y proporciona un conjunto de servicios, como análisis de datos, almacenamiento de datos, visualización de datos históricos y en tiempo real, e integración con los sistemas de administración. El concentrador Azure IoT recibe los datos desde los dispositivos de IoT. Los datos de los

dispositivos se pueden almacenar o transmitir para más procesamiento, análisis y administración. Los dispositivos se pueden comunicar con el concentrador de IoT mediante los protocolos HTTP, MQTT o AMQP. Microsoft ofrece un conjunto de SDK de dispositivos Azure IoT a través de GitHub. (National Instruments, 2019)

2.8.5 Ubidots

Ubidots es una plataforma que ofrece una experiencia segura y con API fáciles de usar a las que se accede a través de protocolos como: HTTP / MQTT / TCP / UDP para conectar de manera simple y segura hardware y entradas digitales a la nube que proporciona Ubidots, donde se puede analizar y formular en soluciones de mejora empresarial. Ubidots es útil para emprendedores, integradores de sistemas, instituciones educativas y permite conectar dispositivos de manera eficiente, administrar datos y visualizar con facilidad en un entorno virtual. (IoTONE, 2020)

La plataforma Ubidots for Education es creada para ofrecer a los entusiastas y estudiantes de IoT un lugar para construir, desarrollar, probar, aprender y explorar el futuro de las aplicaciones y soluciones conectadas a Internet. (IoTONE, 2020)

Esta plataforma nos ofrece:

- Bibliotecas de dispositivos que nos permite conectar rápidamente cualquier dispositivo habilitado para Internet a la plataforma Ubidots. (IoTONE, 2020)
- Paneles de control en tiempo real para para el análisis y control de dispositivos. Permite compartir datos a través de aplicaciones móviles o en la web. (IoTONE, 2020)
- Data Analytics, permite a los usuarios de Ubidots ver los resultados de su trabajo y facilita la toma de decisiones gracias a Ubidots Analytics Engine. Este motor de cálculo permite a los usuarios calcular funciones matemáticas, estadísticas y

lógicas programadas para entregar los datos y la información impulsando a mejora de eficiencia en una empresa. (IoTONE, 2020)

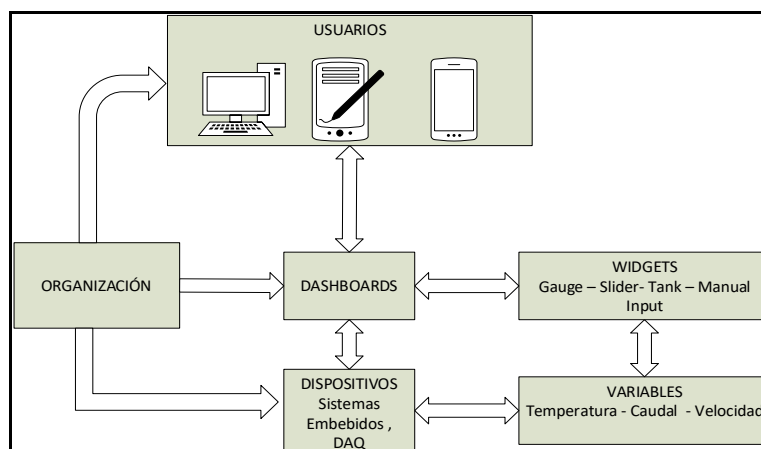
- Ubidots Events Engine permite a los usuarios mantenerse informados mediante la función de eventos y alarmas. Proporciona un sistema de envío de mensajes de texto, correos electrónicos, telegramas a personas y M2M para recibir notificaciones de datos y controles de automatización eficientes. (IoTONE, 2020)
- Acceso de usuario permite acceder a sus aplicaciones de IoT directamente a aquellos que necesitan con la autenticación y administración de usuarios de Ubidots. (IoTONE, 2020)

Dentro de Ubidots se cuenta con una estructura detallada de la siguiente manera:

- Una organización tiene dispositivos y dashboards dentro de ella, por último tendrá usuarios que dependiendo el rol que se les asigne tendrán acceso a visualizar estos dispositivos y Dashboards.
- Los dispositivos tienen variables donde se alojan los datos que captan o envían al exterior.
- Los dashboards permiten alojar estas variables dentro de widgets que reciben datos o envían datos.

Figura 14

Estructura Plataforma Ubidots



2.9 Wireshark

Wireshark es un analizador de paquetes de red. Un analizador de paquetes de red presenta los datos de los paquetes capturados con el mayor detalle posible.

Se podría pensar en un analizador de paquetes de red como un dispositivo de medición para examinar lo que está sucediendo dentro de un cable de red. (Wireshark, 2020)

Hace algunos años, las herramientas de este tipo eran costosas, propietarias o ambas cosas. Sin embargo, con la llegada de Wireshark, eso ha cambiado ya que se encuentra disponible totalmente gratis, es de código abierto y hoy en día es uno de los mejores analizadores de paquetes. (Wireshark, 2020)

Estas son algunos de los beneficios que proporciona Wireshark:

- Los administradores de red lo utilizan para solucionar problemas de red.
- Los ingenieros de seguridad de redes lo utilizan para examinar problemas de seguridad.

- Los ingenieros de control de calidad lo utilizan para verificar aplicaciones de red.
- Los desarrolladores lo utilizan para depurar implementaciones de protocolos.
- La gente lo usa para aprender los aspectos internos del protocolo de red.

Wireshark también puede ser útil en muchas otras situaciones.

2.9.1 Características

Las siguientes son algunas de las muchas características que proporciona Wireshark:

- Disponible para UNIX y Windows.
- Capturar de paquetes de datos en vivo desde una interfaz de red.
- Mostrar paquetes con información de protocolo muy detallada.
- Guardar los paquetes de datos capturados.
- Exportar algunos o todos los paquetes en varios formatos de archivo de captura.
- Filtrar paquetes según muchos criterios.

CAPÍTULO III

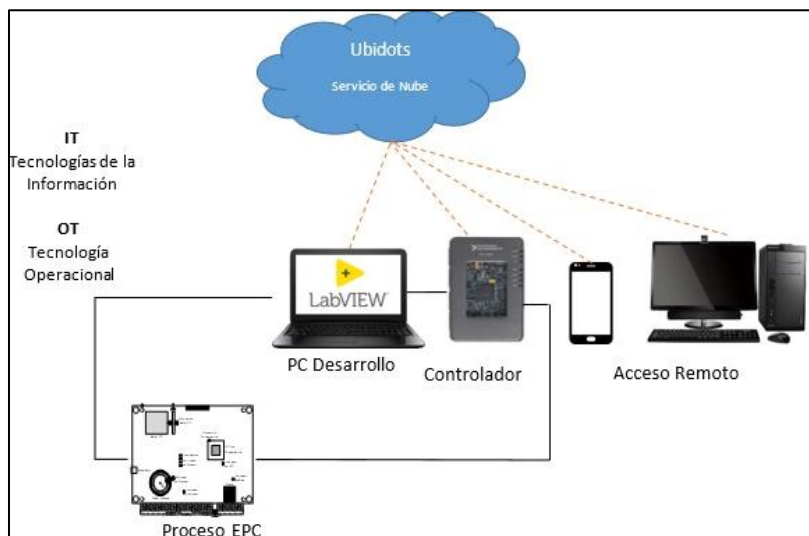
3. DESARROLLO E IMPLEMENTACIÓN

En este capítulo se describe el desarrollo del código e implementación del mismo en el software LabVIEW, que permite la conexión de la PC, Sistema Embebido NI myRIO y el EPC (Entrenador de Planta de Control) con la Plataforma de servicios en la Nube Ubidots bajo el marco tecnológico IoT usando el protocolo de comunicación HTTP y el formato de transmisión de datos JSON.

Al igual que la utilización de la Plataforma Ubidots, en el sentido de creación de una cuenta, dispositivos variables, dashboards, widgets, usuarios y organizaciones. En la *Figura 15* se muestra el esquema de integración de los componentes.

Figura 15

Esquema de integración de los componentes



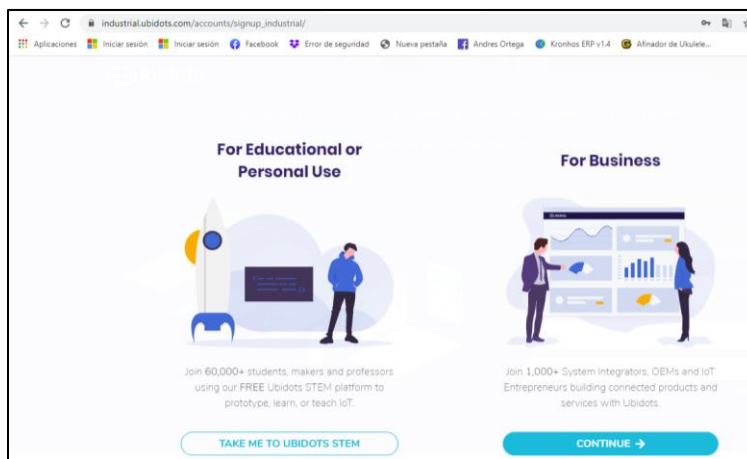
Nota. El gráfico representa el esquema de integración de los componentes de la prueba de concepto en la conexión con la Plataforma Ubidots. Adaptado de DemoScript SystemLink Cloud – Versión Publica (p.3), por S.Orellana , 2020, Datalighths.

3.1 Creación de una Cuenta en la Plataforma Ubidots

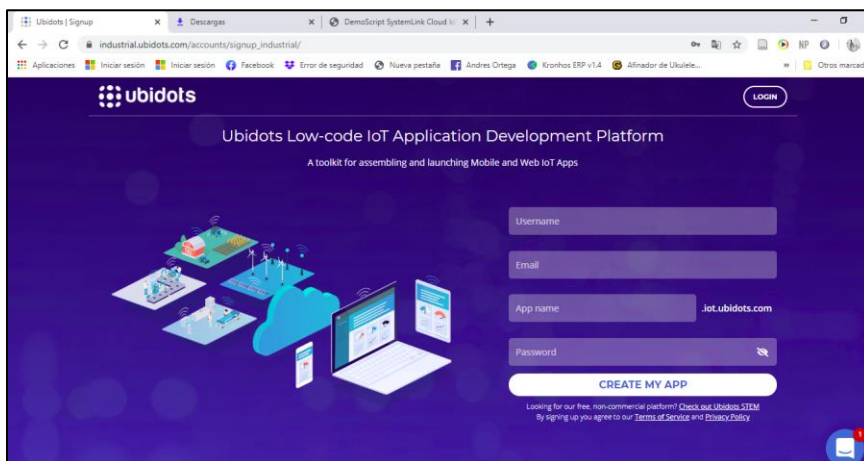
Para iniciar la cuenta en Ubidots, se ingresó a <https://ubidots.com> en la sección "SIGN UP". En la siguiente interfaz seleccionamos una cuenta For Businnes ya que así tendremos un poco más de opciones habilitadas. A continuación, damos clic en "CONTINUE" (Figura 16) y luego de llenar los datos correspondientes de manera personal (Figura 17), queda creada la cuenta en la Plataforma.

Figura 16

Interfaz sing up Ubidots

**Figura 17**

Interfaz creación de cuenta Ubidots



3.1.1 Visualización de Api Keys y Tokens de la cuenta

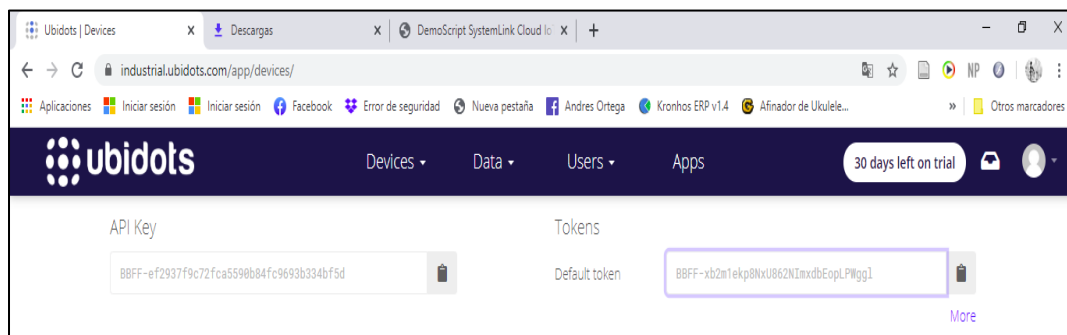
Una vez ingresado a la cuenta de Ubidots que se ha creado, nos dirigimos al icono de usuario y se ingresa en Api Credentials, se podrá visualizar el API Key y el Token. Como se observa en la *Figura 18*.

API Key: **BBFF-ef2937f9c72fca5590b84fc9693b334bf5d**

Default token: **BBFF-xb2m1ekp8NxU862NImxdbEopLPWgg1**

Figura 18

Interfaz Api credentials



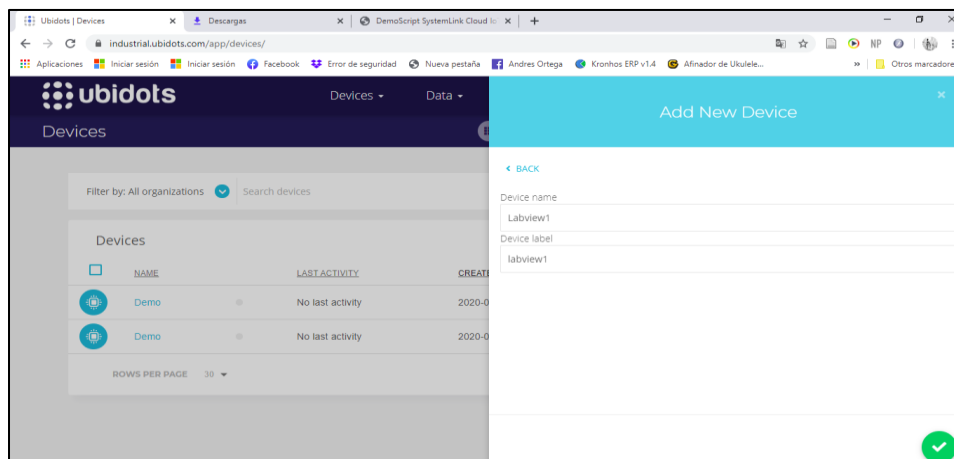
3.2 Creación de un Dispositivo

Se ingresa a la pestaña “DEVICES”, a continuación, damos clic en el símbolo “+” para agregar un dispositivo nuevo como se observa en la *Figura 19*. A continuación se establece un nombre para el dispositivo, en este caso el nombre es “Labview1” y el label es “labview1”. Al ingresar en el dispositivo creado se puede observar la imagen que aparece en la *Figura 20*.

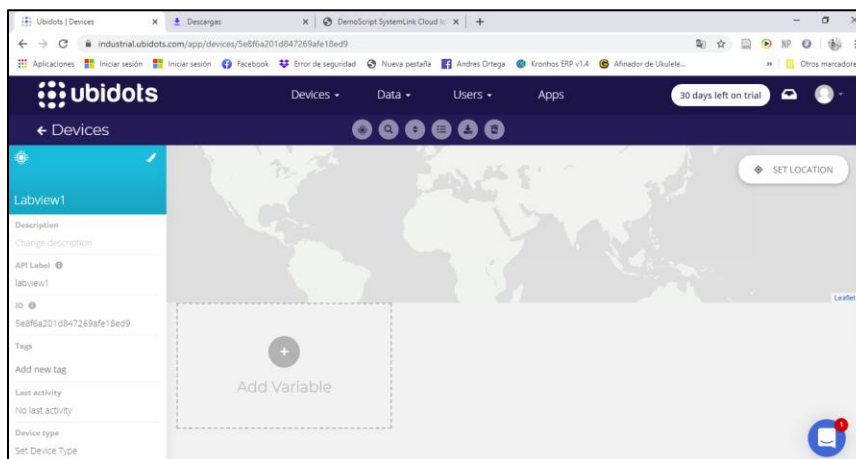
Al dar clic en “SET LOCATION” aparecerá la ubicación de nuestro dispositivo.

Figura 19

Creación de un dispositivo

**Figura 20**

Panel del dispositivo



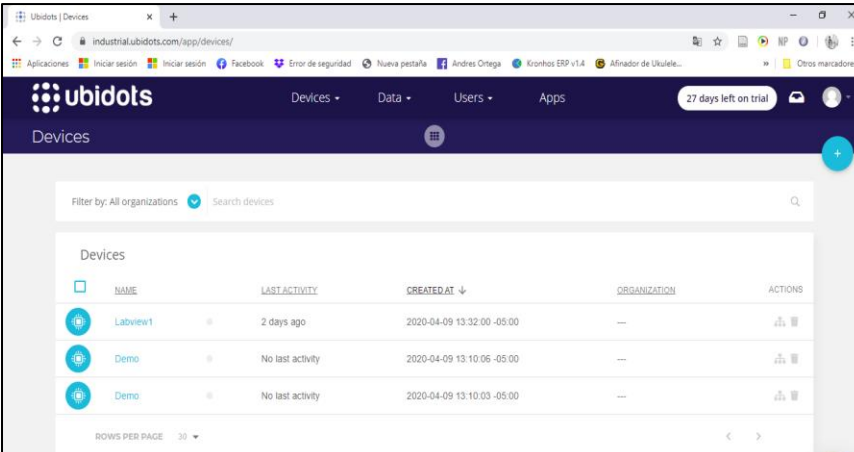
3.3 Creación de Variables en Ubidots

Para la creación de variables es necesario dirigirse a la pestaña “DEVICES” y seleccionar el dispositivo creado. En este caso Labview 1 como se observa en la *Figura 21*. Al seleccionar el dispositivo creado aparece la siguiente interfaz como muestra la

Figura 22, consecutivamente se debe seleccionar “+ Add Variable”. Posteriormente al seleccionar la opción “RAW” como se observa en la Figura 23, aparecerá el recuadro con la leyenda “New Variable” que se mira en la Figura 24, aquí se puede modificar el nombre de la variable y todos los parámetros necesarios. De esta manera se crean todas las variables para las aplicaciones que se deseen crear, esto se observa en la Figura 25.

Figura 21

Panel de dispositivos creados



The screenshot shows the Ubidots web interface for the 'Devices' section. The header includes the Ubidots logo, navigation tabs for 'Devices', 'Data', 'Users', and 'Apps', and a '27 days left on trial' notification. Below the header, there is a search bar and a table listing devices. The table has columns for 'NAME', 'LAST ACTIVITY', 'CREATED AT', 'ORGANIZATION', and 'ACTIONS'. Three devices are listed: 'Labview1', 'Demo', and another 'Demo'.

	NAME	LAST ACTIVITY	CREATED AT	ORGANIZATION	ACTIONS
	Labview1	2 days ago	2020-04-09 13:32:00 -05:00	---	
	Demo	No last activity	2020-04-09 13:10:06 -05:00	---	
	Demo	No last activity	2020-04-09 13:10:03 -05:00	---	

At the bottom of the table, there is a 'ROWS PER PAGE' dropdown set to 30 and navigation arrows.

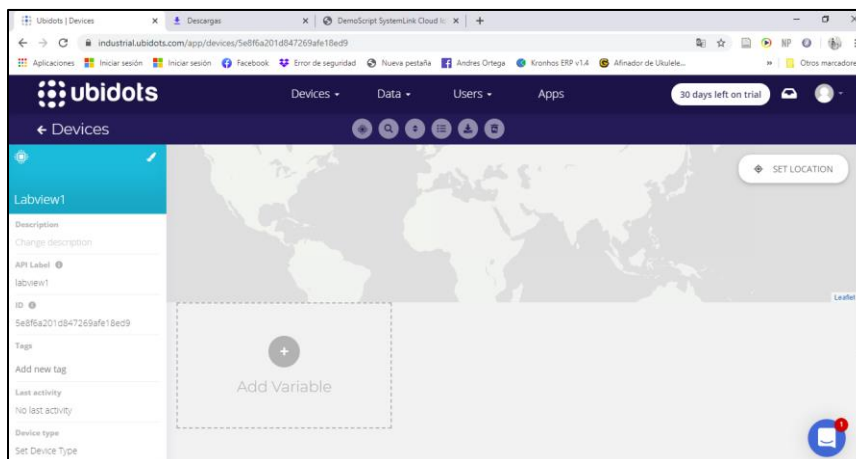
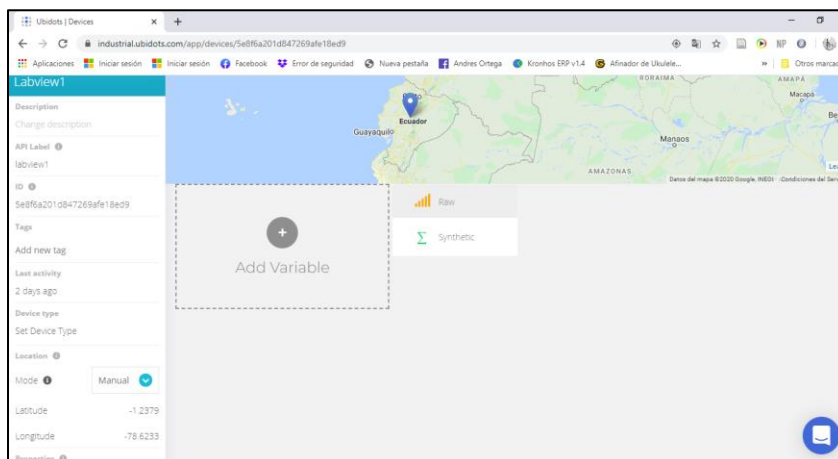
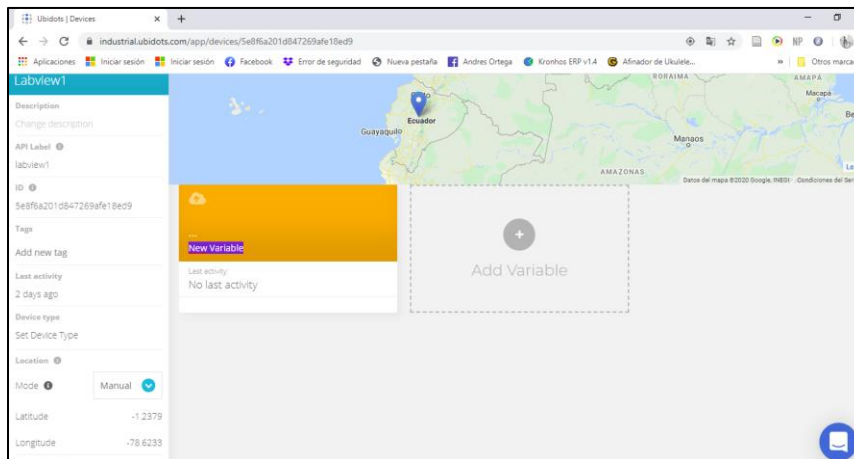
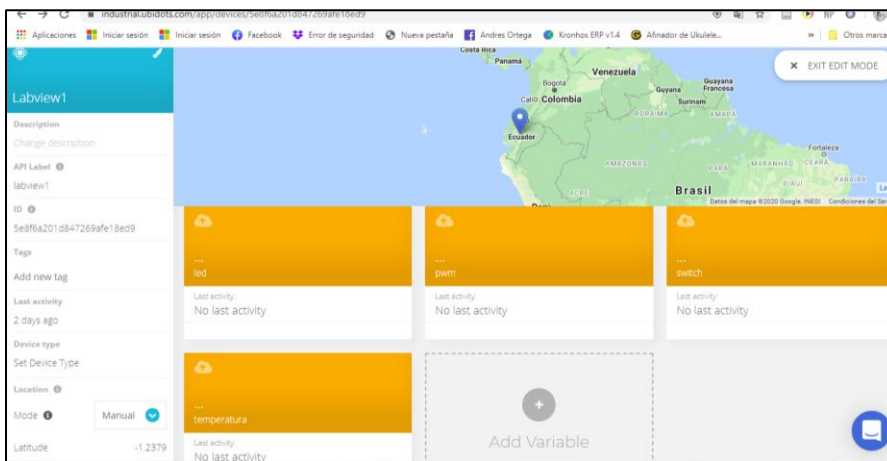
Figura 22*Creación variable***Figura 23***Añadir variable*

Figura 24

Modificar nombre de la variable

**Figura 25**

Panel de variables creadas



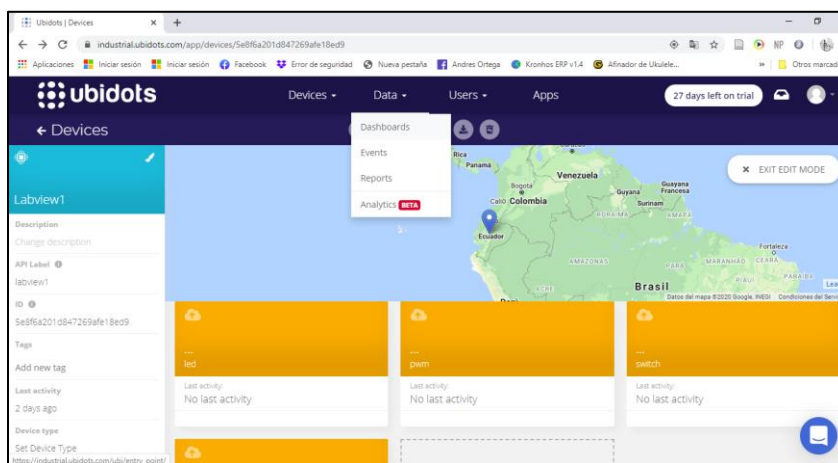
3.4 Creación de Dashboards y asignación de variables a los Widgets

3.4.1 Creación Dashboards

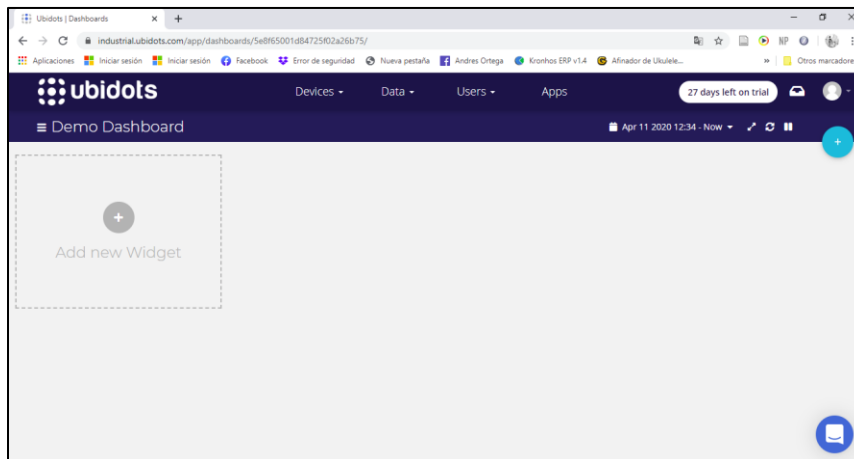
Al dirigirse a la sección “DATA” y dar clic en la opción “Dashboards”, tal y como se observa en la *Figura 26*.

Figura 26

Ingreso sección dashboards



Por defecto nos aparecerá un Demo Dashboard, al dar clic en el símbolo de menú que esta junto al nombre DEMO DASHBOARD (*Figura 27*).

Figura 27*Panel demo dashboard*

Aparecerá la siguiente interfaz (*Figura 28*), aquí se puede dar clic en el símbolo “+” para agregar un nuevo Dashboard. En esta sección se establece un nuevo nombre y se puede modificar ciertos parámetros como se observa en la interfaz de la *Figura 29*, al finalizar los cambios que se deseen se debe dar clic en el símbolo check.

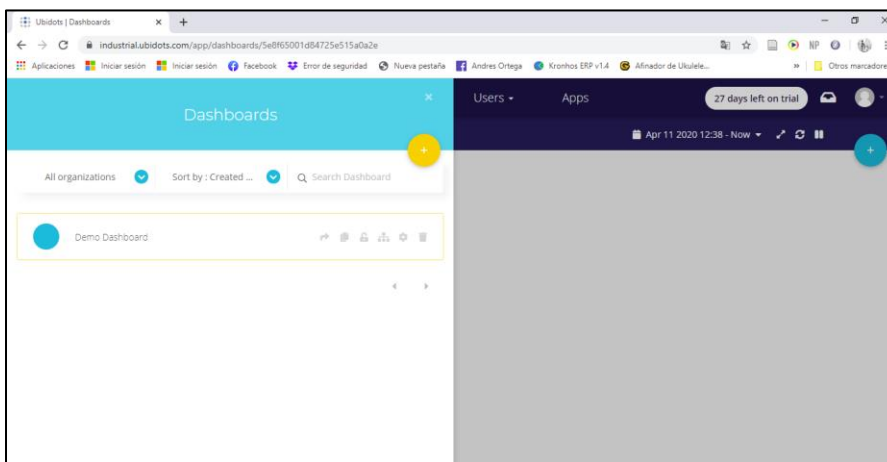
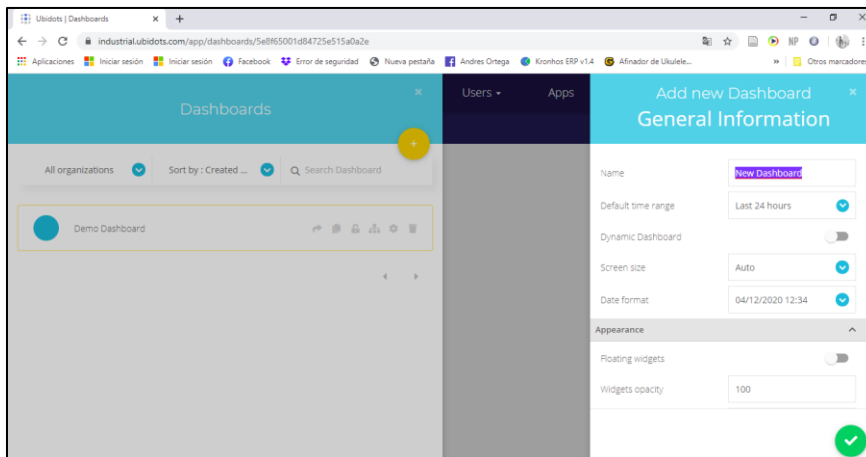
Figura 28*Agregar nuevo dashboard*

Figura 29

Interfaz propiedades del nuevo dashboard

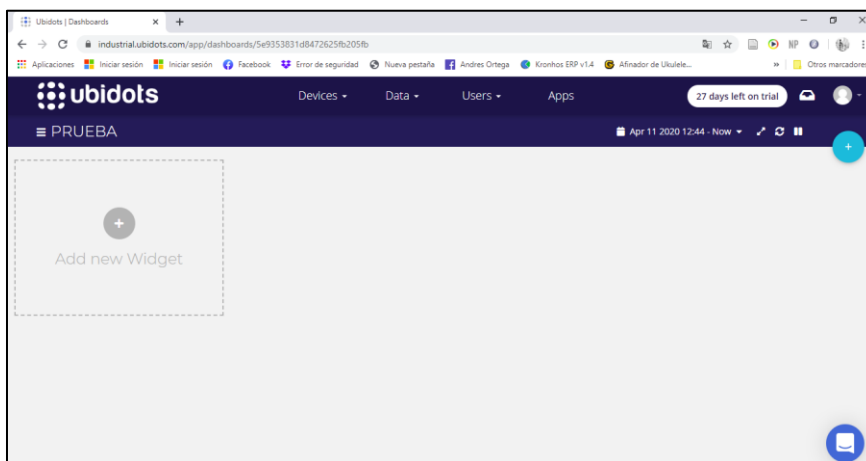


3.4.2 Asignación de Variables a Widgets

Al guardar el Dashboard, se presenta la siguiente interfaz, se puede añadir un Widget con el cual se visualiza las variables creadas, para agregar un nuevo widget se debe dar clic en “Add new Widget” o en el símbolo “+”, como se observa en la *Figura 30*.

Figura 30

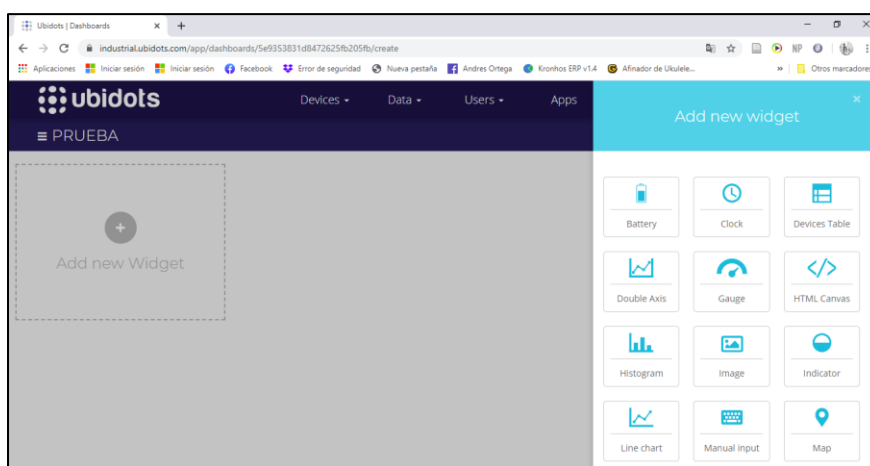
Interfaz dashboard creada



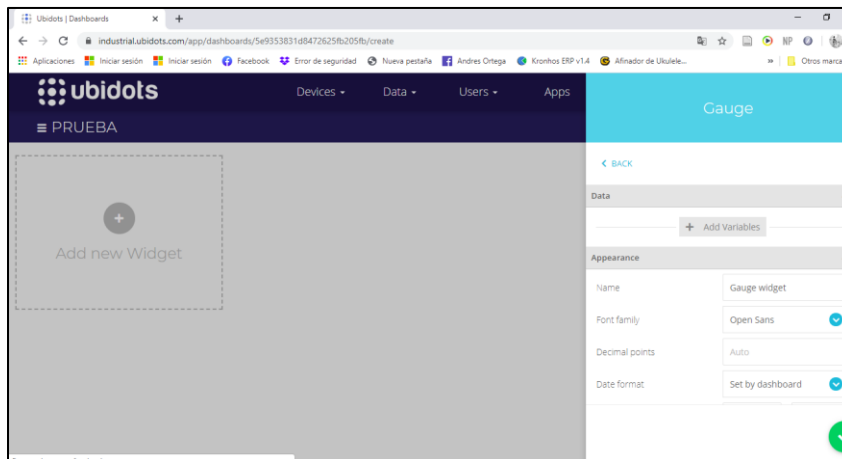
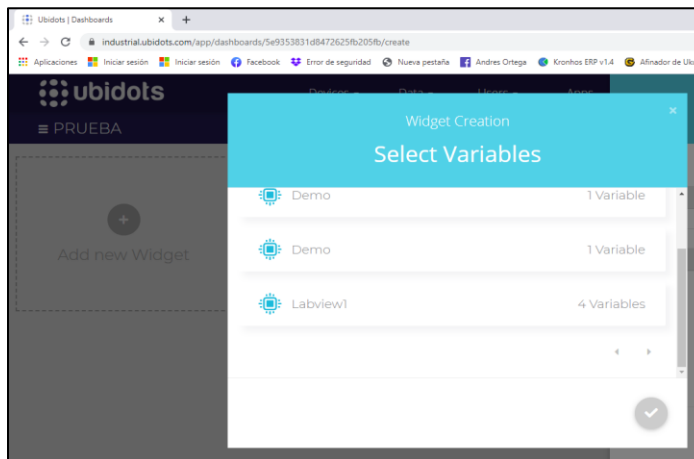
Se tendrá la siguiente interfaz (*Figura 31*) donde se escogen los widgets que se ajusten a la necesidad de la aplicación y que guarden relación con las variables creadas, es decir si se recibe un dato booleano lo más correcto sería visualizarlo en el widget “Indicator” y si se envía un dato booleano sería necesario usar el widget “Switch”.

Figura 31

Panel de widgets



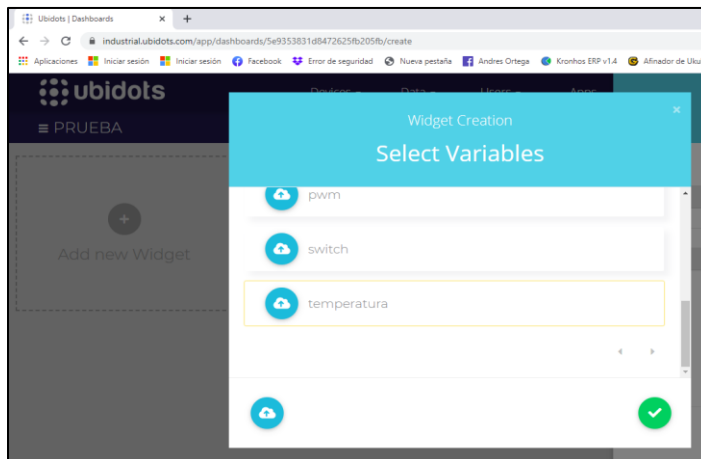
Seleccionar un Gauge en este caso como en la *Figura 32*, para la variable temperatura, lo más lógico y siguiendo el consejo anterior sería colocar un “termometer” pero no se encuentra disponible en esta versión de prueba. Dar clic en “+ Add Variables” y seleccionar el dispositivo creado donde se encuentran las variables creadas, en este caso el dispositivo es Labview1, tal y como aparece en la *Figura 33*.

Figura 32*Interfaz widget gauge***Figura 33***Asignación de variables al widget*

Se debe seleccionar el dispositivo y buscar la variable que se quiera asignar al widget como se mira en la *Figura 34*, en este caso se asigna la variable temperatura y damos clic en el símbolo check.

Figura 34

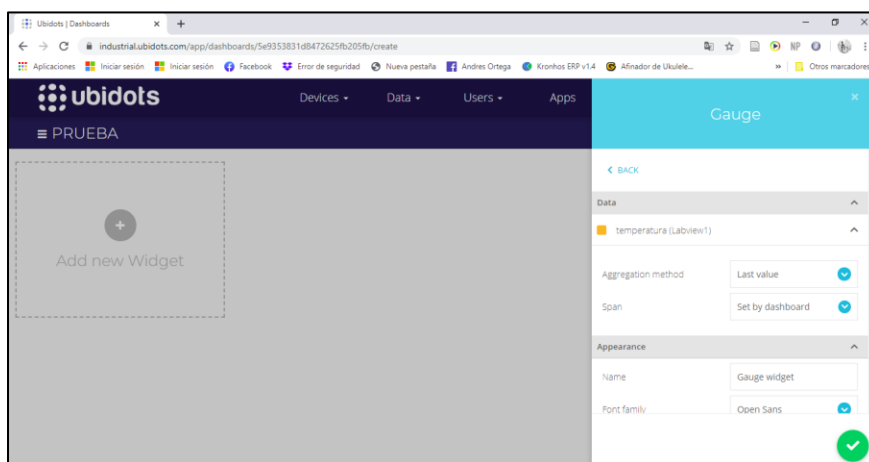
Selección de variables



De esta manera se tiene ya asignada la variable al widget, uno de los parámetros más importantes que se puede modificar es el rango y se lo puede hacer navegando más abajo en el apartado “Range Value” de la *Figura 35*. Cuando se tiene todo acorde a las necesidades de la aplicación daremos clic en el símbolo check.

Figura 35

Propiedades del gauge

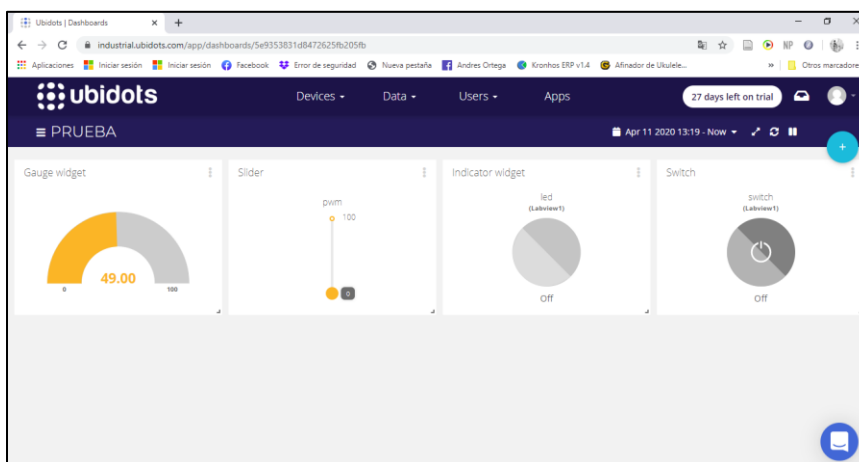


En la *Figura 36* se observa un conjunto de Widgets creados de ejemplo que son:

- Gauge: variable temperatura
- Indicator: variable led
- Switch: variable Switch
- Slider: variable RPM

Figura 36

Dashboard con diferentes widgets

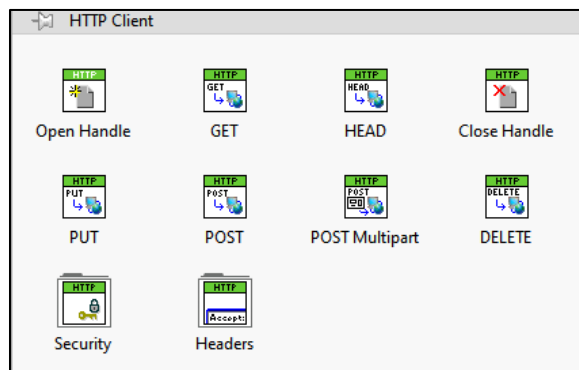


3.5 Desarrollo del código de conexión LabVIEW - Ubidots

Para establecer la comunicación entre el software LabVIEW alojado en el PC de Desarrollo y del Sistema Embebido NI myRIO con la plataforma Ubidots, se utiliza el protocolo de comunicación HTTP, este protocolo posee un conjunto de herramientas en la paleta de Funciones Data Communications > Protocols> HTTP Client como se muestra en la *Figura 37*.

Figura 37

Paleta de funciones http



Nota. El gráfico muestra la paleta de funciones HTTP Client del Software Labview.

Tomado de Software LabVIEW 2019, 2020.

Para poder desarrollar este código de comunicación se usan las funciones Open Handle, Headers, GET, POST y Close Handle. La plataforma Ubidots requiere un formato especial para poder enviar o recibir datos desde un dispositivo estas funciones son GET para recibir datos desde la Plataforma y POST para enviar datos hacia la Plataforma. La estructura para el comando POST se muestra en la *Figura 38*.

Figura 38

Estructura post

```
POST /api/v1.6/devices/{DEVICE_LABEL} 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>
```


Nota. El gráfico muestra la estructura para poder ejecutar el comando POST y enviar datos a la Plataforma de Ubidots. Tomado de Ubidots API Docs por Ubidots , 2020, Ubidots(<https://ubidots.com/docs/hw/#send-data>).

La estructura para el comando GET se muestra en la *Figura 39*.

Figura 39

Estructura get

```
GET /api/v1.6/devices/{DEVICE_LABEL}/{VARIABLE_LABEL}/values 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><CR><LN>
```

Nota. El gráfico muestra la estructura para poder ejecutar el comando GET y recibir datos desde Plataforma de Ubidots. Tomado de Ubidots API Docs por Ubidots , 2020, Ubidots(<https://ubidots.com/docs/hw/#retrieve-data>).

Se usó esta Paleta de Funciones HTTP Client que se muestra en la *Figura 37* como base para el desarrollo de un conjunto de Sub VI's que permiten la conexión con la plataforma Ubidots.

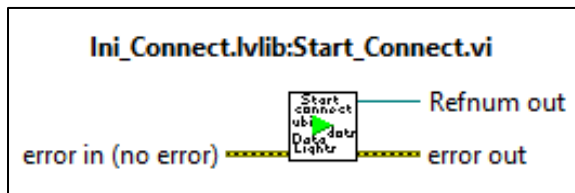
3.5.1 Abrir y Cerrar conexión

a) Abrir Conexión(Start_Connect.vi)

Como se observa en la *Figura 40* tiene una entrada y 2 salidas que son error in, HTTP network connection, out (Refnum out), error out.

Figura 40

Sub vi inicio de conexión

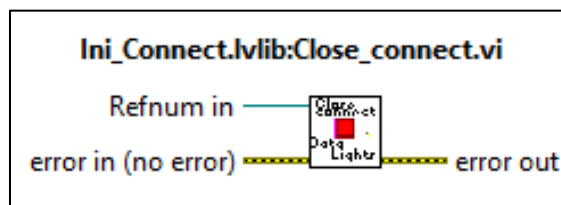


b) Cerrar Conexión(Close_connect.vi)

Como se observa en la *Figura 41* se tiene solamente 2 entradas y una salida que son HTTP network connection in (Refnum in), error in y error out respectivamente.

Figura 41

Sub vi cierre de conexión



Estos Sub Vi's están alojados dentro de una Biblioteca **In Connect.lvlib**.

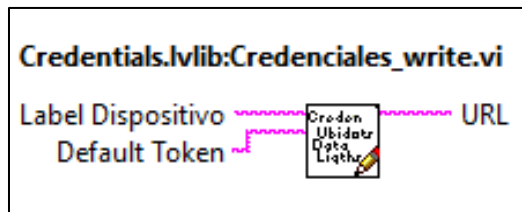
3.5.2 Credenciales de Escritura y Lectura

a) Credenciales de Escritura (Credenciales_write.vi)

Como se observa en la *Figura 42* se tiene 2 entradas y una salida, las entradas son el Label o etiqueta del Dispositivo y el Default Token, en ese orden, la salida es el conjunto de estas credenciales conocidas como URL.

Figura 42

Sub vi credenciales de escritura

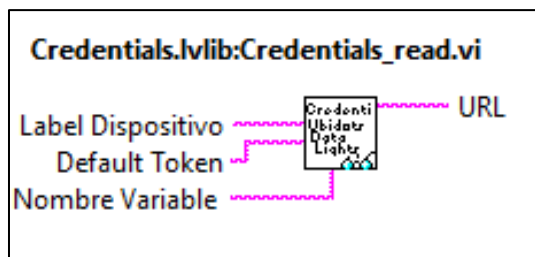


b) Credenciales de lectura(Credenciales_read.vi)

Como se observa en la *Figura 43* se tiene 3 entradas y 1 salida, las entradas son el Label o etiqueta del Dispositivo, el Default Token Y nombre de la variable, en ese orden, y la salida es el conjunto de estas credenciales conocidas como URL.

Figura 43

Sub vi credenciales de lectura



Estos Sub Vi's están alojados dentro de una Biblioteca **Credentials.lvlib**.

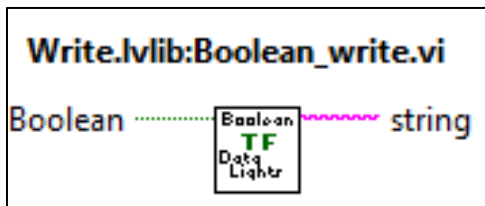
3.5.3 Conversión de variables numéricas, booleanas y Strings

a) Datos booleanos (Boolean_write.vi)

Como se observa en la *Figura 44* tenemos 1 entrada y 1 salida, la entrada hace referencia al controlador booleano y la salida al valor convertido.

Figura 44

Sub vi conversión booleano a string

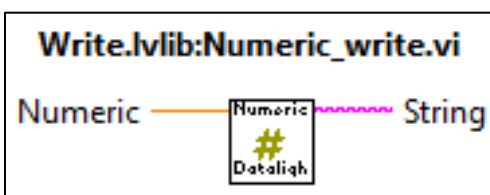


b) Datos Numéricos(Numeric_write.vi)

Como se observa en la *Figura 45* se tiene 1 entrada y 1 salida, el tipo de datos numérico establecido por defecto es DBL, ya que la mayoría de datos y sensores que se pueden llegar a enviar son de este tipo, pero si se requieren enviar datos numéricos de otro tipo no existe ningún problema, además del punto de coerción que aparecerá, la entrada es el control numérico que se quiera enviar y la salida el valor convertido.

Figura 45

Sub vi conversión dato numérico a string



c) Datos String

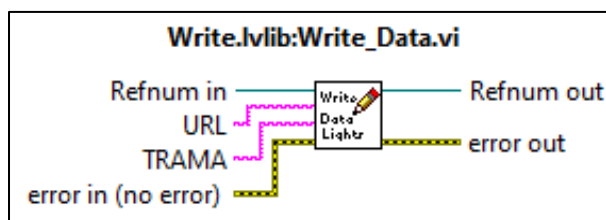
Este tipo de datos no necesitan ningún tratamiento así que no existe un Sub Vi para los mismos.

3.5.4 Escritura de los datos (Write_Data.vi)

Como se observa en la *Figura 46* se tiene 4 entradas y 2 salidas, las entradas son HTTP network connection in (Refnum in), credenciales de escritura (URL), trama de datos (Trama) y el error in, en ese orden, y las salidas son HTTP network connection out (Refnum out) y el error out.

Figura 46

Sub vi escribir datos a Ubidots



Estos SubVis están alojados dentro de una Biblioteca **Write.lvlib**.

Para la **trama de datos** no existe un Sub Vi como tal para entramar los datos y debido a errores con datos booleanos no se usa la función JSON que viene pre construida en LabVIEW pero necesariamente se debe usar este formato de datos.

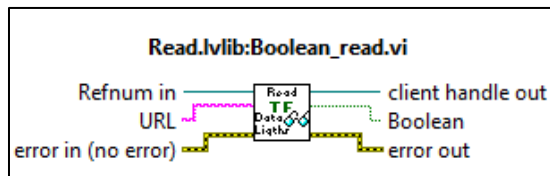
3.5.5 Lectura de variables booleanas, numéricas y strings

a) Lectura variable booleana (Boolean_read.vi)

Tiene 3 entradas y 3 salidas como se observa en la *Figura 47*, las entradas son HTTP network connection in, credenciales de lectura (URL), HTTP error in, en ese orden y las salidas son, dato booleano leído, HTTP error out y HTTP network connection out en ese orden.

Figura 47

Sub vi lectura de dato booleano desde Ubidots

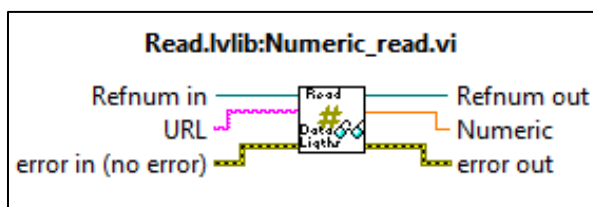


b) Lectura variable numérica (Numeric_read.vi)

Tiene 3 entradas y 3 salidas como se observa en la Figura 48, las entradas son, HTTP network connection in, credenciales de lectura (URL), HTTP error in, en ese orden y las salidas son, dato numérico leído, HTTP network connection out, HTTP error out en ese orden.

Figura 48

Sub vi lectura de datos numéricos desde Ubidots



Estos SubVi's están alojados dentro de una Biblioteca **Read.lvlib**.

3.6 Programa PING desde PC a la Plataforma Ubidots

Antes de comenzar con el envío y recepción de datos, se debe comprobar la conexión de la PC a la plataforma de Ubidots y a la red de Internet en general, para esto se realizó un programa para ejecutar un comando PING desde la PC hacia la

Plataforma de Ubidots. Para esto se utiliza la función SYSTEM EXCEC de la Paleta de Funciones Connectivity.

Se deben crear los siguientes Controles e Indicadores:

Controles

- String: DIRECCION IP
- Numérico: Número de Paquetes
- Numérico: Timeout
- Control de Error

Indicadores

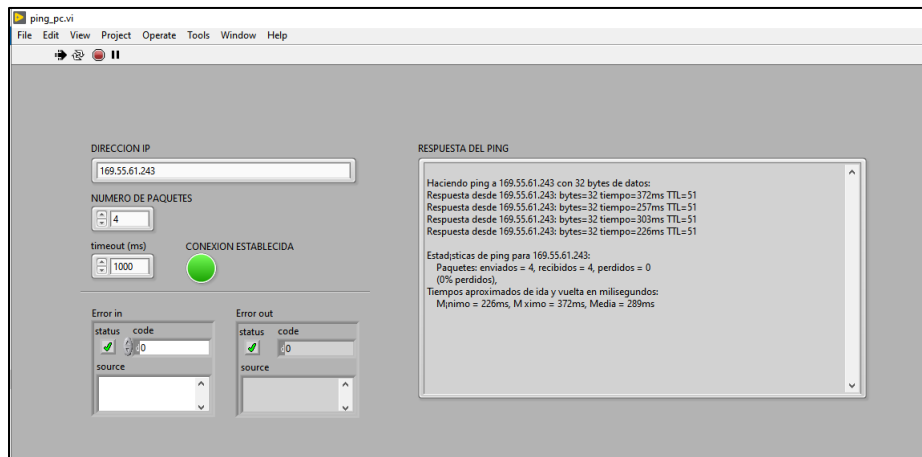
- String: RESPUESTA DEL PING
- Boleano: CONEXIÓN ESTABLECIDA
- Indicador de Error

La dirección IP de la plataforma Ubidots es 169.55.61.243, esa dirección se introduce en el Control String “DIRECCION IP” y en el control numérico “Numero de Paquetes” se coloca 4, es el número de paquetes recomendado, el valor del Control Timeout es de 1000 ms.

El programa ejecutará el comando PING a la dirección Especificada y mostrará la respuesta en el Indicador RESPUESTA DEL PING, además que se encenderá el Indicador Boleano CONEXIÓN ESTABLECIDA como se observa en la *Figura 49*.

Figura 49

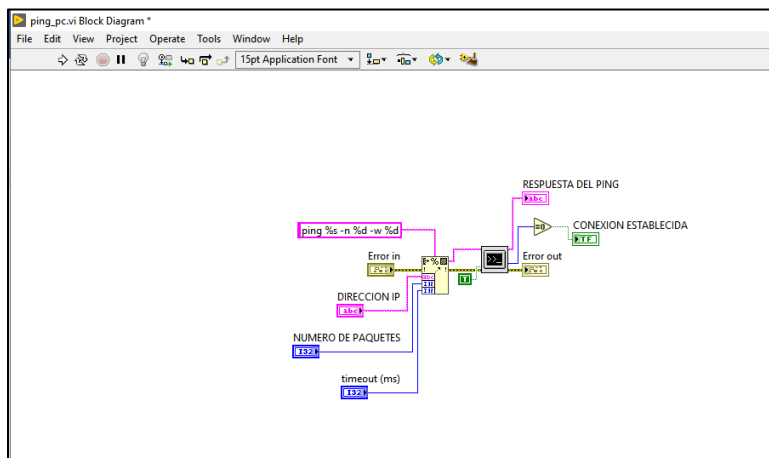
Panel frontal programa ping desde pc



En la *Figura 50* observamos el Diagrama de Bloque del programa, donde se usa la ya mencionada función SYSTEM EXECUT.

Figura 50

Diagrama de bloques programa ping pc a Ubidots



3.7 Programa que Escribe Datos a la Plataforma de Ubidots

Se desarrollará un programa que escriba 2 variables, una numérica y otra booleana desde el PC con LabVIEW hacia un Dashboard en la Plataforma Ubidots.

En la plataforma de Ubidots se debe crear un Dispositivo con el Nombre “**Labview**” y 2 variables: **temperatura** y otra con el nombre **led**, como se indicó en la sección 3.2 y 3.3 respectivamente, adicionalmente se crea un Dashboard y se asignan las variables a los widgets como en la sección 3.4.

Widgets Creados:

- Gauge: Asignar variable temperatura
- Indicator: Asignar variable led

También se deben crear los siguientes controles en la Interfaz de LabVIEW:

Controles

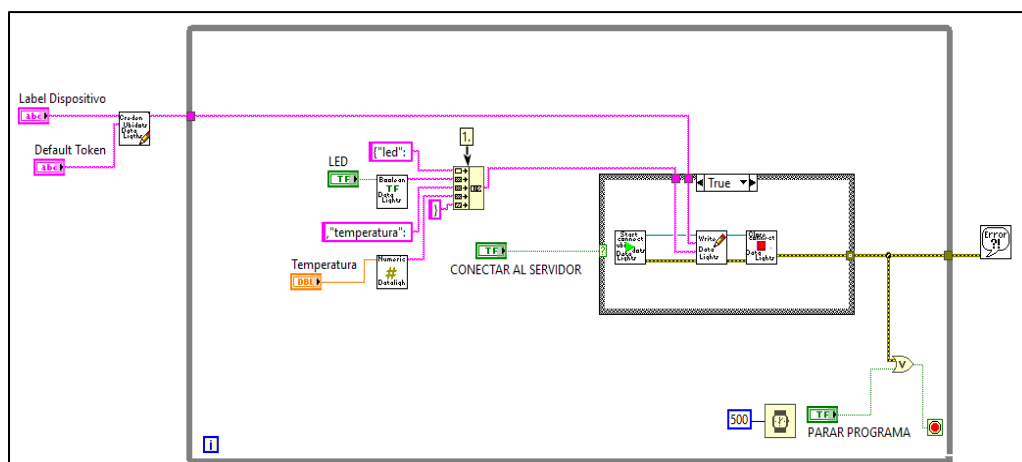
- String: Label Dispositivo
- String: Default Token
- Numérico: Temperatura
- Boleano: LED
- Boleano: Parar Programa
- Boleano: CONECTAR AL SERVIDOR

Es necesario hacer uso de las bibliotecas **Credentials.lvlib**, **Write.lvlib**, **Ini_Connect.lvlib**. Estas bibliotecas contienen los siguientes Sub VI's: **Start_connect.vi**, **Close_connect.vi**, **Credenciales_write.vi**, **Write_Data.vi**, **Boolean_write.vi** y **Numeric_write.vi**

En la *Figura 51* se observa el Diagrama de Bloques del Programa.

Figura 51

Diagrama de bloques programa enviar datos desde pc a Ubidots



Para iniciar la ejecución del programa abra que ingresar el Label Dispositivo y Default Token, datos disponibles en la Creación del Dispositivo en la sección 3.2. Esta escritura se activará solamente cuando este activado el control Booleano **“CONECTAR AL SERVIDOR”**.

Se enviará a través del control numérico valores de temperatura de 0 a 100 ° y se escriben en el Dashboard, por otro lado, se tiene un control booleano que prende o apaga un led en el Dashboard.

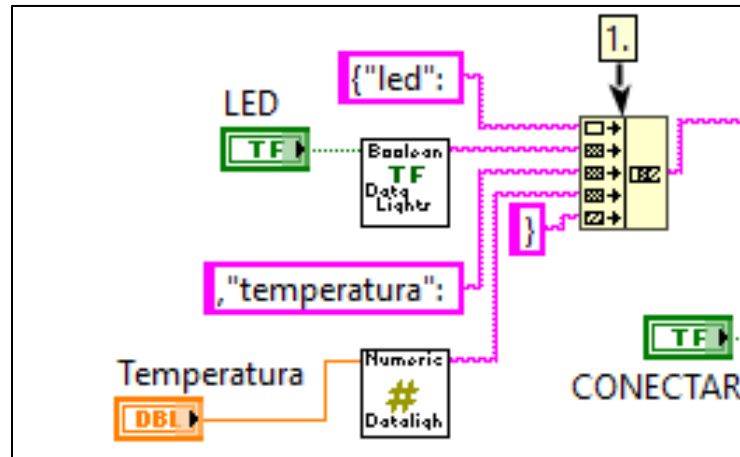
Entramado de datos (formato Json): Se entrama cada uno de los datos bajo el formato JSON de la siguiente manera, como se muestra en la *Figura 52*.

```
{"variable1:", "variable2:", "variable3:", "variableN:"}
```

No se utiliza la función JSON directamente por inconveniente en los datos boolean.

Figura 52

Entramado de datos



En este caso la trama que se armó es la siguiente:

`{"led: estado del led","temperatura: número generado"}`

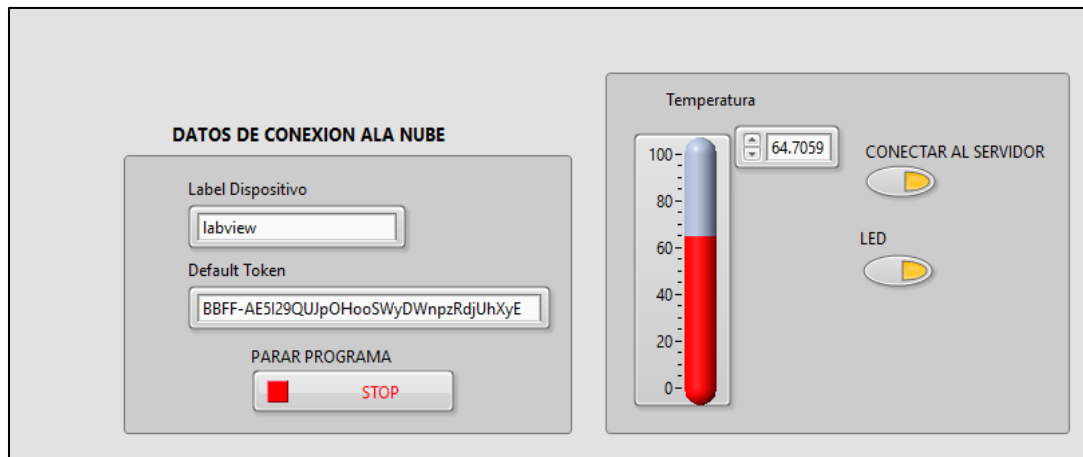
Un ejemplo de trama enviada sería:

`{"led: 1","temperatura: 45.7"}`

En la *Figura 53* se observa el panel frontal del programa.

Figura 53

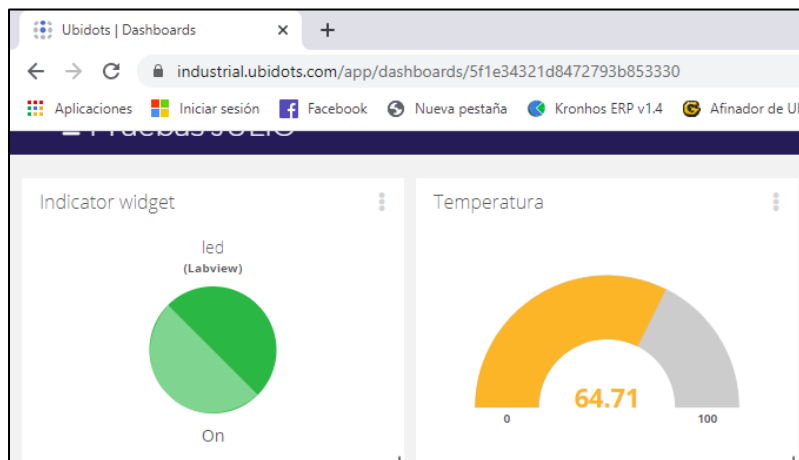
Panel frontal del programa enviar datos desde pc a Ubidots



De esta forma se envían datos desde el PC hacia el Dashboard creado en la Plataforma Ubidots como se observa en la *Figura 54*.

Figura 54

Dashboard plataforma Ubidots



3.8 Programa que Recibe Datos desde la Plataforma de Ubidots

Se desarrolló un programa que reciba 2 variables, una numérica y otra booleana desde un Dashboard en la Plataforma Ubidots hacia el PC con LabVIEW.

En la plataforma de Ubidots se creó un Dispositivo con el Nombre “**Labview**” y 2 variables una es **RPM** y otra con el nombre **Switch**, como se mostró en la sección 3.2 y 3.3 respectivamente, adicionalmente un Dashboard y se asignan las variables a los widgets como en la sección 3.4.

Widgets Creados:

- Slider: Asignar variable RPM
- Switch: Asignar variable Switch

Adicionalmente se crearon los siguientes controles e indicadores en la Interfaz de LabVIEW:

Controles

- String: Label Dispositivo
- String: Default Token
- Nombre Variable 1
- Nombre Variable 2
- Boleano: Parar Programa
- Boleano: CONECTAR AL SERVIDOR

Indicadores

- Numérico Gauge: RPM
- Boleano: Switch

Se necesitan 3 bibliotecas **Credentials.lvlib**, **Read.lvlib**, **Ini_Connect.lvlib**.

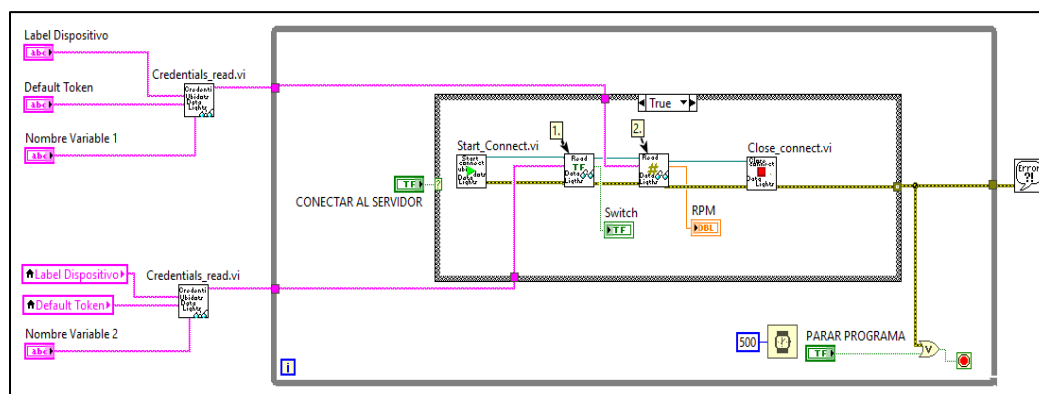
Estas bibliotecas contienen los Sub VI's: **Start_connect.vi**, **Close_connect.vi**

Credenciales_read.vi, **Boolean_read.vi** y **Numeric_read.vi**

En la *Figura 55* se observa el Diagrama de Bloques.

Figura 55

Diagrama de bloques programa recibir datos



Para iniciar la ejecución del programa se ingresó el Label Dispositivo y Default Token, datos disponibles en la Creación del Dispositivo en la sección 3.2 y el nombre de las variables creadas en la Plataforma de Ubidots. Esta escritura se activará solamente cuando este activado el control Booleano **“CONECTAR AL SERVIDOR”**.

Se lee el valor de una slider conocido como Rpm y se lee el estado de un Switch visualizado en un led en el panel frontal, como se observa en la *Figura 56*.

Figura 56

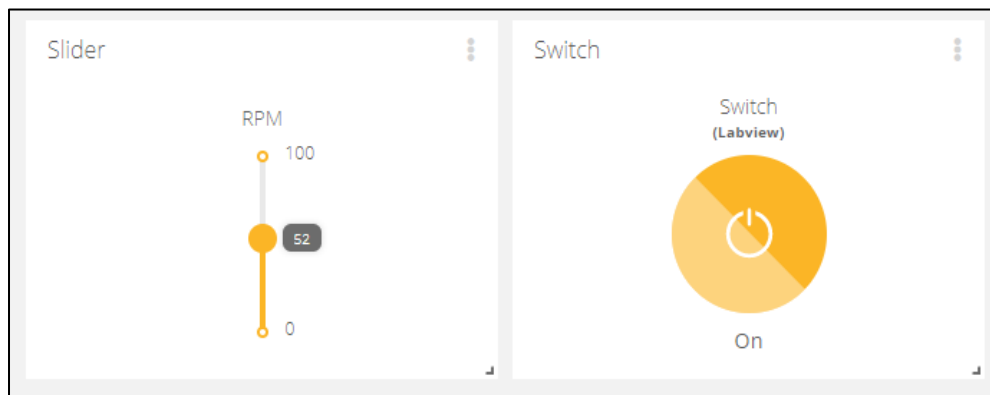
Panel frontal programa recibir datos



De esta manera se envían datos desde el Dashboard de Ubidots como se puede observar en la *Figura 57*.

Figura 57

Dashboard Ubidots

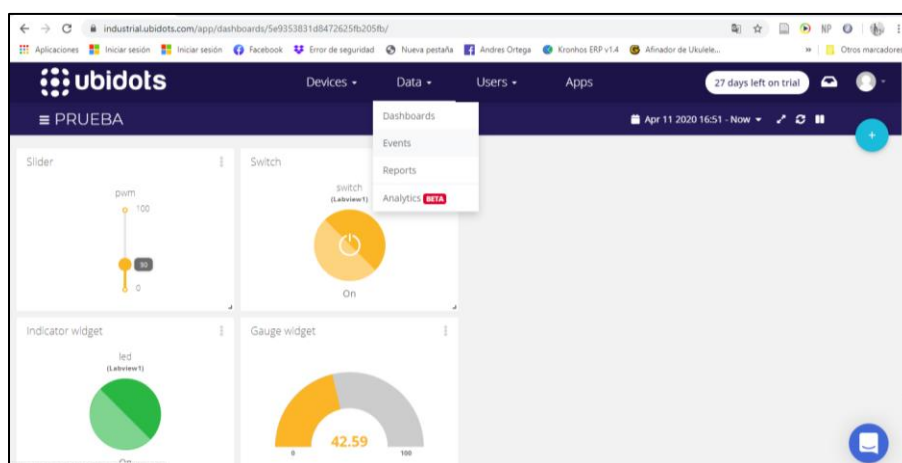


3.9 Eventos dentro de Ubidots

La plataforma Ubidots nos proporciona la función de programar eventos cuando algunas de las variables entren en una condición específica. Para crear un evento nuevo se debe ir a la sección “DATA” y seleccionar la pestaña “Events”, como muestra la *Figura 58*.

Figura 58

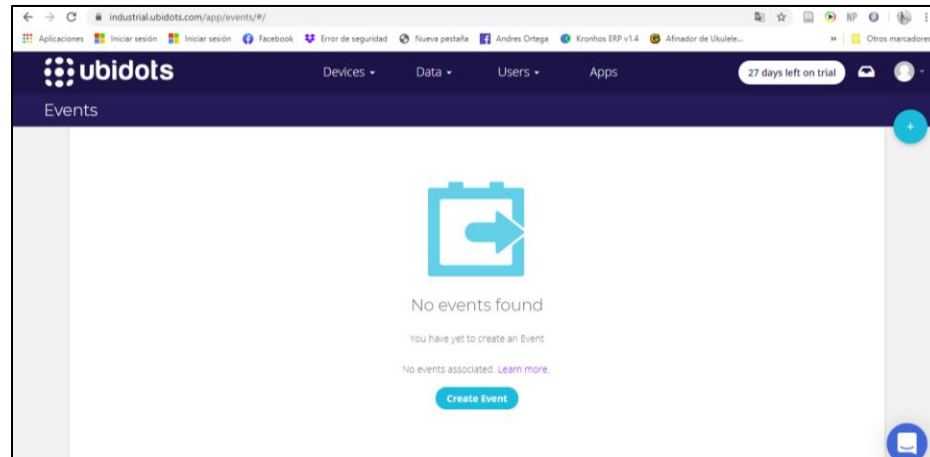
Creación eventos



Se debe tener la siguiente interfaz de la *Figura 59* y daremos clic en “Create Event”

Figura 59

Interfaz crear evento

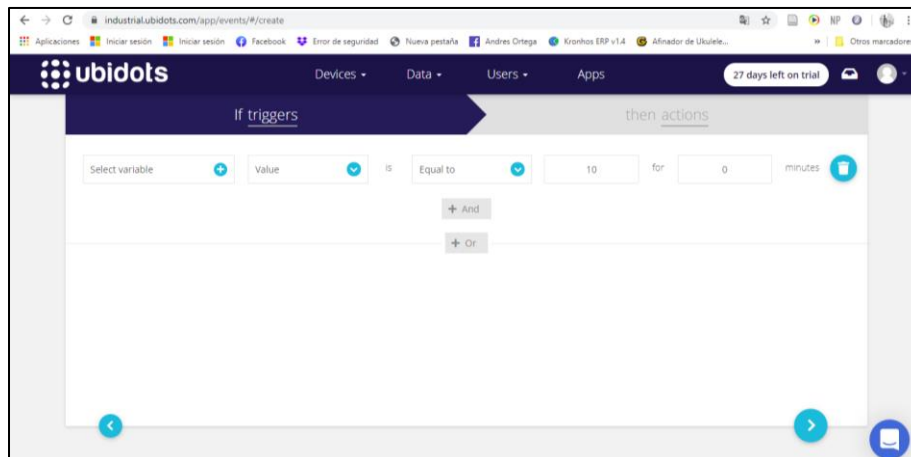


Se obtiene la siguiente interfaz que muestra la *Figura 60* donde, se usa la estructura IF y THEN para programar el evento requerido los datos requeridos son:

- Variable
- Valor
- Condición
- Tiempo

Figura 60

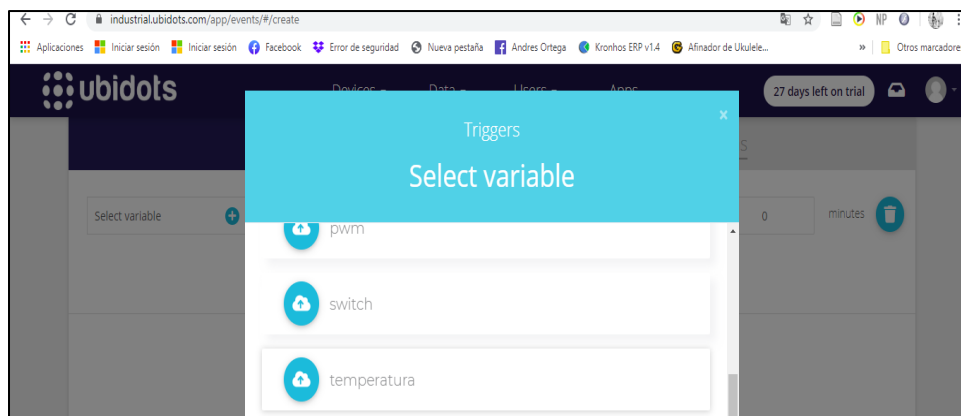
Interfaz de condición de evento



En este caso se debe seleccionar la variable temperatura, como muestra la *Figura 61*.

Figura 61

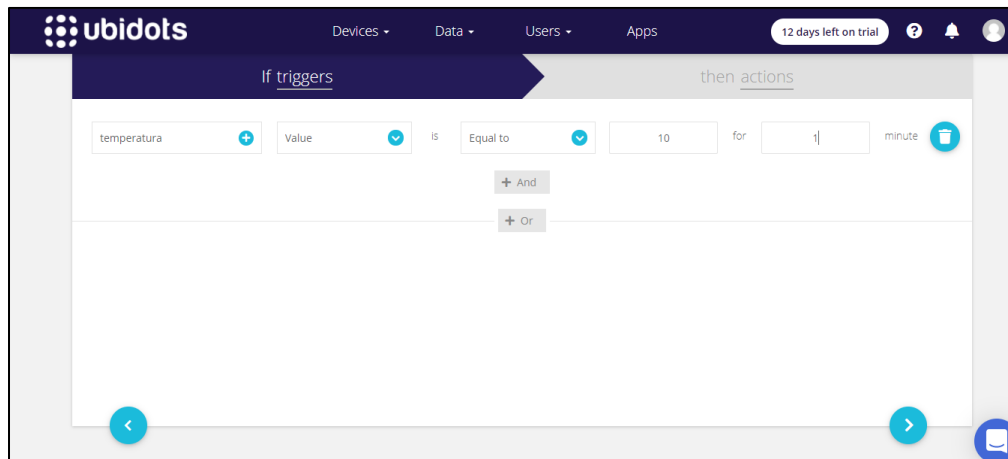
Selección de variable para el evento



La condición es que se producirá una acción cuando la temperatura sea igual a 10°C, como se observa en la *Figura 62*.

Figura 62

Ingreso de condiciones para evento



Se debe seleccionar “then actions” y dar clic en “Add new action”, como se observa en la *Figura 63*.

Figura 63

Ingresar una nueva acción

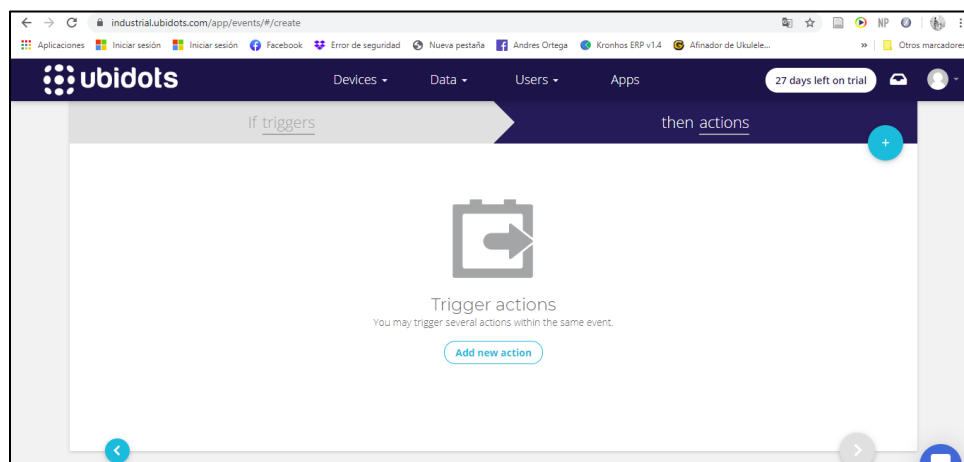
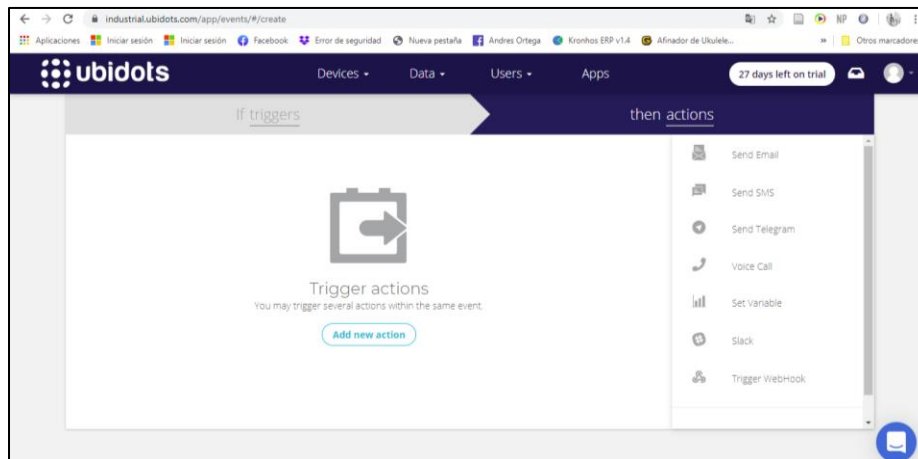
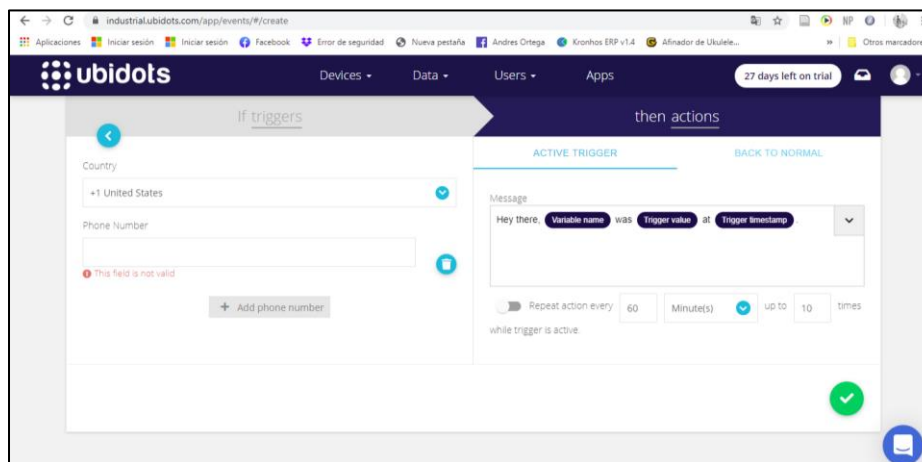


Figura 64*Acciones para el evento*

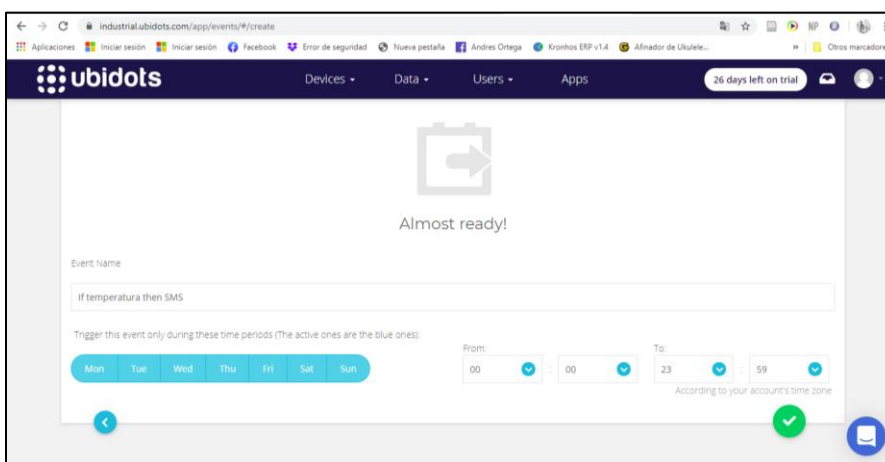
Por ejemplo, al enviar un SMS de texto como se observa en la *Figura 64*, se observan las siguientes configuraciones; la que se repite en todas es la frecuencia con la que se repetirá la acción que dependerá de nuestra necesidad. Después de tener todo listo se selecciona el símbolo check, como se mira en la *Figura 65*.

Figura 65*Configuración sms*

Dar clic en la flecha que apunta a la derecha, seleccionar los días y el intervalo de tiempo en el día que requiera que el evento se produzca, como se observa en la *Figura 66*. Y dar clic en el símbolo check.

Figura 66

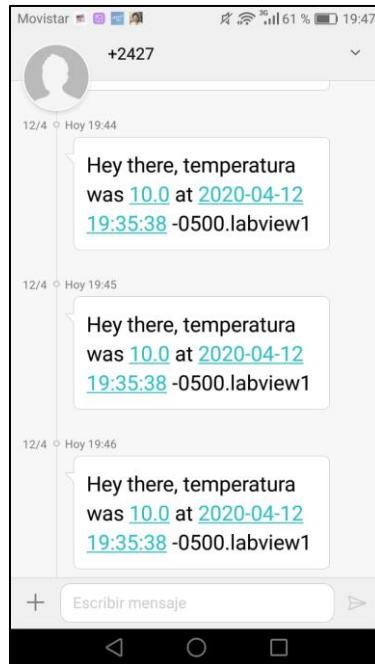
Días y frecuencia que se produce la acción



En este caso cuando la temperatura sea igual a 10°C por minuto se producirá el evento, y llegara un SMS al número celular ingresado, como se evidencia en la *Figura 67*.

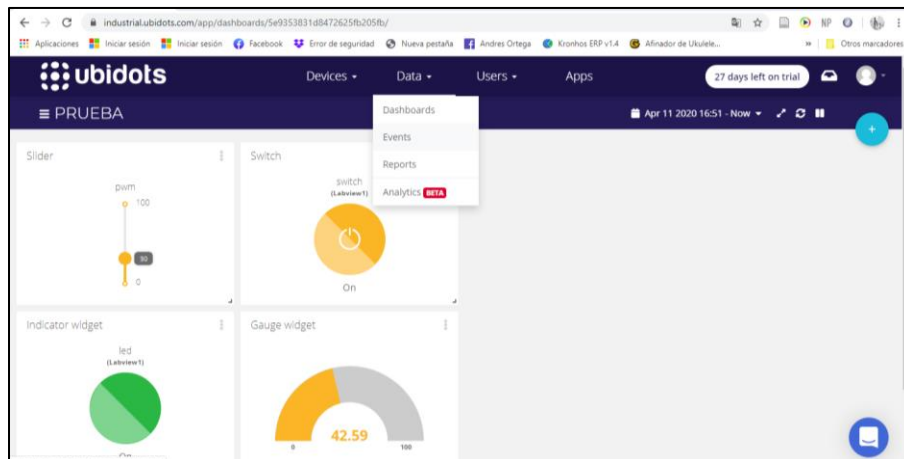
Figura 67

Sms de texto que llega al celular

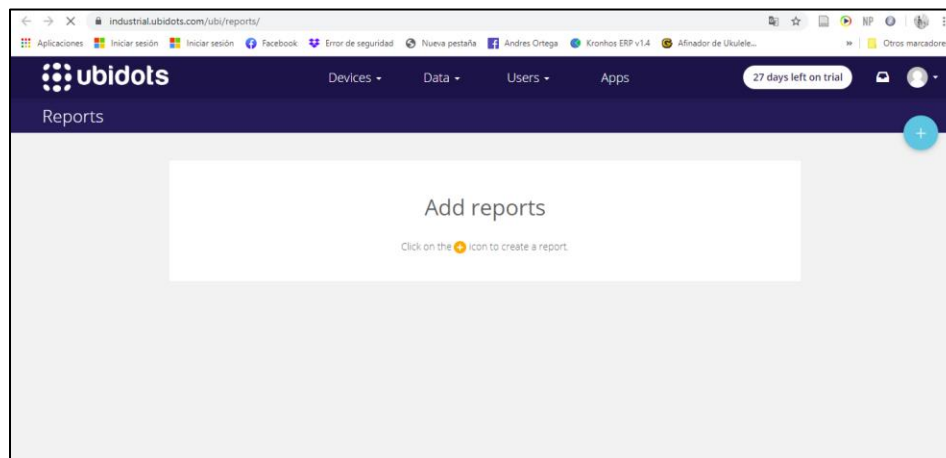


3.10 Creación de Reportes en Ubidots

Para crear un reporte nuevo se debe ir a la sección "DATA" y seleccionar la pestaña "Reports", como se observa en la *Figura 68*.

Figura 68*Creación reporte*

Se obtiene la siguiente interfaz como muestra la *Figura 69* y se procede a dar clic en “Add reports”

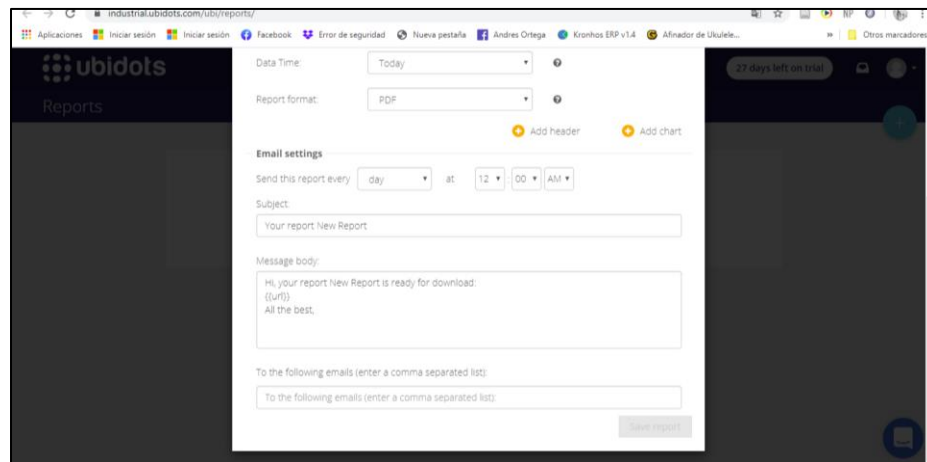
Figura 69*Adición de reporte*

En la siguiente interfaz de la *Figura 70*, se deben seleccionar los parámetros de configuración entre los más importantes se tienen:

- Formato del reporte (Excel, PDF)
- Frecuencia de envío del reporte
- Email donde llegara el link de descarga del reporte

Figura 70

Configuración de reporte



The screenshot shows the 'Reports' configuration page in the Ubidots interface. The page is titled 'Reports' and has a dark sidebar on the left. The main content area is white and contains the following settings:

- Data Time:** A dropdown menu set to 'Today'.
- Report format:** A dropdown menu set to 'PDF'.
- Buttons:** Two yellow buttons labeled '+ Add header' and '+ Add chart'.
- Email settings:**
 - Send this report every:** A dropdown menu set to 'day'.
 - at:** A dropdown menu set to '12'.
 - AM/PM:** A dropdown menu set to 'AM'.
 - Subject:** A text input field containing 'Your report New Report'.
 - Message body:** A text area containing 'Hi, your report New Report is ready for download: {{url}} All the best.'.
 - To the following emails (enter a comma separated list):** A text input field.

Al dar clic en “+ Add chart”, como muestra la *Figura 71*, se puede configurar las tablas, gráficos y valores de las variables que sean necesarias, se usa el Type “Widget” para realizar el reporte.

3.11 Programa Control Motor DC con Lectura y Escritura de Variables desde NI myRIO en conjunto con planta EPC a Ubidots.

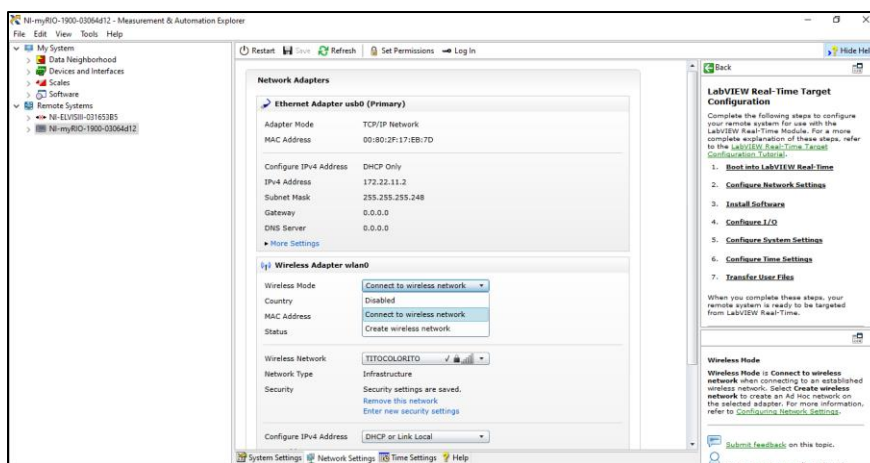
3.11.1 Configuración del Sistema Embebido NI myRIO.

Antes de empezar con el desarrollo del programa y el código de programación del mismo, se configuró el Sistema Embebido NI myRIO de la siguiente manera:

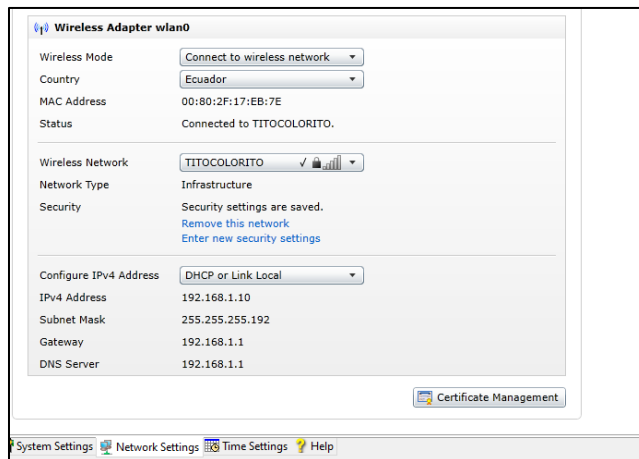
1. Se conectó la tarjeta NI myRIO al computador mediante el cable USB, posteriormente se ingresó a NI MAX. Ir a la sección **Remote Systems**, y seleccionar el NI myRIO e ingresar a **Network Settings**, en la sección **Wireless Mode** seleccionar **Connect to Wireless network** y conectar a la tarjeta a nuestra red de Internet. Por ultimo presionar Save para guardar los cambios, como se muestra en la *Figura 73*.

Figura 73

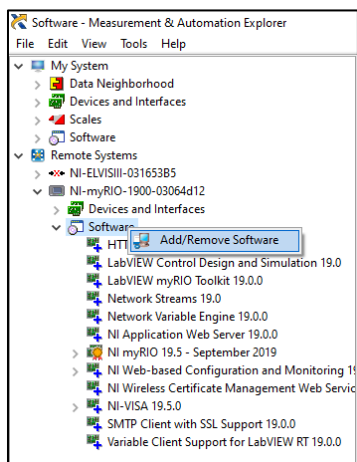
Interfaz NI Max



Tener en cuenta la dirección IP que se asigna a la tarjeta al momento que se conecte, se puede revisar en la Sección Network Settings que muestra la *Figura 74*.

Figura 74*Network Settings*

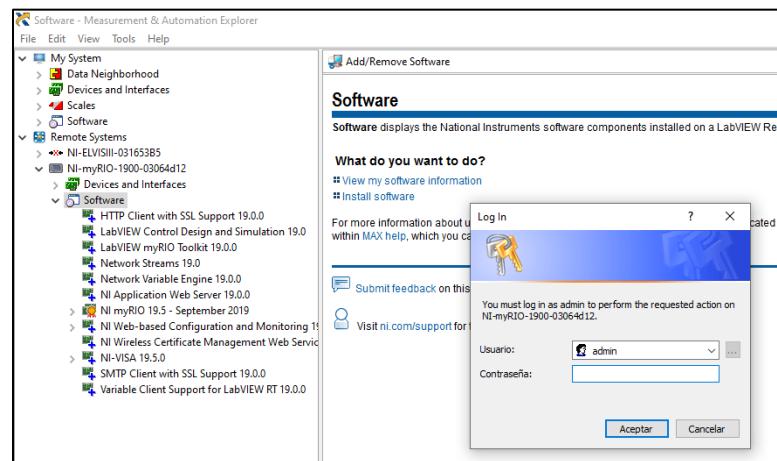
2. Desplegar la pestaña de la tarjeta myRIO y en la sección **Software** instalar los siguientes complementos: **HTTP Client with SSL Support** y **Labview Control Desing and Simulation**, dar clic derecho en **Software** y dar clic en **Add/Remove Software**, como se muestra en la *Figura 75*.

Figura 75*Software instalado en NI myRIO*

En esta sección se puede observar una ventana como en la *Figura 76* para ingresar usuario y contraseña, el usuario es **admin** y la contraseña se deja en blanco.

Figura 76

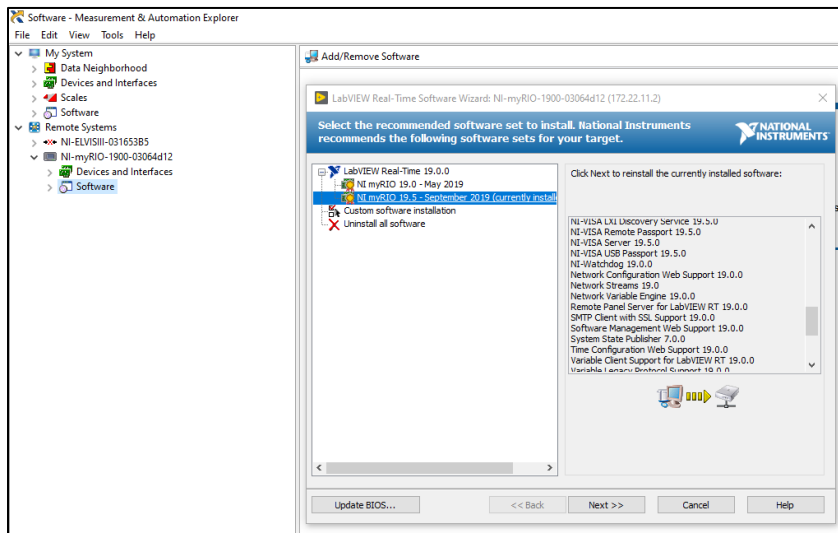
Interfaz de Log in



Se debe seleccionar la versión de software más actual correspondiente al mes en el que se está realizando la configuración y posteriormente dar clic en **Next**, como se puede visualizar en la *Figura 77*.

Figura 77

Interfaz de instalación



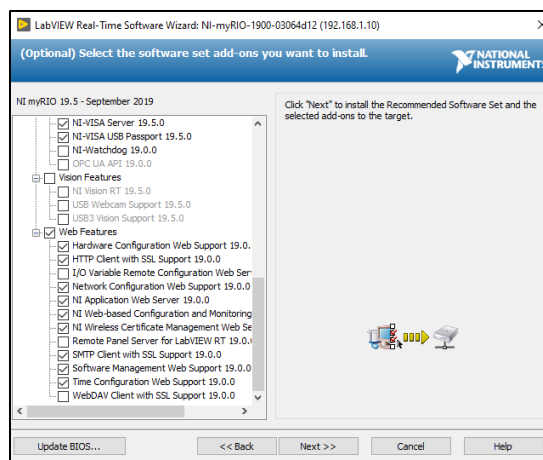
Seleccionar los complementos **HTTP Client with SSL Support y Labview**

Control Desing and Simulation, dar clic en **Next** como se observa en la Figura

78 y se procede a finalizar la instalación.

Figura 78

Finalización de instalación



3.11.2 Programa para comprobar la conexión a la plataforma Ubidots desde NI myRIO

Antes de comenzar con el envío y recepción de datos, se debe comprobar la conexión del Sistema Embebido NI myRIO a la plataforma de Ubidots y a la red de Internet en general, para esto se debe realizar un programa para ejecutar un comando PING.

Para esto se utiliza la función SYSTEM EXCEC de la Paleta de Funciones Connectivity, tomando en cuenta que el SO de la tarjeta NI myRIO es Linux. Se deben crear los siguientes Controles e Indicadores:

Controles

- String: DIRECCION IP
- Numérico: Número de Paquetes
- Numérico: Timeout
- Control de Error

Indicadores

- String: RESPUESTA DEL PING
- String: IP
- Numérico: Tiempo de Respuesta
- Boleano: CONEXIÓN ESTABLECIDA
- Indicador de Error

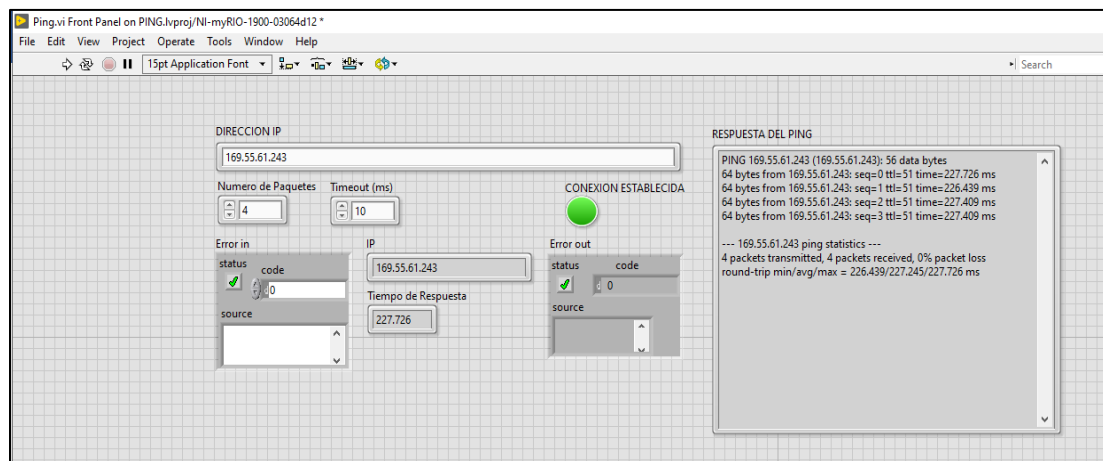
La dirección IP de la plataforma Ubidots es 169.55.61.243, esa dirección se introduce en el Control String “DIRECCION IP” y en el control numérico “Número de

Paquetes” se coloca “4” que es el número de paquetes recomendados, el valor del Control Timeout es de 1000 ms.

El programa ejecutará el comando PING a la dirección especificada y mostrará la respuesta en el Indicador RESPUESTA DEL PING, además que se encenderá el Indicador Boleano CONEXIÓN ESTABLECIDA como se observa en la *Figura 79*.

Figura 79

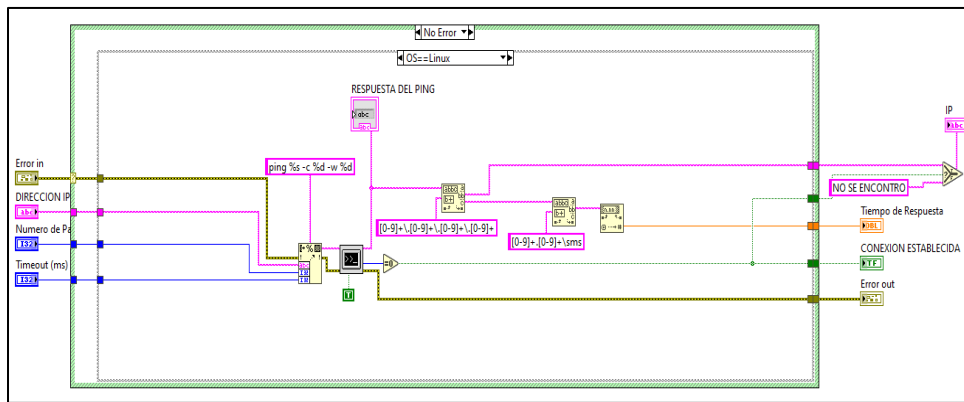
Panel frontal programa ping NI myRIO



En la *Figura 80* se observa en Diagrama de Bloques del Programa.

Figura 80

Diagrama de bloques programa ping NI myRIO

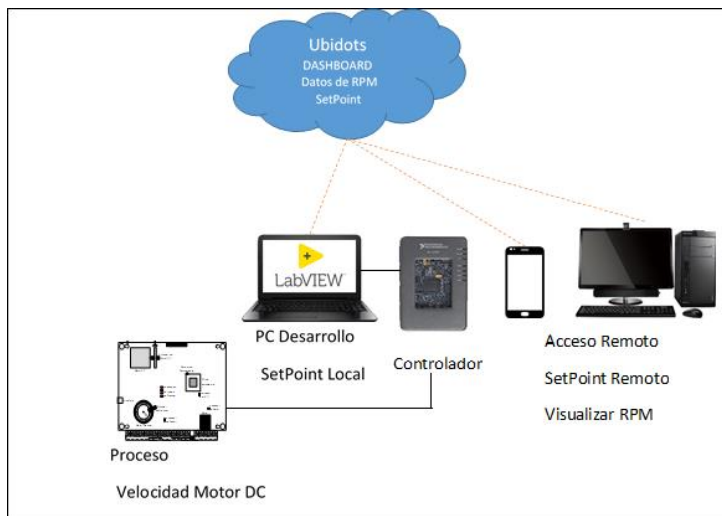


3.11.3 Programa control Velocidad Motor DC de la planta EPC conexión con Ubidots.

En el siguiente programa se desarrolló un Control de Velocidad del Motor DC de la Planta EPC, el cual enviará los datos de velocidad (RPM) a un Dashboard en Ubidots, de igual manera recibirá el valor de SetPoint de forma Remota desde el Dashboard en Ubidots, además de también poder cambiar el valor del Setpoint de forma local. En la *Figura 81* se observa el Esquema del Programa.

Figura 81

Esquema de funcionamiento programa control de velocidad EPC



Nota. El gráfico representa el esquema de funcionamiento del Programa. Adaptado de DemoScript SystemLink Cloud – Versión Publica , por S.Orellana , 2020, Datalights

En la plataforma de Ubidots se creó un Dispositivo con el nombre “**Labview**” y 2 variables una es **RPM** y otra con el nombre **SetPoint**, como lo hicimos en la sección 3.2 y 3.3 respectivamente, adicionalmente un Dashboard y se asigna las variables a los widgets como en la sección 3.4.

Widgets Creados:

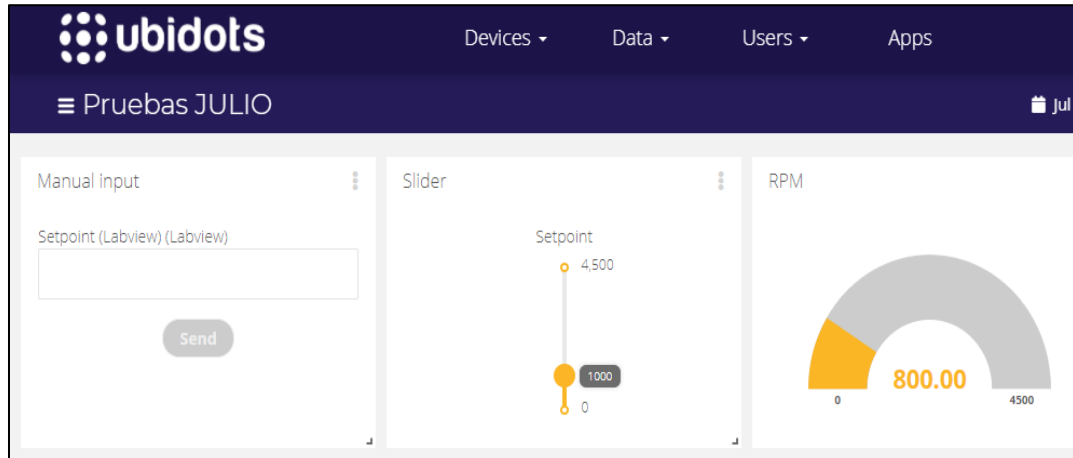
- Slider: Asignar variable SetPoint
- Manual Input: SetPoint
- Gauge: Asignar variable RPM

Con esto se puede cambiar el SetPoint de forma remota desde la slider o ingresando el valor por teclado y visualizar los datos de Velocidad en RPM en el Gauge.

Las variables creadas y asignadas a los widgets dentro del Dashboard se observan en la *Figura 82*.

Figura 82

Dashboard programa control de velocidad de motor dc del EPC



También se crearon los siguientes controles e indicadores en la Interfaz de LabVIEW:

Controles

- String: Label Dispositivo
- String: Default Token
- Boleano: Parar Programa
- Boleano: HABILITAR SP REMOTO
- Numérico: Set Point
- Numérico: PID GAINS

Indicadores

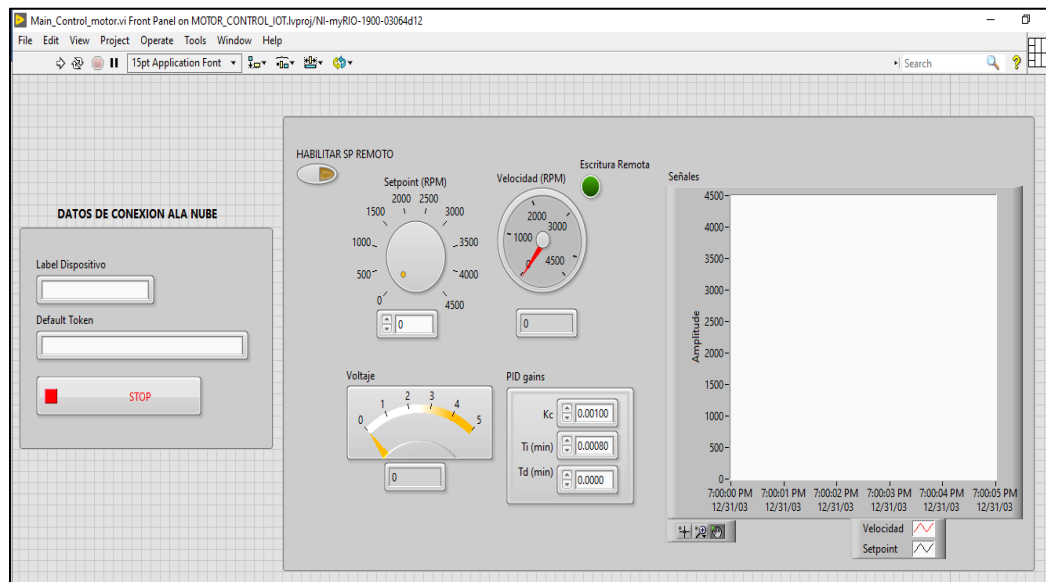
- Numérico Gauge: RPM

- Numérico Gauge: Voltaje
- Boleano: Escritura Remota
- Chart: Señales

Estos controles e indicadores se muestran en la *Figura 83*.

Figura 83

Controles e indicadores programa control de velocidad EPC



Se realizó una programación de lazos while paralelos, en un lazo se tiene la comunicación con la plataforma Ubidots denominado Lazo de Comunicación y otro lazo para realizar el control de velocidad del Motor DC del EPC que se denominó como Lazo de Control.

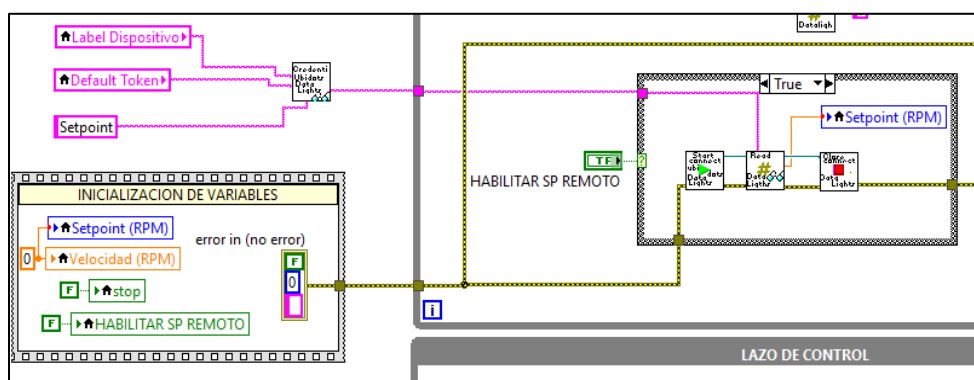
Se necesitan 4 bibliotecas **Credentials.lvlib**, **Read.lvlib**, **Ini_Connect.lvlib** y **Write.lvlib**. Estas bibliotecas contienen los Sub VI's: **Start_connect.vi**, **Close_connect.vi**, **Credenciales_read.vi**, **Credenciales_write.vi**, **Write_Data.vi**, **Numeric_write.vi**, y **Numeric_read.vi**.

3.11.3.1 Lazo de Comunicación

Se necesita recibir la variable SetPoint desde el Dashboard de Ubidots como se muestra en la *Figura 84*. En este caso se usará **Start_connect.vi**, **Close_connect.vi**, **Credenciales_read.vi** y **Numeric_read.vi**.

Figura 84

Recepción de variable setpoint desde dashboard Ubidots

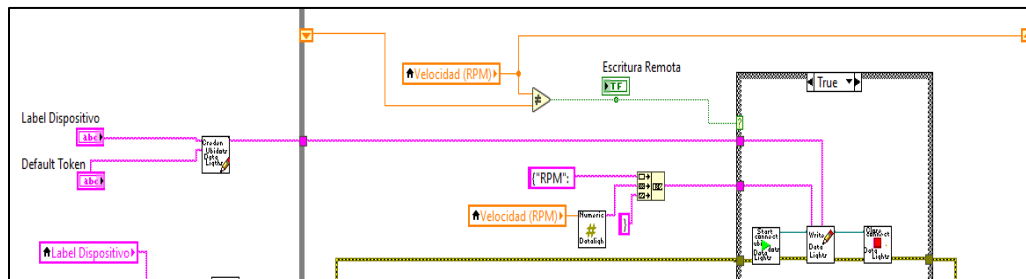


Se enviará a la variable RPM al Dashboard como se muestra en la *Figura 85*. En este caso se usará **Start_connect.vi**, **Close_connect.vi**, **Credenciales_write.vi**, **Write_Data.vi** y **Numeric_write.vi**.

La escritura solamente se activa cuando el valor de la velocidad cambia de esta forma se evita que se escriban datos de forma innecesaria y se produzca saturación de la Red.

Figura 85

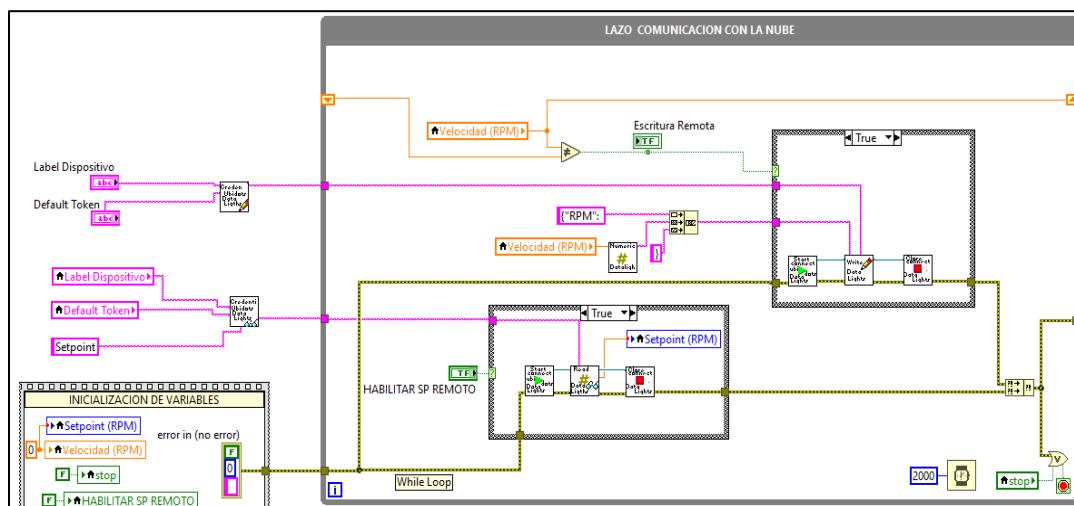
Escritura de velocidad rpm al dashboard de Ubidots



De esta manera queda definido el lazo de comunicación del programa como se observa en la *Figura 86*.

Figura 86

Lazo de comunicación programa control de velocidad motor dc del EPC

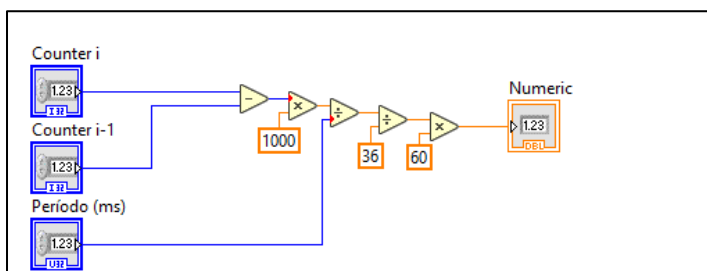


3.11.3.2 Lazo de Control

En este lazo se realizará el control de Velocidad del Motor DC del EPC, se recibirá la medición del Encoder del EPC y se realizará la conversión para obtener la velocidad en RPM, como se muestra en la *Figura 87*. Esta señal entra al canal DIO 2 del NI myRIO.

Figura 87

Conversión medición encoder a rpm



Se usará la Función PID de la Paleta de Funciones Control & Simulation, sus entradas serán:

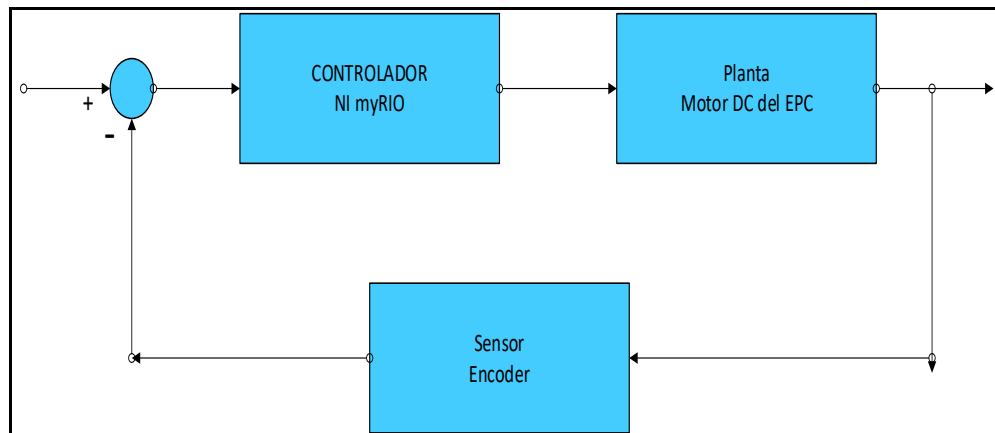
- SetPoint: Control Numérico.
- Variable del Proceso: Señal del Encoder transformada a velocidad en RPM
- PID Gains: Las constantes PID

Y su salida es la variable manipulada que en este caso será el Voltaje que sale del canal analógico AO 0 del NI myRIO.

En la *Figura 88* se observa el Diagrama del Lazo de Control. En donde la variable controlada es la Velocidad y la variable manipulada es el Voltaje.

Figura 88

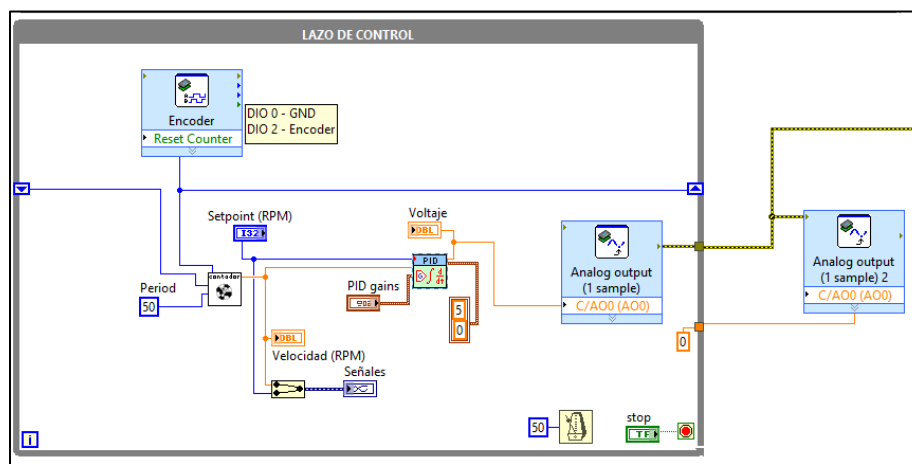
Diagrama del lazo de control de velocidad motor dc



En la *Figura 89* se observa el lazo de Control.

Figura 89

Diagrama de bloques del lazo de control



Para poder empezar la Ejecución del programa habrá que ingresar el Label del Dispositivo y el Default Token del Dashboard de Ubidots, así como las ganancias PID, las ganancias utilizadas fueron proporcionadas por la Empresa Datalights, estas ganancias fueron calculadas por la empresa en proyectos anteriores a este, en este

caso el motor DC es representado por un Sistema de Primer orden así que se tiene un control PI en realidad cuyas constantes son:

$$k_c = 0.00135$$

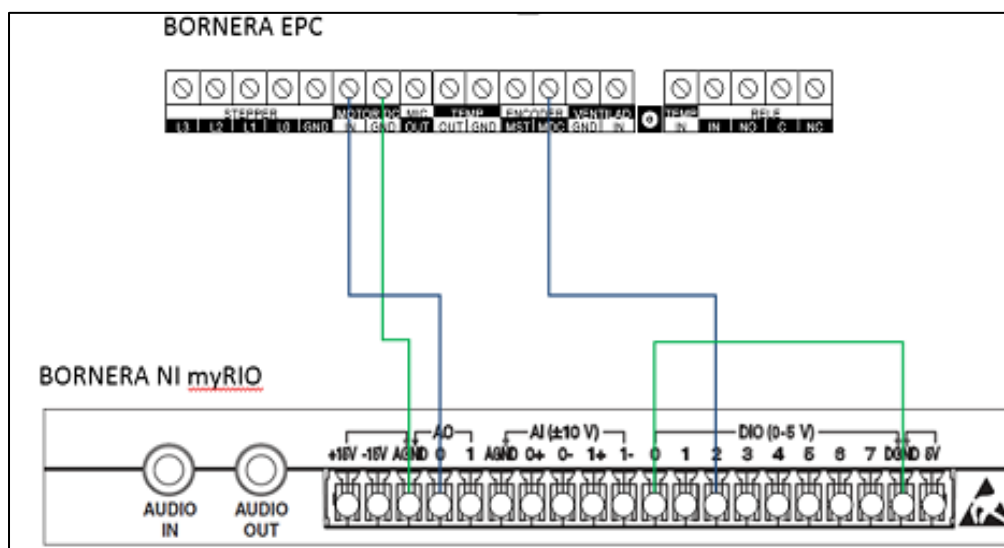
$$T_i = 0.00283$$

$$T_d = 0$$

Habr  que realizar la conexi3n del EPC con el myRIO como se muestra en la *Figura 90*.

Figura 90

Conexi3n de NI myRIO con el EPC



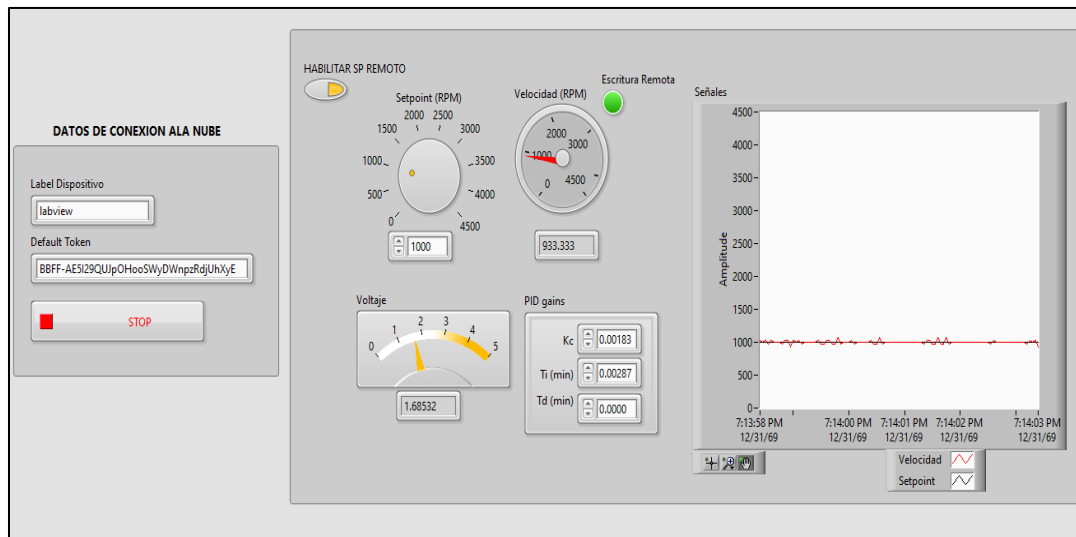
Nota. El gr fico representa el esquema de conexi3n entre el NI myRIO y el EPC.

Adaptado de DemoScript SystemLink Cloud – Versi3n P blica, por S.Orellana , 2020, Datalights

En la *Figura 91* se puede observar el Panel Frontal del programa ejecut ndose.

Figura 91

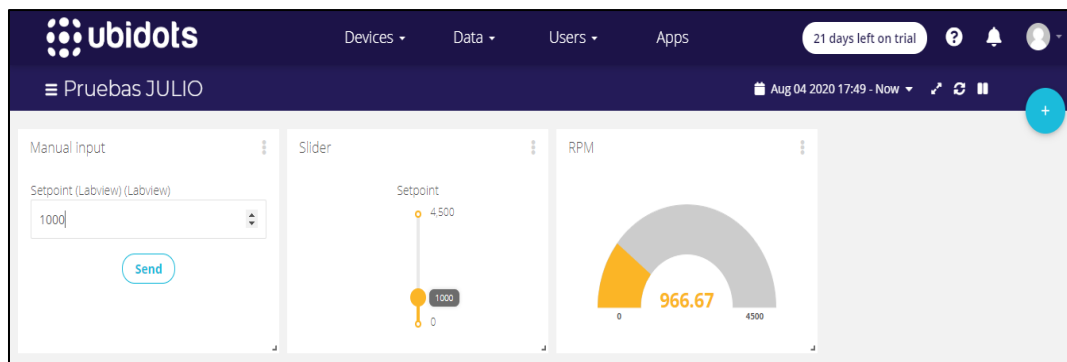
Panel frontal del programa control de velocidad motor dc en ejecución



En la *Figura 92* se observa el Dashboard en Ejecución.

Figura 92

Dashboard de Ubidots programa control de velocidad en ejecución



3.12 Interactuar con Dashboards desde un web browser y desde un Smartphone.

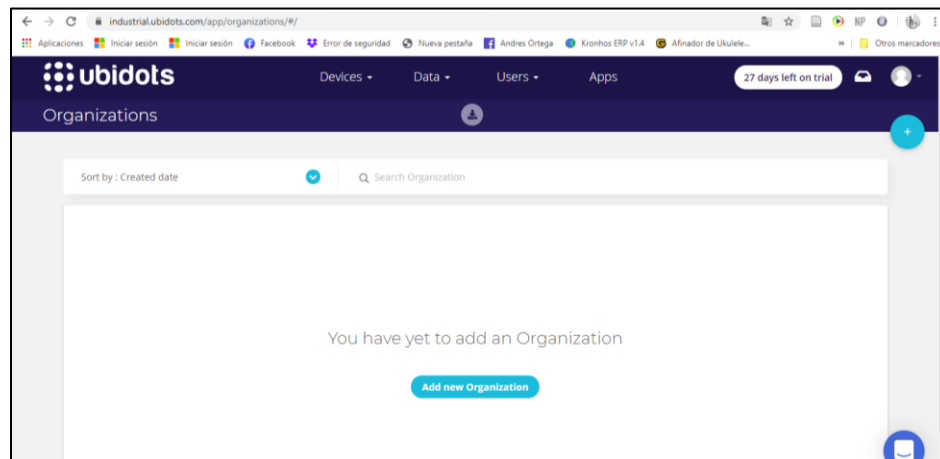
Para usar estas funciones se creó una organización que contenga al Dispositivo y Dashboard que fueron creados anteriormente y crear un usuario nuevo que pueda acceder al Dashboard.

3.12.1 Crear Organización.

Dar clic en “USERS” y en la pestaña “Organizations”, como se observa en la *Figura 93*.

Figura 93

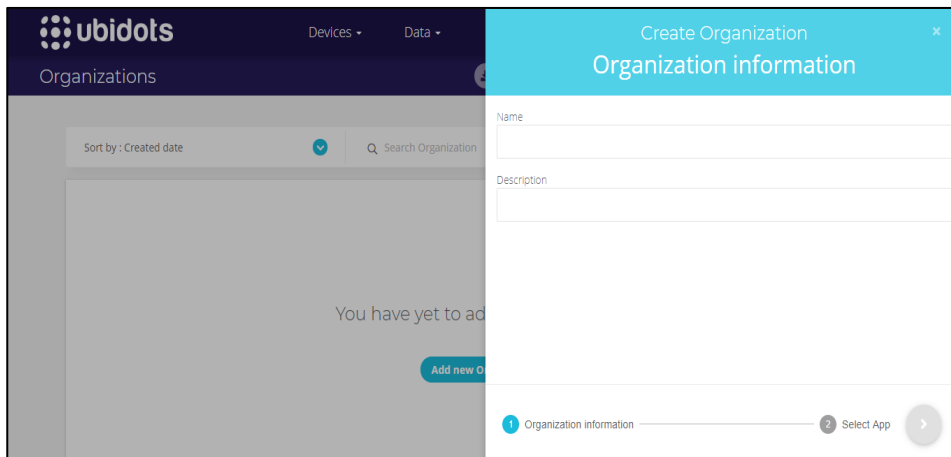
Crear organización



Dar clic en “Add new Organization” y establecer un nombre, como se muestra en la *Figura 94*.

Figura 94

Nombre de la organización

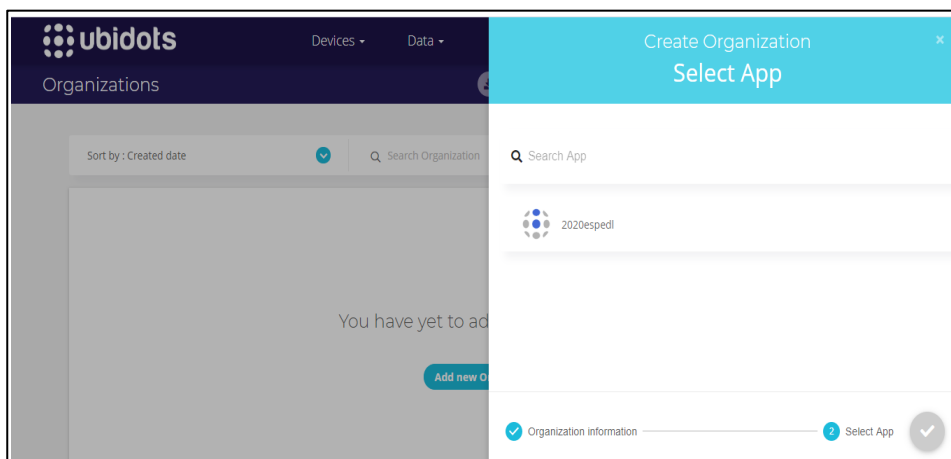


The screenshot shows the Ubidots interface with a modal dialog titled "Create Organization" and a sub-header "Organization information". The dialog contains two input fields: "Name" and "Description". Below the fields is a progress indicator with two steps: "1 Organization information" (active) and "2 Select App". The background shows the "Organizations" page with a search bar and a message "You have yet to add..." with an "Add new Organization" button.

A continuación, en la opción de seleccionar aplicación, aparecerá el nombre de la aplicación que se haya establecido en la creación de la cuenta de Ubidots, y se debe seleccionar esa como se observa en la *Figura 95*.

Figura 95

Seleccionar la aplicación para la organización



The screenshot shows the Ubidots interface with the modal dialog titled "Create Organization" and a sub-header "Select App". The dialog contains a search bar labeled "Search App" and a list of available applications, with "2020espedi" selected. Below the list is a progress indicator with two steps: "1 Organization information" (checked) and "2 Select App" (active). The background shows the "Organizations" page with a search bar and a message "You have yet to add..." with an "Add new Organization" button.

Luego de tener la organización creada dirigirse al icono que tiene forma de lápiz para editar la organización como se observa en la *Figura 96*, en la sección Dashboards en el símbolo “+”, como se observa en la *Figura 97*, agregar el Dashboard creado anteriormente, como se muestra en la *Figura 98*.

Figura 96

Editar organización

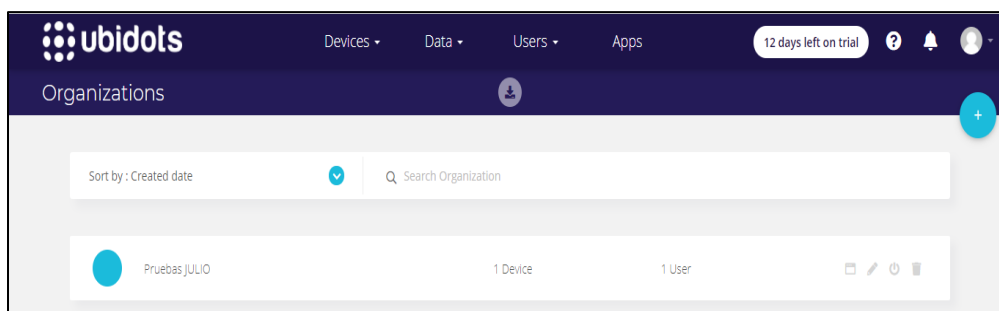
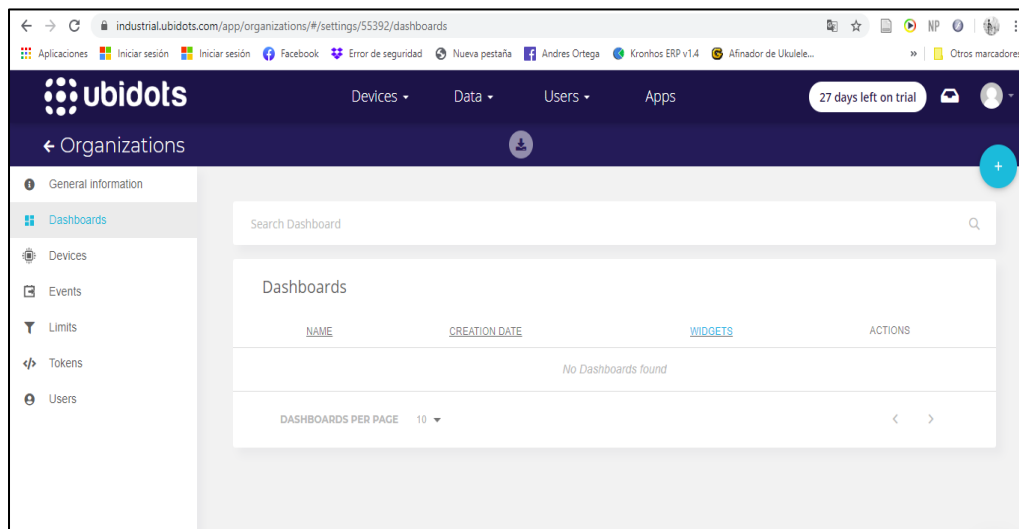


Figura 97

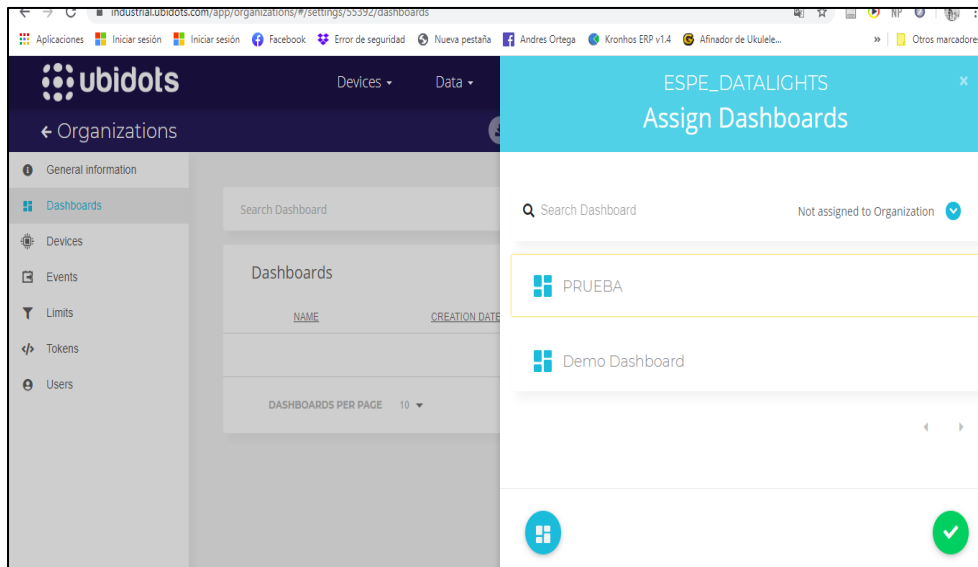
Enlazar un dashboard a la organización



Dar clic en el símbolo check.

Figura 98

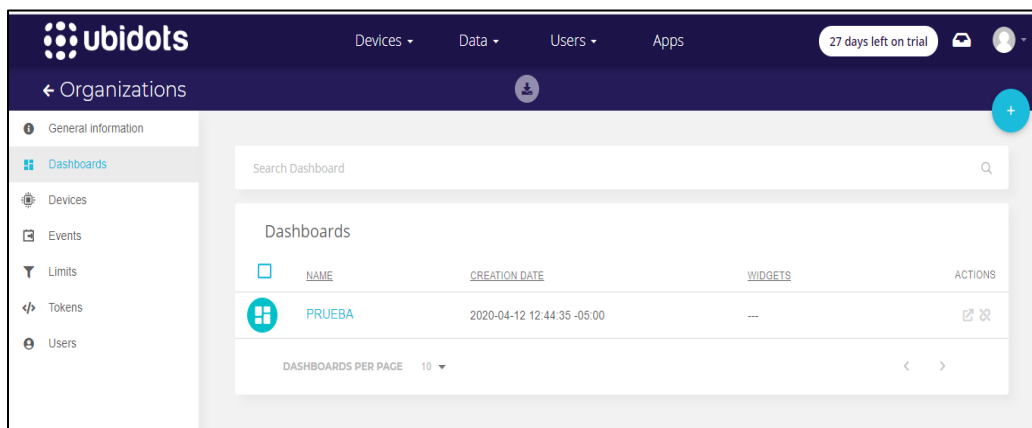
Selección del dashboard para la organización



A continuación, se asigna el Dashboard a la organización, como se puede observar en la *Figura 99*.

Figura 99

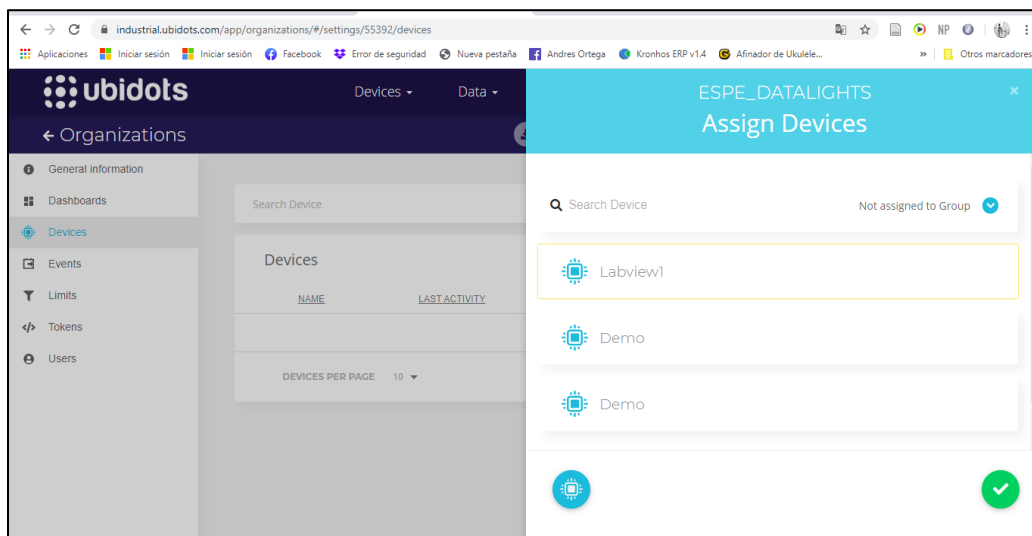
Dashboard asignado a la organización



En la sección Devices seleccionar el dispositivo creado, como se puede ver en la *Figura 100* y dar clic en el símbolo check.

Figura 100

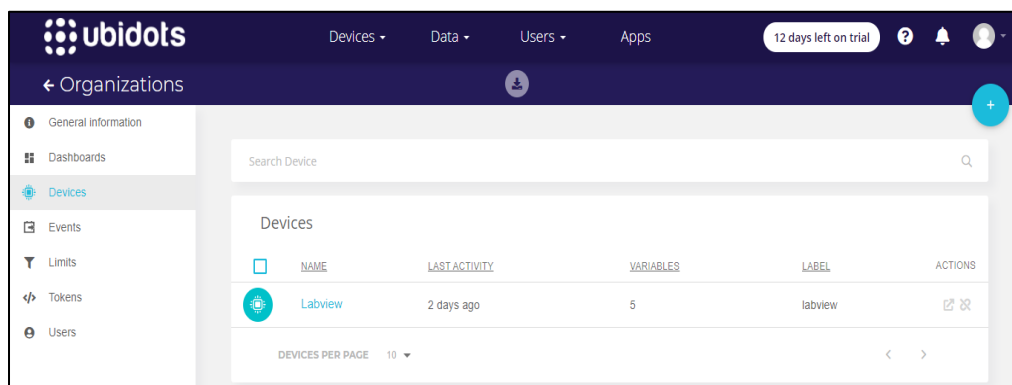
Selección de dispositivo para la organización



De esta manera queda asignado el dispositivo y el Dashboard a la Organización como se observa en la *Figura 101*.

Figura 101

Organización creada con todos sus elementos

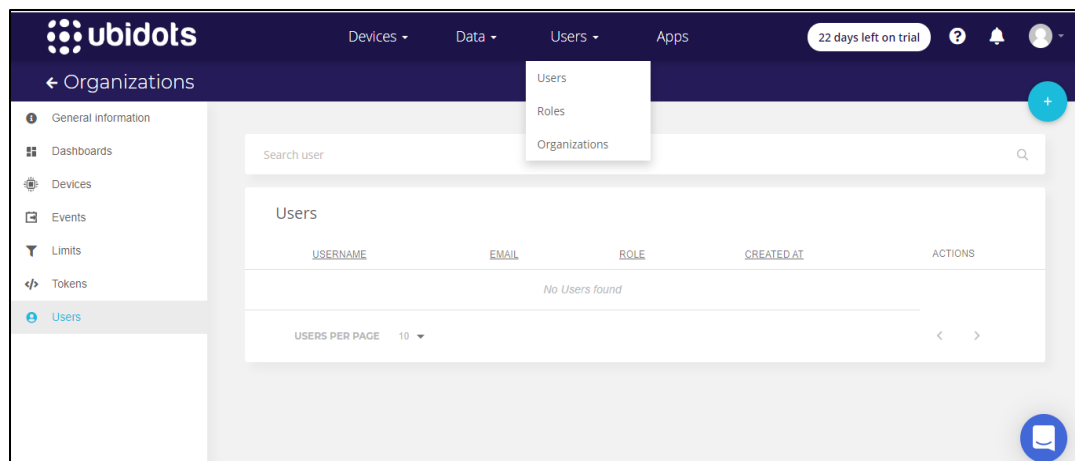


3.12.2 Creación Usuario

En la sección Users seleccionar “Users”, dar clic en el símbolo “+” y añadir el usuario, como se observa en la *Figura 102*.

Figura 102

Creación de usuario



Colocar un nombre de usuario y un email como se observa en la *Figura 103*. En este caso el usuario creado es “espe1”, la invitación con la dirección llegará al email establecido.

Figura 103

Ingreso de nombre de usuario y email

The screenshot shows the 'Add new User' modal in the Ubidots interface. The modal has a light blue header with the title 'Add new User' and a close button. Below the header, there are two input fields: 'Username' and 'Email'. Both fields are currently empty and have red error messages below them: 'Please enter a valid username' and 'Please enter a valid email'. At the bottom of the modal, there is a progress indicator with four steps: 1. User information (checked), 2. Select organizations (current step), 3. Select role, and 4. Send invitation. A right arrow button is also present.

A continuación, seleccionar la Organización que se observa en la *Figura 104*, aparecerá el nombre de la Organización creada anteriormente.

Figura 104

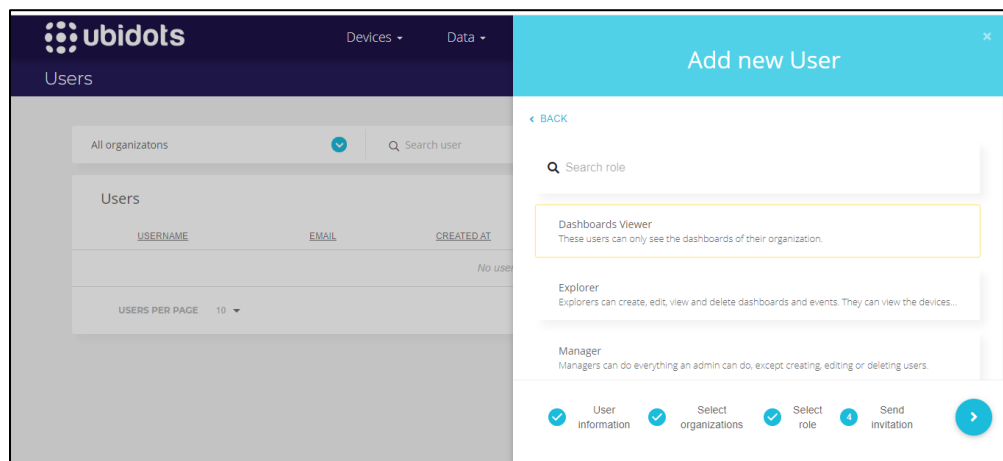
Asignar una organización a un usuario

The screenshot shows the 'Add new User' modal in the Ubidots interface, now in the 'Select organizations' step. The modal has a light blue header with the title 'Add new User' and a sub-header 'Select organizations'. Below the header, there is a search bar labeled 'Search Organization' and a 'Select All' checkbox. A single organization is listed: 'Pruebas JULIO'. At the bottom of the modal, the progress indicator shows that step 2, 'Select organizations', is now checked, and step 3, 'Select role', is the current step. A right arrow button is also present.

En la interfaz de la *Figura 105* se selecciona el rol que le queremos entregar al usuario.

Figura 105

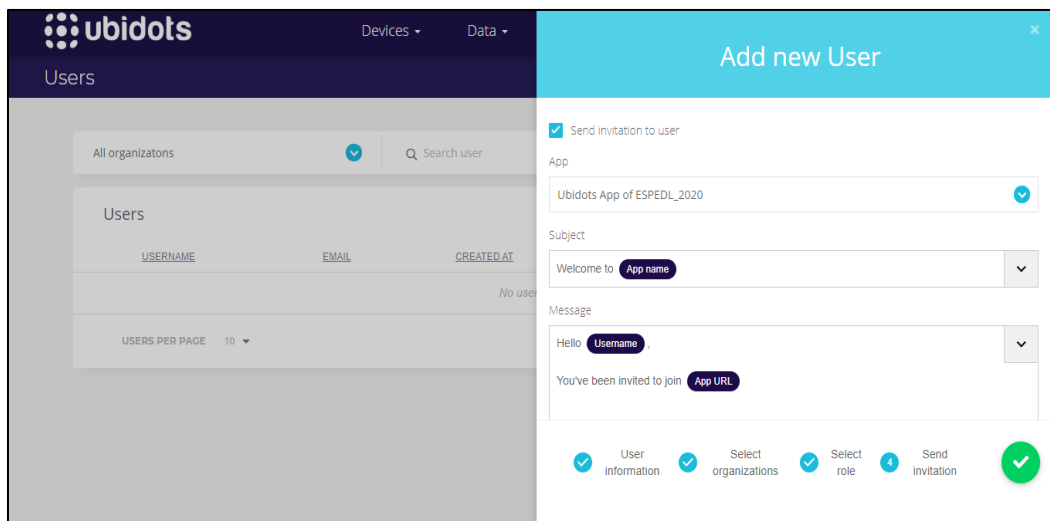
Selección de rol para el usuario



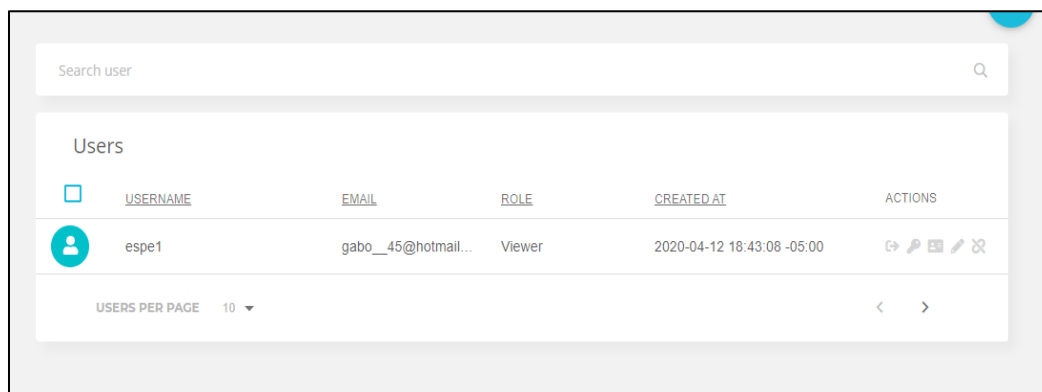
Al seleccionar el tipo de rol para el usuario dar clic en el símbolo check que se observa en la *Figura 106*.

Figura 106

Guardar el usuario

**Figura 107**

Establecer contraseña de usuario



Al dar clic en la llave como se observa en la *Figura 107* se puede establecer una contraseña para el acceso del usuario al Dashboard. En este caso los datos son:

Usuario: espe1

Contraseña: espe1

Finalmente, con este usuario se puede acceder al Dashboard deseado de manera remota desde un Web Browser o Smartphone, y se realiza las acciones permitidas respecto al rol que se asignó al usuario.

3.12.3 Web Browser

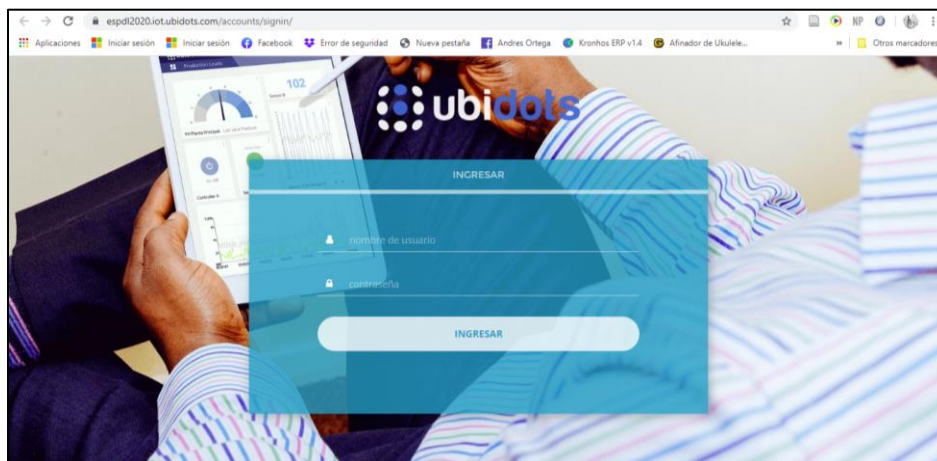
Para hacerlo desde un web browser se tiene que ingresar al dominio conformado por el nombre de la aplicación en conjunto con `iot.ubidots.com` es decir (El nombre de la aplicación es el nombre que se asignó en la creación de la cuenta):
`appname.iot.ubidots.com`.

En nuestro caso sería: **`espdl2020.iot.ubidots.com`**

Se ingresa con el usuario y contraseña creados anteriormente para el usuario “espe1”, como se observa en la *Figura 108*.

Figura 108

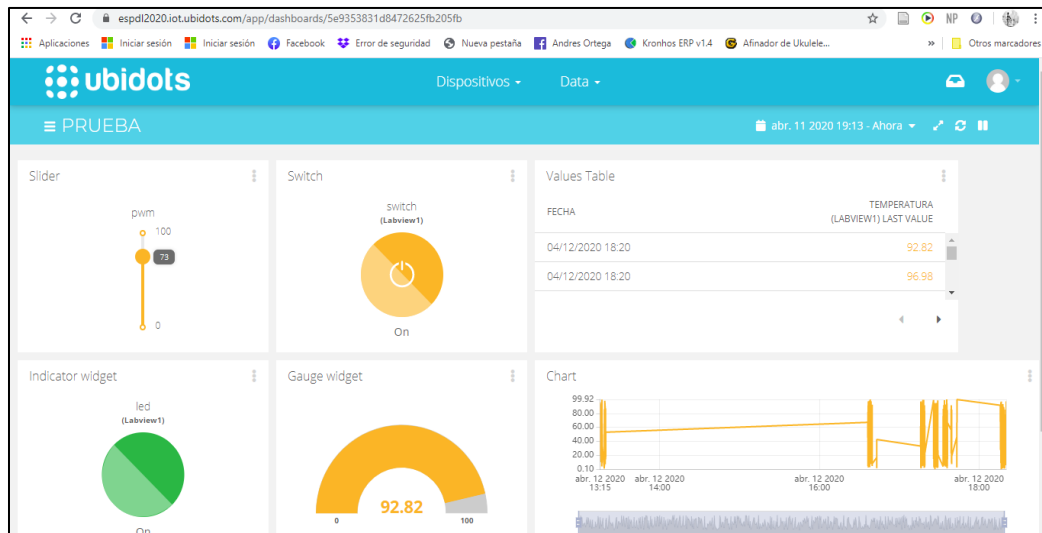
Web browser para ingresar al dashboard de forma remota



En la *Figura 109* se observa el dashboard ingresado de una PC en forma remota.

Figura 109

Dashboard del programa visualizado de forma de remota

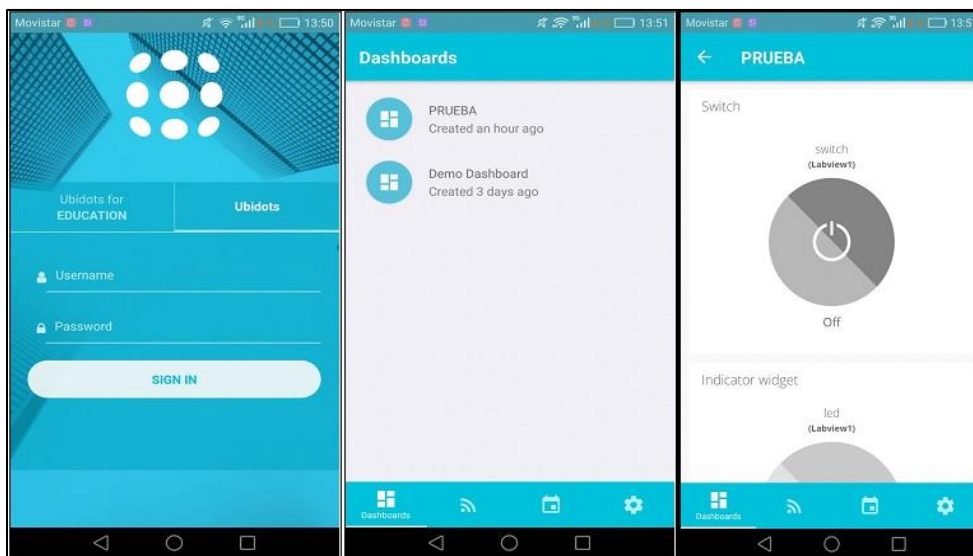


3.12.4 Smartphone

La aplicación está disponible solamente para Smartphone con SO Android, y está disponible en la tienda Play Store con el nombre Ubidots Explorer. Se inicia sesión con el usuario y contraseña creados anteriormente, como se observa en la *Figura 110*.

Figura 110

Acceso al dashboard desde un smartphone en forma remota



CAPÍTULO IV

4. PRUEBAS Y RESULTADOS DEL PROYECTO

4.1 Pruebas de funcionamiento

4.1.1 Programa Enviar datos desde PC a Ubidots.

En la *Figura 53* se puede observar el programa en ejecución, después de mantenerlo en Ejecución durante 2 horas y evaluado bajo distintas situaciones se tienen los siguientes resultados evidenciados en la *Tabla 2*.

Tabla 2

Resultados de la prueba del programa enviar datos desde pc a Ubidots

RENDIMIENTO	BUENO	REGULAR	MALO
Network Delay	X		
Velocidad de Ejecución	X		
Ejecución en una Red de Internet con velocidad baja	X		
Ejecución en una Red de Internet con velocidad alta	X		
Ejecución en una Red compartida con gran cantidad de dispositivos		X	

RENDIMIENTO	BUENO	REGULAR	MALO
Ejecución en una Red compartida con pocos dispositivos	X		

4.1.2 Programa Recibir datos desde Plataforma Ubidots a la PC

En la *Figura 56* se puede observar el programa en ejecución, después de mantenerlo en Ejecución durante 2 horas y evaluado bajo distintas situaciones se tienen los siguientes resultados evidenciados en la *Tabla 3*.

Tabla 3

Resultados de funcionamiento programa recibir datos desde Ubidots a pc

RENDIMIENTO	BUENO	REGULAR	MALO
Network Delay	X		
Velocidad de Ejecución	X		
Ejecución en una Red de Internet con velocidad baja	X		
Ejecución en una Red de Internet con velocidad alta	X		
Ejecución en una Red compartida con		X	

RENDIMIENTO	BUENO	REGULAR	MALO
gran cantidad de dispositivos			
Ejecución en una Red compartida con pocos dispositivos	X		

4.1.3 Programa Control de Velocidad del Motor DC del EPC conectado a la Plataforma Ubidots

En la *Figura 91* se puede observar el programa en ejecución, después de mantenerlo en Ejecución durante 2 horas y evaluado bajo distintas situaciones se tienen los siguientes resultados evidenciados en la *Tabla 4*.

Tabla 4

Pruebas de funcionamiento programa control de velocidad del motor DC

RENDIMIENTO	BUENO	REGULAR	MALO
Network Delay	X		
Velocidad de Ejecución	X		
Ejecución en una Red de Internet con velocidad baja		X	
Ejecución en una Red de Internet con velocidad alta	X		

RENDIMIENTO	BUENO	REGULAR	MALO
Ejecución en una Red compartida con gran cantidad de dispositivos			X
Ejecución en una Red compartida con pocos dispositivos	X		
Funcionamiento del Control del Motor	X		

4.2 Pruebas de Latencia de PC y NI myRIO en relación a la conexión con la Plataforma Ubidots

En la *Tabla 5* observamos los resultados de la ejecución del comando PING tanto en el PC y el Sistema Embebido NI myRIO.

Tabla 5

Prueba de latencia en pc y NI myRIO conexión Ubidots

Latencia	PC	myRIO
Paquetes Enviados	4	4
Paquetes Recibidos	4	4
Paquetes Perdidos	0	0
Tamaño Paquetes(bytes)	32	64

Latencia	PC	myRIO
TTL	51	51
Tiempo ida y vuelta MAX (ms)	220	378.149
Tiempo ida y vuelta MIN (ms)	220	220.450
Latencia (Promedio ms)	220	261.645

4.3 Pruebas de Trafico de Red

4.3.1 Programa Enviar datos desde PC a Ubidots.

En la *Figura 111* se observa los resultados de la prueba de trafico de red de la PC conectada a la Plataforma Ubidots.

Figura 111

Resultados tráfico de red programa Wireshark al enviar datos

The screenshot shows the 'Wireshark - Capture File Properties - Microsoft Wi-Fi' window. The 'Details' pane is active, displaying the following information:

- Name:** C:\Users\LENOVO\AppData\Local\Temp\wireshark_Wi-Fi_20200811174258_a10232.pcapng
- Length:** 25 MB
- Hash (SHA-256):** 28605cd3bd8b7e1ea5d904c04bb20ef49f5669b703e78e9eb242b076811afe
- Hash (RIPEND 160):** f8be199b2db1748d2ba0ffe7fd7b6c518c0b87a
- Hash (SHA 1):** 2015790f4ab0537807ece79715bcabfc480b99b
- Format:** Wireshark/... - pcapng
- Encapsulation:** Ethernet

Time

- First packet: 2020-08-11 17:42:58
- Last packet: 2020-08-11 17:53:13
- Elapsed: 00:10:15

Capture

- Hardware: Intel(R) Core(TM) i3-4010U CPU @ 1.70GHz (with SSE4.2)
- OS: 64-bit Windows 10 (19H3), build 18362
- Application: Dumpcap (Wireshark) 3.2.5 (v3.2.5-0-ged20d5ea8138)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

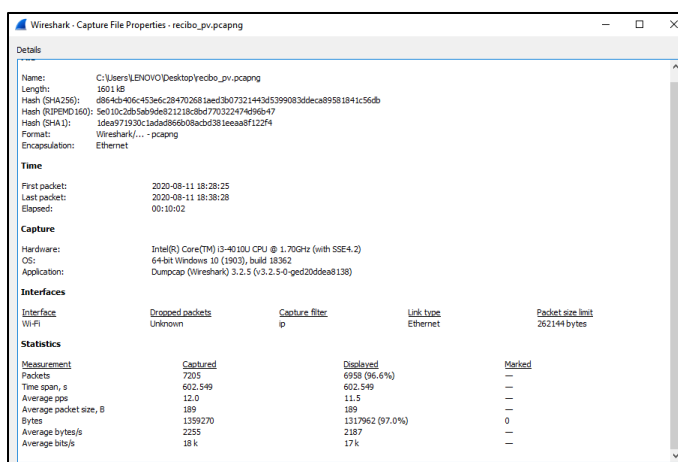
Measurement	Captured	Discarded	Marked
Packets	51402	35292 (68.6%)	—
Time spent, s	615.125	615.011	—
Average pps	83.6	57.3	—
Average packet size, B	470	274	—
Bytes	24183346	9663590 (40.0%)	0
Average bytes/s	39 k	15 k	—
Average bits/s	314 k	125 k	—

4.3.2 Programa Recibir datos desde Plataforma Ubidots a la PC

En la *Figura 112* se observa los resultados de la prueba de tráfico de red de la PC conectada a la Plataforma Ubidots.

Figura 112

Resultados tráfico de red en el programa Wireshark al recibir datos



En la *Tabla 6* observamos los resultados de las pruebas de latencia realizadas, ejecutando los programas de envío y recepción de datos hacia y desde la Plataforma Ubidots.

Tabla 6

Prueba tráfico de red en envío y recepción de datos pc-Ubidots

	Envío	Recepción
# Paquetes Generados	3696	7205
Cantidad de Datos Generados	774 kB	1601 kB

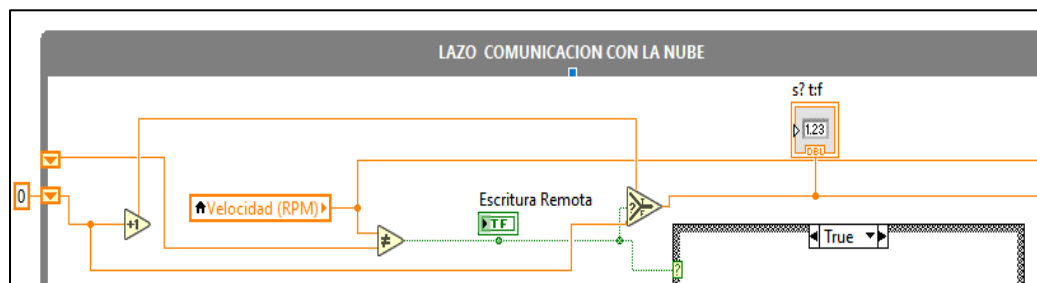
	Envío	Recepción
Velocidad Promedio	1.08kB/s	2.25kB/s

4.3.3 Programa Control de Velocidad del Motor DC del EPC conectado a la Plataforma Ubidots

Para medir la cantidad de Datos que el programa escribe al Dashboard de Ubidots se usó el número de veces que se activó el indicador Booleano **Escritura Remota**, en conjunto con el tamaño de la trama que se escribe, como se observa en la *Figura 113*.

Figura 113

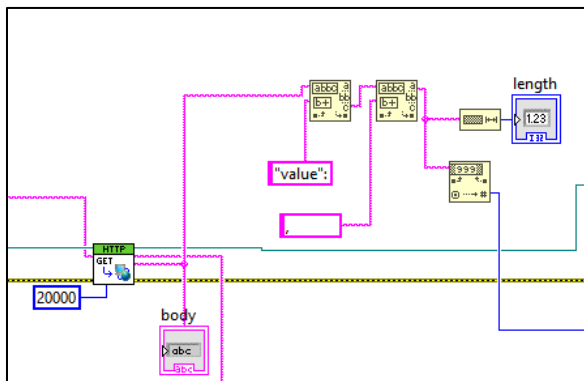
Código para determinar la veces que se enciende la escritura remota



Para medir la cantidad de datos recibidos se midió la longitud del dato recibido en cada interacción de un segundo (*Figura 114*).

Figura 114

Código para medir cantidad de datos recibidos



La prueba se realizó por 60 minutos y se obtuvo los resultados que se muestran en la *Tabla 7*.

Tabla 7

Resultados de tráfico del control de velocidad del motor dc

	Envío	Recepción	Envío y Recepción
Velocidad	120 bits/s	48bits/s	
Cantidad de Datos Generados	21.33 kB	21.6kB	42.93kB

4.4 Validación de la hipótesis

La hipótesis “Realizar la conexión de un Sistema Embebido de National Instruments con una Plataforma de Servicios en la Nube usando lenguaje de Abstracción medio en el Software Labview”, ha sido comprobada a través del conjunto

de Sub VI's desarrollados que permiten una conexión estable con la Plataforma Ubidots utilizando el protocolo de comunicación HTTP y el formato de escritura de datos JSON.

El desarrollo de este conjunto de Sub VI's usando únicamente las Funciones básicas de la Paleta disponible en el Software LabVIEW nos permite evidenciar, la posibilidad de usar un lenguaje de abstracción de nivel medio para realizar conexiones entre Sistemas Embebidos de NI y en este caso la Plataforma de Servicios en la Nube Ubidots.

Como podemos observar en la *Tabla 5*, la latencia del Sistema Embebido NI myRIO en conexión con la Plataforma Ubidots es de 261 ms y de los 4 paquetes enviados ninguno se perdió, manteniendo de esta forma una conexión estable y exitosa.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Debido al momento actual que vive el mundo en donde la Educación tuvo que convertirse forzosamente en Virtual, la aplicabilidad académica del código desarrollado tiene un gran potencial para ser implementado en laboratorios remotos y, además, para empezar con pruebas que permitan hacer aportes a la aplicación de los nuevos marcos tecnológicos ya sea el Internet de las Cosas (IoT) o el Internet Industrial de las Cosas (IIoT).

Se logró desarrollar un conjunto de Sub VI's que usan como base el protocolo de comunicación HTTP y el formato de transmisión de datos JSON que permiten la conexión entre el Software LabVIEW, el Sistema Embebido NI myRIO y la Plataforma de Servicios en la Nube Ubidots.

Se realizó únicamente una prueba de concepto aplicable al Internet Industrial de las Cosas(IIoT) con la Plataforma Ubidots, debido a que esta plataforma permitió la conexión usando un lenguaje de abstracción medio del software LabVIEW. Existen otras plataformas que tienen Toolkits predefinidos para entablar comunicación que no se pueden modificar ni permiten experimentar con los mismos.

Se desarrolló el código de conexión usando las Funciones HTTP Client, Matemáticas, Comunicación de Datos, entre otros; disponibles en las Paletas predefinidas del Software LabVIEW lo que le da al código desarrollado un nivel de Abstracción Medio, que permite sin ningún problema establecer la conexión.

El conjunto de Sub VI's del presente trabajo de investigación, fue desarrollado para Sistemas Embebidos de National Instruments, específicamente para la tarjeta NI myRIO. Este código permite su reutilización en dispositivos de otros fabricantes que tengan compatibilidad de conexión con el Software LabVIEW.

La aplicabilidad Industrial del código desarrollado, puede ser considerado como experimental por el momento, debido a cuestiones de seguridad en la Transmisión de Datos. Se usó el protocolo http para la conexión y no el protocolo https ya que el Sistema NI myRIO no permitió la instalación del certificado de seguridad del mismo.

5.2 Recomendaciones

Para desarrollar pruebas o ejecutar aplicaciones tanto en el ámbito académico como industrial, es importante trabajar con una red de Internet exclusiva para evitar cualquier tipo de problemas en la comunicación con la Plataforma de servicios en la Nube.

Otro factor importante a considerar en el momento de realizar conexiones remotas es la velocidad del servicio de Internet, más aún si las aplicaciones que se desarrollen con lleven un gran procesamiento de máquina.

Conjuntamente con lo descrito anteriormente, la Revisión de los niveles de seguridad que tiene el proveedor de Internet y el programa Antivirus (en el caso de usar un PC para el desarrollo) es significativo para evitar conflictos en la comunicación con Plataformas de Servicios en la Nube.

La investigación realizada abre el camino para una exploración en dos ámbitos. En primer lugar, se encuentra el ámbito de seguridad en la transmisión de datos, esta debe contar con la encriptación necesaria para poder aplicar el código desarrollado a nivel Industrial y en segundo lugar tenemos los protocolos de comunicación, es preciso

utilizar un protocolo mejor adaptado para el uso del Internet de las Cosas (IoT) y el Internet Industrial de las Cosas (IIoT).

REFERENCIAS BIBLIOGRÁFICAS

Wortmann, F., & Flüchter, K. (2015). Internet of Things Technology and Value Added.

Springer, 2-3.

(2018). Recuperado el 4 de Febrero de 2020, de RS : [https://es.rs-](https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things&redirect-relevancy-data=636F3D3126696E3D4931384E53656172636847656E65726963266C753D6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4C7D5C707B4E647D2D2C2F255C2E5D2B2426)

[online.com/web/generalDisplay.html?id=i/iot-internet-of-things&redirect-](https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things&redirect-relevancy-data=636F3D3126696E3D4931384E53656172636847656E65726963266C753D6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4C7D5C707B4E647D2D2C2F255C2E5D2B2426)

[relevancy-](https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things&redirect-relevancy-data=636F3D3126696E3D4931384E53656172636847656E65726963266C753D6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4C7D5C707B4E647D2D2C2F255C2E5D2B2426)

[data=636F3D3126696E3D4931384E53656172636847656E65726963266C753D](https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things&redirect-relevancy-data=636F3D3126696E3D4931384E53656172636847656E65726963266C753D6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4C7D5C707B4E647D2D2C2F255C2E5D2B2426)

[6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4](https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things&redirect-relevancy-data=636F3D3126696E3D4931384E53656172636847656E65726963266C753D6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4C7D5C707B4E647D2D2C2F255C2E5D2B2426)

[C7D5C707B4E647D2D2C2F255C2E5D2B2426](https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things&redirect-relevancy-data=636F3D3126696E3D4931384E53656172636847656E65726963266C753D6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4C7D5C707B4E647D2D2C2F255C2E5D2B2426)

Bitter , R., Mohiuddin, T., & Nawrocki, M. (2007). *LabView: Advanced Programming*

Techniques, Second Edition. Boca Raton: Taylor & Francis Group.

Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of

things (IIoT): An analysis framework. *ScienceDirect*.

Damian, O. (2020). *Academia.edu*. Recuperado el 16 de Febrero de 2020, de

[https://www.academia.edu/8013013/Cap%C3%ADtulo_5_El_paradigma_cliente_](https://www.academia.edu/8013013/Cap%C3%ADtulo_5_El_paradigma_cliente_servidor)

[servidor](https://www.academia.edu/8013013/Cap%C3%ADtulo_5_El_paradigma_cliente_servidor)

Datalights. (2020). *Datalights*. Recuperado el 12 de Febrero de 2020, de

https://datalights.com.ec/site2/?option=com_content&view

Hejazi, H., Rajab, H., Cinkler, T., & Lengyel, L. (2018). Survey of Platforms for Massive

IoT. *IEEE International Conference on Future IoT Technologies (Future IoT)* (pp.

1-8). Eger: IEEE.

IEEE. (2015, Mayo). *IEEE Internet of things*. Recuperado el 25 de Febrero de 2020, de

<https://iot.ieee.org/definition.html>

Instruments, N. (2020). *NI*. Recuperado el 16 de Febrero de 2020, de NI:

<https://www.ni.com/es-cr/support/model.myrio-1900.html>

Instruments, N. (2020). USER GUIDE AND SPECIFICATIONS NI myRIO-1900.

Recuperado el 17 de Febrero de 2020, de <https://www.ni.com/es-cr/shop/engineering-education/portable-student-devices/myrio-student-embedded-device/what-is-myrio.html>

IoTONE. (2020). *IoTONE*. Recuperado el 4 de Marzo de 2020, de

<https://www.iotone.com/software/ubidots-iot-application-development-platform/s71>

Kodosky, J. (2020). LabVIEW. *ACM Magazines*, 23.

Llanos, A. (2020). *Outsourcing Automation Systems*. Recuperado el 16 de Julio de

2020, de oasys: <https://oasys-sw.com/diferencias-iiot-e-iiot/>

Microsoft Azure. (2020). *Microsoft Azure*. Recuperado el 10 de Marzo de 2020, de

<https://azure.microsoft.com/es-es/overview/internet-of-things-iiot/iiot-technology-protocols/>

Microsoft Azure. (2020). *Microsoft Azure*. Recuperado el 7 de Marzo de 2020, de

<https://azure.microsoft.com/es-es/overview/internet-of-things-iiot/iiot-security-cybersecurity/>

National Instruments. (2019). *National Instruments Corp*. Recuperado el 5 de Marzo de

2020, de <https://www.ni.com/es-cr/innovations/white-papers/17/a-practical-guide-for-connecting-labview-to-the-industrial-iiot.html>

National Instruments. (2020). *National Instruments Corp*. Recuperado el 15 de Marzo de

2020, de <https://www.ni.com/es-cr/shop/labview.html>

- Optical Networks . (2020, Agosto 9). *Optical Networks*. Recuperado el 14 de Agosto de 2020, de <https://www.optical.pe/internet-industrial-de-las-cosas-iiot/>
- Orellana, S. (2020). *Datalights*. Recuperado el 2 de Febrero de 2020, de <https://datalights.com.ec/site2/>
- Sáez, I. P. (2019). *Incibe-cert*. Recuperado el 7 de Febrero de 2020, de <https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones>
- Schneider, S. (2016). The industrial internet of things (IIoT): Applications and taxonomy. In H. Geng, *Internet of Things and Data Analytics Handbook* (pp. 41- 42). Sunnyvale: Wiley.
- Semle, A. (2016). Protocolos IIoT para considerar. *AADECA REVISTA*, 32 - 35.
- Serpanos, D., & Wolf, M. (2018). *Internet of Things (IoT) Systems*. Cham : Springer International Publishing.
- UIT. (2012). *Visión general de la Internet de las cosas*. Ginebra.
- Wireshark. (2020). *Wireshark*. Recuperado el 12 de Abril de 2020, de <https://www.wireshark.org/download/docs/user-guide.pdf>

ANEXOS



Los Juglares 1-17 y Francisco de Orellana
 PBX: (+593) 7 288 2288 ■ Fax: (+593) 7 281 7800
 www.datalights.com.ec
 info@datalights.com.ec
 P.O.BOX: 010201 ■ Cuenca - Ecuador

Cuenca, 25 de agosto del 2020

CARTA DE CONFORMIDAD

En calidad de Gerente Departamental de la empresa DataLights Cia. Ltda. certifico que:

El señor ORTEGA VINTIMILLA HUGO GABRIEL portador de la cédula de ciudadanía N° 1804656708, estudiante de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga, realizó el proyecto de titulación denominado. "DESARROLLO DE UNA PRUEBA DE CONCEPTO FUNCIONAL APLICABLE AL INTERNET INDUSTRIAL DE LAS COSAS(IIIOT) UTILIZANDO SISTEMAS EMBEBIDOS DE NATIONAL INSTRUMENTS, LENGUAJE DE NIVEL DE ABSTRACCION MEDIO Y UNA PLATAFORMA DE SERVICIOS EN LA NUBE PARA PROYECCION DE APLICABILIDAD TANTO ACADÉMICA COMO PROFESIONAL EN DATALIGHTS DISTRIBUIDOR AUTORIZADO DE NATIONAL INSTRUMENTS" en el departamento de Automatización a partir del 3 de febrero del 2020 hasta el 3 de agosto del 2020 con una duración de 6 meses.

El proyecto se encuentra dentro de los lineamientos establecidos por la Empresa DataLights Cia. Ltda. Demostrando responsabilidad y cumplimiento de los objetivos. Es todo cuanto puedo certificar.

PAUL
 SANTIAGO
 ORELLANA
 VINTIMILLA

Atentamente
 Firmado digitalmente por PAUL
 SANTIAGO ORELLANA VINTIMILLA
 DN: cn=PAUL SANTIAGO
 ORELLANA VINTIMILLA, o=IC
 HIGUAYO-URBANO CENTRAL DEL
 ECUADOR, ou=ENTIDAD DE
 CERTIFICACION DE
 INFORMACION EC/IC/IC/IC
 Motivo: Soy el autor de este
 documento.
 Ubicación:
 Fecha: 2020.08.28 17:08:05.00

Ing. Santiago Orellana
 GERENTE DEPARTAMENTAL
 DataLights Cia. Ltda.
 Los Juglares 1-17 y Francisco de Orellana
 Tel. (+593) 7 288 2288
 Cel. (+593) 994 11 33 44
 Cuenca – Ecuador

Cuenca, 25 de agosto del 2020

CARTA DE CONFORMIDAD

En calidad de Gerente Departamental de la empresa DataLights Cia. Ltda. certifico que:

El señor VASQUEZ DEL HIERRO NELSON ANDRÉ portador de la cédula de ciudadanía N°0503498719, estudiante de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga, realizó el proyecto de titulación denominado. "DESARROLLO DE UNA PRUEBA DE CONCEPTO FUNCIONAL APLICABLE AL INTERNET INDUSTRIAL DE LAS COSAS(IIoT) UTILIZANDO SISTEMAS EMBEBIDOS DE NATIONAL INSTRUMENTS, LENGUAJE DE NIVEL DE ABSTRACCION MEDIO Y UNA PLATAFORMA DE SERVICIOS EN LA NUBE PARA PROYECCION DE APLICABILIDAD TANTO ACADÉMICA COMO PROFESIONAL EN DATALIGHTS DISTRIBUIDOR AUTORIZADO DE NATIONAL INSTRUMENTS" en el departamento de Automatización a partir del 3 de febrero del 2020 hasta el 3 de agosto del 2020 con una duración de 6 meses.

El proyecto se encuentra dentro de los lineamientos establecidos por la Empresa DataLights Cia. Ltda. Demostrando responsabilidad y cumplimiento de los objetivos. Es todo cuanto puedo certificar.

**PAUL
SANTIAGO
ORELLANA
VINTIMILLA**

Firmado digitalmente por PAUL
SANTIAGO ORELLANA VINTIMILLA
DN: cn=PAUL SANTIAGO
Atentamente:
+59372817800
Ecuador, ou=INSTITUTO DE
CERTIFICACION DE
INFORMACION ACIBCA
Móvil: 09 91 434 40 444
Ecuador
Fecha: 2020.08.26 17:00:03.00

Ing. Santiago Orellana
GERENTE DEPARTAMENTAL
DataLights Cia. Ltda.
Los Juglares 1-17 y Francisco de Orellana
Tel. (+593) 7 288 2288
Cel. (+593) 994 11 33 44
Cuenca – Ecuador