

Aplicación de internet industrial de las cosas al monitoreo y supervisión de un proceso de manufactura simulado que integra tecnologías de comunicación heterogéneas

Lema Molina, Henry David y Morales Jordán, Carlos Daniel

Vicerrectorado de Investigación, Innovación y Transferencia de Tecnologías

Centro de Posgrados

Maestría en Electrónica Mención Redes Industriales

Trabajo de Titulación, Previo a la Obtención del Título de Magister en Electrónica Mención
Redes Industriales

Dr. Minchala Ávila, Luis Ismael

16 de mayo del 2022

Latacunga



AGENDA

Introducción, objetivos e hipótesis

Desarrollo del proyecto

Comunicación OPC UA

Virtualización de los procesos

Análisis de resultados

Conclusiones y recomendaciones

Trabajos futuros



Introducción

El IoT industrial se define como la **red de dispositivos, maquinaria y sensores** conectados entre sí y a internet, con el objetivo de recoger datos y analizarlos para aplicar esta información a una mejora en los procesos.

Las aplicaciones del IoT industrial son numerosas y han empujado a un creciente número de empresas a impulsar este nuevo paradigma para mejorar su productividad y optimizar sus gastos y beneficios.

Principales aplicaciones del IoT industrial:

- Gestión y monitoreo automatizada y remota de los equipos
- Mantenimiento Predictivo
- Aplicación de mejoras de procesos mas rápida
- Inventarios a punto
- Control de calidad
- Optimización de la cadena de suministro
- Mejoras en la seguridad de las plantas



Objetivos

GENERAL:

Implementar una solución de internet industrial de las cosas en el contexto de monitoreo y supervisión de un proceso de manufactura simulado que integra tecnologías de comunicación heterogéneas.

ESPECÍFICOS:

- Investigar y seleccionar el software de simulación apropiado para el desarrollo del proceso industrial simulado.
- Desarrollar un proceso industrial simulado, integrando características de un proceso real.
- Diseñar una interfaz hombre máquina (HMI, por sus siglas en inglés), a través de dashboards.
- Realizar la comunicación del PLC con la plataforma IoT mediante el protocolo de comunicación OPC-UA.
- Implementar IoT con la plataforma de internet Thinger.io para manejo de la planta desde un móvil.
- Evaluar el control de la planta simulada y la adquisición de datos para su gestión desde la aplicación móvil.

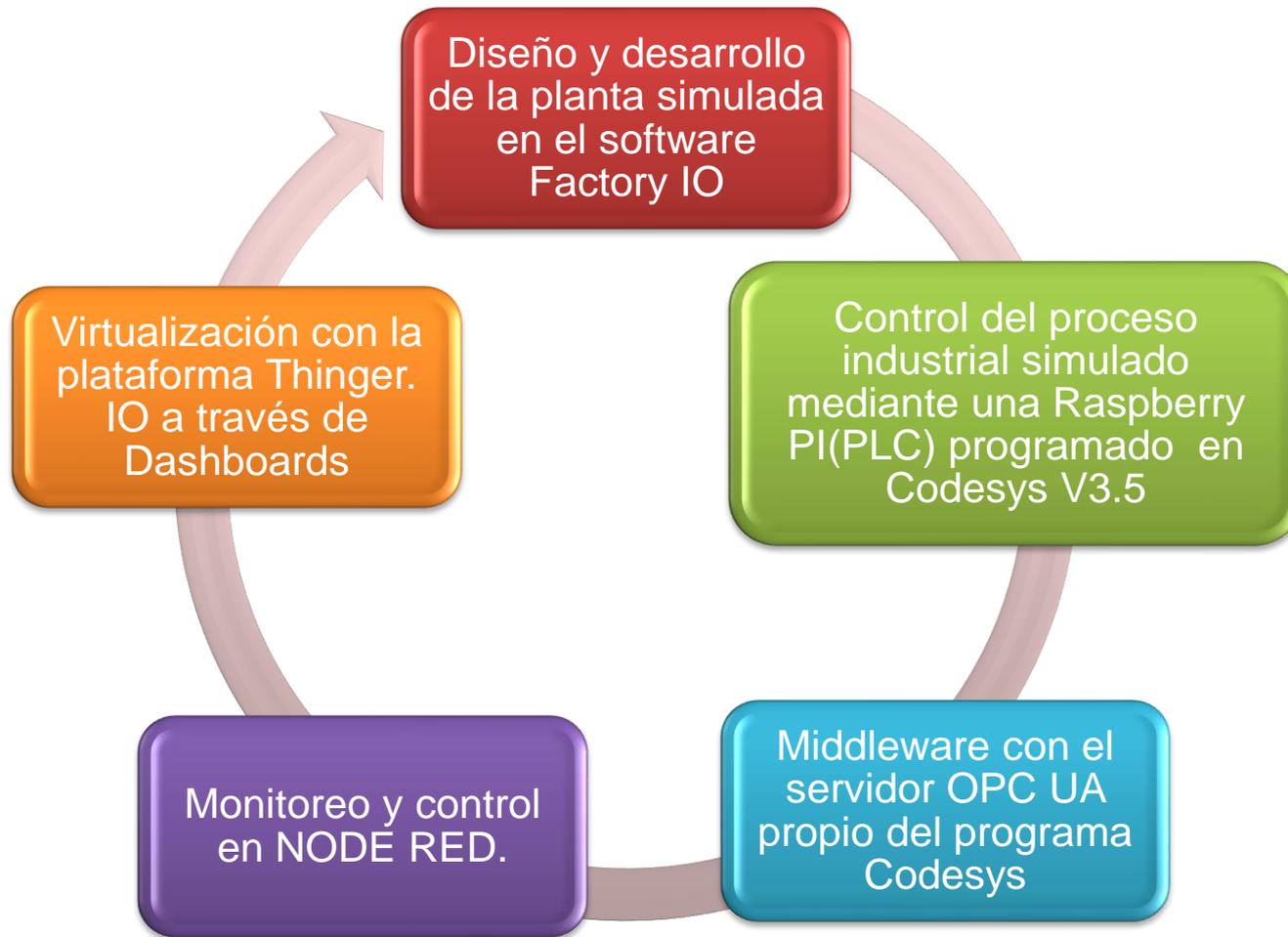


Hipótesis

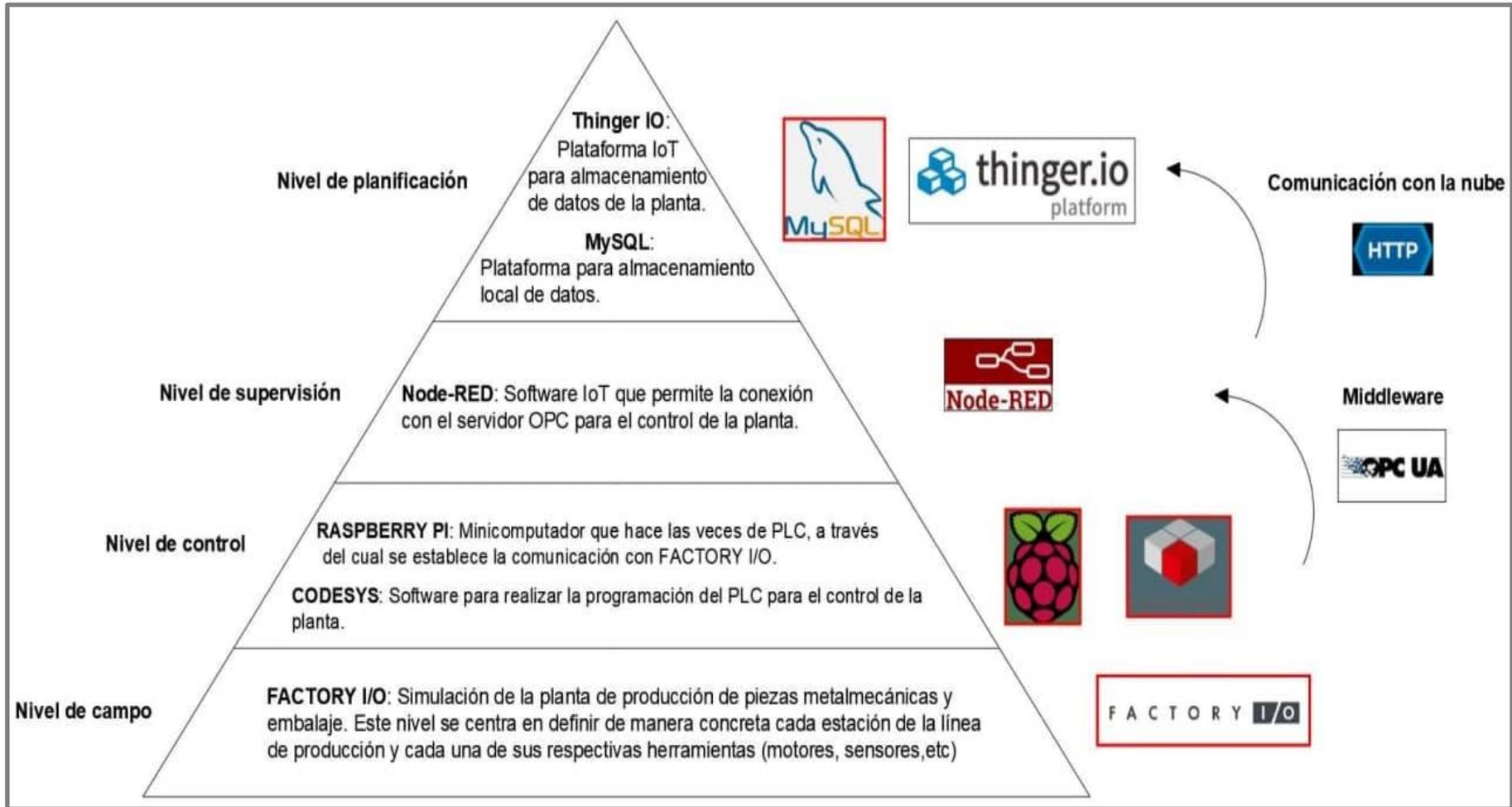
La implementación del IOT Industrial en los procesos industriales facilita la comunicación de sistemas de automatización industrial con interfaces de comunicación heterogéneos



DESARROLLO DEL PROYECTO



ARQUITECTURA EN BASE A PIRAMIDE CIM



DESCRIPCIÓN DEL FUNCIONAMIENTO DE LA PLANTA

PRODUCCIÓN

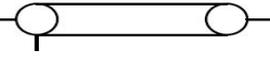
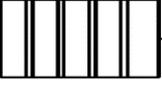
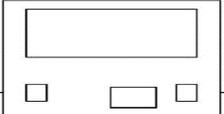
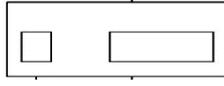
- Fabricación
- Embalaje y empaquetado

FACTORY IO

- Procesos enumerados y diferenciados
- Cuatro procesos



SIMBOLOGIA SEGÚN NORMA ISA/ANSI 5.1

Categoría	Descripción	Simbología
Elementos primarios	State Sensor (YE) montado en campo	
	Motor montado en campo	
Control compartido	Control Station (LK) Ubicación accesible normalmente al operario	
Componentes adicionales	Banda transportadora con rodillos giratorios	
	Banda transportadora lisa	
	Ascensor	
	CNC	
	Brazo robótico	



CONFIGURACIÓN DEL DISPOSITIVO EN FACTORY IO

The screenshot displays the 'CONFIGURATION' screen in Factory IO. On the left, a sidebar lists various device types, with 'OPC Client DA/UA' highlighted in red. The main area is titled 'Server' and contains the following fields and buttons:

- Host Name:** An empty text input field.
- BROWSE SERVERS:** A button to search for servers.
- OPC Server:** A dropdown menu currently showing 'CoDeSys.OPC.DA'.
- Items/Nodes:** A section with a 'BROWSE' button.
- Limit:** A text input field containing the value '256'.
- Filter names that start with:** A text input field containing 'No filter'.
- Filter names that contain:** A text input field containing 'No filter'.
- DEFAULT:** A button to reset settings.

Enlace del PLC por medio de comunicación OPC DA/UA

La Raspberry PI 4 mantiene el principio de heterogeneidad realizando las veces de PLC. El procesamiento es suficiente para el funcionamiento de la planta.



Configuración del entorno de programación Codesys V.3.5

- Instalación de paquetes y licencias adicionales para Raspberry Pi y Servidor OPC DA
- Configuración de la comunicación con el PLC

The screenshot displays the Codesys V.3.5 environment. On the left, the 'Dispositivos' tree shows the project structure, including 'Device (CODESYS Control for Raspberry Pi SL)' and 'Lógica PLC'. The central editor shows a ladder logic program with variables like 'Temporizador1' and 'Temporizador2'. A 'Herramientas' menu is open, listing options such as 'Administrador de paquetes...', 'Repositorio de bibliotecas...', and 'Update Raspberry Pi'. On the right, the 'Configuración de comunicación' window is active, showing a network diagram and configuration parameters for 'Gateway-1' and 'PLC1 (activo)'. The configuration details include IP-Address: localhost, Port: 1217, Nombre del dispositivo: PLC1, Dirección del dispositivo: 0301.A066, ID del target: 0000 0010, Tipo de sistema destino: 4102, Fabricante del sistema de destino: 3S - Smart Software Solutions GmbH, and Versión del sistema de destino: 4.0.0.0.



Configuración del servidor OPC DA

OPCConfig - C:\ProgramData\CODESYSOPC\OPCServer.ini

File Edit ?

Server

- PLC_GW3
 - Connection

Settings for PLC_GW3

Interface: GATEWAY3

Project name:

Timeout (ms): 10000

Number of Tries: 3

Buffer Size (Byte): 0

Wait Time (s): 10

Reconnect Time (s): 15

Active:

Motorola Byteorder:

No Login-Service:

Logging

Enable logging (Defaultevents)

Log Additional Events

Add Debug Events (slow)

La adquisición de los datos es realizada a través de un *middleware*, para lo cual haremos uso de un servidor con el estándar de comunicación OPC-DA/UA.

En la configuración de la conexión, la herramienta asigna la dirección IP de manera dinámica.

PLC address

Gateway

IP address: localhost

Port: 1217

OK

PLC name or address: PLC1

Use TCP/IP blockdriver:

IP address of PLC: localhost

IP port of PLC: 11740



DESIGNACIÓN DE VARIABLES EN CODESYS

Administrador de bibliotecas Configuración de símbolos

Ver Configuración Herramientas

Configuración modificada de símbolos se transmitirá con la siguiente descarga o cambio en línea.

bolos	Derechos de acceso	Máximo	Atributo	Tipo
Variables_Globales				
<input checked="" type="checkbox"/> BRAZOUNOPOS				BOOL
<input checked="" type="checkbox"/> CONTADORCAJAS				WORD
<input checked="" type="checkbox"/> CONTADORCNC1				WORD
<input checked="" type="checkbox"/> CONTADORCNC2				WORD
<input checked="" type="checkbox"/> EMISORPAQUETES				BOOL
<input checked="" type="checkbox"/> FINCARRERAASCENSOR				BOOL
<input checked="" type="checkbox"/> INICIOCNC				BOOL
<input checked="" type="checkbox"/> LIMITELEVADOR				BOOL
<input checked="" type="checkbox"/> MOTOR10P10				BOOL
<input checked="" type="checkbox"/> MOTOR10P2				BOOL
<input checked="" type="checkbox"/> MOTOR11P10				BOOL
<input checked="" type="checkbox"/> MOTOR11P2				BOOL
<input checked="" type="checkbox"/> MOTOR12P10				BOOL
<input checked="" type="checkbox"/> MOTOR12P2				BOOL
<input checked="" type="checkbox"/> MOTOR13P10				BOOL
<input checked="" type="checkbox"/> MOTOR13P2				BOOL
<input checked="" type="checkbox"/> MOTOR14P10				BOOL
<input type="checkbox"/> MOTOR15P10				BOOL
<input type="checkbox"/> MOTOR16P10				BOOL
<input checked="" type="checkbox"/> MOTOR1P1				BOOL
<input checked="" type="checkbox"/> MOTOR1P10				BOOL
<input checked="" type="checkbox"/> MOTOR1P2				BOOL

Tesis.project - CODESYS

Archivo Edición Ver Proyecto Compilar En línea Depuración Herramientas Ventana Ayuda

Dispositivos

La configuración modificada

Soportar características OPC UA

Símbolos

- Constantes
- IoConfig_Globales
- PLC_PRG
- Variables_Globales

Incluir comentarios en XML

Incluir los indicadores de nodo en XML

Configurar comentarios y atributos...

Configurar sincronización con tareas IEC...

Diseño optimizado

Utilizar por defecto espacios de nombres vacíos (compatibilidad con V2)

Activar acceso E/S directo

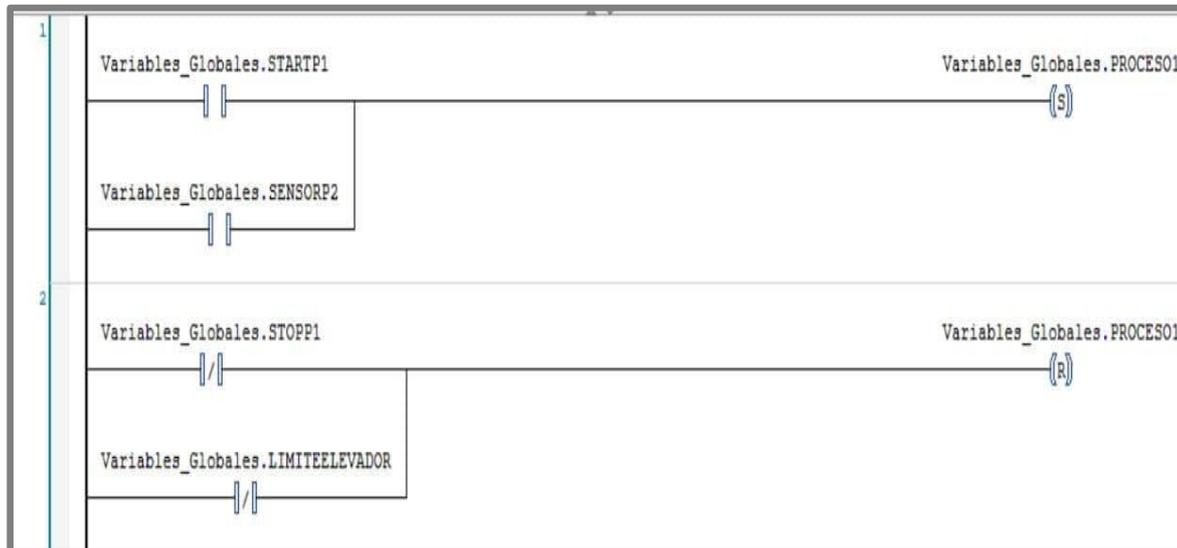
Incluir la información de llamada en XML

Habilitar conjuntos de símbolos

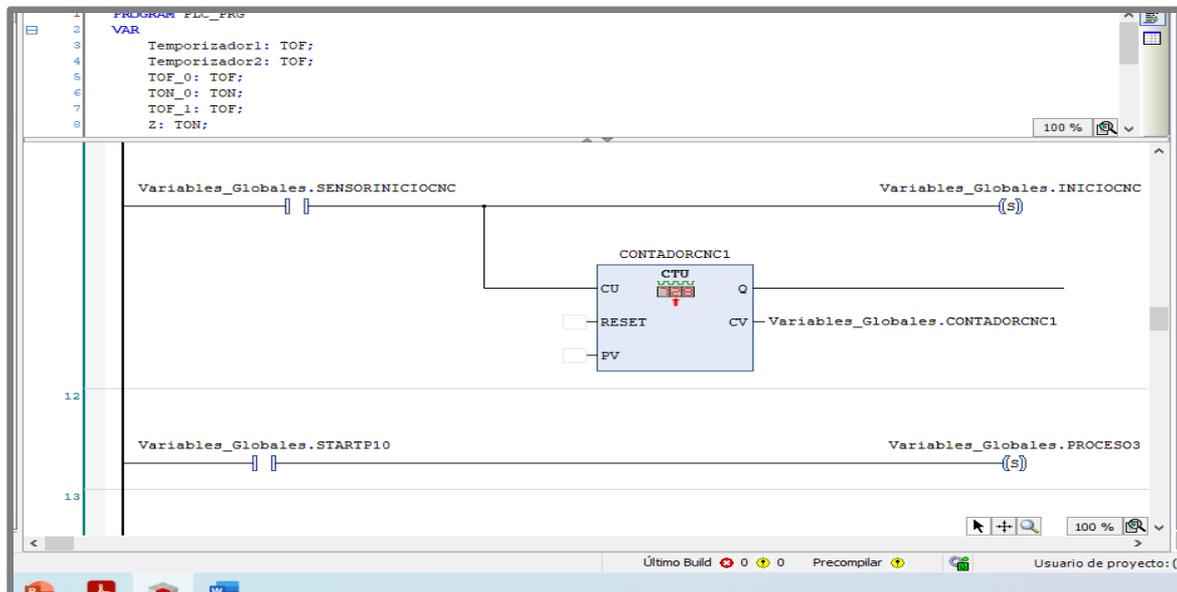
Creación de variables globales con características OPC UA para ser administradas por el servidor y sus clientes OPC.



PROGRAMACIÓN EN CODESYS



La programación es realizada en un ambiente gráfico de tipo escalera.



El control de los sensores y actuadores de la planta simulada en Factory IO se realiza mediante el uso de los diferentes módulos como SET/RESET, contadores, TON, TOF, y varios otros disponibles dentro de Codesys V 3.5.



TABLA DE VARIABLES EN FACTORY IO

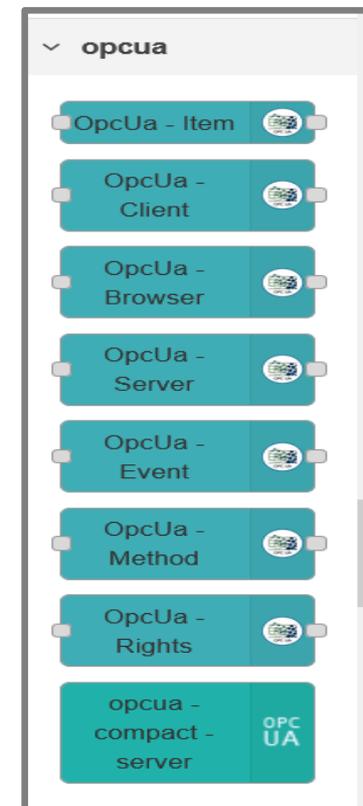
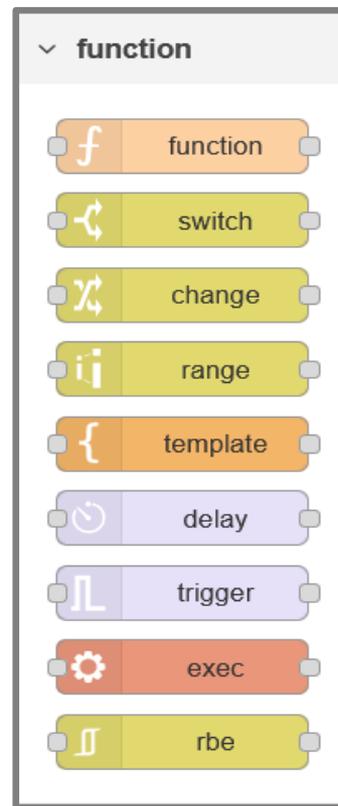
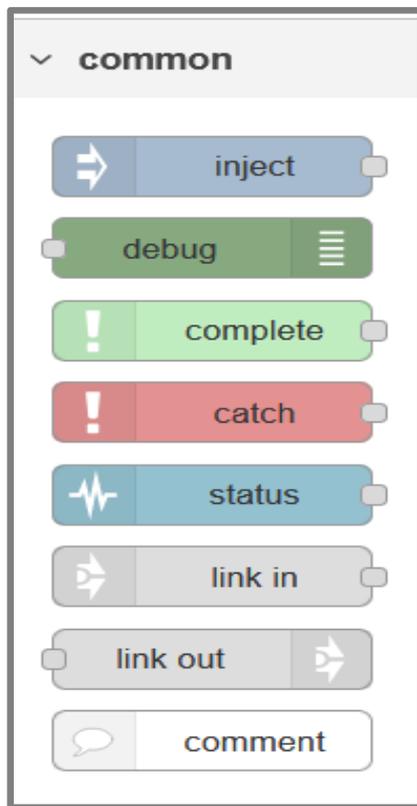
Server: CoDeSys.OPC.DA (79)		
	BRAZOUNOPOS	Brazo1pos
	EMISORPAQUETES	EmisorPaquetes
FinCarreraAscensor	...RRERAASCENSOR	
	INICIOCNC	InicioCNC
Limite Elevador	LIMITELEVADOR	
	MOTOR10P10	Motor10P10
	MOTOR10P2	Motor10P2
	MOTOR11P10	Motor11P10
	MOTOR11P2	Motor11P2
	MOTOR12P10	Motor12P10
	MOTOR12P2	Motor12P2
	MOTOR13P10	Motor13P10
	MOTOR13P2	Motor13P2
	MOTOR14P10	Motor14P10
	MOTOR1P1	Motor1P1
	MOTOR1P10	Motor1P10
	MOTOR1P2	Motor1P2
	MOTOR1P5	Motor1P5
	MOTOR1PEMPA	Motor1P4
	MOTOR2P1	Motor2P1
	MOTOR2P10	Motor2P10
	MOTOR2P2	Motor2P2
	MOTOR2P5	Motor2P5
	MOTOR2PEMPA	Motor2P4
	MOTOR3P1	Motor3P1
	MOTOR3P10	Motor3P10
	MOTOR3P2	Motor 3P2
	MOTOR3P5	Motor3P5
	MOTOR4P1	Motor 4P1
	MOTOR4P10	Motor4P10
	MOTOR4P2	Motor4P2
	MOTOR4P5	Motor4P5

Factory IO desde su instancia de cliente OPC-UA reconoce el servidor y establece la conexión de variables en Codesys con los sensores y actuadores en este programa.

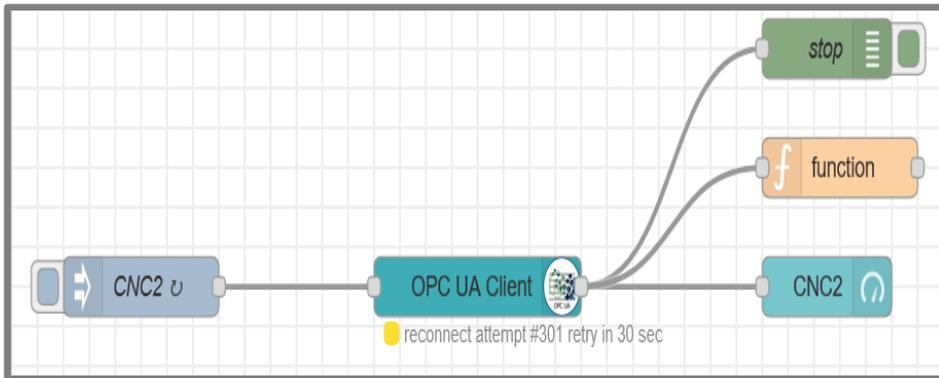


NODOS O HERRAMIENTAS DE Node-RED

- Herramienta basada en navegador web
- Editor de flujo para enlazar nodos para su comunicación
- Descarga de nodos de comunicación, almacenamiento, telegram, paneles de visualización, etc.



COMUNICACIÓN PLC- Node RED



Se establece el flujo de información a través de la conexión de los nodos de Node RED

Edit OpcUa-Client node

Delete Cancel Done

Properties

Endpoint:

Action:

Certificate:

Local certificate file with absolute path:

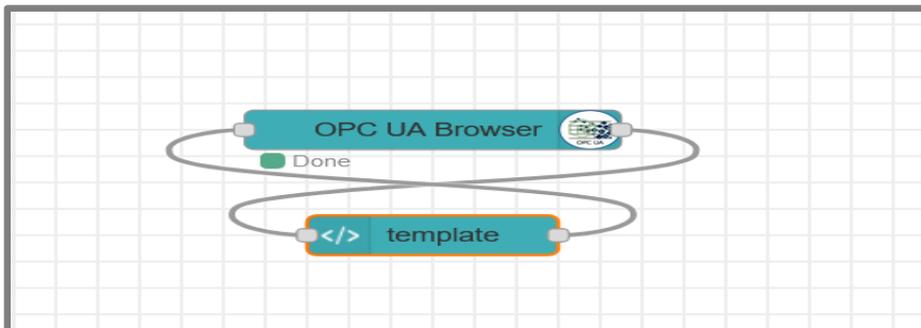
Local private key file with absolute path:

Name:

En las propiedades del nodo OPC Ua-Client se asigna la dirección del Controlador (Raspberry Pi) como Endpoint y se elige la acción de lectura (*READ*) o escritura (*WRITE*) para la visualización o manipulación de las direcciones OPC de las variables del PLC



DIRECCIONES DEL NODO OPC-UA



TESIS

-- Objets Root --

Press to Update:

Browse

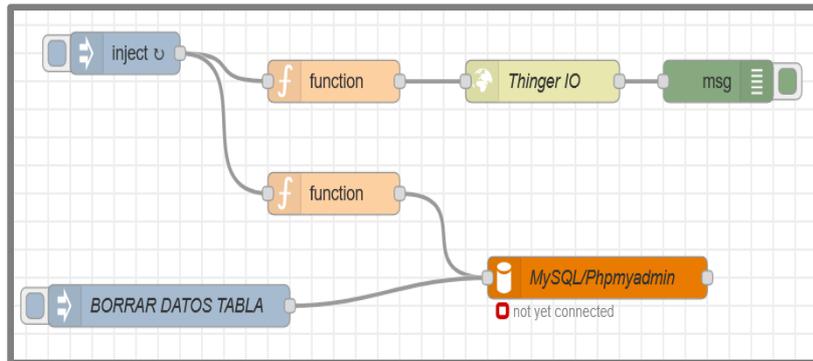
- BRAZOUNOPOS: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.BRAZOUNOPOS
- CONTADORCAJAS: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.CONTADORCAJAS
- CONTADORCNC1: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.CONTADORCNC1
- CONTADORCNC2: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.CONTADORCNC2
- EMISORPAQUETES: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.EMISORPAQUETES
- FINCARRERAASCENSOR: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.FINCARRERAASCENSOR
- INICIOCNC: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.INICIOCNC
- LIMITELEVADOR: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.LIMITELEVADOR
- MOTOR10P10: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.MOTOR10P10
- MOTOR10P2: ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.MOTOR10P2

Uso del nodo Opc-Ua Browser para determinar las direcciones Opc de las variables globales del PLC



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

CONFIGURACIÓN DEL NODO MYSQL EN Node RED



Edit mysql node > Edit MySQLdatabase node

Delete Cancel Update

⚙ Properties ⚙ 📄

🌐 Host	127.0.0.1
🔌 Port	3306
👤 User	root
🔒 Password	
🗄 Database	datafact
🕒 Timezone	
🗂 Charset	UTF8
📄 Name	Name

flujo de información a través de la conexión de los nodos de Node RED para la plataforma Thingier IO y MySQL

Se define la dirección de *localhost*, puerto y usuario de la base de datos. El envío de información desde Node-RED se realiza cada 2 segundos, tiempo que ha sido estimado como adecuado de acuerdo con los requerimientos de esta planta de producción.



ENVÍO DE DATOS A LA PLATAFORMA THINGER IO

Edit http request node

Delete Cancel Done

Properties

Method: POST

URL: https://backend.thinger.io/v3/users/Caylos82/devic

Enable secure (SSL/TLS) connection

Use authentication

Type: bearer authentication

Token:

Enable connection keep-alive

Use proxy

Return: a UTF-8 string

Name: Thinger IO

El envío de datos se ejecuta desde el cliente implementado en Node-RED, mediante el protocolo de transferencia de hipertexto (HTTP) desde un nodo *http request*.



ENVÍO DE INFORMACIÓN DESDE NODE RED

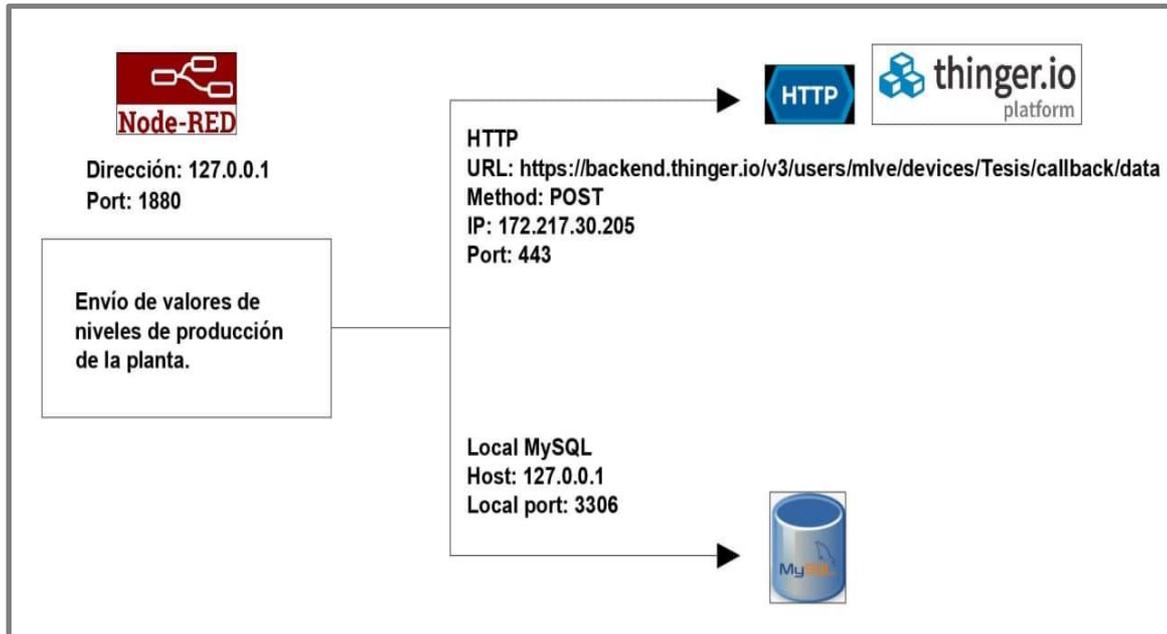


Diagrama explicativo que detalla la información relevante sobre el envío de datos desde Node-RED hacia las plataformas de almacenamiento de información, MySQL y Thinger.io



CONFIGURACIÓN DEL NODO TELEGRAM RECEIVER EN NODE RED

Edit receiver node > Edit telegram bot node

Delete Cancel Update

⚙ Properties

🔍 Bot-Name CFabricabot

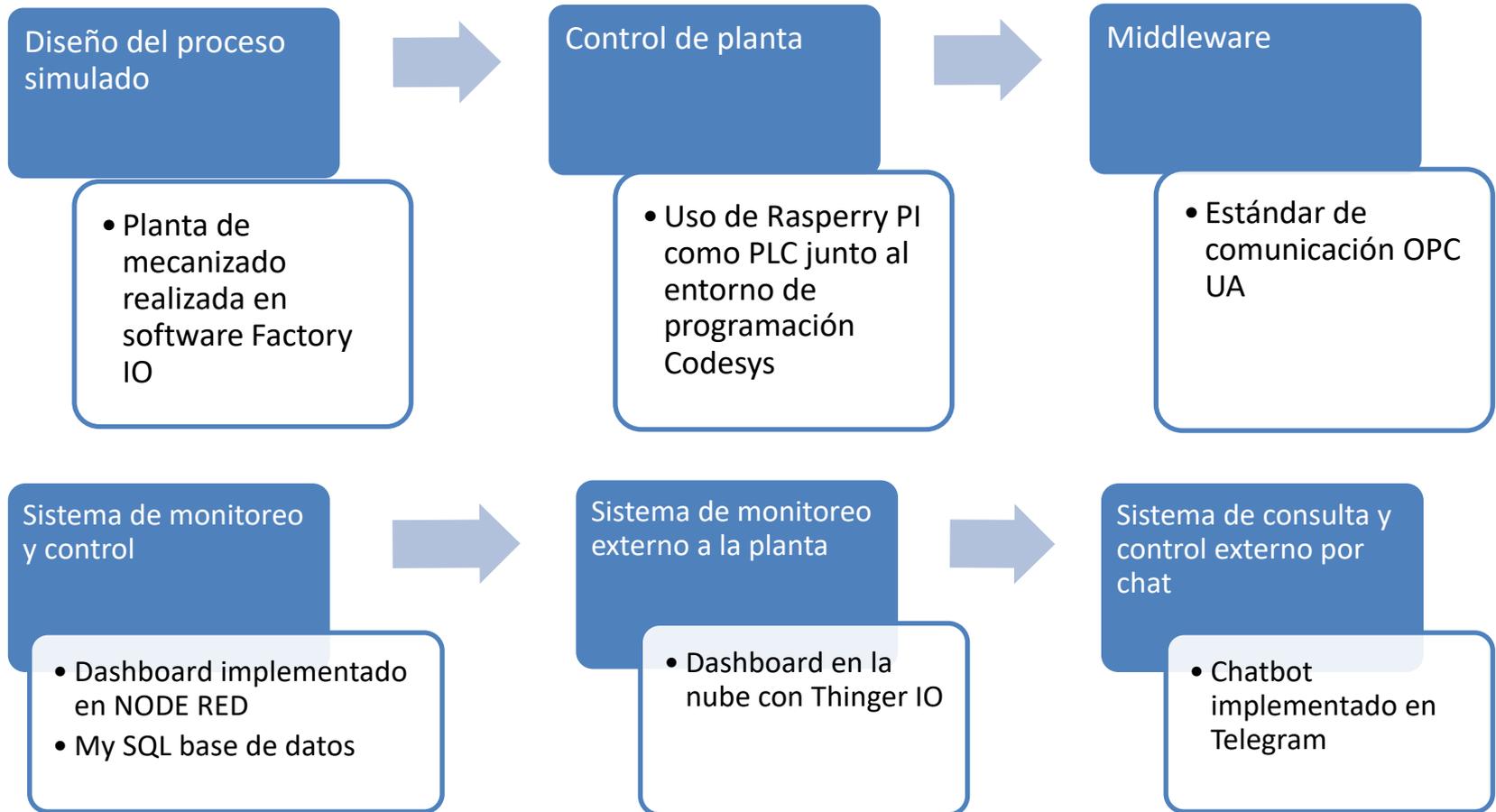
🔍 Token 1938484188:AAFNqaNHXo7MCxUk6wQcgv11aJXyIK9b6JQ

Tip: If you don't have a token yet, you can create a new one here:
[@BotFather](#).

Crea un *token* que sirve para comunicar el lenguaje de programación Python, bajo el que envía la información de Node-RED hacia Telegram. La conectividad entre Python y Telegram se efectúa a través de una API



RESULTADOS

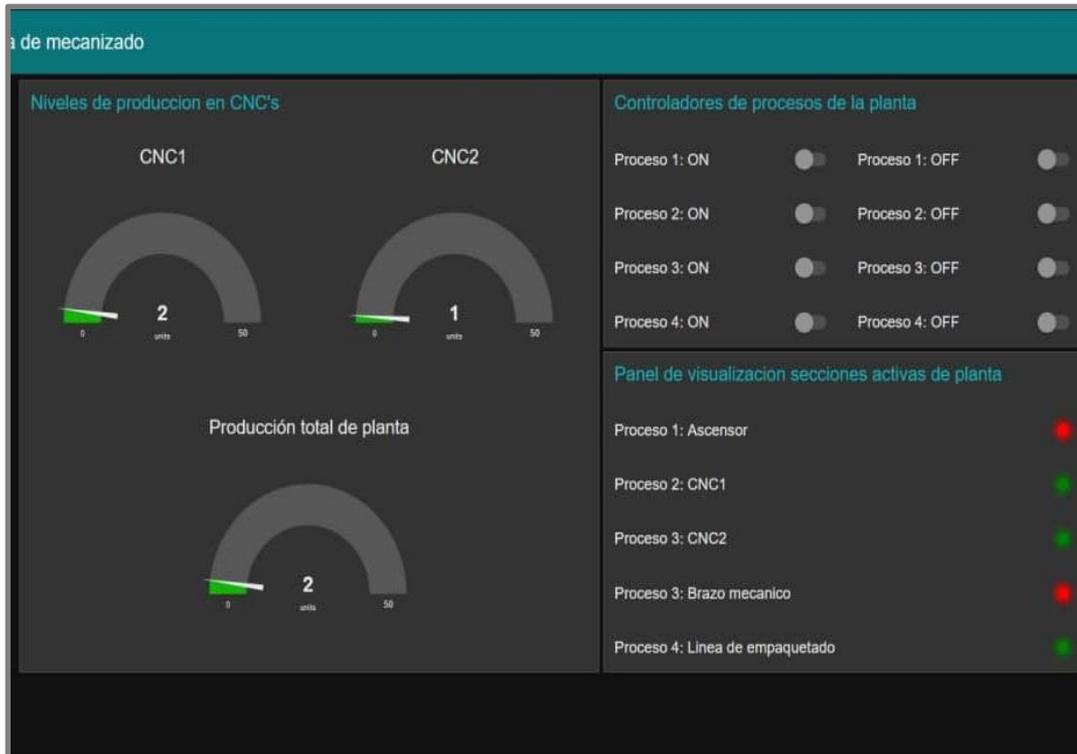




La plataforma Factory IO, utilizada para el diseño de la planta de producción, ostenta ventajas significativas con respecto a otros *softwares* de simulación de ambientes industriales, como PROModel, FreeCAD, ARENA simulation, etc. Cuenta con diversos escenarios previamente configurados para utilizarlos en distintas aplicaciones y se caracteriza por ser una plataforma amigable con el usuario.



SISTEMA DE MONITOREO Y CONTROL DE LA PLANTA

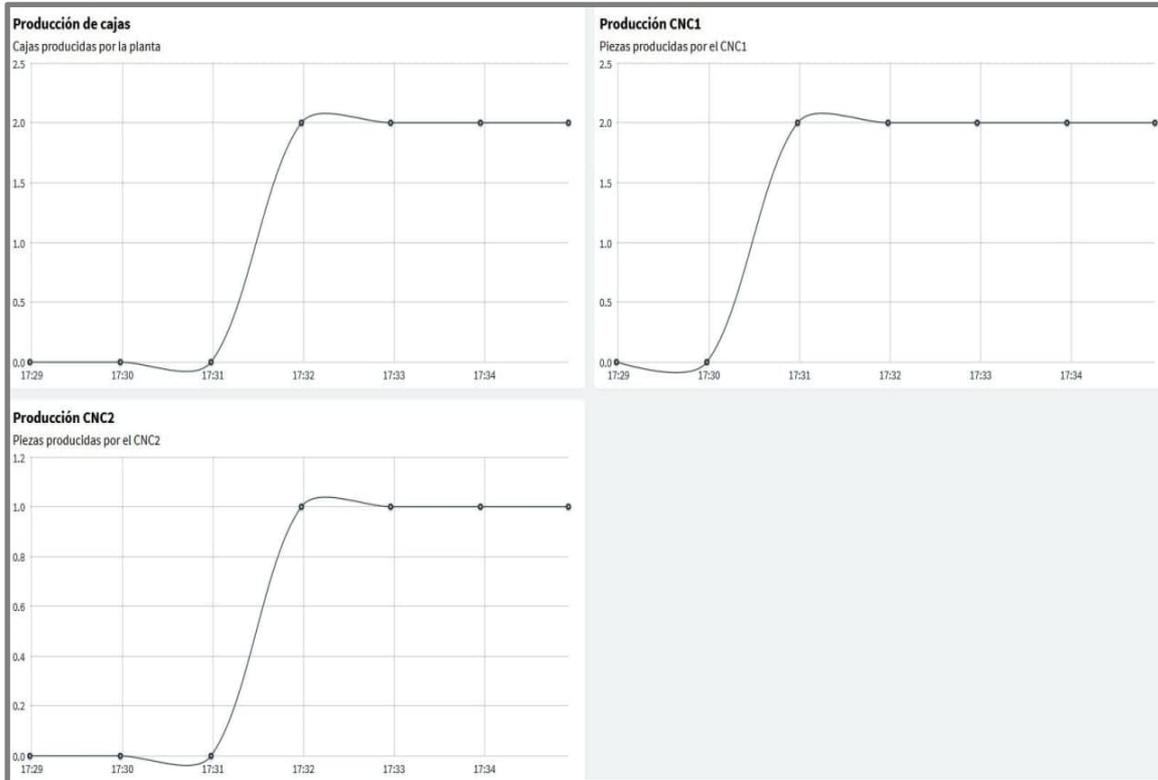


Los sistemas de interacción con la planta implementados son tres y se listan a continuación.

- Sistema de monitoreo y control en planta: *dashboard* implementado en Node-RED.
- Sistema de monitoreo externo a la planta: *dashboard* implementado de la nube de almacenamiento Thinger.io.
- Sistema de consulta y control externo por *chat*: *chatbot* implementado en Telegram.



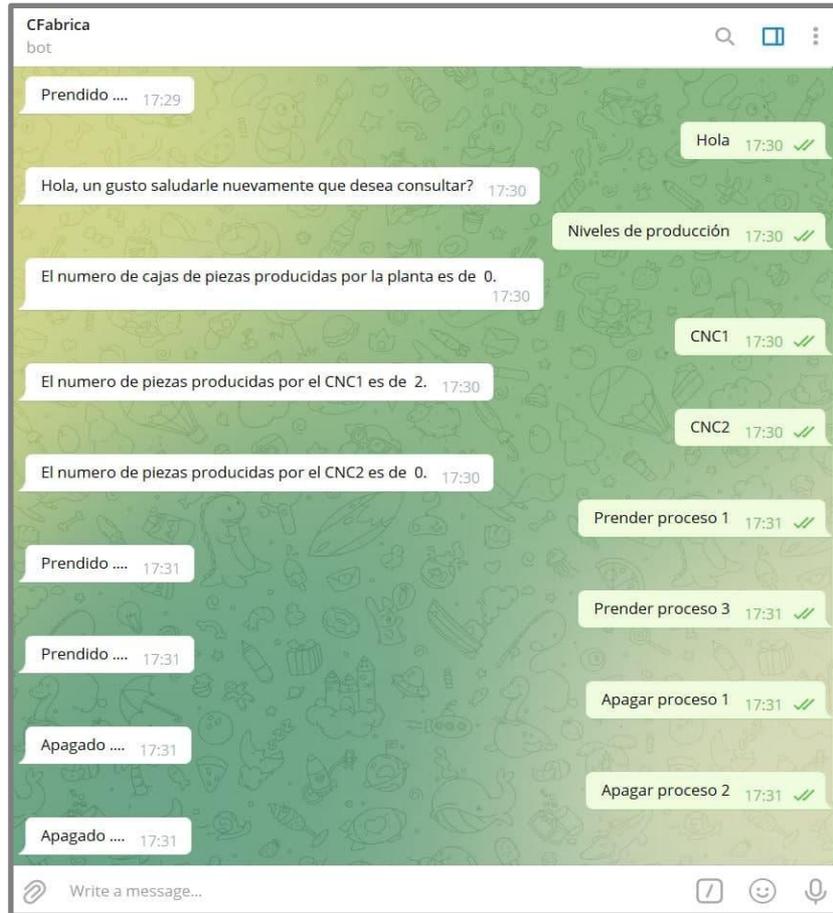
SISTEMA DE MONITOREO EXTERNO EN THINGER IO



En el sistema de monitoreo externo a la planta se presentan los valores de fabricación de manera gráfica. Debido a que no cuenta con la opción de controlar la planta, esta plataforma se concibe como un sistema de monitoreo desde la nube.



SISTEMA DE CONSULTA Y CONTROL EXTERNO POR CHAT A TRAVEZ DE TELEGRAM



El *chatbot*, destinado para monitoreo y control externo está disponible para consulta por cualquier personal fuera de planta, con la finalidad de conocer información en el instante de la consulta y controlar el estado de los procesos.



Análisis de comunicación y protocolos de los sistemas de interacción con la planta simulada

PAQUETE SERVIDOR OPC UA A NODE RED EN WIRESHARK

15451	62.168099	192.168.0.103	192.168.0.108	OpcUa	232 UA Secure Conversation Message: ReadRequest
15452	62.168231	192.168.0.103	192.168.0.108	OpcUa	231 UA Secure Conversation Message: ReadRequest
15453	62.168342	192.168.0.103	192.168.0.108	OpcUa	237 UA Secure Conversation Message: ReadRequest
15454	62.168450	192.168.0.103	192.168.0.108	OpcUa	238 UA Secure Conversation Message: ReadRequest
15455	62.168566	192.168.0.103	192.168.0.108	OpcUa	235 UA Secure Conversation Message: ReadRequest
15456	62.168667	192.168.0.103	192.168.0.108	OpcUa	234 UA Secure Conversation Message: ReadRequest
15457	62.168766	192.168.0.103	192.168.0.108	OpcUa	234 UA Secure Conversation Message: ReadRequest
15458	62.168865	192.168.0.103	192.168.0.108	OpcUa	233 UA Secure Conversation Message: ReadRequest
15459	62.168879	192.168.0.108	192.168.0.103	UDP	102 1740 → 1743 Len=60
15460	62.168965	192.168.0.103	192.168.0.108	OpcUa	236 UA Secure Conversation Message: ReadRequest
15461	62.169062	192.168.0.103	192.168.0.108	OpcUa	237 UA Secure Conversation Message: ReadRequest
15462	62.171295	192.168.0.108	192.168.0.103	OpcUa	133 UA Secure Conversation Message: ReadResponse

▼ Frame 1753: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{DB5CAEC1-39E3-4A90-95CD-374E8F...}

- > Interface id: 0 (\Device\NPF_{DB5CAEC1-39E3-4A90-95CD-374E8F...})
- Encapsulation type: Ethernet (1)
- Arrival Time: Jan 6, 2022 14:59:50.161164000 Hora est. Pacifico, Sudamérica
- [Time shift for this packet: 0.00000000 seconds]
- Epoch Time: 1641499190.161164000 seconds
- [Time delta from previous captured frame: 0.013200000 seconds]
- [Time delta from previous displayed frame: 0.013200000 seconds]
- [Time since reference or first frame: 7.022475000 seconds]
- Frame Number: 1753
- Frame Length: 55 bytes (440 bits)

la IP para Node-RED, 192.168.0.103, corresponde a la IP del ordenador ya que el cliente se aloja en éste. Se verifica el uso correcto del protocolo OPC UA.



Análisis de comunicación y protocolos de los sistemas de interacción con la planta simulada

PAQUETE DE Node RED AL CHATBOT EN WIRESHARK

No.	Time	Source	Destination	Protocol	Length	Info
70295	151.730246	192.168.0.103	193.123.28.242	UDP	1114	62065 → 8801 Len=1072
70296	151.730389	192.168.0.103	193.123.28.242	UDP	1114	62065 → 8801 Len=1072
70298	151.741290	192.168.0.103	193.123.28.242	UDP	1114	62065 → 8801 Len=1072


```
> Frame 70296: 1114 bytes on wire (8912 bits), 1114 bytes captured (8912 bits) on interface \Device\NPF_{DB5CAEC1-...}
> Ethernet II, Src: IntelCor_fb:a9:55 (38:00:25:fb:a9:55), Dst: Tp-LinkT_24:ed:7c (ac:84:c6:24:ed:7c)
v Internet Protocol Version 4, Src: 192.168.0.103, Dst: 193.123.28.242
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1100
  Identification: 0xe0c1 (57537)
  > Flags: 0x00
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: UDP (17)
```

El *bot* de consulta se aloja en un servidor propio de Telegram, por lo tanto, corresponde a una red externa, con la IP 193.123.28.242.



Análisis de comunicación y protocolos de los sistemas de interacción con la planta simulada

PAQUETE DE Node RED A THINGER IO EN WIRESHARK

```
14337 21.850127 192.168.0.103 34.107.221.82 HTTP 378 GET /success.txt?ipv4 HTTP/1.1
14363 21.880006 34.107.221.82 192.168.0.103 HTTP 274 HTTP/1.1 200 OK (text/plain)
14402 22.012804 192.168.0.103 192.168.0.103 NCSO 1000 Response

> Frame 14337: 378 bytes on wire (3024 bits), 378 bytes captured (3024 bits) on interface \Device\NPF_{0B5CAEC1-39E3-4A90-95CD-374E8FC800F3}, id 0
> Ethernet II, Src: IntelCor_fb:a9:55 (38:00:25:fb:a9:55), Dst: Tp-LinkT_24:ed:7c (ac:84:c6:24:ed:7c)
> Internet Protocol Version 4, Src: 192.168.0.103, Dst: 34.107.221.82
> Transmission Control Protocol, Src Port: 63926, Dst Port: 80, Seq: 325, Ack: 221, Len: 324
v Hypertext Transfer Protocol
  > GET /success.txt?ipv4 HTTP/1.1\r\n
  Host: detectportal.firefox.com\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101 Firefox/95.0\r\n
  Accept: */*\r\n
  Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3\r\n
  Accept-Encoding: gzip, deflate\r\n
  Connection: keep-alive\r\n
  Pragma: no-cache\r\n
  Cache-Control: no-cache\r\n
  \r\n
```

En este caso, en la nube se aloja en un servidor propio de Thinger.io; por lo tanto, corresponde a una red externa, con la IP 34.107.221.82. Se verifica el uso correcto del protocolo HTTP



Análisis del envío de información a la base de datos local y en la nube de almacenamiento en Thinger IO

BASE DE DATOS THINGER IO

Date	Cajas	Cnc1	Cnc2
2022-01-06T15:33:27.098Z	18	12	12
2022-01-06T15:32:27.550Z	18	12	12
2022-01-06T15:31:27.158Z	18	12	12
2022-01-06T15:30:28.389Z	18	12	12
2022-01-06T15:29:29.227Z	18	12	12
2022-01-06T15:28:27.987Z	18	12	12
2022-01-06T15:27:28.742Z	18	12	12
2022-01-06T15:13:23.310Z	14	12	8
2022-01-06T15:12:23.362Z	13	12	7
2022-01-06T15:11:23.105Z	12	12	6
2022-01-06T15:10:24.805Z	12	12	5
2022-01-06T15:09:25.095Z	10	12	4
2022-01-06T15:08:26.182Z	9	12	3
2022-01-06T15:07:26.762Z	9	11	3
2022-01-06T15:06:28.079Z	9	10	1
2022-01-06T15:05:27.749Z	4	8	1

BASE DE DATOS MY SQL

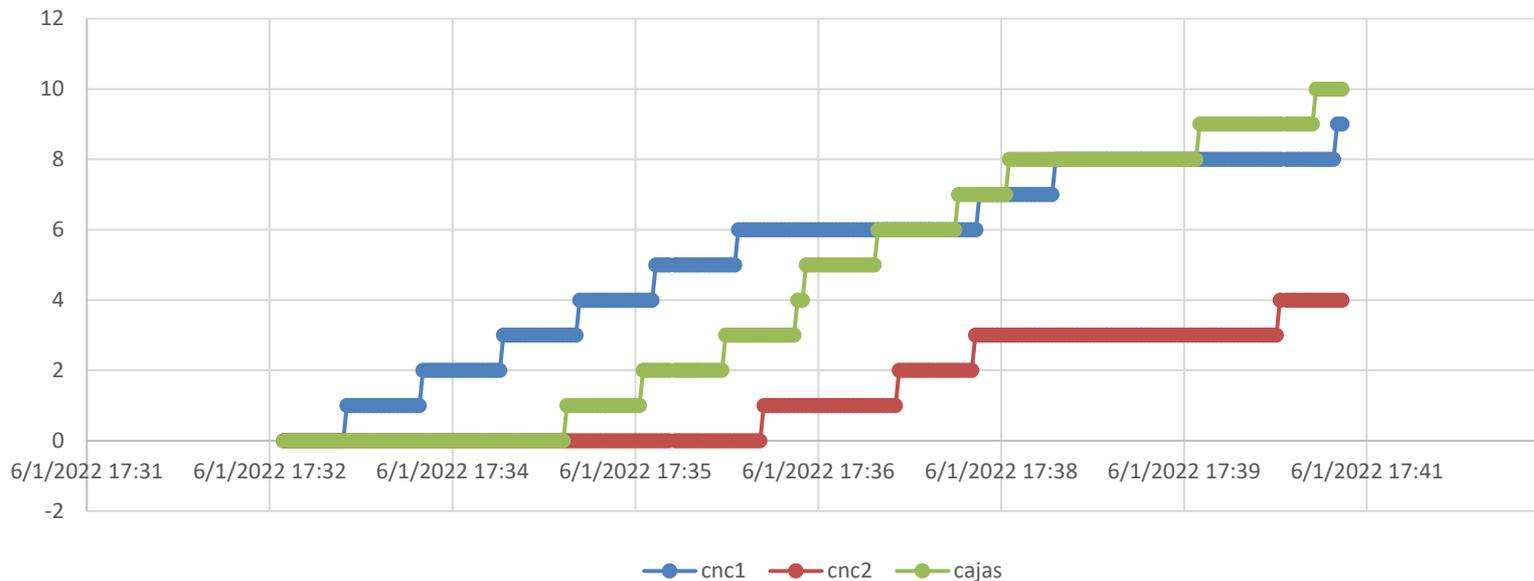
p_id	Atime	cnc1	cnc2	cajas
1840	2022-01-06 16:41:58	2	0	0
1841	2022-01-06 16:42:00	2	0	0
1842	2022-01-06 16:42:02	2	0	0
1843	2022-01-06 16:42:04	2	0	0
1844	2022-01-06 16:42:06	2	0	0
1845	2022-01-06 16:42:08	2	0	0
1846	2022-01-06 16:42:10	2	0	0
1847	2022-01-06 16:42:12	2	1	0
1848	2022-01-06 16:42:14	2	1	0
1849	2022-01-06 16:42:16	2	1	0
1850	2022-01-06 16:42:18	2	1	0
1851	2022-01-06 16:42:20	2	1	0
1852	2022-01-06 16:42:22	3	1	0
1853	2022-01-06 16:42:24	3	1	0
1854	2022-01-06 16:42:26	3	1	1
1855	2022-01-06 16:42:28	3	1	1
1856	2022-01-06 16:42:30	3	1	1
1857	2022-01-06 16:42:32	3	1	1
1858	2022-01-06 16:42:34	3	1	1

Se encuentra que la base de datos local recibe 1802 paquetes y la nube de información recibe sólo 60. Esto se debe a que la nube de información Thinger.io actualiza los datos que Node-RED envía cada minuto



ANÁLISIS DE INFORMACIÓN ALMACENADA DE FORMA LOCAL

Datos almacenados en MySQL

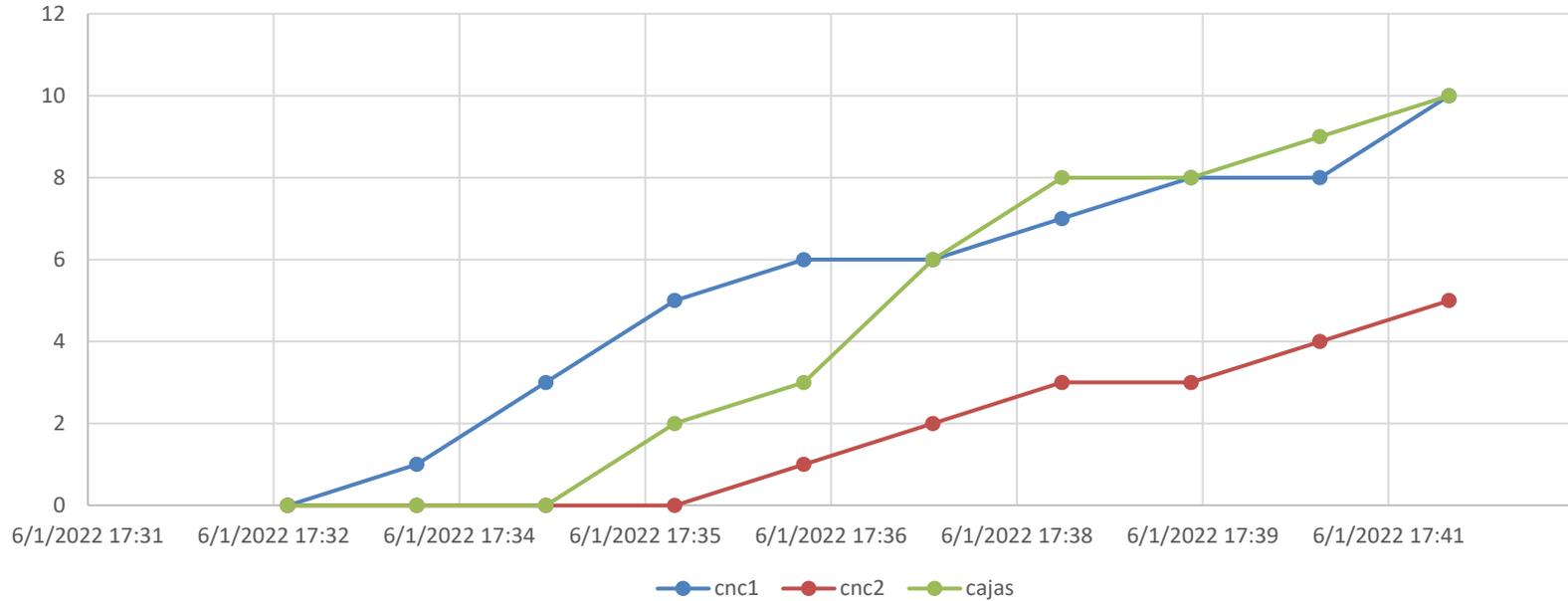


Se aprecia el marcado crecimiento de los totales de piezas y cajas. es clave para analizar el caso de haber un problema en el transporte de las piezas que pudiera traducirse en fallos de las bandas u otros elementos



GRÁFICA DE CRECIMIENTO DE PIEZAS MECANIZADAS Y CAJAS DE ACUERDO CON LOS DATOS ALMACENADOS EN THINGER.IO

Datos almacenados en Thinger IO



Se tiene una idea general del crecimiento de los totales de piezas y cajas; sin embargo, se pierde información, lo que no permite realizar el mismo análisis a profundidad de la gráfica anterior



ANALISIS DE TIEMPO DE RESPUESTA EN LA COMUNICACIÓN CON EL PLC

Tipo de Variable	Descripción	Dirección servidor OPC	Tiempo Wireshark / cronometrado
Bool	Inicio proceso 1	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.STARTP1	17ms / <1s
Bool	Paro proceso 1	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.STOPP1	21ms / <1s
Word	Contador CNC1	ns= 4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales. CONTADORCNC1	8ms / <1s
Word	Contador Cajas	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales. CONTADORCAJAS	9ms / <1s
Bool	Accionamiento de motores	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales. MOTOR1P1	19ms / <1s

Se aprecia que los tiempos cumplen métricas menores a 1 segundo para todas las variables analizadas. Cabe recalcar que estos tiempos dependen en gran medida del ancho de banda que maneje la implementación del sistema de red diseñado



COMPROBACIÓN DE LA HIPÓTESIS

Implementación de IoT industrial

El principal beneficio a partir de la virtualización de los servicios de IoT industrial, que constituye una red de comunicación de datos sobre el funcionamiento de la planta simulada, es que se lleva la etapa de supervisión a otro nivel. Todo esto para optimizar el control de las operaciones de planta; además de obtener ventajas en cuanto a tiempos y costos de implementación.

Software	Precio por licencia
Factory IO	\$30,00 / mensual
Servidor OPC-UA Codesys	\$56,63
TOTAL	\$86,63



COMPROBACIÓN DE LA HIPÓTESIS

Facilidad de comunicación de los sistemas de automatización industrial con interfaces heterogéneas



CONCLUSIONES

- La propuesta de simulación de este trabajo permite evidenciar la comunicación en los niveles mencionados, esto gracias a la implementación de un servidor OPC UA que sirve de base para la unión de los protocolos y servicios IIoT contemplados en el desarrollo de este trabajo.
- Se adquieren los conocimientos necesarios para manejar equipos y protocolos de gama industrial de gran capacidad. Su simplicidad de uso y su compatibilidad con demás *software* y *hardware* lo convierte en la opción ideal para un despliegue de una planta de manufactura en un entorno simulado.
- Para este trabajo se implementa gran variedad de *software*, entre ellos Codesys. Esta herramienta permite levantar el servicio base y fundamental de este proyecto, el servidor OPC UA. Para esto, en *hardware* se hace uso del dispositivo Raspberry Pi4, que hace las veces de controlador PLC. Es en éste en el que se ejecuta el servidor mencionado. La utilización de OPC UA permite una comunicación entre todos los niveles de la arquitectura propuesta. La propuesta de un servidor OPC UA como base fundamental que permite la comunicación de tecnologías heterogéneas, se concibe como un *middleware* que habilita implementar las soluciones IIoT.



CONCLUSIONES

- El manejo y gestión de la información del *dashboard* de la nube de almacenamiento desde una aplicación móvil no se realiza debido a que la plataforma Thinger.io no dispone del *kernel* requerido para su despliegue. Por este motivo se utilizan otras herramientas para gestión de los datos como Node-RED, la base de datos local implementada en MySQL y el *bot* de consulta. Sin embargo, para llevar a cabo el monitoreo, tanto en Thinger.io como en Node-RED, únicamente se requiere de una conexión a internet y un navegador web.



RECOMENDACIONES

- La simulación implementada de la planta de fabricación de piezas metalmecánicas y embalaje requiere ejecutar varios programas al mismo tiempo; por tanto, se recomienda contar con una conexión estable a internet y un ordenador que cumpla con las exigencias de funcionamiento del proyecto.
- Adicionalmente, el funcionamiento adecuado de este trabajo depende de que la conexión a internet sea de banda ancha, debido a que la red maneja una consistente cantidad de datos provenientes de las comunicaciones entre varios dispositivos y servicios.
- Con la finalidad de mejorar los resultados de almacenamiento obtenidos con Thinger.io, es recomendable investigar otras plataformas de almacenamiento en nube.



TRABAJOS FUTUROS

- Implementación a nivel de *hardware* para control físico de un modelo similar a la planta diseñada en el presente trabajo. La variedad de tecnologías heterogéneas permite llevar a la práctica su integración, teniendo las facilidades de acceso a *hardware* con características similares a las simuladas en la planta.
- Seguridad para el envío de información, tanto para el servidor local como para el almacenamiento de datos que se aloja en las nubes de información.
- Integración de nuevos equipos dentro del ambiente simulado, para generar un agente de control de fallos dentro de la planta, de esta manera se puede escalar un nivel más dentro de la pirámide de automatización CIM, llegando así a tener un nivel de supervisión y control de fallos virtualizado.



MUCHAS GRACIAS



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA