

## **POSGRADOS**

MAESTRÍA EN-

## ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-30-No.507-2019

OPCIÓN DE TITULACION:

PROYECTO DE DESARROLLO

TEMA:

PANEL DE MONITOREO Y CONTROL REMOTO UTILIZANDO LA PLATAFORMA IBM CLOUD. APLICACIÓN A UNA LINEA DE PRODUCCIÓN VIRTUALIZADA EN 3D

AUTOR:

CARLOS MOISÉS QUIROGA CENTENO

DIRECTOR:

BYRON XAVIER LIMA CEDILLO

GUAYAQUIL-ECUADOR 2022

#### Autor:



#### Carlos Moisés Quiroga Centeno

El autor es estudiante del programa de Maestría en Electrónica y Automatización META. Obtuvo su título de Ingeniero Eléctrico en 2018 por la Universidad Politécnica Salesiana. Actualmente se desempeña como Eléctrico de Turno en el Área de Ingeniería del Puerto de Aguas Profundas DP-World Guayaquil. cquirogac@est.ups.edu.ec

#### Dirigido por:



#### Byron Xavier Lima Cedillo

Ingeniero Electrónico con mención en Sistemas Industriales. Magister en Automatización y Control Industrial. Actualmente se desempeña como docente en el área de Pregrado y Posgrado de la Universidad Politécnica Salesiana. blimac@ups.edu.ec

Todos los derechos reservados. Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS ©2022 Universidad Politécnica Salesiana. GUAYAQUIL – ECUADOR – SUDAMÉRICA CARLOS MOISÉS QUIROGA CENTENO.

PANEL DE MONITOREO Y CONTROL REMOTO UTILIZANDO LA PLATAFORMA IBM CLOUD. APLICACIÓN A UNA LINEA DE PRODUCCIÓN VIRTUALIZADA EN 3D

# Índice general

Ín	dice	de Fig	uras	$\mathbf{V}$
Ín	dice	de Tal	olas	$\mathbf{X}$
D	edica	toria		XI
Re	esum	en		XI
Al	ostra	$\mathbf{ct}$		XI
1.	Intr	oducci	ión	1
	1.1.	Reseña	a generalizada de la problemática	2
	1.2.		VOS	
		1.2.1.		
		1.2.2.	Objetivos específicos	
	1.3.	Contri	buciones	
	1.4.		tura del Hológrafo	
<b>2</b> .	Mai	co Teá	órico	5
	2.1.	Estado	o del Arte	6
	2.2.	Definio	ciones Previas	7
		2.2.1.	Definición y Evolución del IoT	7
		2.2.2.	Arquitectura del IoT	10
		2.2.3.	Arquitectura de Tres Capas	11
		2.2.4.	Diseño de Cinco Capas	12
		2.2.5.	Tecnología Bluetooth	
		2.2.6.	Identificación por radiofrecuencia	
		2.2.7.	Red de sensores inalámbricos (WSN)	
		2.2.8.	Wireless Fidelity (Wi-Fi)	
		2.2.9.	Redes 5G	
		2 2 10	Node-Red	14

ÍNDICE	GENERAL		IV

4.	Res	3.3.10. B ultados	ase de Datos para la producción		104 <b>107</b>
			reación de dispositivo IoT en IBM Cloud		
			BM Cloud		
			esarrollo del Dashboard		
			omunicación entre Node-Red y Tia Portal		
			rocedimiento para instalación de NODE-RED		
			7-PLCSIM Advanced		
			reación de un nuevo proyecto		
	ა.ა.		ia Portal V16		
	3.2. 3.3.		${f o}$		
			el experimento		
			el Proyecto		32
		2.3.15. D	atos Técnicos de elementos y equipos		27
			ode-RED Dashboard		
			Qué es un Nodo?		
			Qué es JSON?		
		2.3.11. V	entajas de Node-RED		25
			Qué es Node-RED?		
			actory I/O $\dots$		
			omunicación S7		
			Qué es un software de base de datos?		
			Qué es un broker?		
			rganización?		
		2.3.4. ¿I	De qué forma integrar plataformas Iot en u	ına	
		_	Cuáles son los elementos de las plataformas IoT? . lases de plataformas Iot:		
			Qué son las plataformas IoT?		
	2.3.		onceptual		
	0.0		oncepto tiempo medio de reparaciones		
			oncepto Tiempo Medio entre fallas		
			adicadores Clave de Rendimiento (KPI)		

# Índice de Figuras

2.1.	Esquemático de Revoluciones Industriales	6
2.2.	Evolución del Internet	8
2.3.	Arquitectura tres, cuatro y cinco capas	10
2.4.	Arquitectura por capas de una red de sensores inalámbricos	14
2.5.	Node-Red Editor de flujo	15
2.6.	Fórmula para cálculo MTBF	17
2.7.	Fórmula para cálculo MTTR	17
2.8.	Esquema de Iot	18
2.9.	Base de Datos	22
2.10.	Capas del protocolo S7	23
2.11.	Ejemplo de fabrica 3D	24
2.12.	Ejemplo del diseño en FBP	25
2.13.	Node-red vs Node.js Programación	27
2.14.	Ejemplo dashboard	28
2.15.	Controlador lógico programable S7-1500	28
2.16.	Módulo de Entradas Digitales	29
2.17.	Bloque de Salidas binarias	30
2.18.	Módulo de Entradas analógicas	31
3.1.	Esquema para red utilizada	34
3.2.	Dashboard en Red Local	34
3.3.	Entorno de Programación	35
3.4.	Verificación de Software instalado	36
3.5.	Vista de dispositivos	36
3.6.	Selección de dispositivo	37
3.7.	Configuración del Proyecto	38
3.8.	Bloque OB01	39
3.9.	Bloque OB30-OB38	39
3.10.	Bloque OB100	40
2 11	Plague OR40	41

ÍNDICE DE FIGURAS	VI
3.12. Bloque OB40	41
3.13. Bloque de Función FB	42
3.14. Bloque de Función FC	43
3.15. Bloque de Datos DB	44
3.16. Vista inicial PLC SIM	45
3.17. Ventana PLC-SIM Advanced	46
3.18. Descargar en dispositivo	46
3.19. Comunicación OPC	47
3.20. Propiedades para simulación	48
3.21. Adaptador de Red virtual Siemens	49
3.22. Sitio web Descargar Nodejs	49
3.23. Instalación Node.js	50
3.24. Selección Carpeta Instalación	51
3.25. Tipos de Instalación	51
3.26. Herramientas nativas para Nodejs	52
3.27. Códigos de Confirmación	52
3.28. Comando de ejecución Node Red	53
3.29. Comando Node-Red Ejecutado	53
3.30. Panel de control	54
3.31. Ventana de opciones Firewall	54
3.32. Opciones avanzadas Firewall	55
3.33. Inbound Rulesl	56
3.34. Creación de nueva regla	57
3.35. Configuración del puerto	57
3.36. Nombre del puerto	58
3.37. Puerto Configurado	58
3.38. Visualización de Node-red en otro dispositivo	59
3.39. Componentes de Node.js	59
3.40. Ventana de trabajo en Node-Red	60
3.41. Vista interior del Menú Principal en Node-Red	61
3.42. Vista de Nodo Inject en Paleta de Nodos	62
3.43. Vista de edición del Nodo	63
3.44. Pestaña Editor Json	63
3.45. Pestaña Editor Visual	64
3.46. Ventana de Flujo	64
3.47. Visualizar mensaje de nodo Inject	64
3.48. Ejecución de Nodo Inject	65
3.49. Data Block sin acceso Optimizado	66
3.50. Variable Estructura	66
3.51. Estructura Copiada	67

ÍNDICE DE FIGURAS	VII
3.52. Variables de lectura y escritura Node-Red	67
3.53. Valores Iniciales	68
3.54. Valores Ingresados para producción	68
3.55. Plano Ocupacional Byte Entradas y Marcas del sistema	69
3.56. Plano Ocupacional Byte de Salidas	69
3.57. Nodo S7-COMM	70
3.58. Nodo S7-COMM edición	71
3.59. Configuración de Dispositivo Nodo S7-COMM	71
3.60. Valores de Offset por defecto en Tia Portal	72
3.61. Nodos de Dashboard	73
3.62. Menú de dashboard	74
3.63. Editar nombre de Dashboard	74
3.64. Editar grupo Dashboard	74
3.65. Editar Tema en Node-Red	75
3.66. Cambio de colores en panel de control Node-Red	75
3.67. Nodo MySql	76
3.68. Nodo Botón	76
3.69. Nodo Texto	76
3.70. Nodo Función	77
3.71. Nodo Plantilla	77
3.72. Ejemplo plantilla	78
3.73. Nodo Switch	78
3.74. Configuración Switch	79
3.75. Nodo Gráficos	79
3.76. Configuraciones de Gráfica	80
3.77. Configuración Nodo Medios	81
3.78. Configuración archivos de Medios	81
3.79. Nodo Medios	82
3.80. Nodo RBE	82
3.81. Configurar nodo RBE	82
3.82. Nodo Catch	83
3.83. Configuración nodo Catch	83
3.84. Nodo S7-Comm-Read	84
3.85. Configurar Nodo S7-Comm	84
3.86. Nodo S7-Comm-Write	85
3.87. Nodo Indicador	85
3.88. Configuración Nodo Indicador	86
3.89. Nodo Subflujo	86
3.90. Ejemplo de Subflujo creado por el autor	87
3.91. Panel de Control Node-Red	88

3.92. Configuración de Cuenta       89         3.93. Datos Personales       90         3.94. Condiciones del Servicio       90         3.95. Revisión de privacidad de la cuenta       91         3.96. Verificación por tarjeta de crédito       91         3.97. Información para facturar       92         3.98. Menú Principal IBM Cloud       92         3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       98         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.15Nodo IBM IoT In       102         3.115Declaración de Tabla e	ÍNDICE DE FIGURAS	VIII
3.94. Condiciones del Servicio       90         3.95. Revisión de privacidad de la cuenta       91         3.96. Verificación por tarjeta de crédito       91         3.97. Información para facturar       92         3.98. Menú Principal IBM Cloud       92         3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.102Desplegar aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.114Credenciales del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.115Nodo IBM IoT In       102         3.117Editar API Key Nodo IBM In       103         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas	3.92. Configuración de Cuenta	. 89
3.95. Revisión de privacidad de la cuenta       91         3.96. Verificación por tarjeta de crédito       91         3.97. Información para facturar       92         3.98. Menú Principal IBM Cloud       92         3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.114Credenciales del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.117Daba para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.120Tabla para a obtener tiempos segmentos 1-4       109         4.1. Data Blocks<	3.93. Datos Personales	. 90
3.96. Verificación por tarjeta de crédito       91         3.97. Información para facturar       92         3.98. Menú Principal IBM Cloud       92         3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.119Declaración de Tabla en Workbench       104         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para K	3.94. Condiciones del Servicio	. 90
3.97. Información para facturar       92         3.98. Menú Principal IBM Cloud       92         3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.111Discositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.115Nodo IBM IoT In       102         3.115Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para para obtener tiempos segmentos 5-7       109         4.4. Programa para o	3.95. Revisión de privacidad de la cuenta	. 91
3.98. Menú Principal IBM Cloud       92         3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8	3.96. Verificación por tarjeta de crédito	. 91
3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para API       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 5-7	3.97. Información para facturar	. 92
3.99. Kit started       94         3.100Empezar instalación Node Red en la Nube       94         3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para API       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 5-7	3.98. Menú Principal IBM Cloud	. 92
3.101Configuraciones para instalar Node-Red       95         3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para bases y Tapas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111 <td></td> <td></td>		
3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para bases y Tapas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111     <	3.100Empezar instalación Node Red en la Nube	. 94
3.102Desplegar aplicación       95         3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para bases y Tapas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111     <	3.101Configuraciones para instalar Node-Red	. 95
3.103Configuración de aplicación       96         3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para bases y Tapas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112 </td <td></td> <td></td>		
3.104Creando API KEY       96         3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para bases y Tapas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTBF       110         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.105Finalizando creación instancia Node.Js       97         3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.120Tabla para bases y Tapas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112	•	
3.106Panel de Instancia para instalar Node RED       97         3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.107Usuario y clave puesta de servicio en Node-Red       98         3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.108Node Red instalado con Link de IBM CLOUD       98         3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.109Dashboard en IBM CLOUD       99         3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.110Acceso a IBM Watson       100         3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.111Dispositivos creados en watson IoT       100         3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.112Nodo Watson       101         3.113Propiedades del Nodo Watson       102         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.113Propiedades del Nodo Watson       101         3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1 Data Blocks       108         4.2 Programa para obtener tiempos segmentos 1-4       109         4.3 Programa para obtener tiempos segmentos 5-7       109         4.4 Programa para obtener tiempos segmentos 8-9       110         4.5 Cálculo de MTBF       110         4.6 Cálculo de MTTR       111         4.7 Segmentos de programación Kpi       111         4.8 Dashboard Prueba 1       112	•	
3.114Credenciales del Nodo Watson       102         3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1 Data Blocks       108         4.2 Programa para obtener tiempos segmentos 1-4       109         4.3 Programa para obtener tiempos segmentos 5-7       109         4.4 Programa para obtener tiempos segmentos 8-9       110         4.5 Cálculo de MTBF       110         4.6 Cálculo de MTTR       111         4.7 Segmentos de programación Kpi       111         4.8 Dashboard Prueba 1       112		
3.115Nodo IBM IoT In       102         3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.116Propiedades Nodo IBM In       103         3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.117Editar API Key Nodo IBM In       103         3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.118Declaración de Tabla en Workbench       104         3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.119Tabla para bases y Tapas       105         3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.120Tabla para usuarios y contraseñas       105         3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
3.121Tabla para KPI       106         4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
4.1. Data Blocks       108         4.2. Programa para obtener tiempos segmentos 1-4       109         4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112	*	
4.2. Programa para obtener tiempos segmentos 1-41094.3. Programa para obtener tiempos segmentos 5-71094.4. Programa para obtener tiempos segmentos 8-91104.5. Cálculo de MTBF1104.6. Cálculo de MTTR1114.7. Segmentos de programación Kpi1114.8. Dashboard Prueba 1112	3.121.1abia para KPI	. 100
4.2. Programa para obtener tiempos segmentos 1-41094.3. Programa para obtener tiempos segmentos 5-71094.4. Programa para obtener tiempos segmentos 8-91104.5. Cálculo de MTBF1104.6. Cálculo de MTTR1114.7. Segmentos de programación Kpi1114.8. Dashboard Prueba 1112	4.1 Data Blocks	108
4.3. Programa para obtener tiempos segmentos 5-7       109         4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
4.4. Programa para obtener tiempos segmentos 8-9       110         4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
4.5. Cálculo de MTBF       110         4.6. Cálculo de MTTR       111         4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba 1       112		
4.6. Cálculo de MTTR		
4.7. Segmentos de programación Kpi       111         4.8. Dashboard Prueba       112		
4.8. Dashboard Prueba 1		
	4.9. Dashboard Prueba 2	

ÍNDICE DE FIGURAS	IX
4.10. Comparación Históricos Base de Datos	 113
4.11. Administrador de Tareas	 114

## Índice de Tablas

2.1.	Brokers Mqtt							2
2.2.	Características Técnicas S7-1500							29
2.3.	Tabla Técnica para Entradas Digitales .							30
2.4.	Tabla técnica de Salidas Digitales							30
2.5.	Tabla Técnica para Entradas Analógicas							3

### Dedicatoria

A lo largo de mi recorrido en la vida y mi paso en las aulas de la universidad politécnica salesiana me pude dar cuenta que hay muchas cosas para las que soy bueno destrezas y habilidades que no creí que se desarrollarían en mí, que a pesar de realizar actividades y proyectos en los que pensaba que estaba solo siempre obtuve un mejor resultado con ayuda de mi familia y amigos este agradecimiento debo darlo a estas maravillosas persona que tengo el lujo de tener en mi vida mis padres Norma Centeno y Manuel Quiroga que con sus palabras me enseño que debo seguir adelante que solo debo ver mi meta sin nunca perder la dignidad ni desfallecer en el intento.

Al Ingeniero Carlos Romero por estar allí incondicionalmente sin esperar nada a cambio le agradezco de corazón por sus consejos de hermano que me permitieron enfocarme en la culminación de este trabajo, gracias mi buen amigo y hermano de vida.

A todos mis irreemplazables amigos que no pude mencionar anteriormente, ya son parte de todo.

## Resumen

Las interfaces en la nube actualmente se proponen en reemplazar a las conocidas HMI (Interfaz Humano Máquina), debido a la constante actualización de las empresas muchos dispositivos industriales ven la necesidad de migrar sus aplicaciones a un entorno web de la nube e incrementar el acceso a documentación para generar indicadores clave de desempeño (KPI) y el control remoto desde cualquier parte del mundo, por ello se realiza el presente trabajo que muestra una alternativa de funcionamiento en la Nube, realizando un Dashboard para el monitoreo remoto de un proceso en el software de simulación (Factory I/O) instalado en el computador.

Node-Red permite la interacción con diferentes programas y protocolos de comunicación debido a su plataforma en código abierto y al uso de interfaces en la nube. El Dashboard en Node-Red, es un módulo adicional que brinda la posibilidad de crear paneles de control en tiempo real, al momento de la instalación del módulo se añade la categoría Dashboard a la paleta de nodos con widgets para realizar la interfaz de usuario. Se realiza la programación del proceso **Production Line** en Tia Portal V16 que se conecta mediante comunicación S7 a Node-Red, se ingresan el numero de piezas por linea de producción a realizar desde el Dashboard desarrollado en Node-red local de la computadora, siendo ejecutado desde una pestaña del explorador Microsoft Edge, posteriormente las variables de producción se acumulan en una base de datos desarrollada en el software MySql, se procede a enviar los datos por medio de la plataforma Watson IoT de IBM. Los datos adquiridos en Watson IoT se transfieren a otro Node-Red instalado en la plataforma IBM Cloud, se ejecuta el Dashboard ingresando a un enlace proporcionado por IBM Cloud y se visualiza la producción desde cualquier parte del mundo, por motivos de seguridad se genera un token por cada elemento que pase por Watson IoT esto garantiza que los datos no puedan ser corrompidos por un agente externo.

**Palabras clave:** KPI, IoT, Dashboard, Node-Red, widgets, MySql, Watson, IBM, IBM Cloud

### Abstract

The cloud interfaces are currently proposed to replace traditional HMI (Human Machine Interface), due to the constant update of the companies many industrial devices have to migrate their applications to a cloud platform to maximize data access to generate key performance indicators (KPI) and remote control around the world. This reason to develop a project to show an alternative operation in the cloud, making a Dashboard for remote monitoring of a process in the simulation software (Factory I/O) installed on the computer.

Node-Red allows interaction with any kind of software and communication protocols due to it's open source platform and the use of cloud interfaces. The Dashboard in Node-Red is an additional module which enables the creation of real-time control panels. When the module is installed, a Dashboard category is added to the node palette with widgets to create an user interface. The programming of the process **Production Line** is performed in Tia Portal V16 connected through S7 communication to Node-Red, the number of pieces per production line to be performed were entered by the Dashboard developed in local Node-network of computer, being executed from a tab of Microsoft Edge browser, then production data are stored in a database developed in MySql software, we proceed to send data through IBM's Watson IoT platform. The data acquired in Watson IoT is moved to another Node-Red installed in IBM Cloud platform, Dashboard is executed by accessing a link provided by IBM Cloud and the production is displayed worldwide, for security reasons a token is generated for each element that passes through Watson IoT, this ensures that the data will not get corrupted by any external party.

**Keywords:** KPI, IoT, Dashboard, Node-Red, widgets, MySql, Watson, IBM, IBM Cloud

### Capítulo 1

### Introducción

En la última década el desarrollo de la industria 4.0 ha permitido la obtención y medición de datos sin la necesidad de intervención humana, se ha automatizado este proceso para que las empresas puedan hacer uso de los mismos y verificar posibles fallos en sus diversas ejecuciones durante la producción.

Se tiene como meta la elaboración de un panel o interfaz de monitoreo y control remoto utilizando la plataforma IBM Cloud para una línea de producción virtualizada en 3D, mediante el uso del editor de flujo Node-Red, el cual utiliza diversos protocolos de comunicación tales como ModBus Tcp/Ip, S7-Comm accesibles a controladores lógicos programables. Este sistema además puede generar una compuerta de enlace a diferentes plataformas en la nube tales como Watson Iot, IBM Cloud, Google Cloud service, Amazon Web Services entre otros para la utilización de dispositivos IoT.

En el desarrollo del proyecto, el inconveniente notable es el uso de un solo computador para las simulaciones debido al consumo de recursos del mismo, se realiza una prueba beta con el PLC-SIM V16, este programa solo se podía comunicar con Node-Red o Factory I/O independientemente y no de manera simultánea, se trato de integrar el software NetToPLCsim para complementar la falencia antes mencionada pero había problemas y se perdían datos durante la transmisión, actualmente se utiliza el programa PLC-SIM Advanced 3.0 en conjunto con el TIA Portal V16, este programa instala un puerto Ethernet virtual para simular la conexión a un S7-1500, esto permite leer los datos que envía Factory I/O a Tia Portal, digitarlos en Node-Red y luego hacer el proceso inverso.

#### 1.1. Reseña generalizada de la problemática

Durante los últimos años se ha tratado de mejorar los procesos, disminuir el tiempo de mantenimientos correctivos y aumentar la productividad, la industria se ha optimizado potencialmente gracias al uso del IoT (Internet of Things); Por ello se aprovecha en gran medida las ventajas que ofrece Node-Red con el acceso remoto para observar el estado del proceso, conocer los tipos de fallas recurrentes dentro del mismo, realizar planes para disminuirlas y que los indicadores claves de desempeño no se vean afectados en gran medida.

El desarrollo del proyecto muestra una nueva perspectiva para los SCADA, permitiendo el desarrollo de aplicaciones con interfaz gráfica sin usar Licencias Runtime o de HMI, estos elementos son reemplazados con el uso del Smartphone o un computador convencional para visualizar el estado del proceso en cualquier momento y lugar, implementando el uso de usuario y contraseña previamente configurados para así evitar accesos indeseados que podrían contribuir al mal funcionamiento de todos los sistemas instalados en el proceso a controlar.

Actualmente dentro del Software Factory I/O desarrollado por la compañía REALGAMES se tienen procesos industriales desarrollados en un ambiente 3D para simular su funcionamiento con el aporte de un controlador lógico programable, ya sea con hardware físico o virtual, actualmente Factory I/O no puede enviar datos a hacia la nube para la lectura de variables, debido a esta limitante se propone la elaboración de la interfaz Hombre-máquina para almacenar datos provenientes desde la etapa de mecanizado en la línea de producción, utilizando la plataforma de desarrollo Node-Red e integrando el software TIA Portal, se pretende obtener dichos datos y crear el indicador clave de desempeño (KPI) simulando una parte de la producción, al final los datos estarán disponibles en la plataforma desarrollada en la nube.

#### 1.2. Objetivos

#### 1.2.1. Objetivo general

Elaborar un panel de monitoreo y control remoto utilizando la plataforma IBM Cloud para una línea de producción virtualizada en 3D.

#### 1.2.2. Objetivos específicos

- Analizar el concepto sobre el internet de las cosas, mediante el reconocimiento de ejemplos y artículos científicos para adquirir las bases conceptuales y metodológicas para la ejecución del proyecto.
- Definir los requisitos de diseño, para el funcionamiento de un Dashboard en Node-Red, con el fin de sentar las bases en la simulación del proyecto
- Implementar un Dashboard en Node-Red, mediante el uso de las variables en TIA Portal, para interacción remota entre usuario y proceso simulado en Factory I/O.
- Implementar un sistema de monitoreo en la nube utilizando la plataforma IBM Cloud para la etapa de mecanizado de una línea de producción, simulado en Factory I/O.
- Configurar el Dashboard para despliegue de gráficas y almacenamiento de variables en la base de datos MySql
- Validar el Dashboard implementado, mediante la simulación en Factory I/O con los datos de la producción total en TIA-Portal, con la finalidad de medir el KPI del proceso.

#### 1.3. Contribuciones

El proyecto permitirá a los investigadores mediante el uso de Factory I/O simular un proceso determinado de ejemplo y explotar las capacidades de la industria 4.0 sin necesidad de tener que estar en una fábrica real o tener una planta a escala. Debido al tiempo de pandemia este tipo de software resulta bastante útil para continuar con la investigación y desarrollo de controladores sin necesidad de exponerse a contagios.

Haciendo uso del editor en Node-red el área de mantenimiento y planificación de las empresas pueden obtener a los datos de producción en ese instante, previamente verificando su identidad en la pagina Web pueden acceder al mismo, el monitoreo se realiza en tiempo real de manera automatizada obteniendo los indicadores clave de desempeño de una forma ágil y organizada.

#### 1.4. Estructura del Hológrafo

La documentación se realiza con cuatro capítulos, a continuación, se expone una breve introducción a cada uno de ellos:

- Capítulo 1: Se plantea la problemática con todos los factores asociados a la misma; se establece el objetivo principal junto a los específicos planteando una solución.
- Capítulo 2: Presentación de los conceptos básicos y teóricos necesarios para desarrollar el problema planteado.
- Capítulo 3: Se desarrolla la solución al problema planteado con la aplicación de herramientas para dispositivos IoT.
- Capítulo 4: Se muestran los resultados obtenidos durante el desarrollo del proyecto, las conclusiones y recomendaciones para otros trabajos a futuro

## Capítulo 2

## Marco Teórico

Dentro del capítulo se logra detallar el análisis teórico para la elaboración del manuscrito en cuestión, se estudia la aplicación de diferentes plataformas y dispositivos IoT que en gran medida contribuyen a la solución de los problemas ya mencionados en el capítulo anterior. Se abordaran temas sobre el uso de la Industria 4.0 en las fabricas actuales y como las empresas están migrando la obtención de datos hacia la nube para luego aplicarlos en los indicadores claves de rendimiento y así mejorar la productividad. En la actualidad para implementar una aplicación orientada al uso de dispositivos IoT, tiene cierto costo dependiendo del fabricante del mismo, por ello en el presente capítulo se realizó una investigación exhaustiva para encontrar alternativas para elaborar el proyecto con software de simulación en tiempo real con lineas de producción en 3D.

El software Factory I/O es una alternativa utilizada en la simulación de procesos, comparar controladores, integrando otros softwares de simulación tales como Matlab, LabVIEW o TIA Portal, un ejemplo es el tema: "Mejora del control PID basado en una red neuronal".

#### 2.1. Estado del Arte

El constante aumento de equipos inteligente con disponibilidad de movimiento como por el ejemplo los celulares(smartphones) y cada uno de los requerimientos en la aplicación de sus factores ya determinados para los futuros sistemas 5G, los principales problemas que puede llegar a presentar los futuros sistemas de red 5g son: un mayor consumo de datos, tener que poder dar soporte a un gran sinnúmero de dispositivos en simultaneo, exceso de trafico en la red, poca intermitencia, mejor intensidad de señal en entornos cerrados y abiertos, ahorro de energía, asimetría y confiabilidad. [Bezeira et al., 2017].

El concepto de IoT permite conectar las cosas físicas con la electrónica integrada, el software, los sensores y la conectividad, proporcionando así el intercambio de datos con los fabricantes, los operadores y/o otros dispositivos conectados. En 2013, la Iniciativa de Normas Globales sobre la Internet de las Cosas (IoTGSI) definió la IoT como "la infraestructura global de TI que proporciona servicios avanzados (físicos y virtuales) mediante la conexión en red de cosas, basada en las tecnologías de la información y la comunicación existentes e interoperables en el desarrollo"[Vermesan et al., 2014].

Muchos son los apelativos proporcionados a la neo-revolución industrial dispuesta actualmente: Industria 4.0, compañías interconectadas a 4.0, fabricas inteligentes, internet de las cosas, entre otros. Haciendo énfasis en lo detallado anteriormente se aporta en parte al proceso de conversiones tecnológicas, comunitarias y económicas, en donde el hombre desarrolla nuevos componentes que hacen más sencilla la fabricación y el crecimiento, con ello se obtiene un progreso tanto cultural y económica en la sociedad, en la figura 2.1 se aprecian los aportes de cada revolución industrial a lo largo de la historia [Mejía Cruz, 2018].



Figura 2.1: Esquemático de Revoluciones Industriales. [Borja, 2019]

En la actual revolución industrial se puede repotenciar los sistemas de producción según las necesidades que presente el mercado en cuestión, para cumplir con la interconexión de los sistemas operacionales y la comunicación con dispositivos que se encuentren de forma remota indistintamente a la localidad, se reafirma el crecimiento de una correcta digitalización para la cooperatividad en cada una unidad de negocio. Uno de los pilares en los que se basa, es la captura de datos de los procesos de fabricación para poder analizar en todo momento el estado del sistema y tomar las acciones más adecuadas [Borja, 2019]. El núcleo es desde donde se receptan los datos que se dirigen a la nube en donde se guarda un respaldo del sistema en si es un hardware que comparte similitud con SIMATIC S7-1500 que se encarga de la ejecución del programa guardado en la máquina. En el desarrollo del proyecto se ha utilizado el editor basado en navegador Node-RED y de IBM Cloud, para tener acceso a IBM Cloud hay que registrarse y crear una cuenta, Node-RED permite arrastrar y soltar nodos que se pueden conectar entre si ademas desplegar opciones con un solo clic. Se crea un servicio Node-RED utilizando la plataforma IBM Cloud, se pueden conectar sensores IBM IoT si el usuario lo requiere, en este caso se publican datos del proceso.

La tecnología en nube representa la entrega de recursos informáticos y capacidad de almacenamiento como un servicio para un grupo heterogéneo de usuarios finales. Por lo tanto, la nube proporciona de alguna manera la prestación de servicios en lugar del producto en sí. Los usuarios pueden tener acceso a las aplicaciones de la nube a través de un navegador web o una aplicación de teléfono móvil, mientras que el software y los datos de los mismos se encuentran en servidores en una ubicación remota [Lekić and Gardašević, 2018].

#### 2.2. Definiciones Previas

#### 2.2.1. Definición y Evolución del IoT

K.a. logro dar un claro entendimiento del IoT en 1999, su respuesta fue definir como objetos identificables que se encuentran conectados de forma única. Sin embargo, dicho concepto sigue en proceso de hipótesis el cual esta sujeto a las perspectivas que se le adopten. Por lo general la IoT se muestra como la infraestructura de una red global que tiene capacidad de auto-configuración y se basa en protocolos de comunicación [Gokhale et al., 2018].

Se observa la evolución de Internet Fig. 2.2 , se puede clasificar en cinco épocas:

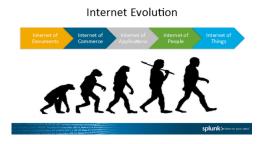


Figura 2.2: Evolución del Internet. [Li et al., 2015]

- 1. La Internet de los Documentos: Bibliotecas electrónicas, páginas web basadas en documentos
- 2. La Red del comercio: sitios web de comercio electrónico, banca electrónica y comercio de acciones.
- 3. La Internet de las aplicaciones: Web 2.0
- 4. La Web de las personas: Redes sociales.
- 5. La interconexión de los objetos: dispositivos y máquinas conectados.

El IoT es un conjunto de sensores, ordenadores y elementos digitales relacionados, localizados por todo el mundo a través del ciberespacio, que se comunican entre si para distribuir y trasladar información usando una tipificación exclusiva que se otorga a todos los dispositivos, como UID (identificadores únicos) [Gupta and Johari, 2019].

La tecnología del Internet de las cosas puede hacer que los objetos sean más inteligentes, entre los objetos más susceptibles de ser conectados de forma inteligente se encuentran los equipos domésticos, los equipos de la red eléctrica, los equipos logísticos, los equipos médicos, los equipos agrícolas y los automóviles [Qian and WANG, 2012].

El IoT tiene una gran aplicación en muchos de los campos, citando los siguientes:

 Evaluación ingeniosa basada en IOT para conocer los niveles de habilidad en los estudiantes y la dificultad a la hora de contestar las preguntas del mismo. [Xheladini et al., 2017]. Este sistema permite analizar la capacidad del estudiante para comprender un determinado tema o materia [Gupta and Johari, 2019].

- 2. Aplicación de monitoreo a la salud del paciente desde un lugar remoto basada en IOT[Minoli et al., 2017].
- 3. Actualmente el sistema de control en automóviles progresa obteniendo la información en directo del movimiento y monitorizar su rendimiento [Wang et al., 2016].

La visión del Internet de las cosas (IoT) ha evolucionado y se está haciendo realidad. El IoT implica varios miles de millones de dispositivos diversos interconectados para 2020 [Biswas and Giaffreda, 2014] vastas cantidades de datos de rápida aparición/versatilidad (es decir, "big data") y muchos servicios. El IoT permite que las cosas se noten y se controlen a distancia por la edificación de red existente, tolerando una incorporación más directa del entorno físico y de los sistemas informáticos, lo que se interpreta en aptitud, exactitud y beneficios económicos, al tiempo que se reduce la intervención humana [Lekić and Gardašević, 2018]. Las IoT poseen sus propias identidades y atributos, son capaces de manejar diferentes interfaces inteligentes e integrarse a una red de información, en resumen las Iot pueden emplearse como un grupo de dispositivos interconectados entre si que son identificables de manera única, un sistema IoT se implementa utilizando dispositivos periféricos con la función de almacenamiento, gestionar/transformar datos y transmitirlos a través de internet con otras aplicaciones [Mehmood et al., 2019]. Sin embargo, lo fundamental de la IoT implica que los objetos en la misma pueden ser identificados de forma única en sus representaciones virtuales. Dentro de una IoT, todas las máquinas son capaces de intercambiar datos y, si es necesario, de procesarlos de acuerdo con esquemas predefinidos [Gokhale et al., 2018].

#### 2.2.2. Arquitectura del IoT

Un requisito fundamental de la IoT es que los elementos de la red tienen que estar conectados entre sí. La arquitectura del sistema IoT debe garantizar que las operaciones de su red, se conecten a los medios físicos y virtuales. El diseño de la arquitectura de IoT implica muchos factores, como la red, la comunicación, los procesos, etc. En el diseño de la arquitectura deben tenerse en cuenta la extensión, la escalabilidad y la operatividad entre dispositivos, debido al hecho de que las máquinas pueden moverse y necesitan interactuar con otras en tiempo real, la arquitectura de la IoT debe ser adaptable para hacer que los dispositivos interactúen con otros dinámicamente y apoyen la comunicación entre ellos, además de poseer la naturaleza descentralizada y heterogénea.

Las IoT poseen muchos diseños arquitectónicos, en el presente documento solo se presentan las dos arquitecturas más comunes y ampliamente consideradas en las empresas, la investigación industrial y sus aplicaciones; conocidas como arquitecturas de tres capas y de cinco capas como se observa en la Fig.2.3.

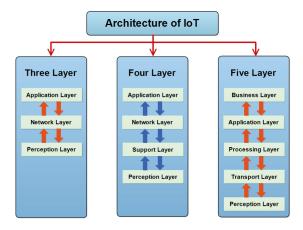


Figura 2.3: Arquitectura tres, cuatro y cinco capas [Burhan et al., 2018]

No existe un acuerdo único y general sobre la arquitectura de la IoT que sea consensuado por todo el todo el mundo y los investigadores, se debe considerar que la arquitectura que se proponga debe cumplir los requisitos de IoT en cuanto a seguridad y privacidad.

#### 2.2.3. Arquitectura de Tres Capas

Capa de percepción: También se conoce como capa sensora, funciona como los ojos, los oídos y la nariz de las personas, tiene la responsabilidad de identificar las cosas y recoger la información de ellas, hay muchos tipos de sensores que se adhieren a objetos para recoger información, como la RFID, el código de barras 2D y los sensores. La información que recogen estos sensores puede ser sobre ubicación, cambios en el aire, entorno, movimiento, vibración, etc. Sin embargo, son el principal objetivo de atacantes que desean utilizarlos para sustituir el sensor por su propia información [Suo et al., 2012].

Capa de red: Se la define como capa de transmisión, actúa como un puente entre las capas de percepción y aplicación, lleva y transmite la información recogida de los objetos físicos a través de sensores. El medio de transmisión puede ser inalámbrico o por cable, también se encarga de conectar las cosas inteligentes, los dispositivos de red y las redes entre sí, Por lo tanto, es muy sensible a los ataques por parte de los atacantes. Tiene problemas de seguridad importantes en lo que respecta a la integridad y la autenticación de la información que se transporta en la red [Burhan et al., 2018]. Las amenazas y los problemas de seguridad para las capas de red son:

- Ataque denegación de servicio (DoS)
- Ataque al medio principal (MiTM)
- Ataque a la memoria
- Ataque de explotación

Capa de aplicación: Se determinan todas las aplicaciones que utilizan la tecnología IoT o en las que se ha desplegado. Las aplicaciones de IoT estan en hogares inteligentes, ciudades, salud, seguimiento de animales. Posee la responsabilidad de brindar los servicios a las aplicaciones, estos pueden variar para cada uso porque dependen de la información que recogen los sensores. Hay muchos problemas en la capa de aplicación en los que la seguridad es clave; En particular, cuando IoT se utiliza para para hacer una casa inteligente, introduce muchas amenazas y vulnerabilidades desde el interior y el exterior [Ali and Awad, 2018].

#### 2.2.4. Diseño de Cinco Capas

El modelo de cuatro capas protagonizó un papel principal en el crecimiento de la IoT, igualmente se tuvieron problemas adjuntos a la seguridad y almacenamiento. Los investigadores propusieron una arquitectura de cinco capas para hacer que la IoT sea segura[Madakam et al., 2015]. Tiene tres capas como las arquitecturas anteriores que son capa de percepción, transporte y aplicación. Además, cuenta con dos capas más, los nombres de estas nuevas capas propuestas son capa de procesamiento y capa de negocio. Se considera que la nueva arquitectura propuesta posee la virtud de ejecutar las condiciones de la IoT [Khan et al., 2012].

Capa de Procesamiento: Es nombrada como capa de middleware, recoge las referencias expedidas desde la capa transporte y ejecuta el encausamiento de la información recolectada. Posee la tarea de suprimir la información extra sin sentido y extrae lo necesario. Hay numerosos ataques que pueden afectar a la capa de procesamiento y perturban el rendimiento de IoT.

Capa de Comercio: Da a conocer el comportamiento previsto en una aplicación y actúa como un gestor de todo, incluye la facultad de gestionar y dominar las app, el negocio y beneficios del IoT. La privacidad del usuario también es gestionada por esta capa, posee la capacidad de determinar cómo la información puede ser creada, almacenada y modificada. Muchos de los inconvenientes vinculado con seguridad son las debilidades en una aplicación que resulta de un control fiable muy ausente.

#### 2.2.5. Tecnología Bluetooth

Bluetooth se utiliza para aplicaciones que quieren comunicarse a corta distancia, ofrece muchos mecanismos de seguridad en la interconexión entre el remitente y el receptor. Proporciona la facilidad de encriptación en la que un mensaje se convierte en otra forma, llamada texto cifrado. Por otro lado, el receptor también tiene la capacidad de cambiar el texto cifrado en un mensaje original, gracias a la encriptación, un mensaje no puede ser entendido por nadie excepto por el usuario que tiene los derechos para ver el mensaje. El emisor debe obtener el permiso del receptor antes de enviar el mensaje; En primer lugar, se envía una solicitud a un receptor que tiene información que el emisor quiere compartir y dependerá del receptor aceptar o rechazar la solicitud del emisor [Padgette et al., 2011].

#### 2.2.6. Identificación por radiofrecuencia

RFID utiliza ondas de frecuencia para la comunicación entre dos dispositivos. Consta de tres partes: etiquetas, un lector y una base de datos; Las etiquetas se adhieren a los objetos y leen su estado, mientras que un lector se utiliza para interpretar el lenguaje de las etiquetas, En almacenamiento de datos se considera una tercera parte y se utiliza para almacenar la información [Morshed et al., 2011]. Proporciona la facilidad de encriptación para transmitir la información, hay tres formas utilizables de encriptación: En la primera, no se utiliza el cifrado, no se utiliza para enviar información innecesaria su objetivo es ahorrar recursos del sistema, el segundo tipo de cifrado es el **Estándar de Cifrado de Datos (DES)**: Es un método de encriptación de datos de clave simétrica, maneja la misma contraseña para codificar y descodificar un mensaje, por tanto el remitente y el receptor deben conocer y utilizar una clave privada[Khoo, 2011].

El tercer tipo es el**Advanced Encryption Standard (AES)**: Es un algoritmo criptográfico más eficiente y elegante, su principal fuerza reside en las opciones de longitud de la clave, el tiempo necesario para romper un algoritmo de cifrado está directamente relacionado con la longitud de la clave, se permite elegir una clave de 128, 192 o 256 bits para crear más fuerza que la clave de 56 bits de DES y por lo tanto, se considera más seguro que DES[Khoo, 2011].

#### 2.2.7. Red de sensores inalámbricos (WSN)

Las WSN tienen muchos nodos y cada uno de ellos tiene cuatro partes: sensores, batería, microcontrolador y memoria, la funcionalidad de las WSN puede entenderse fácilmente de esta manera: los sensores se utilizan para recoger la información y almacenarla en su memoria para volver a utilizarla y la información se envía al servidor, también se utilizan baterías que proporcionan facilidades para trabajar de forma continua. La red inteligente, la monitorización medioambiental y un sistema de detección de intrusos son las aplicaciones del IoT en las que se utiliza un esquema de sensores inteligentes sin cableado. En la arquitectura de las WSN consta de cinco capas, denominadas física, de enlace, de red, de transporte y de aplicación, como se observa en la Fig.2.4.

#### 2.2.8. Wireless Fidelity (Wi-Fi)

Es una red de comunicación no cableada que transmite la comunicación en forma de señales de radio. Tiene algunos problemas de seguridad y no ofrece un mecanismo de cifrado, por lo tanto es fácil para el atacante cambiar el mensaje.

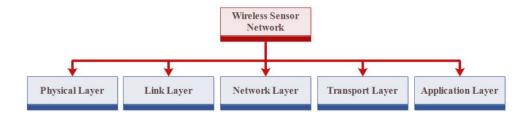


Figura 2.4: Arquitectura por capas de una red de sensores inalámbricos [Burhan et al., 2018]

El segundo inconveniente es el espionaje esto se refiere a la observación y monitorización no autorizada de la comunicación de otras personas, este proceso se encarga de recoger la información del usuario transmitida a través de una red[Grabovica et al., 2016].

#### 2.2.9. Redes 5G

Hay muchas tecnologías de comunicación como 2G/3G/4G, ZigBee, Bluetooth, RFID y WSN, que se utilizan en las aplicaciones de IoT, estas tecnologías de comunicación se enfrentan a muchos desafíos debido al gran número de dispositivos de conexión, los hilos de seguridad, las nuevas normas, la capacidad computacional de los dispositivos y las complicadas comunicaciones[Akpakwu et al., 2017]. Para hacer cara a estos desafíos la red 5G, se está convirtiendo rápidamente en una con IoT que abarca las necesidades del entorno y la industria inteligente[Nunez, 2018]. La 5ta generación de tecnología móvil asegura perfeccionar la velocidad, la cobertura y la capacidad de respuesta en la red inalámbrica. Debido a la consideración de la comunidad investigadora y del mundo académico sobre el IoT basado en el 5G, se informa de que el 5G completo estará disponible después de 2025[Association et al., 2015].

#### 2.2.10. Node-Red

A principios de 2013, Nick-O'Leary y Dave Conway-Jones, del equipo de servicios de Technology Services Team crearon Node-RED, Originalmente, era sólo una prueba de concepto (PoC) para ayudar a visualizar y entender el mapeo entre temas para traslado en telemetría de colas en mensajes (MQTT),

pero pronto, se convirtió en una herramienta muy popular que podía ampliarse fácilmente a diversos usos.

Node-RED fue creado por el grupo de Servicios de Tecnologías Emergentes de IBM, en otras palabras es un implemento de programación visual destinado al Internet de las cosas, también es utilizado para diferentes aplicaciones, y ensamblar muy rápidamente flujos de diferentes servicios. Está basado en Node.js (una plataforma de javascripting del servidor). Node-RED permite a los usuarios unir servicios web y dispositivos mediante la sustitución en tareas de codificación comunes y es posible con una interfaz visual de arrastrar y soltar [Kodali and Anjum, 2018].

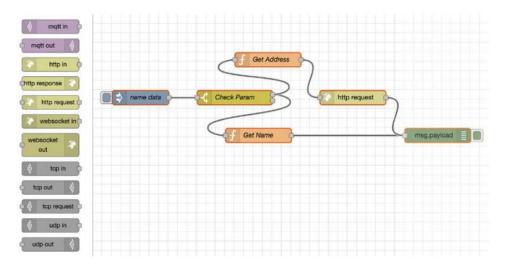


Figura 2.5: Node-Red Editor de flujo [Kodali and Anjum, 2018]

#### 2.2.11. Proyectos de Desarrollo con Node-Red

En [Bezeira et al., 2017] se desarrolla un sistema rentable que utiliza energía al mínimo y puede manejar diversos sitios como casas, industrias y otros lugares públicos, todos los componentes están conectados a la nube, los datos llegan a la nube y su respuesta es inmediata, el usuario puede establecer sus preferencias según su necesidad.

En la investigación de [Chayapathy et al., 2017] indican que los efectos de instalar tecnología doméstica inteligente debe justificar sus costes. La automatización del hogar de la casa debe ser rentable, costos mínimos, fácil de

usar, de instalar y flexible con diversas infraestructuras de red y aparatos.

#### 2.2.12. Indicadores Clave de Rendimiento (KPI)

Aparece como un número o valor que puede compararse con un objetivo interno, o con un objetivo externo "benchmarking" para dar una indicación del rendimiento. Ese valor puede referirse a los datos recogidos o calculados de cualquier proceso o actividad, la selección de una serie de medidas de rendimiento que sean apropiadas y concretas para una empresa debe hacerse a la luz de las intenciones estratégicas de la misma, que se habrán formado para adaptarse al entorno competitivo en el que opera y a la naturaleza del negocio. Por ejemplo, si el liderazgo técnico y la innovación de los productos van a ser la fuente clave de la ventaja competitiva de una empresa manufacturera, ésta debería medir sus resultados en este ámbito en relación con sus competidores. Pero si una empresa de servicios decide diferenciarse en el mercado acerca de la calidad en el servicio, entonces, entre otras cosas, debería supervisar y controlar el nivel de calidad deseado [Ahmad and Dhafr, 2002].

Tanto si la empresa pertenece al sector manufacturero como al de servicios, al elegir una gama adecuada de medidas de rendimiento será necesario equilibrarlas, para asegurarse de que no se hace hincapié en una dimensión o conjunto de dimensiones del rendimiento en detrimento de otras. Estas medidas pueden representar[Zairi, 1994]

- un indicador de rendimiento financiero (rendimiento empresarial);
- un indicador de rendimiento técnico (medición de la productividad);
- un indicador de eficiencia (medición de la contribución humana).

Está muy claro que al centrarse sólo en los resultados, se puede conseguir más perjuicio que beneficio. Las métricas serán absolutas y no explican si el desempeño es bueno o malo, esto podría ser muy perjudiciales en los negocios en cuestión a largo plazo[Zairi, 1994].

#### 2.2.13. Concepto Tiempo Medio entre fallas

El Tiempo Medio entre Fallas es aquel momento medio que sucede en medio de dos defectos/daños del equipo. Esto representa la confiabilidad en la operatividad del activo mientras más alto sea el MTBF, más confiable será el equipo[Alavian et al., 2019].

El MTBF se calcula mediante la diferencia entre el tiempo global de trabajo del activo (esto es el número de horas que funcionaría si no sufriera averías) y el tiempo de fallas, dividido por el número de averías por los que pasó. La fórmula sería:

$$MTBF = \frac{Tiempo\ Total\ de\ Trabajo\ -\ Tiempo\ de\ Avería}{Número\ de\ Fallos}$$

Figura 2.6: Fórmula para cálculo MTBF [Infraspeak, 2020]

#### 2.2.14. Concepto tiempo medio de reparaciones

El MTTR presenta el tiempo medio esencial en la reparación de las averías que sufre el equipo hasta que vuelva a una condición plenamente funcional. Por ello, es un gran indicador para llevar el performance del grupo de mantenimiento[Schneeweiss, 1981].

Se lo calcula dividiendo el tiempo general del mantenimiento correctivo en el transcurso de un determinado tiempo tomando en cuenta el número de intervenciones realizadas. La fórmula expresada:

$$MTTR = \frac{Tiempo\ Total\ de\ Mantenimiento}{Número\ de\ Intervenciones}$$

Figura 2.7: Fórmula para cálculo MTTR [Infraspeak, 2020]

Mientras más escaso el MTTR, superior será la perspectiva, si el MTTR es muy alto significa que entre mostrar y anunciar sobre la falla pasa mucho tiempo. Para dicho caso, se tiene que optar por la admisión de un Sistema para Gestión del Mantenimiento Asistido por computador y facilitar los reporte de fallas por parte de cualquier empleado.

#### 2.3. Marco Conceptual

#### 2.3.1. ¿Qué son las plataformas IoT?

A pesar de que la palabra **plataforma** es bastante negligente, con "**plataforma IoT**" se refiere a la base para que los dispositivos se conecten

entre sí y generen su propio entorno, tiene que ser apta para tolerar millones de conexiones sincronizadas de dispositivos y ser simple de configurar. Las plataformas IoT se basan en un programa que deja enlazar dispositivos tales como: sensores, actuadores y dispositivos industriales dentro de un medio digital, produciendo su propia red permitiendo a los mismos comunicarse y producir información.

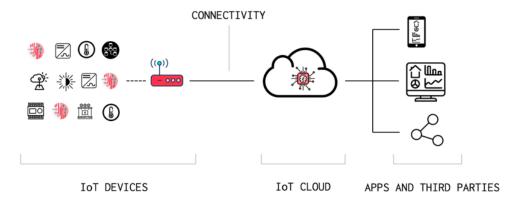


Figura 2.8: Esquema de Iot [Pérez-Bedmar, 2019]

La tecnología IoT comenzó a revolucionar la industria, ha favorecido la conexión a la red de un gran número de dispositivos que ejecutan funciones interconectadas y originan datos de gran valor dentro de la empresa. También permite producir aplicaciones para monitorizar todos estos dispositivos y administrarlos de la forma más adecuada y potenciar el negocio. A partir de la visualización técnica, gestionar a la práctica la implementación del IoT en la Industria no sería concebible sin que exista una plataforma idónea para lograrlo.

#### 2.3.2. ¿Cuáles son los elementos de las plataformas IoT?

A pesar de la existencia de múltiples tipos (plataformas IoT, plataformas IoT cloud) los elementos más importantes dentro de las diversas clases de plataformas IoT incluyen:

■ Hardware: Son los instrumentos y sensores con la función de captar la información en un entorno y desenvolverse en él. Un ejemplo, el sensor que se encarga de leer la temperatura ambiente o un instrumento que la mantiene siempre en determinados parámetros.

- Software: Es aquel con la misión de revisar la información enviada por un dispositivo local y la toma de decisiones, generalmente el programa se encuentra en la nube desarrollando Plataformas IoT.
- Conectividad: El uso de la red computacional facilita la transmisión de datos y organización desde el hardware a la nube.
- Interfaz de acceso: Es aquella que posibilita la comunicación entre usuarios y red IoT. Definiendo es el punto accesible del equipo humano a los datos recolectados por los otros elementos antes mencionados.

Los diferentes Brokers IoT permiten la conexión entre los elementos estudiados, impulsando el paso a los datos en las empresas. Desde ahora, es de mucha importancia considerar los datos y cambiarlos a información adecuada para la empresa, con ello se abre paso a tecnologías de adiestramiento y análisis de datos como el aprendizaje automático.

#### 2.3.3. Clases de plataformas Iot:

Esta no es una clasificación exhaustiva pero, más o menos se pueden dividir las plataformas de IoT en uno de estos diferentes tipos:

- 1. Apartado/Apartado:Brinda herramientas de equipos, programas, conectividad, fiabilidad y gestión de elementos para administrar millones de conexiones, facilita integraciones gestionadas (OTAS, gestión de dispositivos, conexiones) para conectar y supervisar una flota de dispositivos y herramientas para explotar los datos recogidos.
- 2. **Plataformas EDGE:**Proporcionan, principalmente, gestión de dispositivos y herramientas para supervisar y operar a distancia los mismos.
- 3. Plataformas de Administración en la conectividad: Muestra soluciones de coordinación a los enlaces con bajo coste por medio de las tecnologías Wi-Fi y celular.
- 4. **Plataformas de Datos**:Mezclan varias de las herramientas apropiadas en el enrutamiento de datos para los dispositivos en administración/visualización y el análisis de los mismos.

## 2.3.4. ¿De qué forma integrar plataformas Iot en una Organización?

La incorporación de plataformas IoT dentro de una organización demanda conocimientos múltiples en ámbitos como IT (Tecnología de Información), seguridad en comunicaciones y dispositivos, entre otras cosas. Es primordial contar con profesionales, calificados para efectuar las conexiones esenciales, elaborar datos, insertar algoritmos y enviar los datos necesarios a la Nube. Actualmente, es posible en las organizaciones que cuentan con otras plataformas y programas, se integren a nuevas plataformas IoT. Lo primordial es que la organización sea dueña de los datos adquiridos y los sistemas sean compaginables. Es de suma importancia un lapso de reflexión de la organización para tener en consideración los diversos factores en la selección de una plataforma IoT.

#### 2.3.5. ¿Que es MQTT?

Es un protocolo de comunicación pub/sub de código abierto creado por Andy Stanford-Clark y Arlen Nipper. Lo más habitual es que se ejecute sobre la pila TCP/IP, pero hay implementaciones de MQTT que utilizan otros protocolos (por ejemplo, Bluetooth). Dado que el protocolo maneja algo más que çolas de mensajesz "transporte de telemetría", se suele utilizar el término MQTT como nombre del protocolo en lugar de pensar en un acrónimo de un nombre más largo.

MQTT es eficiente en su utilización del ancho de banda y los recursos del sistema. Como resultado, es popular en casos de uso que van desde sistemas integrados de baja potencia hasta microservicios nativos de la nube. El modelo pub/sub es el núcleo de la comunicación MQTT, la información se organiza por temas y los dispositivos clientes pueden publicar datos en un tema para enviar mensajes a cualquier suscriptor. Del mismo modo, los clientes pueden suscribirse a un tema para ser notificados cuando se publique un mensaje.

#### 2.3.6. ¿Qué es un broker?

Un broker MQTT es una entidad de software central en la arquitectura MQTT. Actúa como un agente inmobiliario que primero comprueba los antecedentes de las partes implicadas y, después de asegurarse de que se cumplen las normas pertinentes, el agente inicia una transacción. Un broker MQTT hace lo mismo pero en lugar de transacciones monetarias, los brokers MQTT manejan transacciones de mensajes. En concreto, así es como los brokers MQTT facilitan las transacciones entre clientes MQTT:

- Permitir que los dispositivos (también llamados dispositivos cliente o simplemente clientes) realicen una solicitud de conexión
- Autenticar los dispositivos basándose en la información de conexión compartida por el o los dispositivos de conexión

- Una vez autenticado, asegurarse de que el dispositivo puede enviar/recibir mensajes hacia/desde otros dispositivos de forma segura utilizando el cifrado de la capa de transporte (TLS) (como una opción)
- Almacenan los mensajes en el servidor para que puedan volver a enviarse en caso de pérdida de conexión no deseada, al conectarse el cliente, al desconectarse el cliente, etc.

Hay un montón de brokers MQTT viables, tanto gestionados como autoalojados. A continuación se ofrece un resumen de algunas de las opciones más utilizadas ver tabla 2.1.

$\operatorname{Brokers}$	Address/Port	WebSocketSupport
AWS IoT Core MQTT	Asignación dinámica	Si, puerto:443
Mosquitto	test.mosquitto.org	Si,puerto:8081,8080
Mosca/Aedes	test.mosca.io	Si, puerto:3000
$\operatorname{HiveMQ}$	broker.hivemq.com	Si, puerto:8000,443
VerneMQ	autoasignado	Si, puerto:9001,9002
Azure IoT Hub	Asignación dinámica	Si, puerto:443
EMQX	autoasignado	Si, puerto:8083, 8084
ejabberd	Autoasignados v asignados	Si

Tabla 2.1: Brokers Mqtt

La lista anterior menciona los brokers MQTT autoalojados y gestionados más populares, y cada broker tiene sus propios pros y contras. Algunos brokers son adecuados para proyectos a pequeña escala, mientras que otros están hechos para aplicaciones a gran escala. Por ejemplo, los brokers como VerneMQ y similares están hechos específicamente para facilitar la escalabilidad del sistema.

#### 2.3.7. ¿Qué es un software de base de datos?

Un programa destinado a bases de datos se usa en la producción, edición y mantenimiento de fichas y registros esto facilita la creación, la incorporación de datos y edición de los mismos además de actualizaciones e informes. El programa maneja el archivo de datos, copias de seguridad, informes, control múltiple de accesos y seguridad, lo último mencionado es importante hoy en día para su uso en base de datos, ya que las filtraciones de información son cada vez más comunes. El software a veces se denomina DBMS (Sistema de Gestión de Bases de Datos).

#### ¿Qué es una base de datos en MySQL?

Se define como un sistema en la gestión de información relacional basado en SQL que es de código sin bloqueo. Esta orientado y desarrollado para la marcha de aplicaciones web y funciona en cualquier plataforma. De la forma que surgieron necesidades nuevas y variadas en Internet, el programa se ha transformado en la plataforma electa para desarrolladores web, esto se debe a su diseño en el manejo de millones de análisis y miles de transacciones, actualmente es la opción más popular en los negocios de comercialización electrónica que requieren administrar muchas transferencias. La transigencia bajo demanda es una característica clave del mismo, Ver Fig 2.9.

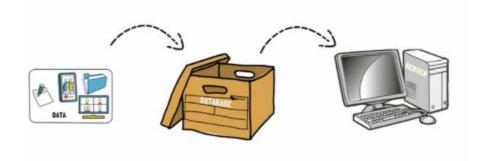


Figura 2.9: Base de Datos [techbf, 2019]

#### 2.3.8. Comunicación S7

#### Definición

Se utiliza para programar PLC, intercambiar datos entre PLC, acceder a datos de PLC desde sistemas SCADA (Supervisory Control and Data Acquisition) y para fines de diagnóstico. El primer byte siempre es 0x32 como identificador de protocolo. Los procesadores de comunicación especiales para la serie S7-400 (CP 443) pueden utilizar este protocolo sin las capas TCP/IP.

El Protocolo S7, sienta las bases en las comunicaciones de Siemens, su puesta en funcionamiento Ethernet se basa en ISO TCP (RFC1006) que, por diseño, está orientado a bloques.

Para establecer una conexión con un PLC S7 hay 3 pasos:

1. Conectar con el PLC en el puerto TCP 102

- 2. Conectar en la capa ISO (COTP Connect Request)
- 3. Conectar en la capa S7comm (s7comm.param.func: 0xf0, Configurar comunicación)

	OSI layer	Protocol
7	Application Layer	S7 communication
6	Presentation Layer	S7 communication
5	Session Layer	S7 communication
4	Transport Layer	ISO-on-TCP (RFC 1006)
3	Network Layer	IP
2	Data Link Layer	Ethernet
1	Physical Layer	Ethernet

Figura 2.10: Capas del protocolo S7

#### 2.3.9. Factory I/O

Este programa es la simulación de fabricas en 3D con la finalidad de que el alumno aprenda las diversas formas de automatización. Diseñado para uso sencillo, brinda la facultad de elaborar rápidamente fabricas virtuales usando la selección de materiales industriales comunes. El software incluye muchas plantas inspiradas en aplicaciones industriales frecuentes, con dificultades que van desde el novato hasta experto en automatización. La forma más común es utilizar Factory I/O como un medio de entrenamiento para controladores lógicos programables, debido a que los mismos son los elementos más comunes encontrados en aplicaciones industriales. No obstante, se pueden utilizar microcontroladores, SoftPLC, Modbus, entre otras periferias de comunicación.

Las principales características que hacen de Factory I/O un buen programa para el aprendizaje en tareas reales de control, a través de PLC, son:



Figura 2.11: Ejemplo de fabrica 3D

múltiples escenarios ambientados en tareas industriales de uso rutinario, se utiliza una biblioteca con varios componentes industrializados, con los mismos se desarrollan escenarios adaptados a un entorno virtual; estrategias en el diagnóstico de fallas, pueden ser a causa de un tomacorriente abierto o un cortocircuito; la seguridad y eficacia del software, ya que muestra aplicaciones industriales reales, evitando los problemas de costo, daños a personas y/o equipos.

## 2.3.10. ¿Qué es Node-RED?

Node-RED es una herramienta de PFC descrita hasta ahora, desarrollado por el grupo de Emerging Technology Services, Node-RED está ahora bajo la OpenJS Foundation.

### Programación en base al flujo (FBP)

El FBP tiene un estilo funcional de alto nivel para que el comportamiento del sistema pueda ser interpretado fácilmente por ejemplo, en un protocolo multipar distribuido, como un modelo de flujo de datos distribuido, para analizar con precisión los criterios y determinar si una variable o declaración se comporta correctamente, Ver Fig 2.12.

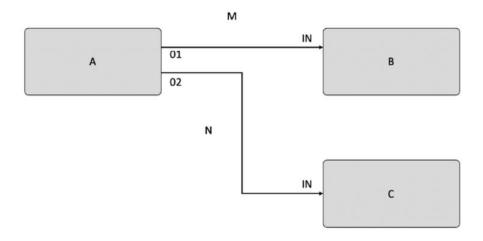


Figura 2.12: Ejemplo del diseño en FBP [Kodali and Anjum, 2018]

## 2.3.11. Ventajas de Node-RED

Es una herramienta FBP, adecuada para hacer control de datos para aplicaciones web e IoT. Su entorno de desarrollo y ejecución son aplicaciones basadas en el navegador hechas con Node.js, lo que hace que su desarrollo sea lo más fácil posible.

## Simplificación:

Al programar con Node-RED, notarás su simplicidad. Como el nombre de no-code-low-code indica, la codificación se elimina y la programación se completa intuitivamente con un número mínimo de operaciones que necesitan ser utilizadas.

### Eficiencia:

El FBP tipificado por Node-RED puede completarse casi sólo con operaciones de la interfaz gráfica de usuario, El editor de flujo de Node-RED se encarga de construir el entorno de ejecución para la aplicación, la sincronización de bibliotecas, el entorno de desarrollo integrado (IDE) y la preparación del editor para que el usuario pueda concentrarse en el desarrollo.

## 2.3.12. ¿Qué es JSON?

JSON es un formato de texto completamente independiente del idioma, pero utiliza convenciones conocidas por los programadores, que incluyen: C. Java javascript Perla Python Entre otros [Barrera, 2020]

#### Características de JSON

- JSON es solo un formato de datos.
- Requiere usar comillas dobles para las cadenas y los nombres de propiedades. Las comillas simples no son válidas.
- Una coma o dos puntos mal ubicados pueden producir que un archivo JSON no funcione.
- Puede tomar la forma de cualquier tipo de datos que sea válido para ser incluido en un JSON, no solo arreglos u objetos.
- A diferencia del código JavaScript, en el que las propiedades del objeto pueden no estar entre comillas, en JSON solo las cadenas entre comillas pueden ser utilizadas como propiedades.

El uso de JSON está creciendo rápidamente en la industria, ya que es una técnica de programación útil. Twitter cambió recientemente de XML a JSON. Google Web Toolkit también funciona con este formato.

## 2.3.13. ¿Qué es un Nodo?

Para entender el concepto de un nodo, se revisa nuevamente el concepto de NODE-RED, el cual es una herramienta para la programación de aplicaciones Node. js con herramientas de interfaz gráfica de usuario (GUI). Node-RED también sirve como un entorno para ejecutar software (Node-RED Flow) programado en Node-RED. Normalmente, cuando se programa con Node. js, el código fuente se escribe con un editor de código o un entorno de desarrollo integrado (IDE). Se genera un archivo ejecutable al construir el código fuente escrito (compilar, asociar con archivos de dependencia, y así etc.). En Node-RED, el procesamiento básico utilizado al programar con Node. js es proporcionado por partes implementadas llamadas nodos. En la programación orientada a objetos normal, estas partes pueden proporcionarse a menudo como archivos de biblioteca en forma de partes comunes.

En otras palabras, como ya hay piezas programadas (nodos), la programación se completa simplemente colocándolas en la GUI. La siguiente

figura 2.13 compara la programación pura de Node.js con la creación de flujos en Node-RED.

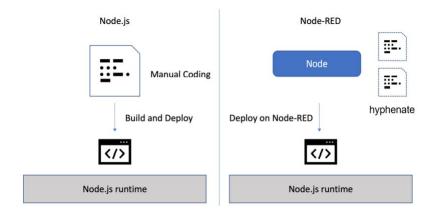


Figura 2.13: Node-red vs Node.js Programación [Kodali and Anjum, 2018]

#### 2.3.14. Node-RED Dashboard

Node-RED Dashboard es un módulo que proporciona un conjunto de nodos en Node-RED para crear rápidamente un panel de datos en vivo. Estos nodos requieren node. js versión 12 o más reciente. La última versión que soportaba node v8 era la 2.30.0.

El diseño de la pantalla debe mostrarse como una cuadrícula. Cada elemento del grupo tiene un ancho predeterminado de 6 ünidades" (una unidad tiene un ancho predeterminado de 48 píxeles, con un espacio de 6 píxeles).

El algoritmo de diseño de matriz siempre intenta colocar los elementos lo más alto y lejos posible en su contenedor; Esto incluye cómo se organizan los grupos en la página y cómo se organizan los widgets en grupos.

#### 2.3.15. Datos Técnicos de elementos y equipos

#### Controlador Lógico Programable (PLC)

Es una computadora industrial que utiliza la ingeniería para automatizar procesos para ayudar a la máquina a desarrollar eficientemente todos sus sistemas. Gracias a estas ventajas, el PLC se ha convertido en una herramienta indispensable para el desarrollo tecnológico de la industria y de todo el entorno



Figura 2.14: Ejemplo dashboard

social, en la fig. 2.15 se observa un modelo de la familia siemens S7-1511-1 PN, será utilizado para el desarrollo de este proyecto.



Figura 2.15: Controlador lógico programable S7-1500 [Siemens, 2021b]

Tabla 2.2: Características Técnicas S7-1500

Datos Técnicos

Datos Techneos	
Marca	Siemens
${f Modelo}$	CPU 1511-1-PN
Tensión de Alimentación	$24 \mathrm{VDC}$
Intensidad de entrada	$0.7 \mathrm{Amp}$
Área de datos remanente total	128 kbytes
Número de puertos	2, Rj45 Ethernet
Módulos por cada rack, máx	$32; \mathrm{CPU} + 31 \mathrm{\ m\'odulos}$
Display	$3,45~\mathrm{cm}$

## Módulo de entradas Digitales

En la fig 2.16 se muestra el módulo de entradas digitales, recibe señales tipo boolenas provenientes de sensores, botones y demás instrumentación del proceso a controlar, se adjuntan características técnicas en la tabla 2.3.



Figura 2.16: Módulo de Entradas Digitales [Siemens, 2021b]

## Módulo de Salidas Digitales

El módulo de salida digital permite al PLC operar con disparadores previos y receptores de comando encendido/apagado. El valor binario de la salida digital se convierte en la apertura o cierre del relé interno Ver Fig 2.17.

Tabla 2.3: Tabla Técnica para Entradas Digitales

Datos Técnicos Siemens, 2021bl

Datos Techneos Stemens, 20216	
Marca	Siemens
${f Modelo}$	6ES7521-7EH00-0AB0
Tensión de Alimentación	24VDC
Intensidad de entrada	3  mA
Bloque de entradas binarias	16



Figura 2.17: Bloque de Salidas binarias [Siemens, 2021c]

Tabla 2.4: Tabla técnica de Salidas Digitales

Datos Técnicos

Marca	Siemens	
${f Modelo}$	6ES7522-5HH00-0AB0	
Tensión de Alimentación	24VDC	
Intensidad de entrada	185  mA	
Número de Salidas Digitales	16	
Tipo de salida	relé	

## Módulo de Entradas Analógicas

El módulo de entradas analógicas convierte la cantidad analógica en un número almacenado en la variable interna del PLC. Lo que hace es conversión A/D, porque los PLC solo funcionan con señales digitales. Esta conversión se realiza con cierta precisión o exactitud (número de bits) y en cada período de tiempo determinado (período de muestreo) las unidades de entrada analógica

pueden leer el voltaje o el amperaje.

Tabla 2.5: Tabla Técnica para Entradas Analógicas

Datos TécnicosSiemens [2021a]		
Marca	Siemens	
${f Modelo}$	6ES7531-7KF00-0AB0	
Tensión de Alimentación	24VDC	
Intensidad de entrada	$240~\mathrm{mA}$	
Número de Entradas Analógicas	8	
Tipo de Entrada	V,I,RTD	



Figura 2.18: Módulo de Entradas analógicas [Siemens, 2021a]

## Capítulo 3

# Desarrollo del Proyecto

Durante este capítulo se detallaran los diferentes softwares utilizados para la simulación del proyecto, configuraciones de cada uno de los programas para realizar el monitoreo de variables y posterior visualización en el dashboard del Node-RED instalado en la plataforma de IBM - Cloud.

## 3.1. Diseño del experimento

En la realización del presente proyecto se utilizan varios tipos de metodologías, una de ellas es la descriptiva que permite analizar las diversas plataformas Iot y de software existentes y escoger la más adaptable para la ejecución del proyecto.

Toda la información se obtuvo por medio de la metodología de investigación bibliográfica, se realizo una discriminación de proyectos científicos, artículos científicos, revistas y libros guía que ayudaron a profundizar en gran medida acerca de las tendencias actuales en Industria 4.0 y su conectividad con las plataformas ya instaladas en la nube, esto permitió al desarrollo eficaz del proyecto.

En los casos desde el segundo hasta el cuarto objetivo específico, planteados en el capítulo uno se utiliza la metodología experimental, en donde se obtienen respuestas acerca del funcionamiento de las plataformas IoT, se comprueba la interacción entre las variables locales y su visualización en la nube por medio de la supervisión, control y monitoreo de forma remota durante la simulación del proceso ejecutado por Factory I/O.

## 3.2. Desarrollo

En la Fig 3.2, se puede observar el esquema red utilizado para el desarrollo del proyecto, se opto por instalar node red en el computador local, se comunica con el simulador PLC-SIM Advanced y a su vez se supervisa el proceso en factory I/O, mediante el nodo IBM Watson se envian los datos de la producción actual y por medio del broker cloud mqtt se leen los datos guardados en MySQL.

Para la base de datos en MySql, el servidor es el mismo computador se considera en el dashboard la opción de una tecla para borrar lo almacenado en la base de datos y así optimizar los recursos del computador ya que no es un servidor para almacenar grandes cantidades de datos.

En la nube de IBM se instala node red para realizar otro dashboard y poder realizar la comunicación vía mqtt y watson iot, esto sirve para que el usuario final pueda visualizar el dashboard de la producción cuando se encuentre fuera de la planta, en este caso se implementa la validación por medio de usuarios y contraseñas, esto se realiza para aumentar los niveles de seguridad y un agente externo no pueda ingresar sin validar su identidad.

En cuanto a los reportes solo se podrán descargar por medio del node red local, no se realiza en el ibm cloud debido a que no se tiene una ruta fija para el almacenamiento del archivo pdf generado desde el dashboard.

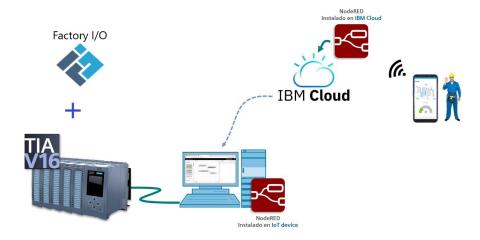


Figura 3.1: Esquema para red utilizada

En la base de datos, se guardan datos de la producción actual en este caso el mecanizado de tapas y bases, estos datos se observan en el dashboard desarrollado en Node-red dando al usuario visualización del proceso, están presentes dos barras verticales que aumentan de acuerdo al producto manufacturado dependiendo la línea de producción a la que esta ligada como se observa en la Fig. 3.2.

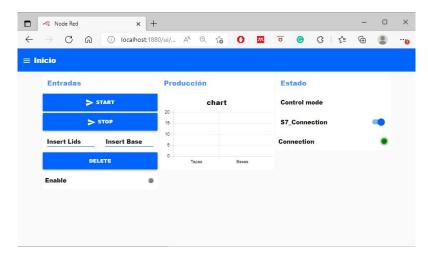


Figura 3.2: Dashboard en Red Local

## 3.3. Software utilizado

### 3.3.1. Tia Portal V16

El software desarrollado por la empresa Siemens, utiliza una automatización totalmente integrada es decir aplica los avances tecnológicos a todos los niveles desee la gestión hasta la fabricación, las innovaciones se centran en la integración, continuidad del desarrollo. Estos incluyen estándares, técnicas de integración distribuida entre equipos y pruebas de integración funcional, esto autoriza a los usuarios aumentar la calidad de su software, acortar el tiempo de actividad y mininizar los costos de ingeniería.

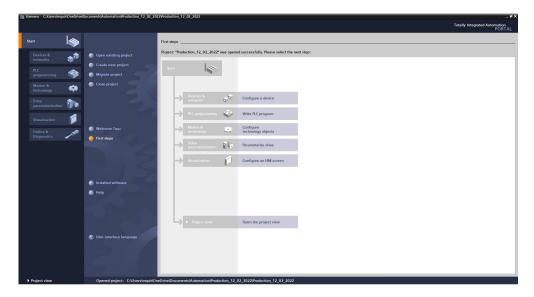


Figura 3.3: Entorno de Programación

En la Fig. 3.3, se observa la vista de configuración para el nuevo proyecto, se puede agregar un nuevo dispositivo, agregar nuevos bloques para programación, configurar objetos tecnológicos, parametrizar drives y configuración de HMI.

## 3.3.2. Creación de un nuevo proyecto

Lo primero a tener en cuenta antes de la creación de un nuevo proyecto, es verificar los paquetes de software instalados para asegurarse de no tener complicaciones durante la configuración del proyecto. Ver Fig. 3.4.

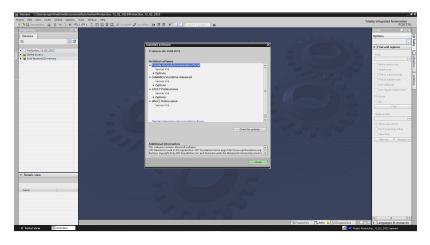


Figura 3.4: Verificación de Software instalado

Una vez realizada la verificación de software, se procede a crear nuestro nuevo proyecto en blanco se despliega una nueva ventana Fig. 3.5, se encuentran las diferentes CPU y drivers fabricados por Siemens con sus versiones mas actualizadas.

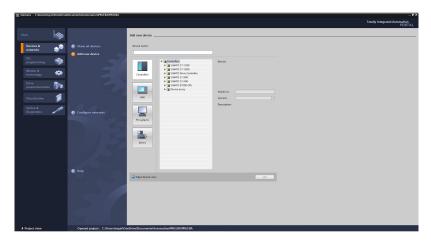


Figura 3.5: Vista de dispositivos

Se selecciona el CPU Simatic S7-1500 a utilizar en el proyecto, para este caso el S7-1511-1-PN con sus módulos de entradas y salidas digitales para realizar la lectura de sensores y accionamiento de motores dentro del proceso en Factory  $\rm I/O$  Ver Fig.3.6

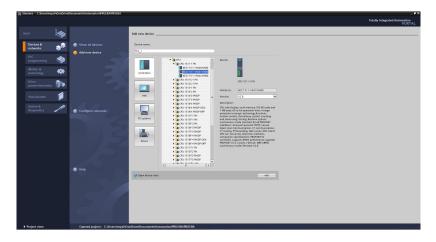


Figura 3.6: Selección de dispositivo

Una vez se agrega el dispositivo en el proyecto, se accede automáticamente a la ventana de configuración de dispositivos, aquí se realizan los ajustes de hardware en el dispositivo tales como:

- Configuración de interfaz Profinet y asignación de IP en el dispositivo
- Asignación de racks para entradas/ salidas digitales y entradas analógicas
- Direccionamiento de todas las entradas y salidas ubicadas en cada rack
- Configuración de velocidad en comunicaciones del dispositivo TCP/IP
- Habilitación de seguridades en hardware para evitar accesos no autorizados

En la pestaña vista de redes, se ven todos los dispositivos creados en el proyecto, en nuestro caso solo se tiene el plc S7-1500 creado para manejar y clasificar todo el proceso de manufactura durante su ejecución y posterior finalización, en caso de requerir otro autómata simplemente se agrega de forma directa desde las librerías pre-configuradas y se realizan las conexiones en caso de comunicarse vía TCP/IP simplemente se da clic en el puerto ethernet de cada dispositivo y se crea la conexión, en casos de configuración de otras redes tales como: Profibus, ASi, Can etc. la puesta marcha cambia, se deben agregar librerías en su mayoría tipo GSD acorde al fabricante del dispositivo.

En la carpeta del proyecto con nombre production, se da clic en propiedades y se realiza la configuración para cargar la programación o cambios que se ejecuten en la misma durante el desarrollo del proyecto, debido a que se utiliza el simulador PLC-SiM Advanced y podrían existir conflictos al cargar nuevas lineas de programación Ver Fig. 3.7.

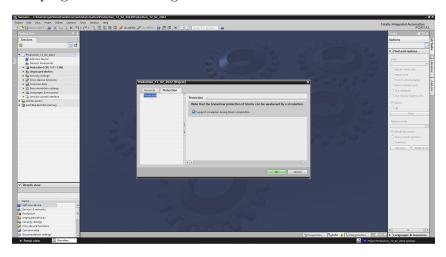


Figura 3.7: Configuración del Proyecto

### OB (Bloques Organizacionales)

Entender el concepto de OB's y el mecanismo de su ejecución es crucial para entender todo el concepto de los PLC de Siemens. Hay un montón de bloques de organización y el propósito principal de todos ellos es correr el programa realizado por el usuario en los intervalos de tiempo dados o en base a diferentes eventos. Así que, en pocas palabras, los OB's le ayudan a organizar su programa de usuario basado en diferentes criterios, estos determinan la secuencia (eventos de inicio) mediante la cual se ejecutan las secciones individuales del programa. Una llamada a un OB puede interrumpir la ejecución de otro dependiendo de su prioridad, los OBs de mayor prioridad pueden interrumpir a los OBs de menor prioridad. Estos bloques son para diferentes tareas, organización del programa, fallos de hardware, fallos de software, interrupción cíclica y mucho más. Para la organización del programa se necesita siempre el OB1 como OB principal. El bloque de organización (OB) es la interacción entre el sistema operativo y el programa madre. El mismo puede almacenarse en el OB1 y ser llamado periódicamente por el sistema operativo (programa lineal) o el programa puede dividirse y almacenarse en varios bloques (programa estructurado).

A continuación se tiene un resumen de todos los bloques organizacionales:

■ OB1 (bloque de organización principal): Se ejecuta todo el tiempo y de nuevo. Puede ser interrumpido por otros bloques de organización Ver Fig. 3.8.

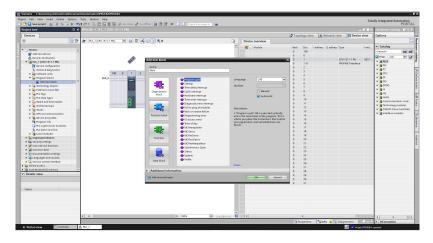


Figura 3.8: Bloque OB01

■ OB30 - OB38 (bloques de organización de interrupciones cíclicas): Interrumpen el OB1 a intervalos de tiempo regulares, si quieres ejecutar tu programa de usuario cada 100ms, se pone el código en el OB35 Ver Fig. 3.9.

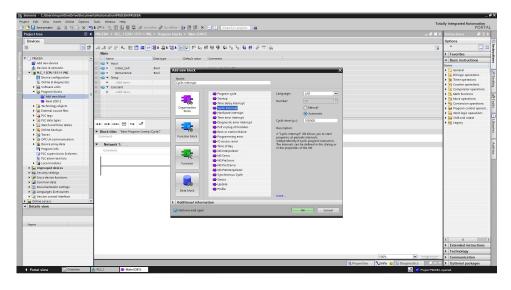


Figura 3.9: Bloque OB30-OB38

■ OB100 (bloque de organización de reinicio completo): Se ejecuta una vez después del reinicio completo de la CPU. Puede utilizar este OB para programar alguna rutina que desee que ocurra sólo una vez al reiniciar la CPU, por ejemplo para cargar los parámetros de planta por defecto Ver Fig. 3.11.

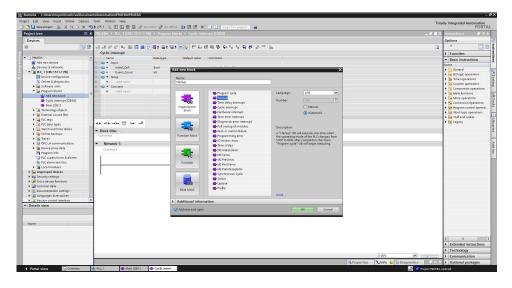


Figura 3.10: Bloque OB100

- OB40 (Interrupción de hardware): Se configura el OB40 para esta entrada y se programa algún código que se ejecutará una vez cada vez que el estado de la señal cambie de 0 a 1. Otro ejemplo puede ser un contador que se incrementa cada vez que el estado de la señal cambia.
- OB10 (Interrupción horaria): Se requiere que algo suceda todos los días a las 22:00. se configura y utiliza el OB10 para este propósito.

Como conclusión los OB's se encargan de la ejecución del programa del usuario, y su existencia le ayudara como programador a organizar tu código de una manera útil y lo más eficiente posible.

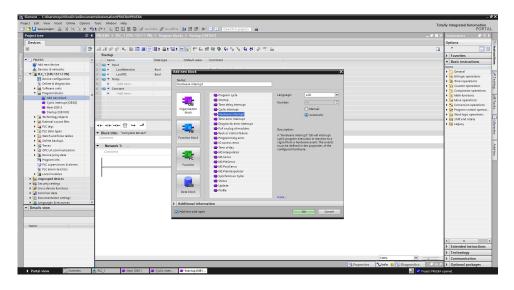


Figura 3.11: Bloque OB40



Figura 3.12: Bloque OB40

## FB (Bloque de función)

Es una subrutina capaz contener una secuencia o acción o ser llamada desde otro bloque como los estudiados anteriormente. A diferencia de FC, el bloque está vinculado por defecto a la instancia de DB que almacena los valores de FB, los datos se llaman STATIC y solo el FB los tiene, este bloque también puede ser creado por el usuario y llamado tantas veces como sea necesario.

También se asignan diferentes parámetros en el bloque de funciones como In, Out, In-Out, Temp, y Stat, que es un parámetro adicional en el FB, los datos guardados en el bloque de datos de instancia no se pierden después de la ejecución del FB. Las variables estáticas se almacenan en el bloque de datos de instancia. Sin embargo, los datos Temp se almacenan en la pila de datos local, que se pierde después de la ejecución del FB. Por lo tanto, cuando se llama a un FB y ejecuta su lógica preprogramada, sus datos se almacenan en el DB de instancia asociado con ese FB, por lo que la próxima vez que se llame al bloque, los datos en vivo primero se recargarán tal como estaban contenidos en el DB.

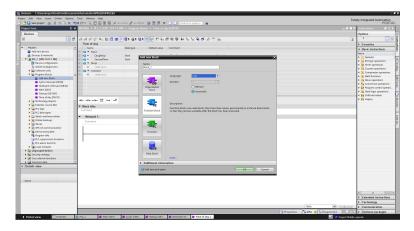


Figura 3.13: Bloque de Función FB

## FC (Función)

Es un subprograma que puede contener una cadena específica y puede llamarse desde otro bloque. La particularidad es que no contiene fichas de datos cíclicos, por consiguiente si en él hay algún tipo de programación de tipo escritura o realiza una operación, puede ser útil en esta relación y ejecutar solo esta y en la siguiente llamada al bloque FC, el resultado de la operación se pierde, este bloque puede ser creado por el usuario y llamado tantas veces como

sea necesario, se pueden asignar diferentes parámetros en la función como In, Out, In-Out, Temp.

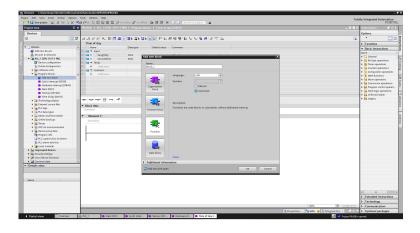


Figura 3.14: Bloque de Función FC

## ¿Que es un Bloque de Datos (DB)?

Es en el que no se pueden escribir ningún tipo de instrucción, solo se los trabaja para almacenar datos de lectura/escritura y se puede acceder desde cualquier parte de la programación, desde otro bloque o desde ciertas operaciones.

### Clases de Bloque de Datos (DB)

Existen en Tia portal dos tipos de bloques de datos:

- Bloque de Instancia: Instance DB es el bloque responsable del almacenamiento de datos y solo está asociado con un FB, el FB vinculado será el único bloque que puede modificar el valor de la instancia de DB.
- Bloque DB Global: Todo DB puede ser leído y escrito por cualquier bloque, y cada bloque u operación puede modificar los valores almacenados en el bloque Global DB, estén relacionados o no.

Antes de programar, es de suma importancia familiarizarse con el significado de variables locales, globales, temporales y estáticas.

Las locales son a las que se accede desde el bloque al que están asignadas, si algún otro intenta acceder o utilizar estas variables, no será accesible

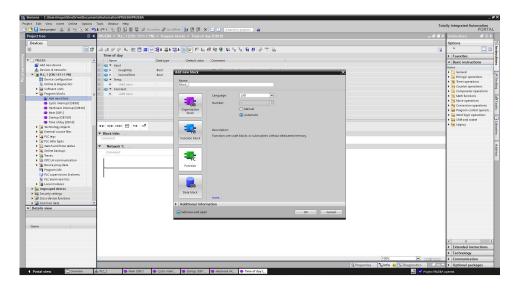


Figura 3.15: Bloque de Datos DB

directamente. Como sugiere el nombre, las variables globales son variables que se pueden usar o acceder desde cualquier bloque del proyecto.

En el TIA Portal, las variables globales son aquellas que se direccionan como entrada, salida o memoria (E, Q, M); por otro lado, en el TIA Portal, las variables locales se agregan a cada FB o FC y se distinguen agregando un "hash" delante de su nombre.

Las etiquetas temporales se encuentran en FC y FB y representan datos que se eliminan después de un ciclo de lectura del PLC en este bloque, ya que solo son válidas en este momento de ejecución. Por otro lado, las variables estáticas son exclusivas de FB porque se almacenan en su instancia de base de datos asociada, por lo que se puede acceder al valor o a los datos almacenados de forma permanente, independientemente de si se vuelve a llamar a FB o no.

Dominar estas capacidades permitirá aprovechar al máximo lo que un PLC tiene para ofrecer mientras se reducen los recursos disponibles en el controlador, como la memoria, la programación estructurada y profesional es un requisito en la industria actual porque reduce los errores, el tiempo de inactividad y la capacidad de moverse. en nuevas industrias de tendencia como la Industria 4.0

### 3.3.3. S7-PLCSIM Advanced

Con SIMATIC S7-PLCSIM Advanced, puede crear un controlador virtual para emular un procesador S7-1500 o ET 200SP y emular todas las funciones. Esto elimina la necesidad de un controlador real para probar los programas de

#### STEP 7.

Los controladores virtuales también se pueden probar y verificar en el contexto del sistema o máquina. Las interfaces de programación de aplicaciones (API) admiten periféricos y se utilizan para conectar controladores virtuales a máquinas virtuales o modelos de fábrica o aplicaciones programadas en lenguajes de alto nivel. Esto hace posible combinar el controlador virtual con la verificación del programa STEP 7 en software de simulación como SIMIT o NX Mechatronics Concept Designer.

S7-PLCSIM Advanced de Siemens permite la simulación de funciones durante la configuración y la ingeniería sin una conexión física. En general, PLCSIM Advanced implementa muchas mejoras con respecto a PLCSIM, incluyendo la simulación para el servidor web, OPC UA, IO distribuida y muchas más.

Pasos a seguir en la instalación y desarrollo de simulación:

## ■ Descarga PLCSIM Advanced:

La descarga en PLCSIM Advanced es un poco diferente para un desarrollador acostumbrado a usar PLCSIM. En el PLCSIM estándar, puedes simular usando el icono en el Portal, mientras que en PLCSIM Adv, tienes que seguir algunos pasos más He expuesto el proceso paso a paso a continuación:

1.-Iniciar una simulación a través del icono de la bandeja



Figura 3.16: Vista inicial PLC SIM

- 2.-Asigne el nombre, la dirección IP, la máscara de subred y arranque el PLC
- **3.-**Descargue desde el Tia Portal como si estuviera descargando a un PLC normal. Asegúrese de utilizar el adaptador de red que seleccionó en



Figura 3.17: Ventana PLC-SIM Advanced

el paso anterior (PLCSIM o PLCSIM Virtual Ethernet Adapter)

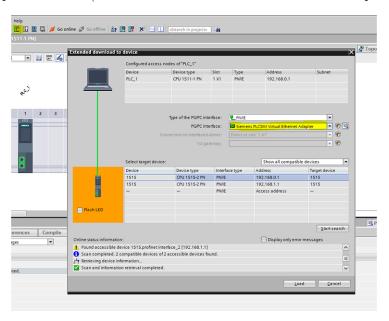


Figura 3.18: Descargar en dispositivo

## • Selección de la simulación

Asegúrese de que está simulando un PLC S7-1500 o un ET200SP. A partir de la V14, PLCSIM Advanced no admite ningún otro PLC o IO distribuido.

#### Versión del firmware del PLC

Tener en cuenta de que el firmware de su PLC es una versión anterior a la 2.1. Actualmente, si tiene un PLC con la versión 2.1, recibirá un código de error "0020 0 2 0". Para solucionar este error, cambie el firmware de su dispositivo a la versión 2.0 o anterior.

## Utilizar el adaptador Ethernet virtual PLCSIM

Si está probando la comunicación OPC o la simulación de un servidor web, asegúrese de utilizar el Adaptador Ethernet Virtual de PLCSIM, este es un requisito para simular un servidor OPC.

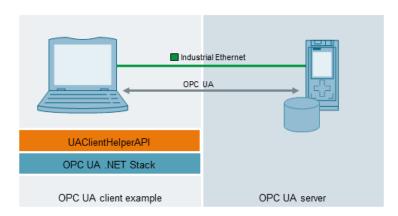


Figura 3.19: Comunicación OPC

### • Seleccione la configuración correcta

Para descargar en PLCSIM Advanced, tendrá que marcar la casilla **apoyar la simulación durante la compilación.** Para utilizar los bloques de la biblioteca, deberá marcar esta casilla y compilar antes de añadir el bloque a la biblioteca.

Para ello, haga clic con el botón derecho del ratón en la carpeta del proyecto en Tia Portal, seleccione la opción propiedades, se desplegará una ventana emergente con dos pestañas; En la primera se tienen las propiedades generalizadas del proyecto, mientras en la segunda pestaña está el apartado de protección del proyecto, aquí se activa la opción asistir la simulación durante la compilación de bloques.

#### Ventajas de PLC-SIM Advanced

El uso de PLC-SIM ofrece numerosas ventajas entre las cuales se tiene:

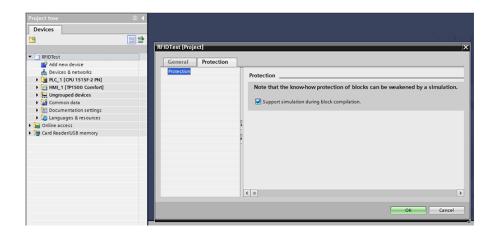


Figura 3.20: Propiedades para simulación

- La mejora en la calidad de nuestros proyectos de automatización gracias a la detección temprana de errores
- Se evitan costos de hardware por el entorno de simulación
- Ofrece tiempos de respuesta reducidos
- Reducción de los riesgos habituales en la puesta de servicio inicial
- Se refuerza la formación del operador antes de la puesta en servicio
- Se aumenta la eficiencia de la producción optimizando los componentes de la programación
- Incremento de la eficiencia al momento de reemplazar los componentes de la maquinaria.
- Adaptación durante la expansión de plantas ya existentes

PLC-SIM Advanced posee comunicación distribuida por TCP/IP, es decir que sus instancias pueden comunicarse por medio de un conmutador virtual con otros dispositivos, la comunicación es posible con CPU y HMI reales o simuladas, el conmutador virtual de PLC-SIM deberá activarse en el adaptador **Ethernet virtual PLC-SIM** para que sea visible en la red.

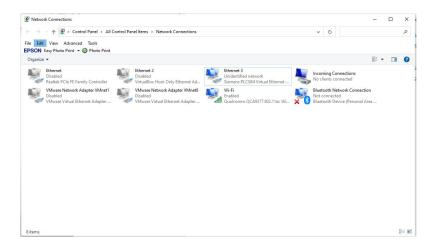


Figura 3.21: Adaptador de Red virtual Siemens

## 3.3.4. Procedimiento para instalación de NODE-RED

En este subtema, se aprenderá a instalar Node-RED en el ordenador local, tanto si ya sea en Windows, Mac o en una Raspberry Pi. Si se desea utilizar Node-RED en Windows, se debe instalar npm y Node.js a través del siguiente sitio web https://nodejs.org/en/home-downloadhead. Ver Fig.3.22



Figura 3.22: Sitio web Descargar Nodejs

Se obtiene el instalador de Node.js para windows directamente desde allí,

luego se comienza el proceso de instalación con los siguientes pasos:

- Hacer clic en el archivo msi que se ha descargado para empezar a instalar Node.js, se incluye la versión actual de versión de npm, Node-RED se ejecuta al mismo tiempo que Node.js, por lo que es necesario.
- Basta con hacer clic en los botones de las ventanas de diálogo según el asistente de instalación, aunque hay que tener en cuenta algunos puntos durante la instalación.
- A continuación, debe aceptar el Acuerdo de licencia de usuario final. Ver Fig.3.23
- También puede cambiar la carpeta de destino de la instalación. Ver Fig.3.24
- En la siguiente pantalla no es necesario realizar ninguna configuración personalizada, se puede seleccionar siguiente con sólo las características por defecto seleccionadas Ver Fig.3.25
- En la siguiente pantalla, puede hacer clic en Siguiente sin marcar nada. Sin embargo, está bien instalar las herramientas que se pueden seleccionar aquí. Esto incluye las instalaciones y la configuración de la ruta de estos entornos (Visual C++, windows-build-tools, y Python). Ver Fig.3.27
- Comprobar las versiones de sus herramientas con los siguientes comandos en la ventana del cmd cuando la instalación de Node.js haya terminado. Ver Fig.3.28

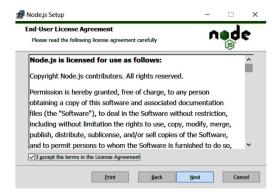


Figura 3.23: Instalación Node.js

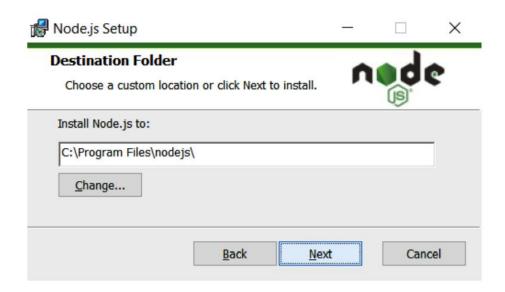


Figura 3.24: Selección Carpeta Instalación [Kodali and Anjum, 2018]

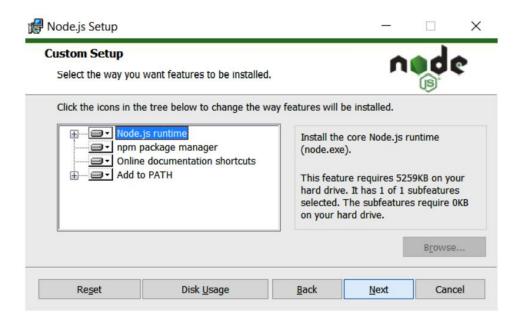


Figura 3.25: Tipos de Instalación [Kodali and Anjum, 2018]

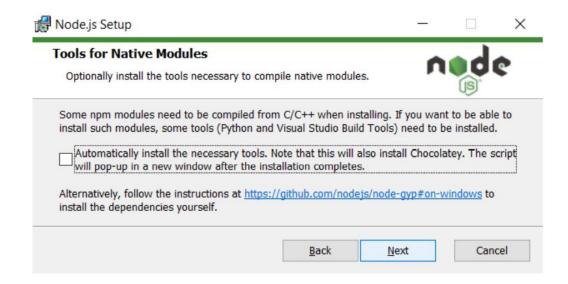


Figura 3.26: Herramientas nativas para Nodejs

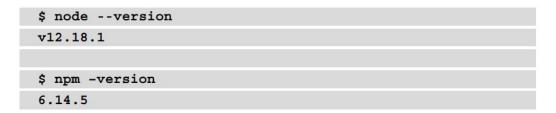


Figura 3.27: Códigos de Confirmación [Kodali and Anjum, 2018]

Cuando las instalaciones de Node.js y npm estén completas, se puede comprobar sus números de versión, con ello ya se puede instalar Node-Red. Dependiendo del proyecto, es común que el funcionamiento sea estable con una versión antigua de Node.js, sin embargo el desinstalar la versión actual de Node.js e instalar la versión deseada de Node.js cada vez que se cambia de proyecto lleva tiempo, por lo tanto al utilizar windows, se recomienda usar una herramienta de gestión de versiones de Node.js como nodist.

Una vez que haya terminado de instalar Node-RED, se puede utilizar ejecutando el comando visualizado en la fig.3.28, después de ejecutar este comando, reconocerá la URL que se utiliza para acceder al editor de flujo de Node-RED. Normalmente, se asignará localhost (127.0.0.1) con el puerto

asignado por defecto 1880.



Figura 3.28: Comando de ejecución Node Red

Figura 3.29: Comando Node-Red Ejecutado

Antes de utilizar un navegador web para visualizar el editor de flujos, asegurarse que el puerto 1880 no este bloqueado para accesos web, se abre el panel de control Fig.3.30 y se selecciona la casilla Windows Firewall, se despliega una ventana con múltiples opciones en donde se hace clic en opciones avanzadas, aparece otra ventana llamada windows defender firewall with advanced fig. 3.32 rules, hacer clic a inbond rules en este caso ver Fig. 3.33

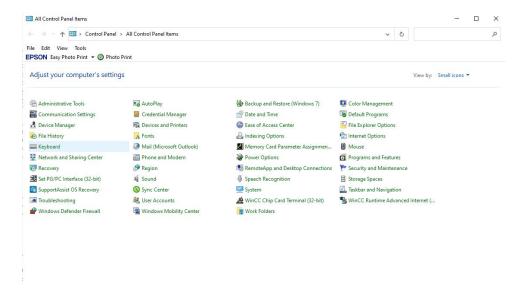


Figura 3.30: Panel de control

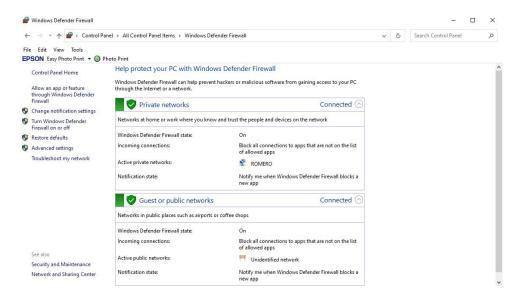


Figura 3.31: Ventana de opciones Firewall

Dado clic en la opción anterior se desplegara una lista con diferentes reglas ya establecidas por el fabricante del sistema operativo como se aprecia en la fig 3.33, se debe crear una nueva regla para dar nombre a nuestro puerto a utilizar se abrirá una nueva ventana con múltiples opciones ver fig. 3.34,

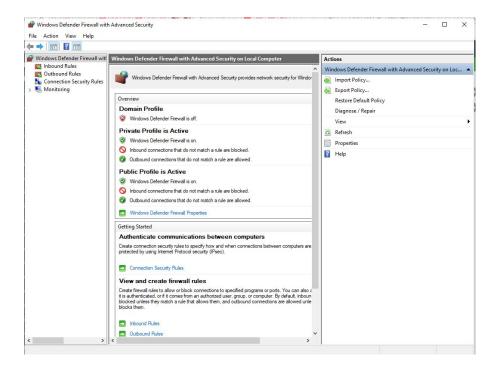


Figura 3.32: Opciones avanzadas Firewall

realiza la pregunta que tipo de regla se va a crear, dar clic en la opción **puerto** clic a siguiente y aparece una nueva ventana de diálogo en donde se específica el número del puerto tipo TCP como se detalla en la Fig. 3.35, dar clic en siguiente hasta llegar a la opción nombre, aquí se otorga un identificativo al puerto que se crea, ver Fig. 3.36, se da clic en finalizar y se creara la nueva regla con el puerto preconfigurado. Ver Fig.3.37

Realizando todos estos pasos se podrá visualizar el diagrama de flujo y el dashboard creado por el desarrollador dentro de la red industrial en cualquier dispositivo que se conecte a dicha red sin la necesidad de tener instalado Node-Red como se observa en la Fig. 3.38, en este caso cada dispositivo tiene su propia dirección Ip, la pc o dispositivo sin node-red accede a la ip del computador principal en el puerto tcp al que esta direccionado la ejecución de Node-Red.

#### 3.3.5. Editor de flujo

Se crea un flujo utilizando el Editor de Flujos de Node-RED. Al crear un flujo sencillo, se entenderá cómo utilizar la herramienta y sus características, a partir de ahora se crean las aplicaciones llamadas flujos utilizando Node-RED,

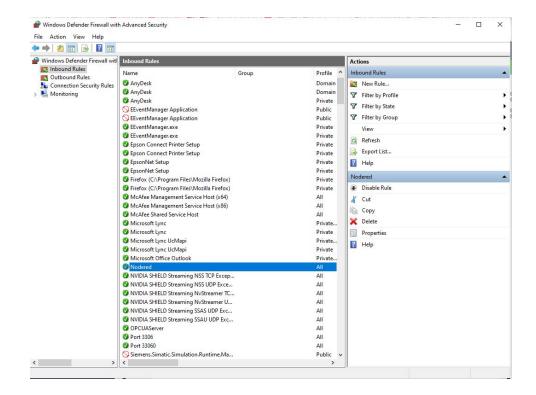


Figura 3.33: Inbound Rulesl

para ello se cubrirán los siguientes temas:

- Mecanismos del editor de flujo de Node-RED
- Uso del editor de flujos
- Realización de un flujo para una aplicación de tratamiento de datos
- Importar y exportar un Editor de flujos

## Mecanismos del editor de flujo de Node-RED

Node-RED tiene dos partes lógicas: un entorno de desarrollo de desarrollo llamado Editor de Flujo y un entorno de ejecución para ejecutar la aplicación que se ha creado allí, estos se llaman el tiempo de ejecución y el editor respectivamente, se describe en detalle a continuación:

■ Tiempo de ejecución: Esto incluye un tiempo de ejecución de la aplicación Node.js, es responsable de ejecutar los flujos desplegados.

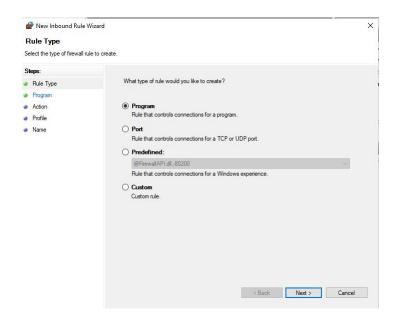


Figura 3.34: Creación de nueva regla

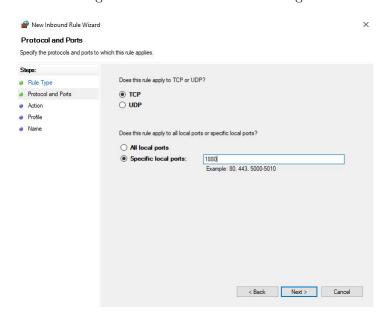


Figura 3.35: Configuración del puerto

■ Editor:Se trata de una aplicación web en la que el usuario puede editar sus flujos.

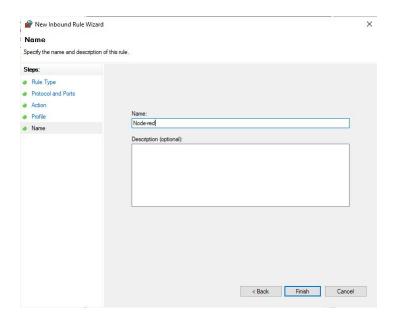


Figura 3.36: Nombre del puerto

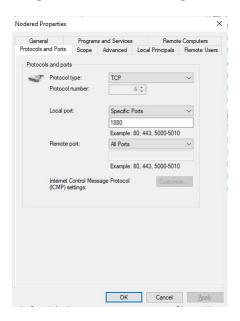


Figura 3.37: Puerto Configurado

El paquete instalable principal incluye ambos componentes, con un servidor web para proporcionar Flow Editor, así como una API REST Admin para



Figura 3.38: Visualización de Node-red en otro dispositivo

administrar el tiempo de ejecución. Internamente, estos componentes pueden ser instalados por separado e incrustados en aplicaciones Node.js existentes, como se muestra en el siguiente diagrama:

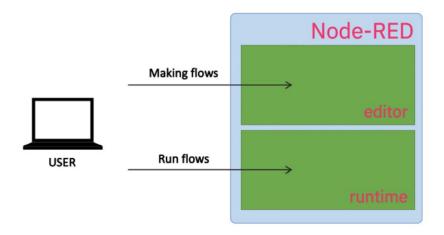


Figura 3.39: Componentes de Node.js [Hagino, 2021]

### Uso del editor de flujos

Las principales características del Editor de Flujo son las siguientes:

- Nodo: El principal bloque de construcción de las aplicaciones Node-RED, representa piezas de funcionalidad bien definidas.
- Flujo: Es una serie de nodos conectados entre sí que representan la serie de pasos que los mensajes pasan dentro de una aplicación.

- Panel de nodos: Una colección de nodos que están disponibles dentro de el editor que puede utilizar para construir cualquier aplicación.
- Botón de compilación: Se Pulsa este botón para compilar las aplicaciones una vez que se editen.
- Barra lateral: Es Un panel para mostrar varias funciones, como la configuración de los parámetros de procesamiento de procesamiento, especificaciones y visualización del depurador.
- Pestañas en barras laterales: Configuración de cada nodo, salida estándar, gestión de cambios, etc.
- Menú Principal: En el se encuentran la eliminación de flujos, importación/exportación de definiciones, gestión de proyectos, etc.

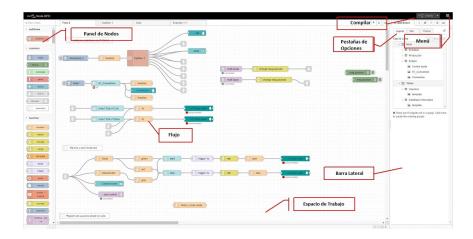


Figura 3.40: Ventana de trabajo en Node-Red

Es necesario entender lo que contiene el menú Flow antes de empezar a utilizar NodeRED, su contenido puede variar, dependiendo de la versión de Node-RED que esté utilizando, pero tiene algunos elementos de configuración como la gestión de proyectos de flujo, la vista de organización, la importación/exportación de flujo, la instalación de nodos publicados en la biblioteca, etc., que son universales.

# Realización de un flujo para una aplicación de tratamiento de datos

Se creara una aplicación en funcionamiento (llamada flujo en Node-RED), ya sea el Internet de las cosas (IoT) o el procesamiento de servidores como una

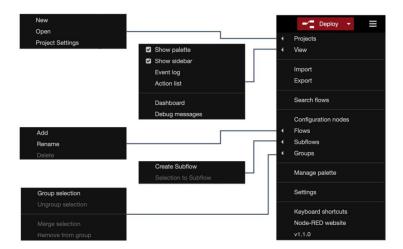


Figura 3.41: Vista interior del Menú Principal en Node-Red [Kodali and Anjum, 2018]

aplicación web, la operación básica que realiza Node-RED es la transferencia secuencial de datos. Aquí se crea un flujo en el que los datos JSON son generados de una forma pseudo-perfecta, y los datos son finalmente a la salida estándar a través de algunos nodos en Node-RED.

Hay muchos nodos en la parte izquierda de la paleta, se encuentra fácilmente el nodo de inyección, como se muestra en la siguiente figura 3.42.

Este nodo puede inyectar un mensaje en el siguiente, para ello se deberán seguir los pasos descritos:

- Arrastrar y soltar en el área de trabajo del Flujo 1, actualmente el nodo está etiquetado con la palabra timestamp. Esto se debe a que su carga de mensaje por defecto es un valor de marca de tiempo, se pueden cambiar los tipo de datos, en nuestro caso se utiliza JSON.
- Hacer doble clic en el nodo y se cambia su configuración cuando se abra el panel de propiedades del nodo:
- Se hace clic en el menú desplegable del primer parámetro y se escoge JSON, se pueden editar los datos JSON haciendo clic en el botón (...) de la derecha.
- Al dar clic en el botón (...), se abrirá el editor JSON donde se podrán crear datos JSON con un editor basado en texto o un editor visual.

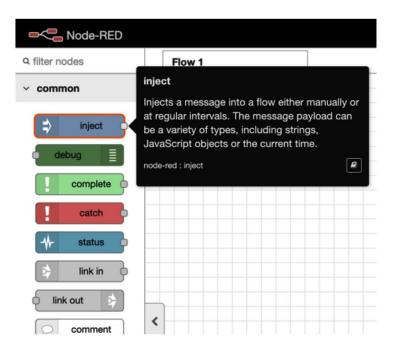


Figura 3.42: Vista de Nodo Inject en Paleta de Nodos

- Esta vez, se escriben datos JSON con un elemento llamado "nombre":
   Carlos y se visualizan las pestañas del editor JSON fig.3.44 y el editor visual fig.3.45
- Se dará clic en el botón **Done** y se cerrará este panel.
- Del mismo modo, se coloca un nodo de depuración en el área de trabajo.
- Después de colocarlo, se conectan los nodos Inject y Debug. Una vez que se ejecute este flujo, los datos JSON que fueron pasados desde el nodo Inject serán enviados a la consola de depuración (salida estándar) por el nodo Debug. Ver Fig. 3.46
- Por último, hay que desplegar el flujo creado, en el Editor de Flujos de Node-RED, se desplegan todos nuestros flujos en el espacio de trabajo en el tiempo de ejecución de Node-RED haciendo clic en el botón desplegar en la esquina superior derecha.
- Antes de ejecutar el flujo, se debrá seleccionar la pestaña de depuración del menú de nodos para habilitar la consola de depuración, como se muestra en la Figura. 3.47

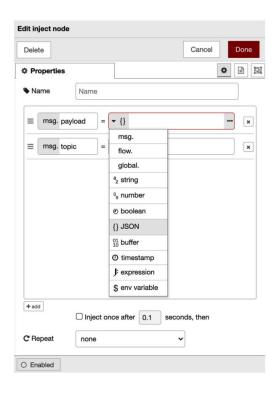


Figura 3.43: Vista de edición del Nodo



Figura 3.44: Pestaña Editor Json

Finalmente se realiza la ejecución este flujo, se da clic en el interruptor del



Figura 3.45: Pestaña Editor Visual

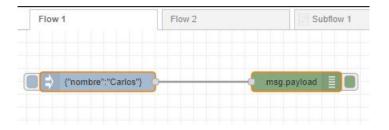


Figura 3.46: Ventana de Flujo



Figura 3.47: Visualizar mensaje de nodo Inject

nodo Inject para ver el resultado de la ejecución de el flujo en la consola de depuración como muestra la Fig. 3.48

El ejemplo que se trato anteriormente es un argumento sencillo en comparación a lo realizado en el desarrollo de proyecto, brinda una idea en la forma de programación con este software basado en flujo gramas.

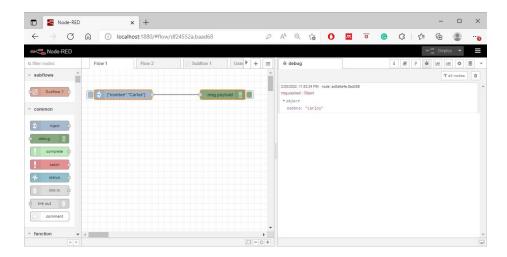


Figura 3.48: Ejecución de Nodo Inject

# 3.3.6. Comunicación entre Node-Red y Tia Portal

En la ejecución del proyecto se utilizan los nodos S7-comm:Read y S7-comm:Write, para leer y escribir datos respectivamente desde un DB (Data Block) en Tia Portal, esto sirve para monitorear el proceso desde Factory I/O, se debe deshabilitar el acceso optimizado al bloque de datos esto se configura en las propiedades del mismo como se muestra en la figura 3.49 dar clic en la opción atributos y se retira el visto de la casilla acceso optimizado al bloque.

Una vez realizado el paso anterior se procede a crear las variables a utilizarse dentro del bloque, se establece un tipo de dato estructura como se observa en la figura 3.50, esta estructura de datos puede repetirse dependiendo del proceso con variables del mismo nombre, se evita crear nuevos nombres para variables similares ejecutadas en un mismo proceso, simplemente se cambia el nombre de la estructura. Ver Fig. 3.51

En comparación con las estructuras, los tipos de datos del PLC se definen en todo el controlador en el TIA Portal y pueden modificarse de forma centralizada, todos los lugares de uso se actualizan automáticamente.

Dentro de la estructura de datos se declaran las variables a enlazarse con el editor tal como se muestra en la figura 3.52, por medio del simulador se observara el estado de cada una y se determinara si hay algún tipo de fallo en la comunicación interna, además se pueden visualizar los valores ingresados y actuales de la producción.

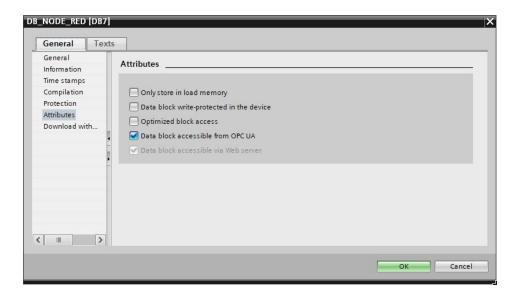


Figura 3.49: Data Block sin acceso Optimizado

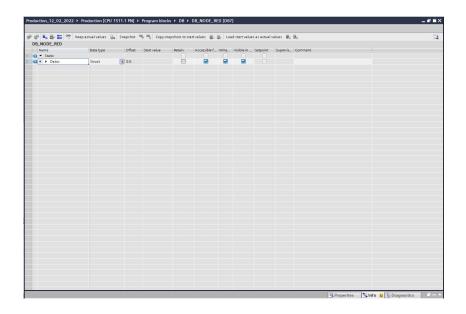


Figura 3.50: Variable Estructura

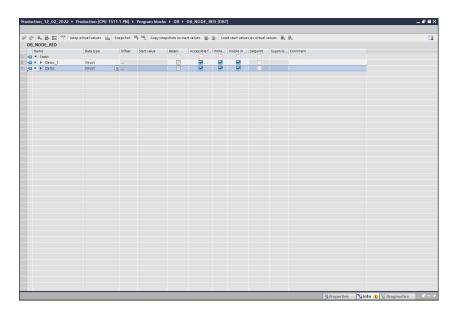


Figura 3.51: Estructura Copiada

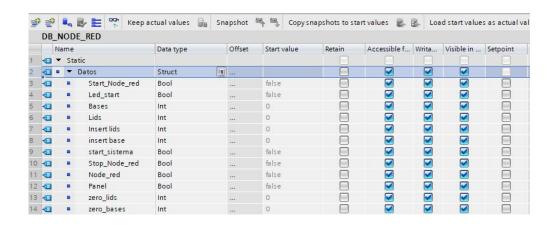


Figura 3.52: Variables de lectura y escritura Node-Red

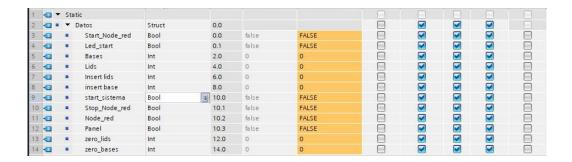


Figura 3.53: Valores Iniciales

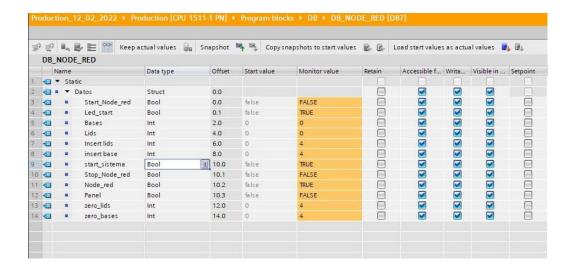


Figura 3.54: Valores Ingresados para producción

En los valores seteados por el usuario, se tiene el número de bases y tapas a realizar durante la producción, también se tiene el feedback de cuantas se van realizando y almacenando, además se pueden observar las variables booleanas para dar marcha o stop a la producción. Ver Fig.3.54

El uso de bloques de datos, simplifica ampliamente el uso de la memoria del dispositivo y el diagrama ocupacional de memoria lo certifica, allí se observa la utilización de bits de memoria interna, timers, entradas y salidas, se puede ver en la fig. 3.55,3.56 que no se han utilizado muchas marcas del sistema.

Cuando se activa el acceso a un DB como acceso estándar, se dispondrá de todos y cada uno de los valores creados dentro del DB individualmente seleccionables en cuanto a su remanencia. Es decir, que se seleccionará uno a uno el dato deseado.

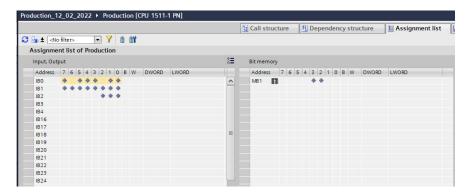


Figura 3.55: Plano Ocupacional Byte Entradas y Marcas del sistema

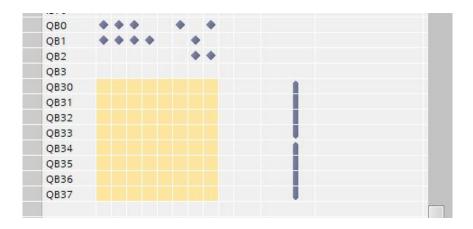


Figura 3.56: Plano Ocupacional Byte de Salidas

#### Nodo s7-comm

Node-red-Contrib-s7comm está diseñado para interactuar con el controlador SIEMENS SIMATIC S7-300/1200/1500 basado en el protocolo de comunicación RFC1006. Node S7comm puede establecer conexión con SIMATIC-S7. Además, se pueden leer/escribir direcciones de PLC con algunos tipos de datos S7.

Se deberán seguir los siguientes pasos:

- Primero instalarlo desde el manage pallete, en la pestaña install se lo busca con el nombre Node-red-contrib-s7comm
- Segundo se procede a instalar el nodo
- Tercero una vez instalado el nodo, se busca en el panel de nodos
- Cuarto se arrastra el nodo al espacio de trabajo. Ver Fig.3.57
- Quinto se da doble clic en el nodo para abrir la ventana de edición. Ver Fig.3.58
- Sexto dar clic sobre el ítem connection, se desplegara una nueva ventana.
- Séptimo en la nueva ventana se agrega la dirección Ip del PLC, el número de puerto, rack y slot, esos datos se pueden obtener en la pestaña configuración de dispositivos en Tia portal. Ver Fig. 3.59

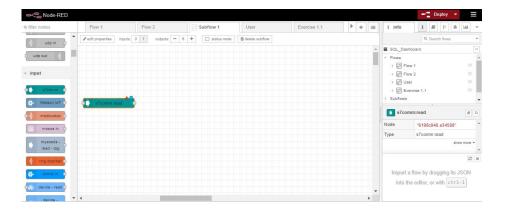


Figura 3.57: Nodo S7-COMM

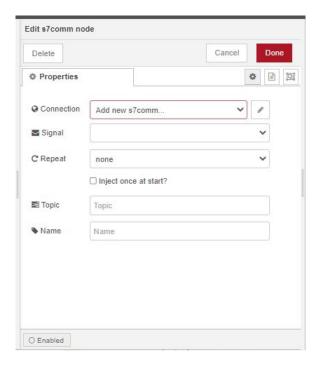


Figura 3.58: Nodo S7-COMM edición

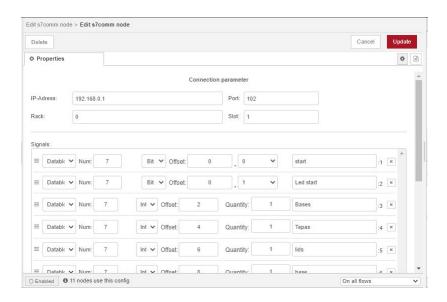


Figura 3.59: Configuración de Dispositivo Nodo S7-COMM

Dentro del apartado señales, se agregan las variables utilizadas en el proyecto, en este caso la primera opción es el bloque de datos, luego se escribe el número que designa Tia Portal por defecto a su vez se selecciona el tipo de dato puede ser un bit, Byte, Word, DWord, Real, entero, etc, se transcribe el offset proporcionado por Tia Portal y al final se utiliza un nombre para definir la variable a emplear, cabe recalcar que se debe tener el mismo nombre designado en el data block para así evitar inconsistencias al momento de la compilación.

	DB_	NOL	E_RED									
	1	lame		Data type	Offset	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint
	<b>1</b>	▼ Static										
2	<b>1</b>		Datos	Struct	0.0						<b>~</b>	
	1		Start_Node_red	Bool	0.0	false	FALSE	-	<b>~</b>	<b>~</b>	<b>M</b>	
	€00		Led_start	Bool	0.1	false	FALSE				<b>✓</b>	
	€		Bases	Int	2.0	0	4			<b>~</b>		
	1		Lids	Int	4.0	0	4			<b>~</b>	<b>~</b>	
	-01		Insert lids	Int	6.0	0	0		$\overline{\mathbf{w}}$	<b>~</b>	<b>~</b>	
	-€00		insert base	Int	8.0	0	0					
	1		start_sistema	Bool	10.0	false	FALSE					
0	-01		Stop_Node_red	Bool	10.1	false	FALSE				<b>~</b>	
1	1		Node_red	Bool	10.2	false	FALSE		$\overline{\mathbf{w}}$	<b>~</b>	<b>~</b>	
2	1		Panel	Bool	10.3	false	FALSE					
3	1		zero_lids	Int	12.0	0	0		<b>✓</b>	<b>~</b>		
4	1		zero_bases	Int	14.0	0	0	[000]		<b>M</b>		

Figura 3.60: Valores de Offset por defecto en Tia Portal

## 3.3.7. Desarrollo del Dashboard

En el desarrollo de la interfaz con Node-red, se tiene el dashboard, se desplaza hacia abajo en la sección de nodos, se observa un conjunto de nodos llamados dashboard como se muestra en la siguiente figura 3.61:

Los nodos de la sección del cuadro de mandos proporcionan widgets que aparecen en la interfaz de usuario (UI) de la aplicación, la interfaz de usuario está organizada en pestañas y grupos, las pestañas son diferentes páginas en tu interfaz de usuario, como varias pestañas en un navegador. Dentro de cada pestaña tienes grupos que dividen las pestañas en diferentes secciones para que puedas organizar tus widgets.

Para crear una ficha y un grupo, se siguen las siguientes instrucciones vease en la Fig. 3.62:

- En la esquina superior derecha de la ventana de Node-RED se tiene una pestaña llamada dashboard.
- Seleccione esa pestaña (1). Para añadir una pestaña a la interfaz de usuario, haga clic en el botón + pestaña (2).

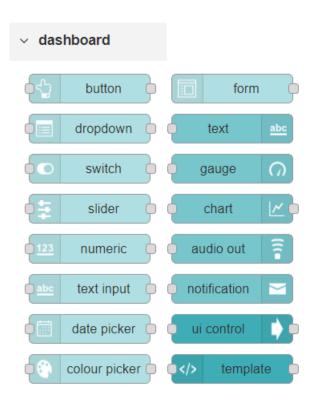


Figura 3.61: Nodos de Dashboard

• Una vez creada, se puede editar la ficha pulsando el botón de edición (3).

Puede editar el nombre de la pestaña y cambiar su icono:

Después de crear una pestaña, se pueden hacer varios grupos bajo esa pestaña, se necesitara crear al menos un grupo para añadir los widgets. Para añadir un grupo a la pestaña creada, tienes que hacer clic en el botón +grupo (4).

A continuación en la Figura 3.64, puede editar el grupo creado haciendo clic en el botón de edición (5).

# Temas en Dashboard

El panel de control de Node-RED tiene un fondo blanco y una barra azul claro por defecto. Puede editar sus colores en la pestaña Tema en la esquina superior derecha como se muestra en la siguiente figura 3.65.

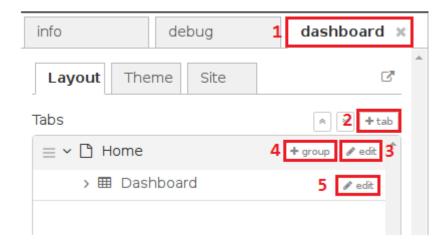


Figura 3.62: Menú de dashboard

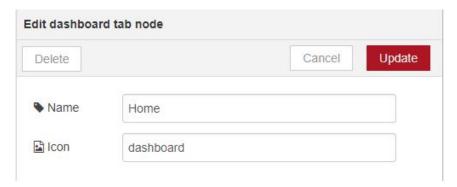


Figura 3.63: Editar nombre de Dashboard



Figura 3.64: Editar grupo Dashboard

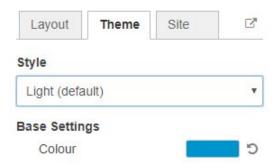


Figura 3.65: Editar Tema en Node-Red

Una vez realizado los cambios en la pestaña temas, se compila con el botón Deploy y se observa cómo la interfaz de usuario del panel de control cambia sus colores, en especial por el fondo y la barra de tareas, cabe recalcar el ejemplo obtenido en la siguiente figura 3.66 es otra configuración que viene por defecto en Node-red, se hacen varias combinaciones de colores acorde a nuestra necesidad.

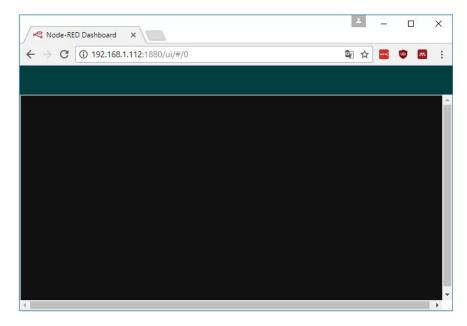


Figura 3.66: Cambio de colores en panel de control Node-Red

A continuación se detallan los nodos utilizados para el desarrollo del dashboard:

# Nodo MySql

Este nodo utiliza la operación de consulta contra la base de datos configurada, esto permite tanto las inserciones como los borrados. El msg. topic debe contener la consulta de la base de datos, y el resultado se devuelve en msg. payload, normalmente el valor devuelto será un array de las filas resultantes.



Figura 3.67: Nodo MySql

## Nodo Botón

Cuando se utiliza, se produce un botón estándar en el dashboard, el diseño del botón es bastante simple actualmente se puede configurar con viñetas y cambiar el color de fondo y así se mejora su visualización.



Figura 3.68: Nodo Botón

# Nodo Texto

Este nodo mostrará los campos de texto que no se pueden editar en la interfaz de usuario, todo mensaje recibido de msg.payload refrescará el texto basándose en el valor de formato dado, el campo Value Format se poda utilizar en cambios al formato de visualización y puede contener filtros HTML y Angular válidos.



Figura 3.69: Nodo Texto

#### Nodo Función

El nodo Función es un arreglo que ejecuta los avisos recibidos de otro nodo, el mensaje se envía como un objeto de JavaScript denominado msg, que por convención tiene una propiedad msg.payload que contiene el cuerpo del mensaje y se espera que la función devuelva un objeto de mensaje (o varios objetos de notificación), pero puede detenerse sin devolver nada al flujo de datos, la pestaña On Start abarca el código que se correrá cada vez iniciado el nodo, mientras la pestaña On Stop alberga el código que se ejecutará cuando se detenga el nodo. Ver fig 3.70



Figura 3.70: Nodo Función

## Nodo Plantilla

El widget de plantilla puede contener cualquier html válido y directivas Angular/Angular-Material, este nodo se puede utilizar para crear un elemento de interfaz de usuario dinámico que cambie su apariencia en función del mensaje de entrada y pueda enviar mensajes de vuelta a Node-RED, en la Fig 3.72 se observa un ejemplo de la plantilla utilizada para las tablas en el dashboard.



Figura 3.71: Nodo Plantilla

## **Nodo Switch**

Este nodo añade un interruptor a la interfaz de usuario, cada cambio realizado en el estado del interruptor generará un msg.payload con los valores On y Off especificados. El color de encendido/apagado y el icono de encendido/apagado son campos opcionales. Si todos ellos están presentes, el interruptor por defecto será reemplazado por los iconos correspondientes y sus respectivos colores, en nuestro proyecto se utiliza como condición para ejecutar la conexiones de red entre Node Red, MySql y Tia Portal. Ver Fig 3.74

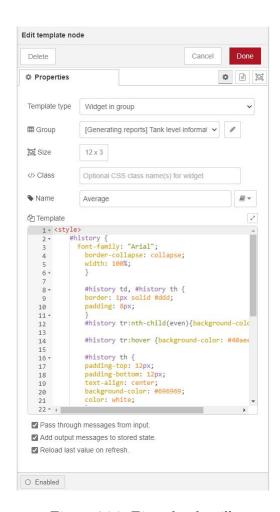


Figura 3.72: Ejemplo plantilla



Figura 3.73: Nodo Switch



Figura 3.74: Configuración Switch

# Nodo Gráficos

En la Fig 3.76 se observa la configuración para la serie proveniente de la base de datos de MySql, es un histórico de la producción y pruebas realizadas para la correcta ejecución del proyecto.



Figura 3.75: Nodo Gráficos



Figura 3.76: Configuraciones de Gráfica

# **Nodo Medios**

Este nodo permite mostrar imágenes, archivos de audio y vídeo en el Dashboard, gestionar la carga de archivos y mostrarlos a los usuarios del Dashboard, en la pestaña de Configuración, además de las reglas comunes del Dashboard, la configuración de Layout permite controlar la disposición de una imagen, y se establecen opciones de Vídeo relativas a la reproducción del mismo

Los archivos multimedia se pueden cargar en la pestaña archivos, donde se pueden crear categorías y cargar los archivos en Node-RED, para que puedan ser visualizados en el Tablero de Control posteriormente. El campo Enlace genera entonces la cadena que debe enviarse a cargar para mostrar ese medio en particular. También se puede dejar sin seleccionar ningún medio (después de subirlos), para que no se muestre ningún medio por defecto al desplegar. Ver Fig 3.77, 3.78



Figura 3.77: Configuración Nodo Medios



Figura 3.78: Configuración archivos de Medios



Figura 3.79: Nodo Medios

# Nodo RBE

El nodo de informe por excepción (RBE), sólo pasa los datos si el mensaje ha cambiado. También puede bloquear a menos que, o ignorar si el valor cambia por una cantidad especificada (modo Dead- y Narrowband).

Los modos de ancho de banda bloquearán el valor entrante a menos que su cambio sea mayor o igual que mas o menos que la diferencia de rango respecto a un valor anterior.

Los modos de banda estrecha bloquearán el valor entrante, si su cambio es mayor o igual que mas-menos la brecha de banda del valor anterior. Es útil para ignorar los valores atípicos de un sensor defectuoso, por ejemplo.



Figura 3.80: Nodo RBE

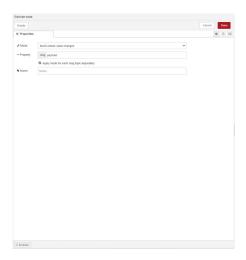


Figura 3.81: Configurar nodo RBE

Esto funciona en base a cada tema del mensaje. Esto significa que un solo nodo rbe puede manejar varios temas diferentes al mismo tiempo.

## Nodo Catch

Si un nodo arroja un error mientras maneja un mensaje, el flujo normalmente se detendrá. Este nodo puede ser utilizado para atrapar esos errores y manejarlos con un flujo dedicado.

Por defecto, el nodo atrapará los errores lanzados por cualquier nodo en la misma pestaña. También puede dirigirse a nodos específicos, o configurarse para que sólo atrape los errores que no hayan sido ya capturados por un nodo de captura "dirigido". Ver Fig3.83

Si se lanza un error dentro de un subflujo, el error será manejado por cualquier nodo de captura dentro del subflujo. Si no existe ninguno, el error se propagará hasta la pestaña en la que se encuentra la instancia del subflujo.



Figura 3.82: Nodo Catch

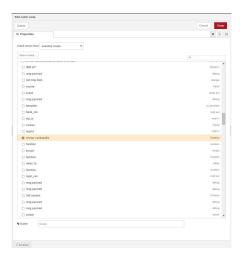


Figura 3.83: Configuración nodo Catch

#### Nodo S7-Comm Read

El nodo lee los datos de los dispositivos Siemens, como el S7-300/400/1200/1500, las señales a leer se configuran en el selector de conexiones del nodo y la función de repetición permite enviar peticiones de forma periódica o si se desactiva activada por cualquier mensaje de entrada del nodo. Ver Fig.3.84



Figura 3.84: Nodo S7-Comm-Read

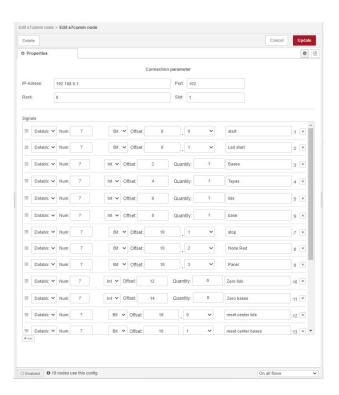


Figura 3.85: Configurar Nodo S7-Comm

En la Fig. 3.85, se observan las señales que se reciben desde el bloque de datos DB-7 configurado en Tia Portal, acorde a los nombres asignados en el DB-7 se colocan los mismos datos en Node-Red para evitar los conocidos traslapes de información.

En la opción IP address se coloca la dirección asignada a la CPU S7-1500 configurada en TIA Portal en este caso es 192.168.0.1, el puerto por defecto es el 102, con el rack 0 y slot 1.

#### Nodo S7-Comm Write

El nodo escribe datos en dispositivos Siemens como el S7-300/400/1200/1500, las señales a escribir se configuran en el selector de conexión del nodo. Para escribir un valor se introduce un mensaje con valor msg.payload como objeto JSON de tipo array. Ver Fig 3.86



Figura 3.86: Nodo S7-Comm-Write

Las configuraciones en este nodo son exactamente iguales al anterior, en este caso se escribe un valor desde NODE-RED hacia la variable direccionada en el DB-7 configurado en Tia Portal.

## Nodo Indicador

Se Añade un widget de tipo indicador a la interfaz de usuario, el msg.payload busca un valor numérico y se formatea de acuerdo con el Formato de Valor definido, que luego puede ser encerado utilizando otra variable a disposición del usuario. Ver Fig. 3.87, 3.88



Figura 3.87: Nodo Indicador

## Nodo SubFlujo

Un flujo secundario es una colección de nodos colapsados en un solo nodo en un lienzo que se puede usar para reducir la complejidad visual de un flujo o para agrupar un conjunto de nodos en un flujo reutilizable que se puede usar en muchos lugares. Ver Fig.3.89

También es posible convertir la selección actual de nodos a un subflujo seleccionando la opción "Subflow -> Selection to Subflow" en el menú. Los nodos

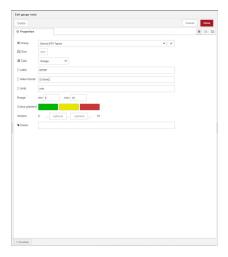


Figura 3.88: Configuración Nodo Indicador

se moverán a un nuevo subflujo y se reemplazarán por un nodo de instancia de subflujo dentro del flujo Ver Fig3.90.

La forma más fácil es usar la propiedad de sublínea. Estas son propiedades que se pueden establecer en una instancia de sublínea y aparecen como variables de entorno en el subflujo.



Figura 3.89: Nodo Subflujo

Haciendo repaso con los nodos ya estudiados anteriormente se desarrollara el dashboard de control local, se tienen campos para insertar texto en este caso el número de tapas y bases que se requieren manufacturar, se tiene un switch como condición preliminar para activar la comunicación S-7 y la comunicación con la base de Datos en MYSql, una vez accionado el switch se habilitan dos botones uno de marcha y uno de paro cambiando su color de fondo indicando que el proceso esta listo para empezar, cabe recalcar que al momento de insertar el número de elementos a manufacturar por cada linea de mecanizado, la simulación en Factory I/O pasará a modo RUN, se habilitan las gráficas de producción, Kpi Tapas y Kpi Bases, en producción se visualizan el número de Tapas y bases que pasaron el proceso de manufactura por medio de la CNC y se están almacenando, actualmente se cuenta con 2 botones para

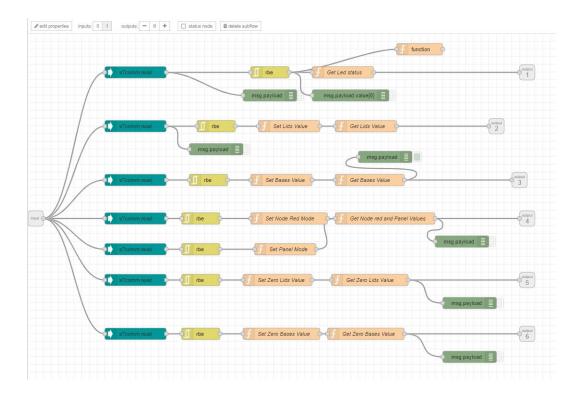


Figura 3.90: Ejemplo de Subflujo creado por el autor

simular la falla en la CNC de cada línea de producción, esto se debe a que es necesario registrar cada vez que la maquinaria sufre algún desperfecto y así se calculan nuestros indicadores claves de desempeño, se visualizan en gráficos indicadores en este caso los MTBF, MTTR y disponibilidad. Ver Fig. 3.91

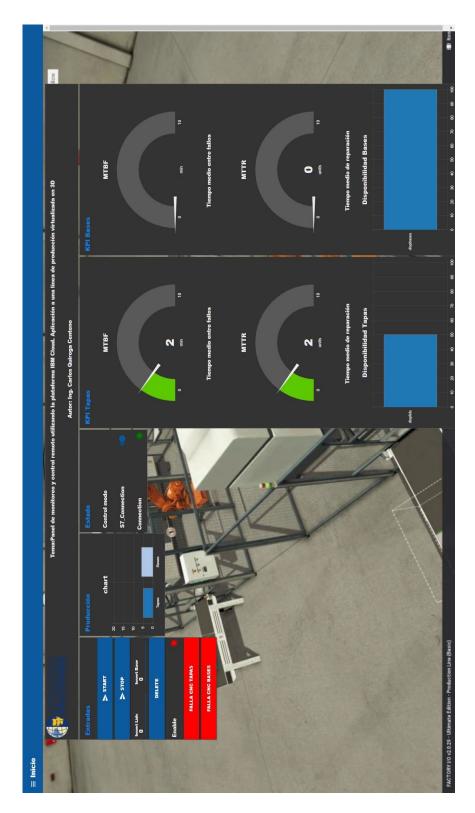


Figura 3.91: Panel de Control Node-Red

## 3.3.8. IBM Cloud

En esta ocasión, se utiliza IBM Cloud. La razón de esto es que IBM Cloud tiene Node-RED Starter Kit. Se trata de una especie de boilerplate de software que incluye los servicios necesarios para Node-RED en la nube, como una base de datos, herramientas CI/CD, y más.

Ahora, se ejecutara Node-RED en IBM Cloud siguiendo estos pasos:

1. Crear cuenta en ibm Cloud como se visualiza en Fig.3.92

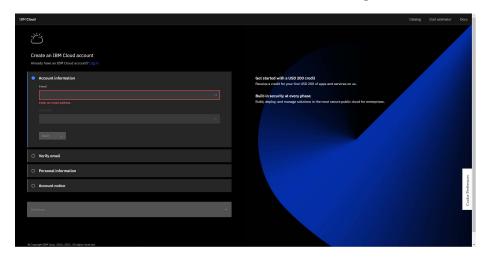


Figura 3.92: Configuración de Cuenta

- 2. Una vez verificado el e-mail, se procede a llenar las casillas con nuestra información personal, en este caso nuestro nombre, apellido y país. Ver fig 3.93
- 3. Se deberá aceptar las condiciones de servicio. Ver fig. 3.94
- 4. Al hacer clic en continuar se despliega una nueva ventana. Ver Fig 3.95
- Nuevamente se hace clic en continuar y se abre la ventana de verificación de identidad por tarjeta de crédito, sin la ejecución de este paso nuestra cuenta no se creara. Ver Fig 3.96
- 6. Se completa la información con los datos personales y dirección de domicilio para que se realice la facturación por el servicio ver Fig. 3.97.
- 7. Realizados los pasos anteriores al final se abre la ventana del menú principal en IBM cloud. Ver Fig 3.98

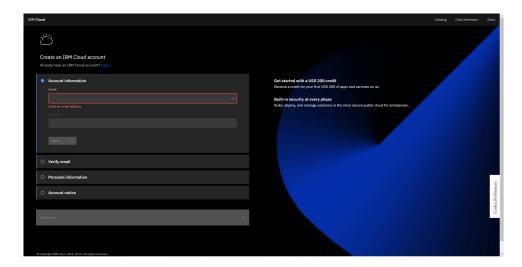


Figura 3.93: Datos Personales

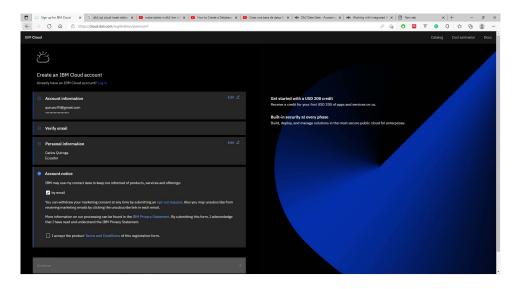


Figura 3.94: Condiciones del Servicio



Figura 3.95: Revisión de privacidad de la cuenta

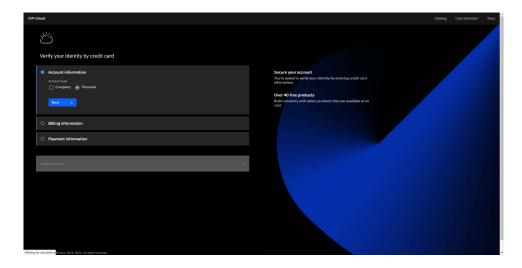


Figura 3.96: Verificación por tarjeta de crédito



Figura 3.97: Información para facturar

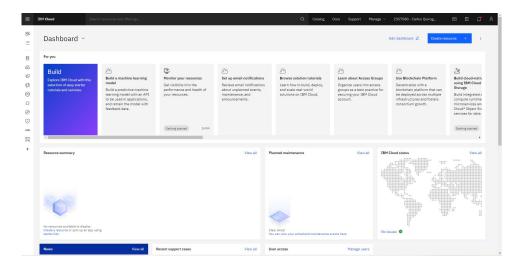


Figura 3.98: Menú Principal IBM Cloud

Ahora se ejecutan los siguientes pasos para instalar el editor de flujos en nuestra cuenta de ibm cloud:

- 1. Después de iniciar sesión en IBM Cloud, se verá su propio panel de control en la pantalla, si es la primera vez que utiliza IBM Cloud, no se mostrará ningún recurso en el panel de control. Ver Fig 3.98
- 2. Se crea Node-RED como un servicio en esta nube, se da clic en desarrollo de aplicaciones en el menú el menú de la parte superior izquierda y luego clic en el botón obtener un kit de inicio, esto permite crear un nuevo servicio de aplicación. Ver Fig. 3.99
- 3. Se puede encontrar el instalador de Node-RED escribiéndolo en el cuadro de texto de búsqueda, una vez que se lo encuentra dar clic en el panel del mismo nombre. Ver Fig.3.100
- 4. Después de hacer clic en el panel Node-RED, se deben configurar algunos elementos, como se observa en la Fig. 3.101
- 5. Cuando todo esté listo, dar clic en el botón desplegar aplicación en el centro de la pantalla para habilitarla. Ver Fig. 3.102
- 6. Después de hacer clic en el botón desplegar aplicación, se muestra la ventana de configuración de la aplicación. Ver Fig. 3.103
- 7. La ventana pedirá que se cree una clave API de IBM Cloud, no hay que preocuparse por esto la clave se generará automáticamente, se da clic en el botón nuevo para abrir una ventana emergente, luego clic en el botón OK en la ventana emergente, una vez hecho esto, se generará una IBM Cloud API Key. Ver Fig 3.104
- 8. En esta ocasión, se utiliza IBM Cloud con una cuenta Lite, por lo que sólo se dispone 256 MB de memoria disponible para todos nuestros servicios en IBM Cloud. Por lo tanto, si se sobrepasa de 256 MB para el servicio Node.js de Cloud Foundry, no se podrá utilizar más memoria para otros servicios. Pero Node-RED necesita 256 MB para ejecutarse en IBM Cloud, así que utiliza 256 MB, la instancia ya tiene asignados 256 MB por defecto, así que se hace clic en el botón Next sin cambiar ningún parámetro. Ver Fig 3.105
- 9. Ahora se muestra la ventana de inicio para entrar a la aplicación de node Red y terminar con la instalación de la misma. Ver Fig. 3.106

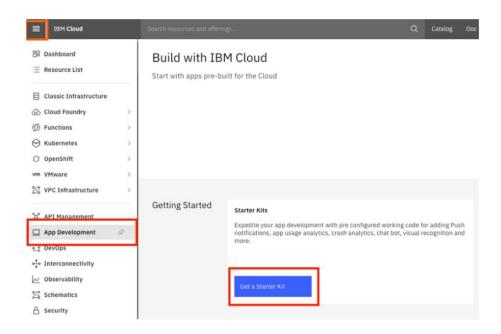


Figura 3.99: Kit started

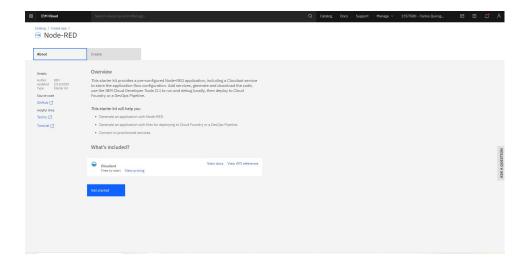


Figura 3.100: Empezar instalación Node Red en la Nube

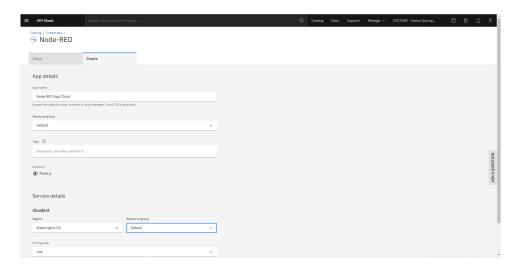


Figura 3.101: Configuraciones para instalar Node-Red

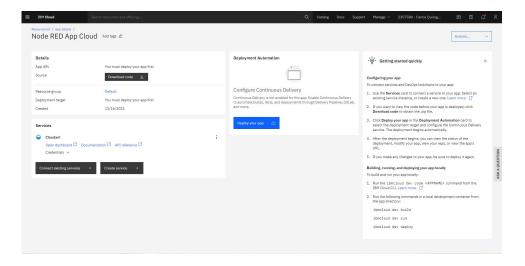


Figura 3.102: Desplegar aplicación

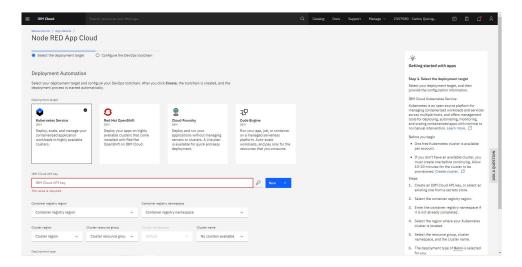


Figura 3.103: Configuración de aplicación

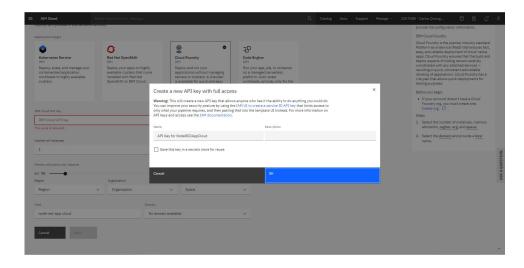


Figura 3.104: Creando API KEY

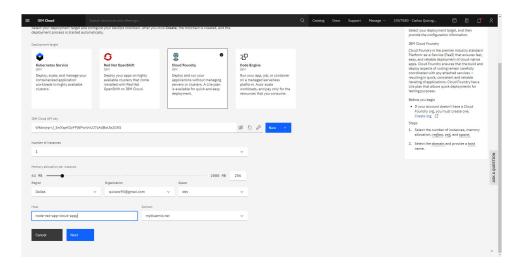


Figura 3.105: Finalizando creación instancia Node.Js

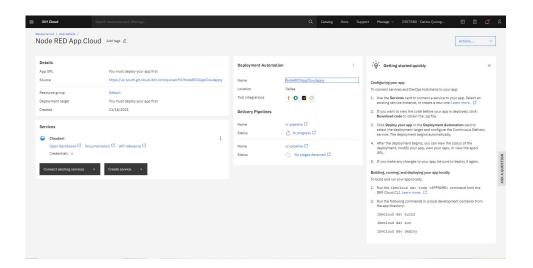


Figura 3.106: Panel de Instancia para instalar Node RED

Una vez instalados los elementos necesarios en IBM Cloud, se procede a configurar Node RED. Ver Fig $3.107\,$ 

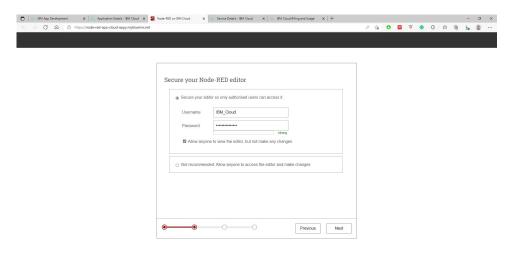


Figura 3.107: Usuario y clave puesta de servicio en Node-Red

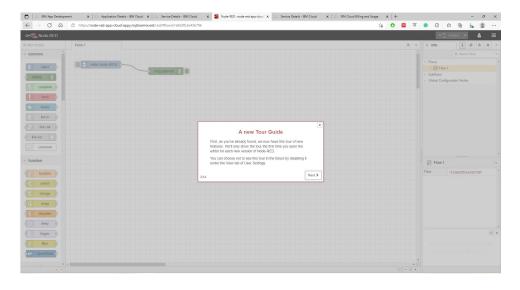


Figura 3.108: Node Red instalado con Link de IBM CLOUD

Acorde a la Fig. 3.108, se puede ejecutar NODE-RED dentro de la plataforma en la Nube, además de realizar nuestros programas y diferentes tipos de dashboard para verificar remotamente el estado de nuestro proceso. Ver Fig. 3.109

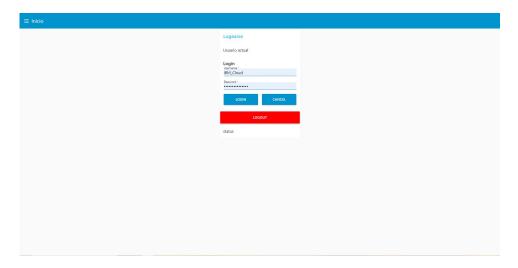


Figura 3.109: Dashboard en IBM CLOUD

#### 3.3.9. Creación de dispositivo IoT en IBM Cloud

IBM Watson IoT Platform es un servicio gestionado alojado en la nube diseñado para simplificar el proceso de obtener valor de los dispositivos IoT.

IBM Watson IoT Platform ayuda a iniciar rápidamente proyectos del IoT, es un servicio alojado en la nube totalmente administrado y diseñado para simplificar la extracción de valor de los dispositivos IoT. Proporciona registro, conectividad, control, visualización y almacenamiento de datos rápidos de dispositivos habilitados para IoT. Primero se direccionará a la plataforma internet of things de IBM, actualmente ya se tiene un usuario y contraseña configurado previamente, para esta ocasión simplemente se accederá con esas credenciales a esta nueva plataforma. Ver Fig 3.110

En la Fig 3.111 se observan todos los dispositivos creados para comunicar las variables del NODE-RED local con el instalado en la nube de IBM Cloud, cabe recalcar que se utiliza tipos de datos JSON para la lectura y escritura de variables utilizando el nodo Watson Iot.

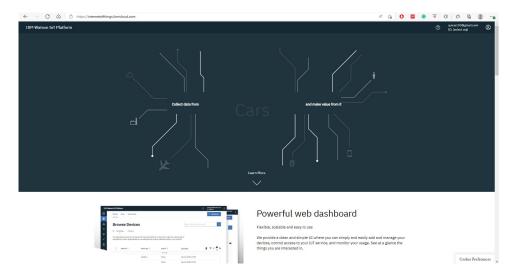


Figura 3.110: Acceso a IBM Watson

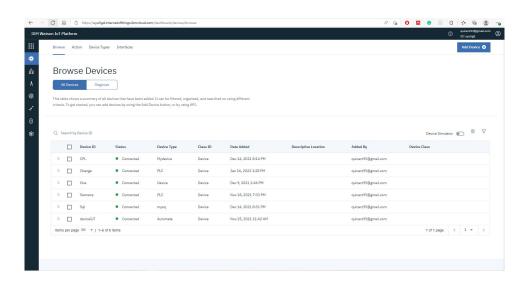


Figura 3.111: Dispositivos creados en watson IoT

## Nodo Watson Iot

El nodo puede conectarse como dispositivo o como puerta de enlace, en modo registrado o utilizando el servicio Quickstart, envía eventos del dispositivo a la plataforma IBM Watson Internet of Things. Ver Fig. 3.112

Cuando se conecta utilizando el servicio Quickstart, la conexión utilizará un tipo de dispositivo llamado node-red-wiotp y un id de dispositivo generado aleatoriamente, que puede configurarse en el nodo. Los eventos del nodo se pueden ver en el panel de control de Quickstart. El formato del evento es por defecto json, pero puede establecerse a otro valor o, si se deja en blanco, puede establecerse mediante la propiedad msg.format. Ver Fig 3.113

Cuando se conecta como un Gateway, el tipo y el id del dispositivo del que se está enviando el evento puede ser configurado en el nodo o dejarse en blanco, puede ser establecido por las propiedades msg.deviceType y msg.deviceId. Si no se proporcionan estas propiedades, ni en el nodo ni en el mensaje, se utilizará el tipo y el id del propio Gateway. Ver Fig. 3.115



Figura 3.112: Nodo Watson

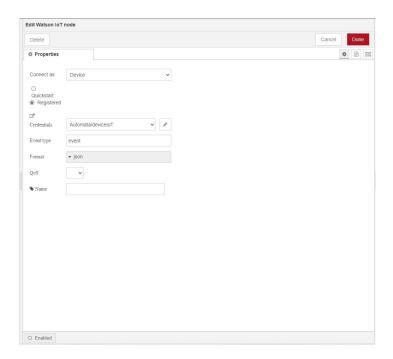


Figura 3.113: Propiedades del Nodo Watson

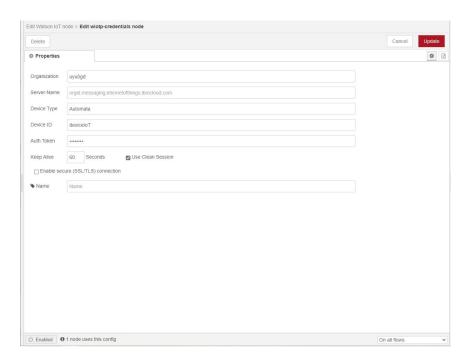


Figura 3.114: Credenciales del Nodo Watson

## Nodo IBM IoT

Nodo de entrada que puede utilizarse con Watson IoT Platform para recibir eventos enviados desde dispositivos, recibir comandos enviados a dispositivos o recibir actualizaciones de estado relativas a dispositivos o aplicaciones. Produce un objeto llamado msg y establece que msg.payload sea una cadena que contenga el contenido del mensaje entrante. Ver Fig 3.115



Figura 3.115: Nodo IBM IoT In



Figura 3.116: Propiedades Nodo IBM In

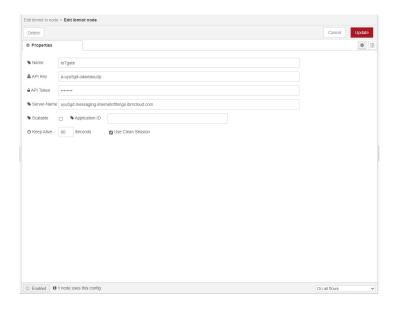


Figura 3.117: Editar API Key Nodo IBM In

#### 3.3.10. Base de Datos para la producción

En programación, el uso de algún tipo de DBMS es casi inevitable. Tarde o temprano, cada programa que se imagina tiene que almacenar datos en algún lugar, al menos una lista de usuarios autorizados, sus permisos y atributos. MySQL es actualmente el sistema de gestión de bases de datos relacionales más utilizado porque es de código abierto, MySQL es un sistema de gestión de bases de datos de doble licencia. Por un lado es de código abierto pero por otro lado tiene una versión comercial gestionada por Oracle Corporation.

En este apartado se adjuntan las tablas creadas en el software MySql para guardar los datos durante la producción, se guardan datos como las bases y tapas que se van produciendo a lo largo de pruebas durante la simulación en Factory I/O, además se crean tablas para el manejo de usuarios y contraseñas en la nube de IBM, también se guardan los datos provenientes de los indicadores clave de desempeño en otra tabla. Ver Fig 3.118, 3.119, 3.120, 3.121

Al momento de crear todas las tablas se debe tener en cuenta la creación de un TimeStamp para registrar la hora y día en que ocurrió algún evento ya sea en este caso el comienzo de la producción o el registro de los indicadores claves de desempeño para un análisis costo beneficio de la maquinaria instalada en la empresa

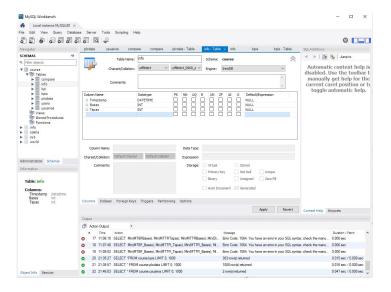


Figura 3.118: Declaración de Tabla en Workbench

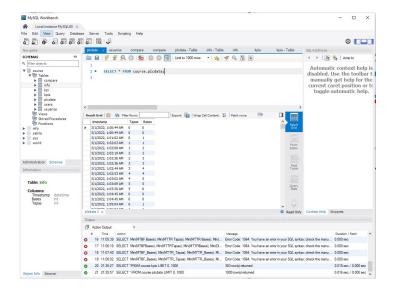


Figura 3.119: Tabla para bases y Tapas

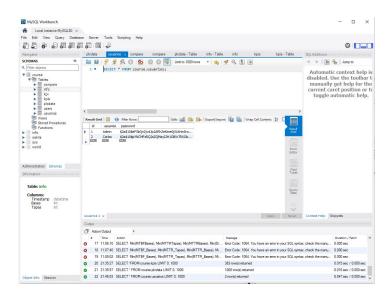


Figura 3.120: Tabla para usuarios y contraseñas

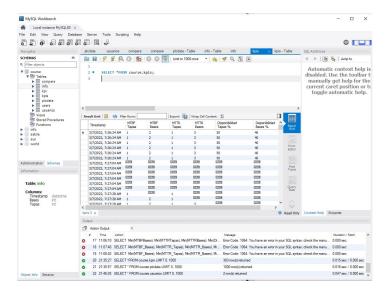


Figura 3.121: Tabla para KPI

# Capítulo 4

# Resultados

En este capítulo se muestran resultados obtenidos a partir de la simulación con los diferentes programas utilizados en el desarrollo del proyecto, se muestran los datos obtenidos desde MySql en gráficas y los cálculos realizados para los indicadores clave de desempeño KPI.

CAP 4.RESULTADOS 108

## 4.1. Resultados obtenidos a partir de la simulación

Al inicio de la etapa de desarrollo en el proyecto, se tuvieron muchos problemas al no contar con las librerías necesarias en NODE-RED, primero se trato de utilizar el protocolo OPC-UA porque tiene la ventaja de no requerir certificados o licencias adicionales para su funcionamiento, aunque el software TIA Portal podía realizar la comunicación con otros software, no fue el caso con NODE-RED se tenía un grave problema nunca se enlazaba, por muchos intentos que se realizaran e incluso se actualizo la versión del software PLC-SIM Advanced, pese a tantos intentos el problema persistía hasta que se probo si NODE-RED podría enlazarse con otro software y allí se determino que el problema era la versión de NODE-RED instalada, durante esas fechas hubo una actualización a la versión 2.0 del software antes mencionado y muchas librerías con nodos en OPC-UA no funcionaban correctamente.

Una vez detectado ese problema se procedió a utilizar el nodo S7-COMM y se configuro variable por variable para enlazarla con TIA PORTAL y obtener los datos para visualizar en NODE-RED y en la base de datos en MySql, se declaran data block con las variables a utilizar durante la simulación, ver fig 4.1.

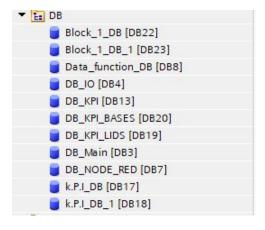


Figura 4.1: Data Blocks

Utilizando bloques de función se desarrollo un programa para guardar el tiempo de las fallas, dentro del mismo se utilizan todas las herramientas conocidas dentro de Tia Portal tales como; comparadores, temporizadores, bloques Move para transferencia de datos, cabe recalcar que este bloque se encarga de guardar los tiempos de falla, luego estos datos serán utilizados para el calculo de los MTTR y MTBF. Ver Fig 4.2, 4.3, 4.4

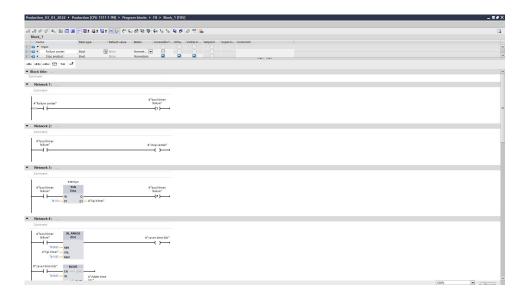


Figura 4.2: Programa para obtener tiempos segmentos 1-4

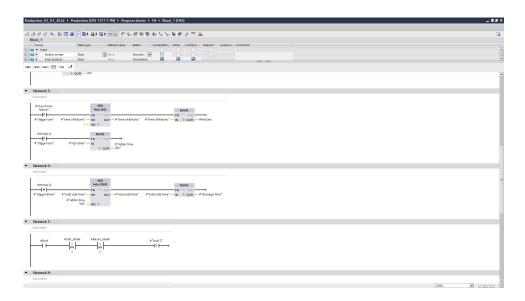


Figura 4.3: Programa para obtener tiempos segmentos 5-7

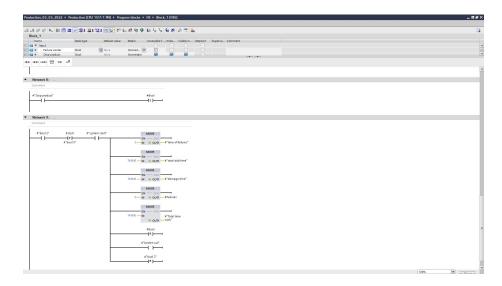


Figura 4.4: Programa para obtener tiempos segmentos 8-9

Dentro de otro bloque se utilizan los datos de tiempo proporcionados para calcular los KPI por cada linea de producción, TIA Portal cuenta con el apartado operaciones matemáticas con ello simplemente se mueven los resultados hacia NODE-RED para visualizarlos en gráficos.

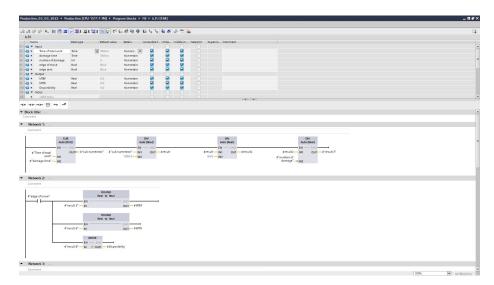


Figura 4.5: Cálculo de MTBF

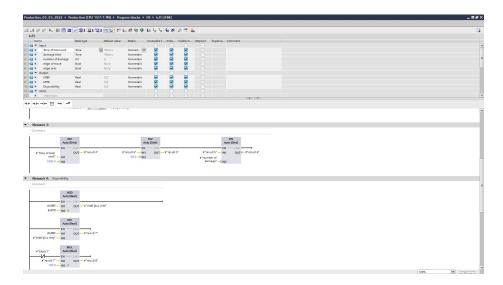


Figura 4.6: Cálculo de MTTR

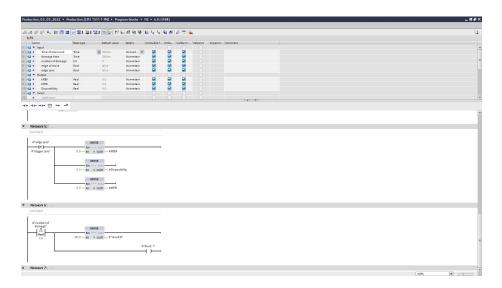


Figura 4.7: Segmentos de programación Kpi

CAP 4.RESULTADOS 112

En la simulación del proyecto, se pide ingresar el número de bases y tapas a manufacturar, para ello se ingresa el valor de 200 unidades en materia prima por cada linea de producción, como muestra la Fig. 4.8 y 4.9 en donde se observa el proceso ya culminado con todos los indicadores mencionados anteriormente.

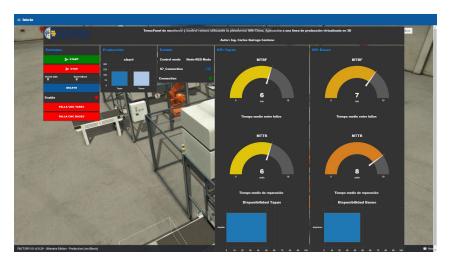


Figura 4.8: Dashboard Prueba 1

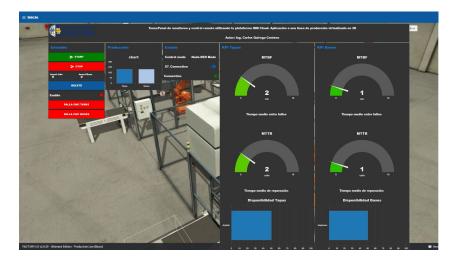


Figura 4.9: Dashboard Prueba 2

Cap 4.Resultados 113

En la Fig.4.10 se observa la interpolación gráfica proveniente de MySql, en la gráfica de histórico de bases se observan que los valores difieren mucho con respecto a la gráfica de histórico de tapas, esto se debe a que al iniciar pruebas se tuvieron problemas en el almacenamiento de datos, el contador de bases no se bloqueaba y seguía funcionando a pesar de terminar el proceso, esto se debe a que los intervalos de comunicación entre varios software diferían en segundos y mili-segundos respectivamente y esto lo provocaba TIA PORTAL por errores en la programación al no considerar bloquear la transmisión de datos al momento de culminar todo el proceso.

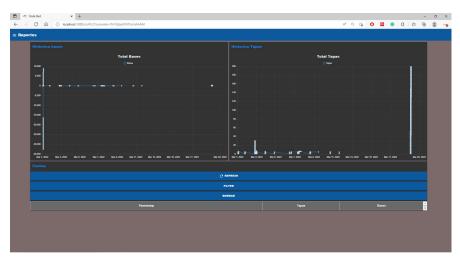


Figura 4.10: Comparación Históricos Base de Datos

Adicional se tomo una captura del rendimiento en la PC al ejecutar varios programas, actualmente al utilizarlos en simultaneo la CPU trabaja al cien por ciento de su capacidad y la memoria a un cuarenta y seis por ciento, para futuros trabajos se deberá trabajar con sistemas embebidos y con una planta a escala para no exceder los limites de rendimiento en el computador.

Si el uso de la CPU está en torno al cien por ciento, esto significa que el equipo está intentando hacer más trabajo del que puede. Normalmente esto es correcto, pero significa que los programas pueden ir un poco más lentos. Los equipos tienen a usar el cien por ciento de la CPU cuando hacen tareas que requieren mucho procesamiento, como pueden ser los juegos, y eso aplica en el caso de Factory I/O debido a que los gráficos son en alta definición.

Si el procesador está al cien por ciento durante mucho rato, esto puede hacer que el equipo vaya sorprendentemente lento, en este caso se observa en la Fig 4.11 los programas que están consumiendo mucho procesamiento en la CPU.

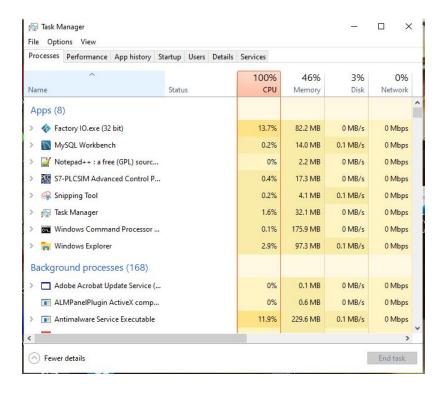


Figura 4.11: Administrador de Tareas

# Conclusiones y Recomendaciones

#### Conclusiones.

- Se analizó el concepto del Internet de las cosas mediante la revisión de Artículos académicos que se encuentran en IEEE, Scopus y Google Schoolar respectivamente para comprender más a fondo del tema a desarrollar.
- 2. Se definieron los requisitos en el diseño para la funcionalidad del editor de flujos NODE-RED mediante la lectura de libros que muestran el proceso de instalación del mismo, además de los requisitos que debe poseer el computador para su correcto funcionamiento, el estudio de los tipos de nodos que existen junto a sus librerías contribuyo en gran parte a desarrollar el dashboard para el control del proceso en Factory I/O.
- 3. Se implemento el uso del dashboard configurado desde el editor de flujos NODE-RED, mediante el Nodo S7-Comm, se utiliza el software PLC-SIM Advanced V3.0 para simular un PLC S7-1500 en red con Factory I/O mientras Tia Portal V16 realiza la lectura de entradas, salidas y marcas del sistema para enviar los estados de las mismas hacia NODE-RED y visualizarlos en el panel de control.
- 4. Se implementa el desarrollo de un panel de monitoreo en la nube de IBM, esto se logra mediante el uso de NODE-RED instalado en la nube, se aperturan puertos para el libre acceso de datos, debido a que todo se realiza en software de simulación no representa mayor riesgo en las actividades del proceso, se utilizan dos brokers para el envío y recepción de datos en la Nubes, uno es Watson IoT de IBM y el otro broker es Cloud MQTT ambos servicios son por contrato mensual.
- 5. Se valida la propuesta utilizando los softwares de simulación antes mencionados: TIA Portal, PLC SIM Advanced, Factory I/O y NODE-RED, se realiza la medición de los KPI por medio del tiempo que toma la manufacturación por medio de las CNC en cada línea de Producción, se simula falla de la máquina para obtener un valor medible debido a las limitaciones del software Factory I/O.

#### Recomendaciones.

- 1. Se recomienda en futuros trabajos, utilizar una pc embebida para el uso de node-red debido a que TIA Portal consume muchos recursos de memoria para ejecutar varios Softwares a la vez.
- 2. Para realizar una aplicación dentro de un proceso real, se debe trabajar en conjunto con el departamento de IT, para desbloquear los puertos correspondientes a NODE-RED si se requiere conexión externa desde cualquier parte con todos las seguridades que se requieran.
- 3. Tener en cuenta trabajar con la versión más actualizada en NODE-RED, debido a que se tuvieron muchos inconvenientes al trabajar con una versión del 2020, no se instalaban todas las librerias y esto ocasiona problemas en la conexión con otros dispositivos.
- 4. Para la base de datos en MYSqL se recomienda tener un computador dedicado para almacenar todos los datos de la producción correspondientes en las diferentes tablas ubicadas en el DATA BLOCK.

GLOSARIO 3

- API Interfaz de programación para el desarrollo de aplicaciones
- **DCUI** Direct Console User Interface (Interfaz de usuario de consola directa).
- AES Advanced Encryption Standard (Encriptación avanzada Estándar)
- DHCP Protocolo de configuración huésped dinámico
- **DES** Estándar de Cifrado de Datos
- HTTP Protocolo de Transferencia para Hipertexto.
- IT Información Tecnológica.
- OT Tecnología de Operación
- **IED** Dispositivo electrónico inteligente.
- **DNS** Sistema de nombres para dominios.
- IP Protocolo de Internet
- IoT Internet de las cosas
- Token Componente léxico es una cadena de caracteres.
- VLAN Red de área local virtual.
- Wifi Wireless Fidelity (Fidelidad sin cables o inalámbrica).
- TCP Protocolo de control de transmisión
- WLAN Wireless local area network (Sistematización de comunicación inalámbrico).
- JSON Formato de texto en JavaScript para el intercambio de datos.
- MQTT Transporte de telemetría de Message Queue Server
- KPI Indicador clave de rendimiento
- MySQL Lenguaje de Consulta Estructurado

## Bibliografía

- M. M. Ahmad and N. Dhafr. Establishing and improving manufacturing performance measures. *Robotics and Computer-Integrated Manufacturing*, 18(3-4):171–176, 2002.
- G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz. A survey on 5g networks for the internet of things: Communication technologies and challenges. *IEEE access*, 6:3619–3647, 2017.
- P. Alavian, Y. Eun, K. Liu, S. M. Meerkov, and L. Zhang. The  $(\alpha, \beta)$ -precise estimates of mtbf and mttr: Definitions, calculations, and induced effect on machine efficiency evaluation. *IFAC-PapersOnLine*, 52(13):1004–1009, 2019.
- B. Ali and A. I. Awad. Cyber and physical security vulnerability assessment for iot-based smart homes. *sensors*, 18(3):817, 2018.
- G. M. S. Association et al. The road to 5g: Drivers, applications, requirements and technical development. A GSA Executive Report from Ericsson, Huawei and Qualcomm, 2015.
- A. Barrera. Json:¿qué es y para qué sirve? page 1, 2020. URL https://www.nextu.com/blog/que-es-json/.
- A. M. Bezeira, M. Villapol, E. Scalise, C. Naranjo, and F. Alvárez. Analysis of the dynamic service flow management transactions protocol for mac ieee 802.16. an aproach. In 2017 XLIII Latin American Computer Conference (CLEI), pages 1–10, 2017. doi: 10.1109/CLEI.2017.8226455.
- A. R. Biswas and R. Giaffreda. Iot and cloud convergence: Opportunities and challenges. In 2014 IEEE World Forum on Internet of Things (WF-IoT), pages 375–376, 2014. doi: 10.1109/WF-IoT.2014.6803194.
- L. C. Borja. Adquisición de datos desde plataforma iot2040. protocolos modbus/tcp y opc ua, Nov 2019. URL https://addi.ehu.es/handle/10810/36491?show=full.

BIBLIOGRAFÍA 5

M. Burhan, R. Rehman, B. Khan, and B. Kim. Iot elements, layered architectures and security issues: A comprehensive survey. *Sensors (Basel, Switzerland)*, 18(9), 2018.

- V. Chayapathy, G. S. Anitha, and B. Sharath. Iot based home automation by using personal assistant. In 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), pages 385–389, 2017. doi: 10.1109/SmartTechCon.2017.8358401.
- P. Gokhale, O. Bhat, and S. Bhat. Introduction to iot. *International Advanced Research Journal in Science, Engineering and Technology*, 5(1):41–44, 2018.
- M. Grabovica, S. Popić, D. Pezer, and V. Knežević. Provided security measures of enabling technologies in internet of things (iot): A survey. In 2016 Zooming Innovation in Consumer Electronics International Conference (ZINC), pages 28–31. IEEE, 2016.
- A. K. Gupta and R. Johari. Iot based electrical device surveillance and control system. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), pages 1–5, 2019. doi: 10.1109/IoT-SIU. 2019.8777342.
- T. Hagino. Node-red book. page 326, 2021.
- Infraspeak. Mantenimiento. page 1, 2020. URL https://infraspeak.com/es/gosario-mtbf/.
- R. Khan, S. U. Khan, R. Zaheer, and S. Khan. Future internet: the internet of things architecture, possible applications and key challenges. In 2012 10th international conference on frontiers of information technology, pages 257–260. IEEE, 2012.
- B. Khoo. Rfid as an enabler of the internet of things: Issues of security and privacy. In 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, pages 709–712. IEEE, 2011.
- R. K. Kodali and A. Anjum. Iot based home automation using node-red. In 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), pages 386–390, 2018. doi: 10.1109/ICGCIoT.2018. 8753085.
- M. Lekić and G. Gardašević. Iot sensor integration to node-red platform. In 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), pages 1–5. IEEE, 2018.

BIBLIOGRAFÍA 6

S. Li, L. Da Xu, and S. Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, 2015.

- S. Madakam, V. Lake, V. Lake, V. Lake, et al. Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3(05):164, 2015.
- M. U. Mehmood, W. Ali, A. Ulasyar, H. S. Zad, A. Khattak, and K. Imran. A low cost internet of things (lciot) based system for monitoring and control of ups system using node-red, cloudmqtt and ibm bluemix. In 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pages 1–5, 2019. doi: 10.1109/ICECCE47252.2019. 8940686.
- J. Mejía Cruz. Historia de la industria 4.0. pages 1-6, 2018. URL https://www.logicbus.com.mx/pdf/articulos/Historia-industria4.0.pdf.
- D. Minoli, K. Sohraby, and B. Occhiogrosso. Iot security (iotsec) mechanisms for e-health and ambient assisted living applications. In 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), pages 13–18. IEEE, 2017.
- M. M. Morshed, A. Atkins, and H. Yu. Privacy and security protection of rfid data in e-passport. In 2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings, pages 1–7. IEEE, 2011.
- Μ. Nunez. What is 5gand how will it make better? 1, 2018. URL https://gizmodo.com/ page what-is-5g-and-how-will-it-make-my-life-better-1760847799.
- J. Padgette, K. Scarfone, and L. Chen. Guide to Bluetooth Security:. US Department of Commerce, National Institute of Standards and Technology, 2011.
- J. Pérez-Bedmar. Things to consider when choosing an iot platform. page 1, 2019. URL https://barbaraiot.com/blog/things-to-consider-when-choosing-an-iot-platform/.
- Z.-H. Qian and Y.-j. WANG. Iot technology and application. *ACTA ELECTONICA SINICA*, 40(5):1023, 2012.
- W. Schneeweiss. Computing failure frequency, mtbf & mttr via mixed products of availabilities and unavailabilities. *IEEE Transactions on Reliability*, 30 (4):362–363, 1981.

BIBLIOGRAFÍA 7

Siemens. Simatic s7-1500, analog input module. page 1, 2021a. URL https://mall.industry.siemens.com/mall/es/WW/Catalog/Product/6ES7531-7KF00-0ABO.

- Siemens. Simatic s7-1500, digital input module. page 1, 2021b. URL https://mall.industry.siemens.com/mall/es/WW/Catalog/Product/6ES7521-7EH00-0AB0.
- Siemens. Simatic s7-1500, digital output module. page 1, 2021c. URL https://mall.industry.siemens.com/mall/es/WW/Catalog/Product/6ES7522-5HH00-0AB0.
- H. Suo, J. Wan, C. Zou, and J. Liu. Security in the internet of things: a review. In 2012 international conference on computer science and electronics engineering, volume 3, pages 648–651. IEEE, 2012.
- techbf. Learn what is database, types of database, dbms. page 1, 2019. URL https://techtfq.com/blog/learn-what-is-database-types-of-database-dbms-omQVL.
- O. Vermesan, P. Friess, et al. *Internet of things-from research and innovation to market deployment*, volume 29. River publishers Aalborg, 2014.
- S. Wang, Y. Hou, F. Gao, and X. Ji. A novel iot access architecture for vehicle monitoring system. In 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pages 639–642. IEEE, 2016.
- A. Xheladini, S. D. Saygili, and F. Dikbiyik. An iot-based smart exam application. In *IEEE EUROCON 2017-17th International Conference on Smart Technologies*, pages 513–518. IEEE, 2017.
- M. Zairi. Benchmarking: the best tool for measuring competitiveness. Benchmarking for Quality Management & Technology, 1994.