



UNIVERSIDAD  
CATÓLICA DE CUENCA  
COMUNIDAD EDUCATIVA AL SERVICIO DEL PUEBLO

UNIVERSIDAD CATÓLICA DE CUENCA SEDE AZOGUES

Unidad Académica de Ingeniería, Industria y Construcción

Facultad de Ingeniería Electrónica

---

“Adaptación de la tecnología estándar de comunicación en el campo del control y supervisión de procesos industriales (OPC), en un prototipo de sistema SCADA basado en la plataforma Ignition y controladores lógicos programables”

Tesis previa a la obtención del título de  
Ingeniero Electrónico

Autor:

Luis Vinicio Ordoñez Lliguicota

Director:

Ing. Trajano Javier González Redrován. MSc

Azogues – Ecuador

2018



## DECLARATORIA DE RESPONSABILIDAD

Yo, Luis Vinicio Ordoñez Lliguicota, autor de la tesis “Adaptación de la tecnología estándar de comunicación en el campo del control y supervisión de procesos industriales (OPC), en un prototipo de sistema SCADA basado en la plataforma Ignition y controladores lógicos programables”, certifico que todas las ideas, opiniones, análisis, desarrollo técnico – lógico y contenidos mostrados en la presente investigación son de mi exclusiva responsabilidad, a su vez que reconozco y acepto el derecho correspondiente de patrimonio intelectual a la Universidad Católica de Cuenca.

Azogues, Marzo del 2018

*Luis Vinicio Ordoñez Lliguicota*

CI: 0302893169



## CERTIFICO

Que el presente trabajo de tesis: “Adaptación de la tecnología estándar de comunicación en el campo del control y supervisión de procesos industriales (OPC), en un prototipo de sistema SCADA basado en la plataforma Ignition y controladores lógicos programables”, fue dirigido por mi persona.

Azogues, Marzo del 2018

Ing. Trajano Javier González Redrován. MSc

Director de Tesis



## Agradecimiento

---

En primer lugar agradezco a Dios por haberme guiado en mi vida, ya que con sus bendiciones me ha protegido y brindado la fuerza y sabiduría para seguir adelante.

En segundo lugar, agradezco a la Universidad Católica de Cuenca, por darme la oportunidad de prepararme profesionalmente.

A mi tutor de tesis, Ing. Javier González quien, con su paciencia y consejo, supo guiarme en este proceso, dándome confianza y motivación. Gracias por toda la colaboración prestada durante el desarrollo de la tesis.

Al Ing. Juan Carlos Ortega Director de la Facultad de Ingeniería Electrónica, quien cumple sus labores de la mejor manera, siempre dispuesto a brindar su ayuda a los estudiantes, de igual manera a todos los profesores de la facultad que siempre estuvieron pendientes, gracias por todas las experiencias y enseñanzas impartidas.

Finalmente agradezco a mis amigos y compañeros de clase, en especial a Homero Landin, Edwin Morquecho y Gustavo Morocho, por su apoyo y colaboración en la terminación de mi tesis y un grato agradecimiento a la señora Anita Tamay, por permitirme usar las instalaciones del laboratorio de Electrónica.



## Dedicatoria

---

Dedico esta tesis a Dios por haberme dado la vida y haberme permitido llegar hasta este momento tan importante de mi formación profesional. Siempre estás conmigo padre mío.

A mis padres Juan Ordoñez y María Lliguicota, por ser los pilares más importantes en mi vida, demostrándome siempre su amor incondicional y comprensión, que con sus sabios consejos me impulsaron a seguir adelante. A mis hermanas Silvia, Vilma y Lucia, a mi hermano Juan, a mis sobrinos Andrés Ordoñez, Aron Vega y a mi sobrina Belén Vega, quienes con su apoyo y compañía me ayudaron a superar los momentos más difíciles de la tesis, demostrándome el verdadero significado de la familia.



---

## Resumen

---

La implementación de una arquitectura de comunicación OPC, permite integrar diversas tecnologías independientes del fabricante, brindando una mejor gestión en el control y monitoreo, de un proceso industrial. La existencia de sistemas propietarios dependientes de hardware y software especializado, precisan de inversiones costosas y de problemas de interoperabilidad ante una deficiente adaptación. En esto, el protocolo OPC presenta una solución firme, confiable y flexible, eliminando estas imposiciones, dotaciones que la *OPC Foundation* compacta en sus versiones de OPC clásico y OPC UA. El presente trabajo muestra, un prototipo de sistema OPC SCADA diseñado en la plataforma Ignition. El cual controla y monitorea desde una red local el micro clima de un modelo a escala de invernadero. Mediante técnicas de control básico se regulan las condiciones climáticas de humedad, temperatura y cantidad de agua. Para ello emplea servidores OPC con diferentes tipos de tecnológicas (PLC, Arduino y Raspberry Pi) que gestionan varios procesos y ejecutados desde distintas plataformas. En conclusión, el prototipo propuesto constituye un tema interesante para explorar nuevas alternativas, evaluando dispositivos embebidos de manera que puedan viabilizarse cómo herramientas provechosas y sostenibles en la industria. Permitiendo la adecuación de esta tecnología con un costo de inversión muy bajo.

**Palabras clave:** Sistema OPC SCADA, dispositivos embebidos, sistemas propietarios, condiciones climáticas, invernadero.



## Abstract

---

The implementation of an OPC communication architecture, allows to integrate diverse independent technologies of the manufacturer, offering a better management in the control and monitoring of an industrial process. The existence of proprietary systems dependent of hardware and skilled software, require of costly investments and of problems of interoperability in front of a deficient adaptation. In this, the protocol OPC presents a firm solution, reliable and flexible, deleting these impositions endowments that the *OPC Foundation* compact in his versions of OPC classical and OPC UA. This graduation work shows a prototype of system OPC SCADA designed in the platform Ignition, which controls and monitor from a local network the micro climate of a greenhouse scale model. By means of basic control techniques, it is possible to regulate the climatic conditions of humidity, temperature and quantity of water. To do it, it is employed OPC servers with different types of technologies (PLC, Arduino and Raspberry Pi) that manage several processes and that are executed from different platforms. In conclusion, the prototype proposed, constitutes an interesting subject to explore new alternatives, by evaluating embed devices so that they can be used as profitable and sustainable tools in the industry, allowing the adequacy of this technology with a very low cost of investment.

**Keywords:** System OPC SCADA, embed devices, proprietary systems, climatic conditions, greenhouse.



## Índice general

Resumen .....	6
Abstract .....	7
Índice general .....	8
Índice de figuras .....	11
Índice de tablas .....	15
<b>CAPÍTULO 1 .....</b>	<b>16</b>
Introducción.....	16
1.1. ANTECEDENTES .....	17
1.2. SITUACIÓN PROBLEMÁTICA .....	19
1.3. LÍNEA DE INVESTIGACIÓN .....	20
1.3.1 <i>Ámbito</i> .....	20
1.3.2 <i>Objeto de estudio</i> .....	20
1.3.3 <i>Campo de acción</i> .....	20
1.4. OBJETIVOS .....	21
1.4.1 <i>Objetivo General</i> .....	21
1.4.2 <i>Objetivo específico</i> .....	21
1.5. HIPÓTESIS .....	21
1.6. JUSTIFICACIÓN .....	21
<b>CAPÍTULO 2 .....</b>	<b>23</b>
MARCO TEÓRICO .....	23
2.1. CONTROLADORES LÓGICOS PROGRAMABLES (PLC) .....	23
2.1.1 <i>Lenguajes de programación</i> .....	25
2.1.2 <i>Medios de programación</i> .....	25
2.2. RASPBERRY PI .....	26
2.2.1 <i>Software</i> .....	26
2.2.2 <i>Características de hardware</i> .....	27
2.3. ARDUINO .....	27
2.3.1 <i>Hardware</i> .....	28
2.4. SISTEMA SCADA .....	29
2.4.1 <i>Arquitectura de hardware</i> .....	29
2.4.2 <i>Arquitectura del Software</i> .....	31
2.5. REDES DE COMUNICACIÓN INDUSTRIAL .....	32
2.6. PIRÁMIDE CIM (COMPUTER INTEGRATED MANUFACTURE) .....	33
2.7. TOPOLOGÍA DE REDES INDUSTRIALES .....	34
2.8. PROTOCOLOS DE COMUNICACIÓN INDUSTRIAL.....	35



2.8.1 Modelo OSI.....	36
2.8.2 Protocolo profibus (Process Fields Bus) .....	37
2.8.3 Modbus.....	38
2.8.4 Hart (Highway Addressable Remote Transducer).....	40
2.8.5 Ethernet.....	40
2.9. OPC (OLE FOR PROCESS CONTROL) .....	41
<b>CAPÍTULO 3 .....</b>	<b>47</b>
Diseño y construcción del prototipo SCADA.....	47
3.1. CONSIDERACIONES DEL MODELO .....	47
3.1.1 Requerimientos del sistema de control automático.....	48
3.2. DISEÑO DE LA INFRAESTRUCTURA DE RED .....	49
3.2.1 Comunicación de implementos del sistema con los servidores OPC.....	50
3.3. IMPLEMENTACIÓN Y CONFIGURACIÓN DE COMPONENTES DEL SISTEMA OPC .....	53
3.3.1 Implementación Arduino Shield Ethernet .....	53
3.3.1.1.Características e instalación de librería OPC Arduino .....	54
3.3.2 Instalación del sistema operativo (Raspberry Pi).....	55
3.3.2.1.Entorno Python .....	56
3.3.3 Propiedades del PLC .....	56
3.3.3.1.Acceso al área de memoria del CPU S71200.....	58
3.4. RED LAN OPC .....	59
3.5. DISEÑO Y CONFIGURACIÓN DE LAS INTERFACES OPC.....	59
3.5.1 Interfaz Arduino Ethernet – Servidor OPC DA.....	59
3.5.1.1.Establecimiento de conexión de red.....	60
3.5.1.2.Creación de Items OPC.....	60
3.6. IMPLEMENTACIÓN DEL SERVIDOR OPC DA.....	62
3.7. CONEXIÓN SERVIDOR OPC DA .....	64
3.7.1 Configuración DCOM.....	65
3.8. PROGRAMACIÓN DEL PLC .....	68
3.9. CONSTRUCCIÓN DEL SERVIDOR OPC UA RASPBERRY.....	71
3.9.1 Configuración de la LAN gobernada por el sistema OPC.....	74
3.10. DESARROLLO DE LA APLICACIÓN .....	74
3.11. DISEÑO DEL SISTEMA SCADA.....	75
3.11.1 Ignition inductive Automation.....	75
3.11.2 Desarrollo de un SCADA en la nube.....	76
3.11.2.1. Configuración de ingreso al sistema .....	76
3.11.2.2. Conexión de las unidades de control.....	77
3.11.2.3. Conexión a base de datos .....	81



3.11.2.4. Diseñador de Ignition.....	83
3.11.2.5. Cliente Ignition .....	96
3.11.2.6. Puerta de enlace Gateway .....	100
3.11.3 Problemas .....	101
3.11.3.1. Incompatibilidad de versión de software del servidor OPC DA .....	101
3.11.3.2. Problemas de visualización de variables OPC DA en el cliente COM Ignition .....	101
3.12. DISEÑO Y ELABORACIÓN DE LA ESTRUCTURA DEL INVERNADERO .....	103
<b>CAPÍTULO 4 .....</b>	<b>108</b>
Evaluación y validación del prototipo SCADA .....	108
4.2. PRUEBAS DE MONITOREO Y FUNCIONAMIENTO .....	108
4.2.1 Interface del sistema de control SCADA OPC .....	108
4.2.1.1. Pruebas de funcionamiento de la aplicación de control SCADA .....	111
4.3. PRUEBAS Y VERIFICACIÓN DE RESULTADOS DE CONEXIÓN DEL SISTEMA OPC .....	114
4.3.1 Pruebas de enlace mediante OPC .....	115
4.3.2 Prueba de enlace por transmisión de mensajería simple ICMP .....	117
4.3.3 Análisis de resultados .....	119
<b>CAPÍTULO 5 .....</b>	<b>120</b>
Conclusiones y recomendaciones .....	120
5.1. CONCLUSIONES .....	120
5.2. RECOMENDACIONES .....	122
Bibliografía.....	123
<b>ANEXOS .....</b>	<b>123</b>
ANEXO A .....	128
Requisitos del servidor OPC UA Raspberry .....	128
ANEXO B .....	129
Diagrama de clases de sistemas embebidos OPC para invernadero.....	129
ANEXO C .....	131
Presupuesto y documentacion sistema Ignition .....	131
ANEXO D .....	132
Aspectos del sistema SCADA OPC.....	132



## Índice de figuras

FIGURA 2.1 ESQUEMA DE E/S DEL PLC AUTOR ().....	23
FIGURA 2.2 ARQUITECTURA INTERNA DE UN PLC (ELECTRIN, 2016). ....	24
FIGURA 2.3 TABLA DE REGISTRO DEL MICROPROCESADOR, CUALQUIER CAMBIO DE ESTADO DEL REGISTRO SE REFLEJA EN LOS BORNES Y LA MEMORIA DEL PLC SIENDO INTUITIVO E INTERACTIVO PARA EL USUARIO (M. MORENO, 2015). ....	25
FIGURA 2.4 LENGUAJE ESCALERA (CON CONTACTORES -  - Y BOBINAS - (-)-) (MITSUBISHI, 2014). ....	25
FIGURA 2.5 DESCRIPCIÓN DE CARACTERÍSTICAS TÉCNICAS Y ELEMENTOS DE RASPBERRY PI 2 B (V_ESTINF, 2013). ....	27
FIGURA 2.6 PROCESO DE LECTURA Y ESCRITURA EN LAS E/S DE ARDUINO (AMANGANDI, 2012). ....	28
FIGURA 2.7 DISPOSICIÓN DE LOS ELEMENTOS DEL ARDUINO UNO R3 AUTOR ().....	29
FIGURA 2.8 SISTEMA SCADA BÁSICO AUTOR ().....	30
FIGURA 2.9 CLASIFICACIÓN DE LA RED INDUSTRIAL EN LA CIM (INGENIERIATIC, 2017) .....	33
FIGURA 2.10 ESTRUCTURA DE LA RED TIPO BUS (FERNANDO & ANÍBAL, 2015). ....	34
FIGURA 2.11 ESTRUCTURA DE UNA RED ANILLO (FERNANDO & ANÍBAL, 2015). ....	35
FIGURA 2.12 TOPOLOGÍA DE RED ESTRELLA (ESPINOZA, 2014). ....	35
FIGURA 2.13 NIVELES DEL MODELO OSI (ERMESH, 2016) & AUTOR (). ....	37
FIGURA 2.14 RELACIÓN MAESTRO-ESCLAVO (NATIONAL INSTRUMENTS, 2014). ....	39
FIGURA 2.15 SEÑAL HART (ESPINOZA, 2014). ....	40
FIGURA 2.16 CONEXIÓN HART EN UNA RED MULTIPUNTO (ESPINOZA, 2014). ....	40
FIGURA 2.17 SOLUCIONES CON OPC (RAÚL COBO, 2015). ....	42
FIGURA 3.1 DIAGRAMA DEL SISTEMA DE COMUNICACIÓN OPC AUTOR () .....	47
FIGURA 3.2 ARQUITECTURA OPC MULTIPLAFORMA AUTOR (). ....	48
FIGURA 3.3 TOPOLOGÍA DEL SISTEMA DE SUPERVISIÓN Y CONTROL HIBRIDO AUTOR (). ....	49
FIGURA 3.4 ESQUEMA DEL MODELO DE COMUNICACIÓN OPC UA DEL PLC CON EL SERVIDOR DE IGNITION AUTOR (). ....	50
FIGURA 3.5 DIAGRAMA DE COMUNICACIÓN OPC ENTRE ARDUINO ETHERNET Y SERVIDOR OPC DA AUTOR (). ....	51
FIGURA 3.6 ESQUEMA DE COMUNICACIÓN OPC UA DE RASPBERRY PI, CON IGNITION AUTOR (). ....	52
FIGURA 3.7 SISTEMA DE CONTROL AUTOMÁTICO, COMUNICACIÓN REMOTA CLIENTE – SERVIDOR OPC EN IGNITION AUTOR (). ....	53
FIGURA 3.8 MONTAJE ARDUINO ETHERNET AUTOR (). ....	54
FIGURA 3.9 LIBRERÍA (#INCLUDE<OPC.H>) OPC DESDE EL GESTOR DE LIBRERÍAS ARDUINO AUTOR (). ....	54
FIGURA 3.10 CONEXIÓN DE INTERNET EN NOOBS AUTOR (). ....	55
FIGURA 3.11 ELECCIÓN DE RASPBIAN AUTOR (). ....	55



FIGURA 3.12 DIRECCIONAMIENTO AL ÁREA DE MEMORIA ENTRADAS (I) EN FORMATO BYTE EN EL PLC (CPU, 2013)	58
FIGURA 3.13 DIRECCIONAMIENTO A UNA AI EN LA CPU (CPU, 2013).	59
FIGURA 3.14 DESCARGA E INSTALACIÓN DEL SERVIDOR OPC DA EN WINDOWS AUTOR ()	62
FIGURA 3.15 INSTALACIÓN DEL MÓDULO OPC CORE AUTOR ()	63
FIGURA 3.16 HABILITACIÓN DE OPCENUM AUTOR ()	63
FIGURA 3.17 ENTORNO Y CONFIGURACIÓN DE ARDUINO ETHERNET CON EL OPCSERVER AUTOR ()	64
FIGURA 3.18 EJECUCIÓN DEL SERVIDOR OPC AUTOR ()	65
FIGURA 3.19 REGISTRO DEL SERVIDOR OPC ARDUINO AUTOR ()	65
FIGURA 3.20 VALIDACIÓN DE SEGURIDAD COM PARA EL OPCSERVER AUTOR ()	66
FIGURA 3.21 HABILITACIÓN DE LA CONFIGURACIÓN DCOM PARA QUE EL “FIREWALL” ADMITA EL ENLACE AUTOR ()	67
FIGURA 3.22 CONFIGURACIÓN DCOM DEL OPCSERVER DE ARDUINO AUTOR ()	67
FIGURA 3.23 CONFIGURACIÓN DE PERMISOS COM AUTOR ()	68
FIGURA 3.24 PROGRAMACIÓN DEL PLC S 7 1200 CON TIA PORTAL AUTOR ()	68
FIGURA 3.25 SELECCIÓN DEL PLC S7 1200 AUTOR ()	69
FIGURA 3.26 CREACIÓN DE VARIABLES EN LA MEMORIA Y VARIABLES DE CONVERSIÓN AUTOR ()	69
FIGURA 3.27 PROGRAMACIÓN KOP AUTOR ()	70
FIGURA 3.28 CONEXIÓN PLC ORDENADOR PARA SUBIR EL PROGRAMA AUTOR ()	70
FIGURA 3.29 CONFIGURACIÓN DE IP ESTÁTICA DEL SERVIDOR OPCUA PARA CONECTARSE CON LA RED LOCAL AUTOR ()	71
FIGURA 3.30 EJECUCIÓN, INTERFAZ DE SERVIDOR OPC UA DESARROLLADO CON QT DE PYTHON EN RASPBERRY AUTOR ()	73
FIGURA 3.31 CONFIGURACIÓN DE RED LAN AUTOR ()	74
FIGURA 3.32 VENTANA LOGIN DE IGNITION AUTOR ()	76
FIGURA 3.33 DRIVER DE CONEXIÓN OPC – UA DEL S7-1200 EN IGNITION AUTOR ()	77
FIGURA 3.34 DETECCIÓN DEL SERVIDOR OPC ARDUINO AUTOR ()	79
FIGURA 3.35 INTERFAZ DEL CLIENTE OPC COM IGNITION PARA EL SERVIDOR OPC DA ARDUINO AUTOR ()	79
FIGURA 3.36 DIRECCIÓN DE RED EN EL FORMATO URL DEL SERVIDOR OPCUA RASPBERRY AUTOR ()	80
FIGURA 3.37 VENTANA DE EDICIÓN DE LA CONEXIÓN DEL SERVIDOR OPCUA RASPBERRY AUTOR ()	80
FIGURA 3.38 MÓDULO DE CONEXIÓN A SERVIDOR MYSQL AUTOR ()	81
FIGURA 3.39 CONFIGURACIÓN Y VERIFICACIÓN DE LA CONEXIÓN A LA BASE DE DATOS AUTOR ()	82
FIGURA 3.40 CONEXIÓN AL PLC SIEMENS VISTA DESDE LA SECCIÓN OPC UA SERVER /DEVICES AUTOR ()	82
FIGURA 3.41 ESTABLECIMIENTO DE LA CONEXIÓN A LA BASE DE DATOS AUTOR ()	83
FIGURA 3.42 CONEXIÓN DE ARDUINO Y RASPBERRY VISTA DESDE LA SECCIÓN OPC CONECTIONS – SERVERS AUTOR ()	83
FIGURA 3.43 ACTIVACIÓN DE LA PLATAFORMA CON LA OPCIÓN RESET TRIAL AUTOR ()	83



FIGURA 3.44 INGRESO AL DISEÑADOR DE IGNITION AUTOR ().....	84
FIGURA 3.45 CREACIÓN DE NUEVO PROYECTO EN EL DESIGNER DE IGNITION AUTOR ().....	85
FIGURA 3.46 COLOCACIÓN DE LOS OBJETOS EN LA VENTANA DE DISEÑO DEL DESIGNER IGNITION AUTOR ()...	86
FIGURA 3.47 ENLACE DE LAS VARIABLES DEL PLC EN EL FORMATO <i>NOMBRE. INSTANCIA [DIRECCIÓN REGISTRO DEL PLC]</i> AUTOR ().....	87
FIGURA 3.48 ENLACE DE LAS VARIABLES OPC DE ARDUINO EN EL DISEÑADOR IGNITION AUTOR ().....	88
FIGURA 3.49 ENLACE DE OBJETOS CON LAS VARIABLES “ <i>BASA DE DATOS SQL DE ETIQUETAS DEL SISTEMA</i> ” AUTOR ().....	89
FIGURA 3.50 CONFIGURACIÓN DE PARÁMETROS HISTÓRICOS DE UN TAG (CONVERSIÓN TAG A HISTÓRICOS) AUTOR ().....	90
FIGURA 3.51 CREACIÓN DE UN GRUPO ESTÁNDAR “ <i>PLC</i> ” EN ÉL RESPALDO DE UN TAG AUTOR ().....	91
FIGURA 3.52 “EASY CHART” ASISTENTE DE CONFIGURACIÓN DE CONTROL DE CURVAS PARA REPRESENTACIONES GRAFICAS AUTOR ().....	92
FIGURA 3.53 CONFIGURACIÓN DEL TRAZADO EN TIEMPO REAL E HISTÓRICO EN EL “EASY CHART” AUTOR ().....	92
FIGURA 3.54 DINAMIZACIÓN DEL OBJETO MEDIANTE STYLE CUSTOMIZER AUTOR ().....	93
FIGURA 3.55 PROGRAMACIÓN MEDIANTE JYTHON EN LAS PROPIEDADES DEL BOTÓN PARA CONFIRMACIÓN DE ACCESO AUTOR ().....	94
FIGURA 3.56 SCRIPTS MEDIANTE EXPRESSION EN LAS PROPIEDADES DE LOS COMPONENTES AUTOR ().....	94
FIGURA 3.57 CONFIGURACIÓN DE ALARMAS CON EL TAG OPC ARDUINO AUTOR ().....	95
FIGURA 3.58 ALMACENAMIENTO Y PUBLICACIÓN DEL PROYECTO AUTOR ().....	96
FIGURA 3.59 EJECUCIÓN DE LA APLICACIÓN SCADA COMO CLIENTE OPC UA AUTOR ().....	97
FIGURA 3.60 MODOS DE EJECUCIÓN DE LA APLICACIÓN CLIENTE “SCADA OPC UA” AUTOR ().....	97
FIGURA 3.61 APLICACIÓN CLIENTE EJECUTADA EN EL MODO DE PANTALLA COMPLETA AUTOR ().....	98
FIGURA 3.62 MODIFICACIÓN DE LA INTERFAZ Y PUBLICACIÓN DE LOS CAMBIOS AUTOR ().....	98
FIGURA 3.63 AVISO DE ACTUALIZACIÓN AUTOMÁTICA AL CLIENTE AUTOR ().....	99
FIGURA 3.64 ACTUALIZACIÓN DEL CLIENTE, DESPUÉS DE MODIFICAR LA INTERFAZ PRINCIPAL AUTOR ().....	99
FIGURA 3.65 CONFIGURACIÓN DE PUERTO PARA SERVICIO WEB EN LA CGU AUTOR ().....	100
FIGURA 3.66 ERROR CON EL BROWSER DE OPC COM AUTOR ().....	102
FIGURA 3.67 HABILITAR PERMISO DE CONEXIÓN POR OPC DA 3 AUTOR ().....	102
FIGURA 3.68 ESTRUCTURA DEL INVERNADERO A ESCALA REAL AUTOR ().....	103
FIGURA 3.69 ESTRUCTURA DEL MODELO A ESCALA DE INVERNADERO AUTOR ().....	104
FIGURA 3.70 LÁMINAS DE MICA EN EL INVERNADERO AUTOR ().....	104
FIGURA 3.71 INSTALACIÓN DE LAS LÁMINAS DE MICA EN LA ESTRUCTURA A ESCALA DEL INVERNADERO AUTOR ().....	104
FIGURA 3.72 MOTOR DE VENTILADOR DE 12 VOLTIOS DC Y SENSOR DE TEMPERATURA DHT11 DE 5 VOLTIOS DC AUTOR ().....	105



FIGURA 3.73 A) SENSOR DE HUMEDAD DEL SUELO Y B) SENSOR DE LLUVIA CON CIRCUITO AMPLIFICADOR AUTOR ().	105
FIGURA 3.74 INVERNADERO CON TODOS LOS ELEMENTOS DE CONTROL Y MONITOREO AUTOR ().	106
FIGURA 3.75 INSTALACIÓN DE EQUIPOS Y SENSORES EN EL MODELO A ESCALA DE INVERNADERO AUTOR ().	106
FIGURA 3.76 MODELO A ESCALA TERMINADO AUTOR ().	107
FIGURA 3.77 VISTA DEL SISTEMA DE VENTILACIÓN Y FUENTE DE CALOR AUTOR ().	107
FIGURA 3.78 VISTA DE CONEXIÓN DE CONTROLADORES OPC A LA RED LAN Y DISPOSITIVOS DE CAPTACIÓN DEL INVERNADERO AUTOR ().	107
FIGURA 4.1 SISTEMAS SCADA OPC, INTERFACE DE MONITOREO Y CONTROL AUTOR ().	109
FIGURA 4.2 INTERFAZ AJUSTES DE PROCESO PARA AUTOMATISMO DE CONTROL OPC AUTOR ().	109
FIGURA 4.3 VENTANA DE GRAFICAS DE PROCESO OPC AUTOR ().	110
FIGURA 4.4 VENTANA DE REGISTRO DE ALARMAS DEL SCADA OPC AUTOR ().	111
FIGURA 4.5 SERVIDOR OPC UA RASPBERRY CONECTADO AL SISTEMA SCADA OPC AUTOR ().	112
FIGURA 4.6SERVIDOR OPC DA DE ARDUINO ETHERNET CONECTADO AL SCADA OPC AUTOR ().	112
FIGURA 4.7 VISTA DEL INVERNADERO CON LAS PRUEBAS DE CONTROL AUTOMÁTICO DEL SCADA AUTOR (). .....	113
FIGURA 4.8 VISTA DE LOS ELEMENTOS DE ACCIONAMIENTO Y SENSORES EN EL INTERIOR DEL INVERNADERO CONECTADOS AL SERVIDOR SCADA OPC IGNITION AUTOR ().	113
FIGURA 4.9 RESPUESTA DEL SISTEMA PARA LA HUMEDAD DE 60% Y TEMPERATURA DE 12 ° C AUTOR ().	114
FIGURA 4.10 ALMACENAMIENTO DE LOS DATOS EN MYSQL EN TIEMPO REAL AUTOR ().	114
FIGURA 4.11 ARQUITECTURA DE COMUNICACIÓN CON LOS MÓDULOS DEL GATEWAY, PARA LOS SERVIDORES OPC HEREDADOS (OPC UA Y OPC DA) Y SERVIDOR OPC UA DEL PLC AUTOR ().	115
FIGURA 4.12 SELECCIÓN DE PUERTO Y PROTOCOLO OPC UA CON WIRESHARK AUTOR ().	116
FIGURA 4.13 CAPTURA DE DATOS EN EL TRÁFICO DE LA RED LOCAL CON EL FILTRO OPC UA AUTOR ().	116
FIGURA 4.14 REVELACIÓN DE INFORMACIÓN EN UN PAQUETE OPC UA AUTOR ().	117
FIGURA 4.15 ESCRITURA EN LAS VARIABLES DE LOS SERVIDORES CON EL SCADA OPC IGNITION DESDE DOS CLIENTES EN ENTORNOS ANDROID Y WINDOWS EQUIPOS CON UN NAVEGADOR WEB AUTOR ().	118
FIGURA B.1 DIAGRAMA DE CLASES ARDUINO ETHERNET PARA CON EL SERVIDOR OPC DA AUTOR ().	129
FIGURA B.2 DIAGRAMA DE CLASES DE SERVIDOR OPC UA RASPBERRY PI AUTOR ().	130
FIGURA C.1 COSTO ESTIMADO DEL PROYECTO AUTOR ().	131
FIGURA D.1 VISTA DEL PLC CONECTADO AL SCADA OPC UA IGNITION AUTOR ().	133
FIGURA D.2 VISTA DE LOS SISTEMAS EMBEBIDOS, ELEMENTOS “MEDICIÓN Y CONTROL” Y ENLACE DE RED LOCAL DEL SCADA OPC UA IGNITION AUTOR ().	133
FIGURA D.3 VISTA GENERAL DEL SISTEMA SCADA OPC UA IGNITION INSTALADO CON EL INVERNADERO EN EL LABORATORIO AUTOR ().	133



## Índice de tablas

---

TABLA 2-1 COMPONENTES DEL SOFTWARE SCADA.....	31
TABLA 2-2 PROTOCOLOS QUE SE DIFERENCIAN POR LA VELOCIDAD DE RESPUESTA, AL TRANSMITIR DATOS ....	41
TABLA 3-1 LONGITUD DEL DATO A LEER/ESCRIBIR EN UN PLC.....	57
TABLA 3-2 ÁREA DE MEMORIA DE UN PLC.....	58
TABLA 4-1 VALORES DE CONFIGURACIÓN DE SETEO AUTOMÁTICO DEL SISTEMA .....	111
TABLA D-1 VALORES DE LOS DISTINTOS SISTEMAS DE GOBIERNO (ARDUINO, RASPBERRY Y PLC) OPC .....	
AUTOR (). .....	132



## Capítulo 1

### Introducción

---

De la mano de las redes de comunicación, la automatización industrial integrada es hoy en día una realidad, permitiendo que sea posible la interconexión de sistemas y procesos en la industria, constituyendo una nueva forma de automatizar procesos industriales, al permitir que sistemas de automatización, puedan intercomunicar el nivel de campo con el sistema de control y supervisión por computadora (SCADA), incluso escalar hacia sistemas empresariales.

Sin embargo, la falta de una integración eficiente, diverge entre diferentes elementos de automatización (como PLC), debido a que no todos disponen de un mismo sistema de comunicación, lo cual evita su adecuado aprovechamiento en un sistema de supervisión, situación que ha llevado a disponer de soluciones costosas, con protocolos de comunicación propietarios, limitando la migración hacia nuevas tecnologías.

Ante la falta de una estandarización y lo costoso de un PLC y sumado a la necesidad de hardware especializado por los sistemas propietarios. Abren las puertas para analizar la posibilidad de encontrar otros métodos. Situación que ha sido atractiva para utilizar, el protocolo de comunicación industrial abierto (OPC) y mitigar hacia nuevas posibilidades de integración con tecnologías contractuales de bajo costo, que ofrece el mercado.

El presente trabajo, trata sobre el desarrollo de un sistema de control y monitoreo interoperable con múltiples tecnologías, mediante el diseño de una arquitectura de comunicación OPC basada en el modelo unificado cliente – servidor. El cual se ha llevado a cabo para la optimización de un proceso de cultivo dentro de un invernadero, a través de la medición de los parámetros de humedad, temperatura y cantidad de agua.



Se ha empleado servidores OPC comprendidos por el protocolo como el OPC DA y OPC UA, quienes proporcionan datos a un cliente OPC UA, integrado en un sistema de gestión. Para ello se ha decidido implementar recursos tecnológicos, combinando equipos industriales como el PLC Siemens S7-1200 y sistemas embebidos como Arduino Ethernet y Raspberry Pi, a fin de explorar nuevas opciones tecnológicas en el campo de sistemas de control y automatización de procesos y ver el alcance que estas tarjetas tienen en la elaboración de proyectos universitarios, así como también para hacerlo más viable se utilizan elementos de medición de uso diario en la gestión del proceso, evitando que el monitoreo requiera de un hardware en concreto.

Para lo cual, se ha buscado establecer enlaces con la plataforma de diseño industrial SCADA Ignition, en la cual se ha construido una aplicación de sistema SCADA, mediante un servidor de aplicaciones en la nube, desde donde se gestiona y proporciona los datos del proceso. El sistema permite el control y monitoreo de elementos de automatización, como también enlaces a distancia, permitiendo que los datos estén siempre disponibles para los usuarios que deseen arribar a la aplicación a través de una interfaz gráfica basada en la web.

El proyecto permite, hacer uso de tecnologías competentes, aprovechando las bondades que ofrece, el hardware y software libre en lo que respecta a Arduino y Raspberry Pi, viéndose como oportunidades integrables a un entorno industrial y como una solución a los problemas de financiamiento e interoperabilidad, a través de una arquitectura de hardware y software accesible.

### **1.2. Antecedentes**

Con el paso del tiempo, el ámbito industrial y los sistemas industriales vienen presenciando importantes avances tecnológicos, en la automatización de sus procesos. La aparición de los controladores lógicos programables (PLC), marcó un cambio protagónico en la revolución industrial. Desde entonces los sistemas de producción implementan, estos elementos de automatización logrando que el control de sus procesos industriales sea más eficiente, rápido, preciso y sobre todo reprogramable.



Desde la óptica del control, el PLC se ha convertido gradualmente en un componente imprescindible de todo sistema de automatización, siendo el promotor para que actualmente las empresas, dispongan de mejor manera la gestión de sus actividades de producción, dando lugar a un desempeño eficiente, flexible y oportuno, que contribuye a mejorar la competitividad y producción, reduciendo costos en la generación de bienes y servicios, incrementando la calidad y el volumen de producción.

Sin embargo cada vez más las exigencias del entorno industrial, apuntan hacia la necesidad de una automatización integrada, que puede ir desde meros procesos de fabricación y manufactura, hasta la incorporación de diferentes áreas dentro de la empresa (como fabricación, gestión de producción, distribución, etc.). Ante esto las redes de comunicación, vienen siendo un componente clave para la intercomunicación de sistemas y procesos dentro del ambiente industrial. Su intervención se fundamenta en el simple hecho de brindar una mayor eficiencia y optimización global de recursos.

Sin embargo el mercado está plagado de varias alternativas tecnológicas, que han influido en un crecimiento industrial desorganizado, dado que nuevos procesos se acoplan a los anteriores con una integración deficiente que imposibilita, el mejor aprovechamiento de recursos. La disposición de diferentes marcas de PLC's, discrepan en una misma industria dando lugar a protocolos propietarios, que el mercado monopoliza a proveedores únicos de soluciones para una marca específica encada producto, al no existir normas que regulen su coexistencia, conllevan al diseño de protocolos propios de comunicación dependientes de un hardware (drivers) o software en concreto. Aquella disposición limita la intercomunicación y por ende la conformidad de la red industrial.

En este aspecto varias organizaciones, suman sus esfuerzos por proveer regulaciones que contrarresten las limitaciones de interconectividad, de los sistemas de procesos a las redes industriales. Una de ellas es la *OPC Foundation*, organización sin fines de lucro, que aportado con una solución firme, confiable y sostenible. Este es el estándar de comunicación en el campo del control y supervisión de procesos industriales (OPC).



El protocolo, ofrece una interfaz común de comunicación, permitiendo integrar cualquier sistema industrial, basado en su estándar a la red de comunicación industrial interconectando todo elemento y sistema del entorno. Su desarrollo, va desde una versión clásica orientada al acceso a datos (OPC DA) a una multiplataforma, bajo una arquitectura unificada (OPC UA), convirtiéndose en uno de los principales referentes, para el desarrollo de sistemas interoperables basado en el modelo cliente – servidor.

Gracias a su atribución de protocolo de comunicación abierto, OPC ofrece la posibilidad de aprovechar opciones tecnológicas más rentables, como son los sistemas *Open Source* o entidades colaborativas a fines, que confieren productos para el libre desarrollo de aplicaciones pudiendo implementarse en casi cualquier sistema de control automático.

Las redes de comunicación y OPC, hacen de su combinación oportunidades para aplicaciones de sistemas interoperables a nivel industrial, ambos conceptos comprenden un amplio campo en el adelanto tecnológico con un constante desarrollo y mejora, proveyendo herramientas, para la interconectividad de varios dispositivos que viabiliza la optimización en la productividad, confiabilidad y operabilidad de los procesos, que se despliegan desde la simple automatización de un proceso, hasta el control y supervisión absoluta de una empresa, permitiendo el apoyo a pequeños y medianos emprendimientos a costos accesibles mediante alternativas tecnológicas asequibles como son los sistemas embebidos.

### **1.3. Situación problemática**

Los sistemas de control de procesos han sido afectados por arquitecturas propietarias en la adaptación de sistemas supervisores y dispositivos de control, lo cual ha impedido la migración hacia soluciones más viables y coherentes, en hardware y software, que permitan mejorar la versatilidad de la función operativa en un proceso.

La presencia de monopolios tecnológicos, no permiten disponer de un mismo medio de comunicación entre aplicaciones y dispositivos de diferentes proveedores dado que cada fabricante diseña un software personalizado para cada producto, bajo los distintos tipos de



protocolos de comunicación, dando lugar a la incompatibilidad entre marcas ajenas a los dispositivos.

Sin embargo, la aparición de controladores (drivers) en el mercado, ha brindado una solución parcial a los problemas de interactividad entre diferentes sistemas de control; no obstante, el implementar varios drivers y software para cada dispositivo, resulta difícil, costoso e ineficiente, ante la aglomeración de interfaces dedicadas para establecer el dialogo con cada dispositivo, que a su vez proveen de información, para el seguimiento y control de los mismos en un sistema dedicado.

La falta de convergencia tecnológica, que ha restringido el despliegue y la integración de nuevas fuentes de información provenientes de dispositivos de adquisición de datos, afectan la eficiencia de un sistema de supervisión y de control al cubrir nuevas áreas de un proceso.

Por ello, se considera como propuesta, la implementación de estándares de comunicación abierta OPC (Object Linking and Embedding for Process Control), que por medio de plataformas y hardware de bajo costo permitan el diseño de un sistema de supervisión, control y adquisición de datos “SCADA” Inter-operativo e interactivo, que resulten en soluciones compactas y específicas para cubrir las necesidades planteadas.

## **1.4. Línea de investigación**

Ciencias exactas, naturales y tecnológicas

### **1.4.1 Ámbito**

Control

### **1.4.2 Objeto de estudio**

Control industrial

### **1.4.3 Campo de acción**

Monitoreo y automatización



## **1.5. Objetivos**

### **1.5.1 Objetivo General**

Realizar la adaptación del estándar de comunicación (OPC) sobre un prototipo de sistema SCADA, utilizando el software Ignition y sistemas embebidos, para el diseño de una infraestructura de intercomunicación versátil de bajo costo para controladores lógicos programables, permitiendo expandir las funcionalidades de automatización.

### **1.5.2 Objetivo específico**

- Fundamentar los conceptos necesarios en las áreas de sistemas embebidos, controladores lógicos programables, redes de comunicaciones industriales, sistemas SCADA y protocolos de comunicación para viabilizar el desarrollo del proyecto.
- Diseñar e implementar un prototipo de sistema SCADA, a través del protocolo de comunicación abierta (OPC), para la integración de varios dispositivos como fuentes de adquisición de datos.
- Evaluar y validar el desempeño del prototipo de sistema SCADA implementado en un ambiente industrial, mediante el análisis de resultados.

## **1.6. Hipótesis**

Este proyecto, permite expandir las capacidades y funcionalidades de sistemas de automatización clásicos, optimizando la gestión operativa en una sola infraestructura de control y monitoreo, basada en sistemas industriales, para diferentes dispositivos como fuentes de información.

## **1.7. Justificación**

Se desea disponer de un medio de comunicación compatible, para la integración de herramientas tecnológicas de diversas marcas y propietarios, que a su vez hagan posible la adaptación de estándares de comunicación abierta “OPC” que permitan el intercambio de



información y el acceso a los datos en tiempo real de forma fácil, rápida y segura. Aprovechando las plataformas abiertas y amigables que faciliten el desarrollo de arquitecturas unificadas de comunicación, sin importar el formato de dato y tipo de protocolo de comunicación, dotando de una alternativa de menor costo que se incorpore a las plataformas industriales convencionales.

La importancia de este proyecto radica, en el hecho de evitar inversiones costosas en hardware y software de tipo propietario, además de las dificultades operativas de programas y al uso recursivo de drivers específicos para múltiples fuentes de información (Controladores lógicos programables (PLC), Sistemas de Control Distribuido (DSC), etc.). Como también reducir la saturación conectiva de equipos de comunicación de datos, ante la concurrencia de peticiones de información y ejecución de protocolos y programas. De forma que se mejoró la versatilidad y el flujo de comunicación de los sistemas SCADA, permitiendo compartir información en forma abierta y efectiva, satisfaciendo las expectativas de control y monitoreo en el despliegue hacia nuevas fuentes de información de diversos dispositivos.

La utilidad que genera este proyecto es que se disponga, de un sistema interoperativo de fácil manejo, que simplifique los enlaces entre los diversos sistemas, reduciendo la dependencia de proveedores de una misma línea de producto o software en particular, fomentado el mutuo acuerdo entre diversas marcas de equipos y sus proveedores en un estándar de comunicación internacional. Utilizando un entorno de enlace universal para transferir datos independientemente de la aplicación y del tipo de conectividad. Abriendo un camino hacia el aprendizaje de nuevos métodos y técnicas de comunicación en el campo del control y supervisión de procesos que se complementen a los conocimientos adquiridos en la formación académica.

## Capítulo 2

### Marco teórico

#### 2.1. Controladores lógicos programables (PLC)

Actualmente los controladores han sido parte indiscutible, de cualquier sistema, que a través de la lectura y escritura de variables, advierten sobre el estado actual de un proceso. La aparición de microprocesadores ( $\mu p$ ), admiten que distintas situaciones de control puedan ser manipuladas por un único sistema, obviando la dependencia de hardware especial para cada aspecto de control.

El PLC es un equipo electrónico, basado en un microprocesador diseñado especialmente para la automatización industrial en ambientes hostiles. Mediante módulos de entrada y salida (E/S) de señales analógicas y digitales, permiten realizar tareas de control concretas en base a condiciones específicas (programas), adecuándose fielmente a las necesidades de automatización. La siguiente imagen presenta los bloques y elementos de este proceso.

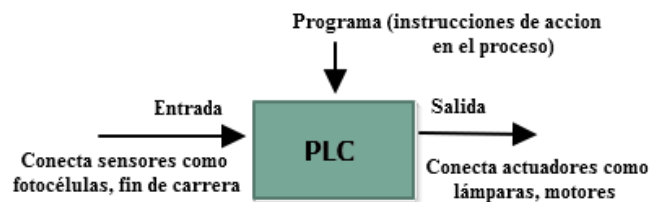


Figura 2.1 Esquema de E/S del PLC Autor ().

Las salidas generadas por el PLC dependen, en esencia de las condiciones de entrada establecidas en la lógica del programa almacenado, que recopilan información del proceso para emprender estrategias de control con distintas maquinarias y procesos industriales. La misma infraestructura del programa puede cambiar de lógica para nuevas tareas. Esto hace que la información puede ser compartida con otras unidades u operarios dentro del proceso industrial a través de buses de comunicación (Hilariona Martínez, 2015).

Todo PLC se halla constituido por: una unidad central de procesamiento, fuente de

alimentación, interfaces de entrada y salida, interfaz de comunicación y unidad de programación, como lo indica la Figura 2.2.

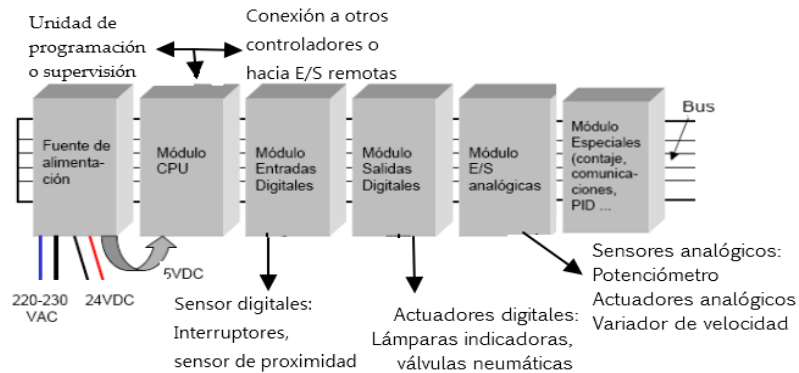


Figura 2.2 Arquitectura interna de un PLC (Electrin, 2016).

A continuación se detalla cada unidad del PLC:

- Unidad central de procesamiento (CPU). Es la parte inteligente del PLC donde el microprocesador consulta el estado de las señales de ingreso y acciones en la salida.
- Fuente de alimentación. Proporciona la alimentación con tensiones adecuadas a los circuitos del sistema y procesador por lo general de 110VAC a 5VDC.
- Unidad de programación. Es la interfaz de comunicación entre el operador y el PLC, ejecuta instrucciones o registros del programa, cíclica e ininterrumpidamente.
- Unidad de memoria. Sitio donde se almacena el programa escrito por el usuario para la aplicación de control dispone de dos tipos de memorias (Electrin, 2016):
  - ROM. Es una memoria solo de lectura, almacena y ejecuta el sistema operativo del PLC.
  - RAM. Es una memoria de acceso aleatorio, permite leer y escribir en diversas posiciones de la memoria, creando imágenes tanto de las entradas como de las salidas sobre sus estados (Villajulca, 2012).



- Interfaces de E/S. Permiten que el PLC pueda comunicarse con el entorno de un proceso, mediante la información proporcionada por señales, de entrada y salida.

El PLC posee direcciones de registros que se vincula con los bornes (véase la Figura 2.3).

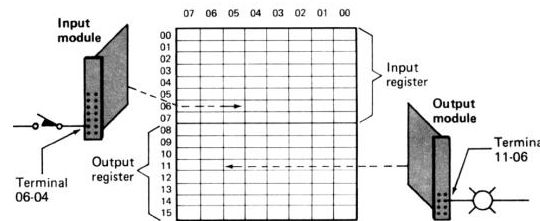


Figura 2.3 Tabla de registro del microprocesador, cualquier cambio de estado del registro se refleja en los bornes y la memoria del PLC siendo intuitivo e interactivo para el usuario (M. Moreno, 2015).

### 2.1.1 Lenguajes de programación

Los PLC disponen de distintos lenguajes de programación. Para ser programado y manipulado con facilidad. Entre ellos el lenguaje de escalera (indicado en la Figura 2.4) que a través, de símbolos normalizados permiten el diseño gráfico que emula el funcionamiento del PLC, permitiéndole al usuario; definir aplicaciones, estructuras del sistema de control y el grado de comunicación, para la lógica del proceso (David & Aguirre Grazio, 2017)& (M. Moreno, 2015).

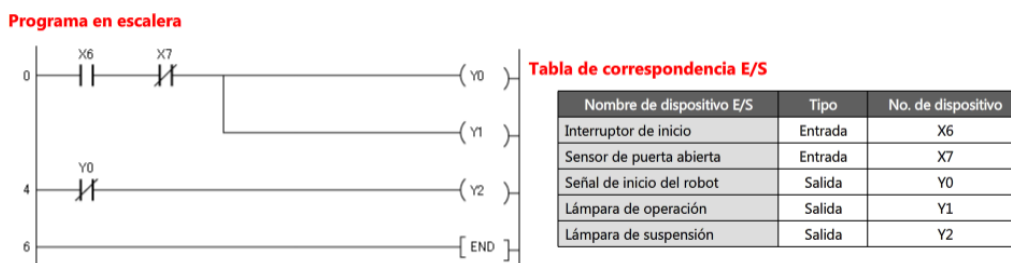


Figura 2.4 Lenguaje escalera (con contactores -| | - y bobinas - (-)-) (Mitsubishi, 2014).

### 2.1.2 Medios de programación

Por lo general se conectan con el PLC, a través del puerto de red (Ethernet) y son indispensables para la grabación y diseño de programas, proporcionando el control, visualización y emulación del estado de operación del PLC. La cual se puede realizar mediante dispositivos (David & Aguirre Grazio, 2017): portátiles, consolas de escritorio y computadores personales.



Aunque son considerados como computadoras divergen, de las prestaciones y capacidades. El PLC está optimizado para tareas de control, en tanto que los ordenadores procesamiento.

## 2.2. Raspberry Pi

Es una microcomputadora de bajo costo del tamaño de una tarjeta de crédito, desarrollada por la *Fundación Raspberry Pi*. Diseñada como medio que facilite e incentive la enseñanza de ciencias computacionales y programación hacia escuelas y universidades. Dispone de los elementos, necesarios para desempeñarse y funcionar como un ordenador, realizando tareas de cálculo y procesamiento, navegación por internet, reproducción multimedia, instalación de programas, y soporte a conexiones para monitor, teclado, mouse y periféricos de E/S.

La principal diferencia y logro de Raspberry Pi sobre los ordenadores es que incorpora un micro procesador (chip Boardcom) de arquitectura ARM (Intruction Set Architecture, ISA), al cual debe su diseño, flexibilidad y bajo consumo de energía (Eben Upton & Gareth Halfacree, 2013). En el cual se ha fomentado el desarrollo de software de código abierto, especialmente bajo distribuciones de Linux, permitiéndole crear su propio sistema operativo y modificar su código fuente como se desee.

### 2.2.1 Software

Raspberry puede explorar las virtudes de entornos de distintas plataformas, principalmente distribuciones basadas en GNU/Linux, bajo la filosofía de software libre y adoptando sus bondades, sean ajustado y atendido para diversos fines. El *Debian* es un sistema operativo Raspbian, siendo el más usado y completo por sus prestaciones multipropósito. OpenElec, es un sistema de propósito específico para aplicaciones domóticas (NERGIZA, 2015).

Otras de las características del software son (NERGIZA, 2015):

- Se ejecuta y almacena en la tarjeta SD, facilitando la transición a varias plataformas.
- Incluye herramientas de programación en Python y Scratck para su aprendizaje.
- Es capaz de recibir, procesar y transmitir, con alto rendimiento la información.



Usualmente Raspberry Pi, puede ser usado en aplicaciones tanto de hardware como de software, disponiendo de periféricos para la conexión de dispositivos externos.

### 2.2.2 Características de hardware

En este proyecto se tendrá en cuenta el modelo B de segunda generación de Raspberry, que dispone de todas las opciones elementales para su uso. La Figura 2.5 muestra su diseño y descripción técnica:

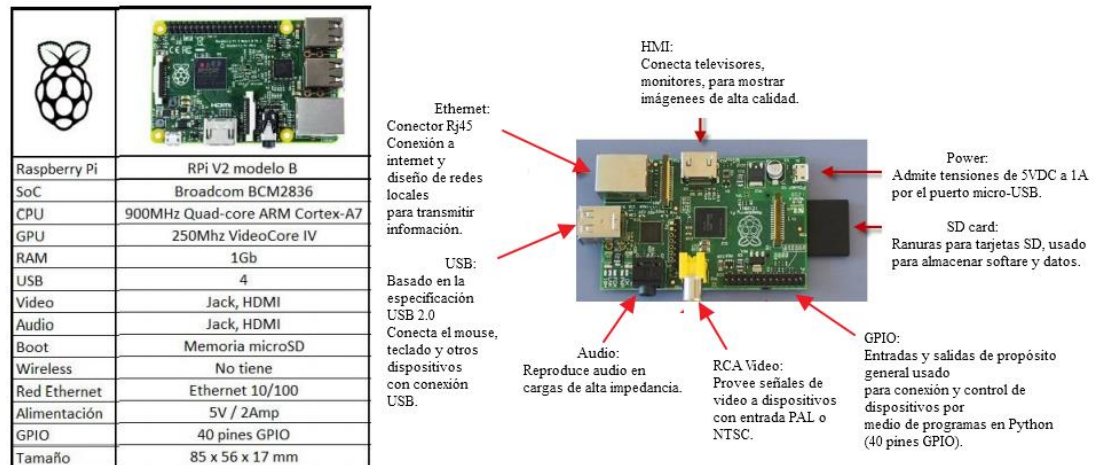


Figura 2.5 Descripción de características técnicas y elementos de Raspberry Pi 2 B (V\_estinf, 2013).

### Limitaciones de hardware y software

- No tiene disco duro, por eso utiliza tarjetas SD.
- Desde su funcionamiento no tiene ningún sistema operativo instalado.
- Es incompatible con software de los ordenadores por las diferencias entre las arquitecturas de los procesadores de ambos.
- Linux no es exclusivo de RPi solo utiliza distribuciones modificadas por las limitaciones de memoria RAM y nivel de procesamiento.
- No contiene convertidores (ADC) y (DAC).

### 2.3. Arduino

Es una plataforma de prototipos electrónica, de fácil uso implantada sobre hardware y software *Open Source*. Está diseñada como herramienta, que facilite y promueva la enseñanza y el aprendizaje de la electrónica y programación, concebida con la idea de que

diseñadores, profesionistas y que cualquier persona, puedan crear entornos u objetos interactivos de forma libre y asequible, sin prohibiciones de ninguna naturaleza, dentro de su entorno de desarrollo integrado (IDE).

El IDE es el software de programación de Arduino y es multiplataforma. Un entorno dinámico, que como producto de las derivaciones de su software base (Processing/C++), la placa puede ser gestionada desde diferentes entornos de programación (como; Python, Visual Basic, etc.), bajo comunicaciones seriales (Amangandi, 2012)& (Paul Lara, 2017).

Su atributo Open Source, le permite dotar de toda información de hardware y software libremente, permitiéndole a terceros crear, modificar y mejorar sus versiones (Fernando Doutel, 2015). Disponiendo de nuevos dispositivos (Shields) y librerías que aportan a incrementar las funcionalidades y prestaciones de la plataforma Arduino.

### 2.3.1 Hardware

La placa se compone de una serie de puertos de E/S, con un microcontrolador programable capaz de ejecutar órdenes grabadas en su memoria, que proveen el control y la autonomía de cualquier dispositivo conectado, e inclusive obtener información de una fuente física (temperatura, humedad, electricidad, etc.). La Figura 2.6 muestra la interacción del sistema con el entorno físico mediante dispositivos de E/S como sensores y actuadores.



Figura 2.6 Proceso de lectura y escritura en las E/S de Arduino (Amangandi, 2012).

Para este proyecto se analiza, la plataforma Arduino Uno R3, que alberga los componentes necesarios, adecuados para su funcionamiento. Este modelo tiene un tamaño de 73X53 mm, contiene el microcontrolador ATmega 328, quien procesa la información adquirida e instrucciones programadas en su memoria. La tarjeta dispone de un LED para pruebas de



funcionamiento conectado al pin digital 13 de la placa. En la Figura 2.7 se presenta la descripción de cada elemento y características técnicas de la tarjeta Arduino UNO R3.

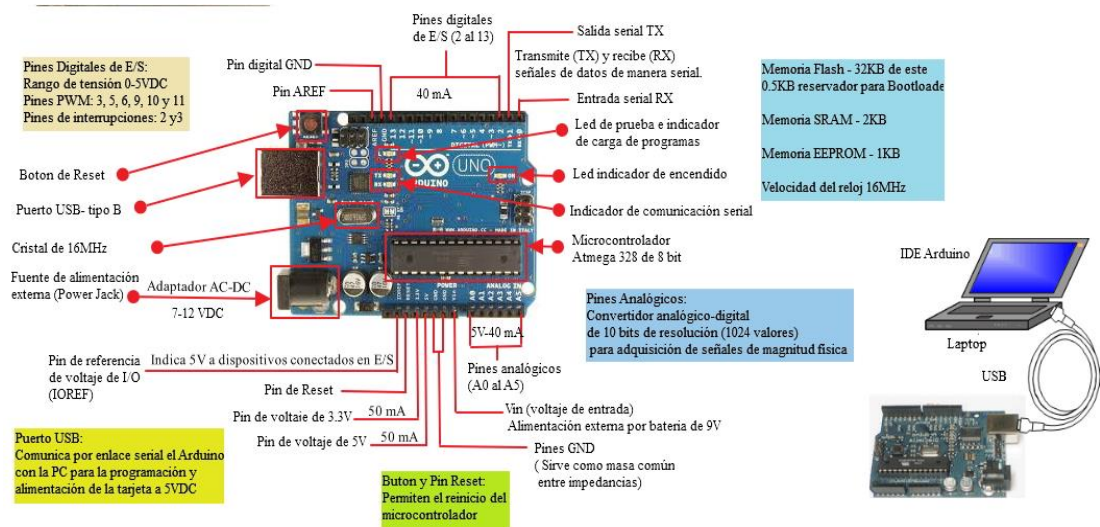


Figura 2.7 Disposición de los elementos del Arduino Uno R3 Autor ().

## 2.4. Sistema SCADA

Un sistema de control, supervisión y adquisición de datos (SCADA), es una aplicación que a través de una interfaz gráfica de usuario, integra y muestra el estado actual de los distintos elementos de un proceso, permitiendo el control y supervisión de equipos de campo, repartidos geográficamente dentro de la industria, donde el operador puede disponer la administración, comunicación e información de forma precisa y actualizada de dispositivos de planta o maquinaria de un proceso, optimizando la operación a través de un análisis y procesamiento de datos desde una estación central/maestra típicamente una computadora, presentando en la pantalla del operador la interfaz hombre – máquina (HMI).

### 2.4.1 Arquitectura de hardware

Un sistema SCADA se comunica siguiendo la arquitectura *cliente – servidor*, para conectarse con estaciones remotas, fraccionados en la capa de cliente que incorpora el HMI, y la capa de servidor que opera sobre los datos generados por el proceso, ambos conectados por la red Ethernet. En si un SCADA puede ser programado en casi cualquier bus de comunicación, gracias a la *estandarización de dispositivos*, permitiendo el control

de un gran número de variables de elementos esparcidos, acoplándose a las necesidades operativas futuras de un proceso, gracias a la adaptación de *arquitecturas abiertas*.

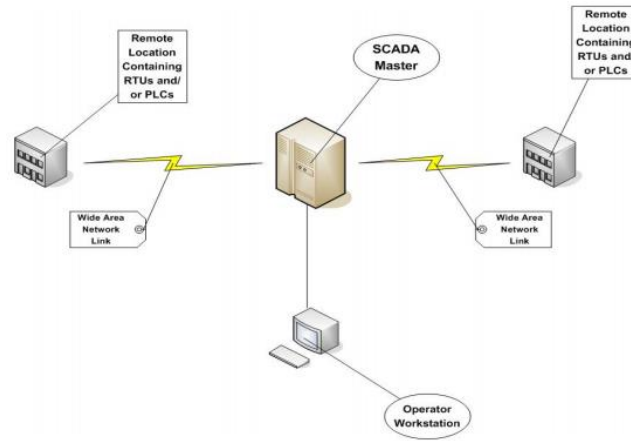


Figura 2.8 Sistema SCADA básico Autor ().

### Elementos de un sistema SCADA

Un sistema SCADA depende de varios componentes, que son necesarios para su correcto accionar (véase la Figura 2.8) dentro de un proceso. Los cuales se organizan según su solvencia. Entre estos están (Pérez López, 2015)& (Sinais, 2014):

- **Equipos de instrumentación.** Son elementos que interactúan directamente con el proceso y cumplen distintas funciones, entre ellos están: sensores (miden variables), indicadores (registran valores), transmisores (envían valores de las variables a otros equipos) y actuadores (regulan el proceso mediante motores, válvulas, etc.).
- **Equipo de campo** o estación remota, es un intermediario entre el SCADA y los dispositivos de campo, para el control y monitoreo, que mediante PLC o RTU (Unidad terminal remota), recolectan datos de las variables físicas del proceso y los envían hacia la estación maestra. Desde el cual se definen instrucciones a través del software SCADA para ejecutar acciones de operación en el PLC o RTU.
- **Red de comunicación,** gestionan la información que aportan los instrumentos de campo. Habitualmente las redes utilizan Ethernet con protocolos TCP/IP y topologías para desarrollar la comunicación. Según la necesidad de enlace se conocen dos tipos de topologías (Sinais, 2014): la punto – punto (altas tasas de transmisión



entre una estación central y una remota) y punto – multipunto (medio compartido bus de comunicación por distintas estaciones remotas, baja tasa de transmisión).

- **Estación central**, soporta el HMI que interactúa con el operador, cuyas funciones son; supervisar y recolectar información de las estaciones remotas y las presenta en tiempo real, administra alarmas, datos históricos y ejecuta el software SCADA. En si crea estadísticas que permiten analizar el desempeño y llevar un registro de fallas del proceso, sobre el cual mantiene una base de datos.

#### 2.4.2 Arquitectura del Software

El software SCADA, es un servidor de datos, que está encargado de la manipulación y recolección de información. En la Tabla 2.1 se muestran los componentes de software.

Tabla 2-1 Componentes del software SCADA

Servidor esclavo/datos SCADA
Administrador de sistema en tiempo real
Aplicación de procesamiento de datos
Generador de reportes
Manejo de alarmas
Control e interfaz para manipular componentes
Hojas de calculo
Históricos
Gráficas y tendencias

Fuente: Autor ()

#### **Funciones del SCADA** (Medina, 2015):

- **Supervisión.** Permite al operario observar desde el monitor el comportamiento de las variables de control del proceso en tiempo real, siendo oportuno y confiable.



- **Control.** El sistema permite activar y desactivar los equipos en forma manual y automática ante funciones operacionales o anomalías en el proceso.
- **Adquisición de datos.** Procesa, almacena y muestra información proveniente de los dispositivos de campo, en forma transparente.
- **Generación de reportes.** Se puede generar tendencias en base a la información recopilada para predicciones, control estadístico, etc.
- **Representación de alarmas.** Advierte al operario de problemas o la presencia de condiciones perjudiciales, mediante señales de sonido y visibles.

## 2.5. Redes de Comunicación Industrial

Es un conjunto de herramientas electrónicas y de control, que proporcionan la integración y consolidación de distintas áreas y dispositivos donde coexisten varios sistemas de control, maquinarias, y procesos del entorno industrial. Estas constituyen el medio o soporte para las estrategias de automatización, que permiten el intercambio de datos, entre diferentes niveles de operación. La red industrial está diseñada para ambientes hostiles por su robustez y confiabilidad en la comunicación de los elementos que la conforman, ocupando al PLC como elemento clave, para la automatización integrada de la red industrial.

El sistema intercomunica en una sola plataforma recursos humanos y la diversidad tecnológica, que interactúan para la gestión y monitoreo del ciclo de vida de los procesos (Conectronica, 2013). Provee nuevas posibilidades de expansión, administra la operatividad de dispositivos y la transmisión de información en tiempo real, rige el intercambio de datos en un protocolo de comunicación, que comprende las reglas y convenciones necesarias que debe seguir cualquier dispositivo como; el medio de transmisión, lenguaje de diálogo, tiempo de conexión, adecuando la transmisión por la flexibilidad de software estandarizado (César Caicedo-Eraso, Rocío Varón-Serna, & Octavio Díaz Arango Profesor, 2015).

Entre las características por optar una red industrial se encuentran:

- ✓ Visualización y control de todos los procesos de forma rápida y eficaz.
- ✓ Se precisa de una velocidad de transmisión y respuesta de datos más rápida y mayor volumen de datos que otras redes (como de las de computación).



- ✓ Mayor rendimiento por componentes que integra (computadoras y controladores).
- ✓ Control distribuido de los sistemas automatizados, que puede ser bajo un soporte de red inalámbrica, de fibra óptica, cable de cobre entre otros.
- ✓ Intercambio de información entre diferentes puntos de conexión (dispositivos).

## 2.6. Pirámide CIM (Computer Integrated Manufacture)

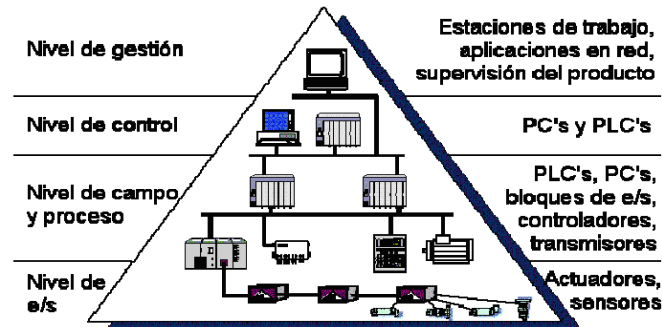


Figura 2.9 Clasificación de la Red Industrial en la CIM (Ingenieratic, 2017)

La pirámide CIM (véase la Figura 2.9) organiza un sistema distribuido, en el cual se muestra cada nivel de la red de comunicación industrial, agrupando jerárquicamente distintos dispositivos en función de la información que manejan, para su adecuada conexión, según el área de aplicación y función que desempeñan dentro de la industria.

Cada nivel supone de distintas soluciones de comunicación, para los distintos equipos. Siendo la red industrial capaz de asociarlos y prepararlos para que se comuniquen directamente entre sí. Actualmente se definen cuatro niveles o redes básicas, en la pirámide CIM, como (Conectronica, 2013):

- **Nivel de gestión o de información.** Integra todos los subsistemas, de los niveles en una estructura de fábrica y gestiona toda la planta. Se halla conformada por computadores de oficina, estaciones de trabajo, que manejan grandes volúmenes de información de producción, para realizar el análisis y control supervisado de los procesos productivos (Fernando & Aníbal, 2015).
- **Nivel de control.** Realiza tareas de mando y administración en diferentes zonas de labor. Con PLC's, sistemas SCADA, computadoras para el control de calidad, etc.

- **Nivel de campo y procesos.** En este nivel se realiza el procesamiento de los datos del campo, e integran y ejecutan pequeños automatismos a través de multiplexores de E/S, controladores PID, etc., sobre los elementos de nivel inferior.
- **Nivel de E/S.** Se hallan los elementos de medición y adquisición de datos como sensores y de mando actuadores, que se encargan debidamente del proceso productivo. Este nivel conecta dispositivos de campo con el nivel de controladores.

## 2.7. Topología de redes industriales

La topología de una red, define la forma en la cual se van a interconectarse y distribuirse los elementos para la comunicación del sistema industrial, a medida que este se agranda. Entre estos se considera tres tipos de topologías, que se describen en seguida (Universidad de Oviedo, 2016):

- **Topología de bus**

En esta red todos los dispositivos están conectados, a un medio de comunicación común. Los datos transmitidos pueden ser vistos por todos los nodos conectados, pero solo el nodo destino tiene acceso a la información. Antes de transmitir un mensaje cada nodo debe consultar si el medio está disponible, así solo un mensaje puede circular en el medio en cada momento (véase la Figura 2.10). Es susceptible a colisión del medio por envío simultáneo de dos estaciones y la alta ocupación.

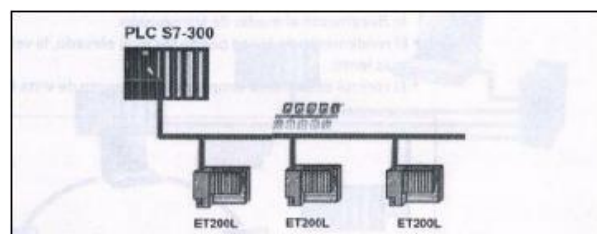


Figura 2.10 Estructura de la red tipo bus (Fernando & Aníbal, 2015).

- **Topología de anillo**

Los elementos se conectan unos a otros formando un anillo como lo indica la Figura 2.11. La transmisión está configurada unidireccionalmente, de tal forma que la



información enviada pasa de un nodo a otro sucesivamente hasta llegar al destino, el cual debe estar habilitado para identificar su dirección en el mensaje. La velocidad de transmisión lo establece el equipo más lento, sin embargo es posible que circule más de un mensaje en la red. Su inconveniente, es que la sola falta de un equipo afecta a toda la red. No existe la alta disponibilidad del medio y la atenuación.

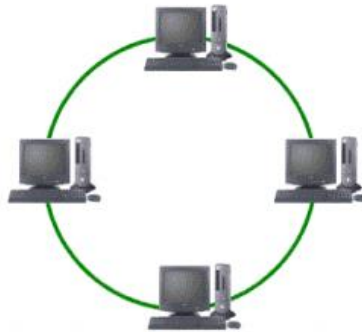


Figura 2.11 Estructura de una red anillo (Fernando & Aníbal, 2015).

- **Topología estrella**

Todos los nodos de la red se conectan a un único dispositivo central llamado concentrador o HUB (véase la Figura 2.12). Este se encarga de gestionar toda la información de los nodos en la red, enviando y recibiendo mensajes simultáneamente, admitiendo distintas velocidades, siendo la red absolutamente dependiente del HUB, si éste falla la red entera también.



Figura 2.12 Topología de red estrella (Espinoza, 2014).

## 2.8. Protocolos de comunicación industrial

Un protocolo de comunicación, es un conjunto de reglas que administran en forma ordenada la transmisión e intercambio de información, entre dispositivos de una red,



implementados tanto en hardware (tarjeta de red) como software (drivers) o la combinación de ambos (Poma, 2013). El avance de la tecnología electrónica en microprocesadores, ha dado origen a distintos protocolos de comunicación con características únicas e incompatibles, optimizados para distintos niveles de automatización (Carolina Lagos, 2016). A continuación se presentan los protocolos más usados en equipos industriales.

### 2.8.1 Modelo OSI

En si no es un protocolo es un modelo de interconexión de sistemas abiertos (OSI), sobre el cual se emplea un conjunto de normas de comunicación como herramientas utilizadas, para el diseño, descripción, implementación, estandarización y el uso de redes de comunicación (CCNA, 2015). Bajo un diseño estructural compuesto de siete capas. El modelo permite la interconexión e intercambio, de información de todos los equipos que conforman una red independiente del protocolo y fabricante de dichos equipos. Cada capa cuenta con funciones definidas, sin establecer la forma en cómo sean cumplidas dentro de la capa.

La Figura 2.13 muestra las capas del modelo OSI, creada por ISO. La cual divide responsabilidades haciendo que cada nivel se ocupe de una sola parte de la comunicación que aporte a la solución de problemas de compatibilidad. Cada capa es independiente de las demás, al modificar un protocolo en una capa no se requiere alteración alguna del resto. Por lo tanto cada nivel puede interactuar con su homólogo en otro equipo. Es usado en las redes industriales y facilita a los proveedores elegir el software de comunicación en sus equipos.

- La **capa física**, define los aspectos eléctricos, mecánicos y físicos del medio de comunicación, y tipo de conexión sobre el cual se envía la señal de información.
- La **capa de enlace de datos**, ordena y administra la transferencia de datos, de forma segura entre los nodos y la red, regulando el tráfico (paquetes), acceso a la red, etc.
- La **capa de red**, controla la operación y congestión de la red, definiendo la ruta que el tráfico generado debe seguir para llegar a su destino, conecta nodos que pueden encontrarse en redes distintas convirtiendo la dirección física (MAC) en lógica (IP)



- La **capa de transporte**, asegura la difusión e integridad de los datos, sin errores o alteraciones en su entrega entre los nodos conectados, manteniendo una transmisión e intercambio confiable mediante protocolos de calidad de servicio (QOS).
- La **capa de sesión**, es responsable de la disponibilidad y establecimiento de sesiones de comunicación entre dispositivos en la red, ante un intercambio de datos.
- En si la **capa presentación**, convierte el formato de información de distintos equipos, a un formato común de datos legibles. Para comunicar dos sistemas.
- La **capa de aplicación**, proporciona el vínculo directo con el usuario, mediante programas de aplicación, disponiendo de recursos y servicios de la red usando protocolos como: correo electrónico, base de datos, etc.

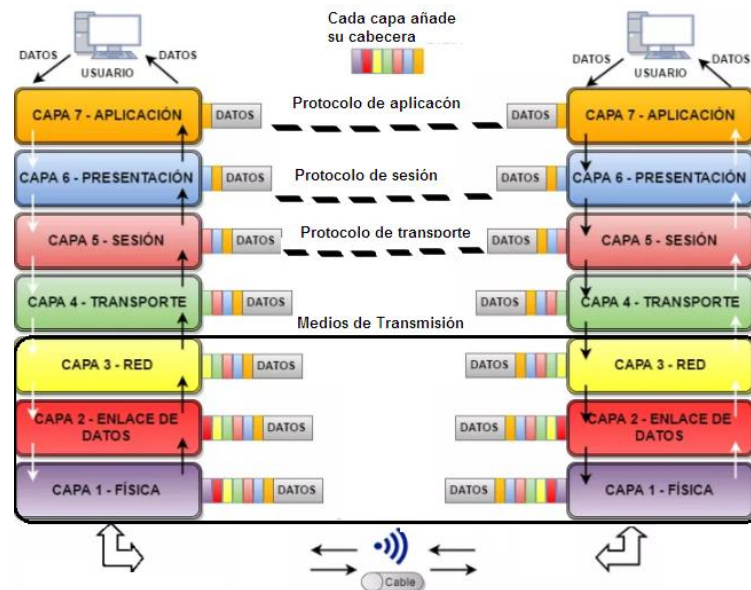


Figura 2.13 Niveles del modelo OSI (Ermesh, 2016) & Autor ().

### 2.8.2 Protocolo profibus (Process Fields Bus)

Profibus es un bus de campo industrial, de estándar abierto obrado bajo normas de comunicación internacional y europea. Brinda soluciones a problemas de conectividad entre equipos del nivel de célula y campo interconectando homogéneamente la estructura funcional de estas áreas de producción industrial, en una arquitectura de red de comunicación unificada. Opera en las capas: física, enlace de datos y aplicación del modelo



OSI (Gabriel Fernandez Perez, 2015). Utilizado en aplicaciones de transmisión de información confiable de alta velocidad, para procesos de automatización y control industrial, con dispositivos PLC, HMI, etc. Está diseñado como protocolo de comunicación unificado, que aloja tecnologías de transmisión complementaria (Karla gaona Hez, 2015). Cuenta con tres variantes, cada una presta un servicio con un tipo de comunicación, enfocados a tareas o dispositivos específicos, entre estos:

- Profibus – PA y Profibus – DP, desarrollados para los dispositivos del nivel de campo (cumplen con tareas de procesos a tiempo real).
- Profibus FMS aplicada en dispositivos del nivel de célula (alto procesamiento y transmisiones de grandes volúmenes de datos).

La red Profibus define dos formas de acceso para la comunicación (Hurtado & Página, 2014); el esclavo – maestro que proporciona el acceso a dispositivos esclavos (Un maestro, varios esclavos), y el paso por testigo (token bus) que brinda el acceso a dispositivos maestros (multimaestro). Soporta comunicaciones serie como la RS485, RS232 y fibra óptica, así como conexiones inalámbricas (Gabriel Fernandez Perez, 2015). La red puede contener un número máximo de 126 estaciones de las cuales 32 son activas (maestro), 32 de segmento, 9 repetidoras y 53 pasivas (esclavos) (Karla gaona Hez, 2015).

### **2.8.3 Modbus**

Es un protocolo de comunicación industrial, utilizado en sistemas de monitoreo y control. Elaborado para enlazar e implementar dispositivos electrónicos del nivel de gestión y de campo, bajo una red de comunicación maestro – esclavo (Xavier Martí Carné, 2015). La red Modbus destina una estructura de mensajes, apoyado en la capa de aplicación del modelo OSI, para el intercambio de información, entre diferentes dispositivos de automatización, sea de punto a punto (un maestro- un esclavo) o difusión (un maestro- varios esclavos) (National Instruments, 2014)& (Collantes & Padilla, 2015).

La Figura 2.14 muestra la metodología de transmisión, mediante solicitud y respuesta entre el maestro y el esclavo respectivamente. Donde el maestro puede seleccionar un dispositivo



conectado a través de una dirección y definir la acción (leer/escribir, control, etc.) a realizar mediante códigos de función (1–255). De forma tal que se encarga de gestionar la comunicación y accionar del esclavo, a efectos de adquisición de datos y control de proceso, reconociendo de que dispositivo proviene la información captada.

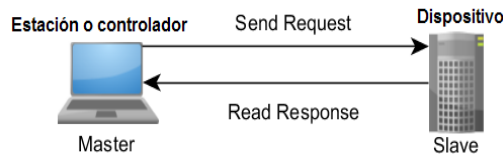


Figura 2.14 Relación maestro-esclavo (National Instruments, 2014).

### Características

- La red Modbus tiene un solo maestro para un máximo de 247 esclavos, asignándoles direcciones específicas en el rango de 1–247, reservando la dirección [0] para difusión sin respuesta. El intercambio de datos puede darse con interfaces RS-485, RS-232, RS-422, fibra óptica y adaptaciones del protocolo Ethernet.
- La comunicación es asíncrona con velocidades de 72 – 19200 baudios. Dispone de un control de errores de los mensajes. Tiene dificultades de seguridad, debido a que fácilmente cualquiera puede modificar los datos y pasarse por el maestro o esclavo.
- Es utilizado en sistemas SCADA, RTU, HMI, PLC entre otros, y los modos de transmisión de datos puede ser de dos formas (Collantes & Padilla, 2015):
  - Modbus – ASCII, utiliza caracteres hexadecimales. Establece el inicio y fin de una trama sin importar el tiempo de transferencia en el canal. Él envió es más lento comparado con RTU por la doble longitud del dato.
  - Modbus –RTU, utiliza comunicaciones seriales con códigos planos, con intervalos de tiempo de carácter silencioso, mejor velocidad de transmisión y densidad de información que ASCII. Pero debe gestionar el tiempo en inicio y cierre de una trama.

### 2.8.4 Hart (Highway Addressable Remote Transducer)

Es un protocolo de comunicación, cuya transmisión de datos lo realiza superponiendo, la señal digital sobre una señal analógica (típicamente de 4 – 20mA) (véase la Figura 2.15), conviviendo ambas sin verse interferidas en el mismo soporte (generalmente par trenzado).

La señal resultante es una onda sinusoidal que cumplen con dos tareas llevar la información y alimentar el dispositivo conectado. Los valores lógicos de la señal digital, están representados por frecuencias de 1200Hz (bit 1) y 2200Hz (bit 0), basadas en la modulación por desplazamiento de frecuencia “FSK”.

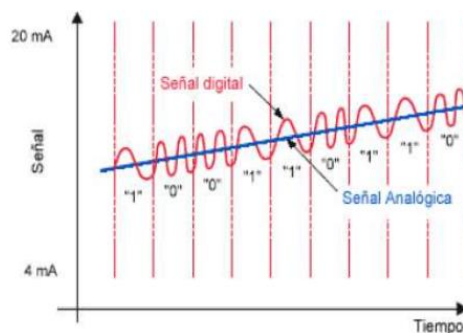


Figura 2.15 Señal Hart (Espinoza, 2014).

En la Figura 2.16 se muestra una red multipunto con el protocolo Hart, incorporando distintos dispositivos de campo (sensor, actuador), conectados directamente sea este un computador o PLC, acondicionados por la comunicación digital maestro/esclavo. El número límite de equipos conectados en esta red es de 15 en un máximo de 2 maestros, con velocidades de enlace de 1200 baudios/seg.

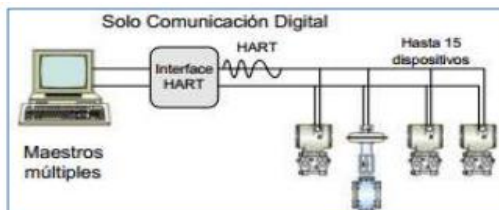


Figura 2.16 Conexión Hart en una red multipunto (Espinoza, 2014).

### 2.8.5 Ethernet

Hoy en día, Ethernet es una tecnología accesible y abierta, sobre todo cuando su estándar está basado en la norma IEEE 802.3. Siendo uno de los más aplicados en las redes



industriales, principalmente en la tecnología de automatización, por sus características de sistema universal, en la conexión de red de área local Ethernet bajo el protocolo de comunicación TCP/IP. Es altamente adaptable a ambientes hostiles, resistente a interferencias electromagnéticas y altas temperaturas, siendo ideales para el nivel de planta (Espinoza, 2014).

Provee una estructura de red flexible y escalable para intercambiar información entre elementos electrónicos (como: PLC, dispositivos de E/S, HMI, entre otros) que pueden ser administrados y monitoreados por un modelo cliente/servidor desde la red. Las tasa de transferencia de esta red comprende el rango de 10Mbps hasta 10 Gbps (Karla Gaona Hez, 2016). El soporte físico puede ser de fibra óptica, cable coaxial, par trenzado. Por lo general su topología es de bus serial, en la cual emplea el protocolo de control de acceso al medio por contienda “CSMA/CD”.

La red industrial Ethernet, ha permitido que algunos protocolos de otras características, incluyan en sus comunicaciones el protocolo TCP/IP, la siguiente Tabla 2.2 muestra los protocolos más comunes.

Tabla 2-2 Protocolos que se diferencian por la velocidad de respuesta, al transmitir datos

Operan a tiempo real	No requieren operar a tiempo real pero son eficaces en la difusión
Ethernet IP	Modbus TCP
ProfiNet	PoweLink
Ethernet Cat	Fieldbus HSE (High-speed-Ethernet)

Fuente: Autor ()

## 2.9. OPC (OLE for Process Control)

La presencia de monopolios tecnológicos, evita la coexistencia de distintas tecnologías en un mismo entorno, confinando a soluciones de un solo proveedor. En este marco los sistemas de automatización, han aquejado por décadas problemas de comunicación, entre

aplicaciones y dispositivos, originados por el uso de protocolos propietarios (Castaño Gómez, Villamarin Meneses, & Rojas Alvarado, 2015).

Ante tal situación surge como alternativa, el estándar de comunicación, en el campo del control y supervisión de procesos industriales (**OPC**) desarrollado bajo la tecnología de Microsoft, fundado y definido por la OPC Foundation, el cual apoyado en una infraestructura de comunicación estandarizada, permite que varias plataformas de hardware y software, de automatización de diferentes fabricantes, se asocien, para poder interactuar e intercambiar información industrial de forma fácil, libre y confiable (Raúl Cobo, 2015).

La Figura 2.17, señala como antes de OPC, el sistema de automatización propietario, establece en cada aplicación su propio controlador (drivers) para realizar la consulta hacia un equipo específico. El estándar elimina el uso de hardware adicional y drivers, fundando todos sus enlaces bajo una interfaz común de transmisión, promoviendo la interoperabilidad entre componentes de automatización, de distintos fabricantes eficientemente.

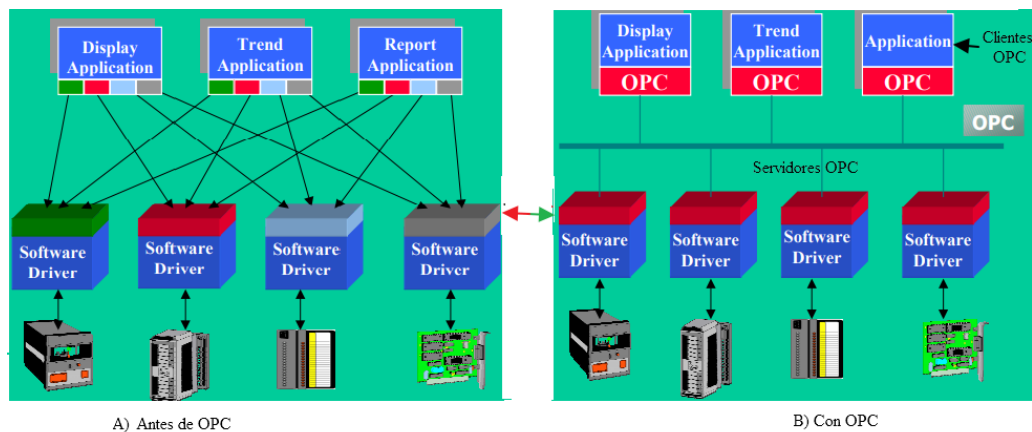


Figura 2.17 Soluciones con OPC (Raúl Cobo, 2015).

### Arquitectura de OPC

Funciona como un bus de comunicación, manteniendo una arquitectura abierta. El protocolo tiene como fin traducir los distintos tipos de datos, de varias fuentes; sean de dispositivos o bases de datos, a formatos estándar OPC. La comunicación se realiza sobre el modelo cliente/servidor, siendo las interfaces del OPC las que entablan los enlaces entre el



cliente – servidor y dispositivo, para el concerniente acceso a los datos. Así el servidor es el encargado de comunicar a los dispositivos con los clientes del sistema OPC, conforme sus necesidades.

El servidor OPC, precisa de un conjunto de interfaces apoyadas en los formatos **COM** (Modelo de objetos de componentes)/**DCOM** (Modelo de objetos de componentes distribuido), que son programadas y activas con el sistema operativo de Microsoft (Windows). Algunos lenguajes de programación como; Visual Basic, C++, Java Script y otros, facilitan el diseño de aplicaciones, cliente y servidor OPC, con interfaces que pueden ser de tipo personalizada o automatizada, dado por OPC. Haciendo que actividades de acceso a datos se torne sencillo y menos complejo para el desarrollador.

### **Servidor OPC**

Es una aplicación de software, utilizada como interfaz de comunicación OPC, la cual cumple con una o con más especificación estipuladas por OPC Foundation (Arencibia, 2016), cubriendo con la difusión de información, únicamente ante solicitudes del cliente.

El servidor utiliza drivers estandarizados para conectarse, con las fuentes de datos (dispositivos; PLC, DCS, etc.). Consiente el acceso y suministro de información de forma transparente al cliente OPC, sobre las variables que ofrece. Admitiendo la comunicación de múltiples fuentes y de clientes simultáneamente.

#### Propiedades del servidor OPC:

- Intercambia información por la interfaz de los objetos COM (enlaces locales) y DCOM (enlaces remotos) de Microsoft. Soportando comunicaciones bidireccionales, entre cliente – servidor para leer o escribir datos, en los dispositivos del sistema OPC sea en forma sincrónica o asíncrona.
- Cada variable (dato físico) es un “ítem” OPC y representa la dirección de conexión con la fuente de datos, dentro del servidor, permitiendo organizar y



contener los datos (ítems) en grupos OPC, que se expone al cliente. También es posible la suscripción, a ítems OPC por parte del cliente (ETI, 2013).

- Se comunica nativamente con los dispositivos, traduciendo los protocolos nativos a protocolos OPC, para la comunicación con el cliente OPC. Acepta protocolos industriales como el serial y Ethernet para la conexión de los dispositivos.

### **Cliente OPC**

Es un fragmente de software, embebido en una aplicación, utilizado para la comunicación con servidores OPC, que utiliza especificaciones de estructuras de mensajería, definidos por la OPC Foundation, que consienten el desarrollo de interfaces para el servidor.

La tarea de cliente OPC es acceder a los datos de las fuentes, traduciendo las peticiones nativas de la aplicación a peticiones OPC equivalentes para el servidor OPC (OPC FOUNDATION, 2014). A fin de que los datos adquiridos sean interpretados adecuadamente, tanto por la aplicación que usa la especificación, como por el servidor que conecta con una fuente específica, así el entorno de la aplicación proporciona el análisis y visualización de datos adquiridos.

Propiedades del cliente OPC:

- Controla e inicia el dialogo (enviando solicitudes.) con cualquier servidor OPC, visible en la red. Pudiendo comunicarse con uno o más servidores OPC compatibles.
- Los clientes OPC son módulos de software usados por aplicaciones como: SCADA, historiadores, HMI, graficadores, etc., haciéndolos compatibles para OPC.
- Configura estructuras de datos en base a ítems, así el cliente dentro de cada grupo puede definir uno o más ítems OPC, accedidos por vía grupos.



## **Tipos de servidores OPC**

Según las especificaciones que define la OPC Foundation, se establecen cuatro tipos de servidores, para diferentes tipos de datos que soporta OPC y que se ajustan a necesidades concretas, como él:

- Servidor OPC DA (Data Access). Brinda un método de acceso común a diversas aplicaciones para diferentes fuentes de consulta. Unifica todos los datos dentro de un grupo OPC. Consta de varias interfaces y conserva un flujo continuo de datos (Arencibia, 2016).
- Servidor OPC HDA (Historical Data Access). Colecciona y permite la elección de Ítems de fuentes, que contienen datos históricos y el tiempo de almacenaje de sus valores. La aplicación solicitante, puede elegir; el tipo de dato y rango de tiempo. Ofrece métodos e interfaces para leer, actualizar, insertar o eliminar datos históricos
- Servidor OPC A&E (Alarms and Event). Provee de mecanismos idóneos para notificar o suscribir al cliente, situaciones de eventos o alarmas que se susciten, en el proceso de control industrial (ETI, 2013). El cliente puede establecer eventos, condiciones y alarmas para conocer el estado del proceso en cada momento.
- Servidor OPC UA (Unified Architecture). Permite la unificación de diversos tipos de datos, integrando múltiples plataformas de control. Adapta las funciones de comunicación de Windows, para usarlo en sus dos tipos de protocolos; orientado a servicios web o el protocolo binario basado en TCP, que es eficiente y optimizado (Alfonso, Noguera, & Valencia, 2015)& (Architecture, 2013).

## **Transición de OPC – UA a OPC clásico**

Inicialmente OPC estuvo hecho para funcionar en sistemas operativos de Windows exclusivamente con la tecnología COM/DCOM, pero este panorama ha cambiado con la nueva versión OPC UA, asumiendo servicios vanguardistas como la tecnología web capaz de comunicarse a través de la red. La cual es compatible absolutamente con la versión clásica de OPC (como: DA, HDA y A&E), y sobre todo goza de un protocolo más abierto



hacia otros sistemas operativos permitiendo soluciones multiplataforma. Del cual se destaca las siguientes características:

- Protocolo de comunicación abierto e independiente que puede funcionar con sistemas operativos Windows, Linux, etc., y sistemas de telefonía móvil como Android, elimina las dificultosas comunicaciones DCOM y las limitaciones, de operación con Windows, permitiendo sistemas interoperables y fácil adaptación al cambio tecnológico, gracias a su interfaz genérica.
- Incorpora mecanismos de seguridad y modelos informáticos orientados a objetos, tipos de datos y métodos, que abarcan todas las funcionalidades tradicionales de OPC clásico, entregando una comunicación más robusta, confiable y de mayor rendimiento para la automatización, aumentando las capacidades de funcionamiento del OPC clásico, a una plataforma de comunicación universal.

### **Ventajas de la tecnología OPC**

- Distintos fabricantes, pueden ofrecer servidores y clientes OPC para el diseño de sistemas interoperables, que se ajustan a diferentes necesidades del usuario.
- El estándar OPC es altamente funcional y versátil, y lo más importante su evolución constante con versiones mejoradas. OPC UA es tolerante a fallas y posee mecanismos de redundancia en la comunicación.
- No necesita re – descripción de drivers, ante cambios en las características del hardware. Pueden utilizar distintas versiones de software OPC sin inconvenientes.
- El apropiado seguimiento del estándar, garantiza la conectividad de cualquier servidor OPC existente en el mercado. Fabricantes de hardware, brindan componentes de software que el usuario puede usar en las aplicaciones, independiente del bus de comunicación y tipo de dato.
- Todas las especificaciones OPC e información pueden ser descargados gratuitamente desde la página web de OPC Foundation para ayudar a los usuarios.



## Capítulo 3

### Diseño y construcción del prototipo SCADA

#### 3.1. Consideraciones del modelo

El estándar OPC presenta las mejores características y ventajas, en la conectividad de aplicaciones industriales, para diseño del sistema SCADA. Es por ello que se propone la implementación de dos tipos de servidores “*OPC UA* y *OPC DA*”, en distintas fuentes de información. Este *modelo de comunicación OPC*, se compone de las siguientes etapas para su adecuado funcionamiento, esquematizadas en la Figura 3.1.

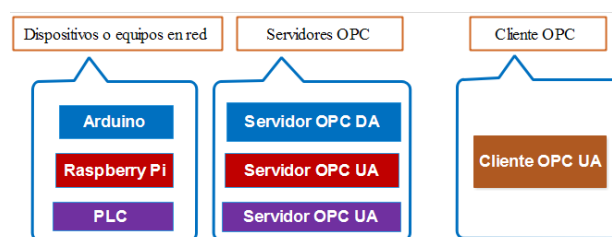


Figura 3.1 Diagrama del sistema de comunicación OPC Autor ()

- **Dispositivos o equipos en red**

La red industrial es el soporte de la comunicación, permitiendo el monitoreo constante de los procesos de automatización ejecutados en planta. Por ende, los dispositivos de control deben, estar conectados a la red para ofrecer al usuario la información generada.

- **Servidores**

A través de la red los servidores OPC UA y DA, acceden a los valores de las variables, que disponen los dispositivos de control (PLC, Arduino, Raspberry), proporcionándolos a un cliente OPC UA. La ventaja de esté junto con el *servidor OPC UA* es que es independiente de la plataforma, en su implementación en cambio el *servidor OPC DA* depende de un sistema de Windows. Por tal razón se utilizará, un servidor OPC UA en la plataforma Raspberry Pi y dos servidores OPC UA y DA, más el *cliente OPC UA* en un ordenador bajo Windows como lo indica el siguiente esquema.

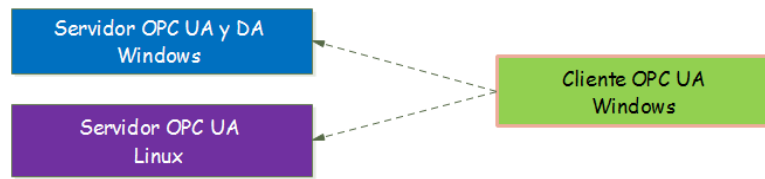


Figura 3.2 Arquitectura OPC multiplataforma Autor ().

- **Cliente OPCUA**

Es el objeto que recibe los datos de los servidores OPC. El cliente mantiene el mismo protocolo de comunicación compatible para ambas especificaciones. La ventaja de esto, es que el usuario no precisa de un cliente en específico en su utilización, puede elegir cualquier opción, sea comercial o libre. En esta ocasión se escogido un cliente OPC UA como aplicación a través de la *plataforma Ignition*.

### 3.1.1 Requerimientos del sistema de control automático

En base a lo expuesto la presente investigación, propone la implementación de un sistema de comunicación OPC básico, en la plataforma industrial Ignition, que permita el desarrollo de aplicaciones de control y supervisión rentables. Disponiendo, en una sola plataforma la combinación de diferentes tecnologías de dispositivos de automatización, para distintas funciones operativas de un proceso, como fuentes de consulta de información.

La propuesta planteada nace con el fin de buscar mediante OPC, aplicar alternativas tecnológicas de bajo costo con plataformas industriales reales, que optimicen y hagan accesible el diseño de sistemas de automatización industrial. El prototipo citado, utiliza un servidor OPC DA y un servidor OPC UA, desplegados para las opciones de control Arduino y Raspberry Pi respectivamente, y un servidor OPC UA estandarizado para PLC's incorporado por la plataforma Ignition.

La comunicación se basa en el sistema cliente – servidor, estando el SCADA a cargo de la gestión y funcionamiento de los procesos respectivos del modelo. El análisis y el diseño del sistema OPC SCADA con la plataforma Ignition, se dedicarán en los apartados posteriores, por lo pronto se explicará ciertos aspectos del prototipo propuesto.



### 3.2. Diseño de la infraestructura de red

El esquema presentado para el desarrollo de este proyecto, tiene por objeto generar una arquitectura unificada e interoperable, que propicie la expansión de las funcionalidades de sistemas de automatización, cimentadas en las especificaciones OPC, que establecen los servidores, que integran y consolidan equipos industriales y sistemas embebidos, que se comunican mediante la red con un sistema de control y supervisión centralizado que se encuentra alojado en la nube. A fin de interconectar, los niveles 1 y 2 del modelo CIM, para compartir y solicitar abiertamente información, del proceso de automatización.

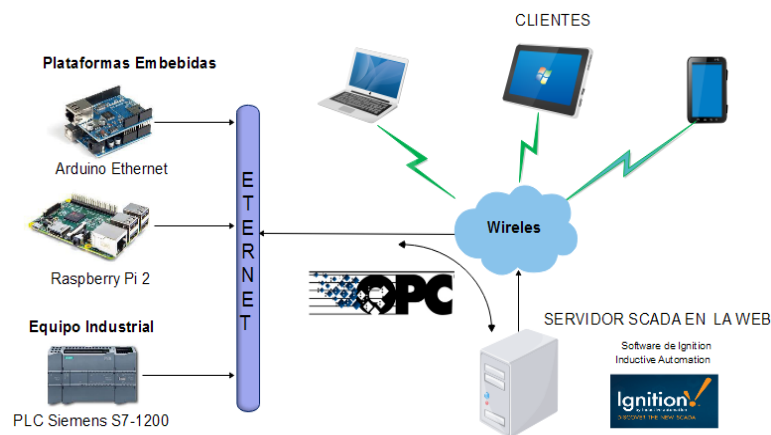


Figura 3.3 Topología del sistema de supervisión y control híbrido Autor ().

Con el afán de que la infraestructura de la red, sea versátil, se realizará la implementación de los servidores OPC y demás elementos del sistema dentro de una *red de área local* (LAN), para la interacción directa con la aplicación de supervisión y control destinada como cliente OPC. Esto es posible siempre y cuando los equipos que conforman el sistema tengan puertos de red y puedan ser direccionados con una dirección única a través de la red, forjando un sistema remoto, cuando el cliente y servidor se encuentran distantes, permitiendo a cualquier operario la administración en tiempo real, de los implementos del sistema.

La arquitectura vista en la Figura 3.3, refleja al sistema OPC SCADA como una estación de admisión web de clientes, gracias a las facultades de servicios web y soporte OPC que reúne la plataforma Ignition, han hecho posible expandir las funcionalidades del sistema OPC propuesto, para que sea gestionable remotamente (*vía red LAN*). Por ello tanto el



servidor Web SCADA Ignition, como los servidores OPC y unidades de control, se han de conectar sobre un mismo medio de transmisión, para que tenga lugar la comunicación OPC.

### 3.2.1 Comunicación de implementos del sistema con los servidores OPC

Los servidores OPC son los encargados de construir las interfaces, para llegar hacia dispositivos de control del proceso y sobre los cuales el usuario, pueda definir las acciones de control o adquisición de datos de los dispositivos conectados, a través del SCADA Ignition. Por ello la adecuada interacción e implementación del prototipo de sistema OPC presentado, recae en las siguientes consideraciones:

#### ■ Comunicación PLC – Servidor OPC UA

La pretensión del servidor OPC UA es manipular las variables internas del PLC. Para llevar a cabo este propósito, se debe crear la interfaz de comunicación del PLC con la plataforma SCADA. Dado que estos son configurados con una *dirección IP* predeterminada. La IP constituye el nexa para identificar, dispositivos que se encuentran conectados en la red y deben estar en el mismo dominio, para evitar incompatibilidades.

La IP de un PLC, está definida por lo general en el rango 192.168.0.1/255, estipulado para las configuraciones. La Figura 3.4 expone la comunicación entre el PLC y el sistema SCADA. Siendo el *servidor OPC UA de Ignition*, el driver de interconexión de estas plataformas.

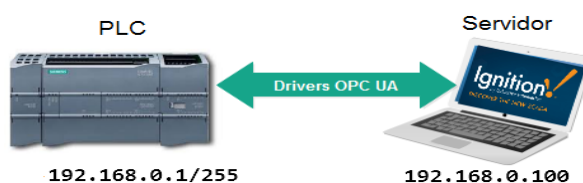


Figura 3.4 Esquema del modelo de comunicación OPC UA del PLC con el servidor de Ignition Autor ().

#### ■ Comunicación Arduino – Servidor OPC DA

El servidor OPC DA tiene como función, ingresar a los registros dentro del Arduino, aquello conlleva a que las operaciones internas de la lógica del programa deban contener, los *ítems OPC* a ser exhibidos al servidor, el cual toma los ítems y los pone a consideración



del sistema Ignition para comunicarse con la aplicación OPC SCADA. Para esto previamente es necesario configurar a nivel de software la interfaz de comunicación del medio OPC, para lograr afianzar los enlaces entre la plataforma Arduino y el servidor, cuya propiedad de red está dada por la *Shield Ethernet*, en la cual se configura y establece la IP para el enlace con la red local.

Uno de los requisitos, para mantener el correcto desempeño del servidor OPC DA, es asegurar su registro en el sistema operativo de Windows. Por razones naturales de la especificación y del propio sistema al utilizar los formatos *COM/DCOM*, que crean las *interfaces OPC DA* para este tipo de servidor OPC. La Figura 3.5 muestra el bosquejo general de la comunicación entre ambas plataformas. Siendo el *módulo OPC* que añade el servidor y la *librería OPC* de Arduino, los encargados de facilitar este propositivo.

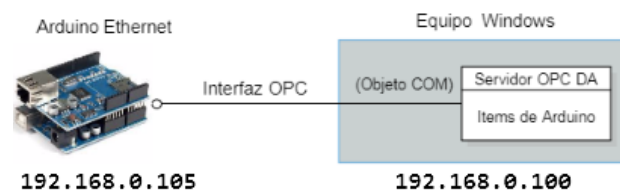


Figura 3.5 Diagrama de comunicación OPC entre Arduino Ethernet y Servidor OPC DA Autor ().

### ■ Comunicación servidor OPC DA – SCADA

La comunicación OPC entre el sistema Ignition y el servidor OPC DA, está basada en el protocolo COM/DCOM. Al operar en entornos de **Windows** en ocasiones es pertinente, realizar las configuraciones **DCOM** del servidor OPC DA, a manera de cumplir con las credenciales del sistema para lograr una interacción eficaz con la plataforma Ignition. Donde el usuario añade excepciones y permita el puerto, *TCP 135 DCOM* definido por el protocolo OPC DA, para comunicarse por la red, al mismo tiempo de ser recomendable de que deban mantenerse instalados los componentes OPC, necesarios para los enlaces a efecto de evitar bloqueos en la conexión por parte del **Firewall**.

### ■ Comunicación Raspberry Pi – SCADA

Las prestaciones de Raspberry le permiten vincularse con el OPC SCADA, a través de su puerto de comunicación de red. Para ello se elabora un servidor **OPC UA**, como driver de

interconexión con el *sistema OPC UA de Ignition*. Al trabajar con el servidor como ruta de enlace con el SCADA, debe comprenderse aspectos de direccionamiento IP y protocolos OPC UA entre ambos, para crear la interfaz de comunicación. La Figura 3.6 muestra la conexión entre Raspberry y el SCADA. Estando a cargo de la comunicación el agente OPC UA de Python.

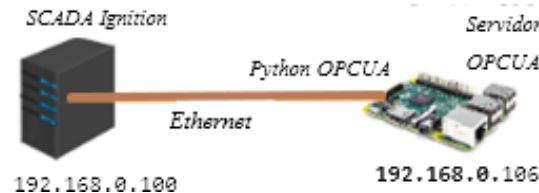


Figura 3.6 Esquema de comunicación OPC UA de Raspberry Pi, con Ignition Autor ().

### ■ Comunicación Cliente – Servidores

Esta comunicación tiene como empeño ofrecer al usuario, toda información acerca de sensores, motores, actuadores, etc., que están conectados a las unidades de automatización. Sea el sistema de control automático, residente en la nube el intermediario hacia dichas unidades asumiendo la propiedad de cliente OPC UA, que actúa desde un sitio Web.

La ventaja de este tipo de conexión, es el enlace con múltiples usuarios de forma inalámbrica aceptando el servidor diferentes peticiones simultáneamente. La única condición es que los usuarios y el servidor, tengan una dirección IP definida dentro del dominio de la red local. Los usuarios para el acceso deben complementar un soporte **JAVA**, entre las funciones de comunicación, que apruebe el enlace vía IP mediante la URL del navegador web. Esto por las facultades Web del *servidor Ignition*, admitiendo con ello la ejecución de la aplicación SCADA en distintos entornos y desde cualquier consola.

La siguiente representación muestra el enlace e interacción de varios usuarios con la aplicación OPC UA Web, en el servidor SCADA a través de la red local acceden a las unidades de control del sistema, atravesando los servidores OPC. Esta será la topología a seguir para el desarrollo del sistema OPC SCADA Ignition.

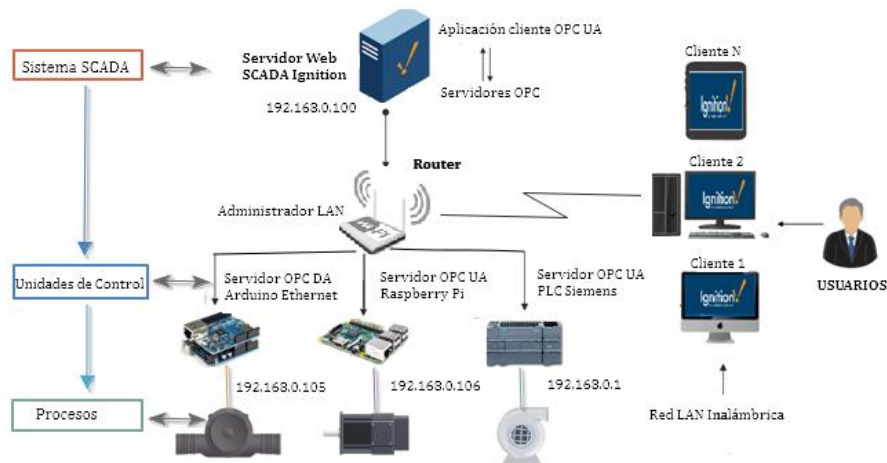


Figura 3.7 Sistema de control automático, comunicación remota cliente – servidor OPC en Ignition Autor ().

### 3.3. Implementación y configuración de componentes del sistema OPC

Se ha visto oportuno en este prototipo, optar por unidades de automatización, que dispongan de puertos Ethernet. Esto con idea de plasmar una red que sea firme y homogénea en el que los datos estén regidos, por un mismo mecanismo de empaquetado (como el protocolo TCP/IP), en su enrutamiento a través de la red LAN. Para que con ello se logre facilitar el direccionamiento de la información a la ubicación del servidor, donde reside la instalación del sistema Web SCADA Ignition. De tal forma que la comunicación de red este bajo la mención OPC *TCP/IP* que encapsulará las tramas de información a formatos legibles, para el servidor SCADA hospedado en la nube.

A continuación se sustenta a detalle en los siguientes puntos, las características y configuraciones necesarias a nivel de hardware y software, para el acoplamiento del protocolo OPC en las unidades de control escogidas. Como fase preliminar de adaptación en los implementos del sistema OPC, para la comunicación con la plataforma Ignition haciendo uso de herramientas compatibles con el estándar.

#### 3.3.1 Implementación Arduino Shield Ethernet

La **Shield Ethernet W5100** es una tarjeta electrónica que añade la funcionalidad de red por el conector Ethernet estándar RJ45, permitiendo la comunicación de Arduino en la red LAN interna. Esta se ha elegido por las bondades que ofrece, al facilitar las configuraciones

de red y de poder adecuar el protocolo de comunicación *TCP/IP* para hacerlo apto a vínculos OPC. Siendo la Shield la que implementa la *pila de protocolos TCP/IP*, cuya capa de aplicación se realiza dentro de Arduino mediante *librerías*. En la Figura 3.8, se explica el montaje y conexión de la Shield Ethernet con Arduino.

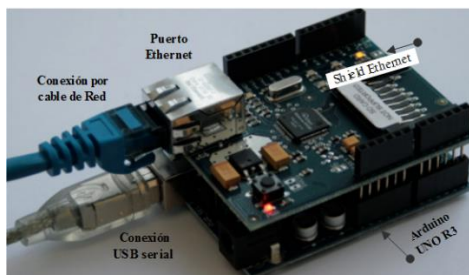


Figura 3.8 Montaje Arduino Ethernet Autor ().

### 3.3.1.1. Características e instalación de librería OPC Arduino

La librería OPC, confiere el modo de soporte OPC entre las funciones de comunicación de Arduino, cuyas señales de datos serán utilizadas por el servidor OPC DA para disponer de la interfaz de comunicación correspondiente, en forma fácil y unificada por los puertos de conexión serial y Ethernet por los cuales intercambiara información.

Como parte de las configuraciones del sistema para los enlaces, la librería adicionalmente dispone de formas seriales o http, para publicar los ítems al servidor OPC DA, respecto a las variables del proceso. La librería puede obtenerse de la página *GitHub* o en su defecto desde el *IDE* de Arduino. En esta aplicación se hará uso de la versión **Arduino 1.6.5.**, y la librería *OPC 2.0.0* elegida como soporte de comunicación entre el servidor OPC DA y el SCADA. La Figura 3.9 muestra la librería OPC obtenida desde el gestor de librerías.

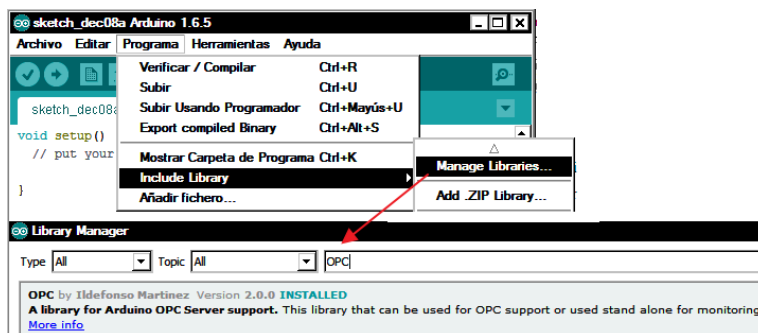


Figura 3.9 Librería (`#include<OPC.h>`) OPC desde el gestor de librerías Arduino Autor ().



### 3.3.2 Instalación del sistema operativo (Raspberry Pi)

La plataforma para su funcionamiento, requiere de un sistema operativo (S.O). Conforme a las capacidades de *Raspberry Pi 2*, en esta ocasión ha sido conveniente implementar el **Raspbian** de Linux, su entorno versátil y funciones multipropósito lo hacen ideal para montar el servidor OPC UA propuesto. La imagen ISO del sistema está disponible en el sitio oficial de Raspberry Pi. Se recomienda utilizar el asistente de instalación del S.O de Raspberry “**NOOBS**” como medio que facilite el proceso. Enseguida se describen los pasos para la instalación del sistema operativo Raspbian:

1. Se requiere una micro SD de al menos 4 GB, formateada en el formato *FAT 32*. El NOOBS se inicia automáticamente, previamente se deberá configurar una conexión a internet para descargar el sistema operativo (véase la Figura 3.10).

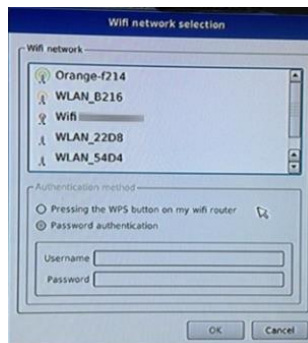


Figura 3.10 Conexión de internet en NOOBS Autor ().

2. Luego se muestra una lista de sistemas operativos, se seleccionará el **Raspbian** y se presionará en **Install** a partir de aquí el proceso es automático, terminada la instalación, ocasionalmente se necesita reiniciar manual el Raspberry para que el sistema quede totalmente funcional (véase la Figura 3.11).

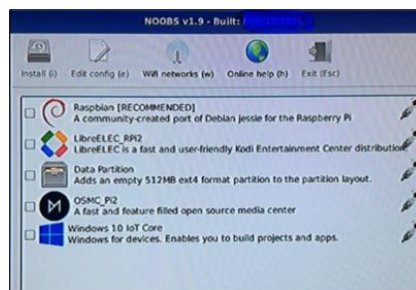


Figura 3.11 Elección de Raspbian Autor ().



Con esto puede utilizarse el S.O. Dando paso a la instalación del lenguaje de programación del servidor OPC, en el cual se construirá el boceto del protocolo OPC UA, y se dispondrán de las configuraciones de red y puertos de comunicación, cuyo programa servirá de pasarela a Raspberry para la correspondiente comunicación OPC UA con el sistema OPC de Ignition.

### 3.3.2.1. Entorno Python

Python es uno de los lenguajes de programación multipropósito, más predilectos actualmente. Permite el desarrollo de aplicaciones en diversas áreas y sobre todo para varias plataformas. El lenguaje de alto nivel que ostenta es simple y efectivo, orientado a objetos. El intérprete de Python es un entorno manipulado por línea de comandos, permitiendo la fácil introducción y ejecución de sentencias. Tiene la colaboración de varios desarrolladores de software disponiendo de una extensa librería y módulos estandarizados.

Para el desarrollo del servidor OPC UA, se ha visto conveniente utilizar la versión **Python 3.4**, y la librería **Freeopcua 0.10.0**. La librería es Open Source y se distribuye bajo la licencia pública general reducida “GLBP” disponible desde GitHub, la cual contiene los métodos que permiten la creación de los objetos OPC UA y la comunicación por Ethernet de Raspberry. También se incorpora una librería para el manejo de los puertos GPIO.

### 3.3.3 Propiedades del PLC

En esta oportunidad se ha escogido el **PLC S7 1200** de la marca **Siemens**, en el cual se realizará la programación mediante el software de *TIA Portal* y se analizará brevemente algunos parámetros de comunicación, así como también las áreas de memoria internas, para la manipulación de los datos, con el sistema OPC de Ignition.

#### Características del PLC Siemens S7-1200:

- El PLC dispone de una CPU que incorpora un puerto Ethernet, para conexiones con cualquier red LAN dándole así, acceso al servidor Web SCADA.



- De fábrica la interfaz viene integrada con un controlador *PROFINET* para comunicaciones de red PROFINET. Sin embargo puede configurarse el servidor OPC UA de Ignition para admitir conexiones nativas de este tipo.
- La CPU se encarga de estructurar y direccionar, zonas de memoria del PLC en direcciones de registro, ofreciendo opciones de almacenamiento de datos en la ejecución de un programa. Vital para establecer con exactitud la región de memoria de donde se obtiene el valor de una variable, esto sea tomando en cuenta para el respectivo acceso del servidor OPC UA.

La Tabla 3.1 y 3.2 (M. Moreno, 2015)& (Cpu, 2013). Se muestra los diferentes tipos de datos y áreas de memoria de un PLC con su descripción ante determinadas funciones.

Tabla 3-1 Longitud del dato a leer/escribir en un PLC.

Tipo de dato	Sintaxis	Tamaño
Bit (Booleano)	<b>X</b>	1- bit
Byte (Unsigned Byte)	<b>B</b>	1-byte
Word (Unsigned Byte)	<b>W</b>	2-byte
Int (Signed Word)	<b>INT</b>	2-byte
DWord (UnsignedDouble Word)	<b>D</b>	4-byte
Real (Real Number)	<b>REAL</b>	4-byte
Long Real Number	<b>LREAL</b>	8-byte
Signed Byte	<b>CHAR</b>	1-byte
Date and Time	<b>DTL</b>	12-byte

Tabla 3-2 Área de memoria de un PLC.

Área de memoria (Nomenclatura)	Descripción	Valor hexadecimal del área
I	Memoria de la imagen del proceso de las entradas provee datos digitales de entradas físicas.	0x81
Q	Memoria de imagen del proceso de salida, provee datos digitales de las salidas físicas.	0x82
M	Memoria de marcas usada para guardar funciones de control y memoria de datos, transfiere en memoria o etiqueta tanto para lectura - escritura.	0x83
T	Memoria de temporizador	0x1C
C	Memoria de contadores	0x1D
DB	Memoria de bloque de datos, utilizada para almacenar todo tipo de dato.	0x84

### 3.3.3.1. Acceso al área de memoria del CPU S71200

Con estas consideraciones, el direccionamiento hacia un área de la memoria interna del controlador, para extraer a los datos se compondrá del formato **[Identificador del área]** **[tamaño]****[dirección del byte]**. **[dirección de bit]**, con el cual se puede direccionar a cualquier posición de la memoria del PLC, incluso acceder a otras áreas con otros formatos de valores para obtener información basada en el formato elegido. Este direccionamiento se explica y representa mejor con la siguiente ilustración.

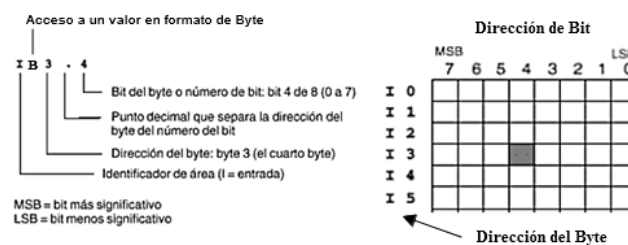


Figura 3.12 Direccionamiento al área de memoria entradas (I) en formato Byte en el PLC (Cpu, 2013)



En el caso de las entradas analógicas [AI], el direccionamiento se realiza siempre con formatos de Word de bytes pares (AIW0, AIW2, etc.), esto por la conversión de un valor analógico a digital. La Figura 3.13 muestra y explica el acceso hacia una entrada analógica.



Figura 3.13 Direccionamiento a una AI en la CPU (Cpu, 2013).

### 3.4. Red Lan OPC

Cabe recalcar que el router es el dispositivo encargado de establecer la red LAN y de proveer conectividad de extremo a extremo, de todos los elementos que conformen el entorno del sistema OPC, fijando direcciones IP automáticamente a los hosts conectados mediante el sistema *DHCP* (*Dynamic Host Control Protocol*). El cual contiene el rango de direcciones que definen el dominio de la red local.

### 3.5. Diseño y configuración de las interfaces OPC

Para crear las interfaces, las unidades de control deben definir dentro de su estructura de programación las variables que presiden su operación, estas se encontrarán almacenadas dentro un registro que define un área de la memoria en concreto. Sobre esto la función del servidor OPC, es obtener las variables que están programadas en las unidades, mediante las cuales se creen las interfaces que gobiernan su operación, admitiendo que el SCADA OPC pueda leer y escribir en dichas variables a través de los servidores OPC.

#### 3.5.1 Interfaz Arduino Ethernet – Servidor OPC DA

Para poder comunicar el Arduino Ethernet con el servidor OPC DA, es necesario construir la interfaz de comunicación. Esta se desarrolla mediante una estructura de programación utilizando las librerías OPC y Ethernet, que instauran los procedimientos de conexión a elaborarse en el IDE de Arduino, sobre los cuales se basa la configuración del servidor, para mostrar las variables creadas al cliente OPC UA Ignition. A continuación, se verán los métodos más importantes.



### 3.5.1.1. Establecimiento de conexión de red

Como primer paso para la administración y uso de Arduino Ethernet. Se incurren a algunos de los métodos de la librería Ethernet conformados en clases, siendo usados para entablar la comunicación con la red local mediante la conexión Ethernet, por la cual se transferirá los datos. Para eso se congregan las librerías OPC, SPI y Ethernet para la operación del montaje.

Es relevante considerar que con frecuencia los pines digitales del 10 al 13, son reservados para SPI y no puede usarse como E/S genéricas. Esto dado por la comunicación serial, de la placa base “Arduino” al transmitir los datos a la Shields, para ser enviados por la red. Se pasa a ver las funciones, de los métodos de configuración IP de la red:

- ❖ *IPAddressClass*: Esta clase emplea el método *IPAddress ()*, con el cual se puede configurar la dirección IP y máscara de red que tendrá Arduino Ethernet, como también las imprescindibles configuraciones de red.
  
- ❖ *Const int listen\_port*. Esta sentencia permite al montaje, admitir la petición de clientes OPC DA. La unidad actúa como un servidor que escucha (conjuntamente con el servidor OPC DA) por un puerto de red las peticiones, por defecto esta se define para el puerto 80.

La librería OPC crea los elementos OPC, que representan las variables programadas que servirán al servidor OPC DA, para acceder y manipular la información que transfieren los procesos que contenga. Ofrece algunos métodos para publicar esas variables, los cuales crean los **objetos OPC** que posteriormente, serán las interfaces COM hacia el servidor. Se presenta un breve análisis de los métodos, usados para la extensión OPC de Arduino.

### 3.5.1.2. Creación de Items OPC

Entiéndase que un elemento, sea este un puerto de entrada o salida real o proceso virtual realizado en Arduino, puede ser declarado como Ítems para ello existen algunos objetos que pueden usarse, para atribuir esta dotación, como pilares del protocolo tenemos:



- ❖ **OPCEthernet**. Este método permite declarar un objeto OPC, definiendo a la interfaz Ethernet como el medio OPC, de intercambio de los datos admitiendo el canal la conexión y mando del servidor OPC DA desarrollado.
- ❖ **Callback ()**. La función se confina en un bucle, permitiendo la declaración y creación de los *Ítems OPC*, reflejando los valores de los procesos internos de Arduino, a ser publicados al cliente OPC DA para su manejo. La función *callback*, utiliza varios objetos en la construcción de Items como: *constchar\*ítemID*, *constopc*, *OperationopcOP* y *value*. Dentro de la función se realiza las operaciones que definen el comportamiento de la variable, bajo el formato OPC aclarando el tipo de dato que se va procesar.
- ❖ **opc\_opwrite, opc\_write, opc\_read y opc\_readwrite, opcOP**. Estos métodos son utilizados con los puertos y permiten que el valor de la variable, que asume un puerto pueda modificar su estado o valor mediante comandos escritos por el cliente OPC DA (o también llamado COM), esto previo a la condición y permisos que enmarque el proceso para llevar a cabo una acción.
- ❖ **Serial.begin ()**. Esta función permite iniciar una conexión serial, para la comunicación con el Arduino, debido a que el servidor envía el flujo de datos a través del puerto USB.
- ❖ **NombreProcesoOPC.setup ()**. Este es un método que permiten inicializar el objeto OPC declarado en las instancias iniciales del programa, con ello la variable determinada en el boceto ya puede utilizarse como un objeto OPC.
- ❖ **AddItem ()**. Este método expone, la variable como OPC Ítem, que servirá de enlace al cliente OPC COM con el servidor OPC DA. El código define la información que va a contener el elemento OPC, dando a conocer el nombre de la variable, los premisos de acción que le dicen a la biblioteca si lee, escribe o ambos en el elemento, el tipo de dato y la devolución de la llamada “*callback*”, cuando el cliente decida leer o escribir un valor, que será



almacenado en el elemento, para corresponder una acción en el Arduino por medio del servidor que anunciará y atenderá dicho Ítems OPC.

- ❖ *processOPC COMmands* (). Este método es usado para inicializar la comunicación y todos los parámetros OPC, lanzado el Ítem OPC mediante los cuales el cliente podrá suscribirse para acceder a la información, con la correspondiente actualización de valores incrustados en los elementos, consultando a las variables declaradas en el boceto del programa, ejecutándose cíclicamente dentro de un bucle principal “*voidloop ()*”.

### 3.6. Implementación del Servidor OPC DA

Se procedió a instalar el servidor OPC en un ordenador portátil HP que alude una versión de 64bit, de 4GB de RAM, con sistema operativo Windows 7 Professional 6.1. El sitio *st4makers* sujeta varias versiones de servidores, que comunican Arduino bajo enlaces OPC DA por lo cual se debe escoger una versión compatible con la Shield Ethernet, para esta ocasión ha sido preferible la versión **OPC Server1.7** que será la pasarela de comunicación de la plataforma, esto se puede observar en la Figura 3.14.

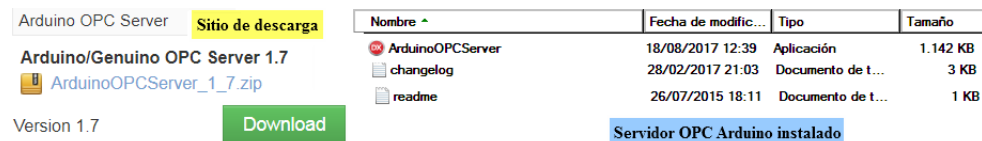


Figura 3.14 Descarga e instalación del servidor OPC DA en Windows Autor ().

Inherentemente se vuelve necesario, para culminar con la incorporación y permitir la comunicación y a la vez el correcto funcionamiento del servidor en el sistema Windows, instalar un módulo OPC llamado *OPC Core Components Redistributable*, que es indispensable cuando se utiliza protocolos OPC clásicos, por el referente objeto COM que contiene, implícitamente estos componentes serán utilizados por el sistema Ignition. El módulo está disponible en la página oficial de *OPC Foundation* (véase la Figura 3.15).

*“Con respecto al sistema operativo del ArduinoOPCServer se ha visto necesario, para compensar y hacer compatible la arquitectura del entorno de operación. Instalar en el*



mismo equipo el servidor Ignition, esto es debido a que el sistema del servidor OPC DA solo funciona y se comunica bajo el componente COM de Windows”.



Figura 3.15 Instalación del Módulo OPC Core Autor ().

A modo de asegurar que la instalación haya sido correcta. La Figura 3.16 señala la forma de cómo verificar que el complemento OPC, haya habilitado el identificador de servidores OPC denotado *OpcEnum*, quien se ejecuta como servicio y proceso en el sistema operativo, el cual se visualiza mediante el administrador de tareas de Windows.

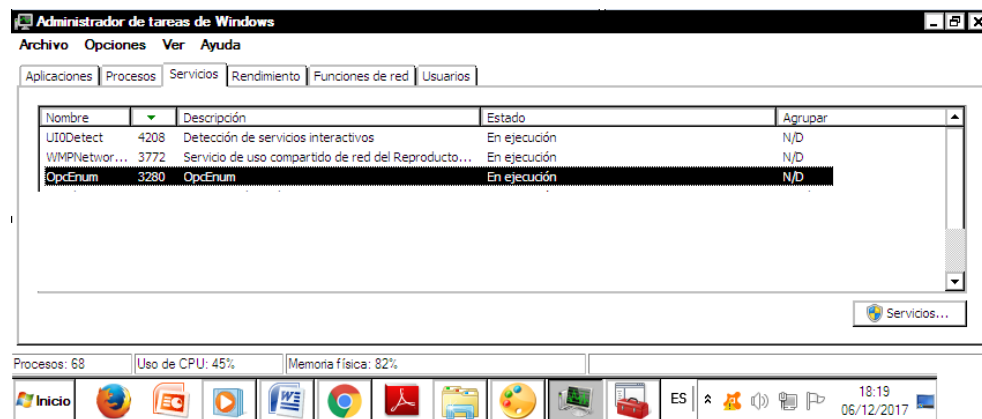


Figura 3.16 Habilitación de OpcEnum Autor ().

En cuanto al módulo OPC Core toma importancia al ser un complemento que usan las interfaces dentro del sistema operativo permitiendo, al servidor OPC Arduino instalado, ser detectado y enumerado por *OpcEnum* haciéndolo perceptible al SCADA Ignition, para realizar el enlace. De esta manera se consolida el protocolo OPC DA mediante COM.

### 3.7. Conexión servidor OPC DA

Definido e incorporado el protocolo OPC en Arduino, mediante librerías, el montaje podrá convertir las variables del proceso a Items OPC, a través de las cuales exhiba y proporcione datos en tiempo real, recuperando información que enviará desde el servidor OPC DA al sistema industrial Ignition, que mediante una aplicación SCADA dedicada como cliente OPC suscribirá y contendrá la información. Para ello hay que configurar y definir los parámetros necesarios, para la puesta en marcha del *OPCServer*, que habría que realizar indicándole al servidor con qué dispositivo de Arduino va a comunicarse consecuentemente está se efectuara mediante comunicaciones Ethernet indicados en los siguientes pasos.

1. En la Figura 3.17 se aprecia la creación de la instancia OPC Arduino con la aplicación portable *ArduinoOPCServer*. La interfaz de configuración, permite escribir la dirección IP y el puerto que asocia la dirección, por el cual transmite los datos Arduino Ethernet, que sin duda debe ser el mismo a lo declarado en el programa. Finalmente debe guardarse para activar la conexión.

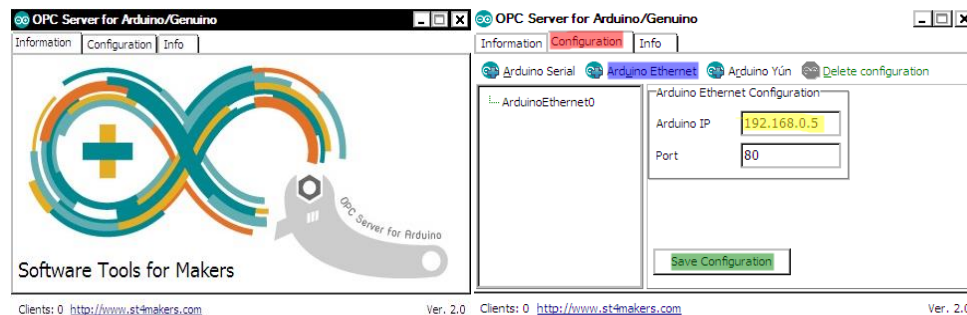


Figura 3.17 Entorno y configuración de Arduino Ethernet con el OPCServer Autor ().

2. A raíz de esta configuración se crean dos archivos. bat, el *register* y *unregister*, los cuales habilitan y deshabilitan la ejecución del servidor respectivamente. Para eso deberá ejecutárselos con *permisos de administrador*. Acto seguido se instala el servidor en el sistema de Windows, cuando se ejecuta por primera vez. Como también se crea un archivo, que contiene información de la configuración realizada, que usa para enlazarse con las variables OPC de Arduino. (véase la Figura 3.18).



Nombre ^	Fecha de modific...	Tipo	Tamaño
ArduinoOPCServer	18/08/2017 12:39	Aplicación	1.142 KB
changeloc	28/02/2017 21:03	Documento de texto	3 KB
readme	26/07/2015 18:11	Documento de texto	1 KB
register		Archivo por lotes de Windows	1 KB
unregister		Archivo por lotes de Windows	1 KB

Figura 3.18 Ejecución del servidor OPC Autor ().

- Lo anterior anotará en el registro del sistema toda, información del servidor sobre su localización en disco duro, así cualquier cliente OPC DA puede preguntar por el servidor para intercambiar información con los dispositivos que asocie. Esto puede ser constatado desde el editor de registro, para su ejecución rápida puede emplear el comando *regedit* (véase la Figura 3.19).

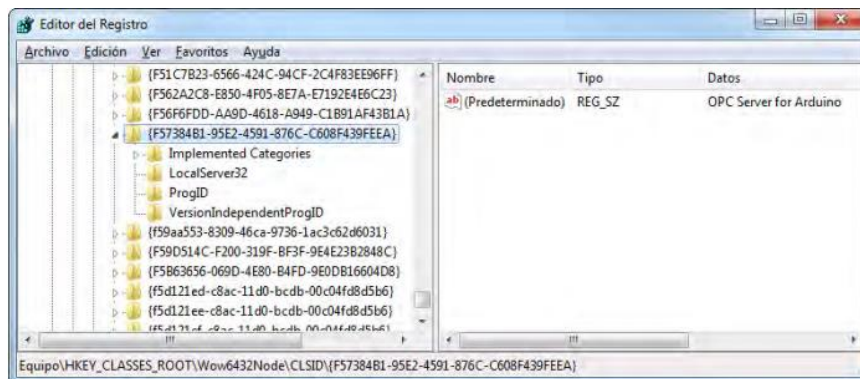


Figura 3.19 Registro del servidor OPC Arduino Autor ().

Con esto el servidor está preparado para comunicarse con cualquier cliente COM, el cual contiene y expone las variables OPC programadas en Arduino Ethernet. Pero por la pertinencia del cliente OPC COM Ignition, se pasa ahora a configurar los derechos y privilegios en el sistema, con el *Firewall* para que se faculte el enlace con el servidor.

### 3.7.1 Configuración DCOM

Relativamente al trabajar con el protocolo OPC DA, se debe conferir credenciales de permiso de seguridad COM en el sistema operativo, para comunicar procesos. El ambiente de Windows facilita el uso de varios recursos para lograr la comunicación entre el *OPCServer* para Arduino Ethernet y la plataforma *Ignition*. La conexión de estos procesos

se lleva a cabo por la razón de que Ignition se ejecuta como un servicio en la cuenta del sistema, que aprovecha para enlazarse con cualquier proceso expectante en la misma línea.

Por eso la configuración **DCOM** del servidor OPC de Arduino, se realiza con la intención de *validar y autenticar la comunicación*, haciendo que ambos componentes de software (*Ignition y servidor OPC Arduino*) apliquen las mismas credenciales para aceptar el enlace.

Consecuentemente se indica el proceso de configuración, de permisos abriendo como primera acción los *servicios de componentes*, de Windows donde se hallan aglutinadas todas las aplicaciones que usan el método COM para su comunicación con el sistema operativo:

1. Dentro de la carpeta *equipos/Mi PC*, ubíquese con un clic derecho en propiedades sobre esta ventana (véase Figura 3.20), diríjase a *seguridad COM*, en los cuatro campos visualizados se agregan en cada uno, las cuentas de *Todos* y *Anonymous Logon* y activase en “*Permitir*” todas las secciones de permisos allí observados, con ello se admiten los *permisos de acceso* y *permisos de inicio y activación*, logrando conformar el canal de enlace **COM** para **OPC DA** .

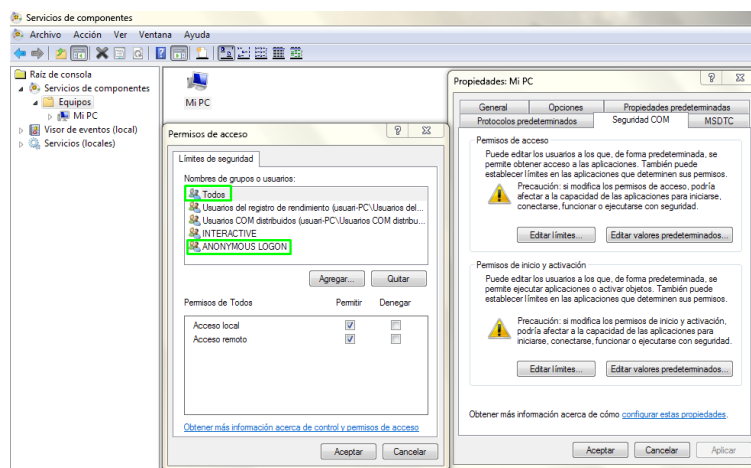


Figura 3.20 Validación de seguridad COM para el OPCServer Autor ().

2. Luego de esto en la pestaña *Propiedades predeterminadas* actívese el recuadro *Habilitar COM distribuido en este equipo*, finalmente acepte las configuraciones. Con estos permisos se validan las credenciales para ejecutar el **ArduinoOPCServer**



en el sistema operativo, de manera que el sistema Ignition pueda comunicarse, evitando así bloqueos por parte del sistema de seguridad al admitir un proceso, como lo indica la Figura 3.21.

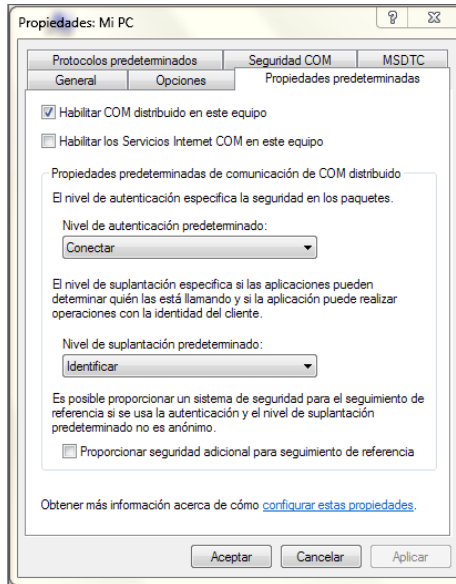


Figura 3.21 Habilitación de la configuración DCOM para que el “firewall” admita el enlace Autor ().

- Adheridos los permisos anteriores (véase la Figura 3.22), en la configuración DCOM se procede a la búsqueda del servidor OPC y en la entrada del mismo elegir propiedades. En este sitio se dará rienda al proceso de privilegios y verificación de las cuentas creadas para otorgar los permisos de seguridad esperados.

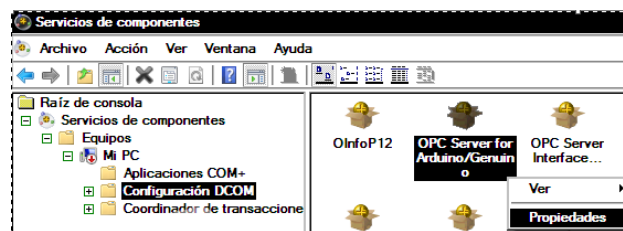


Figura 3.22 Configuración DCOM del OPCServer de Arduino Autor ().

- En la pestaña de **seguridad** solo se validarán las configuraciones realizadas acogiendo la opción *predeterminadas*, en cambio en la de **identidad** se modificará la cuenta actual por una de *usuario interactivo*, esto para obligar la comunicación de Ignition, bajo la instancia actual del servidor, se aplica y acepta el arreglo efectuado, como lo indica la Figura 3.23.

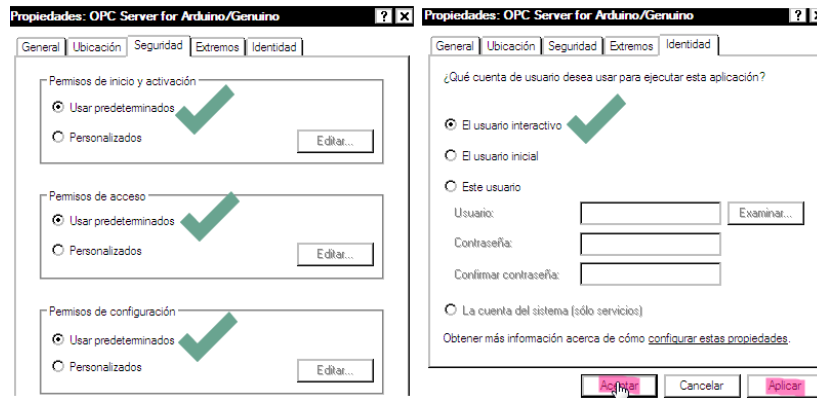


Figura 3.23 Configuración de permisos COM Autor ().

### 3.8. Programación del PLC

TIA Portal ofrece un medio confortable y de fácil manejo, al usuario para establecer la lógica de control que se apresta, a determinar la respuesta del sistema ante distintos estímulos de entrada. Como se había manifestado la programación es simbólica y en el software se dispone del lenguaje grafico **KOP**, apegado al de escalera que se programa por esquemas de contactos o circuitos. En el cual se confecciona un simple programa para hacer la configuración y uso del *servidor OPC UA Ignition* para los enlaces hacia el PLC. El desarrollo de este procedimiento de programación se observa a continuación:

1. En vista de ello la programación inicia estableciendo la comunicación mediante direcciones IP, entre el PLC y el ordenador en el que se encuentra el software.

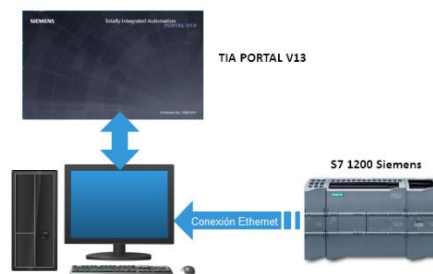


Figura 3.24 Programación del PLC S 7 1200 con TIA Portal Autor ().

2. En el entorno de configuración del software, se crea un nuevo proyecto, al cual se le proporciona un nombre y una ruta de almacenamiento. Acto seguido se agrega el dispositivo y se selecciona el tipo de CPU con el código de referencia que el PLC asignado tenga, esto se puede ver en la parte frontal y lateral respectivamente.

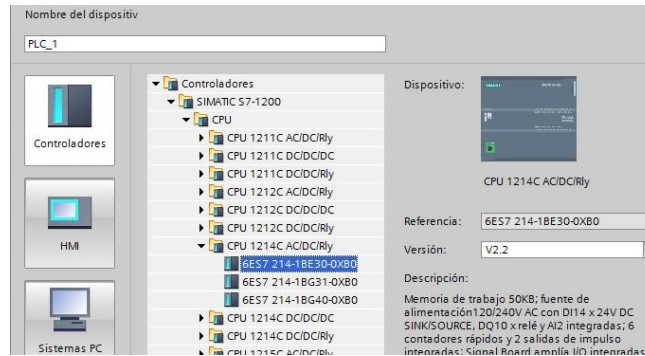


Figura 3.25 Selección del PLC S7 1200 Autor ().

**Nota.** En el laboratorio se cuenta con una **CPU 1214C** en configuración **AD/DC Rly** con número de referencia **BES7 214-1BG30 – 0XB0**.

3. En la “*tabla de variables estándar [16]*” se crean las variables, al complementar los parámetros ahí observados, iniciando por nombrar y establecer el tipo de variable y la “*Dirección*” de las memorias de marca (M0.0), salida (Q0.1) y la entrada analógica (IW64) y demás variables como; “*Escala y Norma*” que sirven para almacenar y convertir los datos enteros de la señal analógica a valores reales de la magnitud física que se mida.

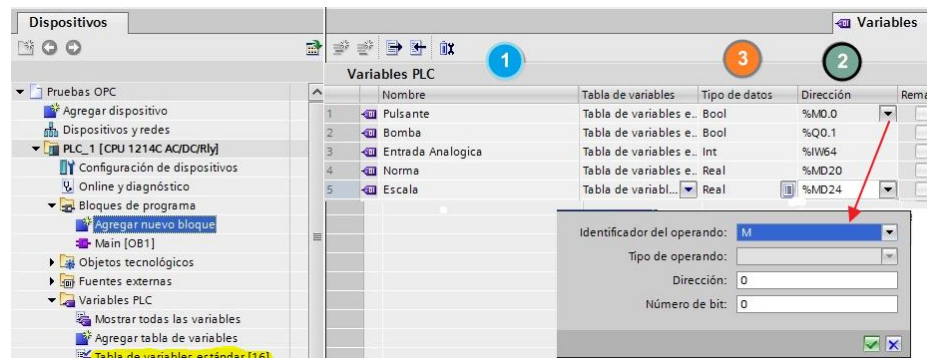


Figura 3.26 Creación de variables en la memoria y variables de conversión Autor ().

4. En consecuencia el **Main** de *Bloques de programa*, permite desenvolver la programación utilizando la lógica de esquemas de contactos, admitiendo la interfaz del bloque la inserción en sus segmentos de los elementos de *contactos*, *bobina* y los bloques de conversión *SCALE\_X* y *NORM\_X* para la captación de señales analógicas, que se arrastrar hasta dichos segmentos aclarando en cada elemento el nombre de la variable que se aplica y las condiciones respectivas que se agreguen.

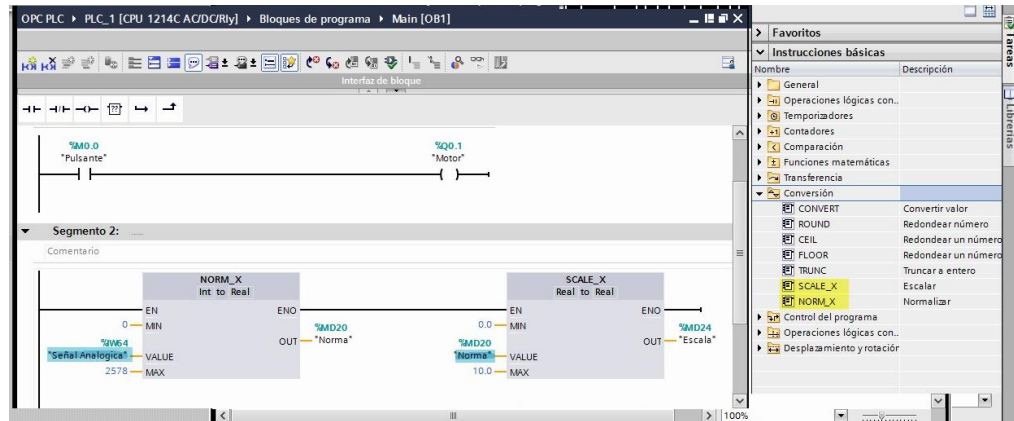


Figura 3.27 Programación KOP Autor ().

5. Compile el programa realizado para verificar que no haya errores, luego cargue el programa en el PLC, guardándolo previamente estableciendo primero la conexión con el computador comprobando, que este tenga una dirección IP configurada en el mismo dominio que el PLC, en este último configure los parámetros de *Conexión de red*, establezca la conexión online y cargue el correspondiente programa.

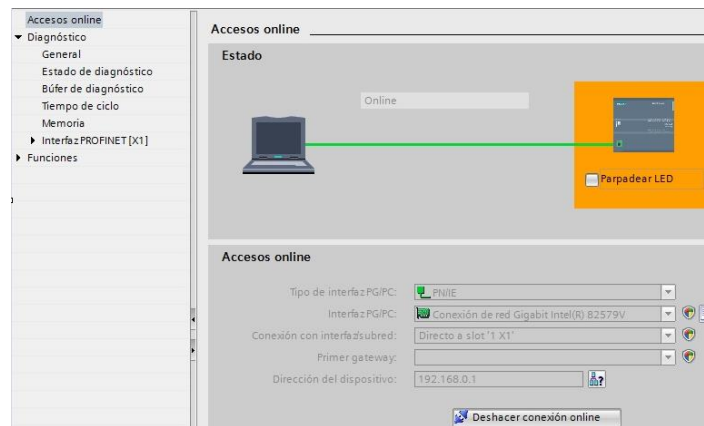


Figura 3.28 Conexión PLC ordenador para subir el programa Autor ().

Se ha escogido la marca *M0.0* para simular virtualmente el estado de un interruptor, como si se tratase de una entrada física “*I*” del PLC, este último acondicionará la activación de la salida física “*Q0.1*”. Es por esto, hay que indicar que los registros configurados en la variable virtual, de marca (M) y entrada analógica serán usados en el *SCADA Ignition* a través del servidor OPC UA. Sin desmerecer que puede obtenerse los estados de operación, de las variables físicas “*I* y *Q*” que simbolizan los puertos del PLC.



### 3.9. Construcción del servidor OPC UA Raspberry

La librería *Freeopcua de Python*, brinda varias funciones y métodos, para convertir a Raspberry en un *servidor OPC UA*. Este paquete de software puede relacionarse con otros módulos, para el gestionamiento de los puertos *GPIO*, permitiendo desarrollar aplicaciones controlables, por medio de variables se puede transformar a formatos *UA* deliberadamente, pudiendo gobernar los periféricos que se añadan a la placa. Además proporcionan facilidades para la transmisión de datos y definición de puertos mediante Ethernet.

#### Establecimiento de conexión a la red local

Para que la Raspberry pueda ser detectada por el *OPC SCADA Ignition*, el servidor OPC UA utiliza una conexión de puerto de red establecida, por la cual transfiera los elementos OPC UA que conectan a la plataforma con el cliente OPC UA Ignition. Esto requiere la configuración del puerto Ethernet, a través de la línea de comandos de Linux de Raspberry en la cual se procede a determinar la IP, mascara de red y el enlace al router. Para ello se emplea el comando *sudo nano /etc/dhcpd.conf*, como se muestra en la Figura 3.29.

```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: /etc/dhcpd.conf
# Some interface drivers reset when changing the MTU so disabled by default.
#option interface_mtu

A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

# A hook script is provided to lookup the hostname if not set by the DHCP
# server, but it should not be run by default.
nohook lookup-hostname
interface eth0
static ip_address=192.168.0.105/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1

```

Figura 3.29 Configuración de IP estática del servidor OPC UA para conectarse con la red Local Autor ().

#### Creación de las instancias y variables del servidor

La librería permite el intercambio de datos fluido y escritura flexible en las variables, entregando a cualquier cliente OPC UA expectante a conectarse, información del servidor y de las variables, recibiendo como parámetros el servidor el estado de conexión de cliente.



Los objetos OPC UA (variables) del servidor se organizan por nodos en su creación. Estos tienen un identificador de nodo, denotado como *NodeId*. El formato de este es también recibido por el servidor, ayudándolo a determinar, cuál es el nodo descubierto por el cliente operativo. Enseguida se describe las clases principales para el diseño de la interfaz de comunicación del servidor:

- **Server ()**. Esta función representa el módulo del servidor OPC UA. Permite la creación de todas las instancias; variables, propiedades y conexiones de red.
- **Set\_endpoint (url)**. Permite definir la dirección de red (IP: puerto de red) unificando el protocolo OPC con TCP/IP, para la conexión del servidor mediante la interfaz Ethernet, mediante la dirección localizador de recursos uniforme [**url**].
- **Register\_namespace (uri)**. Este parámetro permite agregar el nombre al espacio del servidor, por medio del cual los objetos y variables OPC UA serán reconocidos. Para eso implementa una dirección de identificador de recursos uniforme [**uri**].
- **Get\_objects\_node ()**. Es una variable usada como intermediario para crear los nodos OPC UA, su función es obtener nodos objetos del servidor y devolver un objeto nodo, donde residen las variables.
- **Add\_object ()**. Es utilizado para crear los objetos, permitiendo establecer sus propiedades como; nombre del objeto, tipo de dato, etc. Son insertados en nodos, haciendo posible que el cliente los observe para conectarse al servidor.
- **Add\_variable ()**. Es usada para declarar y agregar variables, dadas en el formato OPC UA, estas se introducen en el nodo objeto el cual contiene el tipo y nombre de la variable.
- **Set\_write ()**. Esta instrucción permite otorgar al cliente, privilegios para que pueda realizar la escritura en la variable a través de un nodo grabable.



- **Get\_value ()**. Se utiliza para obtener el valor de un nodo OPC UA, en el formato de Python. Funciona solo con variables y propiedades, ya que estas tienen los valores, admitiendo la comunicación nativa de Python.
- **Server.start ()**. Esta sentencia habilita al servidor, para escuchar a través de la red las peticiones de clientes compatibles con UA, iniciando todos los objetos y variables disponibles que presentará al cliente.
- **Server.stop ()**. Está instrucción termina o interrumpe la ejecución del servidor, desasando todas las conexiones de red de las variables y objetos, que se mantengan comunicadas con el cliente OPC UA.

El protocolo OPC UA tiene definidos los puertos [4840:4841:4048], para comunicarse razón por la cual resulta conveniente aplicarlos, para el servidor. Python tiene aportaciones de varios desarrolladores y colaboradores de software, proveyendo de librerías de diseño gráfico como *PyQT* para diseño de interfaces de usuario interactivas.

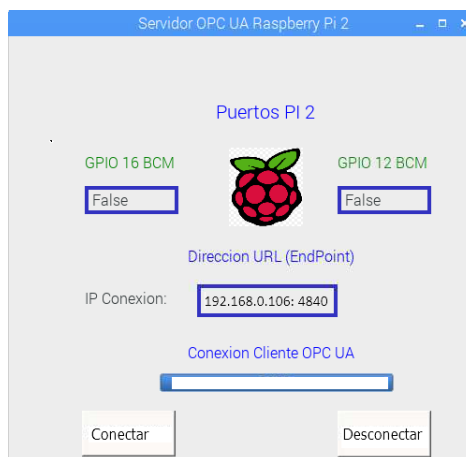


Figura 3.30 Ejecución, interfaz de servidor OPC UA desarrollado con Qt de Python en Raspberry Autor ().

La Figura 3.30 muestra la interfaz del servidor OPC UA desarrollada. Como soporte que favorezca, para una mejor comprensión y detalles del protocolo, se disponen de páginas y documentación que aportan con información de *Python OPC UA* que pueden visitarse en la sesión desde los siguientes *Links*;

Github: <https://github.com/FreeOpcUa>

Media: <https://media.readthedocs.org/pdf/python-opcua/latest/python-opcua.pdf>

### 3.9.1 Configuración de la red LAN gobernada por el sistema OPC

La naturaleza dinámica del sistema DHCP del router en la asignación IP, ha sido conveniente que se establezca en IPs estáticas en las unidades de control y el SCADA, para disponer siempre de la misma dirección que otorgue el router para la configuración IP de los dispositivos. Dado que esto trae consigo problemas en la identificación y por ende en la comunicación de los dispositivos. Aquello supone la configuración del servidor DHCP. La Figura 3.31 explica esta configuración con un router D – LINK.

**ROUTER SETTINGS**

Use this section to configure the internal network settings of your router. The IP address that is configured here is the IP address that you use to access the Web-based management interface. If you change the IP address here, you may need to adjust your PC's network settings to access the network again.

Router IP Address: 192.168.0.20  
Default Subnet Mask: 255.255.255.0  
Local Domain Name:   
Enable DNS Relay:

**DHCP SERVER SETTINGS**

Use this section to configure the built-in DHCP server to assign IP address to the computers on your network.

Enable DHCP Server:   
DHCP IP Address Range: 01 to 199 (addresses within the LAN subnet)  
DHCP Lease Time: 10080 (minutes)

**DHCP CLIENT LIST**

Host Name	IP Address	MAC Address	Expired Time

**25 - DHCP RESERVATIONS LIST**

Host Name	IP Address	MAC Address	
<input checked="" type="checkbox"/> Servidor-Ignition	192.168.0.100	d4-c9-ef-7e-08-8f	<< Computer Name ▼
<input checked="" type="checkbox"/> PLC-Siemens	192.168.0.1	00-1C-06-03-61-68	<< Computer Name ▼
<input checked="" type="checkbox"/> Arduino-Ethernet	192.168.0.105	90-a2-da-0e-ad-8d	<< Computer Name ▼
<input checked="" type="checkbox"/> Raspberry-Pi	192.168.0.106	c5-30-0b-5f-00-0a	<< Computer Name ▼

Save Settings Don't Save Settings

- 1 Cambiar la dirección de acceso al Router
- 2 Establecer la máscara de subred
- 3 Habilitar el Servidor DHCP
- 4 Especificar las direcciones IPs en los dispositivos host de la red
- 5 Guardar las configuraciones realizadas

Figura 3.31 Configuración de red LAN Autor ().

### 3.10. Desarrollo de la Aplicación

Se ha visto oportuno como aplicación de este sistema OPC, tenga lugar en un invernadero en el cual se realice el control y monitoreo de las condiciones climáticas. Por medio de los servidores OPC desarrollados, adaptando la arquitectura OPC dada con el sistema SCADA Ignition. Se dispone de variables analógicas y digitales enlazadas a los controladores, que adquieren datos y admiten el mando, de los dispositivos que serán aprovechados para diversos procesos, asistiendo cada uno una tarea en concreto.

Ante esto se ha precisado de algunas etapas e instrumentos para captar las señales físicas y dar órdenes de control, en las unidades que cumplirán con las siguientes funciones descritas a continuación en el prototipo planteado:



- La unidad de control Arduino Ethernet hará mediciones de la *temperatura* por medio de un *sensor DHT11* y de la *precipitación* a través del *sensor de lluvia YL83*.
- Al PLC S71200 Siemens le corresponde, capturar valores de la humedad del terreno con el sensor *Funduino* y gobernar el riego del invernadero, mediante una *bomba* con depósito de agua para el abastecimiento del cultivo.
- Finalmente, a Raspberry Pi se le dotara el control de una fuente de calor “*un foco*”, y la *ventilación* del invernadero, regulando las condiciones ambientales. Al sistema se suma una base de datos, para respaldo de información fidedigna.

### 3.11. Diseño del sistema SCADA

En el sistema concurrirán todas las conexiones de red. Al ser planteado como controlador centralizado, que gestione los procesos de los equipos mediante la nube. Esto propulsa a determinar un punto en la red, en la cual se aloje el servidor de aplicación de control y monitoreo Web, cuyos clientes tendrán acceso al cumplir con las credenciales requeridas.

#### 3.11.1 Ignition inductive Automation

Ignition es una plataforma de software *SCADA*, elaborada bajo **JAVA**. Permite el diseño de aplicaciones industriales, independientes del sistema operativo. La plataforma esta implementada en un sistema basado en la web, seguro de tiempo real y escalable. De manera que su estructura es portable, facilitando que todos los recursos y módulos necesarios para ejecutar la aplicación se carguen en el lado del cliente al exhortar dicha aplicación. Por lo cual el usuario no requiere la instalación de la aplicación y de ninguna otra plataforma que sirva de base. Solo es necesaria una versión compatible con **JAVA**.

El software tiene una licencia comerciable, no obstante dispone de versiones *Demo* que pueden obtenerse desde la página oficial de *Ignition Inductive*. En vista de ello se ha instalado la versión de Ignition 9.7.3, que presenta una arquitectura moderna, estable y mejorada.

### 3.11.2 Desarrollo de un SCADA en la nube

El sistema de gestión Ignition obvia la necesidad de trabajar física y justamente en el equipo que contiene el software, gracias a los módulos que dispone la plataforma y su entorno web arranca desde cualquier navegador, apoyándose en la tecnología *Java Web Start* y la arquitectura centrada *SQL Data Base* que posee. El portal o *Gateway Igniton*, permite la invocación y configuración de aplicaciones, incluso la incorporación de dispositivos. Este entorno gestiona todas las conexiones y demás aspectos del servidor SCADA. Se realiza la configuración y creación del proyecto para el control y supervisión de procesos, analizando algunos módulos utilizados en este trabajo en los siguientes pasos:

1. Configuración básica de ingreso al sistema Login
2. Conexión de los dispositivos basados en OPC (para cliente OPC COM y UA)
3. Conexión a una base de datos
4. Inicio del Diseñador de Ignition
5. Ejecución de un cliente
6. Configuración del servidor Web

#### 3.11.2.1. Configuración de ingreso al sistema

En la sección de configuración, se tiene acceso a funciones avanzadas para agregar dispositivos, conexiones a base de datos entre otros. Desde aquí se puede restaurar o guardar una copia del archivo global de configuración y la ejecución como tal del servidor. La Figura 3.32 muestra las credenciales de ingreso y en el cual se realizará la configuración de vínculo de los equipos del menú configuración (M2MTechnologies, 2013).

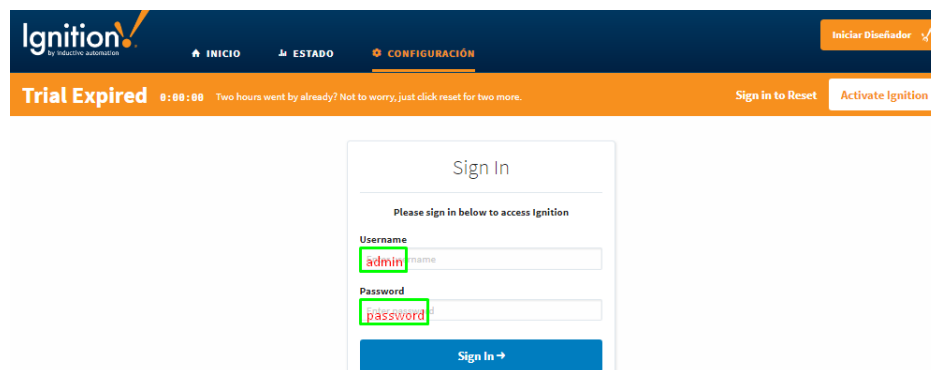


Figura 3.32 Ventana Login de Ignition Autor ().



### 3.11.2.2. Conexión de las unidades de control

Ignition, asume una estructura de comunicación universal, al ser un software SACADA pensado para la industria, conjunta la mayoría de protocolos de comunicación industrial en un solo paquete en la plataforma, disponiendo de varios drivers de comunicación precargados que la hacen multifuncional. El interés en esta aplicación recae, en que posee módulos de comunicación OPC-UA y OPC-COM, mediante los cuales los datos del proceso se adquieren de los distintos dispositivos que se conectan al sistema de gestión.

#### ➤ Módulo OPC UA

Pues así el módulo OPC-UA es un servidor OPC que admite la conexión nativa de controladores industriales de las marcas más reconocidas como Siemens, Allen Bradley entre otros y toda la gama de equipos y conexiones de red que puedan enlazarse a través del protocolo de comunicación OPC UA. Dotes que le otorgan a Ignition, ser el primer sistema que utiliza una pila de protocolos nativa OPC UA hecha en Java. En la Figura 3.33 se tiene el proceso de configuración, para la conexión del PLC Siemens.

El PLC S7-1200 se comunica con la aplicación a través del driver OPCUA. Para eso se inicia por insertar el nuevo dispositivo mediante la opción *Create new Device* de la sección *OPC-UA SERVER/Devices*. Posteriormente el sistema muestra las opciones de drivers que se encuentran disponibles en el entorno de configuración.

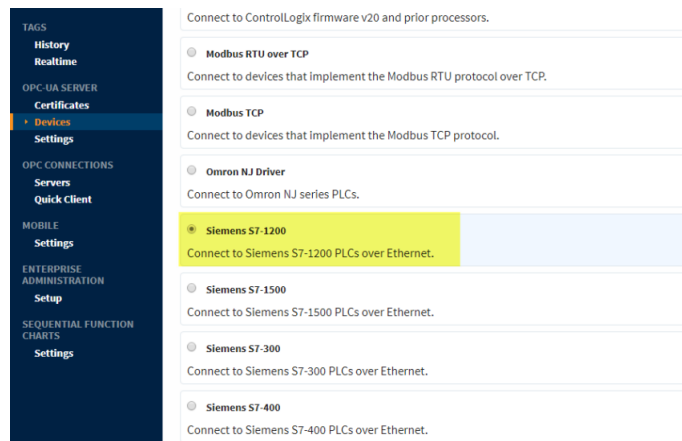


Figura 3.33 Driver de conexión OPC – UA del S7-1200 en Ignition Autor ().



Escogido el driver de comunicación a utilizar. El enlace se levanta al realizar el ingreso de los parámetros de conexión. La conexión se configura mediante puertos Ethernet, mediante la dirección IP y nombre del dispositivo.

➤ Módulo OPC – COM

Ignition aparte de los drivers OPC UA que contiene, puede admitir la comunicación y asociación de servidores OPC de terceros, para las instancias de OPC clásicos y OPC UA. De tal manera que expande sin restricciones de tipo o marca de plataforma, el desarrollo de arquitecturas OPC anti propietarias, está es una ventaja que prevalece en el software y lo han distinguido de los demás sistemas SCADA industriales.

Así el módulo OPC COM es utilizado para entablar enlaces con servidores OPC DA de forma local o remota. Con este y el módulo OPC UA que tiene instalado la plataforma Ignition, abarcan un completo sistema OPC para ambos tipos de estándares usados como puentes de conexión. En buena hora permitirá la comunicación del sistema OPC de Arduino Ethernet y del Raspberry.

La configuración mediante OPC/COM, se lleva cabo en el perfil de *OPC CONECTIONS / Servers*. El proceso de conexión del OPCServer Arduino empieza con la creación de un nuevo enlace, con la opción *Create new OPC Server Connection*, enseguida se procede con los pasos de configuración del enlace descritos a continuación:

- 1- Tras la creación del enlace, el sistema proporciona la opción de comunicación *OPC DA – COM Connection* que soporta las versiones **OPC DA 2** y **3** para el servidor OPCServer Arduino.
- 2- Se establece la localización del servidor, se elige la conexión local por encontrarse en el mismo equipo, donde reside el servidor Ignition.
- 3- Si todo se ha hecho bien, se observará el servidor OPC. Al escogerlo se carga automáticamente información del servidor en la ventana de la interfaz de configuración, como lo muestra la Figura 3.34.

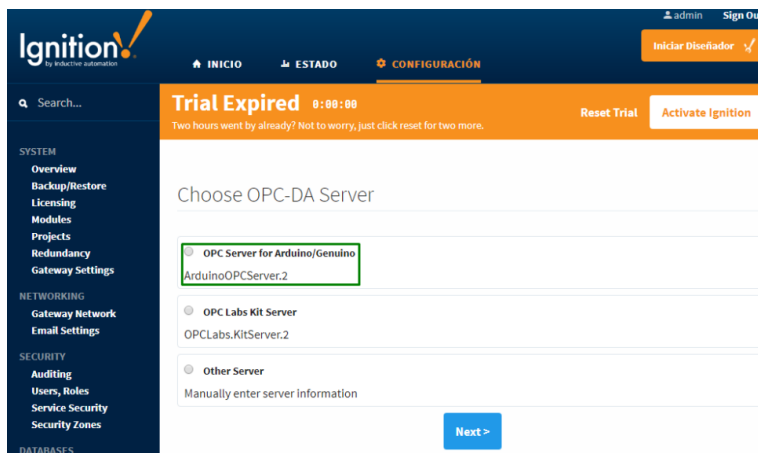


Figura 3.34 Detección del servidor OPC Arduino Autor ().

- 4- Finalmente se define, un nombre al enlace y se verifica la dirección Progid del servidor sea correcta, para habilitarla conexión como lo indica la Figura 3.35.

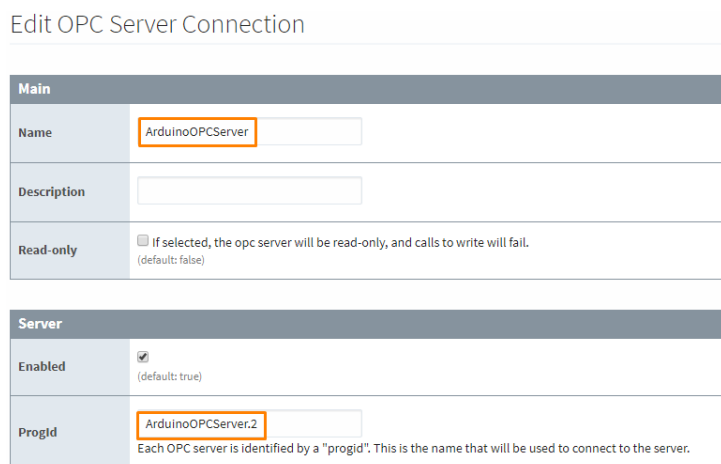


Figura 3.35 Interfaz del cliente OPC COM Ignition para el servidor OPC DA Arduino Autor ().

Dicha configuración entraña algunos aspectos a tenerse en cuenta. Como al navegar por los diferentes pasos de configuración, al establecer una conexión local, el sistema crea una puerta de enlace, siendo la instancia que lanza para comunicarse con el servidor, también el identificador del servidor denotado como *Progid* es un archivo de registro utilizado tanto por el S.O como por Ignition, para asociarla aplicación y controlar su comportamiento manifestando la compatibilidad entre los objetos COM, es decir entre el servidor – cliente.

De igual manera, para anexas el servidor OPC UA de Raspberry, se hace uso del módulo OPC UA, para lo cual se fijará una nueva conexión al dispositivo, definiendo como tipo de conexión la opción OPC UA. A continuación, se citan los pasos de configuración:

- 1- Encaminando el enlace se introduce la dirección de punto final “*Endpoint*”, que pertenece a la dirección de red con el cual se identifica el servidor para su detección.
- 2- Después, el sistema muestra una lista de opciones de seguridad se elige una para el servidor siempre y cuando, se haya adoptado una para tener acceso al servidor y autenticar la conexión, pero este no siempre es el caso, en los servidores OPC UA.
- 3- Finalmente, se define un nombre para levantar el enlace y sí es necesario y acorde a la opción de seguridad escogida se ha de configurar credenciales. Las Figuras 3.36y 3.39 muestran esta configuración para conectarse con el cliente OPC UA Ignition.



Figura 3.36 Dirección de red en el formato URL del servidor OPCUA Raspberry Autor ().

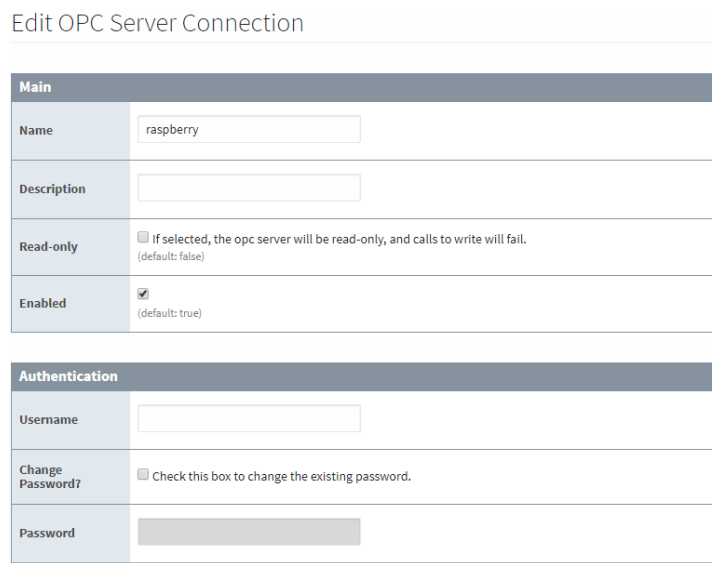


Figura 3.37 Ventana de edición de la conexión del servidor OPCUA Raspberry Autor ().



### 3.11.2.3. Conexión a base de datos

Dentro de las configuraciones básicas, el servidor SCADA tiene la posibilidad de vincularse con uno o varios gestores de bases de datos de diferentes marcas. Esta es otra ventaja del sistema frente a otros tipos de software SCADA, ya que la mayoría de estos trabajan con plataformas de manejo de información afines a su producto. Una base de datos provee un historial de datos que pueden evaluarse y ser útiles, para prevenir futuros fallos o corregir errores que producen los procesos que encargan las unidades de control.

La medida adoptada, permite realizar análisis de los datos mediante tendencias gráficas sobre el funcionamiento de los controladores conectados, al momento de entregar la información. Con ello se asiste al operario informando el comportamiento, de los dispositivos con las variables del proceso en un intervalo de tiempo definido, mismas que se insertan y almacenan en la base de datos.

Ignition puede conectar nativamente plataformas de gestión de base de datos, para respaldar datos históricos. Ante esto se hizo la instalación de un servidor de base de datos local **MYSQL Server**, junto al gestor de base de datos **MYSQL Workbench**, cuya conexión se realiza a través de un módulo nativo de MYSQL de Oracle que incorpora el paquete del sistema. A continuación, se puede observar el proceso de configuración de MYSQL para trabajar con el sistema de archivos históricos de Ignition:

1. En la ventana se muestra varias opciones configurables para conectar bases de datos entre ellas se escoge **MSQL Conector J**. Está creara una conexión a la respectiva base de datos apuntando al servidor en la que reside, como lo indica la Figura 3.38.



Figura 3.38 Modulo de conexión a servidor MYSQL Autor ().

2. En esta ventana (véase la Figura 3.39) se configura la interfaz del enlace a MYSQL, entre los parámetros más importantes está, el nombre de la conexión por medio del cual se enlaza las variables históricas del proceso, la URL de conexión, las credenciales de acceso para el enlace con la base de datos y el nombre de la base de datos que se ha creado en MYSQL para respaldar la información.

The screenshot shows the 'Edit Database Connection' dialog box. The 'Name' field is 'BD\_SCADAOPC'. The 'JDBC Driver' is 'MySQL Connector/J'. The 'Connect URL' is 'jdbc:mysql://localhost:3306/ignition'. The 'Username' is 'root'. There are two password fields, both with green checkmarks indicating they are filled. A red arrow points to the 'Connect URL' field with a text box that says 'Por defecto MYSQL crea una base denominada Test cambiamos por el nombre de la muestra Ignition'.

Figura 3.39 Configuración y verificación de la conexión a la base de datos Autor ().

Terminadas las configuraciones en los enlaces realizados, se pasa a comprobar que estos se hayan establecido con el sistema Ignition. Las instancias creadas, disponen de opciones para editar o eliminar inclusive crear nuevos enlaces. Hay que aclarar oportunamente que la versión demo involucra un tiempo de operación de la plataforma durante 2 horas, por ello se debe activarla, para su adecuado uso. Las siguientes Figuras ilustran lo mencionado.

Devices

Name	Type	Description	Enabled	Status	
Siemens			true	Connected	<a href="#">delete</a> <a href="#">edit</a>

[→ Create new Device...](#)

Figura 3.40 Conexión al PLC Siemens vista desde la sección OPC UA Server /Devices Autor ().



### Database Connections

Name	Description	JDBC Driver	Translator	Status	
BD_SCADAOPC		MySQL ConnectorJ	MYSQL	Valid	delete edit

→ [Create new Database Connection...](#)

**Note:** For details about a connection's status, see the [Database Connection Status](#) page.

Figura 3.41 Establecimiento de la conexión a la base de datos Autor ().

### OPC Server Connections

Name	Type	Description	Read-only	Status	
ArduinoOPCServer	OPC-DA COM Connection		false	Connected	delete edit
Ignition OPC-UA Server	OPC-UA	A connection to the OPC-UA server provided by Ignition's OPC-UA module.	false	Connected	More edit
Raspberry PI	OPC-UA		false	Connected	More edit

→ [Create new OPC Server Connection...](#)

**Note:** For details about a connection's status, see the [OPC Connection Status](#) page.

Figura 3.42 Conexión de Arduino y Raspberry vista desde la sección OPC Conecctions – Servers Autor ().

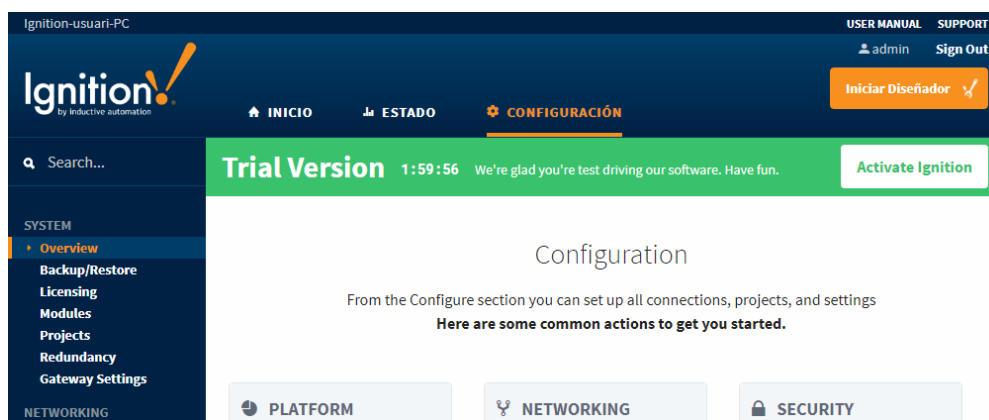


Figura 3.43 Activación de la plataforma con la opción Reset Trial Autor ().

El sistema, crea automáticamente el enlace *OPC UA* con las instancias dadas, esto es para compaginar y acordar un solo protocolo OPC, con ambos tipos de servidor OPC resultando el cliente OPCUA del cual depende la comunicación del sistema de control automático.

#### 3.11.2.4. Diseñador de Ignition

El *Diseñadores* el entorno de desarrollo de la aplicación “*cliente OPC UA*”, que se inicia por medio del Gateway como un servicio Web, siendo este el que guarde el componente de



acceso, a la interfaz de configuración grafica e importe los paquetes para correr su entorno. Está dotación hace que la interfaz de diseño sea portable, desempaquetando, en el ordenador que la invoca al llamarla por primera vez.

Ignition protege la integridad de cambios en un mismo diseño de proyecto, restringiendo el acceso de otras terminales diferentes al Diseñador, cuando existe un primer acceso a la interfaz en cuestión.

El Diseñador se acontece por medio de la opción *Iniciar Diseñador* en la ventana del Portal, tras esto se descargan los paquetes necesarios para ejecutar su entorno de configuración, cuya aplicación (JNLP) escribirá las directrices y configuraciones de acceso hacia el servidor, para la aplicación *Ignition Designer* (véase la Figura 3.44). En el cual se puede seleccionar y predisponer de todos los proyectos creados en el Diseñador, que asocian a distintas aplicaciones accesibles a través del *Gateway Ignition*.



Figura 3.44 Ingreso al Diseñador de Ignition Autor ().

Ingresadas las credenciales se visualiza la interfaz de *open/create Project*. Por medio de la opción “New Project” y “Create New Project” se crea un nuevo proyecto, localizándose en frente el cuadro de ingreso de parámetros del proyecto, que abarca las secciones; de nombre del proyecto que se completa con caracteres alfanuméricos sin espacio entre ellos (validados por el identificador gráfico), el ingreso de un título del proyecto que acepta el



espaciado y por medio del cual es identificado en las interfaces. El resto de medidas son el perfil de autenticación, la base de datos, el proveedor de Tag (variables), la elección de plantillas graficas a aplicar en la interfaz y la descripción para más detalles del proyecto, esto se esquematiza en la Figura 3.45.

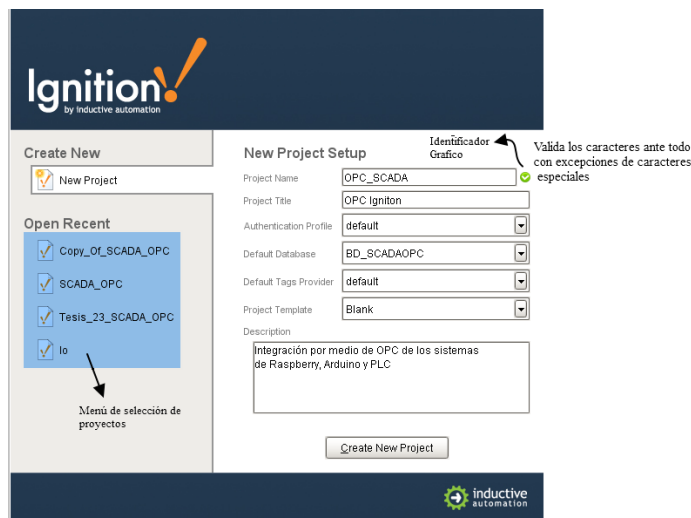


Figura 3.45 Creación de nuevo proyecto en el Designer de Ignition Autor ().

El entorno del diseñador de aplicaciones del servidor Ignition, dispone de opciones generales (véase la Figura 3.46) que permiten el ingreso a varias de las funciones y opciones de cuadros control. Como base técnica para el desarrollo de la aplicación, se tiene los siguientes bloques:

- ✓ **Project Browser** - Comprende todas las provisiones de gestión del proyecto, como, grupos de transiciones de bases de datos, ventanas del proyecto y otros.
- ✓ **SQL Tag Browser**- En él se administra y configura todas las variables del proyecto mediante *Tag* como: OPC y otros tipos de datos que pueden definirse.
- ✓ **Property Editor**- En este se hallan todas las propiedades de un elemento en específico. En el cual se puede organizar las propiedades y mostrar u ocultar por niveles de complejidad, gracias a los filtros que coloca la ventana.
- ✓ **Component Palette**- Está ventana tiene todos los objetos gráficos y controles que pueden insertarse en la aplicación botones, gráficos, entre otros.

- ✓ **Ventana gráfica de aplicación (Vision)** - Presenta, el ambiente de diseño de la interfaz gráfica (HMI) de la aplicación implementando los controles y objetos gráficos, que flexibilizan y dan la apariencia a la aplicación SCADA

La interfaz gráfica, permite que los objetos sean integrados en forma fácil y dinámica, esto hace posible que el esquema de monitoreo se implemente y desarrolle acorde a las necesidades, para el control de los parámetros que se midan (como: humedad, temperatura y precipitación) y de los elementos de maniobra. A su vez estos poseen varias propiedades que pueden configurarse, ofreciendo maniobrabilidad de las variables OPC, de tal manera que se ajustan a los requerimientos del sistema, como lo indica la Figura 3.46.

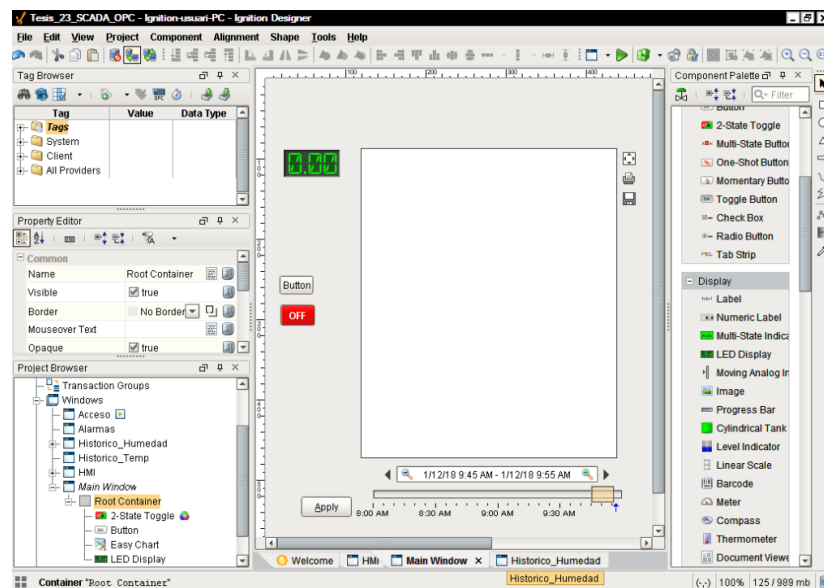


Figura 3.46 Colocación de los objetos en la ventana de diseño del Designer Ignition Autor ().

Para disponer de las variables de cada proceso en el diseñador se tienen que crear los enlaces con el *Editor de Tag*. Ciertamente es que cada una tiene propiedades configurables en el mismo editor; en él se establece el nombre simbólico de la variable, el formato de dato de presentación, el tipo de acceso y las propiedades del servidor OPC para acceder a las mismas. Las variables creadas llevarán el nombre de su correspondiente controlador, para su fácil identificación. Entre los campos de mayor relevancia en la configuración están:

- ❖ El Access Rights. Establece los derechos de lectura/escritura que admitan los servidores OPC respecto a sus variables.



- ❖ El OPCServer. Muestra la lista de servidores OPC conectados al sistema.
- ❖ El OPC Ítem Path. Es en donde se establece la dirección del enlace, que tienen las variables, para tomar su valor en el dispositivo (controlador del proceso).

Las variables de los elementos de maniobra son definidas en el *editor de Tag*. El driver del servidor OPC UA del PLC es inherente en la plataforma y se acontece mediante el OPC Server, en él se visualiza las características y cualidades que puede definirse en el servidor.

Para que el servidor ingrese, hacia a las fuentes de datos del PLC, el contexto de configuración, le permite a este trabajar con las propias direcciones de las variables declaradas en el dispositivo, creando así el enlace nativo hacia el PLC, obteniendo directamente los valores de las variables. A esto se suma también el poder definir el tipo de acceso para leer/ escribir en las variables en la aplicación cliente OPC UA.

En la Figura 3.47 se explica la configuración OPC UA de las variables del PLC, cuya dirección de la variable ha sido definida siguiendo el formato de acceso, al área de memoria interna que se habían explicado en el apartado **3.3.3.1 Acceso a la memoria del PLC**, del mismo modo el tipo de dato, acorde a la variable del PLC.

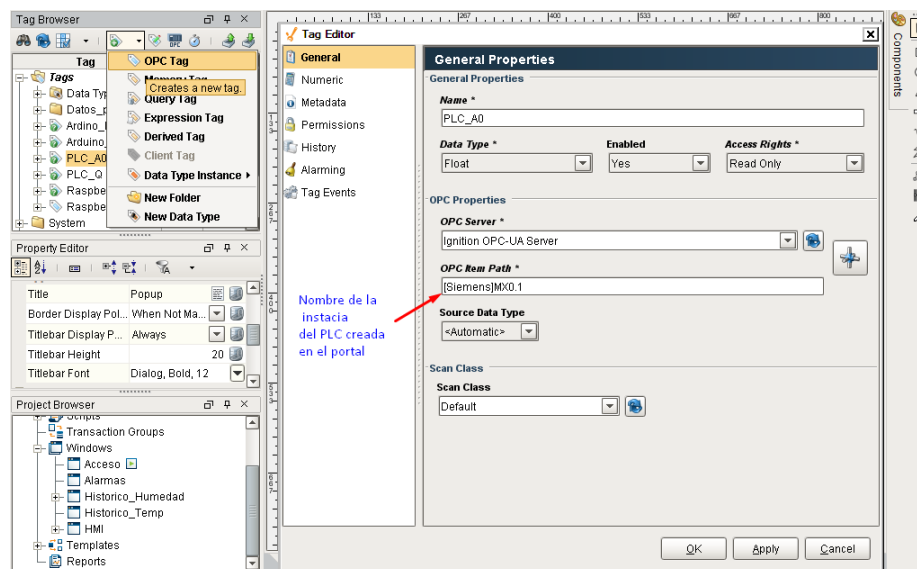


Figura 3.47 Enlace de las variables del PLC en el formato *Nombre. Instancia [dirección registro del PLC]*  
Autor ().



El enlace de las variables de Arduino Ethernet y Raspberry, difiere ligeramente a lo visto con el PLC. Sobre todo con la dirección de las variables, dado que estas son otorgadas directamente desde los servidores OPC que las contienen. Para eso el *Choose OPC Item* es una herramienta del editor, que permite mostrar todos los nombres de las instancias del sistema actualmente comunicadas con los servidores OPC externos, que se habían establecido en el Portal Web, dando a conocer cada una de ellas las variables OPC, disponibles junto a sus propiedades, para entablar los enlaces hacia cada dispositivo.

En la Figurar 3.48 se presenta la configuración del enlace, para elegir variables OPC siendo el mismo proceso en ambos sistemas embebidos.

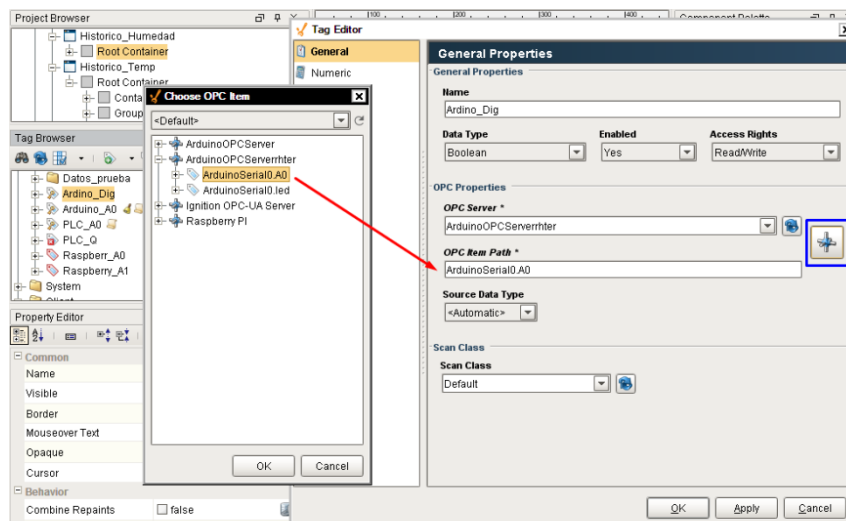


Figura 3.48 Enlace de las variables OPC de Arduino en el Diseñador Ignition Autor ().

Los parámetros vistos de la imagen tienen características que valen la pena aclararlas. En los sistemas embebidos, las variables OPC representan procesos internos de otras variables que se habían programado en estos elementos, de ahí que estas variables no constituye la fuente de datos real, sino la ruta que muestra el servidor OPC para acceder a ese valor en el dispositivo.

Los privilegios de acceso a las variables, deben ser coincidentes a los otorgados en cada dispositivo, respecto al tipo de dato y derecho de modificación y legibilidad que admite la variable, pues así se evitan incoherencias en su maniobra, al momento de programar en el



Diseñador. Al caso contrario del PLC que es de libre acción escoger que sus variables puedan ser solo de escritura/lectura o ambas en su servidor OPC UA.

Seleccionados los proveedores de variables OPC, estas se destinan a un navegador de Tags y una carpeta **Tags** que el sistema reservada y archiva, en la cual pueden observarse todos sus atributos y estados en tiempo de ejecución.

La Figura 3.49 muestra la inclusión de una variable (con arrastrar y soltar la etiqueta) en un componente grafico desde la tabla Tags, en el entorno del cliente OPC SCADA, proporcionando información del proceso que se encomienda cada controlador respectivamente.

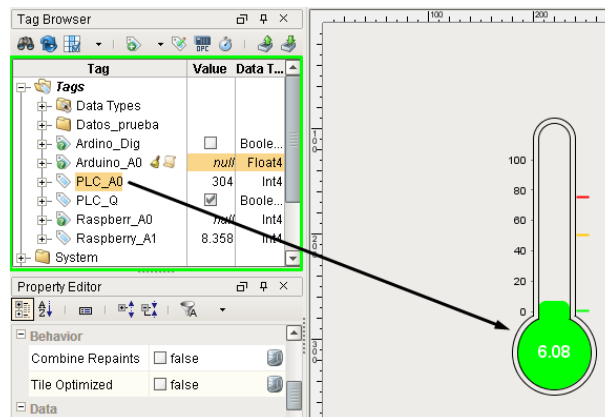


Figura 3.49 Enlace de objetos con las variables “Basa de datos SQL de etiquetas del sistema” Autor ().

La propiedad de valor en la tabla cambia en tiempo real, y en función de este va depender el objeto a elegir, para representar apropiadamente el valor de una variable. Además en la misma pestaña donde está la tabla se puede elegir, muchas propiedades más que están preestablecidas que ayudan a entender mejor un tipo de dato.

### Inserción de datos en la base de Datos

El monitoreo de un proceso en el tiempo, conlleva a un registro histórico de variables que adquieren información de relevancia del proceso que se desean archivar en el tiempo. Así el Diseñador permite hacer uso de funciones OPC HDA en su sistema de archivos históricos, individualmente con cualquier variable del proceso. La configuración de almacenamiento

consiente operar con consultas de estado. Todo esto de algún modo permite comprobar, si surten el efecto esperado en base a las contemplaciones y perspectivas del usuario al darse una modificación o ajuste en el funcionamiento del proceso.

Si bien es cierto, los datos históricos pueden guardarse en cualquier base de datos de acuerdo a las conexiones creadas en el Gateway, quien gestiona todos los procesos relevantes vía web del servidor. Con esto los valores de las variables se pueden almacenar de dos posibles formas mediante Tags históricos o grupos de almacenamiento. En las Figuras 3.50 y 3.51 se muestra la configuración de las variables para cada forma de almacenamiento.

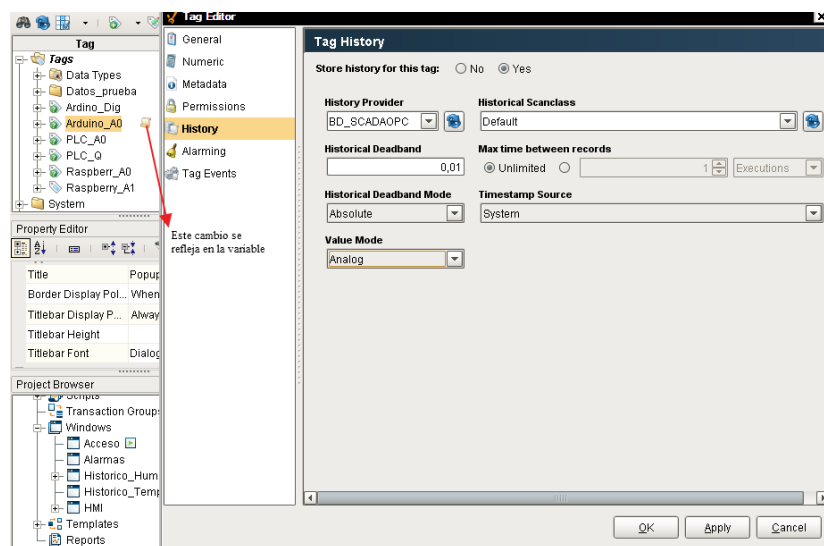


Figura 3.50 Configuración de parámetros históricos de un Tag (conversión Tag a históricos) Autor ().

En el cuadro de configuración del historial Tag OPC. Se puede seleccionar; la base de datos de archivos históricos, ciclo de muestreo para almacenamiento ininterrumpido, incluso definir la hora y fecha de medición, resultando óptimo para medir perturbaciones. Además de poder realizar consultas SQL para valores específicos, extraídos de la base de datos.

El *Transaction Groups*, permite guardar Tags directamente en una tabla personalizada, mediante la creación de grupos de almacenamiento en los cuales se puede configurar, la forma de cómo se transmite y obtiene un dato, elegir la tabla y los campos que contiene en la base de datos, inclusive poder visualizar la inserción de los datos en tiempo real, hacer consultas de datos por medio de SQL entre otros procedimientos más.

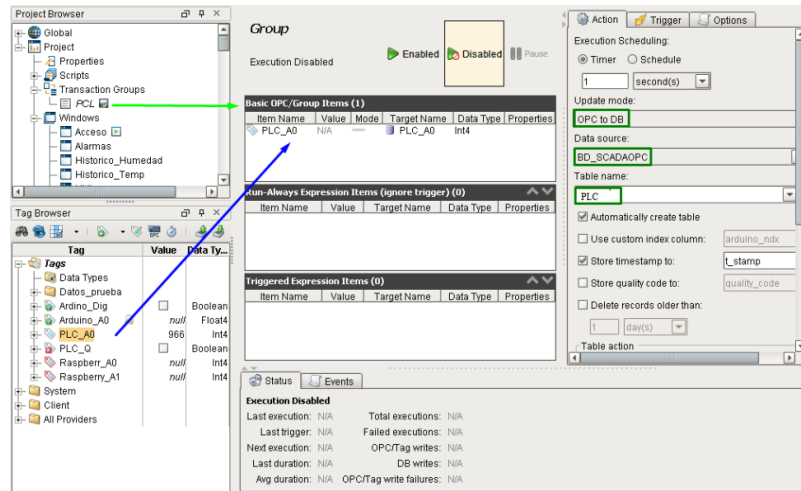


Figura 3.51 Creación de un grupo estándar “PLC” en el respaldo de un Tag Autor ().

En la venta de la Figura 3.541 se ha configurado un enlace de OPC a BD con la base MSQl, a través de un grupo estándar. Hay que tomar en cuenta que mientras el grupo este activo el proceder del almacenamiento será continuo.

Estas atribuciones, son importantes ya que es posible elegir ubicaciones con mecanismos de seguridad distintos al servidor web, haciendo que los datos históricos estén a salvo ante cualquier daño en el equipo principal usado como servidor web. Los datos históricos son vitales para evaluar la eficiencia y efectividad del sistema de control OPC y ayudan a la toma de decisiones.

Aquellas variables que sean históricas, las tendencias permitirán el trazo de los valores de los parámetros medidos en el tiempo (real o histórico). La Figura 3.52, exhibe el entorno de edición de una tendencia “como el controlador Easy Chart”, donde se pueden ver las variables de trazo, junto a su nombre y forma del trazo, como también es posible agregar nuevas variables históricas o adquirir datos desde la base MYSQL. Permitiendo por medio de tablas hacer consultas SQL de un tipo específico de dato, para presentarlo en un cuadro histórico.

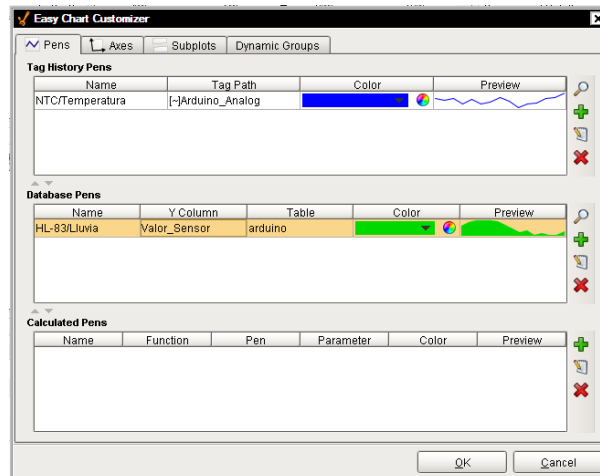


Figura 3.52 “Easy Chart” Asistente de configuración de control de curvas para representaciones graficas Autor ().

A través del control de curvas, la configuración puede ser por defecto o programación, para que varias propiedades como el rango de la gráfica, exportación de datos, ventana de tiempo, color, nombre y forma del trazado, entre otros atributos más, puedan ser modificadas dinámicamente en el tiempo de ejecución mediante controladores externos. La Figura 3.52 muestra la configuración para tres tendencias distintas en la misma ventana, hechas en el editor a través de “Subplots”.

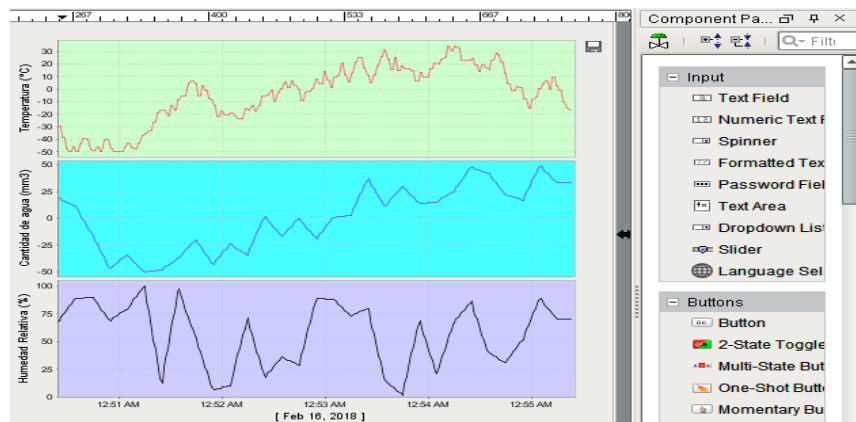


Figura 3.53 Configuración del trazado en tiempo real e histórico en el “Easy Chart” Autor ().

El dinamismo de un objeto gráfico constituye en el sistema de monitoreo de proceso, una forma básica de comunicarse con el usuario que lo gestiona, esto se produce a través de los distintos valores que asumen las variables del proceso, pudiendo ser estados lógicos o valores numéricos, acondicionando cambios de color, intermitencia, alarmas, etc.



Para la dinamización se tiene herramientas, que flexibilizan y facilitan la creación de dinámicas en los objetos, como el *Customizer*, el cual permite entre sus funciones agregar nuevas propiedades hacia un tipo concreto de dato, a un objeto inclusive reclutar varias propiedades del mismo para realizar dinámicas complejas de forma unificada.

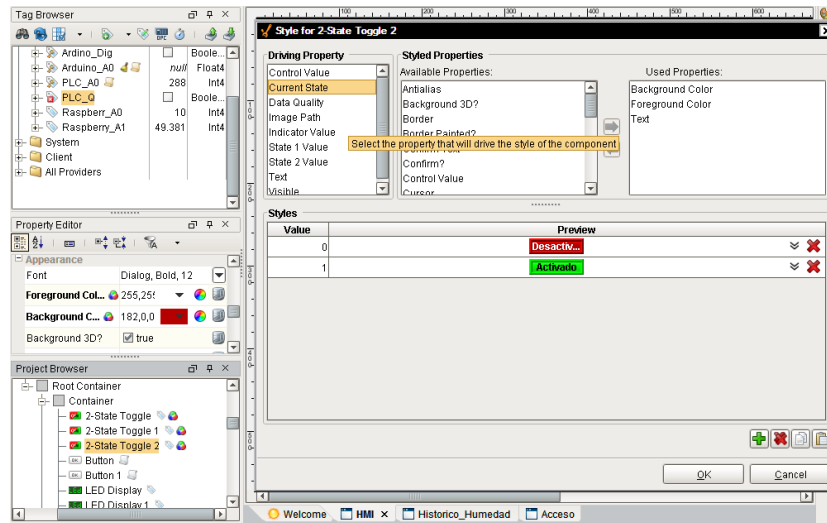


Figura 3.54 Dinamización del objeto mediante Style Customizer Autor ().

La dinámica suscitada de la Figura 3.54 hará que los colores de un botón de estados, cambie en función del valor de un Tag. La configuración establecida, hará que el botón sea rojo para el valor de 0 y verde para 1, así también el texto será blanco cuando sea 0 y negro con 1, empleando algunas propiedades para darle esta característica.

Sin embargo, este tipo de configuración no es la única utilizada en este aspecto. La plataforma de Ignition admite la programación mediante *Script*, que pueden usarse para agregar dinamismo y personalización en el entorno de diseño gráfico hacia los procesos. Para esto el Diseñador dispone de lenguajes de programación basados en Python y Excel como lo es *Jython* y *Expression* correspondientemente. Los cuales están dotados de un amplio juego de librerías y funciones dedicadas para cumplir estas tareas. En las Figuras 3.55 y 3.56 se expone la configuración de dinamización de componentes gráficos y Tags, con Scripts usados para el buen desempeño del sistema de gestión con los procesos.

La plataforma Java de Ignition puede compilar, sin dificultad secuencias de códigos de Python usados principalmente para el manejo de eventos de componentes. La composición de estas plataformas es a lo que se le llama Jython, un lenguaje orientado a eventos basado en Python. Dentro de este entorno se puede configurar el objeto, para diferentes tipos de eventos que admita realizar acciones concretas, como al abrir una ventana emergente al hacer clic en un objeto grafico o pasar de un formulario de Diseño a otro.

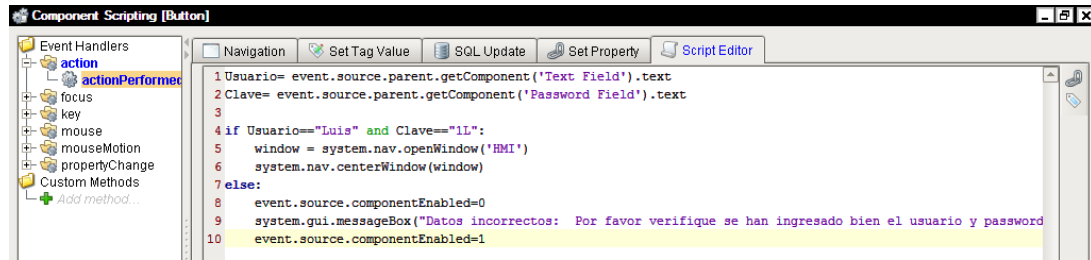


Figura 3.55 Programación mediante Jython en las propiedades del botón para confirmación – acceso Autor ().

La dinámica tendiente hace uso de un Scripts para las credenciales de acceso, en un botón a través de cuadros de ingreso de texto, cuyo código comprueba que el usuario y contraseña sean las correctas, para el ingreso al HMI del invernadero al iniciar la aplicación.

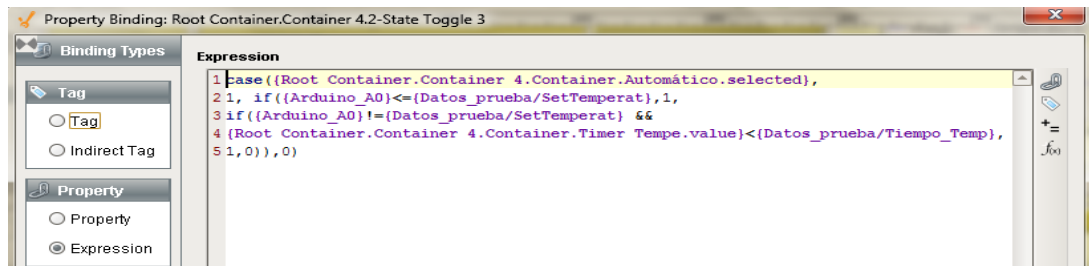


Figura 3.56 Scripts mediante Expression en las propiedades de los componentes Autor ().

Expression es un lenguaje que permite realizar dinámicas de valores con las variables OPC, de forma simple y sencilla a través de sentencias que actúan directamente en los datos, haciendo que no se requiera de una declaración previa. Siendo ocupado en el control automático de los procesos y dinámica de colores, realizando dinámicas simples por medio del editor de propiedades de los componentes, permitiendo combinar propiedades de los objetos y Tags mediante funciones especiales, en los controles.

Por otro lado un aspecto, fundamental del sistema de monitoreo y control de procesos es la capacidad para detectar anomalías y dar aviso, ante estas situaciones. Esto debido a que



pueden suscitarse ciertos eventos adversos al proceso, que se contrastan con las características del clima dentro del invernadero, que incidan negativamente en el cultivo. Las alarmas son un mecanismo clave de advertencia, que bajo condiciones específicas de los parámetros del proceso, ayudan a determinar la incidencia de eventos que lo pongan en peligro.

Para eso el Diseñador Ignition, permite hacer uso de las cualidades del formato OPC A&E con el sistema de alarmas. Admitiendo que los Tags puedan configurarse como alarmas dentro de sus propiedades; permitiendo definir un grado de riesgo, la operación de activación con el valor de proceso que se mide, entre otras opciones más.

La Figura 3.57 muestra la creación de una alarma con este atributo para la temperatura, en la cual se establecido el grado de riesgo acrítico, cuyo seteo de la alarma para su activación puede ser cambiado por el usuario. De igual modo se ha creado una alarma, en el sensor de precipitación que se activase cuando no existe agua para el riego.

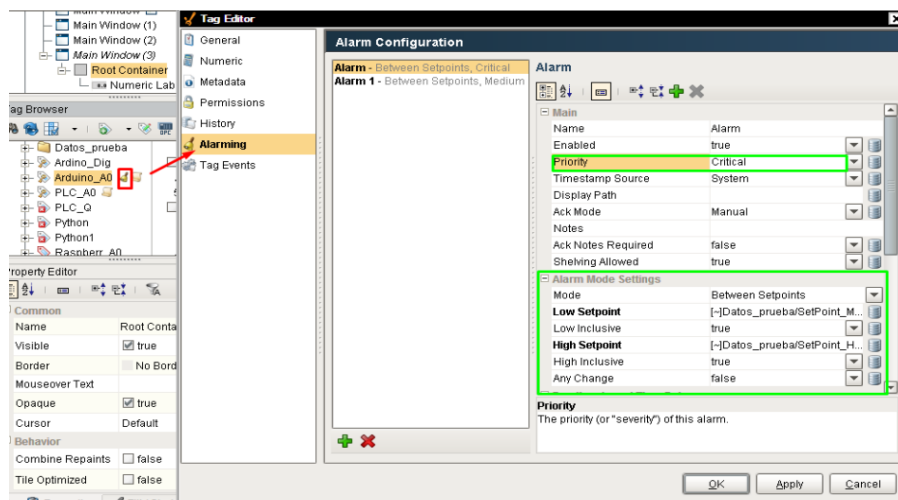


Figura 3.57 Configuración de alarmas con el Tag OPC Arduino Autor ().

Está opción configurable que atribuyen las propiedades de las variables OPC, en el entorno grafico del cliente OPC UA Ignition, permitirán llevar un registro de las anomalías originadas en el proceso, dándoles seguimiento para controlarlas a tiempo. Como se habría podido observar en la estandarización está la facilidad y libertad del diseño, al permitir que las variables sean capaces de operar, con las funciones de otras especificaciones OPC.



Los lenguajes de programación de Ignition, hacen posible la emulación a semejanza de la aplicación y ocurrencia de funciones operativas del proceso, brindando al operario una clara comprensión de cada elemento, permitiendo la creación y combinación de Scripts más complejos, trabajando con los Tag OPC y demás recursos realizando tareas específicas eficientemente creando la interoperabilidad de las unidades, al compartir información.

Una vez dinamizado el entorno grafico a través de las diferentes Tag, se puede guardar el proyecto, esto se realiza con el botón “**Save and Publish**” (véase la Figura 3.58), el cual guarda las configuraciones realizadas y publica en la web los cambio efectuados, estos se ven reflejados al instante en el portal tras guardar el proyecto.

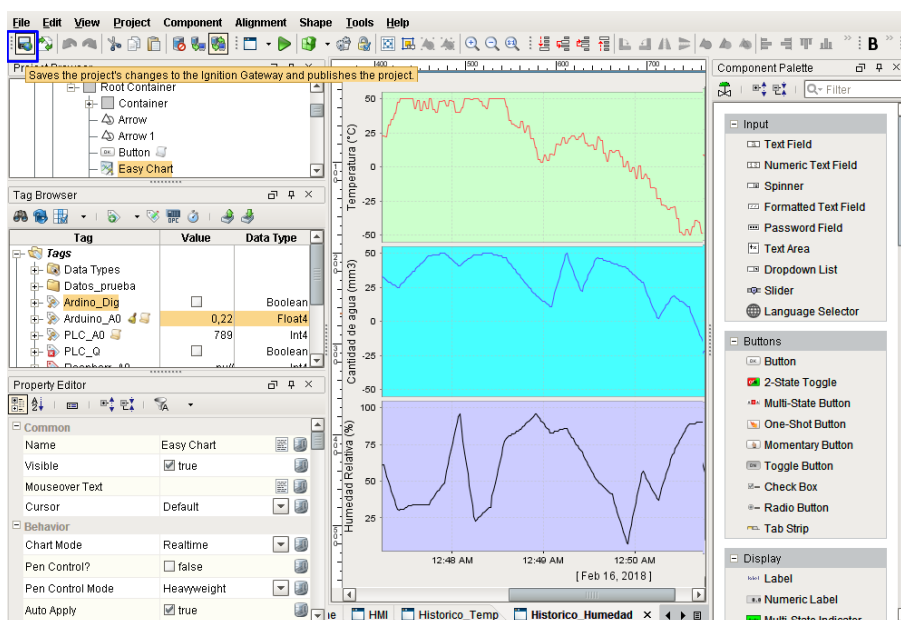


Figura 3.58 Almacenamiento y publicación del proyecto Autor ().

### 3.11.2.5. Cliente Ignition

Basta con solo presionar el botón de **Launch**, para que se invoque a la aplicación que carga e instala los recursos necesarios, para correr el proyecto en la terminal que hace la petición. Este proceso inicia al enviar el servidor un paquete de instalación, que se ejecuta automáticamente y el cual configura el arranque del ambiente de la interfaz gráfica. Con ello el usuario de destino no tiene que dotar el sistema de ningún complemento especial adicional para poder ejecutar el programa excepto el soporte JAVA. Instalados los recursos



necesarios se ejecuta la aplicación gráfica, sin antes haber ingresado en el menú de acceso las credenciales de autorización. La Figura 3.59 muestra la aplicación cliente OPCUA.



Figura 3.59 Ejecución de la aplicación SCADA como cliente OPC UA Autor ().

Existen tres formas de lanzar una aplicación, desde la ventana de configuración del proyecto, así la aplicación puede ejecutarse y mostrarse en distintos modos que son seleccionados en las opciones del proyecto entre estos están; el modo de **venta**, pantalla **completa** y **applet**. Esto se debe realizar antes de ejecutar la aplicación de forma habitual como lo indica la Figura 3.60.



Figura 3.60 Modos de ejecución de la aplicación cliente “SCADA OPC UA” Autor ().

El servidor Web luego de llamar y ejecutar la aplicación, tiene como tarea transferir los complementos de instalación de la aplicación, o en su lugar actualizar su conjunto de archivos, esto cuando ya ha sido ejecutada de tal manera que esta se presenta en la interfaz del cliente final tal como se la ha solicitado. Al tratarse de un cliente fijo como el tipo PC, es común que este se ejecute en el modo de pantalla completa, de tal forma que al ejecutarse el SCADA central de la operación, en la terminal puede observarse los cambios

de estado y fluctuación de las variables sin la intervención de otras aplicaciones que puedan influir en elementos distractores al usuario (véase la Figura 3.61).

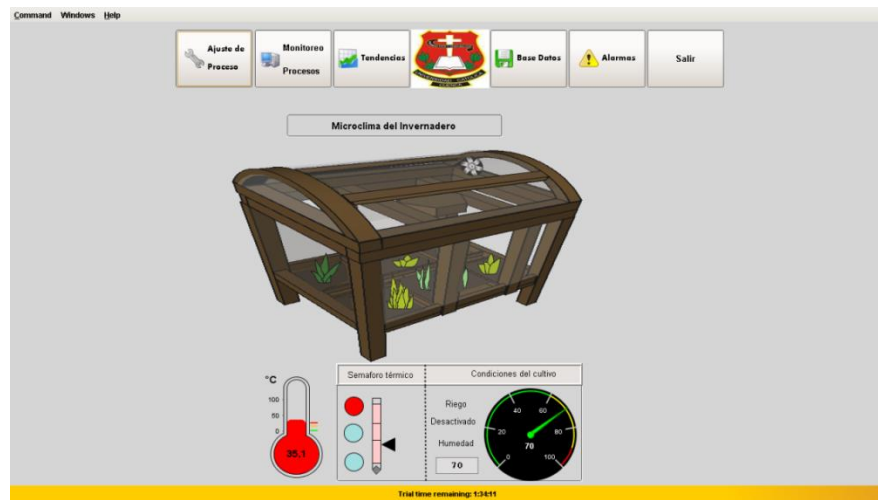


Figura 3.61 Aplicación cliente ejecutada en el modo de pantalla completa Autor ().

Gracias a la arquitectura web del sistema, se puede dar cabida a varios clientes. Un aspecto importante de la plataforma, es que al realizar un cambio en el entorno de diseño este automáticamente, publica un aviso a todos sus clientes activos, indicándoles de que existe una nueva versión de la aplicación, esta actualización se realiza en el propio marco de la aplicación en forma intuitiva mediante un cuadro de diálogo informativo. Este procedimiento se observa en las siguientes ilustraciones.

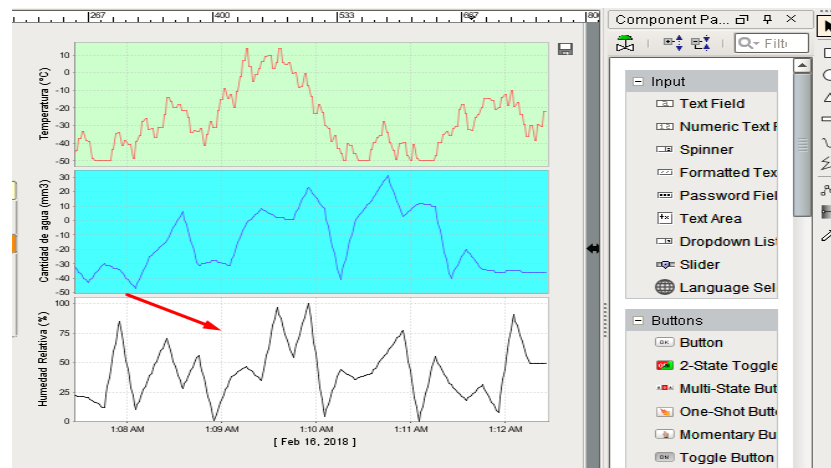


Figura 3.62 Modificación de la interfaz y publicación de los cambios Autor ().

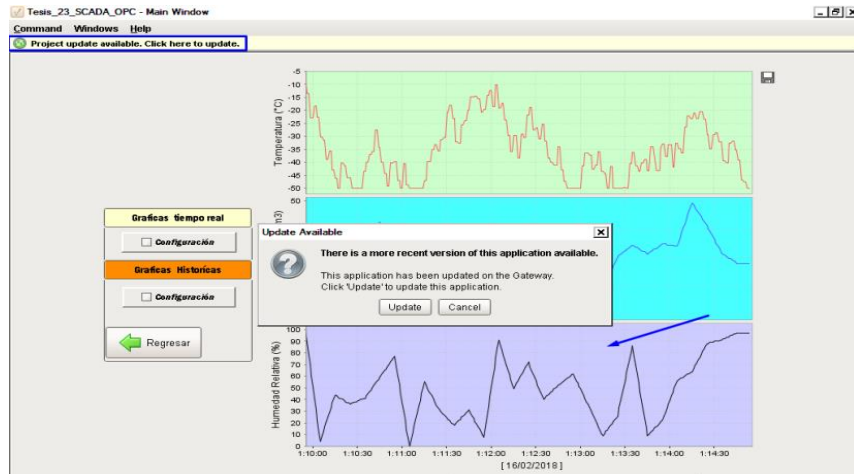


Figura 3.63 Aviso de actualización automática al cliente Autor ().

Al aceptar la actualización la aplicación se vuelve a cargar, reflejándose en la interfaz de diseño todos los cambios realizados, que pueden ver los clientes conectados. En cuanto a los nuevos clientes, al abrir la aplicación se empaquetan los archivos más recientemente actualizados. Es así como se puede observar la enorme capacidad y versatilidad de la plataforma, para actualizar un proyecto (véase la Figura 3.64)

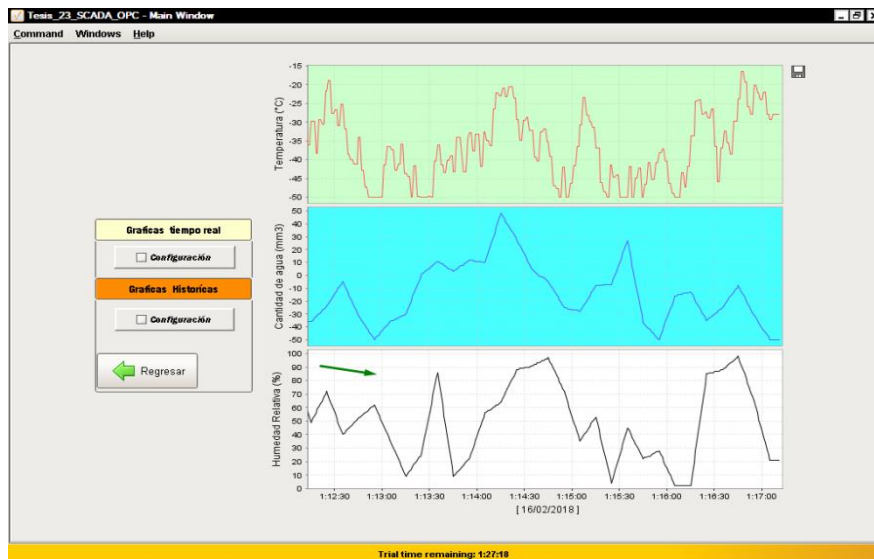


Figura 3.64 Actualización del cliente, después de modificar la interfaz principal Autor ().

### 3.11.2.6. Puerta de enlace Gateway

La *Unidad de Gateway Ignition (UGI)* es el motor del servidor Ignition. De ella depende el funcionamiento como tal de la plataforma y todas las comunicaciones de los dispositivos enlazados. Está se encomienda en habilitar la operación del Gateway, poniendo en marcha el servicio web y los servidores de aplicaciones y comunicación, empleando el sistema un servidor **TOM CAT**, integrado en el paquete de instalación, con el cual puede mediar con las fluctuaciones de los datos provenientes de elementos de mando o de clientes.

La UGI permite realizar configuraciones especiales, que comandan el funcionamiento del portal, logrando cumplir varios procesos como: habilitar o deshabilitar servicios de Ignition, restablecimiento del sistema, configuración de puertos *web* y de seguridad *SSL*, entre otros. Por defecto el servidor Ignition designa los puertos *8088* y *8043*, usados para el servicio web y conexión segura por telnet respectivamente. La Figura 3.65 muestra la configuración del servidor con la UGI, para habilitar el servicio web de Ignition.

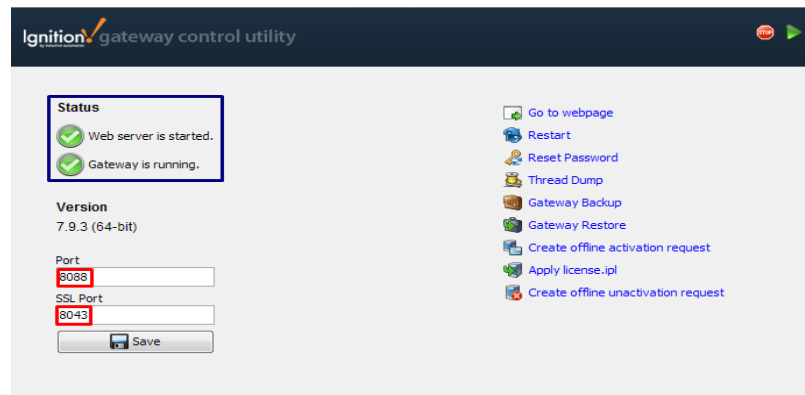


Figura 3.65 Configuración de puerto para servicio web en la CGU Autor ().

Se ha visto idóneo como así lo indica la Figura 3.65, adoptar el puerto 8088 en el servidor de Ignition para las comunicaciones remotas, a través del cual se recepte las peticiones de los clientes y por el cual se transmita la información. Para ponerlo en operación se guarda la configuración de la UGI, la cual restablecerá los servicios y el servidor de la plataforma, en cuya disposición se consigue establecer la consolidación, de la dirección IP del equipo servidor Ignition con el puerto dado, como en todo servidor web lo que hace es ligar la IP con un puerto, la cual se verifica por el estado activo del servidor web y del portal.



La ventaja de esta configuración, es que comprime la comunicación entre la puerta de enlace y los clientes, esto hace posible que el servicio web sea también para dispositivos móviles como; iPad, iPhone que manejen un sistema operativo **Android** o **IOS**.

### 3.11.3 Problemas

Durante el desarrollo de este proyecto, se presentaron varias dificultades que tuvieron que ser solucionados para el buen funcionamiento de la plataforma de Ignition, respecto a la comunicación con las unidades de control bajo el protocolo OPC. En este punto se enlistan los problemas más importantes, puesto que los demás han sido solucionados y explicados conforme al documento.

#### 3.11.3.1. Incompatibilidad de versión de software del servidor OPC DA

El servidor OPC de Arduino Ethernet utiliza el módulo OPC Core, para operar con los componentes COM mediante el cual el sistema OPC COM de Ignition detecta el servidor. Esto ha provocado dificultades, con Ignition para encontrar el componente COM correcto que dé con el servidor. Esto se origina por la diferencia de versiones de 32 y 64 bits de los componentes de software.

Como solución, es apropiado que se mantenga la misma arquitectura del sistema operativo, del equipo que soporte las plataformas. Es decir que todo el mecanismo abstracto “*software*” este en la misma versión de 64 bits y en el ámbito de Windows, esto al fin y al cabo conllevará a reducir enormemente, alteraciones en la comunicación o fallos en la detección del servidor OPC que repriman, su estabilidad por incompatibilidad en los procesos.

#### 3.11.3.2. Problemas de visualización de variables OPC DA en el cliente COM Ignition

Este ha sido uno de los problemas que más dificultad ha presentado. Al tener que buscar, analizar y estudiar, todas las cualidades del sistema y del módulo OPC COM de Ignition, para entender la configuración del módulo. Por suerte ha sido facilitada por la empresa **Inductive Automation**, responsable de la fabricación del sistema Ignition que pone a libre disposición del usuario, toda información sobre este producto de software SCADA industrial a través de su página oficial *Inductive Automation*.



La versión OPCSever 1.7, del servidor Arduino Ethernet resulta compatible con el formato OPC DA a pesar de las versiones 2 y 3 que soporta el módulo OPC COM de Ignition. Sin embargo el problema nace al no poder tener acceso a las variables OPC del servidor. Esta restricción es producto de que al utilizar una conexión Ethernet, el browser del Gateway no detecta como un enlace admitido en el sistema, lanzando el siguiente error, que puede observarse en la Figura 3.66.

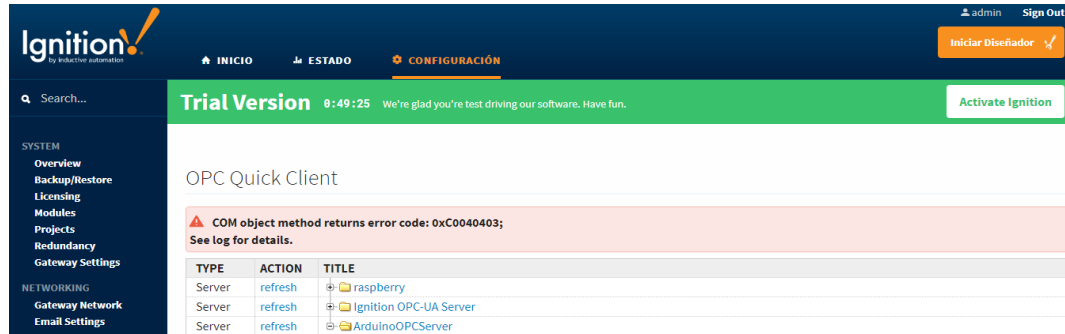


Figura 3.66 Error con el browser de OPC COM Autor ().

Esto se da por utilizar una versión OPC DA 2 en lugar de la versión 3, que le corresponde para admitir la conexión por puerto de red a través del browser OPC COM del portal web. Como solución encontrada, se debe dar premisos para que el Gateway reconozca el enlace de Arduino Ethernet por la red. Esto lleva al uso de las opciones avanzadas, de la interfaz de configuración para conexión del servidor, como lo muestra la Figura 3.67.

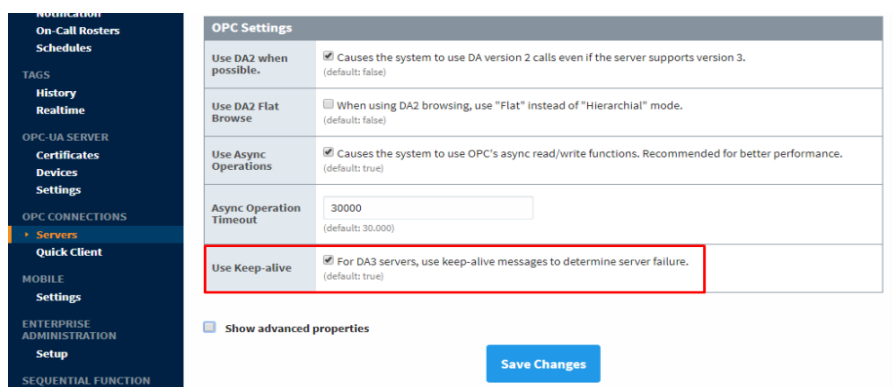


Figura 3.67 Habilitar permiso de conexión por OPC DA 3 Autor ().

Sobre lo visto, vale la pena hacer mención de otras facultades que tiene el sistema Ignition. Dentro del Gateway existe una sección denominada (**Log**), diseñada especialmente para



brindar información detallada, ante problemas detectados en la comunicación de los dispositivos conectados, además de las ventanas de aviso de fallas que proporcionan un resumen del problema.

### 3.12. Diseño y elaboración de la estructura del invernadero

Como estructura del invernadero, se ha planteado que esta sea diseñada en madera, recubierta con láminas de mica, que aislarán el viento y la humedad externa. Esto tiene como propósito brindar la hermeticidad necesaria, reduciendo el impacto de las condiciones climáticas externas. Al ser el recinto impermeable, se consigue tener la maniobrabilidad suficiente de los patrones de humedad y temperatura en el clima interno generado.

No obstante, podrían influir parcialmente ciertas condiciones climatológicas exteriores “como: humedad y temperatura” en el clima en forma de perturbaciones, que el sistema de control automático OPC debe ser capaz de corregirlo, hasta lograr la condición climática deseada según la configuración que se realice. En las Figuras 3.68 y 3.69 se muestra el diseño de la estructura del invernadero con las medidas reales dadas en milímetros (mm).

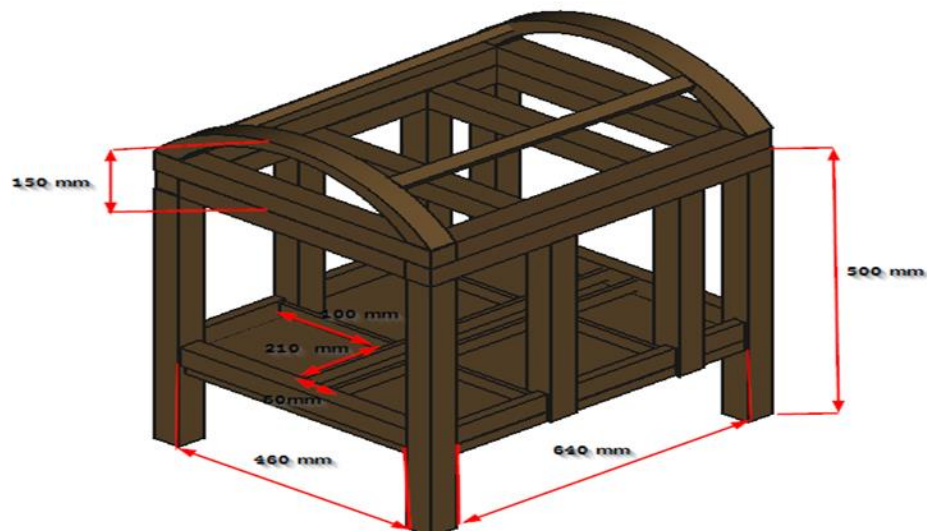


Figura 3.68 Estructura del invernadero a escala real Autor ().



Figura 3.69 Estructura del modelo a escala de invernadero Autor ().

En las Figuras 3.70 y 3.71, se tiene la estructura propuesta con las láminas de mica, para el invernadero, que permiten el paso de la luz solar, que elevan a un cierto nivel el calor en el clima interno. Las variaciones están medidas, por el sensor DHT11 que se ubica dentro del invernadero. Siendo el control de temperatura, el que se encargue de regular el calor transferido, proveyendo un ambiente acogedor para el desarrollo de la vegetación que recibe la luz solar, y se la ha ubicado en el clima interno de la estructura.

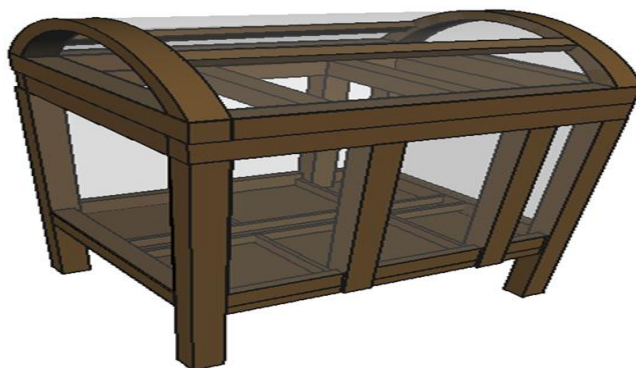


Figura 3.70 Láminas de mica en el invernadero Autor ().



Figura 3.71 Instalación de las láminas de mica en la estructura a escala del invernadero Autor ().



El sistema de ventilación, genera aire fresco en el interior de la estructura. Con el cual se podrá controlar la aeración del clima del invernadero, al regular las altas temperaturas refrescando el ambiente interno. Para ofrecer una mejor renovación del aire, se ha dejado pequeños espacios, en varios rincones de la estructura para permitir el escape del caudal de aire generado hacia el exterior del invernadero, que ayudan también a evitar una excesiva humedad. En la Figura 3.72 se muestra el dispositivo de ventilación junto al sensor DHT.



Figura 3.72 Motor de ventilador de 12 voltios DC y sensor de temperatura DHT11 de 5 voltios DC Autor ().

Para un monitoreo completo en el desarrollo de la vegetación, se tiene instalado un sensor de humedad en el suelo del cultivo, que proporciona información sobre estado del terreno y un sensor que mide la cantidad de agua que recibe la vegetación, para mediciones pluviométricas. En función de los datos que estos provean, el control permitirá inyectar agua o la ventilación en base a los parámetros configurados, para el microclima correspondiente en el invernadero. La Figura 3.73 indica los dispositivos de medición.

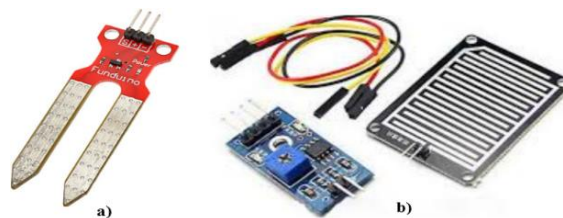


Figura 3.73 a) Sensor de Humedad del suelo y b) sensor de lluvia con circuito amplificador Autor ().

En la próxima Figura 3.74 se muestra el invernadero armado en su totalidad, con una sola abertura de ingreso que sirve como puerta, situada en la parte superior dando la forma de capsula a la estructura, permitiendo la observación del proceso interno referente al control climático y monitoreo de la vegetación que acoge el sistema OPC.

Las vigas superiores en la estructura, permite la instalación de un elemento de radiación luminosa (foco), utilizado como fuente de calor artificial, que incrementa la temperatura del

microclima interno del invernadero. Los segmentos laterales del espacio de separación mostrados entre un cultivo y otro, son utilizadas para instalar los conductos que llevan el agua proveniente de una bomba, que genera una fuerza de succión para extraer el agua de un depósito, con la cual se puede predisponer su control para inyectar pequeños volúmenes de agua, que regulan la humedad del terreno y del ambiente interno del invernadero.

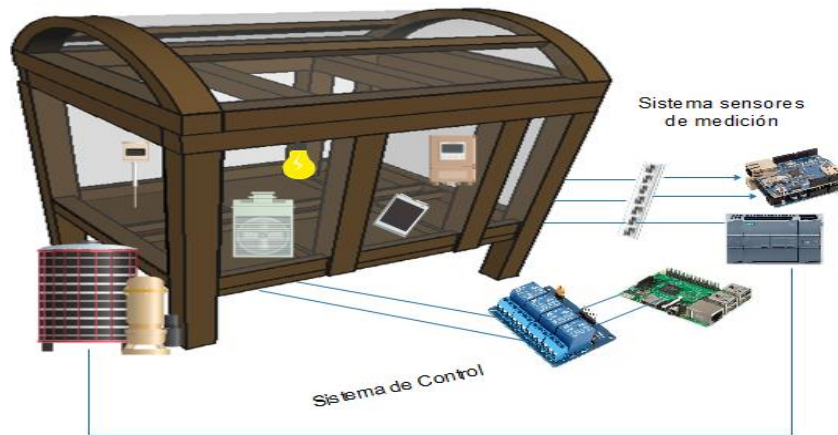


Figura 3.74 Invernadero con todos los elementos de control y monitoreo Autor ().

En las siguientes imágenes se observan, los trabajos realizados para la construcción física del invernadero, como aplicación de este proyecto, con el producto final acabado. Los elementos y fuente artificial, han sido incorporados para generar un ecosistema controlable en el invernadero, sirviendo para conocer y evidenciar la velocidad de respuesta de los elementos de control, de los procesos que asumen las variables OPC de los servidores al compartir información con el sistema de control y monitoreo OPC mediante Ignition desarrollado como aplicación SCADA de cliente OPC UA.

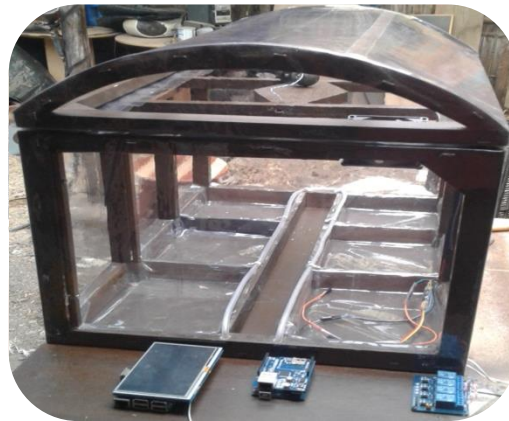


Figura 3.75 Instalación de equipos y sensores en el modelo a escala de invernadero Autor ().



Figura 3.76 Modelo a escala terminado Autor ().



Figura 3.77 Vista del sistema de ventilación y fuente de calor Autor ().



Figura 3.78 Vista de conexión de controladores OPC a la red LAN y dispositivos de captación del invernadero Autor ().



## Capítulo 4

### Evaluación y validación del prototipo SCADA

---

#### 4.2. Pruebas de monitoreo y funcionamiento

En el presente apartado se muestra la interface, que observará el usuario final de la aplicación prototipo de invernadero. Cuyas pruebas con el sistema SCADA Ignition, como cliente OPCUA comunicado con los servidores OPC UA y OPC DA, se dieron en base a dos pretensiones. La una prueba mediante la interfaz gráfica para el control y monitoreo de los dispositivos y procesos y la otra mediante pruebas de conexión del sistema OPC. Estas pruebas se describen brevemente en las siguientes subsecciones:

##### 4.2.1 Interface del sistema de control SCADA OPC

La interfaz muestra, las pantallas y elementos que han sido objetos primordiales en la dinamización. En la aplicación desarrollada se gestiona los procesos, dando lugar a la interoperabilidad, en base a los controles presentados que hacen la presentación de la interfaz en cuestión.

El control principal de proceso mediante OPC, se lleva a cabo en la interfaz *gráfica de monitoreo de los procesos*. Está pantalla, es la principal y cuenta con todos los controles indispensables para una efectiva gestión, asociando a cada unidad de control la representación de un proceso que le compete. A esto también se presentan todos los actuadores con las opciones de poder intervenirlos de forma manual y automática, así como también controles de aviso de alarmas y cuadros de visualización numérica de tiempo de ejecución autónoma del sistema de gestión.

En la Figura 4.1 se muestra el sistema SCADA basado en OPC con todos sus atributos para la administración del proceso, constando de controles *Easy Chart* configurados con los parámetros subyacentes al proceso y estados de operación en tiempo de ejecución.



Figura 4.1 Sistemas SCADA OPC, interface de monitoreo y control Autor ().

La interfaz dispone de varias secciones, de forma organizada y dedicadas a diferentes áreas operativas del proceso, y en las cuales se pueden configurarse el comportamiento del sistema y analizar la información, que provee los factores climáticos en el ambiente del invernadero.

Así la pantalla *ajustes de proceso*, permite el entrenamiento remoto del sistema mediante la configuración de respuesta de los parámetros de control, al modificar la configuración de las variables actuadoras del proceso. Dado que cuenta con controles de *seteo* que el usuario puede ajustar a su criterio, aconteciendo los parámetros característicos del proceso, “*humedad, temperatura y precipitación*”, permitiendo cambiar el funcionamiento del proceso hacia una nueva respuesta de control. En la Figura 4.2 se muestra la interfaz de ajustes de proceso.



Figura 4.2 Interfaz Ajustes de Proceso para automatismo de control OPC Autor ().



Las tendencias gráficas son un componente relevante del proceso. La trazabilidad de las condiciones climáticas del ambiente dentro del invernadero, hace posible la comparación de sucesos positivos o negativos en el desarrollo del cultivo, ante las condiciones ambientales en un intervalo de tiempo definido.

La *ventana de gráficas*, permite realizar consultas históricas, almacenados en la base de datos de manera que se puede conocer el comportamiento que mantiene el ambiente en un periodo específico, brindado información de importancia al operador del proceso, contribuyendo a mejorar su intervención, disponiendo de un menú de selección de ventana de tiempo de forma histórica y real, mediante el cual se realiza el reporte de las variaciones medidas, la cual se expone en la Figura 4.3.

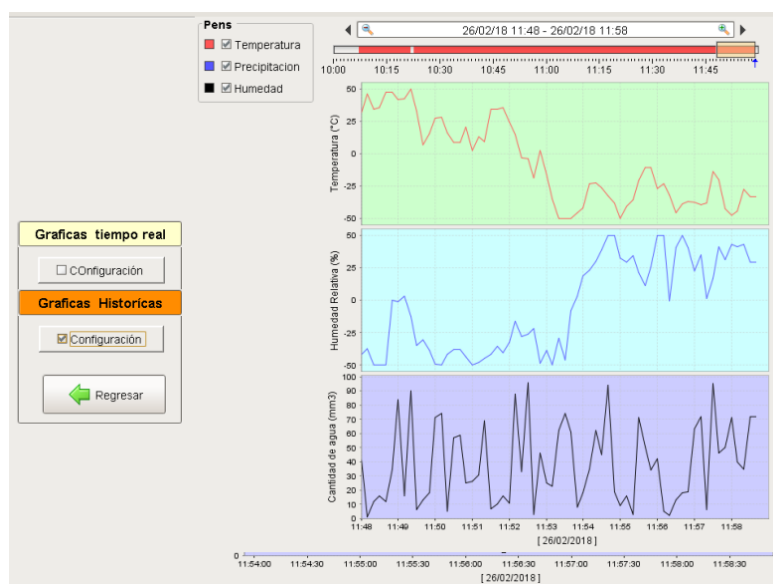


Figura 4.3 Ventana de graficas de proceso OPC Autor ().

Otro aspecto importante en el monitoreo del proceso, es el reporte de novedades y registro de eventos, con respecto a la condición climática y estado del cultivo en el interior del invernadero. Esto permite corroborar con situaciones que han afectado el proceso al producirse en un instante de tiempo determinado en las condiciones del ambiente, dándole aviso al operario y que este pueda tomar las medidas correctivas pertinentes. A través de la ventana de alarmas se exponen las incidencias, y condiciones del proceso, ante las alteraciones que suscitan sus parámetros. Está consta de controles de estado *ON/OFF* de las

alarmas para la activación, y configuración respectiva para condiciones climáticas especiales. En la Figura 4.4 se muestra la interfaz de las alarmas del proceso.

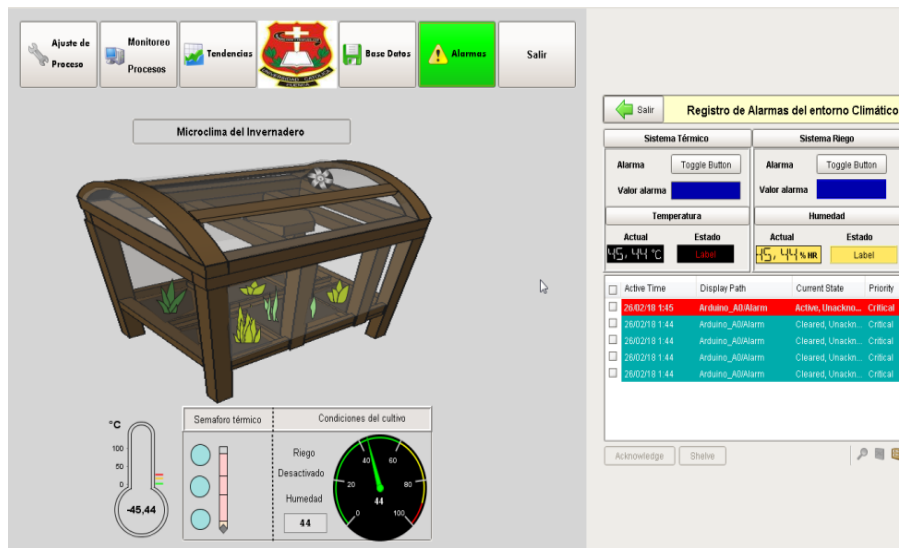


Figura 4.4 Ventana de registro de alarmas del SCADA OPC Autor ().

#### 4.2.1.1. Pruebas de funcionamiento de la aplicación de control SCADA

En la realización de las pruebas se definieron parámetros de configuración que puede ser visto en la Tabla 4.1. Con esto se asegura que el sistema se comporte apropiada y oportunamente, haciendo que los parámetros de temperatura y humedad de la superficie estén dentro de los límites tolerables del proceso, para un seteo de control automático de las variables definido, para esta aplicación corresponde al siguiente.

Tabla 4-1 Valores de configuración de seteo automático del sistema

Temperatura	Humedad	Alarmas	Tiempo Activo
12 °C	60 % HR	15 °C	1 minuto

Fuente: Autor ().

Basado en estos valores, el sistema actuara en el proceso, elevando la temperatura del clima hasta las 12 °C, y ante la humedad de la superficie del cultivo del 60% activara el sistema de riego. Ejecutando todas las funciones de control por el lapso de un 1 minuto, cuando las condiciones planteadas estén dadas. En el caso de que temperatura alcanzase los 15 °C se activa una alarma, la cual expone un cuadro de aviso de regulación térmica con el sistema de ventilación que el operario accionara para validar la alarma.



Para probar la operación del sistema, se ejecutaron los diferentes servidores OPC con el cliente OPC UA de Ignition. La Figura 4.5 muestra la interfaz de comunicación del servidor OPC de Raspberry con sus variables definidas en los puertos GPIO, como también el estado y dirección de conexión al servidor bajo el enlace OPC UA.



Figura 4.5 Servidor OPC UA Raspberry conectado al sistema SCADA OPC Autor ().

Del mismo modo en la Figura 4.6 se indica la comunicación del servidor OPC Arduino con las variables de plataforma bajo el enlace OPC DA.

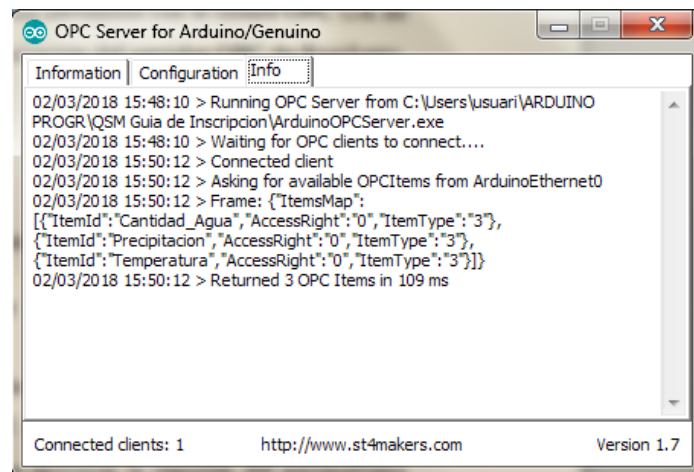


Figura 4.6 Servidor OPC DA de Arduino Ethernet conectado al SCADA OPC Autor ().

Las Figuras 4.7 y 4.8 muestran la realización de las pruebas, utilizando para la simulación de la energía solar un foco incandescente de 100W de potencia. Mediante el cual se generó el calor que necesita el sistema como estímulo para provocar la reacción del automatismo del sistema OPC. Haciendo evidente la administración del SCADA con los diferentes sistemas de control que funcionan con distintos procesos.



Figura 4.7 Vista del invernadero con las pruebas de control automático del SCADA Autor ().



Figura 4.8 Vista de los elementos de accionamiento y sensores en el interior del invernadero conectados al servidor SCADA OPC Ignition Autor ().

Como se ha observado en las imágenes anteriores, el SCADA OPC emula la operación del proceso real en su entorno virtual con cada dispositivo de control. De algún modo esto facilita la identificación rápida ante fallas en el proceso o problemas en el funcionamiento en un equipo de control ofreciendo una interfaz netamente interactiva y de fácil manejo al operador.

En la Figura 4.9 se indican los resultados obtenidos con los datos de humedad, temperatura y precipitación [*Cantidad de agua*], almacenados en la base de datos y en la Figura 4.10 se muestra el progreso de la medición histórica y en tiempo real, de la sección *base de datos*,



mediante el sistema SCADA. En esta última se dispone de controles, que permite compartir la información con otros sistemas de análisis más críticos como Excel para reportes.

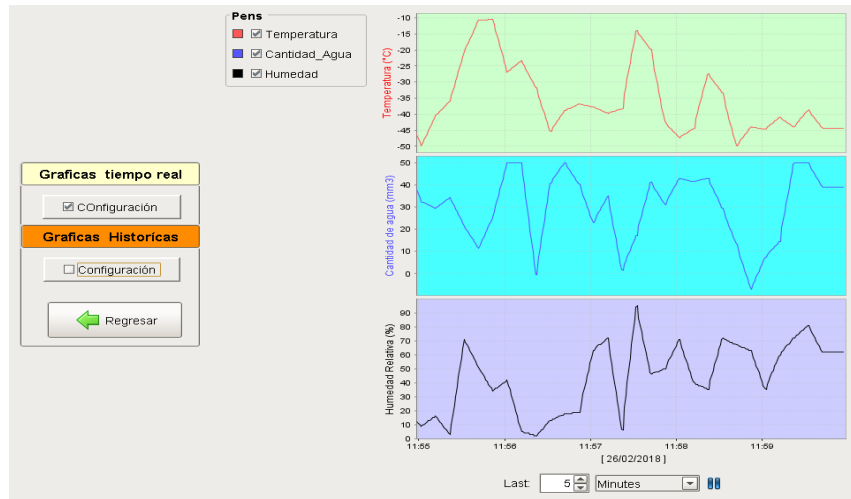


Figura 4.9 Respuesta del sistema para la humedad de 60% y temperatura de 12 ° C Autor ( ).

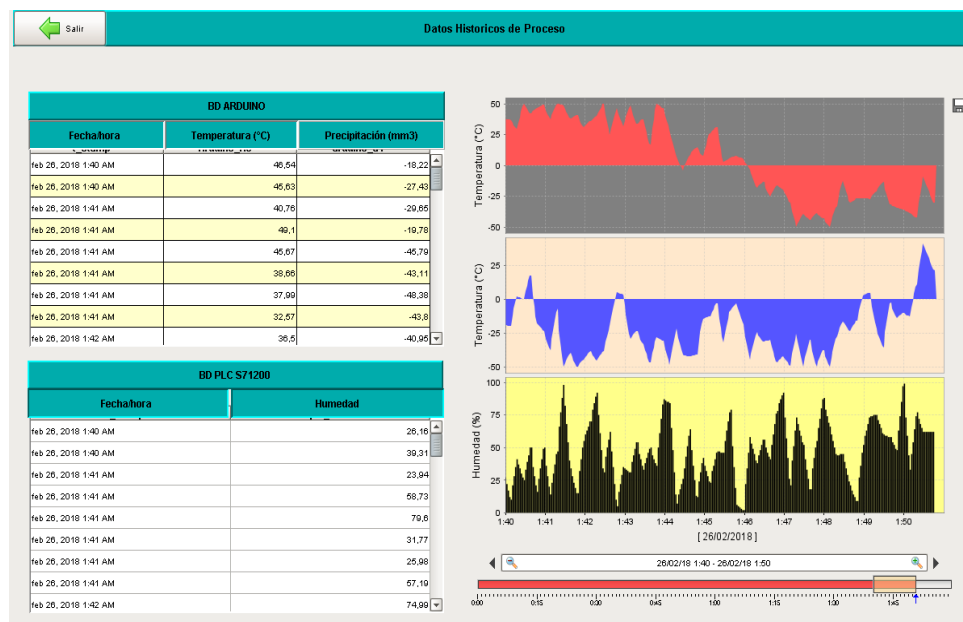


Figura 4.10 Almacenamiento de los datos en MYSQL en tiempo real Autor ( ).

### 4.3. Pruebas y verificación de resultados de conexión del sistema OPC

Esta sección comprende los mecanismos de verificación del sistema OPC, del SCADA y los servidores, a fin de comprobar que la arquitectura de comunicación proyectada, este trabajando bajo el protocolo OPC mediante la captura y análisis del tráfico generado en la red local. Así mismo se realizan pruebas de transmisión de datos a través de estructuras de

mensajería simple (ICMP), con la cual se puede determinar la accesibilidad a las interfaces del SCADA y la congestión de tráfico en el enlace. Todos estos ensayos de comunicación se realizan con el sistema SCADA OPC activo y operando.

### 4.3.1 Pruebas de enlace mediante OPC

La plataforma Ignition componelos enlaces hacia los servidores OPC, en un solo formato de protocolo a través del Gateway, el sistema idóneo fija el enlace OPC UA porque facilita y flexibiliza la comunicación con los distintos dispositivos. El traspaso de OPC DA a UA, lo realiza mediante los módulos del servidor Ignition, así evade las dificultades de conexión del COM de OPC DA, al conjuntarlo permitiendo la interacción de enlaces OPC UA.

En la Figura 4.11 se aprecia la conversión, del enlace OPC clásico a UA y resto de enlaces viéndose necesario, para entender el contexto de comunicación OPC del sistema Ignition.

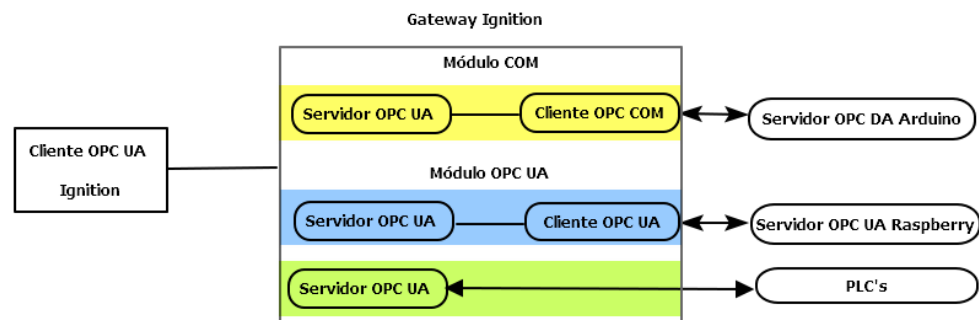


Figura 4.11 Arquitectura de comunicación con los módulos del Gateway, para los servidores OPC heredados (OPC UA y OPC DA) y servidor OPC UA del PLC Autor ().

Como se aprecia en el esquema. Cada módulo tiene un servidor OPC UA con su respectivo cliente interno, mediante el cual adquiere los datos y es compatible con los diferentes servidores OPC externos al Gateway. Excepto con el PLC ya que la conexión al servidor es directa. Así es como se origina y utiliza el cliente OPC UA y se ha podido verificar la prioridad y prevalencia de este estándar que trabaja con el sistema OPC Ignition.

Para corroborar con esta comunicación OPC UA se hizo la captura de paquetes del tráfico generado, esto fue posible gracias a que el protocolo basa su comunicación mediante **TCP/IP**, que puede capturarse en la red local por medio del software *WireShark*.



Wireshark es una herramienta de monitoreo de redes que permite capturar cualquier tipo de tráfico. Disponiendo de varios filtros predeterminados para diferentes estándares de protocolos entre estos OPC UA. Claro está que para eso, se requieren de las siguientes configuraciones para el seguimiento del tráfico de entrada y salida del sistema OPC:

- Se configura el puerto del protocolo OPC UA, para escuchar el tráfico del servidor – cliente al fijar la conexión, capturando todas las llamadas del servicio en la red. En este caso el puerto 4840 del servidor OPCUA Raspberry. Respecto al cliente de Ignition y resto de servidores el sistema establece y comparte por defecto el puerto 4841 entre ambas plataformas. Esta configuración se indica en la Figura 4.12.

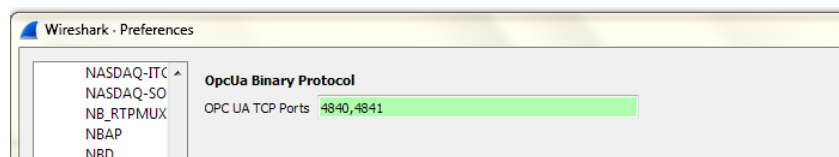


Figura 4.12 Selección de puerto y protocolo OPC UA con WireShark Autor ().

- Se comienza la captura del tráfico, en el cual se obtiene normalmente todo tipo de datagrama o tramas con distintos protocolos. Al aplicar filtros OPC UA a los paquetes obtenidos se obtiene las comunicaciones basadas en este protocolo para su análisis, seleccionando el tipo de captura en este caso por la interfaz Ethernet.

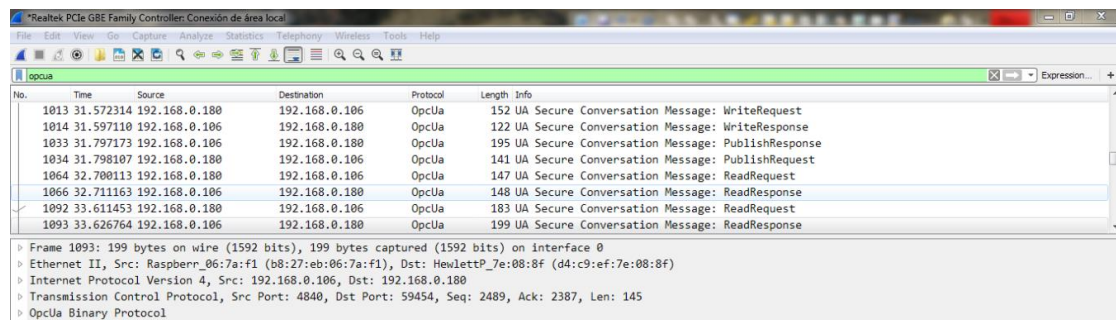


Figura 4.13 Captura de datos en el tráfico de la red local con el filtro OPC UA Autor ().

Cuando WireShark admite la información del tráfico engendrado, captura los paquetes UA de cada parámetro para visualizar su contenido en el registro, mostrando varias opciones de análisis e información del paquete al filtrarlo, como lo indica la Figura 4.13. De cliente a servidor la comunicación es bidireccional. La Figura 4.14 muestra la acción de escritura en un nodo del servidor de Raspberry al cual suscribe el cliente OPC UA Ignition.

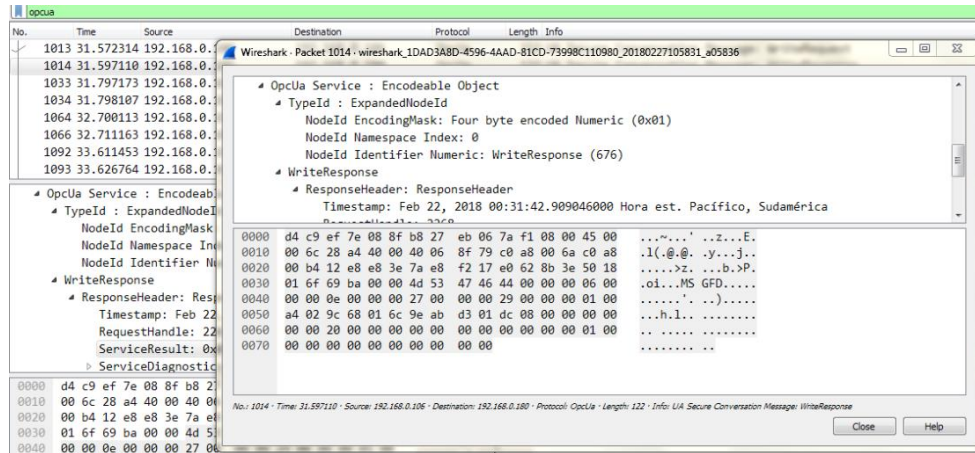


Figura 4.14 Revelación de información en un paquete OPC UA Autor ().

### 4.3.2 Prueba de enlace por transmisión de mensajería simple ICMP

El objetivo de estas pruebas, es establecer la disponibilidad del enlace y congestión del tráfico dentro de la red por TCP/IP. Tomando como base los tiempos de respuesta, ante los datos transmitidos, hacia el sistema SCADA y en los servidores OPC con las unidades de control para la concurrencia de nuevos clientes OPC a los servidores o los usuarios al conectarse a dicho sistema.

La prueba se lleva a cabo mediante comandos de “ping”. Desde el terminal de origen donde se halla el servidor Web de la aplicación SCADA OPC hasta el lugar donde se encuentran vinculados a la red LAN los controladores Arduino Ethernet, Raspberry Pi y PLC S71200 contenidos en los respectivos servidores OPC, que administran los procesos de operación.

La prueba de mensajes simple ICMP, por medio de enlaces definidos para los distintos componentes de gobierno, en el invernadero con el servidor Web de la aplicación en el cual se realiza la consulta y entrega de paquetes, se ha determinado un tiempo promedio de entrega de la información de un paquete de **128 ms**, esto según siempre y cuando el tráfico de envió de paquetes se haga vía OPC en el mismo enlace. Como se observa la entrega de paquetes a través de la red no tiene saturación ni interrupción, siendo él envió de paquetes rápido y consistente en la arquitectura del sistema SCADA OPC.



Con esta interfaz del SCADA puede ser vista y ejecutada sin mayor problema, también en plataformas móviles basados en un entorno JAVA. Bajo las condiciones de operación de la aplicación con el proceso, ante la disposición de un medio de enlace óptimo que evade los tiempos de espera extensos o fallos en la conexión de la red, agilitando la administración de los procesos en el invernadero y de la aplicación misma. Poniendo en eminencia las funcionalidades y capacidades de la plataforma con equipos de bajo nivel de procesamiento y en entornos multiplataforma y lo que es más la gestión remota a través de la web.

La Figura 4.15 muestra la aplicación ejecutada desde un dispositivo móvil, en un entorno Android compatible con el sistema Ignition mediante el módulo *Mobile* de la plataforma y también desde un entorno Windows. Digitando la dirección *IP* del servidor y el puerto de conexión, *vía web* se accede al sistema en este caso en particular *192.168.0.100:8088*.

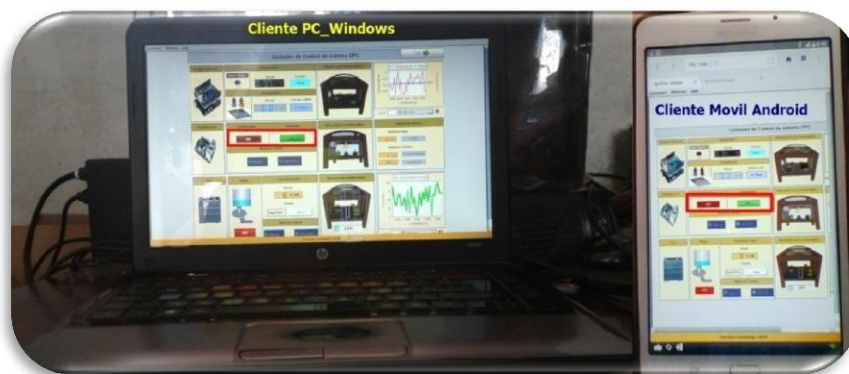


Figura 4.15 Escritura en las variables de los servidores con el SCADA OPC Ignition desde dos clientes en entornos Android y Windows equipos con un navegador web Autor ().

Cualquier modificación realizada en las variables o nodos en los servidores, repercute inmediatamente en todos los clientes conectados. De igual modo un cambio, desde un cliente se ve reflejado, tanto en los servidores como en el resto de clientes según lo indicado en la figura anterior.

Las pruebas llevadas a cabo han sido satisfactorias, respecto a la conexión permitiendo analizar y comprender la transmisión de paquetes de datos. Concluyendo con ello que el sistema SCADA, efectivamente está funcionando y operando con el protocolo OPC concretamente con la estribación UA, en ambos servidores externos y con el servidor



interno del sistema Ignition. Al mismo tiempo, se evidencia la eficacia del sistema cumpliendo con los objetivos al trabajar en diferentes dispositivos a la vez.

#### 4.3.3 Análisis de resultados

Las comprobaciones vertidas anteriormente en el sistema OPC, son realizados para conocer el grado de disponibilidad de la aplicación para los usuarios en la red local, mediante análisis de tráfico y rutas de acceso, así también su funcionamiento mediante OPC. De ello se han obtenido los siguientes resultados:

- Se ha confirmado que el tráfico del sistema SCADA basado en el modelo cliente – servidor se produce con el estándar OPC específicamente la derivación OPC UA.
- Al tener menor tasa de tráfico el servidor SCADA puede soportar un mayor número de peticiones de los usuarios desde varias plataformas.
- La comunicación y respuesta de los dispositivos gobernados por el SCADA OPC es casi al instante. La lectura y escritura de variables ha funcionado en base a los principios y requerimientos otorgados por el protocolo OPC UA y OPC DA.
- Cualquier cambio en los nodos OPC de los servidores, han repercutido inmediatamente en todos los clientes conectados, operando con distintos dispositivos.
- La concurrencia de nuevos clientes OPC en los servidores dentro de la arquitectura no es del todo admitida, solo unos cuantos. Así también con la transferencia de datos se ha asegurado un control apropiado del proceso.
- En una de las últimas pruebas de conexión se tuvo la desconexión de las variables de los servidores, esto se produjo al terminarse el tiempo de 2 horas de uso bajo las capacidades básicas del sistema Ignition. Los sistemas embebidos Arduino y Raspberry intentaron recuperar la conexión pero, esto solo fue posible al tener que reiniciar el proceso de comunicación. Cabe destacar que la plataforma mantuvo un funcionamiento normal sin inconveniente alguno dentro de ese lapso de tiempo.



## Capítulo 5

### Conclusiones y recomendaciones

---

#### 1.1. Conclusiones

Los sistemas OPC, constituye un pilar fundamental en la integración y convivencia de diferentes tecnologías y de una nueva perspectiva de conceptos de aplicaciones industriales. La arquitectura de comunicación OPC refleja la importancia a la hora de simplificar la integración de otras aplicaciones como HMI, SCADA y sistemas MES del ámbito industrial.

El sistema OPC SCADA desarrollado, con la ayuda de la plataforma Industrial Ignition ha permitido corresponder eficiente con la integración, de un PLC y servidores OPC para Arduino Ethernet y Raspberry Pi. De tal forma que dichos dispositivos de control han podido interconectarse e interoperar en conjunto, controlando la operatividad y monitoreo de diversos procesos por medio de una red local (LAN).

Referente al software se ha tratado de enfocarse, en distribuciones libres de tal manera que los componentes de programación, no representen ninguna restricción, en cada servidor OPC. Con ello las tecnologías implantadas como alternativas, están basadas y desarrolladas con librerías de código abierto, operando bajo sus lenguajes nativos como lo es C++ y Python correspondientemente, logrando su ejecución en cualquier plataforma y la obtención de un sistema Open Source en su totalidad.

Al implementar el sistema propuesto, se ha podido determinar grandes beneficios del estándar de comunicación OPC, ante la convergencia y versatilidad de diferentes dispositivos y sistemas, usados para la automatización siendo equipados con el protocolo;

- Como la portabilidad, de poder llevarla a aplicaciones más concretas y una oportunidad interesante para el diseño de sistemas escalables, mediante la integración de tecnologías, de modo que cada nueva implementación pueda



ocuparse de un área específica, con una inversión final bastante accesible. Por el módico costo que tienen los materiales, software, licencias de sistemas operativos y bases de datos.

- Brinda oportunidades de nuevos emprendimientos, que a largo plazo pueden ser provechosos y de menor costo, al comparar nuevas alternativas con el escenario donde disponer de un PLC simplemente resulta, demasiado costoso y muchas veces se sobredimensiona su utilidad y aprovechamiento al máximo, ante pequeñas aplicaciones sobre todo si se tratase con un sistema HMI resulta en extremo costoso.
- La base datos ha permitido, un mayor control de los procesos desarrollados principalmente usados para la medición, almacenando las variaciones de las variables, con lo cual a través de la interfaz gráfica del sistema el usuario puede informarse de los cambios su citados, a través de una ventana de reportes histórica de una fecha seleccionada.
- La factibilidad de operación multiplataforma, que no es admisible para otros sistemas SCADA similares, es debido a las facultades y versatilidad de los softwares y hardware estandarizados.

A pesar de que se consideró el caso específico de un invernadero. La arquitectura OPC de este sistema puede ser aplicado en realidad en cualquier tipo de proceso, usando los mismos recursos en lo que respecta a tecnologías de la información y comunicación (TIC) y bajo el mismo esquema. Teniendo una operación similar, inclusive de utilizar el mismo servidor de aplicaciones para varios procesos.

Se ha observado que el grado de comunicación, con esta arquitectura y cantidad de dispositivos y procesos es adecuado, sin ser un problema los retardos en la transmisión de los paquetes. Aunque es preciso considerar evaluaciones más prolijas, para establecer el grado óptimo de conexiones, en el canal de datos antes de la saturación de la red con un mayor número de controladores, procesos y peticiones de clientes. Con estas conclusiones se logró verificar y validar la hipótesis planteada, al diseñar una arquitectura supervisora versátil para diferentes protocolos nativos de comunicación en varias unidades de gobierno.



## 5.2. Recomendaciones

Se tiene en cuenta las siguientes recomendaciones, con la finalidad de obtener mejores resultados en el prototipo realizado y en su implementación:

- Conforme a la sección anterior, se observó la posibilidad de utilizar la misma arquitectura OPC, en otros procesos debido a que las unidades de control son reprogramables. Sin embargo, se debe investigar bien todas las características y requerimientos, respecto al software de programación de tal manera que el sistema se acople adecuadamente.
- Si bien el sistema OPC SCADA es efectivo en a nivel local, su funcionalidad puede ser mejorado al expandirse, a áreas de cobertura mucho más grandes como internet. Actualmente se habla mucho del internet de las cosas industriales “IIOT”. De conseguir una conexión en línea, un operario tendrá acceso y podrá controlar todos los procesos del sistema desde cualquier lugar y en cualquier instante. Para cumplir este propósito, es primordial construir mecanismos de seguridad más restringidos, como establecer enlaces vía túneles VPN (Virtual Private Network).
- En el tiempo de diseño de este prototipo, se apreció las prestaciones y capacidades multiplataforma del sistema SCADA Ignition. En este aspecto se ha visto una oportunidad, para soportarlo en el sistema embebido Raspberry aunque se encuentra limitado en capacidad de procesamiento. No obstante con el adelanto tecnológico actual, dispone de una versión Raspberry Pi de tercera generación con capacidades superiores de procesamiento. La ejecución del servidor Ignition en esta plataforma, traería como beneficio abaratar aún más el costo final de la aplicación.
- Una forma de mejorar las funcionalidades de comunicación del sistema, es implementando un servidor OPCUA para Arduino. Por la gran dificultad de conectar la versión OPC DA basada en el COM/DCOM de Windows. Con esto la arquitectura será más versátil y permitirá con facilidad la adaptación de nuevas tecnologías, incluso acoplar protocolos de comunicación de IoT como MQTT.



---

## Bibliografía

---

- Alfonso, V., Noguera, B., & Valencia, J. F. (2015). Servidor OPC UA. *Protocolo OPC UA*. Retrieved from <https://riunet.upv.es/bitstream/handle/10251/56295/VILLARROYA-SERVIDOR-OPC-UA-SOBRE-BEAGLEBONE-BLACK.pdf?sequence=2>
- Amangandi, J. (2012). ¿Qué es arduino? Retrieved from <https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>
- Architecture, O. P. C. U. (2013). OPC Unified Architecture – La plataforma de comunicación universal para modelos estándar de información . *OPC, 1.1*, 2. Retrieved from [https://opcfoundation.org/wp-content/uploads/2014/05/OPC-UA\\_CollaborationOverview\\_ES.pdf](https://opcfoundation.org/wp-content/uploads/2014/05/OPC-UA_CollaborationOverview_ES.pdf)
- Arencibia, S. N. (2016). OLE-DDE-OPC. Retrieved from [http://www.iuma.ulpgc.es/~avega/int\\_equipos/trab9899/ole\\_dde\\_opc/index#\\_Toc409073150](http://www.iuma.ulpgc.es/~avega/int_equipos/trab9899/ole_dde_opc/index#_Toc409073150)
- Carolina Lagos. (2016). Revista Electroindustria - Protocolos de Comunicación Industrial. Retrieved from <http://www.emb.cl/electroindustria/articulo.mvc?xid=562>
- Castañó Gómez, C. A., Villamarin Meneses, E. F., & Rojas Alvarado, O. A. (2015). Principios de aplicación del estándar OPC UA en un caso de estudio. *Revista Vector*, 8. Retrieved from [http://www.unicauca.edu.co/ai/publicaciones/Estudiantes/Castano\\_Villamarin.pdf](http://www.unicauca.edu.co/ai/publicaciones/Estudiantes/Castano_Villamarin.pdf)
- CCNA. (2015). Modelo OSI: la guía definitiva para aprender el Modelo OSI !!! Retrieved from <http://www.seaccna.com/modelo-osi-guia-definitiva/>
- César Caicedo-Eraso, J., Rocío Varón-Serna, D., & Octavio Díaz Arango Profesor, F. (2015). Redes industriales. *Revista Vector*, 7 (7), 12–17. Retrieved from [http://vip.ucaldas.edu.co/vector/downloads/Vector7\\_3.pdf](http://vip.ucaldas.edu.co/vector/downloads/Vector7_3.pdf)



- Collantes, M. H., & Padilla, A. L. (2015). Protocolos y seguridad de red en infraestructuras SCI. Retrieved from [https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/incibe\\_protocolos\\_seguridad\\_red\\_sci.pdf](https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/incibe_protocolos_seguridad_red_sci.pdf)
- Conectrónica. (2013). Redes Profinet: Generalidades - Conectores-Redes-Fibra óptica-FTTh-Ethernet. Retrieved March 15, 2018, from <https://www.conectronica.com/redes-industriales/redes-profinet-generalidades>
- Cpu, L. (2013). Autómata Siemens S7-200 AC / DC / RLY, 1–11. Retrieved from [http://isa.uniovi.es/docencia/ra\\_marina/cuatrim2/Temas/s7200.pdf](http://isa.uniovi.es/docencia/ra_marina/cuatrim2/Temas/s7200.pdf)
- David, I., & Aguirre Grazio, J. (2017). SISTEMAS DE CONTROL AUTÓMATAS PROGRAMABLES P L C, 24. Retrieved from [https://catedras.facet.unt.edu.ar/sistemasdecontrol/wp-content/uploads/sites/101/2017/05/4\\_Autómatas-Programables\\_2017.pdf](https://catedras.facet.unt.edu.ar/sistemasdecontrol/wp-content/uploads/sites/101/2017/05/4_Autómatas-Programables_2017.pdf)
- Eben Upton, & Gareth Halfacree. (2013). *RaspberryPiGuíaDelUsuarioParteIyIIFull.pdf*. *Raspberry Pi guía del Usuario*. Buenos Aires . Retrieved from [http://190.24.150.73/seda/seda\\_wp\\_00/wp-content/plugins/pdfjs-viewer-shortcode/web/viewer.php?file=http://190.24.150.73/seda/seda\\_wp\\_00/wp-content/uploads/pdftutorials/RaspberryPiGuíaDelUsuarioParteIyIIFull.pdf&download=true&print=true&openfile=false](http://190.24.150.73/seda/seda_wp_00/wp-content/plugins/pdfjs-viewer-shortcode/web/viewer.php?file=http://190.24.150.73/seda/seda_wp_00/wp-content/uploads/pdftutorials/RaspberryPiGuíaDelUsuarioParteIyIIFull.pdf&download=true&print=true&openfile=false)
- Electrin. (2016). Estructura básica de un PLC |. Retrieved October 26, 2017, from <https://electrinblog.wordpress.com/2016/04/24/post-2/>
- Ermesh. (2016). El modelo OSI - Parte 1: Aspectos generales - Ermesh. Retrieved from <http://www.ermesh.com/modelo-osi-parte-1-aspectos-generales/>
- Espinoza, Z. C. R. (2014). “redes industriales para la calidad y la manufactura esbelta (lean) .” *Redes Industriales*, 113. Retrieved from <http://www.bib.uia.mx/tesis/pdf/015895/015895.pdf>



- ETI, C. I. . (2013). Opc: un estandar en las redes industriales y buses de campo. Retrieved from <http://www.etitudela.com/entrenadorcomunicaciones/downloads/labviewintroduccionopcserver.pdf>
- Fernando, A., & Aníbal, A. (2015). PLANTA INDUSTRIAL A ESCALA ACADÉMICA PARA EL ESTUDIO DE CONCEPTOS Y APLICACIONES EN REDES DE CONTROLADORES (PLCs) Y REDES DE CAMPO INDUSTRIAL, 98. Retrieved from <https://www.dsi.fceia.unr.edu.ar/images/downloads/CIM/ProyectoComunicacionesIndustriales.pdf>
- Fernando Doutel. (2015). Guía del Arduinomaníaco: todo lo que necesitas saber sobre Arduino. *Ww.Xataka.Com*. Retrieved from <https://www.xataka.com/especiales/guia-del-arduinomaniaco-todo-lo-que-necesitas-saber-sobre-arduino>
- Gabriel Fernandez Perez. (2015). PROFIBUS | INFORME. Retrieved from <http://informefebrerotanquehidraulico828826.blogspot.com/2015/09/profibus.html>
- Hilariona Martínez. (2015). *Modulo didáctico para prácticas de laboratorio con controladores lógicos programables*. Universidad Autónoma De Nuevo León. Retrieved from <http://eprints.uanl.mx/9535/1/1080214939.pdf>
- Hurtado, J. M., & Página, T. (2014). TUTORIAL REDES PROFIBUS-DP. Retrieved from [http://www.infoplcn.net/files/descargas/siemens/infoPLC\\_net\\_tutorial-redes-profibus-dp.pdf](http://www.infoplcn.net/files/descargas/siemens/infoPLC_net_tutorial-redes-profibus-dp.pdf)
- Ingenieriatric. (2017). Redes Industriales - Página web de ingenieriatric. Retrieved from <https://www.ingenieriatric.com/redes-y-comunicaciones/redes-industriales/>
- Karla Gaona Hez. (2015). PROFIBUS | Mecatronica. Retrieved from <http://karlagaona.blogspot.com/2015/09/profibus-introduccion-el-bus-de-campo.html>
- Karla Gaona Hez. (2016). INFORME:Ethernet | Mecatronica. Retrieved March 15, 2018,



- from <http://karlagaona.blogspot.com/2016/01/informeethernet.html>
- M. Moreno. (2015). Controlador Lógico Programable. *Automación Micromecánica S.a.i.c*, I, 84. Retrieved from <http://www.microautomacion.com/capacitacion/Manual061ControladorLgicoProgramablePLC.pdf>
- M2MTechnologies. (2013). Ignition : El nuevo SCADA | MIOS. Retrieved March 16, 2018, from <http://mios-group.com/es/product/uncategorized/ignition-el-nuevo-scada/>
- Medina, R. (2015). SISTEMA SCADA. Retrieved from [http://www.academia.edu/5635397/SISTEMA\\_SCADA](http://www.academia.edu/5635397/SISTEMA_SCADA)
- Mitsubishi. (2014). PLC: Conceptos Básicos, 52. Retrieved from [http://www.mitsubishielectric.com/fa/assist/e-learning/pdf/spa/1-GX\\_Works2\\_Basics\\_fod00064\\_spa.pdf](http://www.mitsubishielectric.com/fa/assist/e-learning/pdf/spa/1-GX_Works2_Basics_fod00064_spa.pdf)
- National Instruments. (2014). Información Detallada sobre el Protocolo Modbus - National Instruments. Retrieved from <http://www.ni.com/white-paper/52134/es/>
- NERGIZA. (2015). Raspberry Pi 2 B: ¿Qué es y para que nos puede servir? | Nergiza. Retrieved from <https://nergiza.com/raspberry-pi-2-b-que-es-y-para-que-nos-puede-servir/>
- OPC FOUNDATION. (2014). OPC DATA ACCESS. Retrieved from <http://isa.uniovi.es/~vsuarez/Download/opc.pdf>
- Paul Lara. (2017). ¿Sabes qué es Arduino y cómo funciona? Retrieved from <https://www.telcelsoluciones.com/articulos/conoce-arduino-la-placa-que-cambiara-tu-vida>
- Pérez López, E. (2015). SCADA systems in the industrial automation. *Tecnología En Marcha*, 28 (4), 3–14. <https://doi.org/10.18845/tm.v28i4.2438>
- Poma, D. V. (2013). Interfaces y Protocolos de Comunicación. Retrieved from



<http://tecsup-r5-ac-interfaces-y-protocolos.blogspot.com/>

Raúl Cobo. (2015). Revista Electroindustria - OPC: El estándar para comunicaciones entre dispositivos y sistemas de control de procesos. Retrieved from <http://www.emb.cl/electroindustria/articulo.mvc?xid=764&edi=43&xit=opc-el-estandar-para-comunicaciones-entre-dispositivos-y-sistemas-de-control-de-procesos>

Sinai. (2014). Sistema de control y adquisición de datos para aplicaciones industriales ASPECTOS RELEVANTES Capacidades. *Sistemas SCADA*. Retrieved from <http://www.sinai.es/Hoja de especificaciones SCADA.pdf>

Universidad de Oviedo. (2016). Redes Locales En Entornos Industriales. Buses De Campo. *Buses de Campo*, 23. Retrieved from <http://www.isa.uniovi.es/docencia/redes/Apuntes/tema9.pdf>

V\_estinf. (2013). RASPBERRY PI – Historia de la Informática. Retrieved from <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>

Villajulca, J. (2012). Estructura de un PLC: modulos de memoria. Retrieved from <http://instrumentacionycontrol.net/estructura-de-un-plc-modulos-de-memoria/>

Xavier Martí Carné. (2015). ModBUS, el protocolo Bueno, Bonito, Barato | Xavier-Martí Carné Mohedano. Retrieved from <http://www.xmcarne.com/blog-tecnico/introduccion-modbus/>

## Anexos



## Anexo A

### Requisitos del servidor OPC UA Raspberry

#### A.1. Librería Free OPC UA

Para instalar esta librería se requiere, de algunos complementos que permiten su correcto funcionamiento y uso de los métodos que contiene para la ejecución del servidor, estos pueden variar de más o menos elementos según la versión de Python. En los siguientes pasos se describe la instalación de los componentes para la versión de Python 3.4:

Antes de instalar cualquier programa se recomienda actualizar todo el software y librería usando los comandos: *sudo apt - get - y upgrade*

- 1- Dentro del editor de Python se instala la librería Freeopua mediante el facilitador de instalaciones de software “PIP”: utilizando el comando *pip install opcua*.
- 2- Se instala las librerías necesarias para se ejecute los métodos del servidor en Python. Estas son:
  - a) **Cryptography**: Librería de Python para uso de métodos primitivos. Se instala con el comando: *pip install cryptography*.
  - b) **XML**: Librería de Python para lenguajes de marcado extensible, definidos por la *World, Wide Web Consortium* utilizado para almacenar datos legibles: Se instala con el comando *pip* y premisos especiales (superusuario):  
*sudo apt-get install libxml2 libxml2-dev libxslt1-dev*  
*sudo pip install lxml*.
- 3- **GPIO**: Librería que permite controlador los puertos GPIO. Se instala con el comando: *sudo apt - get install python - rpi.gpio*.

#### A.2 Librería PyQt Python

Es una biblioteca para diseño gráfico, mediante Qt que se realiza a través de la programación de Python, permite la integración de métodos a la interfaz gráfica del servidor. Para su descarga se utiliza el comando: *sudo apt - get install python - qt4*.



## Anexo B

### Diagrama de clases de sistemas embebidos OPC para invernadero

#### B.1. Diagrama de clases de Arduino OPC DA

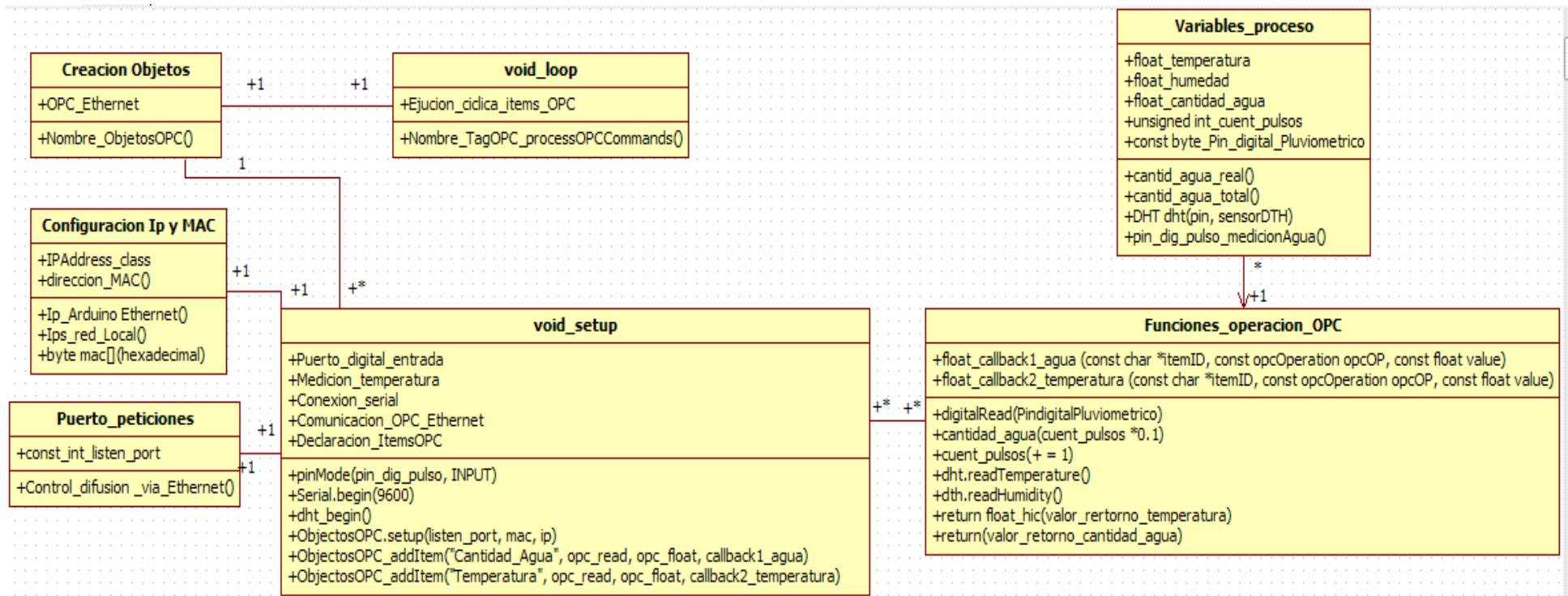


Figura B.1 Diagrama de clases Arduino Ethernet para con el servidor OPC DA Autor ( ).



### B.2. Diagrama de clases del Servidor OPC UA Raspberry Pi

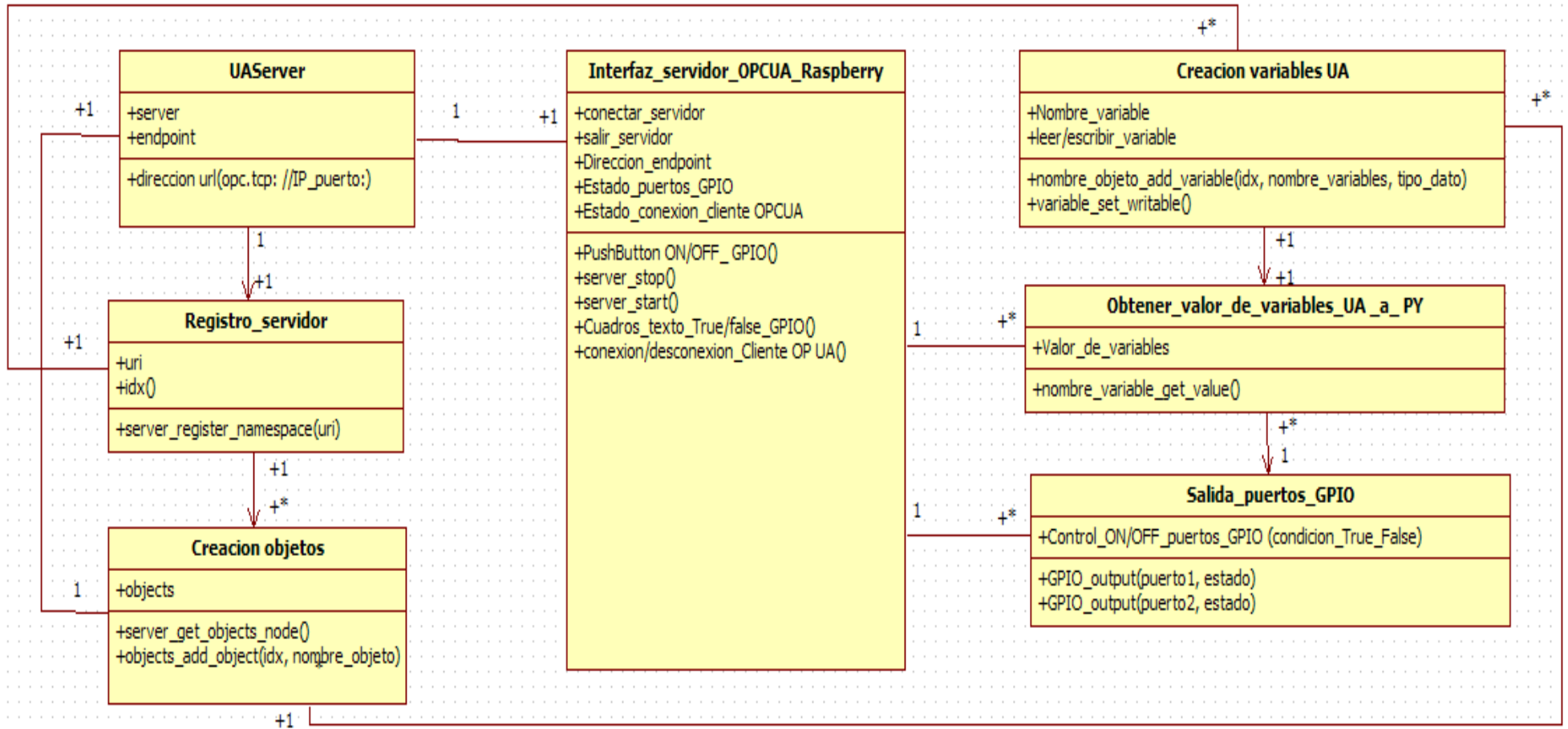


Figura B.2 Diagrama de clases de servidor OPC UA Raspberry Pi Autor ( ).



## Anexo C

### Presupuesto y documentacion sistema Ignition

#### C.1. Presupuesto del proyecto

Descripción	Cantidad	Costo Unitario [USD]	Costo Total [USD]
PLC S7 – 1200 CPU1214C	1	\$ 500,00	\$500,00
Arduino UNO R3	1	\$ 20,00	\$ 20,00
Ethernet Shields	1	\$ 25,00	\$ 25,00
Raspberry Pi 2	1	\$ 80,00	\$ 80,00
Pantalla para Raspberry	1	\$ 34,00	\$ 34,00
Sensor de temperature DHT11	1	\$ 11,00	\$ 11,00
Sensor de Humedad Funduino	1	\$ 5,00	\$ 5,00
Sensor de Lluvia YL 83	1	\$ 5,00	\$ 5,00
Modulo de 4 relés	1	\$ 10,00	\$ 10,00
Ventiladores	2	\$ 2,00	\$ 4,00
Maqueta del Invernadero	1	\$ 90,00	\$ 90,00
Dispositivos y accesorios eléctricos y electrónicos	1	\$ 30,00	\$ 30,00
Diseño, desarrollo y pruebas del sistema OPC [días]	130	\$ 2,00	\$ 260,00
Otros	1	\$ 10,00	\$ 10,00
<b>Presupuesto Total (USD)</b>			<b>\$ 1084,00</b>

Figura C.1 Costo estimado del proyecto Autor ().

#### C.2. Documentación del sistema Ignition

El sistema Ignition, contribuye con información completa y recabada de todas las funciones básicas de sus entornos y módulos en su entorno de desarrollo grafico a través del servidor Gateway. Las cuales se categorizan en las áreas:

##### Universidad Ignition

Esta plataforma está dotada de un área de aprendizaje de Ignition a través de materiales multimedia. La cual está disponible desde el link:

<https://docs.inductiveautomation.com/display/DOC/Scripting>

##### Documentación Ignition

Provee toda informa de sus módulos y funciones sobre lenguajes de programa *Jython* y *Express* y demás componentes de desarrollo, la cual puede visitarse desde el link:

<https://www.inductiveuniversity.com>



## Anexo D

### Aspectos del sistema SCADA OPC

#### D.1 Valores de parámetros característicos del proceso

Fecha	Temperatura ° C	Humedad % RH	Cantidad de agua mm <sup>3</sup>
2018-02-18 13:25:00	29.986784	25.299999237060	22.84082
2018-02-18 13:25:01	23.299805	25.329999923706	45.652058
2018-02-18 13:25:02	20.90043	30.280000686645	50
2018-02-18 13:25:03	21.347527	30.280000686645	50
2018-02-18 13:25:04	19.841526	25.329999923706	46.15304
2018-02-18 13:25:05	18.5595555	25.329999923706	38.626736
2018-02-18 13:25:06	28.986784	25.3600006103516	44.328007
2018-02-18 13:25:07	25.299805	25.3600006103516	48.315575
2018-02-18 13:25:08	24.90043	30.4099998474121	39.71861
2018-02-18 13:25:09	21.347527	24.2800006866455	22.862713
2018-02-18 13:25:10	19.841526	24.2800006866455	26.410442
2018-02-18 13:25:11	17.5595555	24.2800006866455	50
2018-02-18 13:25:12	24.986784	23.1299991607666	42.766033
2018-02-18 13:25:13	23.299805	23.1299991607666	34.188572
2018-02-18 16:02:55	20.90043	24.2299995422363	29.160683
2018-02-18 16:03:05	21.347527	23.1800003051758	39.949642
2018-02-18 16:03:15	19.841526	23.1599998474121	14.076967
2018-02-18 16:03:25	17.5595555	37.5723233415679	26.410442
2018-02-18 16:03:35	29.986784	27.3299999237061	50
2018-02-18 16:03:45	25.299805	28.8600006103516	42.766033
2018-02-18 16:03:55	22.90043	28.9400005340576	34.188572
2018-02-18 16:04:05	21.347527	28.9400005340576	29.160683
2018-02-18 16:04:15	19.841526	28.8600006103516	39.949642
2018-02-18 16:04:25	15.5595555	28.8600006103516	14.076967

Tabla D-1 Valores de los distintos sistemas de gobierno (Arduino, Raspberry y PLC) OPC Autor ().



## D.2 Sistema SCADA OPC implantado



Figura D.1 Vista del PLC conectado al SCADA OPC UA Ignition Autor ().

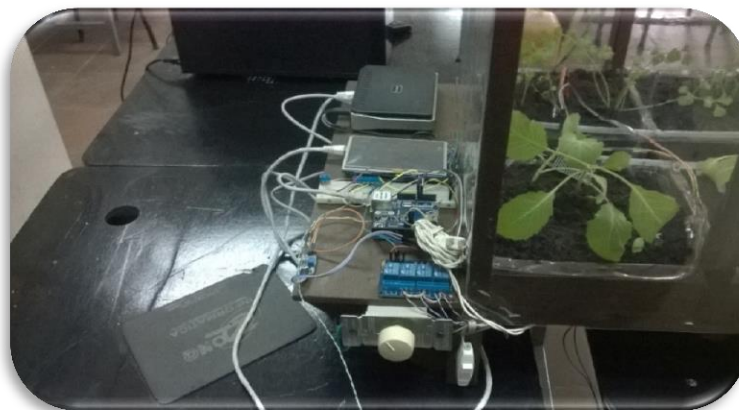


Figura D.2 Vista de los sistemas embebidos, elementos “medición y control” y enlace de red local del SCADA OPC UA Ignition Autor ().



Figura D.3 Vista general del sistema SCADA OPC UA Ignition instalado con el invernadero en el laboratorio Autor ().