

Analysis and Interface Design for Arduino-based PLC connection with Allen Bradley PLC via Ethernet

Análisis y Diseño de Interfaz para conexión de PLC basado en Arduino con PLC Allen Bradley a través de Ethernet

J. M. Valencia-Henao , G. R. Solarte-Martínez , L. E. Muñoz-Guerrero 

Abstract— In this research, the creation of an interface that allows the connection between PLC equipment from different manufacturers and the philosophy of profit of software is proposed, on the one hand we have the industrial automation giant Allen Bradley, who has quite a presence in our region, and it has a suite of proprietary software that increases its cost proportionally with its functionality, and on the other hand Industrial Shields, a company that has emerged in the search for low-cost equipment, sheltered by the apogee and boom of technology based in open source, which offers equipment with industrial support features, programmable in Open Source platforms; The development of this methodology is divided into 2 stages, divided into two stages, it is translated into an essay, it is created in an API, SOA and web interface, with all this information is collected in the operation of the two devices, and currently. The interconnection compatible protocols, the bases for a second stage, the design of a connection protocol, which can be managed by Allen Bradley PLC from an-Arduino -based PLC; ¿What is the problem? How does it work? How does it work? The complete development of the project that could find the style of architecture that allows interoperability between equipment and software from different manufacturers.

Index Terms— Arduino, Industrial Automation, Open Source, OPC Interface PLC, SCADA.

Este manuscrito fue enviado el 01 de mayo de 2018 y aceptado el 26 de junio de 2019. Este trabajo fue soportado por la Universidad Tecnológica de Pereira. Facultad de Ingeniería de Sistemas, Universidad Tecnológica de Pereira, Risaralda Colombia es un trabajo derivado del proyecto de investigación, estrategia aprendizaje a través del TIC, tipo de trabajo es, de investigación, sobre Análisis y Diseño de Interfaz para conexión de PLC basado en Arduino con PLC Allen Bradley a través de Ethernet.

J. M. Valencia-Henao, Ingeniero de sistema y computación. (sejoma989@utp.edu.co)

G. R. Solarte-Martínez, Doctor en Informática de la Universidad Pontificia de Salamanca con sede Madrid España Suficiencia investigativa, D.E. A Universidad Pontificia de Salamanca con sede Madrid España, Magister en Investigación de Operativa y Estadística de la Universidad Tecnológica de Pereira. (roberto@utp.edu.co)

L. E. Muñoz-Guerrero, Msc. En Ingeniería de Sistemas Universidad Nacional de Bogotá, profesor Asociado, de planta, del programa en ingeniería de Sistemas y Computación. (lemunozg@utp.edu.co)

Resumen— En esta investigación, se plantea la creación de una interfaz que permita la conexión entre equipos PLC de diferente fabricante y filosofía de lucro de software, por un lado tenemos al gigante de la automatización industrial Allen Bradley, que tiene bastante presencia en nuestra región, y cuenta con una suite de software privativo que incrementa su costo proporcionalmente con su funcionalidad, y por otro lado Industrial Shields, una empresa que ha surgido en la búsqueda por equipos de bajo costo, cobijada por el apogeo y el boom de la tecnología basada en código abierto, la cual ofrece equipos con características de soporte industriales, programables en plataformas Open Source; el desarrollo de esta metodología se divide en 2 etapas la primera consiste en investigar documentación relacionada sobre la creación de una API, SOA e interfaz web, con toda esta información recolectada sobre el funcionamiento de los dos dispositivos, y las tecnologías actuales en cuanto a los protocolos compatibles de interconexión, se sientan las bases para una segunda etapa de Diseñar un protocolo de conexión, que permita administrar PLC Allen Bradley desde un PLC basado en Arduino; uno de los retos más interesantes en esta investigación fue la formulación del proyecto, debido a que se tenía desconocimiento sobre el tipo de interfaz que debía usarse, y otros modelos fueron ideados, y tratados de llevar a la conceptualización, sin embargo, fue solo hasta el desarrollo completo del proyecto que se pudo encontrar el estilo de la arquitectura que permite la interoperabilidad entre, equipos y software de diferentes fabricantes.

Palabras claves— Arduino, Automatización Industrial, Interfaz OPC, Open Source, PLC, SCADA.

I. INTRODUCCIÓN

EN el mundo de la industria, es común encontrar dentro de las grandes fábricas y empresas, que muchos de sus procesos, desde los operacionales hasta los de producción, han sido tecnificados y estandarizados utilizando herramientas en el mercado como Computadores, Sistemas embebidos, lógica cableada y Controladores Lógicos Programables. Estos últimos, mundialmente conocidos como PLC¹, se han vuelto el pilar de la industrialización de procesos de producción, puesto que son computadoras sin periféricos E/S, optimizadas con

¹ CTIN. Centro de Tecnología e Innovación. (2013). ¿Qué es un PLC? Ciudad de México, <http://www.ctinmx.com/que-es-un-plc/>

mucha robustez para la transmisión y recepción de datos en ambientes hostiles y de mucho ruido en términos de señales, que brindan a quien lo usa, la seguridad y confiabilidad de los datos procesados que viajan desde la máquina a la estación de control, y viceversa.

La fabricación y programación de controladores lógicos programables, o PLCs, ha sido comandada por los grandes fabricantes internacionales, Siemens en Europa, Allen Bradley en estados unidos, Omron en Japón, Delta en China, entre otros, desarrollan junto con sus equipos, software[que se encarga de gestionar los datos y la comunicación entre estos, por consiguiente cada fabricante desarrolla software que es ajustado a las características de cada dispositivo, sin embargo, un sistema basado en PLCs, es un sistema cerrado, hermético, que poco contacto tiene con el mundo exterior, solo lo que el fabricante y las normativas lo permitan, en conclusión, el desarrollo con un PLC es muy ajustado en términos de escalabilidad y alcance a las limitantes de cada fabricante, comparado a los grandes costos que acarrea.

Hoy en día el software libre ha llegado a lugares impensables², y, es en este punto en donde un fabricante de PLCs se propuso a instalar placas arduino en estos controladores lógicos, desde el 2006, abriendo paso a un nuevo mundo de automatizaciones industriales basadas en internet de las cosas, este es un gran aliciente para los profesionales en sistemas que tenemos afinidad o interés con el mundo de la automatización, no sólo para desarrollar nuevos sistemas automatizados [1]bajo la filosofía del software libre, sino también para migrar los sistemas actualmente instalados y aumentar las funcionalidades ya obtenidas, con las ventajas de la programación en plataformas abiertas.

Por años, el sector industrial ha automatizado operaciones usando dispositivos PLC, en Latinoamérica el fabricante líder ha sido Allen bradley, con tecnología de punta, y equipos compactos de mediana gama, como el Micrologix 1100, pero costoso, alrededor de 4 millones por equipo, y software como el RS Logix 500 [2] que comienza desde los 15 millones de pesos en adelante por estación, es decir, de manera local, lo que hace que la automatización de cualquier proceso signifique una mediana o gran inversión, y pensar en un sistema distribuido con varias estaciones hace que sea una inversión monumental.

II. FORMULACIÓN DEL PROBLEMA

En nuestra región, debido al nublado concepto cultural que se ve en la inversión en tecnología, puesto que el retorno a capital en algunos casos puede tomar décadas, son contadas las empresas que cuentan con sistemas de automatización industrial estandarizados a nivel mundial, importados e

instalados por empresas como G&L Ingenieros, que llevan gran trayectoria en la instalación y acoplamiento de sistemas automatizados, pero que por consiguiente suponen grandes inversiones en capital para quien los contrate, no sólo para adquirir estos costosos equipos, también para suplir los costos de mano de obra tecnificada.

El inconveniente principal a la hora de pensar en migrar sistemas completos que ya cumplen de manera óptima el objetivo que fueron diseñados, a software libre, es el costo, puesto que los directivos ya han aprobado los presupuestos, y las fábricas poseen equipos e infraestructura de alguno(s) de los fabricantes ya mencionados anteriormente y que llamaremos “cerrados”, y por esto, plantear un cambio en el cerebro (PLC) del proyecto, significaría idear un proyecto nuevo con costos que ya fueron asumidos y ejecutados, para solucionar un problema que ya fue corregido; ningún directivo en sus 5 sentidos estaría dispuesto a re - asumir costos innecesarios.

1. Arduino [3], por qué se usó?

Costos: Un PLC basado en arduino [3] puede estar alrededor del millón de pesos (\$1'000.000), sin embargo, su programación se puede simular con la facilidad de acceder a una placa de arduino Mega, por \$40.000 y agregar una placa de conexión ethernet por \$25.000, comparar a los costos de un PLC Allen Bradley Micrologix 1100 alrededor de cuatro millones de pesos (\$4'000.000) hace que sea la opción más directa al hablar de presupuestos. No hay mucho que decir del software de programación gratuito Arduino IDE, que está disponible para ser descargado desde la red, mientras que la suma de los licenciamientos de Rockwell Ronda los treinta millones de pesos (\$30'000.000). Programación y conexión transparente: Solo es requerida una conexión a través de un cable serial de datos desde la placa Arduino Mega³, con salida a puerto USB para su alimentación de energía y programación desde un PC, se configura el software Arduino IDE con 3 clicks al seleccionar el modelo de la tarjeta, y la placa está lista para recibir órdenes. Los dispositivos de salida como diodos led o motores que funcionan a un máximo de 12 V, son muy baratos y fáciles de conseguir en los almacenes del centro de la ciudad.

¿Qué es el microframework Flask y por qué fue elegido?

Microframework⁴ es el término que ha sido usado para referirse a Frameworks minimalistas de aplicaciones web, en contraste con Frameworks Full-Stack. Un microframework carece de la mayoría de funcionalidades que es común esperar en un Full-Stack, (como lo es Django) tales como manejo de cuentas, autenticación, autorizaciones, roles, permisos,

² Rockwell Automation. (s.f). Sistemas de controlador lógico programable Micrologix 1100, <http://ab.rockwellautomation.com/es/Programmable-Controllers/MicroLogix-1100#overview>.

³ ARDUINO. (s.f). ¿Qué es arduino? <https://arduino.cl/que-es-arduino/>

⁴ Automation Networks. (s.f). RSLinx Classic and enterprise software, <http://automation-networks.es/glossary/rslnx-classic-and-enterprise-software>.

abstracción de Bases de datos a través de un mapeo de objetos relacionales, motores de plantillas web, entre otras.

Las ventajas de tener un microframework se hacen mayores cuando la tarea de programar se vuelve simple, al ser escalable, Flask también permite añadir módulos, servicios o librerías, puesto que por defecto no está atado a ningún motor de plantillas o bases de datos, inclusive deja el manejo de excepciones a gusto del programador.

III. DESARROLLO DE LA METODOLOGÍA

Es imprescindible saber cómo dar y recibir órdenes del PLC basado en arduino [4], y también conocer sobre los protocolos que se utilizan actualmente como interfaces de comunicación compatibles con el dispositivo cerrado AB, a fin de permitir que las entradas de este. que es el dispositivo que se encuentra instalado en las fábricas, sean las salidas del PLC arduino [5], y a su vez las salidas de AB. sean entradas para el PLC basado en placas de software libre.

Con la información recolectada sobre el funcionamiento de los dos dispositivos, y las tecnologías actuales en cuanto a los protocolos compatibles de interconexión, se sientan las bases para una 2a etapa, la cual encierra los conocimientos adquiridos y plantea un sistema en el cual los datos que son recolectados en campo, por los periféricos conectados al PLC AB. pueden ser procesados, guardados, organizados y mostrados a través de Arduino, con una interfaz desarrollada en software libre.

A. Etapa 1: Investigación y captación de información.

En esta etapa se realiza la disposición y recolección de la información necesaria para el planteamiento de este proyecto, que abarca desde la conceptualización de herramientas y arquitecturas de actualidad para la creación de interfaces multiplataforma, hasta el reconocimiento de los ítems necesarios para el entendimiento tanto del PLC Allen Bradley como del M-Duino. Se divide en 3 actividades principales, la cuales se describen detalladamente a continuación:

B. Actividad 1, Investigar documentación relacionada a la creación de una API, SOA e interfaz web:

Al comenzar el proyecto no había mucha claridad a la hora de la etiqueta que se le daría al tipo de arquitectura del software que se diseña para encargarse de permitir una conexión multiplataforma, en este caso entre RSLogix 500 y Arduino IDE, sin embargo con el paso del tiempo, las horas de estudio y recolección de datos en el curso, fue posible descubrir otro estándar de comunicación que ha sido desarrollado en el campo del control y supervisión de procesos industriales, llamado OPC (OLE for Process Control) [5] Según la fundación OPC (2018), OPC es un protocolo basado en una tecnología Microsoft, que ofrece una interfaz común para comunicación la cual a su vez permite que componentes de software individuales interactúen y compartan datos, permitiendo que un dispositivo de hardware del nivel de planta de producción actúe como servidor de cualquier aplicación

que se programe para leer/escribir cualquier variable que este ofrezca.

OPC está basado en OLE, software de tipo cliente pesado de Microsoft que permite crear un archivo principal, al cual se puede hacer una referencia de datos donde todo cambio posterior en el archivo principal se refleja en el documento donde se hizo referencia. A pesar de que los grandes fabricantes como Allen Bradley han incluido OPC en sus productos, y que hace posible la extracción de datos desde el dispositivo Micrologix 1100 a una hoja en excel como lo veremos más adelante, se hace necesaria la creación de un servidor programado como OPC que sea externo a la Suite de Rockwell Automation.

OPC plantea la interconexión de todos los dispositivos a través del protocolo ethernet, por lo que se plantea la creación de un servidor local en el micro framework de programación para python flask, el cual ha tomado gran fuerza en los últimos tiempos para la creación de API en python, ya ha sido trabajado y probado en conjunto con Arduino en un proyecto para la materia cliente servidor y es un excelente recurso a la hora de proyectos que tienen propuesto un servidor de características ágiles y livianas. Fig. 1, Código en pyhton usando microframework flask para crear un servidor HTTP, el código de un servidor que hace peticiones a la API publica de Twitter, para que estos datos sean consumidos por un dispositivo Arduino Fig. 1,

```
#include <SPI.h>
#include <Ethernet.h>

// mac dispositivo arduino
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// Dirección ip en caso de que falle la conexión por DHCP
IPAddress ip(192, 168, 0, 45);
// Dirección ip del servidor al cual se está conectando
IPAddress server(192, 168, 0, 152);

// Inicialización del servicio ethernet
EthernetClient client;

bool Flag = false;
char oldc = '0';
String cadena = ""
String last_id = ""; // Último id procesado

void almacenarCadena(String st)
{
  char buf[st.length()]; // Serial.println(st);
  st.toCharArray(buf, st.length());
  char * com_id = strtok(buf, ".");
  String id_actual = String(com_id);

  if ((last_id.toInt()) == (id_actual.toInt()))
  {
    Serial.println("Este twitter ya fue procesado por el arduino");
  }
  else if (last_id.equals(""))
  {
    // Acá se asigna el último ID válido recibido dentro de la var
    last_id = id_actual;
  }
}
```

Fig. 1, Código en Python usando microframework Flask para crear un servidor HTTP. Fuente

Actividad 2, Investigar y conocer el funcionamiento interno de un dispositivo PLC [6] Allen Bradley y de un PLC basado en arduino.

PLC Allen Bradley.

El micrologix 1100 [7] ha sido diseñado para monitorización remota y para aplicaciones que requieran bastante disponibilidad de memoria, pero requiere pocos puertos de entradas y salidas. Posee un socket Ethernet/IP de 10/100 Mbps para mensajería P2P (Peer to peer) y cobertura de conectividad para toda la familia de los productos PLC de la familia Allen Bradley bajo las limitantes del software producido por Rockwell Automation.

El punto de quiebre está en los costos de adquisición tanto del hardware, que oscila alrededor de US\$1,300, mil trescientos dólares, alrededor de cuatro millones de pesos (\$4'000.000), como del software que se requiere para el aprovechamiento de esta herramienta, es toda una suite de utilidades, RsLogix 500 [8], RSLinx RSEmulate, RsView, los cuales con todas las características y funcionalidades pueden tener un costo de Alrededor treinta y ocho millones quinientos mil pesos (\$38'000.000).

PLC Industrial Shields.

En este caso se toma la referencia del M-DUINO PLC Arduino Ethernet 21 I/Os Analog/Digital PLUS, el cual se puede apreciar seguido a este párrafo en Figura 5 Imagen de M-DUINO PLC Arduino Ethernet 21 I/Os Analog/Digital PLUS.

Está especialmente diseñado para su uso en un entorno profesional y cuenta con 13 entradas y 8 salidas (dispositivo de similares características al AB. que cuenta con 10 entradas y 6 salidas), de las cuales 7 entradas y 3 salidas son digitales configurables por software, cuenta con puertos de conexión para comunicación tipo Ethernet, USB tipo B, puerto I2C que ofrece la posibilidad de expandirse con 127 módulos y gobernar hasta 6604 E/S en modo maestro esclavo, además de módulos adicionales de sensores, o de expansión de la misma marca. También cuenta con dos puertos seriales, un RS-232 y un puerto HALF/FULL Duplex RS-485. Soporta protocolos TCP/IP, Modbus TCP y Modbus RTU y tiene la capacidad de comportarse como un Web server, lo que le ofrece una gran flexibilidad y control.

Actividad 3, Analizar cómo cada dispositivo PLC ejerce control sobre el sistema.

PLC Allen Bradley.

Para tomar control del sistema, después de que todas las conexiones físicas han sido instaladas, el PLC Allen Bradley debe ser configurado en la plataforma RSLinx, la cual permite la comunicación entre los dispositivos de control a través del protocolo Ethernet.

Luego de que se selecciona la comunicación entre la tarjeta del controlador del cual se van a extraer los datos el PLC y el computador o la pantalla táctil con la que se va a descargar el programa, se procede a descargar el programa lógico en lenguaje ladder que ha sido programado en la plataforma

RSLogix 500. El PLC recibe las órdenes a ejecutar en este programa a través del cambio de modo de Offline a Descarga, y una vez se vuelva a cambiar el modo a Online de inmediato comienza a ejecutar los datos de las órdenes de control que se indique en el código de lógica de contactos Ladder, sobre los actuadores del sistema, y recoger los datos que le entregan los sensores para su posterior análisis en el SCADA. En este punto es opcional el uso del software RSEmulate, debido a que se puede o no simular el proceso con el PLC, ya depende de la experticia y la planeación de quien ejecuta el proyecto.

Después de que el sistema ya está probado y funcional, el control pasa al SCADA que se haya elegido, en la suite de Rockwell se ofrece RSView, sin embargo, en este punto cualquier SCADA que tenga compatibilidad con este equipo puede tomar el mando. El SCADA se encarga de las ejecuciones de órdenes bien sea para la activación de máquinas, para el funcionamiento de una electroválvula, etc. Esto depende la naturaleza del proyecto. Paralelo a ejecutar las órdenes, el dispositivo se encuentra recibiendo datos en tiempo real de los sensores, si el proyecto cuenta con ellos, (por ejemplo, un flujómetro que se encarga de la medición de flujo de determinado líquido que pasa por un área en determinado tiempo) el SCADA también según su programación estará almacenando información en determinada Base de Datos, para luego proceder a analizar los datos y generar informes gerenciales, de producción, administrativos.

PLC Industrial Shields.

La primer gran diferencia que se encontró al trabajar con estos dispositivos, en relación a los tradicionales, es que su lenguaje de programación es Arduino IDE, cuya adquisición es gratuita y tiene una estructura bastante simple, que divide la ejecución en dos bloques principales, setup() y loop(), la primer función que se ejecuta en el programa y solo una vez, es setup() donde se definen las variables pinmode (qué pines digitales son de entrada o salida) y se inicia la comunicación serial. Loop() contiene todo el código que será ejecutado en cascada, donde se configurará el comportamiento de las entradas y las salidas ver código Fig.2

```
int Q06 = 3;
// Select the Analog (0-10Vdc) / PWM (10 OR 24Vdc) / Digital
// (24Vdc)OUTPUT
int Q05 = 5;
// Select the Analog (0-10Vdc) / PWM (10 OR 24Vdc) / Digital
// (24Vdc)OUTPUT

void setup()
{
  pinMode(Q06, OUTPUT);
  pinMode(Q05, OUTPUT);
}

void loop()
{
  analogWrite(Q06, 254);
  digitalWrite(Q05, HIGH); // Relay ON
  // turn the LED on (HIGH is the voltage level)
  delay(1000); // Wait for a second

  analogWrite(Q06, 0);
  digitalWrite(Q06, LOW); // Relay OFF
  // Turn the LED off (LOW is the voltage level)
}
```

Fig.2: Ejemplo de programación de dos salidas y su activación. Fuente: Industrial Shields By Boot & Work Corp. S.L.

La segunda gran diferencia tiene que ver con el uso de los protocolos de comunicación de manera libre para el intercambio de datos físicos entre diferentes PLC, los cuales son: el protocolo I2C, protocolo propiedad de arduino el cual funciona sólo bajo sus dispositivos y la arquitectura maestro - esclavo, protocolo Serial, el cual es el más usado en la búsqueda de estabilidad, confiabilidad y seguridad en los datos sacrificando un poco la velocidad y rapidez entre ellos y, solo entre dispositivos M-Duinos, protocolo Ethernet.

Etapa 2: Diseño y divulgación.

Actividad 1, Diseñar una interfaz que permita administrar PLC Allen Bradley desde un PLC basado en arduino de una interfaz, sin embargo, con el desarrollo de la investiga. Está actividad fue planeada teniendo en cuenta los conceptos básicos de análisis y desarrollo ción, se descubrió el estándar OPC el cual surgió en 1995 con la unión de los grandes fabricantes de hardware y software industriales, que junto con Microsoft desarrollaron la primera versión de lo que sería el estándar mundial de protocolos de comunicación e interoperabilidad entre dispositivos de automatismos industriales. Es por esta razón que se decide planear la interfaz en base a esta arquitectura, la cual tiene su primer bosquejo graficado seguido en Fig. 3 Bosquejo del sistema en base a un gráfico de arquitectura OPC.

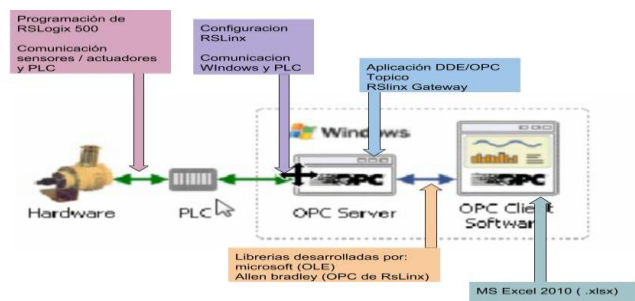


Fig.3. Bosquejo del sistema en base a un gráfico de arquitectura OPC. Fuente: Wolfgang Mahnke Stefan-Helmut Leitner [9]

La interfaz para desarrollar debe tener un módulo Acceso _Datos _Sincronos que permita leer, escribir y monitorizar variables que contienen datos de proceso actuales en el PLC Allen Bradley Micrologix 1100. Su principal uso será el transmitir datos de tiempo-real entre PLC AB. Y PLC M-Duino 21 I/Os plus, y otro dispositivo de control en un PC pantalla cliente. Se debe poder seleccionar explícitamente las variables o tópicos (elementos OPC) que quiera leer, escribir o monitorizar en el servidor desde el PLC AB, bien sea a través de la interfaz, Configurar un tópico OPC/DDE en RsLink para obtener datos de un controlador Logix 500 y agregarlo a una hoja de cálculo de excel, al final de este documento. A través de la interfaz OPC se establece una conexión al servidor M-Duino creando un objeto OPCServer, donde los datos de salida del Allen Bradley son las entradas de la interfaz, y contiene los métodos navegar_direcciones y buscar_dato OPC y sus propiedades, como el tipo de datos, o los permisos de acceso, los cuales se guardarán en un diccionario de datos.

Se debe poder escoger entre 2 modos, el primero monitorizar los cambios de los valores por el M-Duino, a fin de que si este detecta un cambio que supere límites establecidos, reaccione con cierto tipo de alerta o mensaje y gestione este problema ofreciendo el instante de muestreo (timestamp) y la calidad a los datos enviados. La calidad especifica si el dato es correcto (bueno), no disponible (malo) o desconocido (dudoso).

El segundo modo, una visualización en pantalla cíclica de los datos en busca de cambios para la supervisión en tiempo real, con una demora de actualización que no debe ser mayor a 2 segundos. Este tiempo de actualización es utilizado en el servidor para comprobar los valores cíclicamente en busca de cambios. A cada ciclo, el servidor envía sólo los valores cambiados al usuario.

Otro módulo que se propone es Acceso_Datos_Historicos, que debe permitir el acceso a datos ya almacenados. Los archivos históricos se deben recuperan de manera uniforme, desde el sistema de registros simple del Micrologix, hacia la interfaz OPC.

Es por esta razón que se decide planear la interfaz en base a esta arquitectura, la cual tiene su primer bosquejo graficado seguido en Fig. 4, Planteamiento inicial del sistema OPC

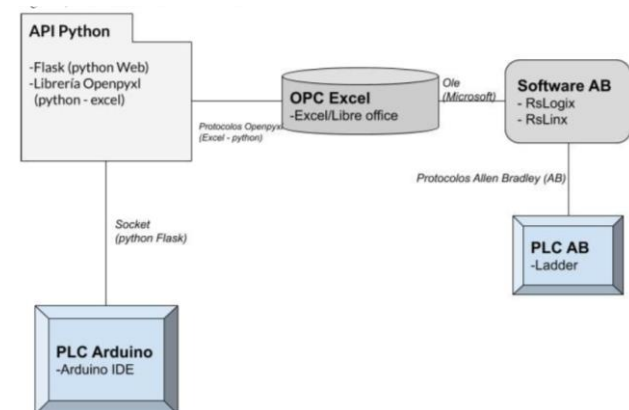


Fig. 4. Planteamiento inicial del sistema OPC, Fuente: Planteamiento inicial del sistema OPC

Tarea 2, Elaboración de casos de uso

La elaboración de los casos de uso gráficamente apreciada a través de Fig. 5. Diagrama UML de los casos de uso, se hace de acuerdo al levantamiento de requerimientos hecho en la tarea anterior y se ve a continuación.

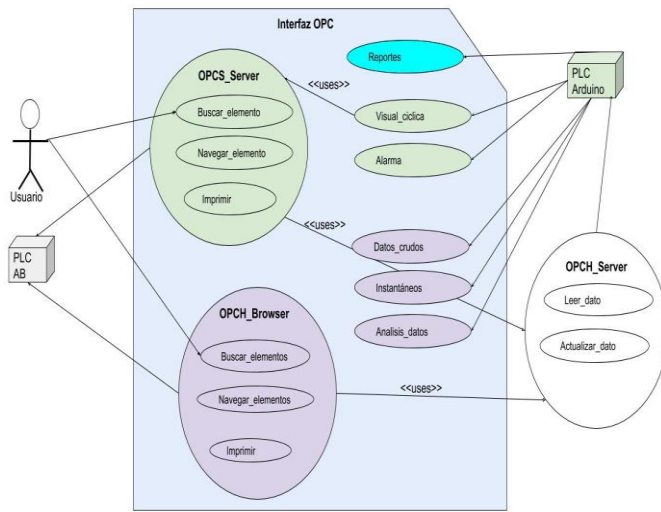


Fig. 5. Diagrama UML de los casos de uso. Fuente: Autores

Tarea 3, Desarrollo de especificación

Para la especificación del sistema se plantea Tabla I Documentación de los actores que interactúan en los casos de uso, donde se aprecian los actores que interactúan más adelante en Tabla I.

TABLA I.

DOCUMENTACIÓN DE LOS ACTORES QUE INTERACTÚAN EN LOS CASOS DE USO

Actor	Caso / uso	Tipo	Descripción
Usuario	2	Primario	Cliente que manipula el sistema
PLC Allen Bradley Micrologix 1100	2	Secundario	PLC que funciona como esclavo y se encarga de recoger los datos de campo
PLC M.Duino 21 I/O	7	Secundario	PLC que funciona como maestro y recoge los datos para administrarlos, analizarlos y mostrarlos

Fuente Autores

IV. RESULTADOS

Los resultados más notables de esta investigación pueden apreciarse en la etapa 2 del procedimiento, puesto que el objetivo principal de este proyecto fue el de analizar los elementos de un sistema de automatización industrial actual, manejado por el común Micrologix 1100 [7], visto desde la perspectiva de un Ingeniero de Sistemas, para así de esta manera diseñar una interfaz de conexión entre el sistema con este robusto equipo, y un dispositivo que está basado en una filosofía de software libre.

A pesar que las tareas de la etapa 2 describen desde el análisis de los requerimientos hasta el diseño de la interfaz, lo hacen de una manera conceptual, visto desde la Ingeniería de Software. Es por esto que para tener una vista más global en

función de la presentación del uso de todas las herramientas presentes, buscando la integración de conocimientos técnicos del área eléctrica de la instrumentación y automatización con área de sistemas y desarrollo, se muestra a continuación la Fig. 6, Esquema gráfico de sistema con PLC Arduino, el cual es un esbozo más completo de todo el ciclo de flujo de datos que se plantea, desde las conexiones físicas con los sensores y actuadores, como se pudo apreciar anteriormente en la Figura 4, pero en este caso agregando el túnel de conexión OPC que brinda RSLinx hacia una hoja de cálculo en excel.

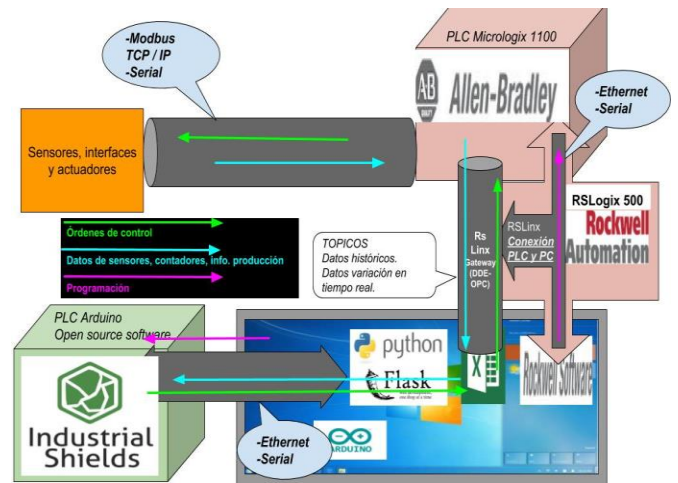


Fig. 6. Esquema gráfico de sistema con PLC Arduino [10].

Cabe resaltar que este puede no ser el esquema final de un producto terminado, esto responde a la corta experiencia, idoneidad y sabiduría de todos los elementos que componen un sistema industrial automatizado, pero es una clara muestra de que es posible utilizar herramientas comunes, como lo es Microsoft Excel, y junto con herramientas libres entrar en el campo del desarrollo de sistemas administrativos y de gestión de la información, cerca de lo que puede ser un SCADA, es posible apreciar este modelo más de cerca en Fig. 7, Esquema gráfico acercado de los elementos de desarrollo, en función de los elementos de desarrollo y programación necesarios que han sido descubiertos y tridoshola colación en el desarrollo de esta investigación.



Fig. 7. Esquema gráfico acercado de los elementos de desarrollo. Fuente: Autores

Es muy bien recibido por el sector de automatización local este tipo de iniciativas y proyectos, pues es visible la falta de herramientas y elementos que permitan una mejora en la optimización de recursos y procesos de la mediana y gran industria de la región cafetera, que se ajusten a las necesidades y los bolsillos de los empresarios, y que a su vez fomenten el desarrollo de proyectos de innovación y emprendimiento regional.

V. CONCLUSIONES

Finalizado el análisis y el diseño de la interfaz para la interconexión y transmisión de datos entre diferentes tecnologías de PLC, lo primero que se trae a colación es el planteamiento de la interfaz, puesto que, al principio de este proyecto, se tenían ideas muy nubladas sobre las tecnologías que usa la industria de la automatización. Uno de los retos más interesantes fue la formulación del proyecto, debido a que se tenía desconocimiento sobre el tipo de interfaz que debía usarse, y otros modelos fueron ideados, y tratados de llevar a la conceptualización, sin embargo, fue solo hasta el desarrollo completo del proyecto que se pudo encontrar el estilo de arquitectura que se ha usado en esta misma industria hace unos años, OPC, que es ahora un estándar mundial para todos los grandes fabricantes de equipos de automatización y que tiene bases muy sólidas desde 1995, es lo que ha permitido la interoperabilidad entre equipos y software de diferentes fabricantes.

El segundo factor a discutir tiene que ver con los límites y el alcance que tuvo el proyecto en su formulación, si bien este proyecto nace de un problema real que se presentó en una empresa, y su componente principal PLC Micrologix 1100 fue el mismo que se pensó para este proyecto, se dificultó demasiado el trabajo en base a este equipo, pues su acceso es costoso, y para el tiempo que el proyecto estaba tomando forma, se habían roto las relaciones comerciales y ya no se posee acceso al equipo ni al software que se necesita para su programación y funcionamiento, aunque este inconveniente tuvo solución por medio de una versión demo por 7 Días, se pudo constatar lo cerrada que es la arquitectura de la empresa Rockwell Automation [8], y los altos costos que puede generar el acceso completo de las herramientas que permiten utilizar un recurso como este, así como la complicada labor de cotizar los costos. El distribuidor oficial, Melexa, ubicado en el sector industrial la popa, nos brindó la información de costes que se puede apreciar en el desarrollo de la metodología, sin embargo, fue tras una espera de alrededor mes y medio que fue posible tener una reunión con el encargado para el Eje cafetero, debido a la complejidad que exige nuestro proyecto en torno a la programación y el conocimiento del equipo. Esto también da una visión de la espera que puede generar plantear un proyecto con una tecnología como la de Allen Bradley.

Otro hecho que ha generado un contraste ha sido la elección del protocolo a utilizar para la comunicación de los equipos, si bien Ethernet ha sido un protocolo que ha tomado mucha fuerza en los últimos años para el desarrollo de redes industriales de velocidad alta para la transmisión de datos, y esto ha tocado la puerta del mundo de la automatización, con nuevos equipos que incorporan puertos Ethernet, se desconocía totalmente el estado del arte de este sector de la industrial, y a través de esta investigación se ha podido constatar que a pesar del avance de nuevos protocolos, los protocolos Serial son aún hoy en día los más usados para la transmisión estable de datos entre equipos o entre PLCs y sensores y actuadores. RS232 y RS485 son aún protocolos vigentes en redes de automatización industrial, pues garantizan la fiabilidad y estabilidad de los datos sin importar los ruidosos o toscos ambientes donde se encuentren instalados, y son los más usados cuando lo que se busca es poca caída de paquetes en la transmisión de datos críticos, como puede serlo cualquier proceso que involucre recursos de una empresa. Es por esto que se encontró mucha documentación sobre la comunicación entre Arduino y el software de Rockwell, sin embargo, las librerías que se ofrecían solamente establecen comunicación a través de protocolos seriales, y este es el motivo por el cual tuvo que ser añadida una herramienta como Excel, ya que, por la colaboración de Microsoft con la industria de la automatización, el protocolo OLE permite la interacción entre productos Microsoft y Allen Bradley. Se pudo constatar lo poderosa que es esta herramienta de ofimática, sin embargo, hace que el planteamiento de este proyecto resulte en muchas rutas adicionales para los datos, lo que puede retrasar bastante el tiempo de respuesta y reacción del sistema, en comparación con una arquitectura con conexión por protocolos seriales.

RECOMENDACIONES

Para proponer nuevos proyectos, se debe seleccionar solo equipos de programación libre, sin la necesidad de agregar dispositivos que se dediquen solo al control de las entradas y salidas y otro al control del sistema completo, pues los dispositivos industriales de arduino cumplen con toda la normativa requerida en términos de seguridad y rendimiento Europeo, y que una adecuación como la vista en todo el desarrollo del procedimiento sería óptima en el momento que solo se quiera adecuar un sistema que ya está funcional, explícitamente como el objetivo general de este proyecto lo índice.

REFERENCIAS

- [1] F. Marsellach, «FMJ Ingenieros (2012,»). *Introducción a la automatización*. Recuperado de <https://www.fmjingenieros.com/servicios/automatizacion-industrial/introduccion-a-la-automatizacion> .
- [2] B. Allen, « Configure RSLinx DDE OPC Topic to Get Data From a Logix Controller to MS Excel,» (2016).

- [3] A. I. S. Prieto, «Controlador PLC Industrial basado en Arduino,» Industrial Shield, (2016).
- [4] I. Shields, «Controlador PLC Industrial basado en Arduino,» . 30 agosto 2018,.
- [5] J. V. Valencia H., «Diseño y aplicación de un sistema bajo arquitectura cliente servidor para la activación de riego automatizado en Arduino a través de Twitter,» vol. 1, n° 1, pp. 5 -6, 2018.
- [6] I. S. (s.f), *Controlador PLC Industrial basado en Arduino*, vol. 1, n° 1, p. 1, 2018.
- [7] R. A. (. S. d. c. I. programable, «Sistemas de controlador logico programable Micrologix 1100,» Rockwell Automation. (s.f). Sistemas de controlador logico programable Micrologix 1100,.
- [8] R. Automation, «Rockwell Automation. (s.f). RS Logix 500 Solutions from Rockwell Software,» 2018.
- [9] S.-H. L. Wolfgang Mahnke, *Arquitectura OPC unificada*, ABB Corporate Research, 2018.
- [10] A. Prieto, «Industrial Shields. (2016) Controlador PLC Industrial basado en Arduino, Presentación modelo propio industria 4.0,» Industrial Shield, ., 2018.



José Mario Valencia Henao, Ingeniero de Sistemas y Computación de la Universidad Tecnológica de Pereira, Semillero de Investigación GNTO Grupo de Nuevas técnicas de búsqueda y de optimización de la Universidad Tecnológica de Pereira.
ORCID: <https://orcid.org/0000-0003-0264-7265>



Guillermo Roberto Solarte Martínez, Doctor en Informática de la Universidad Pontificia de Salamanca con sede Madrid España Suficiencia investigativa, D.E.A Universidad Pontificia de Salamanca con sede Madrid España, Magister en Investigación de Operativa y Estadística de la Universidad Tecnológica de Pereira. Ingeniero de Sistemas de la Universidad Cooperativa de Colombia. Profesor ISC-UTP, Transitorio Titular en la Facultad de Ingenierías. Ha publicado artículos en revistas nacionales especializadas. Libros publicados: Guía didáctica de estructuras de datos, Pertenece a los grupos de investigación Grupo Grande Grupo de Avanzada en Desarrollo de Software, Grupo GIA - Grupo de Investigación En Inteligencia Artificial, Semillero de Investigación GNTO Grupo de Nuevas técnicas de búsqueda y de optimización.
ORCID: <https://orcid.org/0000-0001-5147-7798>



Luis Eduardo Muñoz Guerrero, (Pasto, Nariño, Colombia 1975). Msc. En Ingeniería de Sistemas Universidad Nacional de Bogotá, profesor Asociado, de planta, del programa en ingeniería de Sistemas y Computación. Ha publicado artículos en revistas especializadas nacionales e internacionales. Pertenece al grupo de investigación GIA: grupo de inteligencia artificial.
ORCID: <https://orcid.org/0000-0002-9414-6187>