

## Desarrollo de una HMI tipo arquitectura abierta para la comunicación Ethernet con PLCs de la marca Allen Bradley

### Development of an open architecture type HMI for Ethernet communication with PLCs from the trademark Allen Bradley

CARMONA-MARTÍNEZ, Luis Alberto<sup>†1,2,3</sup>, SOTO-CAJIGA, Jorge Alberto<sup>1</sup>, RODRÍGUEZ-OLIVARES, Noé Amir<sup>1,2,3</sup> y ORTÍZ-ORTÍZ, Tania Judith<sup>2,3</sup>

<sup>1</sup>Centro de Ingeniería y Desarrollo Industrial (CIDESI), Av. Playa Pie de la Cuesta No. 702, c.p. 76125, Col. Desarrollo San Pablo, Querétaro, Qro. México

<sup>2</sup>Universidad Tecnológica de Querétaro (UTEQ), Av. Playa Pie de la Cuesta No. 2501, c.p. 76128, Col. Unidad Nacional, Querétaro, Qro. México

<sup>2</sup>Universidad Tecnológica de Querétaro (UTEQ)-Cuerpo Académico de Automatización y Control

ID 1<sup>er</sup> Autor: Luis Alberto, Carmona-Martínez / ORC ID: 0000-0002-7925-1229, Researcher ID Thomson: H-1068-2018, CVU CONACYT ID: 486128

ID 1<sup>er</sup> Coautor: Jorge Alberto, Soto-Cajiga / ORC ID: 0000-0002-1737-5197, Researcher ID Thomson: I-5468-2018, CVU CONACYT ID: 167955

ID 2<sup>do</sup> Coautor: Noé Amir, Rodríguez-Olivares / ORC ID: 0000-0001-5892-0625, Researcher ID Thomson: I-5012-2018, CVU CONACYT ID: 444191

ID 3<sup>er</sup> Coautor: Tania Judith, Ortiz-Ortiz / ORC ID: 0000-0001-6712-9237, Researcher ID Thomson: H-6539-2018, CVU CONACYT ID: 426398

Recibido Enero 30, 2018; Aceptado Marzo 30, 2018

#### Resumen

Este artículo presenta el desarrollo de una HMI de tipo arquitectura abierta basada en una tarjeta de desarrollo Raspberry Pi 3, una pantalla táctil y una aplicación desarrollada bajo la plataforma de programación Python 2. El objetivo de la interfaz es contar con una alternativa libre y de código abierto capaz de extraer, modificar y visualizar la información de las variables creadas en los PLCs de la marca Allen Bradley de las familias ControlLogix, CompactLogix, MicroLogix y SLC 500. La comunicación fue realizada utilizando sockets TCP/IP en conjunto con una serie de librerías de comunicación que facilitan la interacción entre el PLC y la HMI por medio del protocolo CIP. Las librerías fueron desarrolladas con las especificaciones del protocolo CIP que aplican en las capas de transporte, aplicación y de usuario del modelo de referencia OSI. Posteriormente se realizó la validación del proyecto mediante la conexión de la interfaz con los PLCs Micrologix 1100 y ControlLogix 1756-L71 por medio de la lectura y escritura de 16 variables distintas para cada PLC. Este trabajo permitirá a futuro desarrollar HMIs que permitan comunicarse con distintos equipos industriales y otros dispositivos bajo el concepto de industria 4.0 en el entorno de arquitectura abierta.

PLC, HMI, Raspberry

#### Abstract

This paper presents the development of an open architecture type HMI based on a Raspberry Pi 3 development board, a touchscreen display, and a Python 2 based application. Our goal was to develop an HMI with open source, which is capable of to extract, modify, and visualize the data of the variables created in the Allen Bradley PLCs of the ControlLogix, CompactLogix, Micrologix, and SLC 500 families. First, we did the communication with TCP/IP sockets in conjunction with our communication libraries that facilitate the interaction between the PLC and the HMI. The interface was through the CIP. Later, we developed the libraries with the CIP protocol specifications, which are executed in the transport, application and user layers of the OSI reference model. Finally, the validation was done by connecting the interface with the PLCs Micrologix 1100 and ControlLogix 1756-L71 by reading and writing 16 different variables for each PLC. This work will allow a future the development of HMIs that will enable communication with various industrial equipment and other devices under the 4.0 industry concept in the open architecture environment.

PLC, HMI, Raspberry

**Citación:** CARMONA-MARTÍNEZ, Luis Alberto, SOTO-CAJIGA, Jorge Alberto, RODRÍGUEZ-OLIVARES, Noé Amir y ORTÍZ-ORTÍZ, Tania Judith. Desarrollo de una HMI tipo arquitectura abierta para la comunicación Ethernet con PLCs de la marca Allen Bradley. Revista del Desarrollo Tecnológico. 2018. 2-6: 24-31.

<sup>†</sup> Investigador contribuyendo como primer autor.

\*Correspondencia al Autor Correo Electrónico: (alberto0488@hotmail.com)

## 1. Introducción

La competitividad altamente creciente en la industria de hoy exige productos con calidad y precios competitivos. Para superar este desafío, muchas industrias han integrado sistemas automatizados, reemplazando la toma de decisiones del ser humano en un proceso mecanizado, por computadoras y dispositivos de automatización, dejando el control de las líneas de producción a una cantidad mínima de operadores. Uno de los dispositivos de automatización más utilizados en la industria actualmente son los PLCs. Los PLCs son computadoras diseñadas para el control de maquinaria de líneas de producción, que interactúan con la maquinaria por medio de múltiples señales de entrada y de salida.

La HMI es un dispositivo o sistema que permite la interacción entre el usuario y un sistema de control, como es el caso de los PLCs. Para que el operador pueda interactuar con la línea de producción por medio del PLC (asignando parámetros y revisando el estado de la línea), requiere de una interfaz hombre-máquina, o más comúnmente conocido por sus siglas en inglés HMI. La comunicación entre las HMIs y los PLCs se puede realizar con una amplia variedad de protocolos de comunicación y, dependerá de la marca y el precio de cada HMI; el protocolo utilizado y los modelos de PLCs disponibles para comunicar.

Este proyecto surge de la necesidad de contar con una HMI de bajo costo, de arquitectura abierta y de buenas prestaciones, como son: conexión a internet, generación de reportes, envío de correos electrónicos, conectividad FTP, comunicación serial, buena resolución de pantalla y que sea capaz de comunicarse vía Ethernet con PLCs SLC 500, Micrologix, CompactLogix y ControlLogix de la marca Allen Bradley. La importancia de este proyecto se centra en dos partes:

La primera parte es la arquitectura abierta y de bajo precio, la cual permite contar con una HMI con capacidad de ser modificada a nivel de hardware y software para adaptarla a necesidades específicas, principalmente en proyectos de bajos recursos o en pequeñas y medianas empresas.

La segunda parte se centra en su posición en la industria, ya que se prevé que el mercado mundial de HMIs llegue a 6.31 mil millones de dólares para antes del año 2022 (Grand View Research, 2016), con un crecimiento en el mercado de software de interfaz estimado superior al 10% promedio anual en los próximos años. También se considera el crecimiento del mercado de la automatización de manera general, el cual espera un crecimiento de 149 mil millones de dólares para el año 2022, con un 6% de crecimiento promedio anual (Market Research Future, 2018). Garantizando así la utilidad actual y a futuro de este trabajo dentro de la industria.

Para el desarrollo de la HMI se realizó un estudio de varios trabajos relacionadas con el desarrollo de interfaces para equipo industrial. La mayoría de los trabajos encontrados relacionados consisten en la implementación de software de paga para el desarrollo de las interfaces, un ejemplo de ello es el desarrollo de una sistema SCADA para monitoreo remoto de grúas RTG (Rodríguez, et al., 2017), el cual consiste en la aplicación del software Ignition para desarrollar una interfaz capaz de comunicarse con un PLC FUJI-NP1S-22 para el monitoreo remoto de los niveles de combustible de una grúa.

Pocos trabajos consistieron en el desarrollo de HMIs de bajo costo y código abierto. Un ejemplo de ello es el desarrollo de un servidor de comunicaciones para el monitoreo, supervisión y control de equipos industriales (Prado, 2005), el cual consiste en la implementación de un servidor con interfaz desarrollada en Linux y con la utilización del protocolo de comunicación EtherNet/IP para el intercambio de datos con un PLC ControlLogix 5555 de Allen Bradley. Otro ejemplo, y muy similar al trabajo desarrollado en este artículo, es la implementación de una HMI para un PLC Micrologix 1100 con Raspberry (Zavala, et al., 2016), el cual consiste en el desarrollo de una HMI de bajo costo capaz de comunicarse con PLCs SLC y Micrologix de la marca Allen Bradley.

El valor agregado de este trabajo respecto a los trabajos encontrados consiste en la diversidad de modelos de PLCs de Allen Bradley que pueden conectarse.

Ya que, en los artículos encontrados el campo de aplicación solo se centra en un cierto modelo de PLCs o se apoyan de programas y librerías de código ya existentes protegidas por derechos de autor, impidiendo o dificultando la modificación del código de comunicación para aplicaciones específicas y futuras mejoras, además, de que limitan la capacidad de la HMI para toda la versatilidad que se pueda obtener.

La estructura de la HMI desarrollada en este trabajo se basa en una computadora de placa reducida Raspberry Pi 3, que incluye un chip de tipo SoC con arquitectura ARM fabricado por Broadcom. El procesador incluye, memoria RAM, una GPU, puertos USB, HDMI, Ethernet, WiFi, 40 pines GPIO y un puerto DSI para la cámara (Raspberry Pi, s.f.). El sistema operativo utilizado es una versión adaptada de Debian, denominada Raspbian. La lógica y el entorno gráfico se desarrollaron utilizando el lenguaje de programación Python 2.

La comunicación con los PLCs se realiza por medio de Ethernet industrial. Este protocolo de comunicación se eligió debido al rápido crecimiento que tiene en el mercado. Con una tasa de crecimiento de un 20%, representando en el 2016 el 38% del mercado global (Hardware Meets Software, 2018). Es necesario mencionar que el Ethernet industrial es la aplicación del estándar Ethernet en un entorno industrial con diversos protocolos que proporcionan determinismo y control.

Dentro de la diversidad del protocolo del Ethernet industrial, se eligió el protocolo EtherNet/IP, el cual ocupa el primer lugar de los protocolos del Ethernet industrial en el mercado (Hardware Meets Software, 2018). El protocolo EtherNet/IP también es conocido como “ControlNet sobre Ethernet” y consiste en la aplicación del protocolo CIP en la capa de transporte, sesión, presentación y aplicación del modelo de referencia OSI. Utilizando como capa física y capa de enlace de datos el estándar Ethernet y respectivamente IP, UDP/TCP; como la capa de red y de transporte.

La visualización se hace por medio de una pantalla táctil de 7 pulgadas conectada al puerto DSI de la Raspberry Pi 3. Pero también se puede conectar vía HDMI hacia cualquier tamaño de pantalla, con resoluciones desde 640x480 pixeles hasta 1920x1080 pixeles.

El artículo está estructurado de la siguiente manera: En la sección 2 se presenta el desarrollo de este trabajo incluyendo la arquitectura de la HMI, el protocolo de comunicación y la implementación en código. En la sección 3 se muestran los resultados obtenidos de las pruebas realizadas. Finalmente en la sección 4 se presentan las conclusiones del trabajo.

## 2. Desarrollo

A continuación se presenta el desarrollo de del trabajo realizado, primero se da a conocer la arquitectura de la HMI, luego la arquitectura del protocolo EtherNet/IP para la comunicación mediante la operación de lectura y escritura de las variables del PLC, y finalmente la aplicación en código para la comunicación con los PLCs y la estructura de la interfaz gráfica.

### 2.1 Arquitectura de la HMI

En la Figura 1 se muestra el diagrama esquemático de la HMI compuesto únicamente por la Raspberry Pi 3 y la pantalla táctil de 7 pulgadas. La Raspberry permite comunicarse a través del puerto Ethernet o de la conexión WiFi con los PLCs Micrologix, SLC, CompactLogix y ControlLogix para una posterior visualización y control de las variables a través de la pantalla táctil, o en otro tipo de pantalla (monitor o TV) que cuente con salida HDMI.

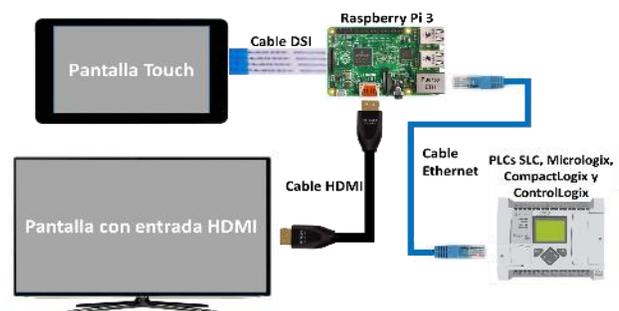


Figura 1 Diagrama esquemático de la HMI

Fuente: Elaboración Propia.

### 2.2. Protocolo EtherNet/IP

ODVA y ControlNet introdujeron el empaquetado de mensajes con TCP/IP. El resultado se conoció como “ControlNet sobre Ethernet” y ahora es llamado EtherNet/IP. El EtherNet/IP usa el Protocolo CIP, este protocolo define el encapsulamiento en la capas de transporte, sesión, presentación y aplicación del modelo OSI (Figura 2). También compartidas por ControlNet y DeviceNet.

Aplicación	Capa de aplicación CIP. Biblioteca de objetos de aplicación.
Presentación	Servicios de gestión de datos CIP. Mensajes explícitos, mensajes de E/S.
Sesión	CIP enrutamiento de mensajes, gestión de conexión.
Transporte	Encapsulación CIP. TCP/UDP.
Red	IP.
Enlace de datos	Ethernet.
Física	Punto a punto, multicast, unicast.

**Figura 2** Modelo de red EtherNet/IP

Fuente: *Elaboración Propia*

La comunicación a través del encapsulado CIP en EtherNet/IP tiene definidos dos métodos de conexión: mensajes explícitos, usando TCP/IP y mensajes implícitos, usando UDP. El método utilizado en este trabajo es el de mensajes explícitos, que sigue el patrón de conexión cliente-servidor o petición-respuesta. Todo mensaje explícito es enviado a través del puerto TCP 44818 y es estructurado de manera general (Tabla 1) por un encabezado de 24 bytes seguido de una porción de datos que varían dependiendo del comando utilizado.

Estructura	Nombre del campo	Tipo de dato
<b>Encabezado</b>	Comando	UINT
	Tamaño	UINT
	Número de sesión	UDINT
	Estatus	UDINT
	Contexto del remitente	Arreglo DE 8 BYTES
	Opciones	UDINT
<b>Datos específicos del comando</b>	Datos encapsulados	Arreglo de 0 a 65511 USINT

**Tabla 1** Paquete de mensaje encapsulado.

Fuente: *(Prado, 2005)*.

Para realizar el intercambio de información de manera coherente entre el PLC y la HMI se requiere primero de una solicitud de sesión a través del comando **RegisterSession**, el cual se estructura como se muestra en la Tabla 2.

Estructura	Nombre del campo	Tipo de dato
<b>Encabezado</b>	Comando	UINT
	Tamaño	UINT
	Número de sesión	UDINT
	Estatus	UDINT
	Contexto del remitente	Arreglo DE 8 BYTES
	Opciones	UDINT
<b>Datos específicos del comando</b>	Versión del protocolo	UINT
	Banderas de opciones	UINT

**Tabla 2** Solicitud de comando RegisterSession

Fuente: *(Prado, 2005)*

Al enviar los datos de la Tabla 2, el PLC deberá indicar a la HMI que está registrado, a través del campo “número de sesión”, en un formato de respuesta como se muestra en la Tabla 3.

Estructura	Nombre del campo	Tipo de dato
<b>Encabezado</b>	Comando	UINT
	Tamaño	UINT
	Número de sesión	UDINT
	Estatus	UDINT
	Contexto del remitente	Arreglo DE 8 BYTES
	Opciones	UDINT
<b>Datos específicos del comando</b>	Versión del protocolo	UINT
	Banderas de opciones	UINT

**Tabla 3** Respuesta del comando RegisterSession

Fuente: *(Prado, 2005)*.

El número de sesión es guardado en la HMI e insertado en las siguientes solicitudes para que puedan ser aceptadas por el PLC.

### 2.3. Operación de lectura y escritura

Las operaciones de lectura y escritura entre la HMI y el PLC se realizan a través del comando **SendRRData** que está estructurado como se muestra en la Tabla 4, y deben ser enviados a través de un mensaje sin conexión UCMR encapsulado en el campo “paquete de solicitud”, el cual está compuesto por un arreglo de datos de longitud variable.

Estructura	Nombre del campo	Tipo de dato
<b>Encabezado</b>	Comando	UINT
	Tamaño	UINT
	Número de sesión	UDINT
	Estatus	UDINT
	Contexto del remitente	Arreglo DE 8 BYTES
	Opciones	UDINT
<b>Datos específicos del comando</b>	Tipo de interfaz	UDINT
	Timeout	UINT
	Conteo de elementos	UINT
	ID del tipo de dirección	UINT
	Longitud de la dirección	UINT
	ID del tipo de dato	UINT
	Longitud de los datos	UINT
	Paquete de solicitud	Arreglo de UINT

**Tabla 4** Solicitud de comando SendRRData.

Fuente: *(Prado, 2005)*.

Al ser enviada la solicitud, el PLC responderá con base en el protocolo encapsulado en el comando **SendRRData** (Tabla 5), el cual será utilizado para extraer la información de la variable en caso de una solicitud de lectura.

Estructura	Nombre del campo	Tipo de dato
Encabezado	Comando	UINT
	Tamaño	UINT
	Número de sesión	UDINT
	Estatus	UDINT
	Contexto del remitente	Arreglo DE 8 BYTES
	Opciones	UDINT
Datos específicos del comando	Tipo de interfaz	UDINT
	Timeout	UINT
	Conteo de elementos	UINT
	ID del tipo de dirección	UINT
	Longitud de la dirección	UINT
	ID del tipo de dato	UINT
	Longitud de los datos	UINT
	Paquete de solicitud	Arreglo de UINT

**Tabla 5** Respuesta del comando SendRRData.  
Fuente: (Prado, 2005).

### 2.4. Aplicación en código

El programa fue desarrollado en Python 2 y está compuesto de dos partes: una parte de comunicación y una parte de interfaz gráfica. La parte de comunicación utiliza una librería desarrollada exclusivamente en este trabajo para comunicarse con los PLCs de Allen Bradley. La librería está compuesta por 4 funciones: 2 para leer y 2 para escribir sobre las variables del PLC. El código de la librería se realizó siguiendo la estructura del protocolo EtherNet/IP explicada anteriormente y la sintaxis de las funciones desarrolladas en este trabajo son las que se explicarán a continuación.

```
#Lectura PLCs ControlLogix y CompactLogix
Dato=C_ReadPLC('192.168.0.10', 0, 'variable00')

#escritura PLCs ControlLogix y CompactLogix
C_WritePLC('192.168.0.10', 0, 'variable01', dato)

#Lectura PLCs Micrologix y SLC
Dato=M_ReadPLC('192.168.0.10', 'B3:0')

#escritura PLCs Micrologix y SLC
M_WritePLC('192.168.0.10', 'B3:1', dato)
```

**Figura 3** Ejemplo de funciones de lectura y escritura.  
Fuente: Elaboración Propia

La función C\_ReadPLC (Figura 3) permite leer los datos de una variable tipo DINT existente en los PLCs ControlLogix y CompactLogix. Los argumentos solicitados por la función son: la dirección IP del PLC, el número de slot en el que se ubica el PLC y el nombre de la variable existente en el PLC.

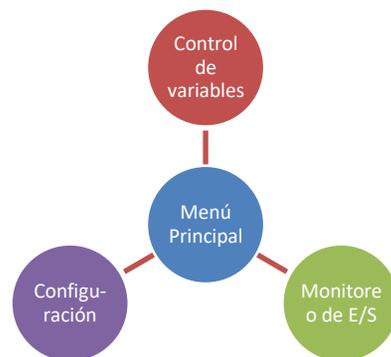
La función C\_WritePLC (Figura 3) permite escribir un valor sobre una variable tipo DINT existente en los PLCs ControlLogix y CompactLogix. Los argumentos solicitados son: la dirección IP del PLC, el número de slot en el que se ubica el PLC, el nombre de la variable a modificar y el valor a escribir.

La función M\_ReadPLC (Figura 3) permite leer los datos de una variable tipo INT existente en los PLCs SLC y MicroLogix. Los argumentos solicitados por la función son: la dirección IP del PLC y el nombre de la variable existente en el PLC.

La función M\_WritePLC (Figura 3) permite escribir un valor sobre una variable tipo INT existente en los PLCs SLC y MicroLogix. Los argumentos solicitados son: la dirección IP del PLC, el nombre de la variable a modificar y el valor a escribir.

### 2.5. Interfaz gráfica

La parte gráfica fue desarrollada mediante la librería Tkinter para la creación de ventanas, controles e indicadores que están vinculados con las variables del PLC a través de las funciones de lectura y escritura. La interfaz gráfica permite moverse entre 4 paneles como se muestra en la Figura 4.



**Figura 4** Diagrama de interconexión de los paneles.  
Fuente: Elaboración Propia

Al iniciar la interfaz el primer panel que se visualiza es el menú principal (Figura 5) que da acceso a cada uno de los otros 3 paneles para poder controlar las variables del PLC, monitorear el estado de las entradas y salidas físicas, y configurar los parámetros de conexión con el PLC.



Figura 5 Menú Principal de la HMI  
Fuente: Elaboración propia

3. Resultados

La implementación de la interfaz para las pruebas de lectura y escritura se realizaron con los PLCs Micrologix 1100 1763 serie B, CompactLogix 1769-L18ERM-BB1B y ControlLogix 1756-L71. Para cada prueba se generaron 16 variables en cada PLC (Figura 6, Figura 7 y Figura 8). Cada prueba consistió en leer y escribir datos desde la HMI sobre cada una de las 16 variables del PLC, utilizando las librerías desarrolladas para este trabajo. Todas las pruebas también se hicieron con el software comercial Interact Xpress Manager para tener un punto de comparación.

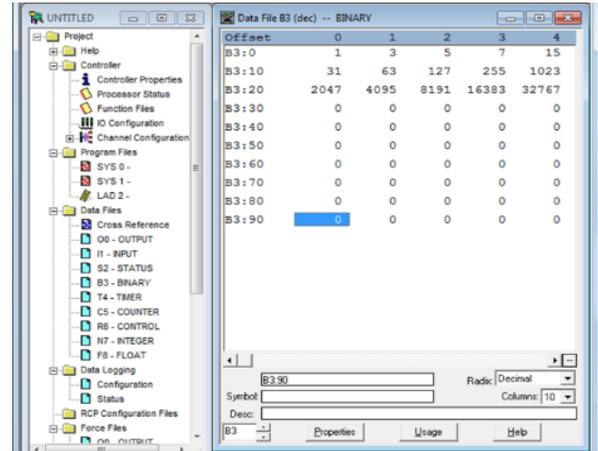


Figura 8 Asignación de variables para MicroLogix.  
Fuente: Elaboración Propia

El resultado obtenido para cada prueba fue la comunicación satisfactoria de la HMI desarrollada en este trabajo, pudiendo leer y escribir números de tipo DINT sobre cada una de las 16 variables creadas en los PLCs CompactLogix y ControlLogix, y pudiendo leer y escribir números de tipo INT sobre cada una de las 16 variables creadas en el PLC Micrologix. En la Figura 9 se muestran un ejemplo de los resultados obtenidos por la HMI de este trabajo, comparados con los resultados del software comercial (Figura 10).

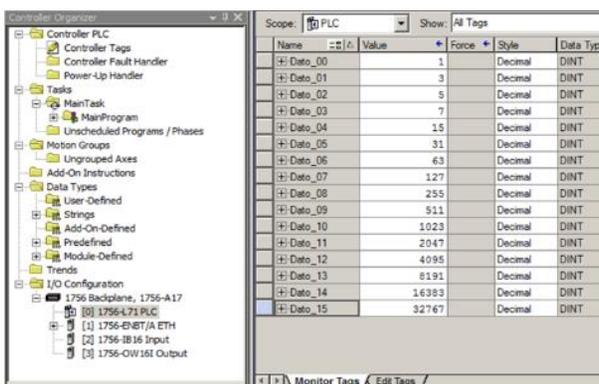


Figura 6 Asignación de variables para ControlLogix.  
Fuente: Elaboración Propia

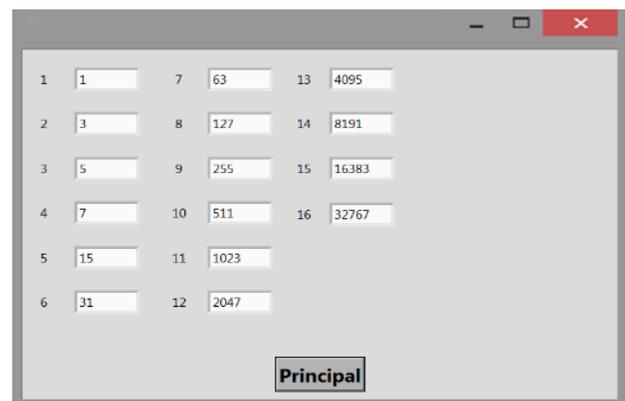


Figura 9 Variables en la HMI de este trabajo.  
Fuente: Elaboración Propia

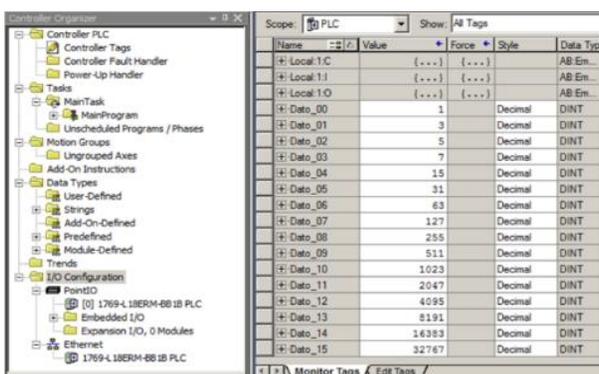


Figura 7 Asignación de variables para CompactLogix.  
Fuente: Elaboración Propia.



Figura 10 Variables en Interact Xpress Manager.  
Fuente: Elaboración Propia.

#### 4. Conclusiones

Los resultados obtenidos avalan el alcance de este trabajo, permitiéndonos contar con una HMI de un costo aproximado de 3 mil pesos, equivalente a una quinta parte del precio de una HMI comercial, con la facilidad de comunicarse con varios modelos de PLCs de Allen Bradley, siendo un resultado satisfactorio y de calidad de un equipo clasificado como minicomputadora con un precio accesible como lo es la Raspberry Pi, presentando ésta un tercio de sus aplicaciones de tipo industrial en una cultura de emprendedurismo comercial denominada: professionals makers, quienes buscan la estandarización e interoperabilidad de equipos que se comuniquen entre sí para el intercambio de información relevante para la toma inteligente de decisiones, tanto para aplicaciones de IoT (Internet of Things) como para su aplicación en el mundo industrial IIoT (Industrial Internet of Things), por lo que además de haber empleado un software libre como Python para el protocolo Ethernet/IP para un monitoreo e intercambio de variables críticas del proceso en planta productiva vía alámbrica e inalámbrica, es posible desarrollar las pantallas para monitoreo, control, históricos, alarmas propias de un HMI comercial.

La continuación del presente trabajo se extiende con un mayor alcance tanto en protocolos industriales como aplicaciones alineadas a la industria 4.0 (cuarta revolución industrial), dándole así una mayor competitividad con los equipos industriales, como también una oportunidad viable económica para las Pymes, las cuales deberán incorporar en sus procesos lo más pronto posible, la digitalización de sus datos, conectividad segura entre equipos industriales y procesos a nivel local y/o remota, ya que esta 4ta revolución exige una manufactura más eficiente para mantener la competitividad en el mercado y el uso más eficiente de los recursos, transfiriendo información relevante entre equipos industriales y móviles, datos que facilitan el conocimiento de lo monitoreado, permitiendo así su análisis y toma estratégica de decisiones en los diferentes niveles del negocio, alrededor del 82% de las organizaciones mundiales aplican la manufactura inteligente; se estima que para el 2020 se conecten 26 billones de dispositivos a nivel mundial.

#### Agradecimientos

Se agradece a CIDESI por el apoyo en la realización del presente trabajo y al cuerpo académico de automatización y control de la universidad tecnológica de Querétaro por el apoyo brindado para la aplicación de la metodología.

#### Anexo 1 Abreviaturas

<b>ARM</b>	Máquina RISC Avanzada (Advanced RISC Machine).
<b>CIP</b>	Protocolo de Control e Información (Control and Information Protocol).
<b>DSI</b>	Pantalla de Interfaz serial (Display Serial Interface).
<b>EtherNet/IP</b>	Ethernet/Protocolo Industrial (Ethernet/Industrial Protocol).
<b>FTP</b>	Protocolo de Transferencia de archivos (File Transfer Protocol).
<b>GPIO</b>	Entrada/Salida de Propósito General (General Purpose Input/Output).
<b>GPU</b>	Unidad de Procesamiento Gráfico (Graphics Processing Unit).
<b>HDMI</b>	Interfaz Multimedia de Alta Definición (High-Definition Multimedia Interface).
<b>HMI</b>	Interfaz Hombre-Máquina (Human-Machine Interface).
<b>IP</b>	Protocolo de Internet (Internet Protocol).
<b>ODVA</b>	Asociación Abierta de Vendedores DeviceNet (Open DeviceNet Vendor Association).
<b>OSI</b>	Interconexión de Sistemas Abierto (Open System Interconnection).
<b>PLC</b>	Controlador Lógico Programable (Programmable Logic Controller).
<b>RAM</b>	Memoria de Acceso Aleatorio (Random Access Memory).
<b>RISC</b>	Computadora con Conjunto Reducido de Instrucciones (Reduced Instruction Set Computer).
<b>SoC</b>	Sistema en Chip (System on Chip).
<b>TCP</b>	Protocolo de Control de Transmisión (Transmission Control Protocol).
<b>UCMM</b>	Administrador de mensaje desconectado (Unconnected Message Manager).
<b>UDP</b>	Protocolo de Datagramas de Usuario (User Datagram Protocol).
<b>USB</b>	Bus Universal en Serie (Universal Serial Bus).
<b>WiFi</b>	Fidelidad Inalámbrica (Wireless Fidelity).

## Referencias

ControlNet International and Open DeviceNet Vendor Association. (2001). *EtherNet/IP Adaptation of CIP Specification*.

Grand View Research. (2016). Human Machine Interface (HMI) Market Analysis By Product (Display Terminals, Interface Software, Industrial PC), By Application (Automotive, Healthcare, Food & Beverages, Oil & Gas, Packaging, Aerospace & Defense) And Segment Forecasts To 2022. *Market Research Report*. Obtenido de <https://www.grandviewresearch.com/industry-analysis/human-machine-interface-market>

Hardware Meets Software. (2018). HMS Release 2016 Industrial Market Share Report. *LC Automation*. Obtenido de [https://www.lcautomation.com/Page/Latest\\_News/2016\\_Industrial\\_Network\\_Market\\_Share\\_Report.aspx](https://www.lcautomation.com/Page/Latest_News/2016_Industrial_Network_Market_Share_Report.aspx)

Market Research Future. (2018). *Industrial Automation Market Research Report- Forecast 2022*. Obtenido de <https://www.marketresearchfuture.com/reports/industrial-automation-market-2212>

Prado Hernández, D. (2005). *Desarrollo de un servidor de comunicaciones para el monitoreo, supervisión y control de equipos industriales*. (Tesis inédita de ingeniería). Universidad Simón Bolívar, Sartenejas, Venezuela.

Raspberry Pi. (s.f.). THE Official RASPBERRY PI PROJECTS BOOK. Obtenido de [https://www.raspberrypi.org/magpi-issues/Projects\\_Book\\_v1.pdf](https://www.raspberrypi.org/magpi-issues/Projects_Book_v1.pdf)

Rodríguez, J., Ramírez, M., Burgara, O., & Escoto, E. (2017). Desarrollo de un sistema SCADA para monitoreo remoto de grúas RTG en la empresa LCTPC del Puerto de Lázaro Cárdenas, Michoacán, México. *Revista de Aplicaciones de la Ingeniería*. 4(12), 36-43.

Zavala, J., Domínguez, R., & Orizaba, J. (2016). Implementación de HMI para un PLC Micrologix 1100 con Raspberry Pi 2 modelo b. *Pistas Educativas*, 38(121), 260-269.