

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Título del trabajo de integración curricular

**Desarrollo de HMI para el Monitoreo de una Central de Control
de Motores para Refinería La Libertad**

Jonathan Vinicio Espín Martín

Ingeniería Electrónica

Trabajo de integración curricular presentado como requisito
para la obtención del título de
Ingeniero Electrónico

Quito, 09 de diciembre de 2019

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE CIENCIAS E INGENIERÍAS

HOJA DE CALIFICACIÓN
DE TRABAJO DE INTEGRACIÓN CURRICULAR

**Desarrollo de HMI para Monitoreo de una Central de Control de Motores
para Refinería La Libertad**

Jonathan Vinicio Espín Martín

Calificación:

Nombre del profesor, Título académico

Diego Benítez Mejía, Ph.D.

Firma del profesor:

Quito, 09 de diciembre de 2019

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante:

Nombres y apellidos:

Jonathan Vinicio Espín Martin

Código:

00125547

Cédula de identidad:

1803933660

Lugar y fecha:

Quito, 09 de diciembre de 2019

RESUMEN

El proyecto tiene como finalidad el desarrollo de una Interfaz Hombre Máquina (HMI) para el Monitoreo de una Central de Control de Motores (MCC) mediante el uso del software Ignition de Inductive Automation® basado en tecnología web, bajo el concepto de HMI de alto desempeño (High Performance HMI). Este proyecto se ha desarrollado en conjunto con el sector empresarial, específicamente con Automation Solutions Ecuador, empresa integradora del software, como un proyecto piloto para la Refinería La Libertad en la provincia de Santa Elena, Ecuador. La Central de Control de Motores está equipada con medidores multifuncionales de potencia y relés de protección digital para manejo de motores en aplicaciones de bajo voltaje, con los cuales se construirá una red de comunicación con arquitectura Modbus RTU sobre TCP/IP en relación maestro-esclavo. Una computadora industrial toma el papel de servidor de Ignition, único cliente de la aplicación, almacenador de la base de datos y maestro/cliente de la red Modbus.

Palabras clave: Base de Datos, Central de Control de Motores, HMI de Alto Desempeño, Ignition, Modbus RTU, Modbus TCP/IP

ABSTRACT

The project aims to develop a Human Machine Interface (HMI) for the Monitoring of a Motor Control Center (MCC) using Ignition by Inductive Automation®, software based on web technology, under the concept of high performance HMI (High Performance HMI). This project has been developed in conjunction with the business area, specifically with Automation Solutions Ecuador, a software integrator, as a pilot project for the La Libertad Refinery in the province of Santa Elena, Ecuador. The Motor Control Center is equipped with multifunctional power meters and digital protection relays for motor management in low voltage applications, with which a communication network with Modbus RTU architecture over TCP / IP in master-slave relationship will be built. An industrial computer takes on the role of Ignition server, the only client of the application, database store and Modbus network master / client.

Keywords: Database, Motor Control Center, High Performance HMI, Ignition, Modbus RTU, Modbus TCP / IP

TABLA DE CONTENIDO

I.	Introducción	10
II.	Metodología.....	11
A.	Medidores de la Central de Control de Motores	11
B.	Protocolo Modbus RTU sobre TCP/IP	11
1)	Trama Modbus RTU.....	12
2)	Trama Modbus TCP/IP.....	13
3)	Compatibilidad Modbus RTU y TCP/IP	13
C.	Base de datos MySQL.....	14
D.	Ignition y High Performace HMI.....	14
1)	Configuraciones de Ignition	14
2)	High Performace HMI.....	14
III.	Implementación.....	15
A.	Iniciando con Ignition y alcance del proyecto	16
B.	Entorno de trabajo en el ordenador para el HMI.....	16
1)	BIOS y sistema operativo	16
2)	Ignition en Panel PC	16
3)	Base de datos MySQL en Panel PC.....	16
4)	Pantalla táctil y teclado virtual para Panel PC.....	17
5)	Configuraciones en sistema operativo Ubuntu	17
6)	Topología de comunicación, dispositivos para HMI.....	17
C.	Dispositivos servidor/esclavo de red Modbus.....	17
1)	Configuración NPort 5232 de Moxa	17
2)	Bloque de datos de dispositivos EMP2200 y MM200	17
D.	Diseño de HMI con Ignition.....	19
1)	Tags en Ignition	19
2)	Ventanas y parámetros.....	19
3)	Componentes inherentes de Ignition	20
IV.	Experimentación y resultados	21
A.	Lectura de tags desde simulador	21
B.	Binding de propiedades de componentes	22
C.	Estructura y resultado final del HMI	22
V.	Conclusiones.....	23
VI.	Referencias	24

ÍNDICE DE TABLAS

TABLA I. Valores nominales fundamentales para MCC. refinería la libertad	11
TABLA II. Bloque de datos protocolo Modbus	12
TABLA III. Funciones de Protocolo Modbus	12
TABLA IV. Algunos colores faciles de distinguir	15
TABLA V. Reacciones psicológicas ante un color	15
TABLA VI. Mapa de registros de EMP2200	18
TABLA VII. Mapa de registros de MM200	18
TABLA VIII. Formato para el mapa de registros del MM200.....	18
TABLA IX. Designador para Modbus en Ignition	19

ÍNDICE DE FIGURAS

Fig. 1. Medidor de Potencia EMP2200 (izquierda), relé de protección digital MM200 (derecha)	11
Fig. 2. Diagrama unifilar de potencia (negro), red de comunicación Modbus (azul).....	11
Fig. 3. Relación de dispositivos en el protocolo Modbus	12
Fig. 4. Contenido de la trama Modbus RTU.....	12
Fig. 5. Contenido de la trama RTU de maestro y esclavo para la función 0x03 o 0x04 de Modbus	13
Fig. 6. Contenido de la trama Modbus TCP/IP.....	13
Fig. 7. Nport 5232 Conversor protocolo Modbus RTU a TPC/IP	14
Fig. 8. Compatibilidad Modbus RTU sobre TCP/IP	14
Fig. 9. Panel PC Advantech PCC-3100-RAE, para la aplicacio HMI.....	16
Fig. 10. Configuración del sistema operativo y último estado por perdida de energía AC	16
Fig. 11. Topología general del HMI y dispositivos	17
Fig. 12. Configuración de puertos TCP/IP y serial del NPort 5232	17
Fig. 13. Flujo de parámetros entre ventanas	20
Fig. 14. Vistazo de interfaz HMI, datos en tiempo real; menús y rango de alarmas	20
Fig. 15. Diagrama unifilar, diagrama fasorial y contenido general	20
Fig. 16 Utilización de componentes de Ignition para tendencias, rangos tolerables y rangos de alarmas	21
Fig. 17. Ruptura de la red Modbus RTU en nodo 11.....	21
Fig. 18. Activación de alarma personalizada por ruptura de nodo	21
Fig. 19. Bloque de registros de retención con valor en dirreccion 1002	21
Fig. 20. Lectura de un valor desde la red Modbus a Ignition	21
Fig. 21. Binding en propiedades para asignar un valor a un componente	22
Fig. 22. Binding para propiedad <i>tag pens</i> para gráfica de tendencias	22
Fig. 23. Estructura de manejo de ventanas de la aplicación	22
Fig. 24. Plano del MCC con cada módulo	22
Fig. 25. Diagrama de potencia con información de acceso rápido	22
Fig. 26. Diagrama de estado de la red Modbus RTU sobre TCP/IP	23
Fig. 27. Tabla de administración de usuarios para la aplicación	23
Fig. 28. Menús para cada tipo de sensor que corresponde a un módulo del MCC.....	23

Fig. 29. Contenido de algunas ventanas de la aplicación, datos en tiempo real, set points de alarmas y tendencias23

Desarrollo de HMI para el Monitoreo de una Central de Control de Motores para Refinería La Libertad, Santa Elena-Ecuador

Jonathan Vinicio Espín Martín
 Universidad San Francisco de Quito
 jespinm@estud.usfq.edu.ec

Resumen— El proyecto tiene como finalidad el desarrollo de una Interfaz Hombre Máquina (HMI) para el Monitoreo de una Central de Control de Motores (MCC) mediante el uso del software Ignition de Inductive Automation® basado en tecnología web, bajo el concepto de HMI de alto desempeño (High Performance HMI). Este proyecto se ha desarrollado en conjunto con el sector empresarial, específicamente con Automation Solutions Ecuador, empresa integradora del software, como un proyecto piloto para la Refinería La Libertad en la provincia de Santa Elena, Ecuador. La Central de Control de Motores está equipada con medidores multifuncionales de potencia y relés de protección digital para manejo de motores en aplicaciones de bajo voltaje, con los cuales se construirá una red de comunicación con arquitectura Modbus RTU sobre TCP/IP en relación maestro-esclavo. Una computadora industrial toma el papel de servidor de Ignition, único cliente de la aplicación, almacenador de la base de datos y maestro/cliente de la red Modbus.

Palabras Clave—Base de Datos, Central de Control de Motores, HMI de Alto Desempeño, Ignition, Modbus RTU, Modbus TCP/IP

I. INTRODUCCIÓN

Uno de los medios por el cual un usuario supervisa un proceso se denomina Interfaz Hombre Máquina (HMI). Se conoce al HMI como un conjunto de representaciones gráficas relacionadas con procesos industriales que utiliza el usuario para interpretar e interactuar con los Sistemas de Control de Supervisión y Adquisición de Datos (SCADA). El método y diseño con el cual se desarrolla el HMI es clave para garantizar la facilidad con la que un operador es capaz de reconocer algún evento dentro del proceso. Aunque no existe una guía exacta con la cual desarrollar un HMI, existen varias pautas que se deben seguir para su desarrollo en base a conceptos de eficacia, es decir, HMI de alto desempeño (High Performance HMI) [1].

En el pasado, gracias al desarrollo en paquetes computacionales, se crearon los primeros HMI en pantalla con gráficos esquemáticos típicamente sobrepoblados y cargados de información innecesaria. La representación del proceso industrial en detalle era posible con la desventaja que se tornó muy gráfico y literal. Como resultado, se generó una saturación de espacio y color en el área de visualización del HMI; dificultando el reconocimiento de eventos normales o anormales. Las industrias que utilizan HMI's en general, con algunas excepciones, persisten en el uso y diseño tradicional que resulta ser un limitante a las posibilidades de apariencia y funcionalidad de un HMI moderno [2].

Uno de los protocolos más utilizado para la adquisición y gestión de datos en un HMI es "Modbus", que fue desarrollado por Modicon en 1979, Modbus es una estructura de mensajería de código abierto que establece comunicación entre dispositivos inteligentes que intercambian información bajo la relación maestro-esclavo/cliente-servidor. Existen tres variaciones de tramas que Modbus utiliza: Código Estadounidense Estándar para el Intercambio de Información (ASCII), Ordenadores Remotos (RTU) y Protocolo de Control de Transmisión sobre Protocolo de Internet (TCP/IP) [3]. Para este trabajo práctico, se utilizó Modbus RTU sobre TCP/IP ya que TCP/IP aumenta la conectividad de la red y RTU por el tipo de comunicación serial de los medidores utilizados. Esta compatibilidad es posible gracias al uso de un dispositivo electrónico, que funciona como puente de comunicación entre los diferentes protocolos, denominado *NPort* de la serie 5200 de Moxa® [4].

Ignition es una herramienta poderosa para el desarrollo del HMI/SCADA moderno. Resulta extenso hablar de todas las posibilidades que ofrece Ignition, pero de ellas destaca su flexibilidad en comparación a paquetes de desarrollo de HMI similares. Sus dos principales aplicativos: Diseñador y Visión de Cliente, permiten diseñar y visualizar el proyecto respectivamente. Una licencia da acceso a un número de clientes, etiquetas y proyectos ilimitados. Ignition utiliza tecnología estándar de web que le permite a un ordenador ser un servidor, sin importar su sistema operativo, y así cumplir con los múltiples propósitos potenciados por el software. Como resultado; este software es sumamente escalable, modular y compatible bajo el marco de seguridad que requiere el medio web, en el cual se basa. Ignition, básicamente, tiene acceso a la interacción con cualquier dispositivo de terceros relacionado al campo industrial aplicable al SCADA. Su compatibilidad se amplía hacia otros servidores, bases de datos y más. Toda esta versatilidad crea varias posibilidades de arquitectura para Ignition y los componentes necesarios para el desarrollo de HMI/SCADA; dando así, paso a que varias compañías hagan uso de todas las herramientas y posibilidades que este software ofrece [5].

El HMI para este proyecto, se basa en los lineamientos y normas utilizados por la compañía Automation Solutions Ecuador. ASE es una compañía ecuatoriana de integración de sistemas de control que empezó sus operaciones en 2004. Uno de los pilares de la compañía es el desarrollo de interfaces bajo el concepto de High Performance HMI y el uso de Ignition como paquete de software industrial [6]. Un contrato realizado con EP Petroecuador; Refinería La Libertad en la provincia de Santa Elena, abre las puertas para incorporar Ignition en su entorno. La contratación del

desarrollo del HMI se aplica para un sistema de monitoreo de una Central de Control de Motores (MCC), con diversas funciones en el proceso de refinamiento del petróleo crudo. Esta central está equipada con dispositivos medidores multifuncionales de potencia y relés de protección digital para manejo de motores en aplicaciones de bajo voltaje. Con estos dispositivos se construye una red Modbus RTU sobre TCP/IP adquiriendo los registros de interés. El servidor; en este caso, es un Panel PC táctil marca *Advantech* configurado para un entorno industrial, con sistema operativo Ubuntu. En este ordenador coexiste el servidor de Ignition, el cliente que ejecuta la aplicación y la base de datos que almacena registros de los dispositivos que utilizan Modbus y eventos sobre el proceso industrial.

II. METODOLOGÍA

A. Medidores de la Central de Control de Motores

El MCC *General Electric Evolution E9000* de EP Petroecuador, Refinería La Libertad; se encarga de proveer energía eléctrica, proteger, monitorear y controlar diversos arrancadores de motores. En el proceso de refinamiento existen varias aplicaciones asociadas al uso de un motor en el bombeo de petróleo crudo y sus derivados. El MCC tiene un arreglo de seis secciones incluyendo la del interruptor termo-magnético principal o disyuntor. Los módulos distribuidos por el panel respetan un mapa de coordenadas alfanuméricas para su fácil identificación, por ejemplo: AAM800 o ABE-30 que representa el módulo que potencia del MCC y el módulo de un arrancador par un motor de 30 HP, respectivamente. En total; existen veintiún módulos arrancadores asignados a cada motor, tres alimentadores de alta corriente, el interruptor principal y canales para cableado de fuerza y control.

Como información básica pero fundamental, este MCC maneja valores nominales de voltaje, corriente y frecuencia como se indica en la TABLA I, para su funcionamiento estándar. Estos tableros son fabricados bajo estándares de protección y desempeño que aseguran que mantendrá o detendrá su funcionamiento ante condiciones adversas, además de la seguridad que brinda a las personas en las proximidades.

TABLA I. VALORES NOMINALES FUNDAMENTALES PARA MCC. REFINERÍA LA LIBERTAD

Valores Nominales para Central de Control de Motores E9000			
Corriente (A)	Fases	Voltaje (V ac)	Frecuencia (Hz)
800	3 Fases, 3 Cables	480	60

Los dispositivos que se utilizan para este trabajo práctico son los medidores multifuncionales de potencia y relés inteligentes de protección digital para motores de baja tensión, instalados en la acometida de potencia del MCC y en cada módulo arrancador de motores; una baja tensión se refiere a un rango de mil voltios ac o inferior [7]. Comercialmente llamados *Multilin™ EPM 2200 Power Meter* y *MM200 Motor Management System* respectivamente, ambos marca *General Electric*, como se indica en la Fig. 1.



Fig. 1. Medidor de Potencia EMP2200 (izquierda), relé de protección digital MM200 (derecha)

A estos dispositivos de medición se los llamará simplemente EMP2200 y MM200. El medidor EMP2200 va anclado a la red principal de distribución de potencia que alimenta el MCC y proporciona información sobre la red de energía. Los medidores MM200 están conectados a cada módulo del MCC y proporcionan información general de cada motor individualmente. La Fig. 2 ilustra un diagrama unifilar las conexiones de potencia y comunicación de los medidores que forman parte del MCC. La red en potencia se representa en negro y la de comunicación Modbus en azul, el medidor EMP2200 es el primer nodo en el recuadro y a partir del dos al veintidós son medidores MM200.

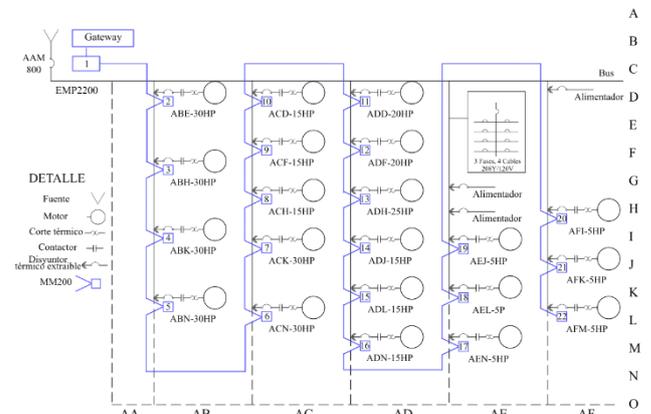


Fig. 2. Diagrama unifilar de potencia (negro), red de comunicación Modbus (azul)

A partir de la comprensión de la red de potencia y estructura de conexión de la red de comunicación, se puede realizar un análisis más detenido sobre la red de comunicación Modbus en cadena. Estos medidores utilizan la comunicación serial a través del par trenzado RS-422 para Modbus RTU y Ethernet para Modbus TCP/IP según la capa física del Modelo *Open System Interconnection* (OSI) [8].

B. Protocolo Modbus RTU sobre TCP/IP

Como se indica en la Fig. 2, la red de comunicación es fundamental para el intercambio de información utilizada en el HMI. El empaquetamiento de bits es distinto en Modbus RTU y TCP/IP, por tal motivo, para mantener la cadena de información se debe conocer el contenido de las tramas de cada variación con el propósito de hacerlas compatibles. Se entiende como trama a un conjunto de bits ordenados que representan o tienen una función en particular en determinado protocolo [8]. El principio del protocolo Modbus se basa en la comunicación entre pares con una relación de maestro-esclavo o cliente-servidor en la que el maestro envía una petición y el esclavo responde como se indica en la Fig. 3.

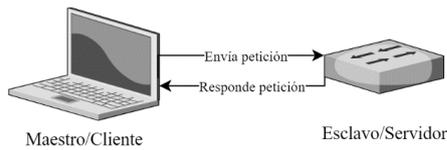


Fig. 3. Relación de dispositivos en el protocolo Modbus

Una característica del protocolo Modbus es dar diferentes funciones a un conjunto de bits transmitidos. Esto permite leer o escribir datos de múltiples formas desde o hacia un esclavo. Previo a la transmisión de bits, estos se almacenan en tablas o bloques de datos Modbus. En la TABLA II se observan los bloques de datos, su acción y dirección. Dos de esas funciones almacenan valores discretos y las otras dos, trabajan con registros de 16 bits. Las que almacenan valores discretos sirven para la lectura de datos mientras que los registros tienen la funcionalidad de lectura y escritura de datos. Cada bloque tiene una tabla de espacios con 65,535 posibles direcciones, pero es usual solamente utilizar 9,999 direcciones.

TABLA II. BLOQUE DE DATOS PROTOCOLO MODBUS

Tabla de dato	Acción	Dirección
Bobinas	Leer/Escribir	00001-09999-065535
Entradas discretas	Solo lectura	10001-19999-165535
Registro de entradas	Solo lectura	30001-39999-365535
Registro de retención	Leer/Escribir	40001-49999-465535

Con el almacenamiento de bits en los diferentes bloques se pueden realizar diversas acciones utilizando funciones Modbus. En la TABLA III se observan las funciones más comunes del protocolo Modbus con su respectivo código funcional. El código funcional determinará la acción del conjunto de bits transmitidos desde o hacia una dirección especificada en un bloque de datos [9].

TABLA III. FUNCIONES DE PROTOCOLO MODBUS

Código (Función)	Acción
1 (0x01)	Leer bits de salida
2 (0x02)	Leer bits de entrada
3 (0x03)	Leer registros de salida
4 (0x04)	Leer registros de entrada
5 (0x05)	Escribir un bit
6 (0x06)	Escritura de un registro
7 (0x07)	Leer estado de excepción
8 (0x08)	Diagnóstico
15 (0x0F)	Escritura múltiples bits
16 (0x10)	Escritura múltiples registros
17 (0x11)	Reporte ID de Esclavo

La dirección en cada bloque de registros de Modbus es de 16 bits, una de las limitaciones de 16 bits es que no permiten representar por sí solos valores de tipo flotante (*float*), doblé (*double*), palabra (*string*), entre otros. Por este motivo, una solución muy simple consiste en agrupar n (bytes), donde n es un número natural, y dependiendo de la función aplicada se pueden empaquetar varios bytes en un solo mensaje. Gracias a esta solución es posible utilizar este protocolo para representar valores de alto o bajo orden de magnitud o strings. Un típico ejemplo es el dato de tipo float, muy utilizado en este trabajo práctico, el cual utiliza el estándar IEEE-754 punto flotante con codificación de 32 bits.

1) Trama Modbus RTU

En la Fig. 4 se indica el agrupamiento del contenido de la trama con protocolo Modbus RTU, para un caso general. Este contenido puede variar según si el maestro, es el que solicita una acción o un esclavo que responde una petición. De igual forma, el tamaño de la trama o número de bits que ejecuta el maestro cambia según la función Modbus utilizada. En el caso de la trama del esclavo, el número de bits cambia en función a la petición del maestro y la función que ejecute. La función con la cual el tamaño de la trama de respuesta del esclavo puede extenderse significativamente, es la 0x03 o 0x04 de Modbus según el apartado II-B TABLA III.

Silencio inicial	Dir. Esclavo	Función	Dato	CRC ^a	Silencio final
3.5T caracteres	8 bits	8 bits	$n \times 8$ bits	16 bits	3.5T caracteres

^a. Revisión Cíclica Redundante

Fig. 4. Contenido de la trama Modbus RTU

a) Silencio de inicio y final

El silencio determina cuando una trama comienza y termina su primer o último bit de transmisión. Esta función es dependiente del tiempo y en este lapso de silencio el transporte de bits es nulo. Se utiliza bits por unidad de tiempo para dimensionar este valor, pero en el contexto de comunicación serial, la tasa de bits es equivalente a la velocidad en baudios (*bauds*) [10]. En (1) se muestra como determinar el tiempo T de silencio de la trama.

$$T(ms) = \frac{1000 \text{ bits}}{\text{baud}} \quad (1)$$

Donde T es el tiempo en milisegundos, *bits* es el número de bits de la trama, llamados caracteres y *baud* según la literatura, son valores estándares de velocidad, para este trabajo práctico se buscará la velocidad óptima entre 9600 a 115200 bauds. Una observación importante es que el maestro y todos los esclavos deben configurarse a la misma velocidad en bauds para evitar inconsistencias y no traslapar la información binaria.

b) Dirección de esclavo

Al ser 8 bits, se puede alcanzar teóricamente 254 esclavos o nodos, cada uno con una dirección única. El conteo de los nodos comienza en 1, ya que en la red Modbus RTU el cero está reservado para una lista de difusión para todos los nodos, manejada por el maestro. En la realidad es recomendable un máximo de 247 esclavos dependiendo del medio físico de comunicación y de la distancia física entre ellos [11].

c) Función y Dato

Como se mencionó en el II-B TABLA II y II-B TABLA III, la función determina, que se debe hacer con los datos y a que bloque acceder para leer o escribir un valor discreto o registro. A este conjunto de bits se los conoce como Unidad de Datos de Protocolo (PDU) y se los utiliza en Modbus para cualquier variación de la trama. Para este trabajo práctico las funciones 0x03 y 0x04 de Modbus son utilizadas, en la Fig. 5 se describe en detalle el contenido de las tramas utilizadas para el maestro y esclavo. Dependiendo de la acción de solicitud o respuestas, el tamaño de la trama puede variar. La trama del maestro tiene un tamaño definido, por

otro lado, la trama del esclavo cambia su tamaño en función de los bits de conteo de registros que solicite el maestro. Para conocer el máximo número de bits del dato de respuesta, se puede observar el valor del conteo de bytes en la sección del dato de la Fig. 5. El conteo de bytes está limitado por 8 bits, es decir, teóricamente hasta 255 registros de 16 bits pueden ser la respuesta del esclavo como dato de respuesta para esta trama, pero en la realidad, un máximo de 125 registros pueden ser transmitidos por trama por las limitaciones impuestas por este protocolo [12].

Modo	Silencio inicio	Dir. Esclavo	Función	Dato		CRC	Silencio final
				Dir. Inicio	# de Registro		
Maestro	3.57 caracteres	8 bits	8 bits	16 bits	16 bits	16 bits	3.57 caracteres
				Contar Byte	Dato		
Esclavo	3.57 caracteres	8 bits	8 bits	8 bits	$n \times 16$ bits	16 bits	3.57 caracteres

Fig. 5. Contenido de la trama RTU de maestro y esclavo para la función 0x03 o 0x04 de Modbus

d) Revisión CRC de 16 bits

Este algoritmo de revisión de errores consiste en el uso de un polinomio de Revisión Cíclica Redundante (CRC) cuyo valor es 0xA001 invertido. El desplazamiento de bits y la operación booleana XOR son utilizados en este algoritmo. Ya que esta revisión se basa en 16 bits, si el mensaje excede o carece de esta cantidad de bits no tiene importancia para este algoritmo. En el caso de escases de bits, el mensaje se rellena con ceros (*zero padding*) y en el caso que exceso de bits, se inicia con los bits menos significativos del mensaje y en cada ciclo se desplaza hacia el bit más significativo, un bit a la vez. Para ambos casos, el ciclo se repite una cantidad de veces igual al tamaño del mensaje sin zero padding. Este método tiene una probabilidad de fallo de 1 entre 65,536 [13]. El algoritmo sigue el siguiente esquema [14]:

- Primero, el algoritmo CRC inicia con dieciséis 1's y se realiza la operación XOR con los 16 bits menos significativos del mensaje.
- Segundo, se debe hacer un desplazamiento (*shift*) del bit menos significativo y rellanar el vacío con el siguiente bit del mensaje (bit 17 en el primer shift). Si no existe más bits del mensaje se rellana con 0's.
- Tercero, se debe notar si el bit desplazado es 1 o 0, si es 0 continua el siguiente shift; si es 1 se realiza la operación XOR entre el polinomio CRC y el mensaje actual sin el bit desplazado, y se continua al siguiente shift.
- Cuarto, se debe realizar el segundo y tercer paso la cantidad de veces del tamaño de bits del mensaje sin zero padding.
- Quinto, se debe intercambiar los 2 bytes del cuarto paso entre sí y obtener el código CRC para la trama Modbus RTU.

2) Trama Modbus TCP/IP

La trama Modbus TCP/IP tiene cierta similitud con Modbus RTU. La diferencia más importante está en que no utiliza CRC y un conjunto de bytes iniciales llamados Encabezado de Protocolo de Aplicación Modbus (MBAP). En la Fig. 6 se observa el contenido general de la trama de

este protocolo; conformado por: el Identificador de transacción, el Identificador de Protocolo, la Longitud y el Identificador de Unidad representa el encabezado MBAP [11].

Identificador Transacción	Identificador Protocolo	Longitud	Identificador Unidad	Función	Dato
16 bits	16 bits	16 bits	8 bits	8 bits	$n \times 8$ bits

Fig. 6. Contenido de la trama Modbus TCP/IP

a) Identificador de Transacción

Este conjunto de bits se emite por el servidor Modbus TCP y se utiliza para asociar la respuesta a una solicitud. Esto permite completar una transacción de información a pesar de que el orden de respuesta no sea el mismo al orden de solicitud [14].

b) Identificador de Protocolo

El protocolo Modbus TCP no es el único que utiliza los principios del protocolo TCP. Para asegurar que se utilizará TCP para Modbus, se llena con dieciséis 0's esta sección de la trama.

c) Longitud

Estos bytes sirven para contar el contenido de bits de los siguientes campos. La acción de conteo de bytes ayudará a estimar el espacio de memoria que necesitará el procesador que se encarga de la transmisión de información. En este punto se completa la sección TCP e inicia la sección IP y el encapsulamiento de Modbus. De igual forma, una inconsistencia con la longitud esperada de la trama sugiere un error en la trama y el dato no es confiable [11].

d) Identificador de Unidad

Este conjunto se parece mucho a la dirección del esclavo de RTU, pero aplicado a TCP. Para esta aplicación, se utiliza características del Protocolo de Internet (IP), por esta razón se denomina al protocolo completo como TCP/IP. De igual forma, la relación cliente-servidor se mantiene en esta variación de Modbus. El uso de este protocolo está relacionado con el dispositivo de compatibilidad de tramas, los puertos físicos del ordenador maestro/cliente y con los medidores EMP2200 y MM200. La topología de esta red de comunicación será descrita a detalle posteriormente en el apartado III.B.6).

e) Función y Datos

Debido a que se utiliza la PDU del protocolo Modbus en TCP/IP y RTU, es decir, la función y datos de la trama Modbus; el conjunto de bits tiene la misma función que se especifica en el apartado II-B-1)c).

3) Compatibilidad Modbus RTU y TCP/IP

Conociendo el contenido de las tramas en los distintos protocolos, es notable que la compatibilidad entre ellas es posible. Desde el punto de vista de sistemas digitales, se puede aplicar hardware para el tratamiento de compatibilidad de las variaciones de Modbus. La industria ha desarrollado equipos electrónicos con este fin, es así el caso de los dispositivos de *Moxa*®. Para este trabajo práctico se utiliza el dispositivo *NPort 5232*, Fig. 7, compatible con el par trenzado RS-422/485 para RTU y cable Ethernet para TCP/IP según la capa física del Modelo OSI [8].



Fig. 7. Nport 5232 Conversor protocolo Modbus RTU a TPC/IP

La forma exacta con la que este dispositivo trabaja con las tramas de ambas variaciones del protocolo Modbus es reservada, pero se puede asumir el posible tratamientos y encapsulamiento de tramas como se observa en la Fig. 8.

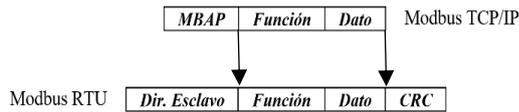


Fig. 8. Compatibilidad Modbus RTU sobre TCP/IP

El dispositivo debe considerar muchos elementos adicionales del protocolo Modbus como revisión cíclica redundante, códigos de excepción, silencios, entre otros; con el fin de que la comunicación entre tramas sea consistente. En este caso, el dispositivo NPort 5232 se utiliza como maestro de la red RTU y de servidor de la red TCP/IP [15].

C. Base de datos MySQL

La necesidad de almacenar información es importante para gestionar y analizar datos, registros e históricos. Existen varios paquetes de software dedicados al procesamiento y almacenamiento de información. Para este trabajo práctico, se utiliza el servicio de Lenguaje de Estructura y Solicitud (*MySQL*) con esquema e instancia local. Por otro lado, *MySQL Workbench* es un software que tiene una Interfaz Gráfica para Usuario (GUI) con herramientas que facilitan el trabajo de escritura de código propio de *MySQL* que sería por sí solo. El beneficio que brinda este software es utilizar el servicio de base de datos para administrar la información de forma dinámica mediante la creación y manejo instancias locales y esquemas. Se entiende por esquema a una estructura lógica de datos cifrada y una instancia local como el servicio de almacenamiento y procesamiento que se encuentran en el mismo ordenador de donde se obtiene los datos [16]. La información almacenada sin transmitirla, procesarla ni interpretarla es inútil; para este trabajo práctico, el protocolo de comunicación entre Ignition y la base de datos para transmitir información es IP, se utiliza el protocolo de huésped local 127.0.0.1 (*localhost*). Además, es necesario especificar el puerto de acceso del protocolo para evitar conflictos con el firewall del ordenador. La información transmitida es leída, administrada e interpretada por Ignition.

D. Ignition y High Performace HMI

1) Configuraciones de Ignition

Ignition es un software que facilita el desarrollo de aplicaciones para HMI/SCADA con posibilidad de potenciarlas para HMI de alto desempeño (*High Performance HMI*). Gracias a su tecnología de desarrollo, Ignition es compatible con múltiples sistemas operativos y

sus arquitecturas. Su alta versatilidad, modularidad y compatibilidad le ha permitido destacar como software industrial. Ignition; convierte un ordenador en una puerta de acceso (*Gateway*) para administrar todos sus recursos como: aplicaciones HMI, proyectos, bases de datos, ejecutar los clientes de las aplicaciones, entre otras. El Gateway es configurable por medio de un navegador web sin necesidad de conexión a Internet. Estas configuraciones se dividen en tres grupos principales: Casa (*Home*), Estado (*Status*) y Configuraciones (*Config*). En “Home” se tiene información sobre las aplicaciones complementarias descargables de Ignition y documentación completa sobre este software. En “Status” se tiene una visión general de los recursos y conexiones que demanda el Gateway del ordenador donde está instalado, además indica el estado de conexiones o desconexiones de bases de datos, dispositivos, y más. En “Config” se tiene disponible de una extensa variedad de opciones que permiten administrar y moldear el desempeño, características y conexiones configurables del servidor para darle al Gateway un uso específico [5]. Las posibilidades que Ignition ofrece como software industrial resultan ser demasiado extensas para su completo análisis. Sin embargo, se pueden describir las más importantes para el desarrollo de este trabajo práctico.

Una función de Ignition, para una aplicación básica de monitoreo, es la conexión con una base de datos y un dispositivo distribuidor o receptor de información gestionable. El protocolo de comunicación industrial que Ignition utiliza para enlazarse con dispositivos de terceros, es el denominado: Unir e Incluir Objetos (OLE), parte del acrónimo, OLE para Procesos de Control con Arquitectura Unificada (OPC UA). Este protocolo estándar de comunicación fue diseñado para HMI/SCADA aplicado al sector industrial, su principio de funcionamiento es genérico, lo cual resuelve problemas relacionados a la compatibilidad y controladores (*drivers*) de dispositivos de diferentes manufacturas. Para este trabajo se utiliza el OPC UA para Modbus RTU sobre TCP/IP [5].

Administrar y almacenar la información es igual de importante que recibirla y transmitirla. La conexión con la base de datos *MySQL* se la realiza levantando un servidor local para enlazarlo a Ignition por medio de sus configuraciones. Una vez creado un esquema para cifra la información e iniciado una instancia local para indicar donde almacenar y desde donde se debe extraer los datos, se crea un ambiente gestionable de información para Ignition. Algunos de los beneficios de mantener un enlace con una base de datos para el proceso de diseño del HMI yace en la presentación de datos por medio de gráficos de tendencia e históricos, además es posible mantener un historial de alarmas que se registran en un diario de alarmas (*journal*). Además, en una base de datos se pueden almacenar perfiles sobre elementos o personas, recetas, inventarios, registros de empleados y sus actividades, entre otras; todo esto gracias al enlace entre Ignition y la base de datos [5].

2) High Performace HMI

Una aplicación de HMI organizada permite presentar de la mejor manera los datos, las representaciones gráficas y el contenido en general de la información gestionable. Una representación intuitiva y concisa, pero útil, califica como de alto desempeño. Sin embargo, para que un HMI cumpla

con los principios de visualización de alto desempeño es necesario que presente ciertas propiedades [2] [17].

- La perceptibilidad, es un medio utilizado para hacer el contenido entendible, asegurando que el HMI responda exactamente según las acciones que el usuario ejecute.
- Interacción, fundamental para dar una perspectiva de flexibilidad, sirve para presentar la información de formas distintas desplegando opciones y con varias formas de visualización. Esto se logra gracias al uso de menús, selección de idiomas, animaciones, entre otras.
- La maniobrabilidad, se refiere a que los elementos destacados del HMI tengan operatividad. De esta forma el usuario puede asociar de forma directa entre lo representado y la vida real.
- La afinidad, con un simple vistazo el usuario del HMI debería entender exactamente lo que observa en pantalla sin importar el grado de complejidad del proceso.

De igual forma se puede describir ciertos principios de diseño para la visualización de la información. La interfaz debería ser adaptada para que un usuario sin práctica pueda entenderlo sin mayor dificultad. La representación del proceso debe ser sistemática y estructurada, de tal forma, se pueda ir incrementando o reduciendo la complejidad de la presentación en la interfaz. Un principio fundamental es diseñar el HMI conforme al comportamiento humano, es decir, el uso de elementos visuales y auditivos que tengan un efecto cognitivo sobre las personas y su sensibilidad [17].

- El principio de adopción de elementos visuales se basa en representaciones. Estas representaciones son gráficos o figuras que deben ser compactas y genéricas. El efecto monolítico sobre las figuras de representación es fundamental para despejar la carga visual en la interfaz.
- El principio de evasión de desorden se basa en la eliminación de objetos redundantes que despiden la atención del usuario. La planificación y entendimiento completo sobre el proceso que se busca representar a través de un HMI, es muy importante para el diseñado desde el inicio. De igual forma, es necesario conocer las funcionalidades de su software de diseño.
- Otro principio es el de uso de color para la distinción entre elementos principales, secundarios y de fondo. El ojo humano es sensible a las percepciones visuales relacionadas al contraste entre colores. Una persona debería notar instintivamente que ha cambiado su panorama visual simplemente por el cambio de colores entre elementos principales y de fondo. Una lista de colores contrastantes que generan este efecto cognitivo, se observa en la TABLA IV.
- El principio de adopción razonable de color, se basa en los efectos emocionales que transmite un color a una persona. Las reacciones psicológicas que generan ciertos colores en una persona, determinan el estado de ánimo y la acción

consecuente al estímulo visual. En la TABLA V se observan los efectos psicológicos de ciertos colores sobre una persona.

- El principio de diseño razonable, se basa en que el diseño debe buscar el agrupamiento de elementos con las mismas características y funciones. La simetría y distribución de estas agrupaciones también es importante para distinguirlos entre sí, sin dispersarlos en el espacio de diseño.
- El principio de emoción y descubrimiento debe generar al usuario una necesidad de exploración y análisis. Se debe transmitir emociones de atención y de calma sin que se genere monotonía ni desenfoque.
- El principio de acceso por múltiples vías ayuda al operador a evitar memorizar los patrones de llegada a alguna parte específica del HMI, como una ventana. Con este principio, el diseño debe permitir al usuario acceder a la información por vías diferentes con el cuidado de no caer en redundancias ni saturación de información.

TABLA IV. ALGUNOS COLORES FACILES DE DISTINGUIR

<i>Color Principal</i>	<i>Color de Fondo</i>
Amarillo	Negro
	Violeta
Azul	Amarillo
Blanco	Negro
	Violeta
	Azul
	Verde
Negro	Amarillo
	Blanco
Verde	Amarillo

TABLA V. REACCIONES PSICOLÓGICAS ANTE UN COLOR

	<i>Reacción Psicológica</i>
Amarillo	Esperanza, alegría, calidez
Azul	Igualdad, calma, frío
Blanco	Inmaculado, santidad, pureza, verdad
Gris	Simple, liviano, suave
Naranja	Placer, irritabilidad, moderado
Negro	Sólido, pesado, marcado
Rojo	Emocionante, brillante, peligro
Verde	Validad, frescura, esperanza
Violeta	Misterio, elegancia, lujoso

El estudio de estas propiedades y principios permiten alcanzar un High Performace HMI. Ignition dispone de elementos y opciones de diseño que permiten fácilmente cumplir con las propiedades y propiedades de un diseño de alto desempeño. Además, Ignition cuenta con una plataforma de aprendizaje virtual en donde se trata el manejo de High Performace HMI y en base a ello, se realiza la selección de colores y distribución de elementos para el diseño del sistema de monitoreo. De igual forma, ASE se basa en los mismos principios de diseño descritos para el desarrollo de sus HMI's.

III. IMPLEMENTACIÓN

El objetivo principal de este trabajo práctico es diseñar un sistema de monitoreo para un MCC, utilizando Ignition de Inductive Automation® como software de diseño

aplicando conceptos de High Performace HMI. El monitoreo contiene datos en tiempo real, históricos, tendencias, alarmas, entre otras. Para cumplir con este objetivo es necesario un proceso meticuloso en cada una de las partes de diseño para obtener el resultado esperado. Previo al desarrollo del diseño, es necesario el entendimiento de las características de un High Performace HMI y comprender el proceso industrial que se representa. Las conexiones entre Ignition con la base de datos y los dispositivos de medición, son primordiales para iniciar el proceso de configuración y diseño del HMI. De igual forma, comprender los protocolos de comunicación de los dispositivos en campo EMP2200 y MM200, así como la compatibilidad de tramas RTU sobre TCP/IP. La configuración de un entorno de trabajo adecuado en el ordenador portable desde el BIOS y el sistema operativo, es necesario para que el HMI sea apto para el uso industrial.

A. Iniciando con Ignition y alcance del proyecto

Ignition cuenta con una plataforma virtual llena de instructivos, guías y documentación sobre el software en su página web oficial [5]. *Inductive University* [20] es una plataforma web con cursos virtuales básicos, de acceso libre, relacionado a las capacidades y funcionalidades de Ignition. En este curso, se describen los conceptos de un High Performace HMI y los diferentes elementos representativos que dispone el software para desarrollar una aplicación de alto desempeño. La interfaz del HMI se desarrolla basándose en los conceptos descritos en el apartado II.D. Una vez comprendido las capacidades de Ignition, se requiere entender el objetivo del proceso industrial que se representa a través del HMI. Una serie de reuniones con los directivos del proyecto fue necesaria para entender el alcance del proyecto. Se espera obtener un sistema de monitores para un MCC sin ninguna acción de control. El sistema de monitoreo se encarga de obtener información sobre el estado de la red de potencia eléctrica que recibe y alimenta al MCC. De igual forma, se encarga de recibir datos de arrancadores vinculados a motores, involucrados en el proceso de refinamiento del petróleo crudo.

B. Entorno de trabajo en el ordenador para el HMI

El dispositivo idóneo para el contenido de los elementos del HMI es un Panel Computadora Portable (*Panel PC*). Marca *Advantech* modelo *PPC-3100S-RAE*, Fig. 9, es un ordenador ultra delgado carente de ventilador, con cuatro procesadores Intel Celeron® N2930 de 1.86 GHz de bajo consumo de potencia para evitar recalentamiento. Con una pantalla táctil de 10.4" LED de resolución 800×600, con frecuencia de actualización de 60,32 Hz y un panel frontal con protección IP65; además incluye puertos estándares USB 3.0, Ethernet, entre otros [18]. Este dispositivo es una opción viable para el uso industrial del HMI desarrollado.



Fig. 9. Panel PC Advantech PCC-3100-RAE, para la aplicacio HMI

1) BIOS y sistema operativo

El contrato para el desarrollo del HMI no contempla una licencia para sistema operativo Windows, MacOS o algún sistema operativo de paga; por esta razón, se utiliza Ubuntu 18.04.2 LTS (*Long Term Support*). La utilización de este sistema operativo presenta ciertos desafíos en el condicionamiento de todas sus características de funcionamiento para ejecutar un HMI utilizable. En el proceso de Resolución de Problemas (*Troubleshooting*) se resolvió problemas de encendido y apagado del Panel PC mediante configuraciones del BIOS. Además, se configuró el ordenador para que recupere el último estado de encendido o apagado, en caso de pérdida de energía AC como se indica en la Fig. 10.

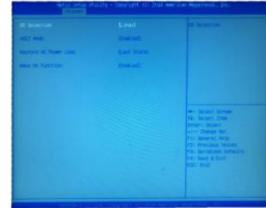


Fig. 10. Configuración del sistema operativo y último estado por perdida de energía AC

Una vez que el ordenador no presenta problemas de encendido ni apagado, se inicia con las configuraciones de instalación de softwares necesarios para una operación industrial. Se procede con la instalación de la base de datos MySQL, Ignition, softwares complementarios y un asistente de teclado táctil virtual; todo con compatibilidad para Ubuntu.

2) Ignition en Panel PC

El instalador de Ignition 8.0.2 para Ubuntu se encuentra fácilmente en la página web. Después de la instalación del Gateway, es necesario instalar las aplicaciones adicionales del Diseñador (*Designer*) y Visualización de Cliente (*Visual Client*) para potenciar todas sus características. Por medio de estas aplicaciones es posible conectarse al Gateway local y fácilmente diseñar y visualizar el proyecto dedicado al monitoreo. Ignition es el motor del sistema HMI, en el gateway se realiza y se guarda todo tipo de transacciones que sean requeridos por el proyecto. Una vez iniciada la base de datos y los dispositivos bajo protocolo OPC UA, se pueden realizar las configuraciones en Ignition para utilizarlos. En esta aplicación de monitoreo, es necesario la configuración de alarmas, Ignition utiliza la base de datos para almacenar estas alarmas y un diario de ellas. Así mismo, por medio de la base de datos almacena datos y registros con los que se podrán mostrar históricos y tendencias. Finalmente, la licencia comprada para este HMI es introducida una vez culminado el proyecto y es enlazada permanentemente con el ordenador que contenga el gateway. Como medida de seguridad es posible realizar un respaldo total del gateway y de esta forma evitar perder el contenido de la aplicación HMI y sus configuraciones.

3) Base de datos MySQL en Panel PC

El levantamiento del servicio MySQL para la base de datos resulta tener una cierta complejidad en sistema operativo Ubuntu. Los comandos del terminal para activar este servicio e instalar MySQL Workbench en Ubuntu son los siguientes:

```

1 sudo apt-get install mysql-server
  mysql-client
2 sudo apt-get install mysql-workbench
3 sudo mysql_secure_installation
4 service mysql stop
5 sudo mkdir /var/run/mysql
6 sudo chown mysql: /var/run/mysql
7 sudo mysqld_safe --skip-grant-tables --
  skip-networking &
8 sudo mysql -u root mysql
9 update mysql.user set
  authentication_string=password('password')
  , plugin='mysql_native_password' where
  user 'root' and host='localhost';
10 sudo mysqldadmin -S
  /var/run/mysql/mysql.sock shutdown
11 service mysql start
12 mysql -u root -p

```

4) Pantalla táctil y teclado virtual para Panel PC

Debido a que la pantalla del Panel PC es táctil, es necesario configurar el driver para la calibración al tacto sobre la pantalla. Los drivers de calibración para esta pantalla táctil del Panel PC, en particular para el sistema operativo Ubuntu, es una cuestión compleja de resolver. Mediante búsqueda e investigación, un contacto directo con el servicio técnico de *Advantech* en línea, se logró obtener el driver adecuado y configurarlo para que la pantalla responda correctamente, el driver calibrador es *PenMount* para Ubuntu. Adicionalmente, se instala el software *Onboard* para Ubuntu cuyo propósito es desplegar un teclado virtual para el Panel PC en pantalla. Esto se debe a que, en campo, este dispositivo solo dependerá de la pantalla táctil para la interacción e interpretación de gestos sobre la pantalla realizados por el usuario.

5) Configuraciones en sistema operativo Ubuntu

Ya que este equipo estará en un entorno industrial y su uso es exclusivo de la aplicación HMI de monitoreo, es necesario deshabilitar y habilitar ciertas opciones del sistema Ubuntu para evitar interrupciones durante la ejecución de la aplicación. Las acciones más importantes son:

- Habilitar el inicio de sesión automático en el encendido de la máquina.
- Revisar actualizaciones de último momento y posteriormente desactivarlas.
- Apagar notificaciones no deseadas que puedan interrumpir mientras se ejecute la operación de monitoreo.
- Por medio de la aplicación *Gnome Tweak* para Ubuntu, configurar la escala de la fuente de la interfaz en general, debido a que el tamaño de la pantalla es pequeño.
- Otorgar permisos y privilegios a carpetas específicas dentro del sistema operativo donde se guarda información importante sobre el HMI como la base de datos y los instaladores de los softwares utilizados.

6) Topología de comunicación, dispositivos para HMI

La topología comprendida por el HMI está compuesta por el gateway, panel PC, con Ignition y la base de datos; conectada con el servidor NPort 5232 con protocolo

Modbus TCP/IP. Seguimiento del dispositivo EMP2200 y los veintidós dispositivos MM200 por medio de protocolo Modbus RTU. En la Fig. 2 se muestra una versión de la topología que el usuario del HMI tendrá en pantalla, en la Fig. 11 se muestra una topología resumida y global sobre la topología del HMI y sus elementos.



Fig. 11. Topología general del HMI y dispositivos

C. Dispositivos servidor/esclavo de red Modbus

Los dispositivos de la red Modbus dominados por el cliente/maestro son los dispositivos NPort 5232 y los medidores EMP2200 y MM200 por medio del Panel PC con Ignition. Para efectos prácticos, se utilizó un simulador Modbus para representar las respuestas del EMP2200 y el MM200 en la red Modbus mientras se desarrolla la aplicación. Para el dispositivo de compatibilidad de protocolos, es necesario configurar en el gateway el dispositivo OPC UA como Modbus RTU sobre TCP/IP, por tal razón, se requiere la utilización física del este dispositivo acoplada al simulador Modbus RTU.

1) Configuración NPort 5232 de Moxa

El medio físico de conexiones del dispositivo para la sección TCP/IP se realiza por cable Ethernet entre el Panel PC y el NPort 5232. Por medio de un navegador web es posible conectarse al NPort 5232 por medio de protocolo IP versión 4 (IPv4) y acceder a su interfaz de configuraciones. Una vez configurada la dirección IP y la máscara de red, se puede establecer fácilmente la comunicación entre el Gateway y el NPort 5232. Como se indicó previamente, se establece que el dispositivo NPort 5232 es un servidor en Modbus TCP/IP para el gateway de Ignition. Además, se utiliza uno de los puertos seriales (puerto 1) donde se configura la velocidad en bauds, cantidad de bits, y el medio físico RS 485 de 2 cables trenzados como se observa en la Fig. 12.

Monitor Line							
Line							
Port	OP Mode	IP1	IP2	IP3	IP4		
1	TCP Server Mode	192.168.10.133	Listen	Listen	Listen		
2	Real COM Mode	Listen					

Serial Settings								
Serial Settings								
	Alias	Baud rate	Data bits	Stop bits	Parity	FIFO	Flow ctrl	Interface
Port 1		9600	8	1	None	Enable	None	RS-485 2Wire
Port 2		115200	8	1	None	Enable	RTS/CTS	RS-485 2Wire

Fig. 12. Configuración de puertos TCP/IP y serial del NPort 5232

2) Bloque de datos de dispositivos EMP2200 y MM200

Los medidores multifuncionales de potencia EMP 2200 y relés de protección digital MM200 son los dispositivos que proporcionaran información sobre el estado del MCC y sus elementos. Cada dispositivo cuenta bloques de registros con información relacionada al estado y función vinculada. Para el caso del dispositivo EMP2200, que monitorea el estado de la red de potencia que energiza el MCC, se extraen los registros de interés como se muestra en la TABLA VI [19]. El bloque de deferencia de estos registros es la de Registro de Retención (*Holding registers*), utilizando el código funcional 0x03 de Modbus.

TABLA VI. MAPA DE REGISTROS DE EMP2200

<i>Dirección</i>	<i>Descripción</i>	<i>Tipo de dato</i>
41000-41001	Voltaje de fase A-N	Float
41002-41003	Voltaje de fase B-N	Float
41004-41005	Voltaje de fase C-N	Float
41006-41007	Voltaje de línea A-B	Float
41008-41009	Voltaje de línea B-C	Float
41010-41011	Voltaje de línea C-A	Float
41012-41013	Corriente de fase A	Float
41014-41015	Corriente de fase B	Float
41016-41017	Corriente de fase C	Float
41018-41019	Potencia Activa Trifásica Total	Float
41020-41021	Potencia Reactiva Trifásica Total	Float
41022-41023	Potencia Aparente Trifásica Total	Float
41024-41025	Factor de Potencia Total	Float
41026-41027	Frecuencia	Float
41028-41029	Corriente en Neutro	Float
42000-42001	Corriente Promedio en fase A	Float
42002-42003	Corriente Promedio en fase B	Float
42004-42005	Corriente Promedio en fase C	Float
44100	Fase de Corriente en A	SInt16
44101	Fase de Corriente en B	SInt16
44102	Fase de Corriente en C	SInt16
44103	Ángulo de Voltaje A-B	SInt16
44104	Ángulo de Voltaje B-C	SInt16
44105	Ángulo de Voltaje C-A	SInt16

TABLA VII. MAPA DE REGISTROS DE MM200

<i>Dirección</i>	<i>Descripción</i>	<i>Formato</i>
Datos sobre último disparo		
30186	Causa de último disparo	FC134
30192-30193	Corriente A previa al disparo	FC10
30194-30195	Corriente B previa al disparo	FC10
30196-30197	Corriente C previa al disparo	FC10
30201	Carga de Motor Previo a disparo	F1
30202	Corriente de desbalance previo al disparo	F1
30203-30204	Corriente Tierra previa al disparo	F10
Contador de disparos		
30230	Número total de disparos	F1
30232	Disparos por sobrecarga	F1
30233	Disparo por atascamiento mecánico	F1
30234	Disparo por Corriente Baja	F1
30235	Corriente de disparo por desbalance	F1
30236	Disparo por falla a tierra	F1
30237	Disparo por aceleración de motor	F1
Tiempos generales		
30257	Número de arranques del motor	F1
30259-30260	Horas de funcionamiento del motor	F9
Estado del motor		
30305	Estado del motor	FC129
30306	Estado extendido de motor	FC178
30307	Capa térmica utilizada	F1
30308-30309	Tiempo para disparo por sobrecarga	F20
30310	Estado de la unidad	FC143
30312	Estado de comando	FC128
Medición de corriente		
30328-30329	Corriente Fase A	F10
30330-30331	Corriente Fase B	F10
30332-30333	Corriente Fase C	F10
30334-30335	Corriente promedio	F10
30336	Carga del motor	F1
30337	Corriente desbalance	F1
30338-30339	Corriente de tierra	F10
Datos alcanzados de arranque de motor		
30467	Tiempo de aceleración alcanzada	F2
30468-30469	Corriente de arranque alcanzada	F10
30470	Capacidad de arranque alcanzada	F1

TABLA VIII. FORMATO PARA EL MAPA DE REGISTROS DEL MM200

<i>Formato</i>	<i>Descripción</i>	<i>Tipo de dato</i>	
F1	Valor sin signo	Int16	
F2	Valor sin signo dividido para 10	Int16	
F9	Valor sin signo en big endian	Int32	
F10	Valor sin signo en big endian dividido para 10	Int32	
F20	Valor en complemento a 2 con signo	SInt32	
<i>Código de Formato</i>	<i>Valor</i>	<i>Descripción</i>	<i>Tipo de dato</i>
FC128, Estado de comando	0x00	Manual	Int16
	0x01	Automático	
	0x02	Manual inhibido	
	0x03	Automático/Manual	
	0x04	Auto cableado	
	0x05	Ninguno	
FC129, Estado rápido	0x01	Alarma	Int16
	0x02	Disparo	
	0x04	Auto prueba de falla	
	0x08	Automático	
	0x10	Contactador A	
	0x20	Contactador B	
	0x40	Contacto de salida 3	
	0x80	Unidad disponible	
FC134, Causa de último disparo	0x8042	Disparo térmico O/L	Int16
	0x8082	Disparo falla a tierra	
	0x80C2	Disparo por aceleración	
	0x8202	Disparo por atasco mecánico	
	0x8242	Disparo baja corriente	
	0x8282	Disparo desequilibrio	
	0x8442	Disparo falla de comunicación	
	0x8482	Relé no configurado	
	0x8742	Disparo por conexión alámbrica	
	0x8782	Disparo de campo	
FC143, Estado de la unidad	0x87C2	Disparo MCC	Int16
	0x8842	Parada de emergencia	
	0x88C2	Disparo control abierto	
	0x8902	Disparo de termistor	
	0x89C2	Disparo de auto revisión	
	0x00	Arranque no disponible	
FC178, Estado del Motor	0x01	Automático disponible	Int16
	0x02	Manual disponible	
	0x03	Disponible	
	0x04	Funcionando	
	0x0001	Bloqueado	Int16
	0x0002	Disparo no bloqueado	
	0x0010	Funcionando	
	0x0020	Pre-contactador	
	0x0080	Arrancando	
	0x0100	Inhibido	
	0x0200	Detenido	
	0x0400	Falla de auto-prueba	
	0x0800	Alarma	
	0x1000	Avance directo	
0x2000	Reversa		
0x4000	Velocidad baja		
0x8000	Velocidad alta		

De igual forma, el dispositivo MM200 dispone de bloques de registros; los de interés para este sensor se clasifican en: datos de último disparo, contador de disparos, tiempos generales, estado del motor, mediciones de corrientes, datos de arranque de motor alcanzados, tal como se indica en la TABLA VII [12]. Cada una de estas clasificaciones tienen registros que serán seleccionados para esta aplicación. Cada uno de los registros pertenece a un tipo de formato de dato, descrito en la TABLA VIII [12].

Existen ciertos formatos especiales que no representan necesariamente un valor número, estos son los formatos con código que representan estados del medidor o estados de motor. El bloque de acceso de los registros del MM200 se encuentran como Registro de Entradas (*Input Registers*), accesibles mediante la función 0x04 de Modbus. En este conjunto de registros existen ciertas particularidades relacionadas con su formato. En algunos casos el dato es exacto al registro del bloque para mostrarlo en el HMI. En otros casos es necesario realizar una división al valor del registro, considerar un bit de signo o relacionarlo con un significado por su formato, para poder desplegarlo en el HMI. De igual forma existen casos en donde el valor representado es un tiempo, por tal motivo, es necesario la aplicación de un código Python dentro de Ignition que ayudará a traducir grandes periodos de tiempo en formatos digeribles.

D. Diseño de HMI con Ignition

La aplicación Designer, propia de Ignition, contiene toda la estructura de diseño que será visible en el HMI. Para desarrollar el proyecto de forma eficiente se debe conocer acciones de optimización que Ignition dispone para los procesos de diseño y desarrollo. Existen varias características de Ignition que permiten acelerar el desarrollo de una aplicación, pero depende del diseñador utilizarlas correctamente.

1) Tags en Ignition

Las etiquetas (*tags*) en Ignition sirven para representar valores numéricos o simbólicos que provienen desde un dispositivo OPC UA. Los tags también pueden ser valores de memoria fijados por el usuario o tomar otro valor según su tipo. Los tipos de tags en Ignition son: tag derivado, tag de expresión, tag de memoria, tag OPC UA, tag de petición a base de datos (*query*) y tag de referencia. Para este trabajo práctico se utilizan tags OPC UA para leer los datos de la red Modbus y tags de memoria para crear setpoints útiles para la configuración de alarmas. Una herramienta que facilita la creación de tags en Ignition es la etiqueta de tipo de dato (*Data Types*). Los Data Types permiten crear un conjunto de tags para usarlos como referencia (*template tag*) que incluyen parámetros variables. Un tag creado en base a un Data Type se conoce como instancia de Data Type (*Data Type Instance*). Es decir, si la aplicación lo amerita, se puede crear un conjunto de Data Types Instance idénticos que obtengan un valor distinto de lectura o escritura a partir de la asignación de valores a sus parámetros, por ejemplo, un parámetro que se refiera al número de esclavo específico de la red Modbus.

a) Tags OPC UA Modbus

Para este trabajo práctico, por ejemplo, se tiene veintinueve dispositivos MM200 de los cuales se leerán el mapa de registros idénticos para cada uno de ellos. Cada dispositivo MM200 está relacionado a un módulo en particular del MCC y por ende el parámetro de variación es el número de esclavo de la red Modbus. Según la documentación, para la comunicación con dispositivos Modbus en Ignition se debe indicar el nombre OPC UA configurado, el número de esclavo, el bloque de datos al que se debe acceder y el tipo de dato que se está leyendo, por ejemplo, como se indica a continuación:

```
[{DEVICE}]{ID}.DESIGNATOR0000
```

En donde, entre corchetes se indica el nombre de configuración del dispositivo OPC UA, seguido del número de esclavo y un punto. A continuación, se indica el acceso al bloque de datos llamado designador. Finalmente, se indica la dirección a la cual acceder dentro del bloque para leer o escribir un dato. Entre llaves se indica el parámetro variable del depósito OPC UA y el identificador del nodo o número de esclavo al que se refiere. Estos parámetros son modificables por el usuario según la necesidad, haciendo que la creación de tags sea una tarea sencilla y dinámica. En la TABLA IX se indica los designadores que Ignition utiliza para Modbus.

TABLA IX. DESIGNADOR PARA MODBUS EN IGNITION

Designador	Descripción
HR	Para registros de retención, 16 bits
IR	Para registro de entrada, 16 bits
C	Para valores de bobina
DI	Para valores entradas discretas
HRBCD	Registro de retención, transformación BCD, 16 bits
HRBCD_32	Registro retención, transformación BCD, 32 bits
IRBCD	Registro de entrada, transformación BCD, 16 bits
IRBCD_32	Registro de entrada, transformación BCD, 32 bits
HRF	Registro de retención, transformación a float, 32 bits
IRF	Registro de entrada, transformación a float, 32 bits
HRD	Registro de retención, transformación double, 32 bits
IRD	Registro de entrada, transformación double, 32 bits
HRUS	Registro de retención, transformación SInt16, 16 bits
IRUS	Registro de entrada, transformación SInt16, 16 bits
HRI	Registro de retención, transformación Int32, 32 bits
IRI	Registro de entrada, transformación Int32, 32 bits
HRUI	Registro de retención, transformación SInt32, 32 bits
IRUI	Registro de entrada, transformación SInt32, 32 bits
HRI_64	Registro de retención, transformación Int64, 64 bits
IRI_64	Registro de entrada, transformación Int64, 64 bits
HRUI_64	Registro de retención, transformación SInt64, 64 bits
IRUI_64	Registro de entrada, transformación SInt64, 64 bits
HRS	Registro de retención con transformación a string

En el Designer, se crean tags OPC UA para cada uno de los registros descritos en el apartado III-C.2), utilizando la referencia del designador para Modbus de Ignition. En cada tag se pueden crear alarmas por medio de la utilización de setpoints establecidos, utilizando de tag de memoria para crear referencias.

b) Tags de memoria

Las referencias de los tags de memoria son los setpoints que permiten establecer condiciones sobre si un valor es igual, mayor, menor o entre un rango tolerable. Los valores de los tags son asignados por el usuario y ayudan a establecer la acción de alarmas de los tags OPC UA. En esta aplicación se utilizan cinco tags de memoria en la mayoría de los tags, por cada tag OPC UA. Se tiene un tag de memoria de valor para valores por debajo del mínimo, un tag por mínimo deseado, un tag por máximo deseado, un tag por máximo superado y un tag con el valor ideal. En algunos casos se utilizan setpoints de referencia igual al valor del tag OPC UA, esto se debe a los valores de los registros que tienen un significado específico no numérico. s el caso de los tags con código de formato FC, como se observa en la TABLA VIII.

2) Ventanas y parámetros

La creación de ventanas ayuda a distribuir la información de forma ordenada en el HMI. Para esta

aplicación se crearon tres tipos de ventanas. Las primeras corresponden a las barras de estado permanentes en los extremos norte y sur en la interfaz de la aplicación. La barra norte proporciona información útil para el usuario, mientras que la barra sur, permite navegar por las múltiples secciones de la aplicación. La barra superior contiene la hora actual, el usuario que inició sesión, el notificador e histórico de alarmas, el cambio de idioma entre inglés y español, bloqueo de pantalla, cambio de usuario y salir de la aplicación. La barra inferior tiene botones que llevan a cinco secciones del HMI que son el segundo tipo de ventanas: la primera es la ventana de bienvenida, la segunda la ventana que muestra el plano del MCC en donde se podrá acceder a toda la información presionando sobre cada módulo, la tercera ventana es un diagrama unifilar en donde se observa una vista rápida de la información más relevante, la cuarta ventana muestra un diagrama de comunicación que permitirá, en caso de ruptura de la red Modbus, conocer que nodo fue desconectado. Finalmente, una ventana de usuarios que permitirá administrar las personas que utilizarán el HMI y sus permisos. El tercer tipo de ventanas son las emergentes, estas ventanas se encargan de mostrar menús y su contenido. Las ventanas emergentes son las mayormente utilizadas en este diseño, ya que ayuda al usuario a mantener la noción de organización y tener claro que información se está observando.

El diseño de todas estas ventanas se lo realizó basándose en los conceptos de High Performance HMI, utilizando el color gris de tono claro como color principal de fondo y color negro para la fuente. Además, la información está bien organizada en menús exclusivos para cada dispositivo. Debido al tamaño pequeño de la pantalla y a la cantidad de información a la que se puede acceder, fue necesario crear políticas de bordes y márgenes para las ventanas emergentes. Todas estas ventanas, tienen la capacidad de ser arrastradas por toda el área visual sin salirse de los márgenes, también de ser redimensionadas según requiera el usuario. Una configuración extensa por medio de código Jython y propiedades de las ventanas propias de Ignition, se logró efectos visuales sofisticados en esta aplicación HMI. Todas las características descritas se pueden observar brevemente en la Fig. 15 y Fig. 15, este vistazo ayuda a tener una idea general sobre el contenido de interfaz del HMI para el sistema de monitoreo.

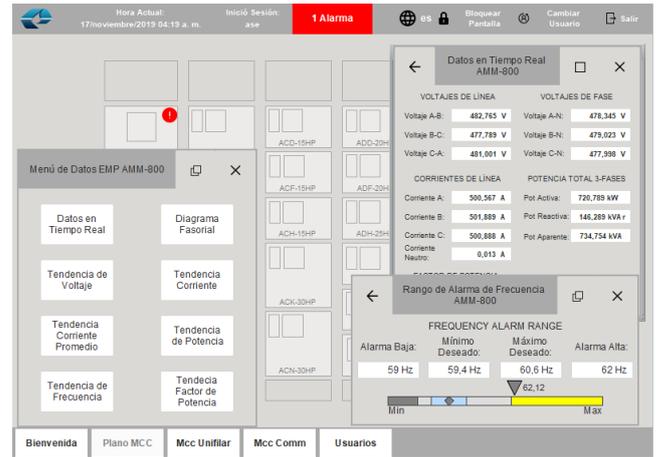


Fig. 14. Vistazo de interfaz HMI, datos en tiempo real; menú y rango de alarmas

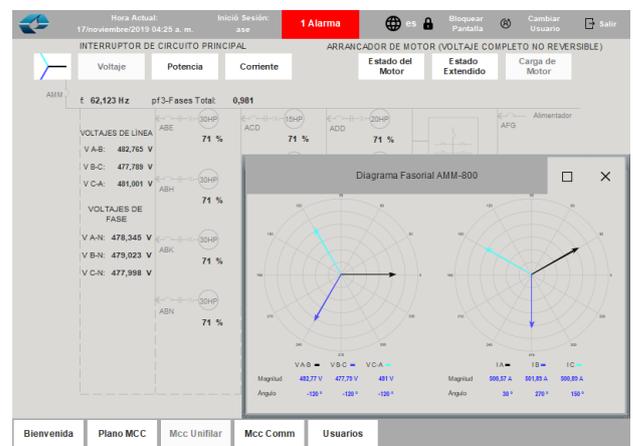


Fig. 15. Diagrama unifilar, diagrama fasorial y contenido general

Para manejar las asignaciones de cada menú al módulo correspondiente se utiliza la transición de parámetros. Pasar un parámetro de una ventana a otra tiene múltiples usos, en este caso, se pasa un parámetro tipo string con el fin de completar la dirección (*path*) de un tag. Con la transición de un parámetro diferente, se puede utilizar la misma ventana diseñada para mostrar valores provenientes de diferentes nodos del MCC. La transición de parámetros se completa mediante la ejecución de una acción, en esta aplicación se utiliza la acción de clic. En la Fig. 13 se muestra el proceso de transición de parámetros entre ventanas, su propósito y a que componentes están dirigidos. En esta descripción se utiliza un ejemplo en particular sobre el plano del MCC y de cómo cada módulo tiene su propio parámetro, y puede ser direccionado a la misma ventana.

3) Componentes inherentes de Ignition

Ignition cuenta con una extensa librería de componentes predefinidos que facilitan el diseño del HMI. En este caso, los indicadores análogos, los diagramas radiales, las gráficas de tendencias, la tabla de alarmas y la interfaz de administración de usuarios; son componentes predefinidos editables al gusto del diseñador. Sin embargo, mucho de los componentes necesarios para el desarrollo de este HMI se realizaron bajo diseño independiente. Componentes con la capacidad de desplegar datos como rótulos (*labels*) provenientes del dispositivo OPC UA, aportan legibilidad y entendimiento al proyecto. En varios de los componentes,

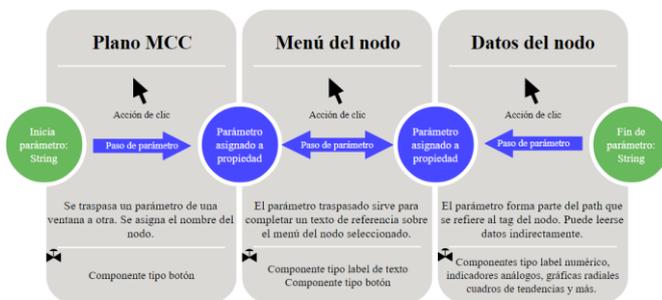


Fig. 13. Flujo de parámetros entre ventanas

como labels, se pueden escribir códigos de expresión propios de Ignition para administrar lo que se debe desplegar. Por ejemplo, para esta aplicación, fue necesario una referencia relacionada con el texto desplegado según el idioma seleccionado o con el texto desplegado según un valor numérico, como se indican en la TABLA VII. Un ejemplo de la utilización de los componentes de Ignition en la aplicación se muestra en la Fig. 16, el gráfico de tendencias ayuda a ver datos a lo largo del tiempo, el gráfico radial ayuda a entender si el valor leído está dentro de lo tolerables, los indicadores análogos ayudan en la entender los rangos de alarmas para un valor, por último, los labels números indican el valor leído directamente del tag o un setpoint configurado por el usuario.

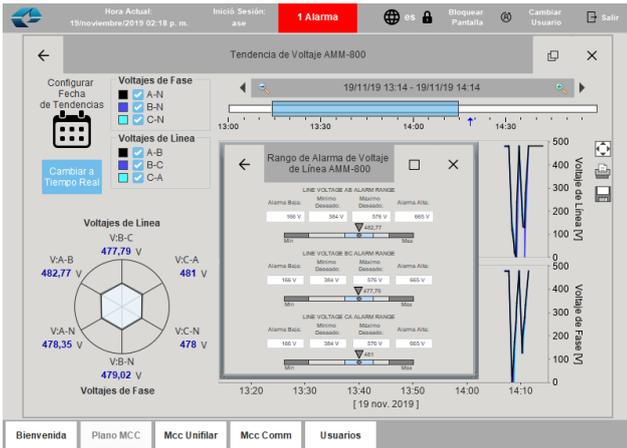


Fig. 16 Utilización de componentes de Ignition para tendencias, rangos tolerables y rangos de alarmas

También hay un componente para desplegar las alarmas, allí se despliegan las alarmas activas e inactivas no reconocidas. Es muy fácil revisar el histórico de alarmas, quien reconoció la alarma, hora de activación y desactivación, descripción y prioridad. Además de las alarmas por valores fuera de los setpoints, en el caso que uno de los nodos de la red Modbus falle, por ejemplo como se observa en la Fig. 17, se emitirá una alarma advirtiendo que nodo ha fallado, Fig. 18.

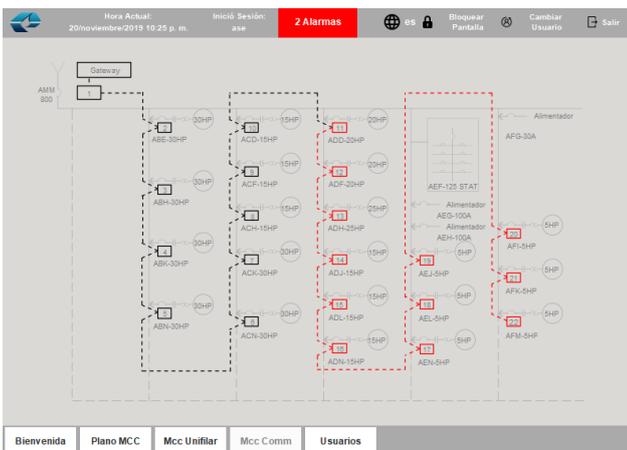


Fig. 17. Ruptura de la red Modbus RTU en nodo 11

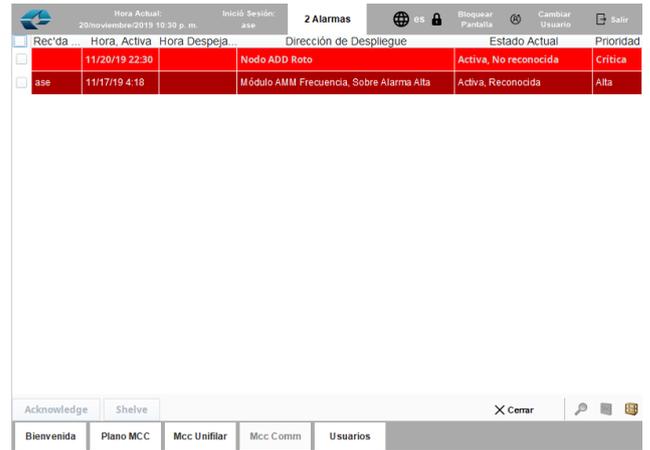


Fig. 18. Activación de alarma personalizada por ruptura de nodo

IV. EXPERIMENTACIÓN Y RESULTADOS

A. Lectura de tags desde simulador

Sin la correcta lectura de un valor asignado a un tag, no es posible desarrollar un HMI para un sistema de monitoreo. En mención al apartado III.D.1)a), el path de lectura de un dispositivo OPC UA configurado en el gateway para Modbus en esta aplicación se realiza asignando los siguientes parámetros:

```
[LIBMCC01]1.HRF1002
```

Según la TABLA VI, accedo al bloque de registros de retención en el dirección 1002, que corresponde al valor de voltaje de fase B-N para el dispositivo EMP2200 en el primer nodo. En el simulador *mod_RSsim* se escribe un valor en formato IEE 747 punto flotante que ocupa dos registros de 16 bits en la dirección especificada como se muestra en el la Fig. 19.

Address	+0	+1	+2
40841-40850	0000	0000	0000
40851-40860	0000	0000	0000
40861-40870	0000	0000	0000
40871-40880	0000	0000	0000
40881-40890	0000	0000	0000
40891-40900	0000	0000	0000
40901-40910	0000	0000	0000
40911-40920	0000	0000	0000
40921-40930	0000	0000	0000
40931-40940	0000	0000	0000
40941-40950	0000	0000	0000
40951-40960	0000	0000	0000
40961-40970	0000	0000	0000
40971-40980	0000	0000	0000
40981-40990	0000	0000	0000
40991-41000	0000	0000	0000
41001-41010	2C36	43EF	82F2
41011-41020	8021	43FA	4893
41021-41030	DC40	4933	6220
41031-41040	nnnn	nnnn	nnnn

Fig. 19. Bloque de registros de retención con valor en dirección 1002

El valor 0x43EF_82F2 corresponde a 479.023 en valor flotante. Este valor es el esperado en el tag asignado, para esta realización que se observa en la Fig. 20.

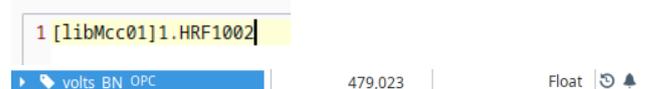


Fig. 20. Lectura de un valor desde la red Modbus a Ignition

Este procedimiento se lo realiza para cada medidor con 696 tags en total y se obtiene el valor de lectura desde el simulador.

B. Binding de propiedades de componentes

Binding se refiere a una unión, vínculo o referencia cruzada que muchos de los componentes disponen en Ignition. La mejor forma de desplegar un valor numérico es por el componente de campo numérico (*numeric text field*) que tiene propiedades de binding para mostrar su valor como se indica en Fig. 21.

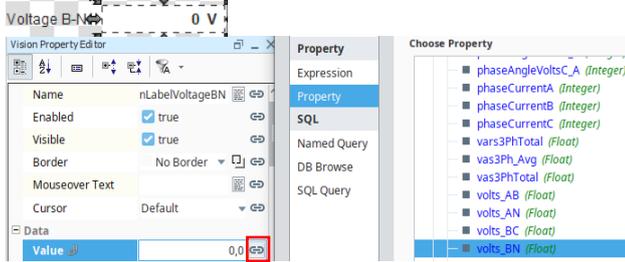


Fig. 21. Binding en propiedades para asignar un valor a un componente

De esta forma un componente adquiere el valor al que es referenciado, este procedimiento se repite para cada uno de los componentes de la aplicación que despliegan un valor numérico.

Para el caso de las gráficas de tendencias, el procedimiento es muy similar pero aplicado a las propiedades de este componente en particular. En las propiedades se muestra bindings para rótulos (*tag pens*), Fig. 22, que leerá el path del tag asignado y podrá desplegar sus valores históricos

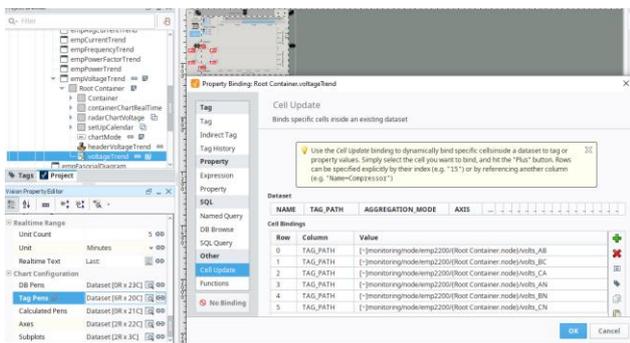


Fig. 22. Binding para propiedad tag pens para gráfica de tendencias

De esta forma y para cada componente es posible darle al HMI funcionalidad y dinamismo para resultados más eficientes.

C. Estructura y resultado final del HMI

La aplicación se ejecuta en una pantalla pequeña para la cantidad de información que proporciona, por esta razón se trabajó con menús y sub-menús para acceder a las distintas partes del contenido. La estructura para esta aplicación se basa en la colocación de dos barras ancladas en los bordes superior e inferior de la aplicación que estarán disponibles todo el tiempo. La barra norte se compone por el logo del cliente final, la hora actual, el nombre del usuario que ejecuto la aplicación, el contador y journal de alarmas, la selección del idioma entre inglés y español, bloqueo de pantalla, cambio de usuario y salir de la sesión de la aplicación. En la barra inferior se encuentran los botones de acceso global de la aplicación, un botón de bienvenida, el plano del MCC para el acceso a los menús de cada módulo,

un acceso al diagrama unifilar de potencia que brindará fácilmente la información resumida y necesaria de cada módulo, un botón para acceder al diagrama de comunicación que ayudará a identificar el estado de la red Modbus he indicará el punto de ruptura de la red si fuese el caso y botón para el acceso al administrador de usuarios que indicara los permisos y horas activadas de cada operador.

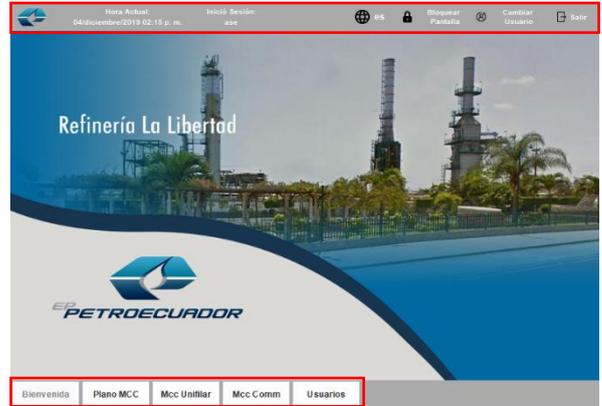


Fig. 23. Estructura de manejo de ventanas de la aplicación

El contenido de cada una de estos accesos se describe en las Fig. 24, Fig. 25, Fig. 26 y Fig. 27, que corresponde al plano del MCC, el diagrama de potencia, diagrama de comunicación y usuarios, respectivamente.

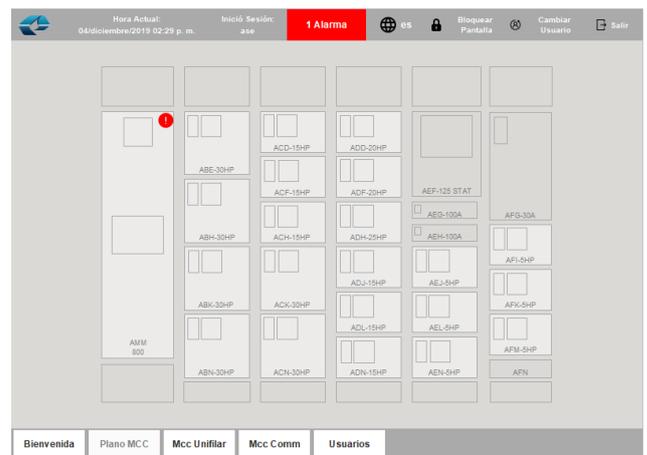


Fig. 24. Plano del MCC con cada módulo

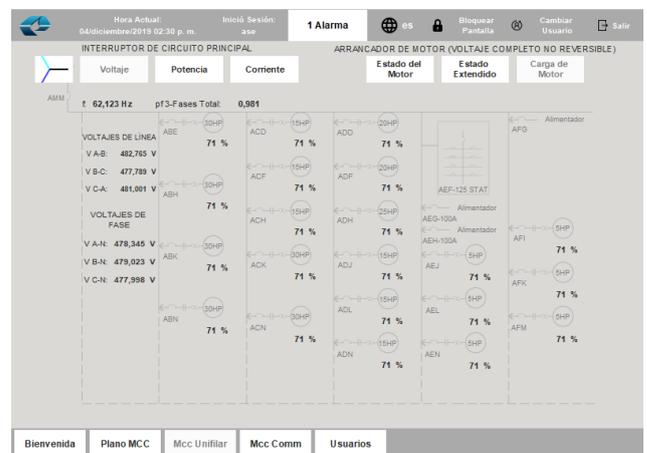


Fig. 25. Diagrama de potencia con información de acceso rápido

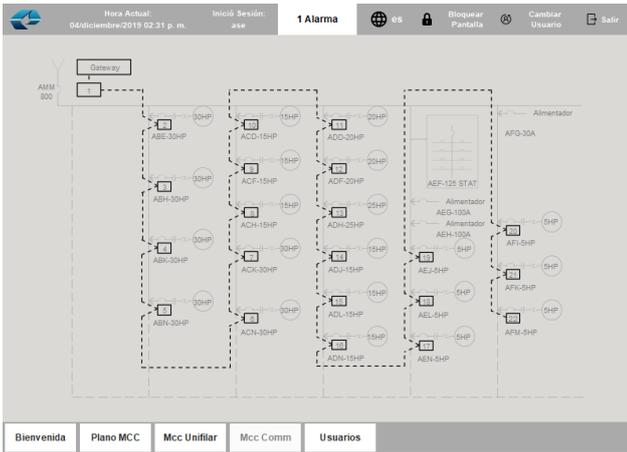


Fig. 26. Diagrama de estado de la red Modbus RTU sobre TCP/IP

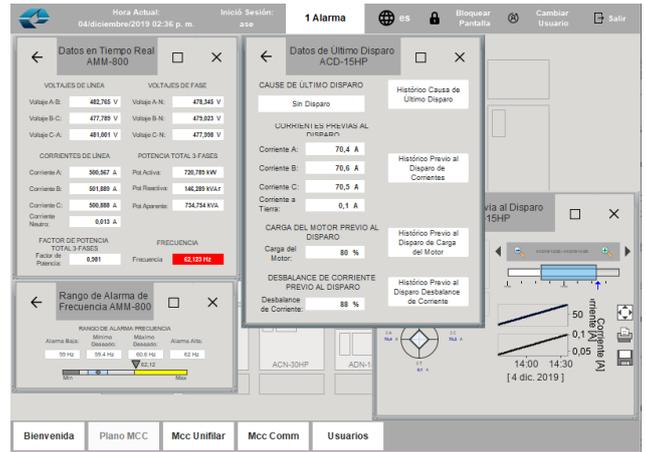


Fig. 29. Contenido de algunas ventanas de la aplicación, datos en tiempo real, set points de alarmas y tendencias

Nombre de Usuario	Nombre	Roles	Info. del Contacto	Horario
ase		Administrator, Op...		Always
operator		Operator		Always
supervisor		Supervisor		Always

Nombre de Rol	# de Miembros
Administrator	1
Operator	2
Supervisor	2

Fig. 27. Tabla de administración de usuarios para la aplicación

También se muestran los menús para cada módulo que corresponde al EMP2200 y MM200, se muestra en la Fig. 28.



Fig. 28. Menús para cada tipo de sensor que corresponde a un módulo del MCC

En la Fig. 29 se muestra algunas vistas de información, por cuestiones de reserva cooperativa de ASE, no se puede mostrar todas las funcionalidades y ventanas para fines académicos.

V. CONCLUSIONES

Existen varios elementos necesarios para alcanzar el objetivo de este proyecto. Inicialmente, se debe comprender el proceso industrial que se busca representar y el alcance del HMI relacionado. De igual forma, la comprensión de todas las funcionalidades de Ignition como software de desarrollo del proyecto; ayudan a alcanzar los objetivos deseados. También, consolidar que el protocolo de comunicación Modbus es el más idóneo para este trabajo práctico para la adquisición de datos. Incluso, la compatibilidad de los dispositivos medidores utilizados, justifica el uso de este protocolo. Por otro lado, una comprensión básica sobre bases de datos como en este caso MySQL enlazada a Ignition, es fundamental para cumplir los propósitos del proyecto. Finalmente, el desarrollo y diseño de la aplicación basadas en conceptos de High Performace HMI y usar las facilidades de Ignition para alcanzar los resultados esperados.

El diseño de un HMI, ligado a los conceptos de High Performace HMI, genera una perspectiva sofisticada y funcional en el sistema de monitoreo implementado. La idea tras un HMI de alto desempeño se basa en presentar información relevante, sin representaciones innecesarias, ayudando al usuario a entender el proceso y reconocer un evento con facilidad. El desafío del desarrollo este proyecto se basa en uso High Performace HMI para una gran cantidad de información en un espacio reducido en pantalla.

Ignition es una herramienta que facilita, en todo sentido, la ejecución del proyecto. Las capacidades de diseño y programación no son suficientes para obtener, desde el punto de vista de la ingeniería, una HMI profesional. El estudio de las tecnologías involucradas permitió un entendimiento razonable en cuanto a los elementos de un HMI industrial. A pesar de que no se realizan acciones de control en este HMI, el monitoreo permite al operador conocer todo lo que sucede en el MCC desde una sola pantalla.

El HMI desarrollado cumple con los requerimientos solicitados por el sector empresarial. Además, como ya se mencionó, es un proyecto que dará a conocer a la empresa contratista las capacidades de Ignition ofrece.

VI. REFERENCIAS

- [1] M. D. Whitt, *Successful Instrumentation and Control Systems Design*, 1ª. ed. EEUU: ISA, 2004, pp 104-106.
- [2] B. Hollifield, "A High Performance HMI: Better Graphics for Operations Effectiveness," en ISA Water & Wastewater and Automatic Controls Symposium, FL EEUU, Agosto 2012, pp. 3-7.
- [3] A. Penin, *Sistemas SCADA*, 1ª. ed. Barcelona: MARCOMBO, 2012, pp. 5.57-5.60.
- [4] Modbus Organization. (2019), Miembros: Moxa [En línea]. disponible: <http://www.modbus.org/about.php#315>.
- [5] Inductive Automation. (2019). Introducing Ignition [En línea]. Disponible: <https://docs.inductiveautomation.com/display/DOC80/Introducing+Ignition>
- [6] Automation Solution Ecuador. (2019). Provisión de equipos de sistemas de control [En línea]. Disponible: <http://www.asecuador.com/servicios/>
- [7] A. Mendoza *Montaje de redes eléctricas aéreas de baja tensión. ELEEO109*, 1ª. ed. Málaga: iceditorial, 2012.
- [8] J. F. Kurose, K. W. Ross, *Redes de computadores un enfoque descendente*, 5ª. ed. Madrid: pp, 50-53.
- [9] Catálogo No. CTI Part # 062-00187-024, 2573-MOD and 2573-TCM2 serial device interface adapter installation and operation guide Version 2.4, Control Technology Inc. Knoxville, TN.
- [10] MathWorks. Baud Rate [En línea]. Disponible: https://es.mathworks.com/help/matlab/matlab_external/baudrate.html
- [11] Catálogo No. 1.0b, Modbus messaging on TCP/IP implementation guide, Modbus Organization, Modbus –IDA, Octubre 2006.
- [12] Catálogo No. GEK-113402C, MM200 revision: 1.2x, General Electric, Markham, ON.
- [13] R. Bacs. (2015, Junio 14). SOT: CRC8 vs. CRC16 [En línea]. Disponible: <https://www.embeddedrelated.com/showthread/msp430/18180-1.php>
- [14] J. Bartolomé. (2011, Enero). El protocolo MODBUS [En línea]. Disponible: <http://www.tolaemon.com/docs/modbus.htm>
- [15] Simply Modbus. (2019). Modbus TCP/IP [En línea]. Disponible: <http://www.simplymodbus.ca/TCP.htm>
- [16] MySQL. (2019). MySQL Document Store. [En línea]. Disponible: https://www.mysql.com/products/enterprise/document_store.html
- [17] G. Chao, "Human-Machine Interface: Design Principles of Visual Information in Human-Machine Interface Design" en 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics, PEK China, 2009, pp. 263-265.
- [18] Advantech Co., Ltd. (2019). PPC-3100S [En línea]. Disponible: https://www.advantech.com/products/1-2jkjm3/ppc-3100s/mod_1a0afa16-fab0-4642-aec4-504d02832d28
- [19] Catálogo No. GEK-113575C, Multilin™ EPM 2200 Power Meter, General Electric revision: 1.0x, Markham, ON.
- [20] Inductive University. (2019). Learn Ignition at Your Own Pace [En línea]. Disponible: <https://www.inductiveuniversity.com/>