

**IMPLEMENTACIÓN DE MODBUS SOBRE TCP/IP UTILIZANDO LABVIEW**

**MARLON JOSÉ MATOSA PÉREZ  
RICARDO ANTONIO MARTÍNEZ BAYUELO**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR  
FACULTAD DE INGENIERÍAS  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
CARTAGENA  
2008**

**IMPLEMENTACIÓN DE MODBUS SOBRE TCP/IP UTILIZANDO LABVIEW**

**MARLON JOSÉ MATOSA PÉREZ  
RICARDO ANTONIO MARTÍNEZ BAYUELO**

**Monografía para optar al título de Ingeniero Electrónico**

**Director:  
JORGE DUQUE  
Magíster en Ingeniería Electrónica**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR  
FACULTAD DE INGENIERÍAS  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
CARTAGENA  
2008**

**Nota de aceptación**

---

---

---

---

---

**Jurado**

---

**Jurado**

---

Cartagena D. T. y C, Junio de 2008

Señores

**COMITÉ CURRICULAR**  
**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**  
La ciudad

Respetados señores:

Con toda atención me dirijo a ustedes con el fin de presentarles a su consideración, estudio y aprobación la monografía titulada **IMPLEMENTACIÓN DE MODBUS SOBRE TCP/IP UTILIZANDO LABVIEW** como requisito para obtener el título de Ingeniero Electrónico.

Atentamente,

---

MARLON JOSÉ MATOSA PÉREZ  
CC 73.201.660 de Cartagena

---

RICARDO MARTÍNEZ BAYUELO  
CC 73.208.562 de Cartagena

Cartagena D. T. y C, Junio de 2008

Señores

**COMITÉ CURRICULAR**  
**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**  
La ciudad

Cordial saludo:

A través de la presente me permito ratificar la asesoría prestada para la monografía titulada **IMPLEMENTACIÓN DE MODBUS SOBRE TCP/IP UTILIZANDO LABVIEW** la cual fue realizada por los estudiantes MARLON JOSÉ MATOSA PÉREZ Y RICARDO ANTONIO MARTÍNEZ BAYUELO, como requisito para la aprobación del “Minor de Automatización Industrial”, y optar el título de Ingeniero Electrónico.

Atentamente,

---

ING. JORGE DUQUE  
Magíster en Ing. Electrónica  
Director

## AUTORIZACIÓN

Por medio de la presente, autorizamos la utilización en las bibliotecas de la Universidad Tecnológica de Bolívar, y la publicación en el catalogo on-line con fines exclusivamente académicos la monografía **IMPLEMENTACIÓN DE MODBUS SOBRE TCP/IP UTILIZANDO LABVIEW**, realizadas por Marlon José Matosa Pérez y Ricardo Antonio Martínez Bayuelo como requisito para obtener el título de Ingeniero Electrónico.

Atentamente,

---

MARLON JOSÉ MATOSA PÉREZ  
CC 73.201.660 de Cartagena

---

RICARDO MARTÍNEZ BAYUELO  
CC 73.208.562 de Cartagena

## TABLA DE CONTENIDO

<b>CONTENIDO .....</b>	<b>1</b>
<b>LISTA DE TABLAS .....</b>	<b>4</b>
<b>LISTA DE FIGURAS .....</b>	<b>5</b>
<b>INTRODUCCIÓN.....</b>	<b>7</b>
<b>CAPITULO I – PROTOCOLO MODBUS.....</b>	<b>9</b>
<b>1.1 DESCRIPCIÓN.....</b>	<b>9</b>
1.1.1 Modelo Cliente/Servidor.....	11
<b>1.2 PROTOCOLO MODBUS TCP/IP.....</b>	<b>13</b>
1.2.1 Construcción de un paquete de datos Modbus TCP.....	14
<b>CAPITULO II – IMPLEMENTACIÓN MODBUS TCP/IP.....</b>	<b>17</b>
<b>2.1 DESCRIPCIÓN DEL PROTOCOLO.....</b>	<b>17</b>
2.1.1 Arquitectura General de Comunicación.....	18
2.1.2 Unidad de aplicación de datos (ADU) en Modbus sobre TCP/IP.....	20
2.1.3 Descripción de Cabecera MBAP.....	22
<b>2.2 DESCRIPCIÓN DE LOS CÓDIGOS DE FUNCIÓN EN MODBUS.....</b>	<b>23</b>
2.2.1 Leer el estado de la bobina.....	26
2.2.2 Leer los registros de espera.....	26
2.2.3 Leer registros de entrada.....	26
2.2.4 Forzar una bobina simple.....	26
2.2.5 Preselección de registro único.....	27
2.2.6 Forzar múltiples registros.....	27

2.2.7	Preseleccionar registros múltiples.....	27
2.2.8	Reportar la dirección del esclavo.....	28
<b>CAPITULO III – DESCRIPCIÓN FUNCIONAL.....</b>		<b>28</b>
<b>3.1</b>	<b>MODELO DE LOS COMPONENTES DE LA ARQUITECTURA DE MODBUS.....</b>	<b>29</b>
<b>3.2</b>	<b>GESTIÓN DE CONEXIÓN TCP.....</b>	<b>34</b>
3.2.1	Módulo de gestión de las conexiones.....	34
3.2.1.1	Descripción General.....	34
3.2.1.2	Descripción de gestión de conexión.....	39
3.2.2	Impacto de los modos de funcionamiento de una conexión TCP.....	41
3.2.2.1	Ruptura de conexión entre puntos finales de operación.....	42
3.2.2.2	Caída y reinicio el punto final del servidor.....	42
3.2.2.3	Caída y reinicio del Cliente.....	43
3.2.3	Modulo de control de acceso.....	45
<b>3.3</b>	<b>EL USO DE LA PILA TCP/IP.....</b>	<b>46</b>
3.3.1	El uso de la interfaz BSD Socket.....	47
3.3.2	Parámetros de la capa IP.....	51
3.3.2.1	Parámetros IP.....	51
<b>3.4</b>	<b>CAPA DE APLICACIÓN DE COMUNICACIÓN.....</b>	<b>52</b>
3.4.1	Cliente Modbus.....	52
3.4.1.1	Diseño del Cliente Modbus.....	53
3.4.1.2	Construir una Solicitud MODBUS.....	54
3.4.1.3	Confirmación de un proceso Modbus.....	60
3.4.1.4	Gestión del tiempo de espera.....	64



3.4.2	Servidor Modbus.....	<b>65</b>
3.4.2.1	Diseño del servidor Modbus.....	<b>67</b>
3.4.2.2	Comprobación de Modbus PDU.....	<b>71</b>
3.4.2.3	Procesamiento del servicio Modbus.....	<b>74</b>
3.4.2.4	Interfaz de Aplicación de usuario (Backend Interface).....	<b>75</b>
3.4.2.5	Construcción de la respuesta Modbus.....	<b>77</b>
<b>CONCLUSIONES.....</b>		<b>80</b>
<b>BIBLIOGRAFÍA.....</b>		<b>81</b>
<b>ANEXO.....</b>		<b>82</b>

## LISTA DE TABLAS

<b>Tabla 1.</b> Descripción de la cabecera MBAP.....	<b>22</b>
<b>Tabla 2.</b> Organización de los registros de Modbus.....	<b>25</b>
<b>Tabla 3.</b> Funciones estándar de Modbus.....	<b>25</b>
<b>Tabla 4.</b> Componentes de la interfaz de Modbus.....	<b>30</b>
<b>Tabla 5.</b> Solicitud Modbus ADU.....	<b>57</b>
<b>Tabla 6.</b> Códigos de excepción.....	<b>78</b>

## LISTA DE FIGURAS

<b>Figura 1.</b> Modelo cliente/servidor.....	<b>12</b>
<b>Figura 2.</b> Construcción de un paquete de datos Modbus TCP.....	<b>14</b>
<b>Figura 3.</b> Arquitectura de comunicación Modbus TCP/IP.....	<b>19</b>
<b>Figura 4.</b> Trama general de Modbus.....	<b>19</b>
<b>Figura 5.</b> Solicitud/Respuesta de Modbus sobre TCP/IP.....	<b>20</b>
<b>Figura 6.</b> Arquitectura del servicio de Mensajería de Modbus.....	<b>29</b>
<b>Figura 7.</b> Modelo de datos de Modbus con bloques separados.....	<b>30</b>
<b>Figura 8.</b> Modelo de datos de Modbus con un solo bloque.....	<b>30</b>
<b>Figura 9.</b> Diagrama de actividades de la gestión de conexión TCP.....	<b>36</b>
<b>Figura 10.</b> Estableciendo la conexión Modbus TCP.....	<b>40</b>
<b>Figura 11.</b> Interfaz de la pila TCP/IP.....	<b>47</b>
<b>Figura 12.</b> Intercambios de Modbus.....	<b>50</b>
<b>Figura 13.</b> Unidad cliente de Modbus.....	<b>52</b>
<b>Figura 14.</b> Diagrama de actividad del cliente Modbus.....	<b>53</b>

<b>Figura 15.</b> Diagrama de actividad de la construcción de la solicitud.....	<b>56</b>
<b>Figura 16.</b> Diagrama de actividad del confirmación del proceso Modbus.....	<b>63</b>
<b>Figura 17.</b> Unidad del servidor de Modbus.....	<b>65</b>
<b>Figura 18.</b> Diagrama de actividad de la indicación del proceso Modbus.....	<b>68</b>
<b>Figura 19.</b> Diagrama de actividad del comprobación de Modbus PDU.....	<b>71</b>
<b>Figura 20.</b> Diagrama de actividad del proceso de servicio Modbus.....	<b>74</b>

## INTRODUCCIÓN

En este documento investigativo se presentara el servicio de mensajería Modbus a través de TCP/IP, con el fin de proporcionar información de referencia que ayude a los desarrolladores de software para implementar este servicio. También se da una precisa y completa descripción de la aplicación de dicho servicio y su objetivo es facilitar la interoperabilidad entre los dispositivos que utilizan este servicio de mensajería.

Modbus TCP/IP es un protocolo de Internet. El hecho de que TCP/IP es el protocolo de transporte de Internet significa que automáticamente Modbus TCP/IP se puede utilizar a través de Internet. Entre otras cosas, fue diseñado para alcanzar este objetivo, y como parte de este objetivo la especificación de este protocolo se ha presentado a la **Internet Engineering Task Force (IETF)**.

En términos prácticos, esto significa que un dispositivo Modbus TCP/IP instalado en Europa puede ser abordado a través de Internet desde cualquier otra parte del mundo.

Modbus TCP/IP se ha convertido en una industria estándar debido a su apertura, simplicidad, bajo costo de desarrollo, y mínimos de hardware necesarios para apoyarlo.

En este momento hay más de 200 disponibles en el mercado. Se utiliza para el intercambio de información entre dispositivos, para supervisar y programar. También se utiliza para gestionar entradas y salidas distribuidas, siendo el protocolo preferido por los fabricantes de este tipo de dispositivos.

El contenido del presente libro está estructurado en capítulos, dispuestos de tal forma, que el lector pueda comprender los planteamientos que se exponen y el proceso de desarrollo del sistema; todo esto de una manera gradual y siguiendo las fases cursadas por los desarrolladores durante la elaboración de la aplicación final:

El **Capítulo I**, contiene una descripción de los protocolos Modbus y Modbus TCP/IP, se explica todas sus funciones y los modelos en los cuales se basan para dicho funcionamiento.

En el **Capítulo II**, se presenta la manera de implementar los protocolos anteriormente mencionados, además, se pretende introducir al lector en la teoría y arquitecturas de los diferentes componentes, explicando de una manera resumida los aspectos de mayor importancia.

En el **Capítulo III**, se detalla la descripción funcional y modelos de arquitectura, como también las especificaciones de la gestión de las comunicaciones y además se explica de forma detallada el funcionamiento y cada una de las partes que intervienen en una comunicación como lo son el cliente y el servidor.

## **CAPITULO I**

### **PROTOCOLO MODBUS**

#### **DESCRIPCIÓN**

El protocolo Modbus fue desarrollado en 1979 por Modicon Incorporated, para los sistemas de automatización industrial y para los controladores programables Modicon. Desde entonces se ha convertido en un método estándar de las industrias para la transferencia de I/O discretas y analógicas de información y registro de datos entre el control y los dispositivos de control industrial. Modbus es ahora de amplia aceptación, abierto, de dominio público que se requiere una licencia, pero no requiere el pago de regalías a su propietario.

Los dispositivos Modbus se comunican a través de un maestro – esclavo (cliente/servidor), técnica en la que sólo un dispositivo (maestro / cliente) puede iniciar transacciones (consultas). Los otros dispositivos (esclavos o servidores) responden mediante el suministro de los datos solicitados al maestro, o por la adopción de las medidas solicitadas en la consulta.

Un esclavo es cualquier dispositivo periférico (I / O transductor, válvula, unidad de red, u otro dispositivo de medición), que procesa información y envía la salida al maestro usando Modbus.

Los maestros pueden direccionar a los esclavos individualmente, o pueden iniciar un mensaje de difusión a todos los esclavos. Los esclavos devuelven una respuesta a todas las preguntas dirigidas a ellos Individualmente, pero no responden a las preguntas de difusión. Los esclavos no inician mensajes por su cuenta sólo han de responder a las preguntas del maestro.

La pregunta de un maestro consistirá de la dirección de esclavo (o dirección de difusión), un código de función definiendo la acción solicitada, los datos que sean necesarios, y el campo de comprobación de error.

La respuesta del esclavo consiste en la confirmación de los campos de la acción tomada, cualquier dato a ser devuelto, y campo de comprobación de error. Tenga en cuenta que la pregunta y la respuesta incluyen la dirección de un dispositivo, el código de una función, más datos aplicables, y campo de comprobación de error.

Si no se produce el error, la respuesta del esclavo contiene los datos que se le han solicitado. Si se produce un error en la pregunta recibida, o si el esclavo no está en condiciones de llevar a cabo la acción solicitada, el esclavo mostrará un mensaje de excepción a su respuesta.



El campo de comprobación de error del mensaje del esclavo permite al maestro confirmar que el contenido de los mensajes es válido. Los mensajes tradicionales de Modbus son transmitidos vía serial y la comprobación de paridad se aplica también a cada carácter transmitido en su trama de datos.

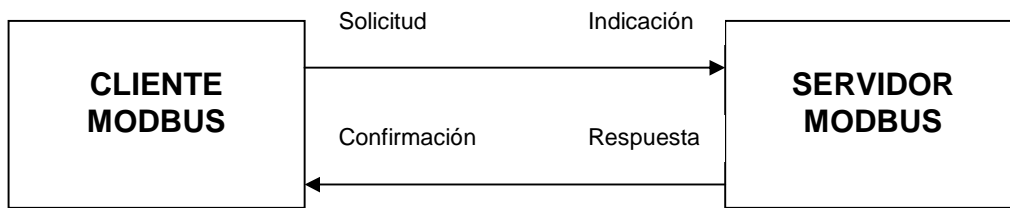
En esta aplicación, se definen las normas para la organización e interpretación de datos, pero sigue siendo simplemente una estructura de mensajería independientemente de la capa física.

Es fácil de comprender y de libre acceso, y accesible a cualquier persona, por lo tanto, es ampliamente apoyado por muchos fabricantes, es importante hacer la distinción de que el mismo Modbus es un protocolo.

### **1.1.1 Modelo cliente/servidor**

El servicio de mensajería de MODBUS provee una comunicación cliente/servidor entre dispositivos conectados en una red Ethernet TCP/IP. El modelo cliente/servidor esta basado en cuatro tipos de mensajes:

- Solicitud Modbus
- Confirmación Modbus
- Indicación Modbus
- Respuesta Modbus



**Figura 1.** Modelo cliente/servidor

Una solicitud Modbus es el mensaje enviado a la red por el cliente para iniciar una transacción.

Una indicación Modbus es el mensaje de solicitud recibido en el lado del servidor.

Una respuesta Modbus es el mensaje de respuesta enviado por el servidor.

Una confirmación Modbus es el mensaje de respuesta recibido en el lado del cliente.

Los servicios de mensajería del modelo cliente/servidor son usados para intercambiar información en tiempo real, entre las cuales se encuentran:

- Intercambio entre dos aplicaciones de dispositivo.
- Entre una aplicación de dispositivo y otro dispositivo.
- Entre aplicaciones HMI/SCADA y otros dispositivos.
- Entre un PC y un programa de dispositivo suministrado en líneas de servicio.

## 1.2 PROTOCOLO MODBUS TCP/IP

Modbus TCP/IP (también Modbus TCP) es simplemente el protocolo Modbus RTU con una interfaz TCP que se ejecuta en Ethernet.

La estructura de los mensajes de Modbus definen las reglas para la organización y la interpretación de los datos independientemente de los datos del medio de transmisión.

TCP/IP se refiere al protocolo de control de transmisión y el protocolo de Internet, que proporciona el medio de transmisión para los mensajes de Modbus TCP/IP.

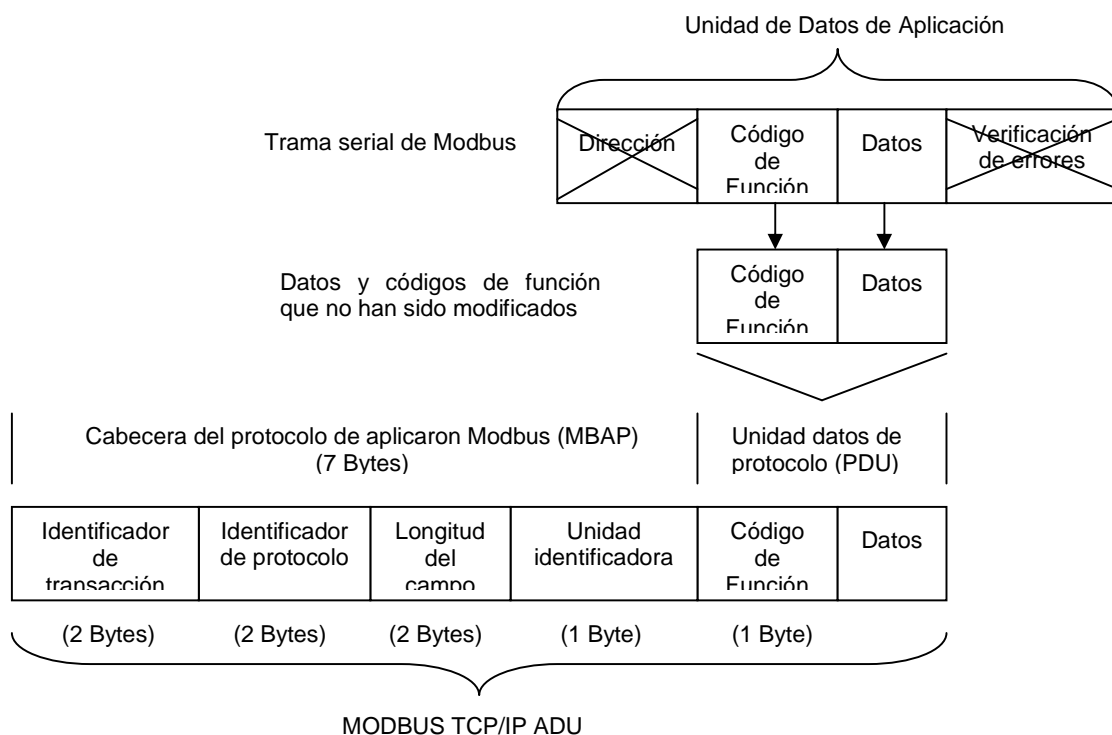
En pocas palabras, TCP/IP permite bloques de datos binarios que se van a intercambiar entre computadoras. También es una norma mundial que sirve de base para la World Wide Web. La función principal de TCP es asegurar que todos los paquetes de datos se reciben correctamente, mientras IP se asegura que los mensajes están correctamente dirigidos y enrutados. Tenga en cuenta que la combinación TCP/IP no es más que un protocolo de transporte, y no define el significado de los datos o como son interpretados (ese es el trabajo de Modbus en este caso).

Así que en resumen, Modbus TCP/IP utiliza TCP/IP y Ethernet para transportar los datos de la estructura de los mensajes de Modbus entre dispositivos compatibles.

Es decir, Modbus TCP/IP combina una red física (Ethernet) con una red de trabajo estándar (TCP/IP), y un método estándar de representación de datos (Modbus como protocolo de aplicación). Esencialmente, el mensaje de Modbus TCP/IP es simplemente una comunicación Modbus encapsulado en una conexión Ethernet TCP/IP.

### 1.2.1 Construcción de un paquete de datos modbus TCP

Modbus TCP incluye una trama de datos Modbus estándar dentro de una trama TCP, sin la comprobación de Modbus, como se muestra en la siguiente figura.



**Figura 2.** Construcción de un paquete de datos Modbus TCP

Los comandos de Modbus y los datos de usuario son encapsulados en un paquete de datos de una red TCP/IP sin ser modificados de ninguna manera.

Sin embargo, el campo de comprobación de errores de Modbus no se utiliza como la capa de enlace estándar de Ethernet TCP/IP por los métodos utilizados para garantía de la integridad de los datos.

Además, el campo de “la dirección de Modbus” es suplantado por la “Unidad identificadora” en Modbus TCP/IP, y se convierte en parte del protocolo de Aplicación de cabecera Modbus (MBAP).

De la figura 2, vemos que el código de la función y los datos de los campos son absorbidos en su forma original. Así, una unidad de datos de aplicación Modbus TCP/IP (ADU) adopta la forma de una cabecera de 7 bytes (Identificador de transacción + identificador de protocolo + longitud del campo + unidad identificadora) y la unidad de protocolo de datos (código de la función + datos).

La cabecera MBAP es de 7 bytes de longitud, e incluye los siguientes campos:

- **Transacción / Identificador (2 Bytes):** Este campo de identificación se utiliza para la operación de parejas cuando varios mensajes son enviados a través de la misma conexión TCP por un cliente sin esperar a una previa respuesta.

- **Protocolo de identificación (2 bytes):** Este campo es siempre 0 para los servicios Modbus y otros valores están reservados para futuras ampliaciones.
- **Longitud (2 bytes):** Este campo es un contador de bytes de los campos restantes e incluye la unidad de identificación de bytes, la función de código byte, y los campos de los datos.
- **Unidad Identificadora (1 byte):** Este campo se utiliza para identificar un servidor remoto situado en una red que no es TCP/IP (para un puente serial). En un típico servidor de aplicación de Modbus TCP/IP, la unidad de identificación está establecido en 00 o FF, ignorada por el servidor, y simplemente se hizo eco de nuevo en la respuesta.

La unidad de datos de Aplicación de Modbus TCP/IP está incrustada en el campo de datos de una trama estándar TCP y enviada vía TCP por un conocido sistema Puerto 502, que se reservan específicamente para aplicaciones Modbus. Los clientes Modbus TCP/IP y los servidores escuchan y reciben datos de Modbus vía Puerto 502.

Podemos ver que el funcionamiento de Modbus a través de Ethernet es casi transparente para el registro Modbus y la estructura de comando. Por tanto, si está ya están familiarizados con el funcionamiento del MOdbus tradicional, entonces usted ya conoce el funcionamiento de Modbus TCP / IP.

## **CAPITULO II**

### **IMPLEMENTACIÓN MODBUS TCP/IP**

#### **2.1 DESCRIPCIÓN DEL PROTOCOLO**

Con el fin de implementar el servicio de mensajería de MODBUS a través de TCP/IP, se proporciona información de referencia que ayuda a los desarrolladores de software a implementar este servicio.

Este documento da una precisa y completa descripción de la aplicación del servicio de mensajería MODBUS. Su objetivo es facilitar la interoperabilidad entre los dispositivos que utilizan el servicio de mensajería MODBUS.

Para la implementación del protocolo Modbus TCP/IP se tienen en cuenta tres aspectos:

1. Una visión general del protocolo MODBUS sobre TCP/IP.
2. Una descripción funcional de un cliente MODBUS, servidor y la implementación de la puerta de enlace.
3. Una guía de aplicación que propone la implementación de Modbus sobre TCP/IP basada en Labview.

### **2.1.1 ARQUITECTURA GENERAL DE COMUNICACIÓN**

Un sistema de comunicación de MODBUS TCP/IP puede incluir diferentes tipos de dispositivos:

- Un Cliente MODBUS TCP/IP y dispositivos del servidor conectados a una red TCP/IP.
- La Interconexión de dispositivos como puentes, enrutadores o puertas de enlace para la interconexión entre la red TCP/IP y una línea serial de sub-red las cuales permiten las conexiones de la línea serial del cliente MODBUS y los dispositivos finales del servidor.



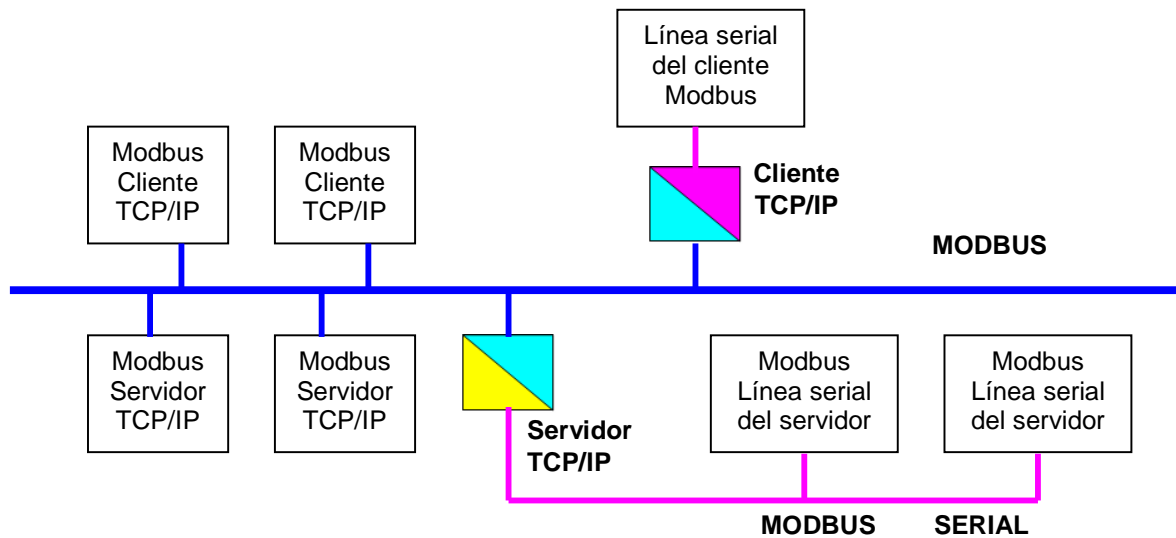


Figura 3. Arquitectura de comunicación Modbus TCP/IP

El protocolo MODBUS define una Unidad de Protocolo de Datos (PDU) independientemente de las capas de comunicación. La cartografía del protocolo MODBUS especifica sobre los buses o redes que pueden introducir algunos campos adicionales en la Unidad de aplicación de Datos (ADU).

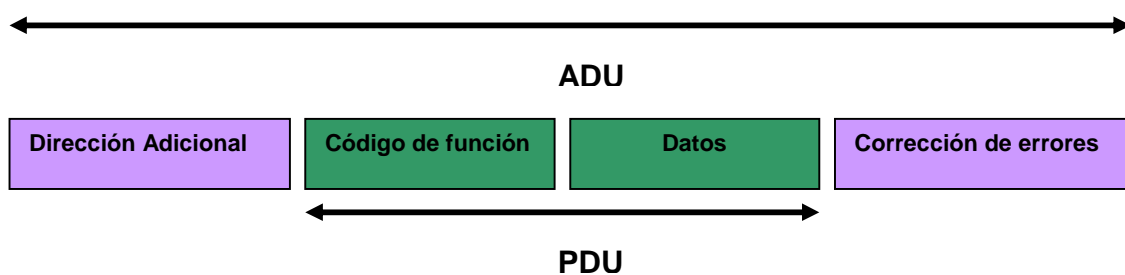


Figura 4. Trama general de Modbus

El cliente que inicia una transacción MODBUS se basa en la unidad de aplicación de datos MODBUS. El código de la función le indica al servidor que tipo de acción se va a realizar.

## 2.1.2 UNIDAD DE APLICACIÓN DE DATOS (ADU) EN MODBUS SOBRE TCP/IP

Esta sección describe el como se encapsula una solicitud o respuesta MODBUS cuando se ejerce MODBUS sobre una red TCP/IP.

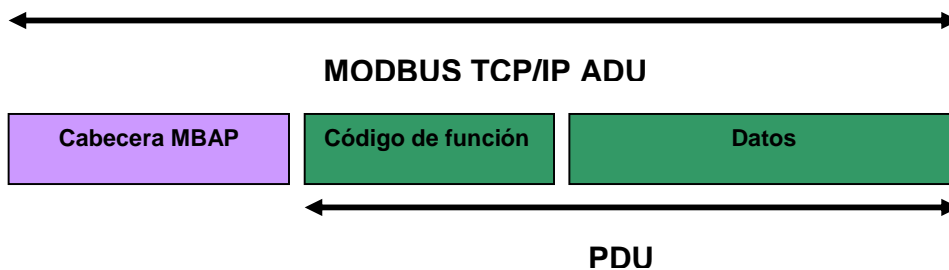


Figura 5. Solicitud/Respuesta de Modbus sobre TCP/IP

Una dedicada cabecera se utiliza sobre TCP/IP para identificar la unidad de aplicación de datos en MODBUS. Se llama la cabecera MBAP (cabecera de Protocolo de Aplicación MODBUS).

Esta cabecera proporciona algunas diferencias en comparación con la unidad de solicitud de datos MODBUS RTU utilizados en la línea serial:

- El campo “dirección de esclavo” de MODBUS utilizada habitualmente en la línea serial MODBUS es reemplazada por un solo byte en “Unidad de Referencia” en la cabecera MBAP.

La “Unidad de Referencia” es utilizada para comunicarse a través de dispositivos tales como puentes, enrutadores y puertas de enlace que utilizan una única dirección IP de apoyo a múltiples unidades independientes finales de MODBUS.

- Todas las solicitudes y las respuestas de MODBUS se han diseñado de tal forma que el destinatario pueda verificar que un mensaje ha terminado. Para los códigos de función donde el MODBUS PDU tiene una longitud fija, el código de la función por sí sola es suficiente. Para los códigos de función que llevan una cantidad de variables de datos en la petición o respuesta, el campo de datos incluye un byte contador.
- Cuando se lleva MODBUS sobre TCP, la longitud adicional de la información se efectúa en la cabecera MBAP para permitir que el destinatario reconozca los límites de mensajes incluso si el mensaje se ha dividido en múltiples paquetes para su transmisión.

La existencia de normas de longitudes explícitas e implícitas, y el uso de un CRC-32 para comprobar el código de error (en Ethernet) en los resultados, una posibilidad infinitesimal de corrupciones detectadas a una solicitud o mensaje de respuesta.

### 2.1.3 DESCRIPCIÓN DE CABECERA MBAP

La cabecera contiene los siguientes campos:

Campo	Longitud	Descripción	Cliente	Servidor
Identificador de transacción	2 Bytes	Identificación de una pregunta de Modbus / transacción de respuesta	Inicializada por el cliente	Recopilada por el servidor de la pregunta recibida.
Identificador de Protocolo	2 Bytes	0 = Protocolo Modbus	Inicializada por el cliente	Recopilada por el servidor de la pregunta recibida.
Longitud	2 Bytes	Numero de los siguientes bytes	Inicializada por el cliente (pregunta)	Inicializada por el servidor (respuesta)
Identificador de unidad	1 Byte	Identificación de un esclavo remoto en una línea serial o en otro tipo de bus.	Inicializada por el cliente	Recopilada por el servidor de la pregunta recibida.

Tabla 1. Descripción de la cabecera MBAP

La cabecera es de 7 bytes de longitud:

- **Identificador de Transacción** - Se utiliza para la operación de sincronización, el servidor MODBUS envía copias de respuesta al identificador de transacción de la solicitud.
- **Identificador de Protocolo** - Se usa para sistema de multiplexado. El protocolo MODBUS es identificado por el valor 0.

- **Longitud** - La longitud de campo es un byte contador de los siguientes campos, incluyendo la unidad identificadora y los campos de datos.
- **Unidad Identificadora** - Este campo se utiliza para el sistema de enrutamiento de acuerdo a su propósito. Por lo general se usa para comunicar a un MODBUS o una línea serial de esclavo de MODBUS a través de una puerta de enlace entre una red Ethernet TCP/IP y una línea serial de MODBUS. Este campo es fijado por el cliente en el MODBUS y la solicitud debe ser devuelta con el mismo valor en la respuesta del servidor.

## 2.2 DESCRIPCIÓN DE LOS CÓDIGOS DE FUNCIÓN EN MODBUS

El protocolo TCP/IP (pila de protocolos independientes) ofrece todos los recursos para que dos dispositivos puedan comunicarse entre sí a través de una red Ethernet de área local (LAN) o una red de área amplia global (WAN). Sin embargo, TCP/IP sólo garantiza que la aplicación de mensajes se transferirá entre estos dispositivos, no es garantía de que estos dispositivos realmente comprenden u operan entre sí. Para Modbus TCP/IP, esta capacidad es proporcionada por la capa de aplicación de protocolo Modbus.

Así, vemos que Modbus que opera de acuerdo al modelo común cliente/servidor (maestro/esclavo). Es decir, el cliente (maestro) envía un mensaje de petición (petición de servicio) al servidor (esclavo), y el servidor responde con un mensaje de respuesta.

Si el servidor no puede procesar la solicitud, la voluntad en lugar devolverá un código de función de error (exceptuando respuesta) que es el código de función original más 80H (es decir, con su más importante bit igual a 1).

Las funciones Modbus operan en los registros de memoria para configurar, monitorear y controlar dispositivos de I/O. Los dispositivos Modbus suelen incluir un mapa de registros. Se debe hacer referencia al mapa de registro de los dispositivos para obtener una mejor comprensión de su funcionamiento. También será útil hacer referencia al mapa de registro a medida que examina las funciones Modbus que se describen más adelante.

El modelo de datos de Modbus tiene una estructura simple que sólo distingue entre cuatro tipos de datos básicos:

Entradas discretas

Bobinas (salidas)

Registros de entrada (datos de entrada)

Registros de espera (datos de salida)

La petición de servicio (Unidad de datos del protocolo Modbus) está compuesta de un código función, y algunos números de bytes de datos adicionales, dependiendo de la función. En la mayoría de los casos, los datos adicionales son variables de referencia, como la dirección de un registro, ya que la mayoría de las funciones de Modbus operan sobre los registros.

Los registros de Modbus de un dispositivo están organizados en torno a cuatro tipos de referencia de datos básicos señalados a continuación con un número de referencia de dirección de la siguiente manera:

Referencia	Descripción
0xxxx	<u>Leer/Escribir salidas discretas o bobinas:</u> Una dirección de referencia "0x" es usada para conducir los datos a un canal de salida digital.
1xxxx	<u>Leer entradas discretas:</u> El estado ON/OFF de una dirección de regencia 1x es controlado por el correspondiente canal de entrada digital.
3xxxx	<u>Lectura de registros de entrada:</u> Un registro de referencia 3x contiene un numero de 16 bit recibido de una fuente externa. Ejemplo, una señal analógica.
4xxxx	<u>Leer/Escribir registros de salida:</u> Un registro 4x es usado para almacenar 16 bit de datos numéricos (binario o decimal) o para enviar los datos de la CPU a un canal de salida.

Tabla 2. Organización de los registros de Modbus

En el siguiente cuadro se pone de manifiesto un subconjunto de las funciones estándar Modbus (el registro de referencia, las direcciones que operan y su función). Las funciones a continuación se utilizan para acceder a los registros e indica en el registro del mapa módulo para enviar y recibir datos.

Código	Función	Referencia
01(01H)	Leer el estado de bobina (salida)	0xxxx
03(03H)	Leer registros en espera	4xxxx
04(04H)	Leer registros de entrada	3xxxx
05(05H)	Forzar una bobina simple (salida)	0xxxx
06(06H)	Preselección de registro único	4xxxx
15(0FH)	Forzar múltiples bobinas (salidas)	0xxxx
16(10H)	Preselección de registros múltiples	4xxxx
17(11H)	Reportar la dirección del esclavo	Escondido

Tabla 3. Funciones estándar de Modbus

### **2.2.1 Leer el estado de la bobina (01)**

Este comando lee el estado del interruptor ON/OFF de las salidas discretas o bobinas (0x direcciones de referencia) en el esclavo/servidor.

### **2.2.2 Leer los registros de espera (03)**

Lee el contenido binario de los registros de espera (4x direcciones de referencia) en el dispositivo esclavo. La transmisión emitida no es compatible.

### **2.2.3 Leer registros de entrada (04)**

Este comando va a leer el contenido binario de los registros de entrada (3x direcciones de referencia) en el dispositivo esclavo. La emisión la transmisión no es compatible.

### **2.2.4 Forzar una bobina simple (05)**

Forzar una sola bobina/salida (0x dirección de referencia) ON/OFF. Con emisiones de transmisión (dirección 0), las fuerzas de la misma bobina conectados en red a todos esclavos (sólo Modbus).



### **2.2.5 Preselección de registro único (06)**

Este comando preselecciona un único registro de espera (4x dirección de referencia) a un valor específico. La emisión de la transmisión es compatible para este comando y actuará para preseleccionar el mismo registro en todas las redes de esclavos.

### **2.2.6 Forzar múltiples registros (15)**

Se fuerzan al mismo tiempo una serie de bobinas (0x dirección de referencia) ya sea en ON o en OFF. La transmisión cuenta con el apoyo de este comando y actuará a forzar al mismo bloque de bobinas de todos los esclavos en la red.

### **2.2.7 Preseleccionar registros múltiples (16)**

Preseleccionar un bloque de registros de espera (4x direcciones de referencia) a determinados valores. La transmisión cuenta con el apoyo de este comando y actuará preestablecido para el mismo bloque de registros en todos los esclavos de la red.

### **2.2.8 Reportar la dirección del esclavo (17)**

Este comando devuelve el modelo, serial, y el número de la empresa a un dispositivo Esclavo/Servidor de Acromag (97xEN para este ejemplo), el estado del indicador de ejecutar y cualquier otra información específica al dispositivo. Este comando no se ocupa de guardar la dirección en el Mapa de registros y la transmisión no es compatible.

## **CAPITULO III**

### **DESCRIPCIÓN FUNCIONAL**

El componente de la arquitectura MODBUS presentada, es un modelo general que incluye tanto al cliente MODBUS como a los componentes del servidor usados en cualquier dispositivo. Algunos dispositivos pueden proporcionar solo el componente del servidor o del cliente.

### 3.1 MODELO DE LOS COMPONENTES DE LA ARQUITECTURA DE MODBUS

Esta sección muestra un breve resumen de los componentes de la arquitectura del servicio de mensajería de MODBUS, seguido de una descripción de cada uno de estos presentados en la siguiente figura.

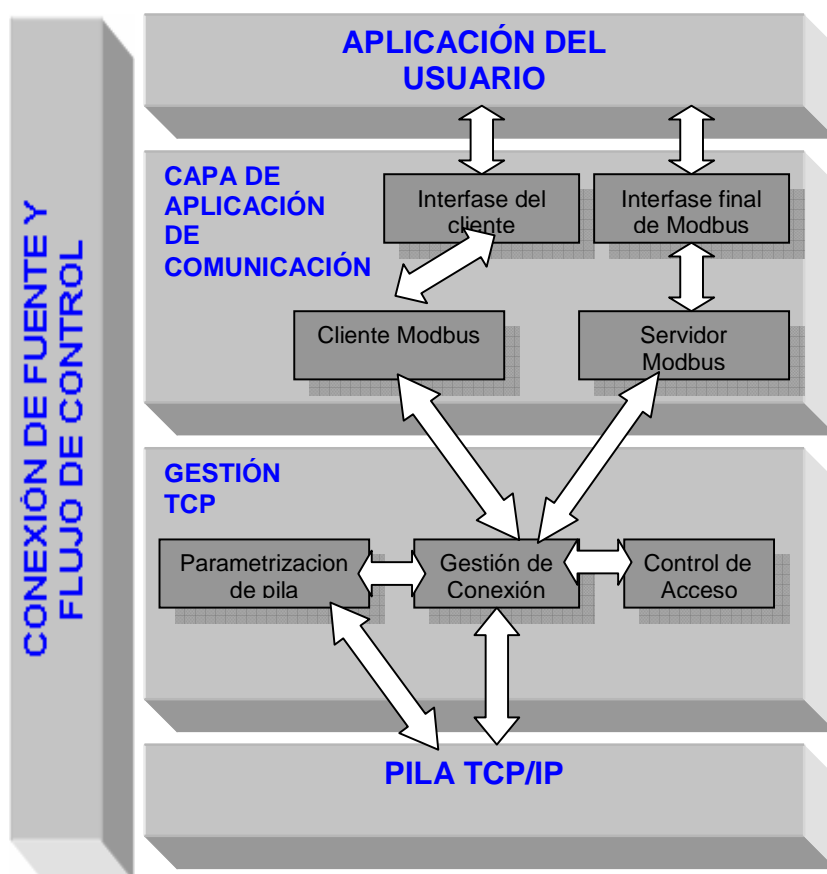


Figura 6. Arquitectura del servicio de Mensajería de Modbus

- **Capa de aplicación de comunicación**

Un dispositivo MODBUS podrá disponer de un cliente y/o un servidor de interfaz MODBUS.

Cuatro áreas pueden componer esta interfaz: entrada discreta, salida discreta (bobinas), los registros de entrada y registros de salida.

Tablas primarias	Tipo de objeto	Tipo de	Comentarios
Entradas discretas	Bit sencillo	Solo lectura	Este tipo de datos pueden ser provistos por un sistema de I/O.
Bobinas	Bit sencillo	Solo escritura	Este tipo de datos pueden ser alterados por un programa de aplicación.
Registros de entrada	Palabra de 16 bits	Solo lectura	Este tipo de datos pueden ser provistos por un sistema de I/O.
Registros de espera	Palabra de 16 bits	Solo escritura	Este tipo de datos pueden ser alterados por un programa de aplicación.

Tabla 4. Componentes de la interfaz de Modbus

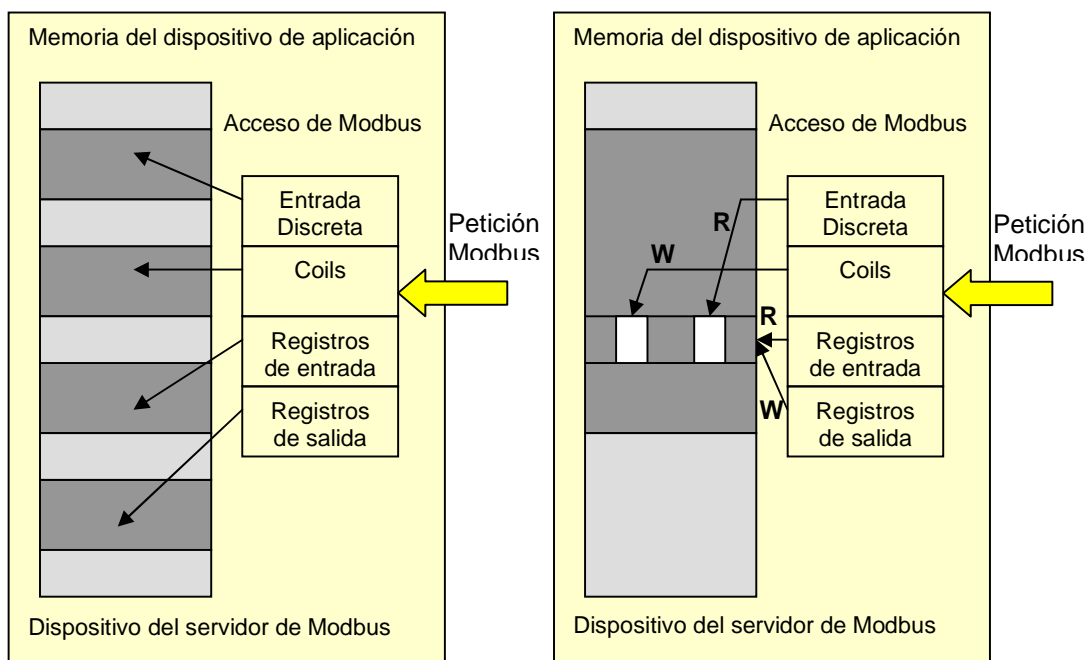


Figura 7. Modelo de datos de Modbus con Bloques separados

Figura 8. Modelo de datos de Modbus con un solo bloque

### ➤ CLIENTE MODBUS

El Cliente MODBUS permite al usuario aplicaciones para el control, explícitamente el intercambio de información con un dispositivo remoto. El Cliente MODBUS construye una solicitud del parámetro de MODBUS que figura en una demanda enviada por el usuario a la solicitud de la interfase de los clientes MODBUS.

El cliente MODBUS utiliza una transacción MODBUS la cual incluye la espera de la gestión y el procesamiento de un MODBUS de confirmación.

### ➤ INTERFASE DE LOS CLIENTES MODBUS

La interfaz del cliente MODBUS proporciona una interfaz de usuario que permite la aplicación de construir las solicitudes MODBUS de diversos servicios, incluido el acceso a los objetos de aplicación MODBUS. La interfaz del cliente MODBUS (API) no se especifica en este pliego de condiciones, si bien se describe un ejemplo en el modelo de aplicación.

### ➤ SERVIDOR DE MODBUS

Para la recepción de una solicitud MODBUS este modulo activa una acción local para leer, para escribir o para lograr algunas otras acciones.

La tramitación de estas acciones se hace totalmente transparente para la aplicación del programador.

Las principales funciones del servidor MODBUS son esperar para una solicitud MODBUS sobre el puerto TCP 502, para el tratamiento de esta solicitud y luego construir una respuesta en función de MODBUS dependiendo del contexto del dispositivo.

#### ➤ INTERFAZ FINAL MODBUS

Es una interfaz del servidor de MODBUS para el usuario en donde en la aplicación son definidos los objetos.

- **Gestión de la TCP**

Una de las principales funciones del servicio de mensajes es la gestión de la comunicación y el establecimiento que termina y para de gestionar el flujo de datos en conexiones de red TCP.

#### ➤ Gestión de la conexión

Una comunicación entre un cliente y el servidor MODBUS requiere el uso de un módulo de gestión de conexión TCP. Es responsable de administrar a nivel mundial la mensajería y conexiones TCP.

Se proponen dos posibilidades para la gestión de la conexión. La propia demanda del usuario gestiona las conexiones TCP o la gestión de la conexión es totalmente realizada por este módulo y por lo tanto, es transparente para la aplicación del usuario. La última solución supone una menor flexibilidad.

➤ **Módulo de control de acceso**

En determinados contextos críticos, la accesibilidad a los datos internos de los dispositivos debe ser prohibidos para los anfitriones. Eso es por qué un modo de seguridad es necesitado y los procesos de seguridad pueden ser implementados en caso necesario.

- **Capa de pila TCP/IP**

La pila TCP/IP puede ser parametrizada con el fin de adaptar el control de flujo de datos, la dirección de gestión y gestión de la conexión a diferentes limitaciones específicas a un producto o a un sistema. Generalmente la interfaz del socket BSD se utiliza para gestionar las conexiones TCP.

- **Gestión de recursos y el control de flujo de datos**

Con el fin de equilibrar la mensajería entrante y saliente y el flujo de datos entre el cliente MODBUS y el servidor, el mecanismo de control de flujo de datos se proporciona en todas las capas de la pila de mensajes MODBUS.

La gestión de los recursos y el control de flujo esta basado en el control de flujo interno de TCP añadido con algunos datos de control de flujo en la capa de enlace de datos y también en el nivel de las aplicaciones de usuario.

## **3.2 GESTIÓN DE CONEXIÓN TCP**

### **3.2.1 Módulo de gestión de las conexiones**

#### **3.2.1.1 Descripción general**

Una comunicación MODBUS requiere del establecimiento de una conexión TCP entre un cliente y un servidor.

El establecimiento de la conexión puede ser activado, ya sea en forma explícita por el usuario del módulo de solicitud o automáticamente por el módulo de gestión de la conexión TCP.

En el primer caso, una aplicación de programación de interfaz tiene que ser proporcionada en la solicitud del modulo de usuario para administrar completamente la conexión. Esta solución ofrece la flexibilidad para la aplicación programador, pero se requiere una buena experiencia en el mecanismo TCP/IP.



En el segundo caso, la gestión de la conexión de TCP esta oculta al usuario para que la aplicación sólo envíe y reciba mensajes MODBUS. El módulo de gestión de conexión TCP es el encargado de establecer una nueva conexión TCP cuando es necesario.

### **Reglas de implementación**

- 1) Sin el requisito explícito del usuario, se recomienda la aplicación automática de la gestión de conexión TCP.
- 2) Se recomienda mantener la conexión TCP abierta con un dispositivo remoto y no de apertura y cerrado para cada transacción MODBUS / TCP.  
**Nota:** Sin embargo, el cliente MODBUS debe ser capaz de aceptar una solicitud de cerrar el servidor y el cierre de la conexión. La conexión puede ser reabierta en caso necesario.
- 3) Se recomienda para una MODBUS Cliente para abrir un mínimo de conexiones TCP con un mando a distancia MODBUS servidor (con la misma dirección IP). Una conexión por solicitud podría ser una buena opción.
- 4) Varias transacciones MODBUS pueden ser activadas simultáneamente en la misma conexión TCP.

**Observación:** Si esto se lleva a cabo, entonces la transacción MODBUS debe utilizarse para identificar de forma exclusiva que coincidan las peticiones con las respuestas.

- 5) En caso de una comunicación bidireccional entre dos entidades MODBUS remotas (cada una de ellas es cliente y servidor), es necesario abrir conexiones separadas para el cliente y el flujo de datos para el servidor de datos.
- 6) Una trama TCP debe transportar sólo una MODBUS ADU. Se aconseja contra el envío de múltiples peticiones o respuestas MODBUS en el mismo TCP PDU.

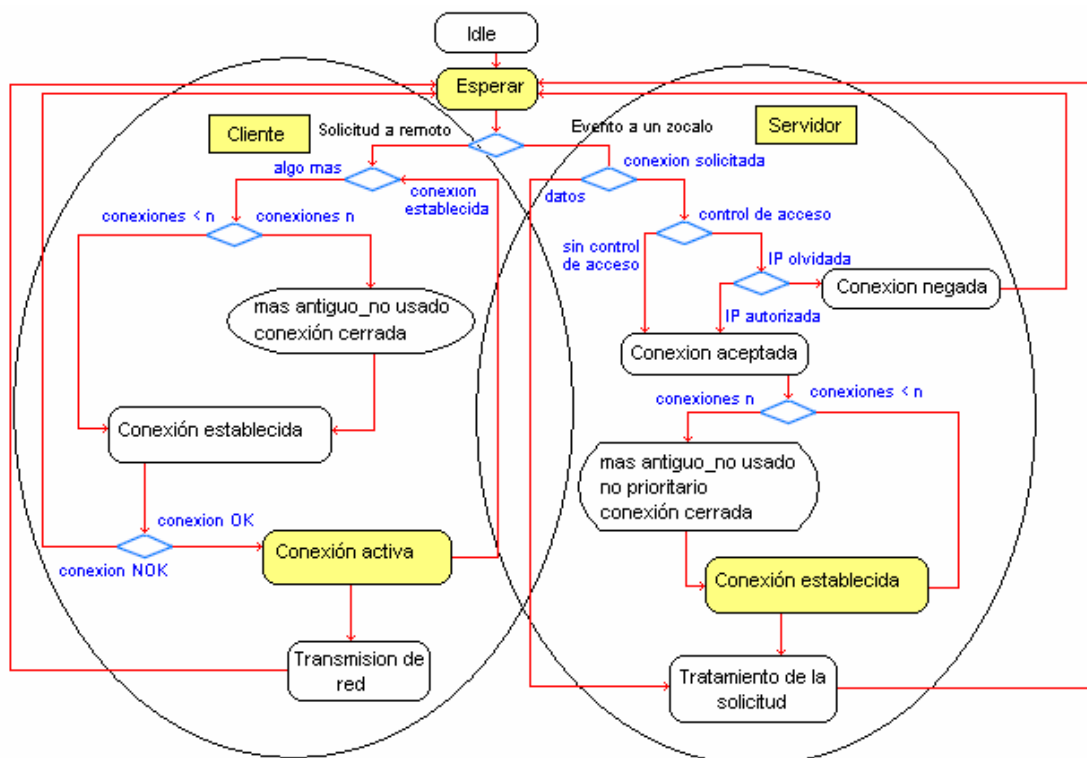


Figura 9. Diagrama de actividades de la gestión de conexión TCP

## **1. Gestión de conexión TCP explícitas**

El módulo de aplicación del usuario se encarga de gestionar todas las conexiones TCP: conexiones activa y pasiva y conexiones de fin, etc. Esta gestión se realiza para todas las comunicaciones MODBUS cliente y un servidor.

La interfaz BSD Socket se utiliza en el módulo de aplicación de usuario para gestionar la conexión TCP. Esta solución ofrece una total flexibilidad, pero que implica que el programador de la aplicación debe tener suficientes conocimientos de TCP.

Un número limitado de conexiones de cliente y servidor tiene que ser configurado teniendo en cuenta las capacidades del dispositivo y exigencia.

## **2. Gestión de conexión TCP automática**

La gestión de conexión es totalmente transparente para el modulo la aplicación del usuario. El módulo de gestión de conexión podrá aceptar un número suficiente de conexiones de cliente y servidor.

No obstante, un mecanismo debe aplicarse en caso de exceder el número autorizado de conexión. En tal caso, le recomendamos que para cerrar la conexión más antigua no utilizados.

Una conexión remota con un socio es establecida en el primer paquete recibido de un cliente remoto o de una aplicación de usuario local. Esta conexión se cerrará si llega un cese de la red o si lo decidió el dispositivo localmente. A la recepción de una solicitud de conexión, la opción de control de acceso puede ser usado para prohibir el acceso del dispositivo a clientes no autorizados.

El modulo de gestión de conexión TCP utiliza el interfaz Stack (por lo general, la interfaz BSD Socket) para comunicarse con la pila TCP/IP.

Con el fin de mantener la compatibilidad entre los requisitos del sistema y los recursos del servidor, la gestión TCP mantendrá 2 grupos de conexión:

- El primer grupo (grupo de conexión prioritaria) está hecha de conexiones que nunca se cierra en una iniciativa local. Una configuración debe ser proporcionada a este grupo para su ajuste. El principio que debe aplicarse es asociar una determinada dirección IP con cada posible conexión de este grupo.

Los dispositivos con estas direcciones IP se dice que están "marcados". Toda nueva conexión que se solicita por un dispositivo marcado debe ser aceptada, y tendrán prioridad de este grupo de conexión prioritaria. También es necesario configurar el número máximo de conexiones permitidas para cada dispositivo remoto para evitar que el mismo dispositivo utilice todas las conexiones del grupo prioritario.

- El segundo grupo (grupo de conexión no prioritario) contiene las conexiones para dispositivos no marcados. La regla que siguen es cerrar la conexión más antigua cuando una nueva solicitud de conexión se reciba a partir de un dispositivo no marcado y cuando no hay más conexiones disponibles en el grupo.

Una configuración podría estar opcionalmente provista para asignar el número de conexiones disponibles en cada grupo. Sin embargo los diseñadores pueden definir el número de conexiones de acuerdo al diseño si es necesario.

### **3.2.1.2 Descripción de gestión de conexión**

#### **➤ Estableciendo la conexión**

El servicio de mensajería MODBUS debe proporcionar un socket de escucha en el Puerto 502, que permite aceptar la nueva conexión e intercambiar datos con otros dispositivos.

Cuando los servicios de mensajería necesitan intercambiar datos con un servidor remoto, debe abrir una nueva conexión de cliente con un 502, a fin de intercambiar datos con esta distancia. El puerto local debe ser superior a 1024 y diferente para cada conexión de cliente.

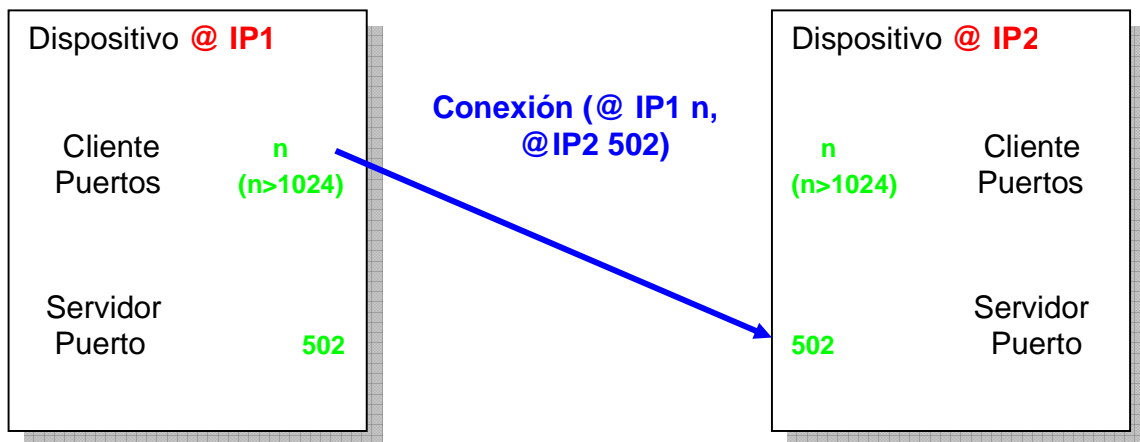


Figura 10. Estableciendo la conexión Modbus TCP

Si el número de conexiones de cliente y servidor es superior al número de conexiones autorizadas, la conexión mas antigua sin utilizar es cerrada. El mecanismo de control de acceso puede ser activado, para comprobar si la dirección IP del cliente remoto es autorizada. De no ser así la nueva conexión es rechazada.

#### ➤ **Transferencia de datos Modbus**

A solicitud MODBUS tiene que ser enviada a la conexión TCP derecha que ha ya sido abierta. La dirección IP remota se utiliza para encontrar la conexión TCP. En el caso de múltiples conexiones TCP abiertas remotas, una conexión tiene que ser escogida y enviada al mensaje de MODBUS, diferentes criterios de elección puede ser usado como el más antiguo, el primero de ellos. La conexión tiene que mantenerse abierta durante todas las comunicaciones MODBUS.

Como se describe en las siguientes secciones, un cliente puede iniciar varias transacciones MODBUS con un servidor sin tener que esperar la finalización del anterior.

➤ **Conexión de cierre**

Cuando las comunicaciones MODBUS se terminan entre un cliente y un servidor, el cliente tiene que iniciar una conexión de cierre de la conexión utilizada para estas comunicaciones.

### **3.2.2 Impacto de los modos de funcionamiento de una conexión TCP**

Algunos Modos de funcionamiento (romper la comunicación entre dos puntos finales de operación, caída y reinicio uno de los puntos finales,...) pueden tener impactos sobre la conexiones TCP. Una conexión puede verse cerrados o abortado por un lado, sin el conocimiento de la otra parte.

La conexión se dice que es "medio abierta". Esta sección describe el comportamiento de cada una de los principales modos de funcionamiento. Se supone que el mecanismo de mantener viva una TCP se utiliza en ambos puntos finales.

### **3.2.2.1 Ruptura de conexión entre dos puntos finales de operación**

El origen de que se rompa una comunicación puede ser la desconexión del cable Ethernet en el servidor. El comportamiento esperado es:

- Si los paquetes no son enviados por la conexión:

La rotura de la comunicación no será vista si dura menos que el valor del temporizador de mantener vivo. Si la rotura de la comunicación dura más, un error se devuelve a la capa de gestión TCP que puede restablecer la conexión.

- Si algunos paquetes se envían antes y después de la desconexión

Los algoritmos de retransmisión TCP (Jacobson, algoritmos de Karn y exponenciales) se activan. Esto puede dar lugar a una capa de reset desde la pila TCP de la conexión antes de que el temporizador de mantener vivo se haya terminado.

### **3.2.2.2 Caída y reinicio el punto final del servidor**

Después de la caída y reinicio del servidor, la conexión es "medio abierta" en el lado del cliente. El comportamiento esperado es:



- Si no se envían paquetes en conexión medio abierta:

La conexión TCP medio abierta es vista abierta desde el lado del cliente siempre y cuando el temporizador de mantener vivo no haya terminado. Después de que un error se devuelve a la capa de gestión TCP se puede restablecer la conexión.

- Si algunos paquetes son enviados en conexión medio abierta:

El servidor recibe datos de una conexión que ya no existe. La pila TCP envía una capa reset para cerrar la conexión medio abierta en el lado del cliente.

### **3.2.2.3 Caída y reinicio del Cliente**

Después de la caída y reinicio del cliente, la conexión es "medio abierta" en el lado del servidor. El comportamiento esperado es:

- No se envían paquetes en conexión medio abierta:

La conexión TCP medio abierta es vista abierta desde el lado del servidor, siempre y cuando el temporizador de mantener vivo no haya terminado. Después de que un error se devuelve a la capa gestión TCP se puede restablecer la conexión.

- Si el cliente abre una nueva conexión antes de mantener vivo el temporizador es más:

Dos casos tienen que ser estudiados:

1. La conexión de apertura tiene las mismas características que la conexión medio abierta en el lado del servidor (el mismo origen y los puertos de destino, el mismo origen y direcciones IP de destino), por lo tanto la apertura de la conexión fallará en el nivel de la pila TCP después de la hora del establecimiento de la conexión (75 segundos en la mayoría de las implementaciones de Berkeley).

Para evitar este largo tiempo durante el cual no es posible comunicar, es muy aconsejable asegurarse que los diferentes números de puerto de origen y que los anteriores sean utilizados para la apertura de una conexión después de un reinicio en el lado del cliente.

2. La conexión de apertura no tiene las mismas características que la conexión medio abierta en el lado del servidor (fuente diferente de puertos, mismo puerto de destino, el mismo origen y dirección IP de destino), por lo que la conexión es abierta al nivel de la pila TCP y señala la capa de gestión TCP del servidor.

Si la capa de gestión TCP del servidor sólo admite una conexión desde una dirección IP de un cliente remoto, puede cerrar la conexión medio abierta más antigua y usar la nueva.

Si la capa de gestión TCP del servidor soporta varias conexiones desde una dirección IP de un cliente remoto, la nueva conexión permanece abierta y la antigua conexión medio abierta expira hasta que el temporizador de mantenerse vivo el temporizador devuelva un error a la capa de gestión TCP. Después la capa de gestión TCP podrá reiniciar la vieja conexión.

### **3.2.3 Modulo de control de acceso**

El objetivo de este módulo es comprobar cada nueva conexión y usando una lista de direcciones IP remotas, el modulo puede autorizar o prohibir a un cliente remoto la conexión TCP.

En el contexto crítico el programador de la aplicación tiene que elegir el modo de control de acceso con el fin de garantizar su acceso a la red. En tal caso, el necesita autorizar o prohibir el acceso remoto para cada dirección IP.

El usuario tiene que proveer una lista de direcciones IP y especificar si cada dirección IP es autorizada o no. Por defecto, a modo de seguridad, las direcciones IP no se configuran por el usuario que las prohíbe.

Por lo tanto, con el modo de control de acceso una conexión procedente de una dirección IP desconocida es cerrada.

### **3.3 EL USO DE LA PILA TCP/IP**

Una pila TCP/IP proporciona una interfaz para gestionar las conexiones, para enviar y recibir datos, y también para hacer ajustar algunos parámetros con el fin de adaptar el comportamiento de la pila al dispositivo o sistema de limitaciones.

El objetivo de esta sección es dar una visión general de la interfaz de la pila y también alguna información concerniente a los parámetros de la pila. Esta visión general se centra en las características utilizadas por la mensajería de Modbus.

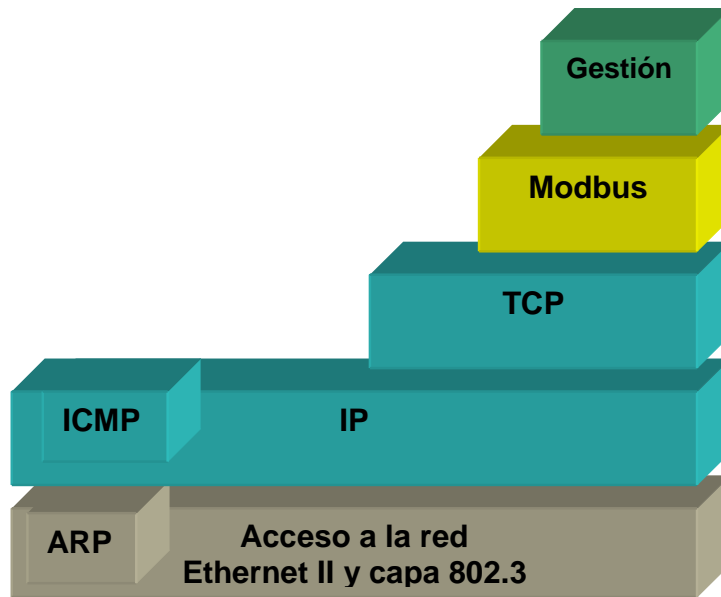


Figura 11. Interfaz de la pila TCP/IP

La interfaz de la pila se basa generalmente en la BSD (Berkeley Software Distribution).

### 3.3.1 El uso de la interfaz BSD Socket

Un enchufe es un punto final de la comunicación. Es el elemento básico para la comunicación. Una comunicación MODBUS se ejecuta mediante el envío y recepción de datos a través de sockets. La librería TCP/IP ofrece sólo sockets corrientes usando TCP y el suministro de una conexión basada en servicios de comunicaciones.

Las sockets son creados a través de la función “**socket ()**”. El número de un socket es devuelto que es luego usado por el creador para acceder al mismo. Los sockets se crean sin direcciones (dirección IP y número de puerto).

Hasta que un puerto es asignado a un socket, este no puede ser utilizado para recibir datos.

La función “**asignar ()**” se utiliza para asignar un número de puerto a un socket. Esta función crea una asociación entre el socket y el número de puerto especificado.

Con el fin de iniciar una conexión, el cliente debe usar la función “**conectar ()**” especificando el número de socket, la dirección IP remota y el numero del puerto del control remoto de escucha (conexión activa).

Con el fin de completar una conexión, el servidor debe usar la función de “**aceptar ()**” especificando el número de socket que se muestra en “**escuchar ()**” (conexión pasiva). Un nuevo socket se crea con las mismas propiedades que el inicial.

Este nuevo socket es conectado al socket del cliente, y su número se devuelve al servidor. De esta manera el socket inicial esta libre para otros clientes que tal vez quieran conectar con el servidor.

Tras el establecimiento de la conexión TCP los datos pueden ser transferidos. Las funciones “**enviar ()**” y “**recibir ()**” son diseñadas específicamente para ser usadas con los sockets que ya están conectados.

El “**setsockopt ()**” le permite al creador de un socket ajustar algunas opciones asociadas a un socket y dichas opciones permiten modificar el comportamiento del socket.

La función “**seleccionar ()**” le permite al programador poner a prueba todos los eventos en los sockets.

La función “**apagado ()**” permite al usuario de un socket desactivar “enviar ()” y/o “recibir ()” en el mismo. Una vez que un socket ya no es necesario, su descripción puede ser descartada mediante la función “**cerrar ()**”.

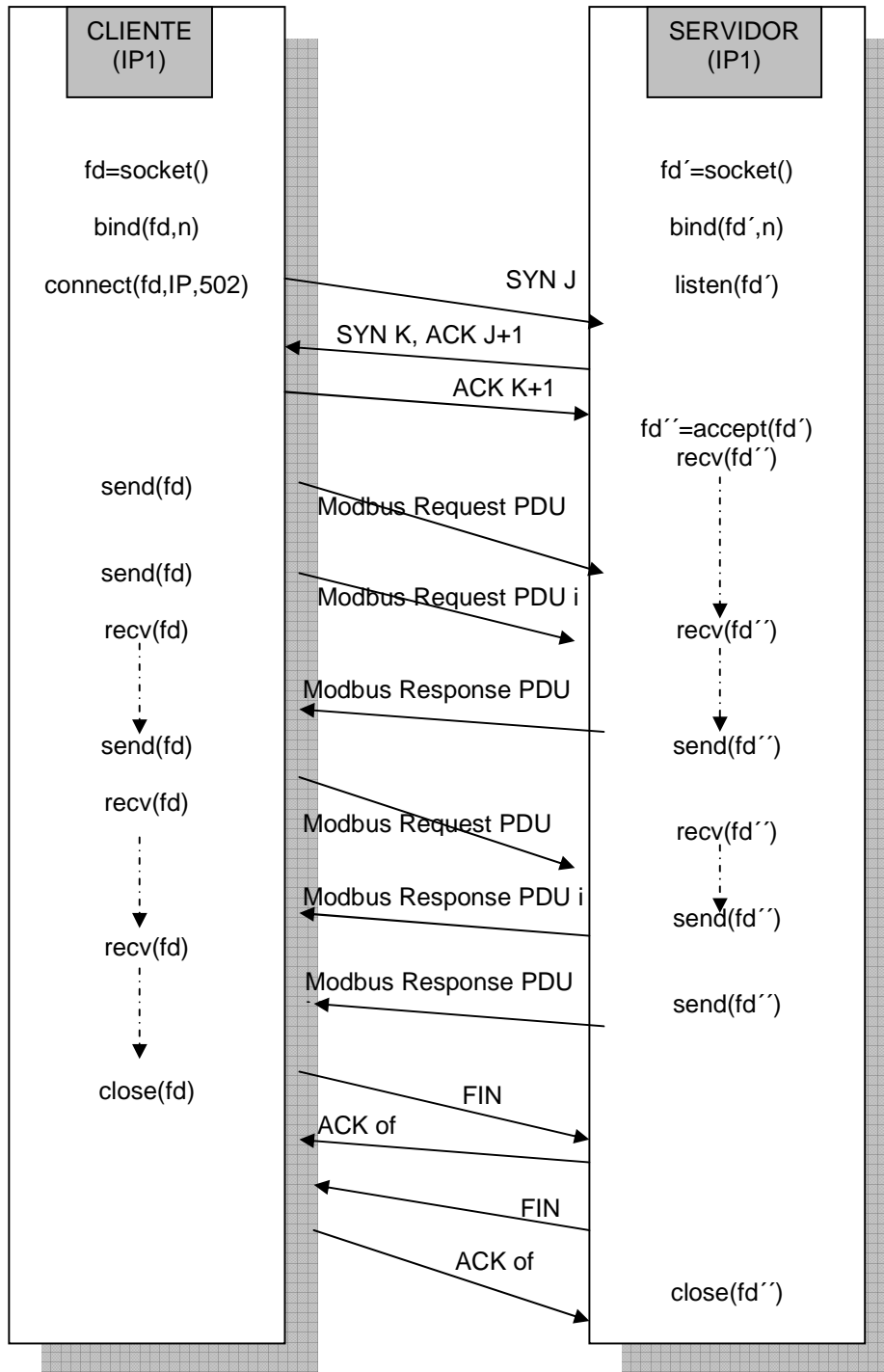


Figura 12. Intercambios de Modbus

La figura describe los intercambios de Modbus y la comunicación entre un cliente y un servidor. El Cliente establece la conexión y envía 3 solicitudes al servidor sin necesidad de esperar la respuesta de la primera. Después de recibir todas las respuestas que el cliente cierra la conexión correctamente.



### 3.3.2 Parámetros de la capa IP

#### 3.3.2.1 Parámetros IP

Los siguientes parámetros deben ser configurados en la capa IP de una implementación Modbus:

- Dirección IP local: La dirección IP puede ser parte de una clase A, B ó C.
- Máscara de subred: Una red IP puede ser hecha para una variedad de razones: el uso de diferentes medios físicos (tales como Ethernet, WAN, etc), un uso más eficiente de direcciones de red, y la capacidad para controlar el tráfico de la red. La Máscara de subred tiene que ser consistente con la clase de dirección IP de la dirección IP local.
- Puerta de enlace por defecto: La dirección IP de la puerta de enlace por defecto tiene que estar en la misma subred que la dirección IP local. El valor 0.0.0.0 está prohibido. Si la puerta de enlace no se define entonces ese valor va a ser establecido ya sea en 127.0.0.1 o el de la dirección IP local.

**Nota:** El servicio de mensajería Modbus no requiere la función de fragmentación en la capa IP. El punto final de la IP local se configura con una dirección IP local y con una máscara de subred y una puerta de enlace predeterminada (diferentes de 0.0.0.0).

### 3.4 CAPA DE APLICACIÓN DE COMUNICACIÓN

#### 3.4.1 CLIENTE MODBUS

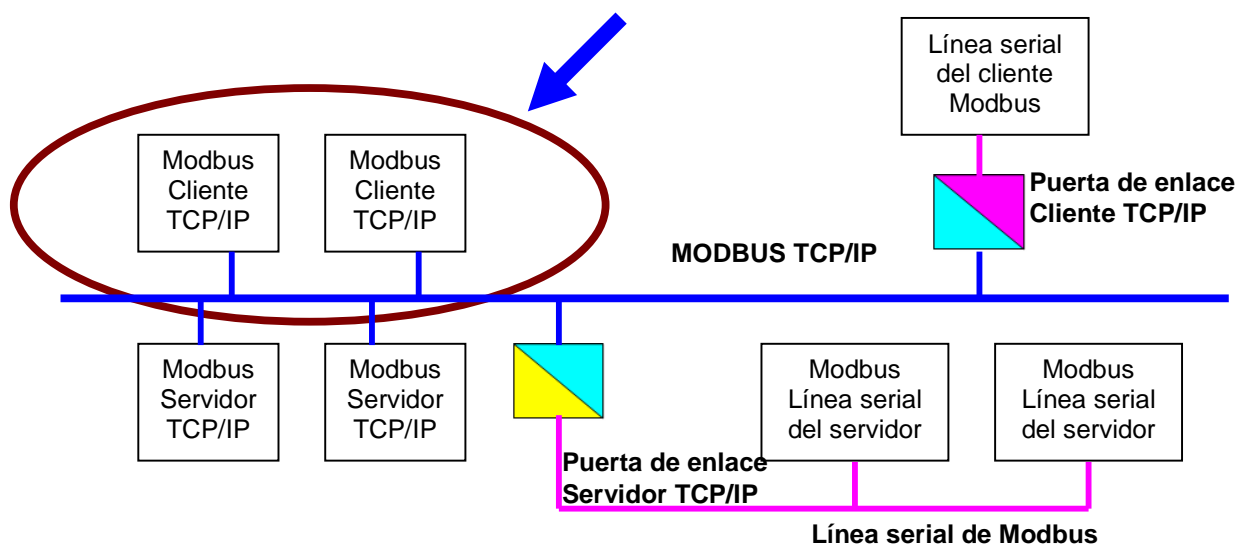


Figura 13. Unidad cliente de Modbus

### 3.4.1.1 Diseño del Cliente Modbus

La definición del protocolo MODBUS/TCP permite el diseño simple de un cliente. El siguiente diagrama de actividad describe los principales tratamientos que son procesados por un cliente para enviar una solicitud MODBUS y para tratar una respuesta MODBUS.

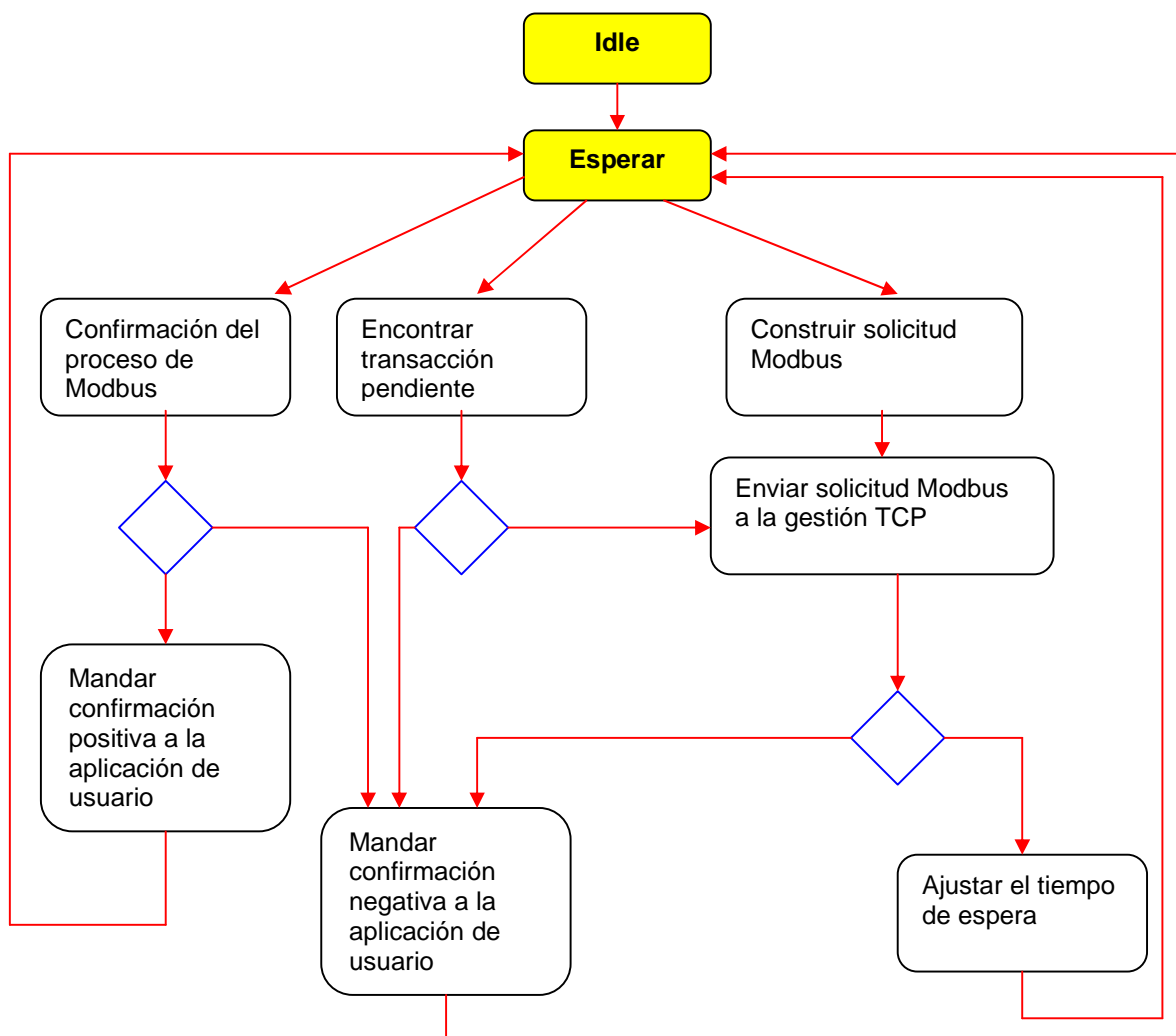


Figura 14. Diagrama de actividad del cliente Modbus

Un cliente de Modbus puede recibir tres eventos:

- Una nueva demanda de la aplicación de usuario para enviar una solicitud, en este caso una solicitud Modbus tiene que ser codificada y enviada a la red usando el componente de gestión de servicio TCP.  
La capa inferior (módulo de gestión TCP) puede devolver un error debido a un error de conexión TCP o algunos otros errores.
- Una respuesta de la gestión de TCP, en este caso el cliente tiene que analizar el contenido de la respuesta y enviar una confirmación a la aplicación de usuario.
- La expiración de un tiempo de espera debido a la falta de respuesta. Un nuevo reintento puede ser enviado a la red o una confirmación negativa puede ser enviada a la solicitud del usuario. **Nota:** Estos reintentos son iniciados por el cliente MODBUS, algunos otros reintentos también puede hacerse de la capa TCP en el caso de que TCP reconozca la falta.

#### 3.4.1.2 Construir una Solicitud MODBUS

Tras la recepción de una demanda de la aplicación de usuario, el cliente tiene que construir una solicitud Modbus y enviarla a la gestión de TCP.

La construcción de la solicitud Modbus se puede dividir en varias sub – tareas:

- La instalación de una transacción de Modbus permite a los clientes memorizar toda la información requerida a fin de obligar más tarde la respuesta a la solicitud y enviar la confirmación a la aplicación del usuario.
- La codificación de la solicitud Modbus (PDU + cabecera MPAB). La aplicación de usuario que inicia la demanda tiene que proporcionar toda la información requerida que permita que el cliente pueda codificar la solicitud. El MODBUS PDU está codificado de acuerdo con las Especificaciones de Aplicación del Protocolo Modbus. (código de función MB, los parámetros asociados y la aplicación de datos).

Todos los campos de la cabecera MBAP están llenos. A continuación, la solicitud MODBUS ADU se construye para prefijar la PDU con la cabecera MBAP.

- El envío de la solicitud MODBUS ADU al módulo de gestión TCP que se encarga de encontrar el socket TCP hacia el servidor remoto. Además de la ADU MODBUS la dirección IP de destino también debe ser aprobada.

El siguiente diagrama describe la actividad más profundamente que la figura anterior, la etapa de construcción de la solicitud.

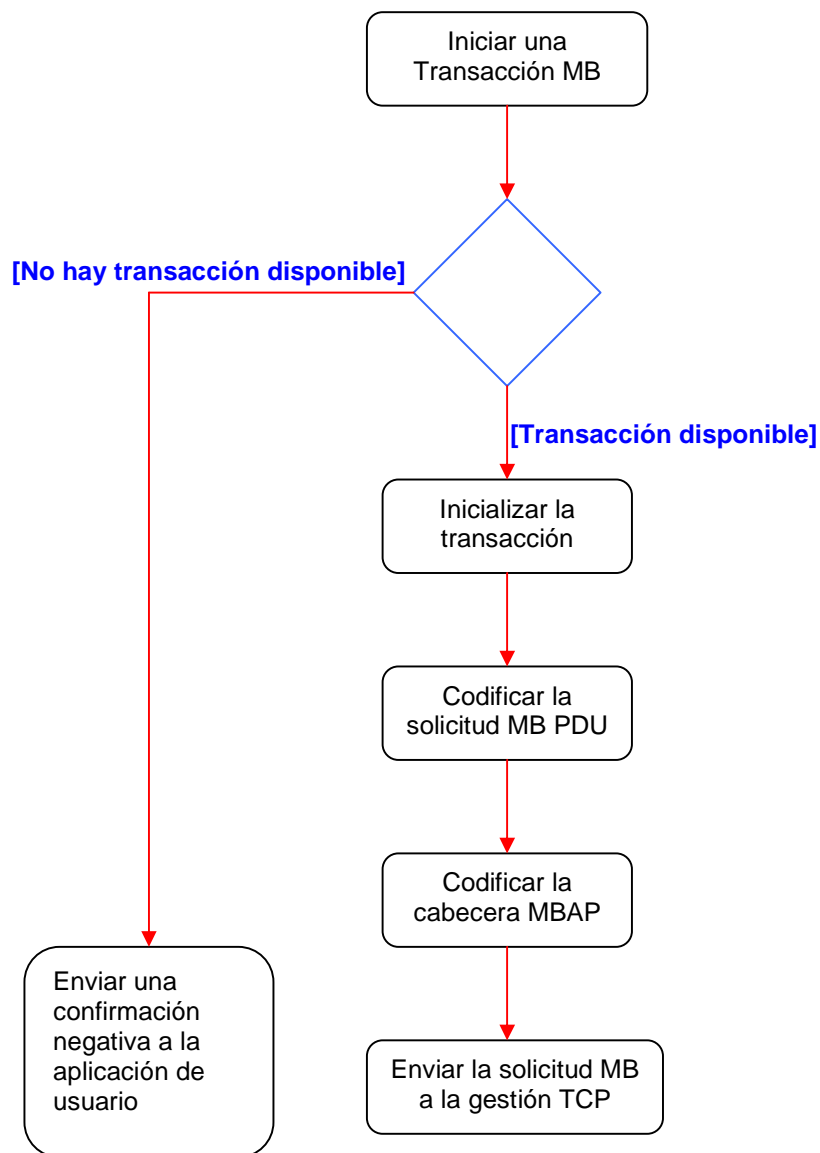


Figura 15. Diagrama de actividad de la construcción de la solicitud

El siguiente ejemplo describe la solicitud MODBUS ADU codificada para leer el registro # 5 en un servidor remoto:

Codificación de la solicitud MODBUS ADU:

	Descripción	Tamaño	Ejemplo
Cabecera MBAP	Identificador de transacción Alto	1	0x15
	Identificador de transacción Bajo	1	0x01
	Identificador de protocolo	2	0x0000
	Longitud	2	0x0006
	Identificador de unidad	1	0xFF
Solicitud Modbus	Código de función	1	0x03
	Dirección de inicio	2	0x0004
	Cantidad de registros	2	0x0001

Tabla 5. Solicitud Modbus ADU

- **Identificador de transacción**

La operación de identificación se utiliza para asociar una futura respuesta a la solicitud. Por lo tanto, a la vez, en una conexión TCP, este identificador debe ser único.

Hay varias maneras de utilizar el identificador de transacción:

- Por ejemplo, puede ser utilizado como un simple "número de secuencia TCP" con un contador que se incrementa en cada solicitud.

- También puede ser utilizado como índice inteligente como un índice o como puntero para identificar una transacción en un contexto a fin de memorizar el servidor remoto actual y la solicitud de MODBUS pendiente.

Normalmente, en una línea serie de MODBUS, un cliente debe enviar una petición a la vez. Esto significa que el cliente debe esperar la respuesta a la primera solicitud, antes de enviar una segunda solicitud. En TCP/MODBUS, varias solicitudes se pueden enviar sin esperar una confirmación del mismo servidor.

El número de solicitudes aceptadas por un servidor depende de su capacidad en términos de número de recursos y el tamaño de las ventanas TCP. De la misma manera el número de transacciones que ha iniciado simultáneamente un cliente depende también de su capacidad de recursos.

Esta aplicación se llama parámetro “**Numero máximo de transacciones del cliente**” y debe ser descrita como una de las características del cliente Modbus. Dependiendo del tipo de dispositivo este parámetro puede tener un valor de entre 1 y 16.



- **Identificador de unidad**

Este campo es utilizado para el enrutamiento cuando se direcciona un dispositivo en Modbus o una sub – red de línea serial de Modbus. En ese caso el identificador de unidad lleva la dirección del esclavo de Modbus del dispositivo remoto:

Si el servidor Modbus está conectado a Modbus o una sub – red de línea serial de Modbus y dirigido a través de un puente o una puerta de enlace, el identificador de unidad de MODBUS debe identificar el dispositivo esclavo conectado a la sub – red detrás del puente o la puerta de enlace.

La dirección IP de destino identifica el propio puente y el puente utiliza el identificador Modbus para que transmita la solicitud al dispositivo esclavo correcto. Las direcciones del dispositivo esclavo MODBUS en línea serie se les asignan de 1 a 247 (decimal). Dirección 0 se usa como dirección de emisión.

En TCP/IP, el servidor MODBUS es dirigido utilizando su dirección IP, por lo el identificador de unidad de MODBUS es inútil. El valor 0xFF tiene que ser usado.

Cuando se dirige un servidor MODBUS conectado directamente a una red TCP/IP, se recomienda no utilizar una dirección de esclavo MODBUS muy significativa en el campo de "Identificador de Unidad".

En el caso de una re-asignación de las direcciones IP dentro de un sistema automatizado y si una dirección IP asignada previamente a un servidor MODBUS se asigna a una puerta de enlace, utilizando una dirección de esclavo significativa podría causar problemas a causa de un mal enrutamiento de la puerta de enlace.

Usando una dirección no tan significativa, la puerta de enlace descarta la MODBUS PDU sin problemas. 0xFF se recomienda para la "Unidad de Referencia" como valor no significativo.

Nota: El valor 0 también se acepta para comunicar directamente a un MODBUS / TCP dispositivo.

#### **3.4.1.3 Confirmación de un proceso Modbus**

Cuando una respuesta se recibe en una conexión TCP, el Identificador de transacción que lleva en la cabecera MBAP se utiliza para asociar la respuesta con la solicitud original y la envía en una conexión TCP:

- Si la operación de identificación no se refiere a ninguna transacción en espera de MODBUS, la respuesta debe ser desechada.
- Si la operación de identificación se refiere a una transacción pendiente de MODBUS, la respuesta debe ser analizada con el fin de enviar una confirmación de MODBUS a la aplicación del usuario (confirmaciones positivas o negativas).

Analizar la respuesta consiste en la verificación de la cabecera MBAP y la respuesta MODBUS PDU:

- **Cabecera MBAP**

Después de la verificación del Protocolo de identificación que debe ser 0x0000, la longitud da el tamaño de la respuesta de MODBUS.

Si la respuesta viene de un dispositivo servidor MODBUS directamente conectado a la red TCP/IP, la identificación de conexión TCP es suficiente para identificar inequívocamente el servidor remoto. Por lo tanto, la Unidad de identificación llevada en la cabecera MBAP no es significativa (valor 0xFF) y debe ser desechada.

Si el servidor remoto se conecta a una sub – red de linea serial y la respuesta viene de un puente, un router o una puerta de enlace, entonces la unidad de identificación (valor!= 0xFF) identifica el servidor remoto MODBUS que en un primer momento había enviado la respuesta.

#### ▪ **Respuesta MODBUS PDU**

El código de función debe ser verificado y el formato de la respuesta MODBUS analizado de acuerdo con el Protocolo de Aplicación MODBUS:

- Si el código de función es el mismo que el utilizado en la solicitud, y si el formato de respuesta es correcto, entonces la respuesta MODBUS se da a la aplicación de usuario como una confirmación positiva.
- Si el código de función es un código de excepción de MODBUS (código de función + 80H), la respuesta de excepción Modbus se da a solicitud del usuario como un hecho positivo de confirmación.
- Si el código de función es diferente de la utilizada en la solicitud (= no espera código de función), o si el formato de la respuesta es incorrecta, entonces un error es señalado a la aplicación de usuario utilizando una confirmación negativa.

Nota: Una confirmación positiva es una confirmación de que el comando se recibió y respondió por el servidor. No implica que el servidor fue capaz de actuar con éxito en el comando (la falla de actuar con éxito en el comando se indica mediante la respuesta de excepción de MODBUS).

El siguiente diagrama de actividades describe más profundamente la fase de confirmación de procesamiento.

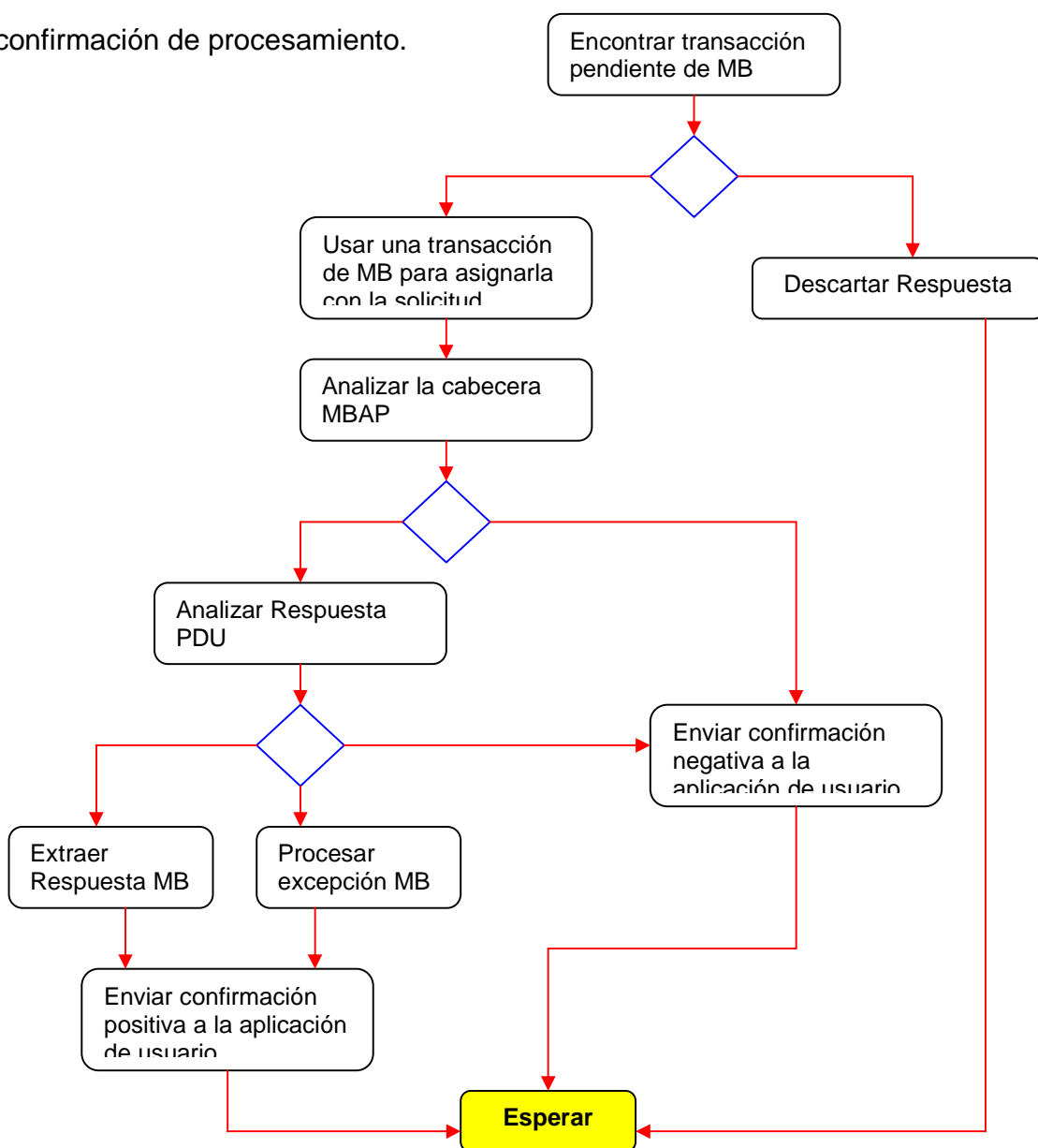


Figura 16. Diagrama de actividad del confirmación del proceso Modbus

#### **3.4.1.4 Gestión del tiempo de espera**

Se ha evitado deliberadamente el pliego de tiempo de respuesta necesario para una transacción sobre MODBUS/TCP.

Esto se debe a que MODBUS/TCP se espera que sea utilizado en la más amplia variedad de situaciones de comunicación, a la expectativa de escáneres de I/O con tiempos de sub – milisegundos a larga distancia con enlaces de radio y retrasos de varios segundos.

Desde la perspectiva del cliente, el tiempo debe tener en cuenta los retrasos del transporte a través de la red, para determinar un tiempo de respuesta “razonable”. Tales demoras de transporte podrían ser milisegundos para una conmutación Ethernet, o cientos de milisegundos para una conexión de área de red amplia.

A su vez, cualquier tiempo de espera utilizado en un cliente para iniciar una aplicación debe ser más grande que el tiempo de respuesta "razonable" máximo esperado. Si esto no es seguido, hay un exceso de congestión en el dispositivo de destino o en la red, que a su vez puede causar más errores. Esta es una característica que siempre debe ser evitada.

Así que en la práctica, los tiempos de espera del cliente utilizados en aplicaciones de alto rendimiento siempre son susceptibles de ser algo dependiendo de la topología de la red y el desempeño del cliente. Las aplicaciones que no están en el tiempo crítico, a menudo pueden salir de los valores de tiempos de espera a los valores normales por defecto de TCP, que presentará una falla de comunicación después de varios segundos en la mayoría de las plataformas.

### 3.4.2 Servidor Modbus

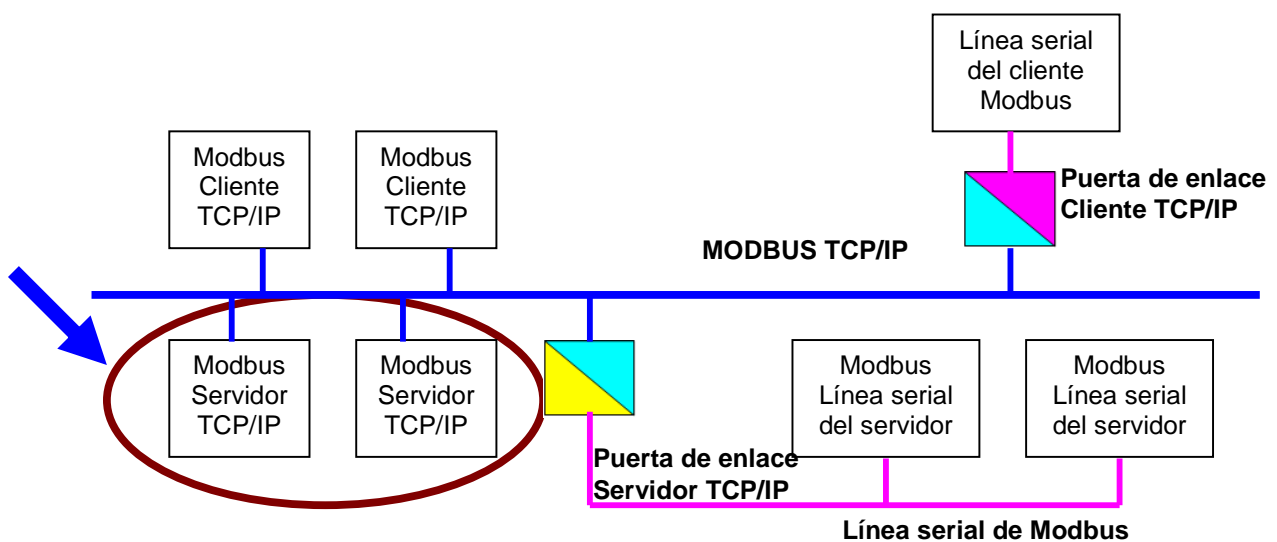


Figura 17. Unidad del servidor de Modbus

La función de un servidor MODBUS es proporcionar acceso a la aplicación de objetos y servicios a los clientes remotos MODBUS.

Distintos tipos de acceso pueden ser proporcionados en función de la aplicación de usuario:

- Acceso simple como obtener y ajustar atributos de los objetos de aplicación.
- Accesos avanzados con el fin de activar servicios de aplicación específicos.

El servidor Modbus tiene:

- Para asignar objetos en aplicación legible y escribible de Modbus, con el fin de obtener y ajustar atributos de los objetos de aplicación.
- Para proporcionar una forma de activar los servicios de aplicación a objetos.

En el tiempo de ejecución el servidor de Modbus tiene que analizar una solicitud recibida, procesar las medidas necesarias, y el devolver una respuesta Modbus.

Nota informativa: Los objetos de aplicación y los servicios de la interfaz Backend obtienen los datos solicitados basados en el código de función, y el usuario es el responsable.



### 3.4.2.1 Diseño del servidor Modbus

El diseño del servidor de MODBUS depende de dos factores:

- El tipo de acceso a los objetos de aplicación (accesos simples a los atributos o accesos avanzados a los servicios).
- El tipo de interacción entre el servidor MODBUS y la aplicación de usuario (sincrónica o asincrónica).

El siguiente diagrama de actividades describe los principales tratamientos que son procesados por el servidor para obtener una solicitud de MODBUS de la gestión TCP, luego analiza la solicitud, procesa las medidas necesarias, y devuelve una respuesta MODBUS.

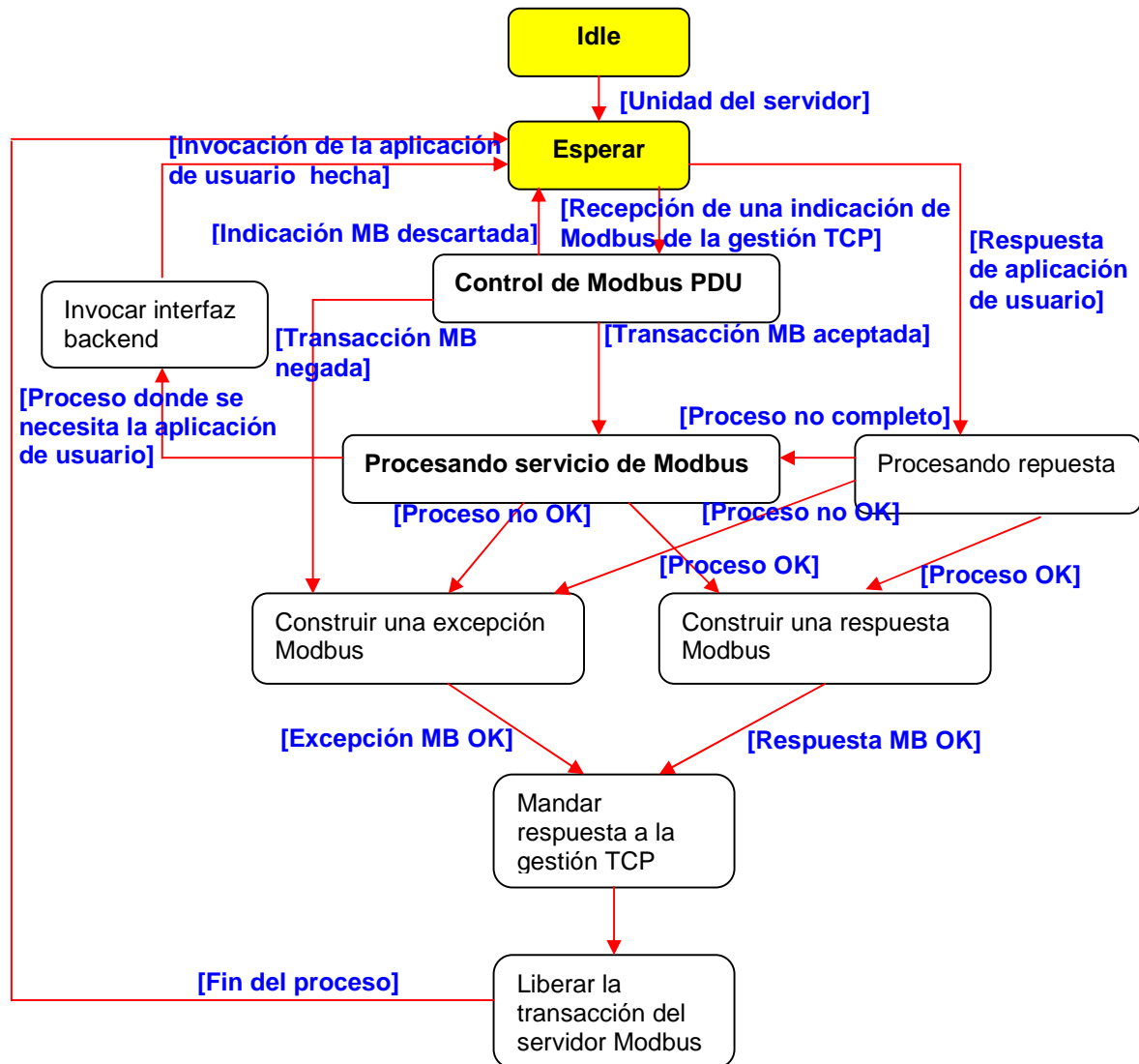


Figura 18. Diagrama de actividad de la indicación del proceso Modbus

El diagrama de actividad anterior nos explica que:

- Algunos servicios pueden ser inmediatamente procesados por el mismo servidor MODBUS, sin la interacción directa con el software de aplicación.

- Algunos servicios pueden requerir también una interacción explícita con el software de aplicación que van a ser procesados.
  
- Algunos otros servicios avanzados requieren la invocación de una interfaz específica llamada MODBUS Back End servicio. Por ejemplo, un servicio de aplicación de usuario de puede ser activado usando una secuencia de varias operaciones de solicitud/respuesta de Modbus realizadas de acuerdo al protocolo del nivel de aplicación del usuario.

El servicio de Back End es responsable del procesamiento correcto de todas las operaciones individuales de Modbus a fin de ejecutar el servicio de aplicaciones del usuario mundial.

El servidor Modbus puede aceptar para servir simultáneamente varias solicitudes Modbus.

El número máximo de solicitudes simultáneas MODBUS el servidor puede aceptar es una de las principales características de un servidor MODBUS.

Este número depende del servidor de diseño y su capacidad de procesamiento y de memoria. Esta aplicación se llama parámetro "Número máximo de operaciones del servidor" y debe ser descrita como una de las características del servidor Modbus. Puede tener un valor de entre 1 y 16 dependiendo de las capacidades del dispositivo.

El comportamiento y el desempeño del servidor Modbus se ven significativamente afectados por el parámetro "Número máximo de operaciones". Particularmente, es importante señalar que el número de transacciones concurrentes Modbus gestionados pueden afectar el tiempo de respuesta de una solicitud MODBUS por el servidor.

### 3.4.2.2 Comprobación de Modbus PDU

El siguiente diagrama describe la actividad de comprobación de Modbus PDU.

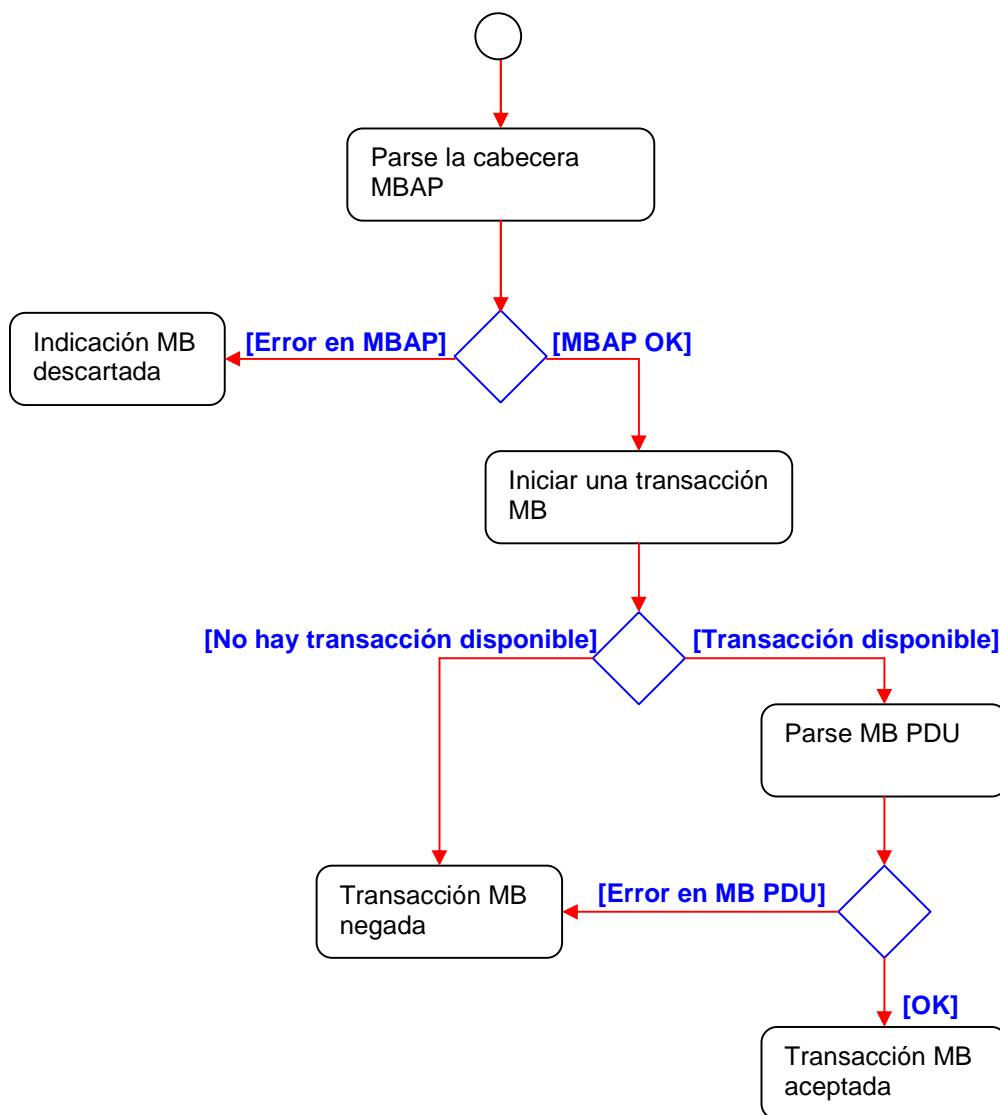


Figura 19. Diagrama de actividad del comprobación de Modbus PDU

La función de comprobación consiste en analizar primero la cabecera MBAP. El campo de identificación del protocolo se ha de comprobar:

- Si es diferente del tipo de protocolo Modbus, la indicación es simplemente descartada.
- Si es correcta (= tipo de protocolo Modbus; de valor 0x00), una transacción es MODBUS instanciada.

El número máximo de transacciones Modbus que el servidor puede ejemplificar esta definido por el parámetro "Numero máximo de transacciones" (un sistema o un parámetro de configuración).

En caso de que no hayan más operaciones disponibles, el servidor crea una respuesta de excepción Modbus (código de Excepción 6: Servidor Ocupado).

Si una transacción Modbus está disponible, es inicializada con el fin de memorizar la siguiente información:

- El identificador de conexión TCP usado para enviar la indicación (dada por la gestión TCP).

- La identificación de transacción Modbus (que figura en la cabecera de MBAP).
  
- La Unidad de identificación (que figura en la cabecera de MBAP).

Entonces el Modbus PDU se analiza. El código de función es primero controlado:

- En caso de invalidez una respuesta de excepción Modbus se construye (Código de excepción 1: Función inválida).
  
- Si el código de función es aceptado, el servidor inicia la actividad de "procesamiento de servicios de Modbus".

### 3.4.2.3 Procesamiento del servicio Modbus

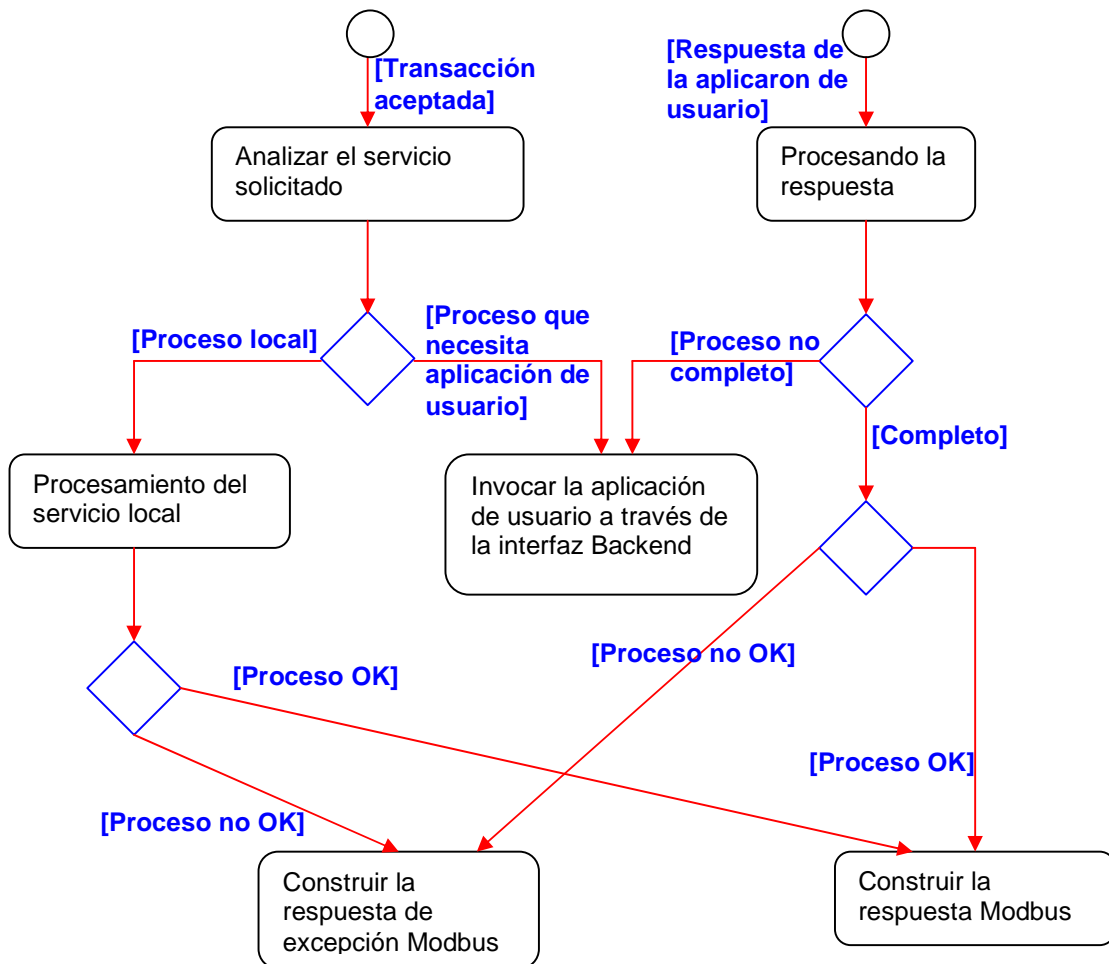


Figura 20. Diagrama de actividad del proceso de servicio Modbus

El procesamiento de los servicios Modbus puede hacerse de diferentes maneras, dependiendo del software del dispositivo y la arquitectura del hardware, tal como se describe en los ejemplos a continuación:



- Dentro de un dispositivo compacto o una arquitectura mono-hilo donde el servidor Modbus puede acceder directamente a los datos de la aplicación de usuario, el servicio requerido puede ser procesado "localmente" por el propio servidor sin invocar el servicio Back End. El proceso es hecho de acuerdo a las especificaciones del protocolo de aplicación Modbus. En el caso de un error, una respuesta de excepción Modbus se construye.
- Dentro de un dispositivo modular multi – procesador o una arquitectura de multi – hilo donde las "capas de comunicación" y la "capa de aplicación de usuario" son 2 entidades distintas, algunos servicios triviales pueden ser procesados completamente por la entidad de comunicación mientras que otros pueden requerir una cooperación con la entidad de aplicación de usuario utilizando el servicio de Back End.

Para interactuar con la aplicación de usuario, el servicio Backend debe aplicar todos los mecanismos adecuados a fin de manejar las operaciones de solicitud del usuario y para gestionar correctamente la aplicación del usuario y las respuestas.

#### **3.4.2.4 Interfaz de Aplicación de usuario (Backend Interface)**

Varias estrategias pueden ejecutarse en el servicio Backend de Modbus para lograr su trabajo aunque no sean equivalentes en términos de rendimiento de la red del usuario, la interfaz de uso del ancho de banda, tiempo de respuesta, o incluso el diseño de la carga de trabajo.

El servicio Backend de Modbus utilizará la interfaz adecuada para el usuario:

- Cualquier interfaz física basada en un enlace serial, o un esquema de puerto dual, o una línea simple de E/S, o una interfaz lógica basada en los servicios de mensajería proporcionada por un sistema operativo.
  
- La interfaz para el usuario puede ser síncrona o asíncrona.

El servicio Backend de Modbus también utilizara el patrón de diseño apropiado para conseguir y ajustar el conjunto de objetos o atributos para activar los servicios. En algunos casos, un simple "modelo de puerta de enlace" será suficiente. En algunos otros casos, el diseñador tendrá que aplicar un "patrón proxy" con una estrategia de almacenamiento en caché, de un simple intercambio de historia a un mecanismo de replicación mas sofisticado.

El servicio Backend de Modbus tiene la responsabilidad de aplicar la transcripción del protocolo con el fin de interactuar con la aplicación del usuario.

Por lo tanto, puede tener que poner en marcha mecanismos para la fragmentación y reconstrucción de paquetes, garantizar la consistencia de los datos, y la sincronización cuando sea necesario.

### 3.4.2.5 Construcción de la respuesta Modbus

Una vez que la solicitud ha sido procesada, el servidor Modbus tiene que construir la respuesta adecuada utilizando la transacción adecuada y tiene que enviarla a la componente de gestión de TCP.

Dependiendo del resultado del procesamiento, se pueden construir dos tipos de respuesta:

- Una respuesta positiva:
  - El código de función de respuesta = El código de función de la solicitud
  
- Una respuesta de excepción:
  - El objetivo es proporcionar al cliente la información pertinente sobre el error detectado durante el proceso.
  
  - El código de función de la respuesta = el código de función de la solicitud + 0x80.
  
  - La excepción código se suministra para indicar el motivo del error.

<b>Código de Excepción</b>	<b>Nombre Modbus</b>	<b>Comentarios</b>
01	Código de función ilegal	El código de función es desconocido por el servidor.
02	Dirección de datos ilegal	Dependiente de la solicitud.
03	Valor de datos ilegal	Dependiente de la solicitud.
04	Falla del servidor	El servidor fallo durante la ejecución.
05	Reconocimiento	El servidor acepta la invocación de servicios, pero el servicio requiere un tiempo relativamente largo para ejecutar. El servidor, por lo tanto, sólo ofrece un reconocimiento de la invocación del servicio recibido.
06	Servidor ocupado	El servidor no ha podido aceptar la Solicitud MB PDU. La aplicación cliente tiene la responsabilidad de decidir si y cuándo debe volver a enviar la solicitud.
0A	Problemas de puerta de enlace	Los caminos de la puerta de enlace no están disponibles.
0B	Problemas de puerta de enlace	El dispositivo objetivo fallo a la respuesta. La puerta de enlace genera esta excepción.

**Tabla 6. Códigos de excepción**

La respuesta Modbus PDU debe ser prefijado con la cabecera MBAP la cual se construye usando los datos memorizados en la transacción.

- **Identificador de Unidad**

El identificador de unidad es copiado ya que se dio dentro de la solicitud Modbus recibida y memorizado en la transacción.

- **Longitud**

El servidor calcula el tamaño de Modbus PDU más el byte del Identificador de unidad. Este valor es ajustado en el campo "Longitud".

- **Identificador de Protocolo**

El campo de identificación de protocolo se ajusta a 0x0000 (protocolo Modbus), ya que se dio dentro de la solicitud recibida de Modbus.

- **Identificador de transacción**

El valor de este campo es ajustado en "Identificación de transacción" y se asocia con la solicitud original y es memorizado en la transacción.

Entonces la respuesta Modbus debe ser devuelto al cliente correcto utilizando la conexión TCP memorizada en la transacción. Cuando la respuesta se envía, la operación debe ser libre.

## CONCLUSIONES

- Ante los avances en los procesos industriales y el gran auge de las comunicaciones el protocolo Modbus es el más utilizado debido a su aceptación y disponibilidad para la conexión de dispositivos electrónicos industriales en donde permite el control de una amplia red de dispositivos y variables.
  
- El servicio de mensajería de MODBUS provee una comunicación cliente/servidor entre dispositivos conectados en una red Ethernet TCP/IP y provee servicios que son usados para intercambiar información en tiempo real.
  
- En la implementación de Modbus TCP/IP con el software Labview se demostró que sistemas de control de procesos ya instalados pueden ser supervisados y controlados desde una red TCP/IP como Internet.
  
- Se demostró la interoperabilidad de la red implementada, de forma que desde clientes Modbus TCP/IP fue posible leer y escribir registros y datos discretos sobre los diversos elementos que conforman la red.

## BIBLIOGRAFÍA

- <http://www.modbus.org>
- Modbus Messaging Implementation Guide v1.doc
- BusWorks® 900EN Series 10/100M Industrial Ethernet I/O Modules w/  
Modbus - INTRODUCTION TO MODBUS TCP/IP
- <http://www.acromag.com>

## ANEXO

### PRACTICA MODBUS CON LABVIEW

#### 1. Objetivo General

- Establecer una comunicación entre dos computadores con el protocolo Modbus TCP/IP mediante LABVIEW, utilizando un cable cruzado que garantiza la conexión física entre ambos PC`s.

#### 2. Objetivos Específicos

- Enviar comandos o registros desde el maestro y el esclavo y comprobar que estos sean leídos por su destinatario.
- Implementar programas de comunicación que utilicen Modbus TCP/IP con respecto a Labview.

#### 3. PRIMERA PARTE – Instalación del Software y Librerías

Instalamos el software (LABVIEW) En los dos computadores que vamos a utilizar, luego procedemos a instalar las librerías, que se obtuvieron previamente de la página Web:

[http://sine.ni.com/apps/we/niepd\\_web\\_display.display\\_epd4?p\\_guid=F1582737BACF5CA8E0340003BA7CCD71?subject=Libreria de labview](http://sine.ni.com/apps/we/niepd_web_display.display_epd4?p_guid=F1582737BACF5CA8E0340003BA7CCD71?subject=Libreria de labview)

#### 4. SEGUNDA PARTE – Hacer el cable de conexión

El cable cruzado es utilizado para conectar dos PCs directamente o equipos activos entre si, como hub con hub, con switch, router, etc.

Un cable cruzado es aquel donde en los extremos la configuración es diferente. El cable cruzado, como su nombre lo dice, cruza las terminales de transmisión de un lado para que llegue a recepción del otro, y la recepción del origen a transmisión del final.

Las redes de computadores no utilizan los 4 pares (8 cables) en su totalidad, utilizan solamente 4 cables: 2 para transmitir y 2 para recibir.



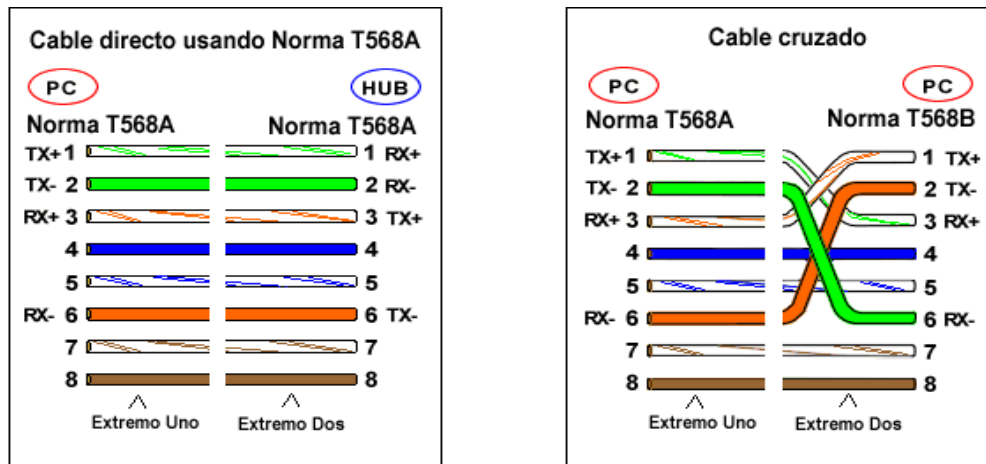


Figura 1. Diagrama del cable cruzado

## 5. TERCERA PARTE – Implementación de MODBUS con LABVIEW

### 5.1 REQUERIMIENTOS DE LA PRÁCTICA

Se desea comunicar dos computadores mediante la herramienta LABVIEW. Ya instalados los programas en ambos computadores y las librerías, procedemos a configurar los computadores, tomamos un computador como maestro que es al que se le va a poner la dirección IP del esclavo, y el otro lo configuramos como esclavo (no es necesario poner la dirección IP del maestro).

## 5.2 Configuración del maestro

Una vez en la página de trabajo de LabView hacemos clic con el botón derecho del Mouse sobre dicha página y aparece la ventana de “Todas las funciones” y hacemos click en el icono de “Todas las funciones”.

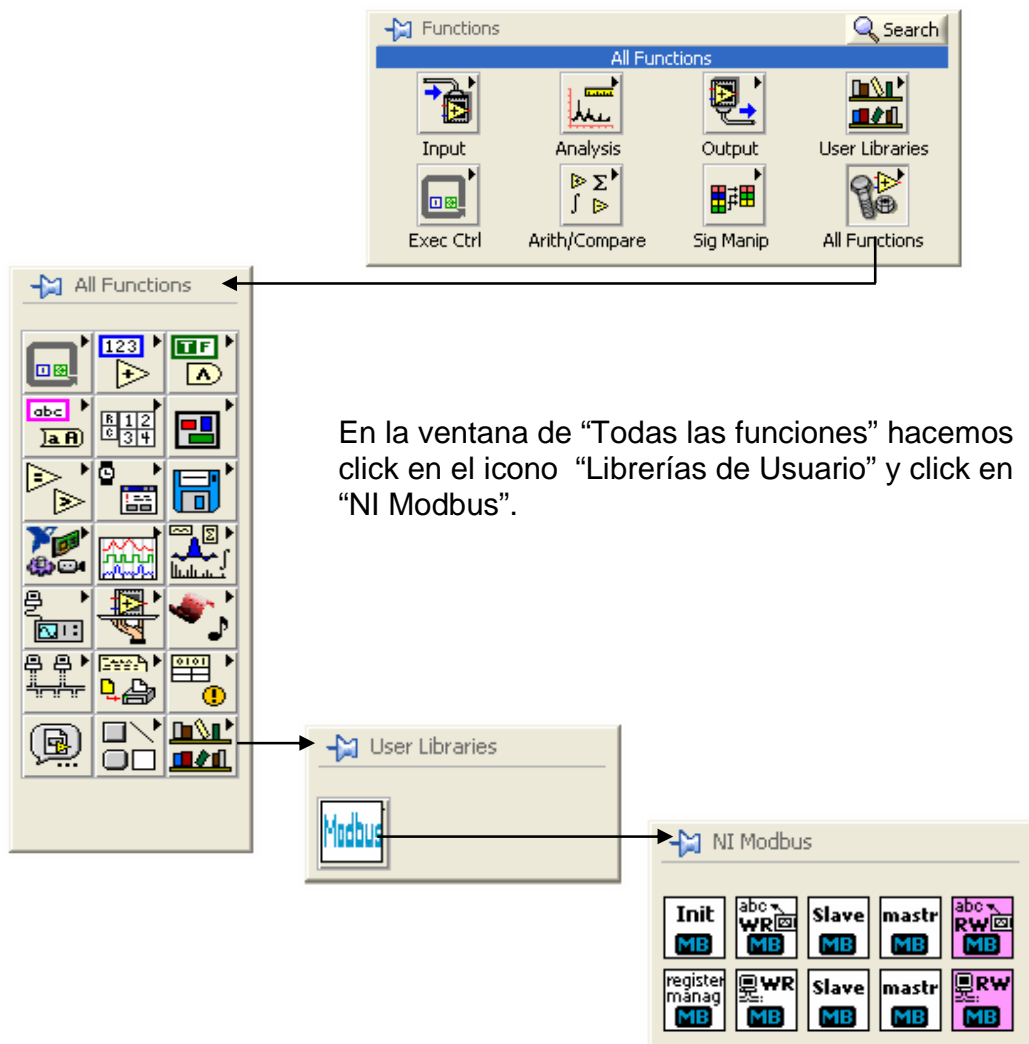


Figura 2

Una vez se llega a la ventana de “NI Modbus”, se selecciona “MB Ethernet Example Master.vi”

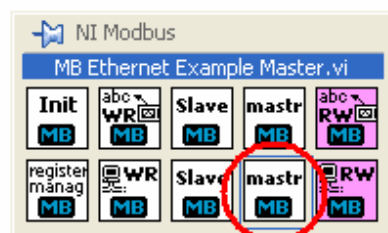


Figura 3

Seguidamente con un click se pega el icono del maestro seleccionado en la página principal de Labview.

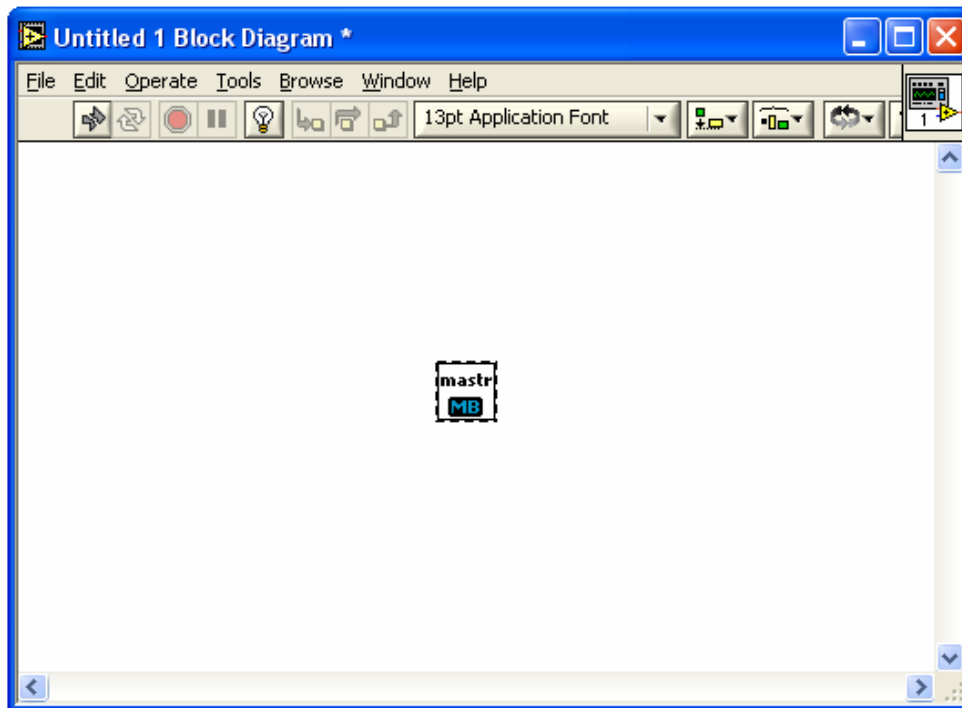


Figura 4

Se hace doble click en dicho icono y obtenemos el panel frontal para su configuración.

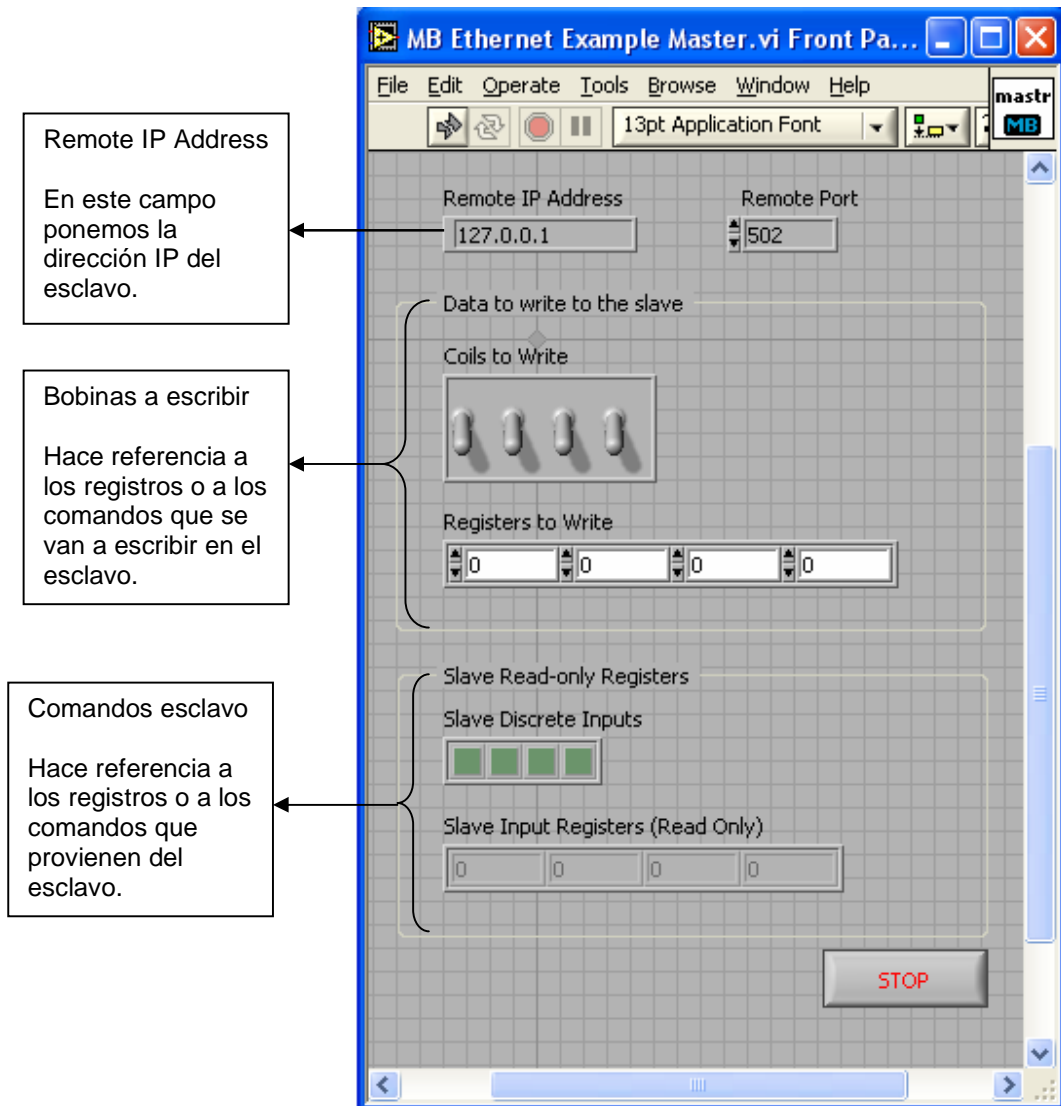


Figura 5. Panel Frontal del Maestro

Haciendo doble click en el panel frontal aparece el diagrama de bloques, hasta este punto todavía no es posible la conexión con el esclavo y para esto hay que crear la cabecera de MBAP que se encuentra en "MB Ethernet Master Quero.vi" y se conecta como indica la figura 6.

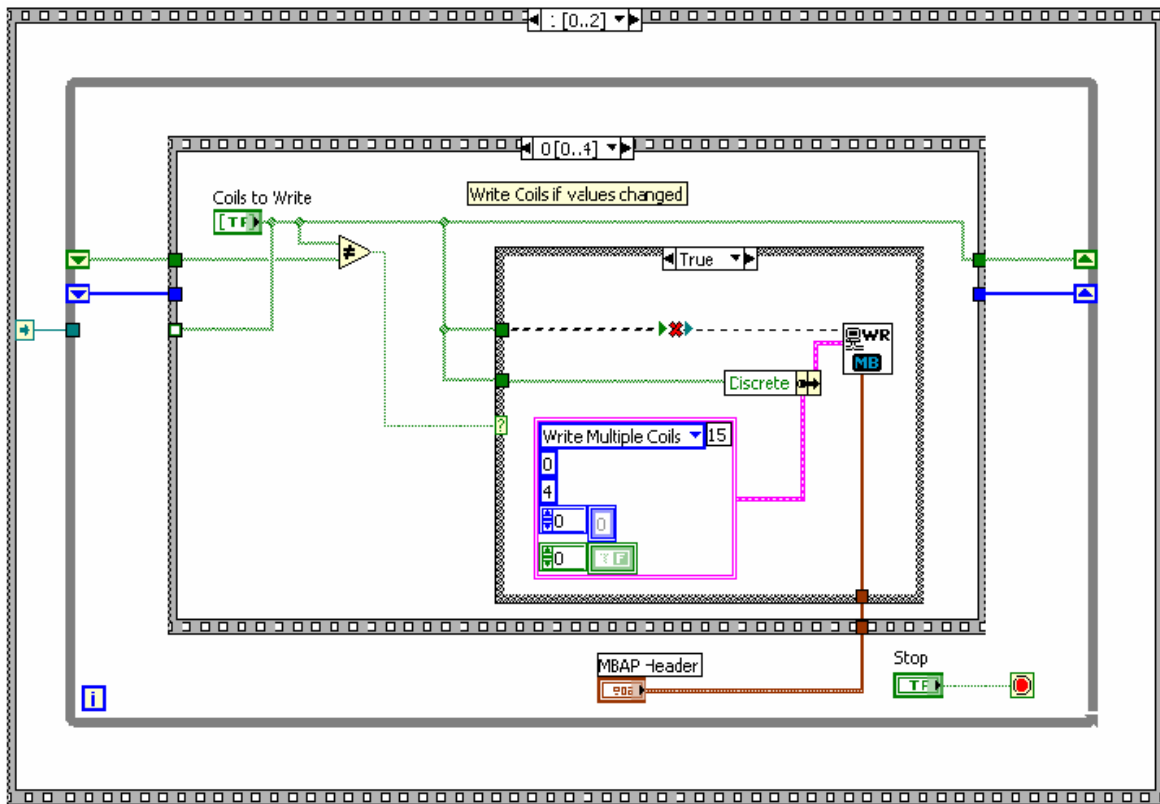


Figura 6. Conexión de Cabecera

Inmediatamente conectamos la cabecera MBAP, en el panel frontal aparece como funciones para la conexión con el esclavo.

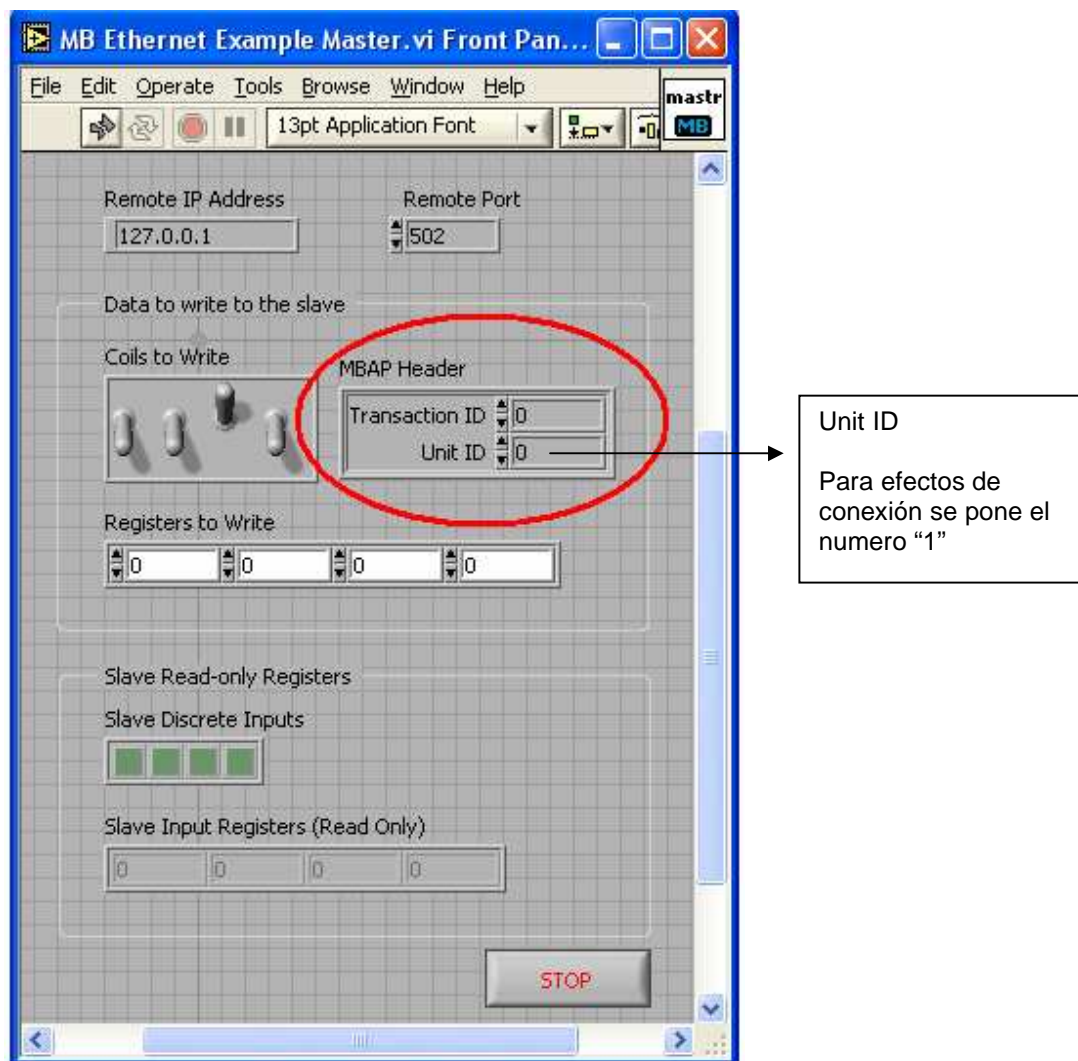


Figura 7

Ya en estas instancias el Maestro esta listo para conectarse.

### 5.3 Configuración del Esclavo

Al igual que con el maestro una vez que se llega a la ventana de “NI Modbus”, seleccionamos “MB Ethernet Example Slave.vi”

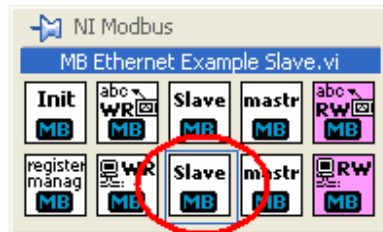


Figura 8

Igualmente nos posicionamos en la página principal de Labview y con un click lo pegamos e inmediatamente hacemos doble click en dicho icono y obtenemos el panel frontal para el esclavo.

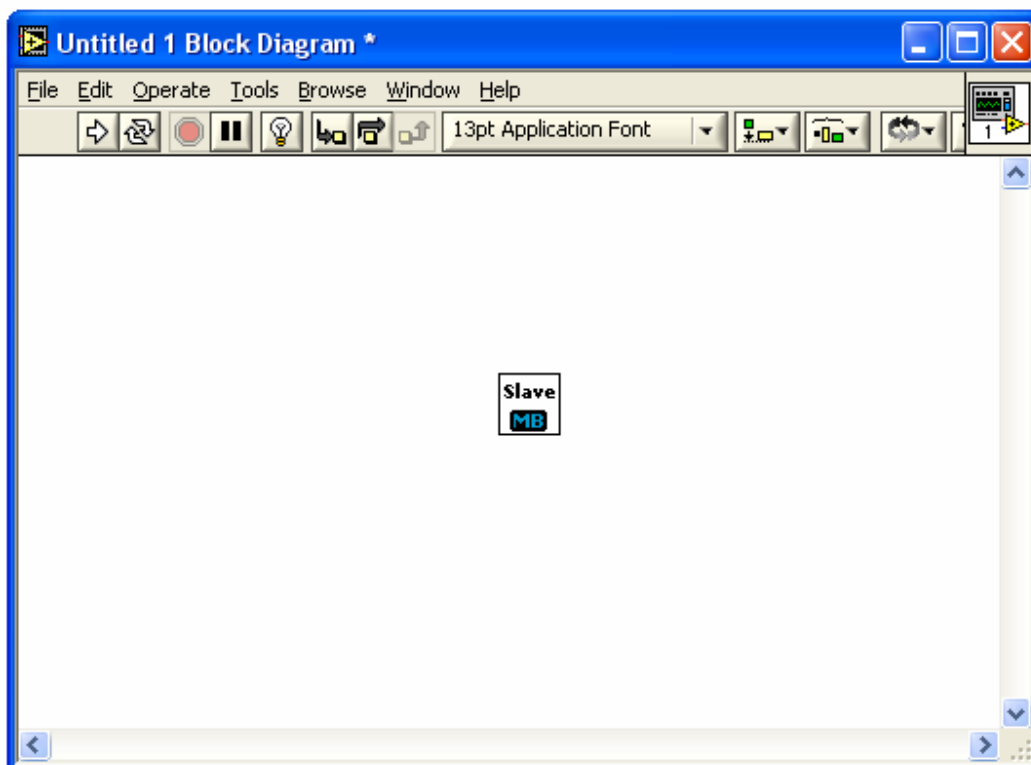


Figura 9

En el panel frontal del esclavo, a diferencia del maestro no hay que cambiar nada, ya esta listo para comunicarse y enviar y recibir comandos.

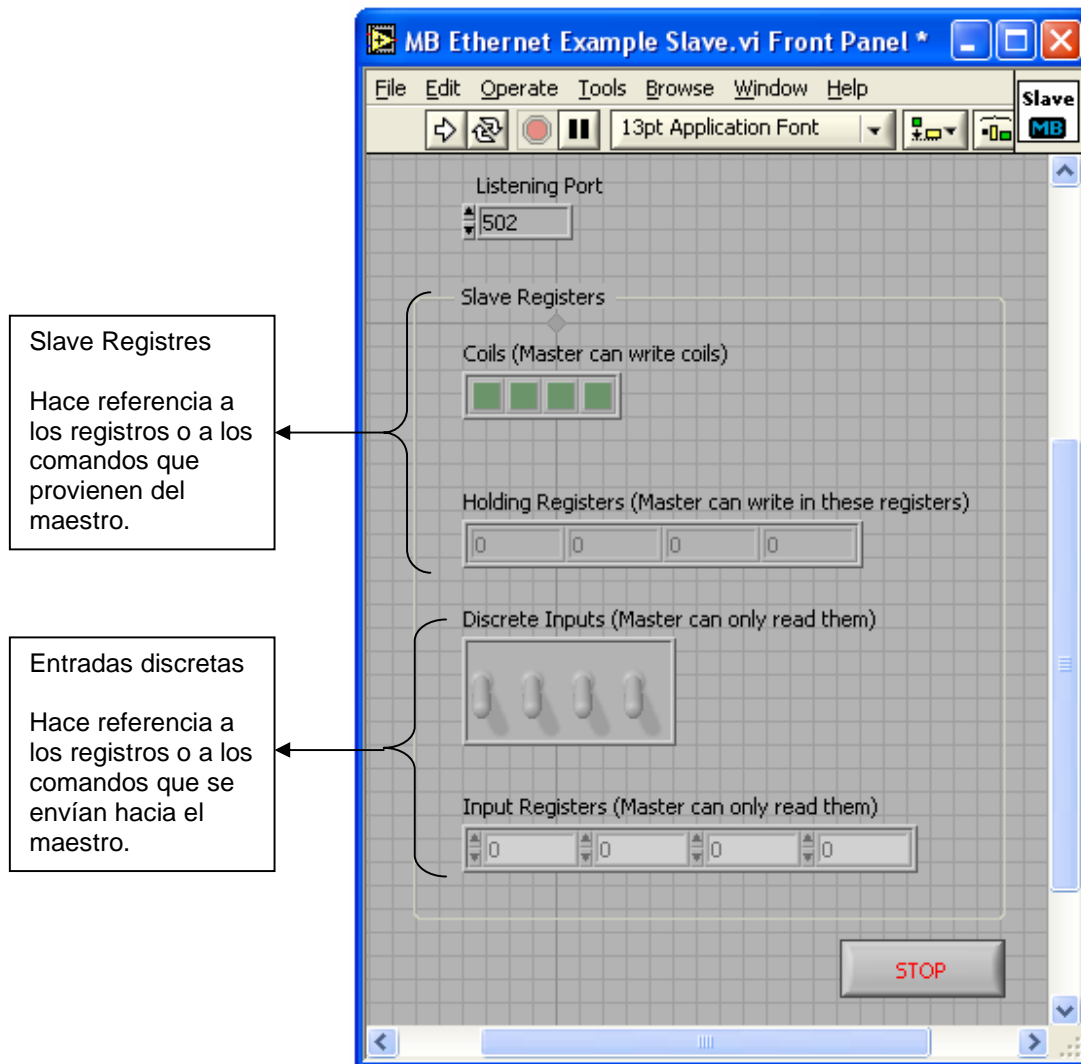


Figura 10. Panel Frontal del Esclavo