

UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS

FACULTAD DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE AUTOMÁTICA Y SISTEMAS AUTOMATIZADOS



TRABAJO DE DIPLOMA RED OPC CON FINES DOCENTES

AUTOR:
RAFAEL PÉREZ CONTRERAS
E-MAIL: RAFAELP0@YAHOO.COM

TUTOR:
DR. ROBERTO BALLESTEROS HORTA
JEFE DPTO. AUTOMÁTICA
E-MAIL: RBALL@FIE.UCLV.EDU.CU

SANTA CLARA, 2004
“AÑO DEL 45 ANIVERSARIO DEL TRIUNFO DE LA REVOLUCIÓN”

*“Es mejor ser osadamente decidido y
correr el riesgo de equivocarse, que
sopesar mil veces las cosas y tomar la
mejor decisión demasiado tarde”*

Marilyn Moats

*Gracias a todos los educadores de quienes
he aprendido que no bastan los libros: my
mom, maría del c, otra maría, que no
recuerdo bien el nombre y
desafortunadamente no sigue educando,
pero que también tiene su pedacito de
tesis, alain, mi grupo anterior, la
universidad, angela, los del pre, incluso
rosa aidée, de la primaria...*

A mi familia toda, y en especial a mi madre, por la paciencia y a ía A mis tias lindas evy, made, ara...

A mi mamá maruchi y a mi mamá dámasa, porque no pierden tiempo de halarme las orejas, voy a tener que pensar que soy un poco regado!

A mis primitos chiquitos emilyjoe.

A mi hermano, incluso cuado tengo ganas de cogerlo por el cuello.

A nilylili, porque esta es su tesis también.

A mis amigos de siempre y a las cuevas, ríos, montañas y mares que nos han hecho familia: yami, el pequeño, tere, felo, lina, krlitos, myrita, yuri, susy, yandoso, alejoymartica, que se casan pronto, kenia, el rolo, ludmi, el negro, yao, julito, greydel, el liba, el seboruco de nuevitas...

A algunos que pasaron y no están, pero que estuvieron.

Gracias a todos ellos, a que gota a gota se llena... y a que la botella a medias está llena es que he llagado a este final del comienzo.

TAREA TÉCNICA

- Análisis y revisión bibliográfica sobre el tema de OPC
- Selección de servidor y cliente OPC para la implementación de una red
- Configuración de los componentes implicados en la red
- Elaboración del informe final

RESUMEN

Dado el auge del protocolo OPC, en cuanto a transferencia de datos de forma transparente y normalizada, salvando las barreras de las diferentes compañías proveedoras de *hardware* y *software* orientados a la industria, contexto administrativo y aplicaciones de oficina; se ha concebido el presente trabajo, de forma tal que constituya la herramienta para llevar este protocolo a los laboratorios de nuestra facultad.

Para ello, se ha confeccionado una síntesis teórica de lo referente a OPC, en cuanto a concepción y alcance. Se ha abordado además el tema de la seguridad, la cual debe tenerse siempre en cuenta cuando se afronta el tema de la transferencia de datos.

Para facilitar la materialización de esta teoría, se ha expuesto una práctica sencilla, de configuración de servidor y cliente OPC para establecer una red y finalmente se ha brindado un ejemplo práctico de transmisión de variables de interés industrial, como son el voltaje de un tacómetro conectado a un servomotor y la temperatura de un local.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1 PROTOCOLO DE COMUNICACIÓN OPC	5
1.1 MISIÓN DE OPC.....	5
1.2 BENEFICIOS DEL USO DE OPC.....	6
1.3 ANALOGÍA DE OPC CON EL DRIVER DE IMPRESORAS EN WINDOWS.....	7
1.4 LA FUNDACIÓN OPC	8
1.5 LAS ESPECIFICACIONES OPC.....	9
1.6 ESTÁNDARES SOBRE LOS QUE SE APOYA OPC	10
1.7 INTERFACES OPC	14
1.8 ARQUITECTURA DE DISTINTOS TIPOS DE SERVIDORES OPC	15
1.8.1 Servidor OPC de Acceso a Datos	16
1.8.2 Servidor OPC de Alarmas y Eventos	17
1.8.3 Servidor OPC de Datos Históricos.....	18
1.8.4 Servidor OPC por Lotes.....	21
1.8.5 Servidor OPC de Acceso a Datos XML.....	22
1.9 DETECCIÓN DE SERVIDORES OPC INSTALADOS	23
1.10 SEGURIDAD EN OPC.....	24
1.10.1 Modelo de Seguridad NT.....	24
1.10.2 Aplicación del Modelo de Referencia NT a OPC.....	25
1.10.3 Niveles de Seguridad.....	26
1.11 CONSIDERACIONES GENERALES.....	27
CAPÍTULO 2 INSTALACIÓN Y CONFIGURACIÓN DE UNA RED CLIENTE/SERVIDOR OPC	29
2.1 SELECCIÓN DE LAS HERRAMIENTAS A UTILIZAR	29
2.2 IMPLEMENTACIÓN DE UNA RED OPC SENCILLA, A MANERA DE INTRODUCCIÓN	31
2.2.1 Instalación del servidor OPC.....	32

ÍNDICE

2.2.2 <i>Microsoft Excel como servidor DDE</i>	33
2.2.3 <i>Configuración del servidor OPC</i>	34
2.2.4 <i>Configuración de la seguridad básica para OPC en W2K</i>	36
2.2.5 <i>Configuración del cliente OPC</i>	42
2.3 IMPLEMENTACIÓN DE UNA RED OPC PRÁCTICA.....	46
2.3.1 <i>Implementación de la red</i>	47
2.4 CONSIDERACIONES GENERALES	49
CONCLUSIONES	51
RECOMENDACIONES	52
REFERENCIAS BIBLIOGRÁFICA	53
ANEXO I APLICACIONES SERVIDOR DDE CONFECCIONADAS EN LABVIEW	56
ANEXO II LABVIEW EN SÍNTESIS	60

INTRODUCCIÓN

El amplio desarrollo tecnológico experimentado en nuestra época ha traído consigo el auge de la informatización y el acceso a ella como una necesidad, ya sea con fines recreativos, educacionales, formativos, profesionales, industriales e incluso para la toma de decisiones comerciales o gubernamentales. Satisfacer esta demanda cualitativa y cuantitativa se ha convertido en un gran reto para los implicados.

El mundo de la informatización, actualmente esta dividido en tres grandes áreas [5]:

- Automatización
- MES¹
- ERP²

Entre estas áreas se precisa de un amplio flujo de información; que hasta hace un tiempo se había logrado mayormente con la intervención de adaptadores específicos a un costo elevado, en tiempo y esfuerzo.

Dado que la competencia en el sector informático ha sido desde siempre muy fuerte, se ha hecho necesaria una reducción en los costos y un aumento de la productividad. Es para ello que se precisa de una integración entre estas tres áreas.

Es Internet, con su desarrollo vertiginoso, la que se considera que sea la encargada de esta integración. Con este objetivo, Microsoft ha definido la Arquitectura Distribuida de Internet (DNA³). Esta estrategia cubre cinco tecnologías básicas:

- Navegador de Internet
- DHTML⁴
- Controles ActiveX
- OPC⁵

¹ Management Execution Systems.

² Enterprise Resource Planning.

³ Distributed interNet Architecture

⁴ Dynamic HyperText Markup Language.

⁵ OLE for Process Control.

- XML⁶

Actualmente, muchos sistemas SCADA⁷ corren sobre el Navegador de Internet, usando DHTML para brindar al usuario la información en forma gráfica, a través de animaciones, video y sonido.

Los Controles ActiveX son usados en algunas aplicaciones de Internet, mientras que XML y OPC constituyen dos estándares abiertos. XML es altamente adaptable en cuanto a plataformas. OPC, provee un intercambio flexible, robusto y simple de datos.[5]

Nuestro país, que nunca ha estado ajeno al acontecer mundial, en el V Congreso del Partido Comunista de Cuba, desarrollado en 1997, se planteó como política económica para la recuperación de la economía cubana:

“El empleo de técnicas modernas de dirección empresarial, adecuadas a nuestras características y basadas en las mejores y más avanzadas prácticas contemporáneas, así como el amplio uso de todas las posibilidades de las tecnologías y servicios de información y las telecomunicaciones, deben constituir prioridad del país a los fines de garantizar la mayor eficiencia en la gestión y los procesos productivos. Para apoyar estos objetivos se deberá desarrollar un amplio movimiento de calificación, desde la formación de estudiantes hasta la recalificación de cuadros de dirección empresariales y estatales y demás trabajadores en todas las instancias.” [34]

Y atendiendo a esto, se trazó como perspectiva que:

“El país debe encaminarse resueltamente a la modernización informática mediante un programa integral que involucre a las organizaciones que deben proveer los recursos materiales, financieros e intelectuales y a las entidades económicas, políticas y sociales que deben traducirlos en más y mejores productos y servicios. La industria de los servicios informáticos deberá asegurar la modernidad de su base técnica y organizativa, y la elevación constante del nivel científico-técnico de sus especialistas con vistas a garantizar esos propósitos.” [34]

⁶ Extended Markup Language.

⁷ Supervisory Control And Data Acquisition.

Actualmente el tema de la informatización se ha visto muy ligado a los programas de la Revolución.

El presente trabajo de diploma abarca al estudio de la tecnología OPC como un estándar en desarrollo para la transferencia de datos de forma normalizada. Algo de suma importancia en el mundo de la automatización. Se ha partido sobre la base de un trabajo de diploma del curso anterior [2].

Se tienen referencias del uso de este estándar en algunas empresas del país, tales como la fábrica de cemento de Nuevitas en Camagüey y el Centro de Ingeniería y Desarrollo para la Automatización Industrial de nuestra provincia⁸; el cual brinda, entre otros, servicios de automatización utilizando esta tecnología.

Dada la falta de familiarización con el protocolo de comunicación OPC en años terminales de la carrera Automática, así como en estudios de postgrado, nos hemos planteado la necesidad de trasladar a nuestras aulas un conocimiento profundo sobre la tecnología OPC, para llevar a nuestras empresas a un personal cada vez más preparado, que pueda asumir el reto de disminuir los costos, aumentar la producción y lograr un producto capaz de insertarse en el mercado.

Vista esta necesidad se plantea el problema de cómo diseñar una red cliente/servidor OPC que permita el conocimiento de esta tecnología, con la finalidad de elevar la calificación de los egresados de la carrera de automática.

Concebido este problema, se ha planteado el objetivo de diseñar e implementar una red cliente-servidor OPC para su utilización como material de estudio en la formación del estudiantado, tanto el de pregrado como el de postgrado, en la Facultad de Ingeniería Eléctrica. Se le da seguimiento con ello al desarrollo del conocimiento que en cuanto a este protocolo de comunicación es necesario, dado su auge y actualidad en los diferentes niveles de gestión industrial. Estos incluyen el nivel de gestión de campo, pasando por el de gestión de proceso, y llegando al de gestión de negocio.

⁸ CEDAI: Sucursal Villa Clara.

Para el cumplimiento del objetivo propuesto, se han planteado los siguientes objetivos específicos:

- Analizar la bibliografía relacionada con la tecnología OPC para introducir un basamento teórico.
- Seleccionar el servidor y cliente OPC para la implementación de la red con fines docentes.
- Configurar el servidor y el cliente OPC para obtener una comunicación satisfactoria usando el protocolo OPC.
- Probar la comunicación para verificar su eficacia.
- Confeccionar el informe final.

Teniendo en cuenta lo anteriormente expuesto se ha estructurado el informe del trabajo de diploma de la siguiente forma:

Un primer capítulo donde se hace un recorrido por todo lo referente al protocolo OPC. Se habla de su fundamento y los beneficios que aporta, se aclara el papel de OPC a través de una analogía, se exponen las especificaciones OPC, concebidas por La Fundación OPC; así como los objetos y las interfaces contenidas en estas. Se enumeran los diferentes tipos de servidores OPC existentes, se aborda el tema de la seguridad en OPC y la forma en que un cliente OPC puede conocer los servidores disponibles en la red.

Un segundo capítulo donde se seleccionan las herramientas a utilizar y se describe la forma de configurarlas, para lograr la comunicación entre los diferentes elementos que compondrán la red. Se plantea la forma de lograr consistencia en la transferencia de los datos, a través de la configuración de la seguridad y se exponen las pruebas de comunicación realizadas.

Capítulo 1 PROTOCOLO DE COMUNICACIÓN OPC

El protocolo OPC, siglas en inglés de OLE⁹ para el Control de Procesos, es una interfaz estándar basada en las tecnologías OLE/COM¹⁰ y DCOM¹¹ de Microsoft. Facilita el intercambio de datos en forma estandarizada y simple entre aplicaciones de control y automatización, entre dispositivos y sistemas de campo y entre aplicaciones administrativas y de oficina. Simplifica la interfaz entre componentes de automatización de distintos fabricantes, así como programas y aplicaciones tales como sistemas administrativos y de visualización [6].

OPC es el resultado de un conjunto de especificaciones, concebidas para normalizar el proceso de intercambio de datos entre aplicaciones orientadas al control, que definen un grupo de interfaces COM orientadas a facilitar la implementación de objetos en aplicaciones encaminadas a la esfera industrial.

1.1 MISIÓN DE OPC

En las industrias actuales ha proliferado el uso de sistemas que usualmente no trabajan de mutuo acuerdo. Hacer que los sistemas trabajen en conjunto es la necesidad más apremiante para los fabricantes. Este problema va más allá de la conectividad, los diversos sistemas operativos y los no tan abiertos sistemas abiertos que supuestamente deben facilitar la interconexión [33].

Una forma de resolver el problema, es mediante el desarrollo de *drivers* (Figura 1.1); pero esto trae algunos problemas intrínsecos, como son:

- Esfuerzo de duplicación: cada aplicación debe desarrollar un *driver* diferente para cada dispositivo al que necesite conectarse.
- Inconsistencias entre los *drivers* de los vendedores: existen características de ciertos dispositivos que no son soportadas por todos los desarrolladores de *drivers*.
- Cambios en las características de los dispositivos.

⁹ Object Linking and Embedding.

¹⁰ Component Object Model.

¹¹ Distributed Component Object Model.

- Conflictos de acceso: dos aplicaciones clientes no podrían acceder a un mismo dispositivo, de forma simultánea, cada una lo haría con un *driver* diferente.

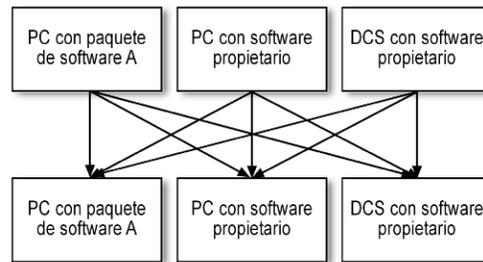


Figura 1.1 Interconexión de diferentes sistemas antes de OPC

La razón principal de este problema reside en la ausencia de una interfaz estándar entre los diferentes proveedores de sistemas, ya que cada cual brinda sus propias soluciones en cuanto a *hardware* y software, limitando la interconexión con sistemas de diferentes propietarios.

Es aquí donde se encuentra en OPC, una manera de resolver el problema (Figura 1.2).

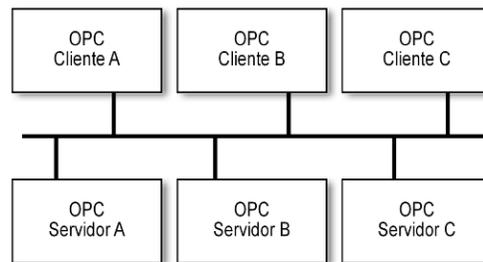


Figura 1.2 Interconexión de diferentes sistemas usando OPC

1.2 BENEFICIOS DEL USO DE OPC

OPC trae beneficios para todos los implicados:

- Los fabricantes de *hardware* solo tendrán que fabricar un conjunto de componentes de *software* para inicializar sus aplicaciones.
- Los desarrolladores de *software* no tendrán que volver a implementar *drivers* debido al cambio de características o incorporaciones en una nueva versión de *hardware*.
- Los compradores tendrán más opciones con las que implementar su sistema de producción integrado en clases e interfaces.

El potencial de innovación de OPC permite hacer realidad la tecnología *Plug & Play* para el mundo de la automatización [6] y posibilita el intercambio de datos a través de una interfaz común entre dispositivos de *hardware* y componentes de *software*, desarrollados por varios fabricantes. Con esto se hace posible la combinación de *hardware* de control programable y *software* sin la necesidad de *drivers* especiales, se facilita el acceso flexible desde productos de Microsoft Office a datos del proceso y representa un complemento importante en las actividades relacionadas con buses de campo, si se tiene en cuenta que el propósito principal de la estandarización en el sector de los buses de campo es la transferencia de datos de forma rápida y fiable.

OPC estandariza la comunicación de tal modo que cualquier servidor OPC y cualquier aplicación OPC pueden trabajar juntos sin ningún problema.

1.3 ANALOGÍA DE OPC CON EL DRIVER DE IMPRESORAS EN WINDOWS

En [32] se hace alusión a una analogía referenciada a la hora de aclarar el papel de OPC en el mundo de la automatización.

En MS-DOS¹² cada aplicación debía saber tratar con la impresora por su propia cuenta. De esta forma, por ejemplo, *PictureMaker* debía desarrollar, además de su propia aplicación, cada uno de los *drivers* de cada tipo de impresora a la que quisiera darle soporte. Por su parte *WordStar*, además de brindar todas las funcionalidades que se esperan de un editor de texto, también debía desarrollar el *driver* de cada tipo de impresora existente en el mercado.

Con la integración del soporte para impresora dentro del sistema operativo, en Windows, el problema quedó resuelto. Ahora es el fabricante de la impresora el que desarrolla el *driver* y este brinda servicio a todas las aplicaciones que requieran de la impresora a través del sistema operativo.

Ahora pasemos al mundo de la automatización. Por ejemplo, Intellution además de desarrollar su Interfaz Hombre-Máquina, HMI por sus siglas en inglés, debe desarrollar su

¹² Sistema operativo de Microsoft sin interfaz gráfica, anterior a Windows.

propio *driver* para cada dispositivo industrial, incluyendo cada rama de PLC¹³. Por su parte, Rockwell debe hacer lo mismo.

Cada fabricante de dispositivos industriales puede desarrollar su propio servidor OPC e integrarlo como un módulo a su dispositivo. De esta forma, cada aplicación que lo necesite puede convertirse en un cliente OPC y acceder a la información que contiene el dispositivo de una forma estándar.

1.4 LA FUNDACIÓN OPC¹⁴

La Fundación OPC es una organización global, independiente y sin fines de lucro. Fue creada para extender las especificaciones OPC, proporcionando al mismo tiempo a sus miembros un gran número de ventajas. Entre los miembros de la Fundación OPC se cuentan varias compañías industriales líderes en su campo que comparten la misma visión de una comunicación estandarizada entre *hardware* y *software* basada en tecnología Windows.

La Fundación OPC fomenta la difusión de las especificaciones del estándar y su conformidad con Windows. Diversas compañías, líderes industriales, están a la cabeza de los comités que se encargan de revisar las extensiones de puntos específicos del estándar. Estos puntos tienen que ver con la implementación de seguridad y confiabilidad dentro del estándar, sistemas de alarmas, protocolos de entrada y salida, compatibilidad con otros sistemas, y mucho más.

Los miembros de la junta directiva de la Fundación OPC incluyen compañías como: Fisher-Rosemount Systems (Austin, Texas); Honeywell (Minneapolis, Minnesota); Intellution (Norwood, Massachusetts); Opto 22 (Temecula, California); Rockwell Software (West Allis, Wisconsin); Siemens (Nuremberg, Alemania) y Toshiba (Tokyo, Japón).

El Consejo Europeo de la Fundación OPC está formado por un consorcio de compañías, cada una de ellas miembro de la fundación. El grupo estratégico europeo de la Fundación OPC está formado por las siguientes compañías: Intellution (Darmstadt, Alemania); National Instruments (Munich, Alemania); Phoenix Contact (Blomberg, Alemania); Rockwell

¹³ Programmable Logic Controller.

¹⁴ The OPC Foundation: <http://www.opcfoundation.org>

Automation (Milán, Italia); Siemens (Karlsruhe, Alemania) y Trebing & Himstedt (Schwerin, Alemania).[6]

1.5 LAS ESPECIFICACIONES OPC

Las especificaciones OPC definen un conjunto de objetos, métodos e interfaces. La Fundación OPC ha tenido en cuenta hasta el momento las siguientes especificaciones [32]:

- Acceso a Datos OPC (OPC DA, por sus siglas en inglés): definen la forma de mover datos en tiempo real desde un PLC, un DCS¹⁵ u otro dispositivo de control, hacia un HMI o algún otro cliente de visualización.
- Alarmas y Eventos OPC (OPC A&E): proporciona la forma de dar notificaciones de alarmas y eventos; incluyendo alarmas del proceso, acciones del operador, mensajes informativos y de rastreo.
- OPC por Lotes¹⁶: lleva la filosofía OPC a las necesidades especializadas de los procesos por lotes.
- Intercambio de datos OPC (OPC DX) [27]: provee la vía de comunicación entre dispositivos o buses, sin la intervención de un HMI u otra herramienta. Ello proporciona interoperabilidad entre diferentes proveedores. Agrega además configuración remota, diagnóstico y servicios de monitoreo y gestión.
- Acceso a datos históricos OPC (OPC HDA): mientras que OPC DA provee acceso en tiempo real, OPC HDA provee acceso a datos almacenados. Desde las actualizaciones de un simple dato, hasta los datos almacenados por un complejo sistema SCADA, pueden recuperarse de manera uniforme.
- Seguridad OPC: Todos los servidores de OPC proporcionan información que es valiosa a la empresa. Si esta fuera inadecuadamente actualizada, podría tener consecuencias significativas para los procesos de una planta. Las especificaciones que tienen en cuenta la seguridad en cuanto a OPC definen cómo controlar el acceso de clientes a estos servidores, de forma tal que se proteja esta información sensible y evitar la modificación desautorizada de parámetros del proceso.

¹⁵ Distributed Control System.

¹⁶ OPC Batch, en inglés.

-
- Acceso a datos XML¹⁷ (OPC XML-DA): Mantiene reglas flexibles, consistentes y formatos para exponer datos usando XML, esto está influenciado por el trabajo hecho por Microsoft y otros en cuanto a SOAP¹⁸ y Servicios Web.
 - Datos OPC complejos (OPC CPX) [26]: es una especificación que acompaña a OPC DA y OPC XML-DA. Brinda la forma de exponer tipos de datos complejos, tales como estructuras binarias y documentos XML.
 - Comandos OPC: en términos de OPC, los comandos representan acciones que pueden cambiar el estado del servidor o el dispositivo. Estas especificaciones definen la forma de controlar los métodos que un servidor provee, a través de comandos OPC; los cuales se limitan a acciones de control tales como: pausa, reavivar o cancelar.

1.6 ESTÁNDARES SOBRE LOS QUE SE APOYA OPC

La gran virtud de OPC es que gravita sobre estándares bien establecidos, como son:

- COM y DCOM
- OLE y *OLE Automation*
- Controles ActiveX
- DDE¹⁹
- XML

COM

Fue creado por Microsoft y define una vía común para que diferentes tipos de software exporten funcionalidades. Especifica una forma de implementar objetos COM que proveen servicios a través de métodos agrupados en interfaces que son accedidas por aplicaciones clientes. Cada objeto COM es una instancia de una clase. La Biblioteca COM está presente en todos los sistemas Windows y tiene acceso a un directorio con todas las clases disponibles en el sistema de objetos COM.

¹⁷ Extensible Markup Language.

¹⁸ Simple Object Access Protocol.

¹⁹ Dynamic Data Exchange.

COM precisa un mecanismo efectivo para la reutilización de *software* mediante la creación de componentes. A diferencia de algunos esquemas de desarrollo basado en componentes y comúnmente inherente a un lenguaje o compilador específico, COM define un mecanismo para enlazar objetos binarios permitiendo la distribución de objetos sin su código fuente, la reutilización de objetos a través de lenguajes diferentes y no tener que recompilar una aplicación entera cuando un objeto cambia. Ofrece, entre otros beneficios, programación orientada a objetos o componentes, independencia del lenguaje y un versionamiento simple y eficiente.

Puntualiza cómo un componente y un cliente interactúan, sin la necesidad de ningún intermediario; precisando que los objetos COM sean accedidos a través de los métodos contenidos en sus interfaces, a las cuales tienen acceso los clientes (Figura 1.3). Véase [33]



Figura 1.3 Interacción cliente-componente usando COM.

A través de COM, una aplicación puede usar funciones de cualquier otro objeto de la aplicación o sistema operativo y se permite la actualización de componentes integrados en la aplicación, sin afectar el funcionamiento de la solución global.

COM puede usarse por diseñadores y desarrolladores de sistema para crear soluciones particulares. Constituye el núcleo de DCOM, ActiveX y OLE [18].

DCOM

COM Distribuido (DCOM) extiende COM a la red. Utiliza llamadas a procedimientos remotos (RPC²⁰), de forma tal que el cliente no necesita hacer distinción entre estar usando un objeto local o remoto (Figura 1.4).

²⁰ Remote Procedure Call.

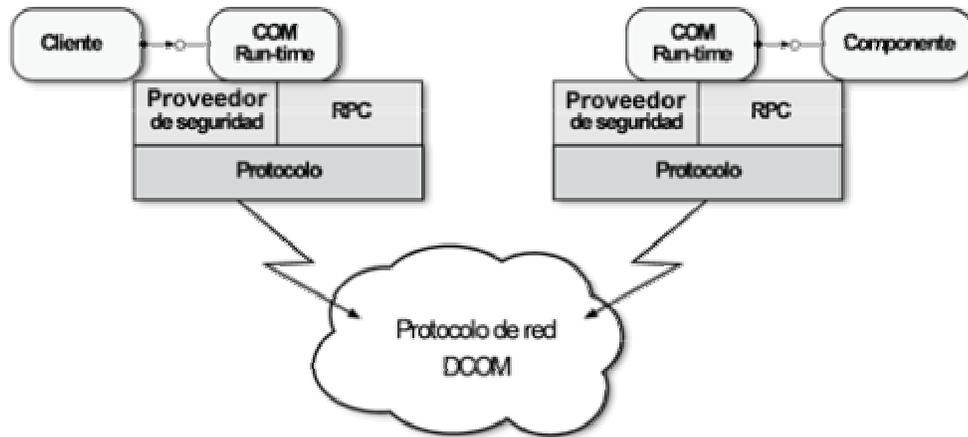


Figura 1.4 Interacción cliente-componente usando DCOM.

De forma general, envuelve tres elementos: creación de objetos remotos, protocolo para invocar métodos y mecanismos para realizar acceso seguro a objetos.

DCOM surge con Windows NT 4.0. Microsoft Java y Visual Basic soportan desarrollo de DCOM y ActiveX. Otras compañías desarrollan versiones de DCOM y ActiveX para las plataformas ajenas a Windows. [7]

OLE

La Vinculación y Empotrado de Objetos (OLE), es una tecnología basada en COM que se usa para proporcionar la integración entre aplicaciones; posibilitando un alto grado de compatibilidad entre aplicaciones, incluso para diversos tipos de información.

Actualmente se refiere solamente a la tecnología de creación de documentos compuestos (OLECD²¹). Esto permite la creación de documentos compuestos, cuyos componentes pueden ser manipulados por aplicaciones diferentes [4].

OLE AUTOMATION

Es una vía genérica para que cualquier tipo de aplicación pueda exponer sus servicios internos a otras aplicaciones. Esto implica que en lugar de constituir herramientas solo para

²¹ OLE Compound Documents.

usuarios finales; las aplicaciones actúen además como cajas de herramientas para los programadores.

Es cierto que para exportar programabilidad una aplicación, puede exponer sus funciones internas como métodos en interfaces COM; pero a pesar de que COM puede ser utilizado para esto, no aporta una forma de que un lenguaje simple como Visual Basic haga uso de forma fácil de esta programabilidad, ya que el mecanismo interno de invocación de métodos de interfaces COM no es utilizable, de forma sencilla desde tales herramientas.

OLE *Automation* es una forma estándar de hacer programable cualquier aplicación y pueda ser utilizada desde cualquier lenguaje, incluyendo los lenguajes más simples.

CONTROLES ACTIVEX

Es completamente posible construir componentes utilizando solo las convenciones de COM; pero es útil definir vías estándares para que los componentes definan sus propias interfaces gráficas de usuario, mecanismos para enviar eventos a sus clientes y/o permitir a sus clientes examinar y modificar sus propiedades. Esto es lo que permiten los Controles ActiveX, facilitándole el trabajo al software que utiliza componentes.

La especificación de ActiveX define un conjunto de interfaces estándares que los objetos COM pueden soportar para realizar acciones particulares y construir componentes más potentes. Además debido a que los controles necesitan una forma de interactuar con el cliente que los usa, también especifican reglas para la creación de los “contenedores²²” de los controles.

Históricamente los Controles ActiveX surgen como el desarrollo de los llamados Controles OLE, los cuales fueron un mecanismo mucho más rígido para alcanzar los mismos propósitos.

²² Software cliente que sabe como utilizar un control ActiveX.

DDE

Es el predecesor de OLE. El Intercambio Dinámico de Datos (DDE), es un método que permite intercambiar datos de forma dinámica entre aplicaciones. El protocolo DDE envía mensajes entre aplicaciones que comparten datos, usando una memoria compartida.

Las aplicaciones pueden usar el protocolo DDE para transferir un dato, una sola vez o de forma continua, en esta última, las aplicaciones se mandan actualizaciones unas a otras, del nuevo dato [20].

XML

Es una familia de tecnologías, entre las que se incluye el metalenguaje de marcado XML que sirve para definir otros lenguajes de marcado y representar la información con la semántica deseada, de tal forma que los elementos que lo componen puedan dar información sobre lo que contienen, y no necesariamente sobre su estructura física o presentación, como ocurre en HTML²³. La familia de tecnologías XML es independiente de la plataforma sobre la que se use, por lo tanto compatible con todo tipo de sistemas. [37]

1.7 INTERFACES OPC

Los objetos COM se acceden a través de interfaces. En las especificaciones OPC, de forma general tienen en cuenta dos tipos de interfaces: las interfaces normales y las automáticas. Una aplicación cliente OPC se comunica con un servidor OPC a través de las interfaces normales o automáticas. [25]

De forma general, las aplicaciones clientes desarrolladas en Visual Basic, Delphi y otras aplicaciones orientadas al control, para interactuar con servidores OPC; lo hacen a través de las interfaces automáticas; mientras que a los desarrolladores en ambiente C, les resulta más sencillo usar las interfaces normales, con un máximo de rendimiento (Figura 1.6). Los servidores OPC deben implementar las interfaces normales y opcionalmente, las automáticas.

Para ayudar a la implementación de clientes OPC por la vía de las interfaces automáticas, La Fundación OPC brinda a sus miembros un paquete, conocido como Wrapper DLL (Figura

²³ HyperText Markup Language.

1.5), que contiene la DLL²⁴ y el IDL²⁵; además del código fuente en C++ de las funciones necesarias en las interfaces automáticas de los objetos OPC; así como un ejemplo en Visual Basic que demuestra como usar estas interfaces. Los desarrolladores y proveedores de OPC, pueden integrar esta a sus productos.

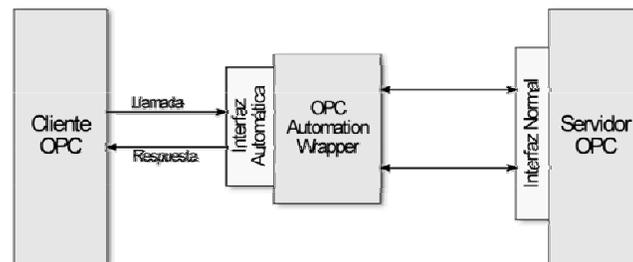


Figura 1.5 Uso de la interfaz automática.

Existe otro grupo de interfaces, a las que se hace referencia, como interfaces opcionales. Estas pueden ser adicionadas por el que desarrolle el servidor. Una aplicación cliente que requiera de las funcionalidades auxiliares añadidas, al igual que en el caso de un servicio de los especificados, debe hacer una llamada a la función que implemente esta interfaz. Un cliente debe estar diseñado para no depender de la existencia de esta interfaz opcional.

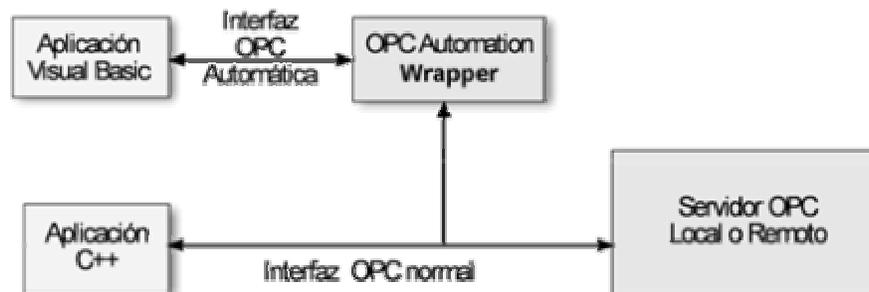


Figura 1.6 Interfaces típicas OPC.

1.8 ARQUITECTURA DE DISTINTOS TIPOS DE SERVIDORES OPC

Existen varios tipos de servidores OPC [32]:

- Servidor OPC de Acceso a Datos (Servidor OPC DA).

²⁴ Biblioteca de Enlace Dinámico, por sus siglas en inglés.

²⁵ Lenguaje de Definición de Interfaces, IDL por sus siglas en inglés.

- Servidor OPC de Alarmas y Eventos (Servidor OPC A&E).
- Servidor OPC de Datos Históricos (Servidor OPC HDA).
- Servidor OPC por Lotes.
- Servidor OPC XML de Acceso a Datos (Servidor OPC XML-DA).

1.8.1 SERVIDOR OPC DE ACCESO A DATOS

Un servidor OPC de Acceso a Datos está compuesto por los objetos (Figura 1.7): [21]

- Servidor: ofrece información propia y sirve como un contenedor de grupos de *items* OPC.
- Grupo: contiene información acerca de sí mismo y proporciona los mecanismos para contener y organizar lógicamente los *items* OPC.

Los grupos OPC representan una forma de organizar los datos de los clientes. Por ejemplo, el grupo podría representar los elementos en una pantalla particular del operador o los datos necesarios para confeccionar un informe. [20]

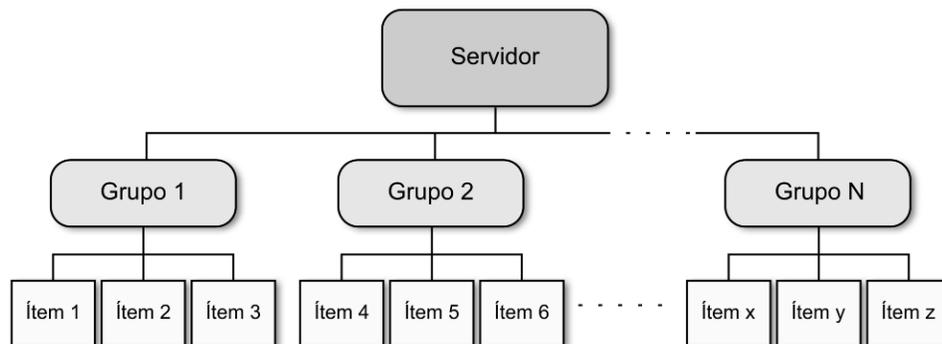


Figura 1.7 Relación Servidor-Grupo-Item.

Existen dos tipos de grupos, públicos y privados. Los públicos son para ser compartidos entre varios clientes, mientras que los privados son locales a un cliente específico.

Dentro de cada grupo, el cliente puede definir uno o más *items* OPC, a los cuales hay asociados un valor, un grado de calidad y la fecha. Los *items* OPC representan conexiones a fuentes de datos dentro del servidor y no son accesibles por el cliente como un objeto; de manera que no hay una interfaz externa definida para ellos. El acceso a un *item* OPC se realizan a través del objeto Grupo OPC que es su contenedor.

Un modelo de clases para la interfaz automática se define como sigue (Figura 1.8): [28]

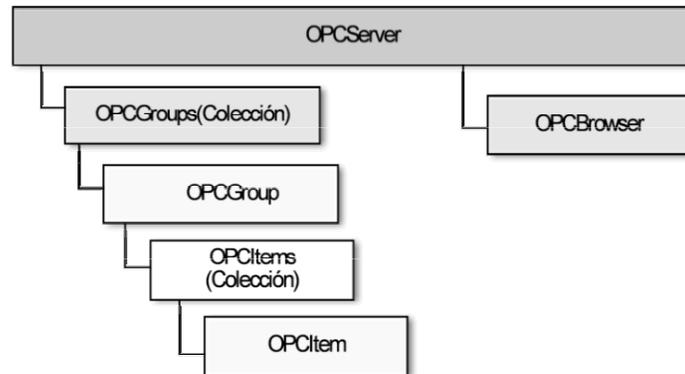


Figura 1.8: Modelo de clases de interfaz automática OPC DA.

- *OPCServer*: se debe crear una instancia de esta clase para poder acceder a los demás objetos. Contiene a *OPCGroups (Colección)* y crea el objeto *OPCBrowser*.
- *OPCGroups (Colección)*: contiene todos los *OPCGroup* creados por el cliente que se conecta, dentro del ámbito del objeto *OPCServer*
- *OPCGroup*: el propósito de éste objeto es mantener información de estado y proporcionar el mecanismo para proveer servicios de adquisición de datos para la colección *OPCItems* que referencia.
- *OPCItems (Colección)*: contiene todos los objetos *OPCItem*, creados por la aplicación cliente en el ámbito del objeto *OPCServer* y el *OPCGroup* correspondiente.
- *OPCItem*: contiene la definición del *item*, su valor actual, información de estado y última hora de actualización. Es de notar que la interfaz normal no define un objeto *OPCItem*; sino que lo trata como una referencia a un dato.
- *OPCBrowser*: visualiza los nombres de los *items* OPC configurados en el servidor. Existe una sola instancia de *OPCBrowser* por *OPCServer*.

1.8.2 SERVIDOR OPC DE ALARMAS Y EVENTOS

Dentro de OPC, una alarma es una situación anormal y por tanto es un caso especial de una condición. Por otro lado, un evento es una ocurrencia detectable que es de importancia para el servidor OPC, para el dispositivo que representa y para el cliente OPC. Un evento puede estar asociado a una condición; o no, en el caso de: acciones de los operadores, cambios en la

configuración del sistema y errores del sistema. Los clientes OPC se deben suscribir para que les sean notificadas las ocurrencias de eventos específicos.

Cualquier objeto COM que implemente una interfaz *IOPCEventServer* es un servidor OPC de eventos. La interfaz *IOPCEventServer* exporta métodos que posibilitan al cliente OPC:

- Determinar el tipo de evento que el servidor de eventos OPC soporta.
- Suscribirse y especificar los eventos de los que desea recibir notificación.
- Especificar la interfaz de llamada a invocar en caso de *shutdown*.

Un servidor OPC alarmas y eventos está compuesto por los siguientes objetos:

- *OPCEventServer*: es el creador de los objetos *OPCEvenSubscriber* y *OPCEventBrowser*. *OPCEvenSubscriber*: se utiliza para la configuración de filtros y otros atributos de un reporte de eventos OPC.
- *OPCEventBrowser*: suministra una manera a los clientes para acceder al área del proceso, organizada por el servidor. (la expectativa es que los eventos y condiciones proporcionadas por el servidor sean organizada en uno o más áreas de proceso, y que el cliente pueda filtrar las subscripciones de eventos según las áreas del proceso especificadas) Este objeto es optativo, y puede no exponerse por los servidores de eventos simples.

Existen varios tipos de servidores de alarmas y eventos. Los tipos que soportan las especificaciones OPC son:

- servidores que detectan alarmas y/o eventos y los reportan a uno o más clientes.
- servidores que pueden coleccionar alarmas y eventos de múltiples fuentes, ya sea suscribiéndose a otro servidor de alarmas y eventos o detectándolos, por si mismo y reportarlos a uno o más clientes.

Para más información véase [22].

1.8.3 SERVIDOR OPC DE DATOS HISTÓRICOS

Los mecanismos de históricos producen una fuente añadida de información que debe ser distribuida a los usuarios y las aplicaciones clientes que están interesados en esta información.

En la actualidad la mayoría de los sistemas históricos usan sus propias interfaces para la distribución de los datos. No hay capacidad para aumentar o utilizar soluciones preexistentes en un ambiente *Plug & Play*. Esto hace que el desarrollador tenga que recrear soluciones independientes para sus productos, como todos los demás vendedores. Como resultado se obtienen sistemas sin interoperatividad con otros sistemas.

Siguiendo con el deseo de integrar los datos de todos los niveles de un negocio, la información histórica podría ser considerada como otro tipo de datos.

Los objetos que se tienen en cuenta en las especificaciones OPC para este tipo de servidor son:

- *OPCHDAServer*: provee la lectura y escritura desde y para el servidor de históricos. El tipo de dato histórico, depende del servidor.
- *OPCHDABrowser*: exporta interfaces que brindan un método al cliente para acceder al espacio de dirección de datos históricos y es esencial en los servidores de los datos históricos de grandes proporciones. Esto le permite a un cliente repasar el espacio de dirección de una manera gráfica simple.
- *OPCHDAClient*: debe exportar interfaces para soportar una llamada de *shutdown* también puede exportar interfaces para varias conexiones asíncronas que un servidor HDA pueda proporcionar.

Para que un cliente pueda usar (siempre que el servidor la exporte) una interfaz asíncrona particular, debe llevar a cabo la petición correspondiente del servicio brindado por esa interfaz.

Se requiere de una llamada de *shutdown*, para que el servidor ejecute las acciones pertinentes, de forma tal que se libere el espacio de memoria asignado al cliente y se cierren las conexiones.

Existen varios tipos de servidores de datos históricos. Los tipos soportados por la especificación OPC son:

- Servidores de curso simple²⁶: Estos servidores proporcionan un almacenamiento de datos sin compresión. Los datos serán de los tipos de datos disponibles en un servidor de datos OPC.
- Servidores de análisis y compresión compleja de datos: Estos servidores proporcionan compresión así como almacenamiento sin compresión. Son capaces de proporcionar resúmenes y funciones de análisis de datos, así como valores intermedios, mínimos, máximos, etc. Estos pueden proporcionar actualizaciones de datos y la historia de las actualizaciones, además de almacenamiento de anotaciones junto a los históricos.

Para la interfaz automática se definen las siguientes clases: (Figura 1.9)

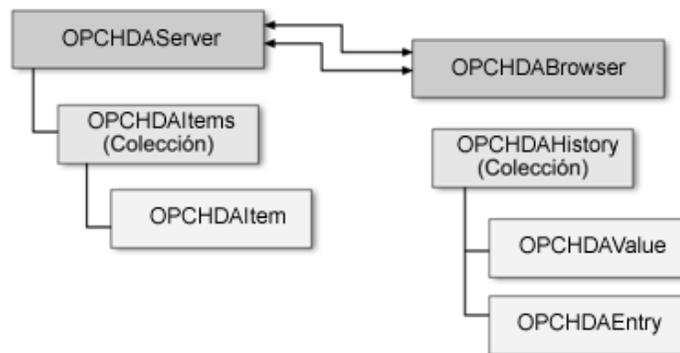


Figura 1.9 Modelo de clases para Servidor y navegador OPC HDA.

- *OPCHDAServer*: Se debe crear un objeto *OPCHDAServer* antes de que se puedan obtener referencias a otros objetos. Contiene la colección *OPCHDAItems*.
- *OPCHDAItems* (Colección): es una colección de objetos OLE Automation que contiene todos los objetos *OPCHDAItem* que un cliente ha creado dentro del contexto del objeto *OPCHDAServer*.
- *OPCHDAItem*: es un objeto OLE Automation que mantiene las definiciones de *item*. La interfaz normal no define a *OPCHDAItem* como objeto.
- *OPCHDABrowser*: navega por los nombres de los *item* en la configuración de *OPCHDAServer*.

²⁶ simple trend, en inglés.

- *OPCHDAHistory* (Colección): es una colección de *OLE Automation* que contiene una serie de objetos *OPCHDAValue* u *OPCHDAEntry*, representando los valores históricos de algunos *item*.
- *OPCHDAValue*: es un objeto que representa un valor histórico discreto para un *item* o un atributo.
- *OPCHDAEntry*: es un objeto que extiende el objeto *OPCHDAValue* para incluir información adicional.

1.8.4 SERVIDOR OPC POR LOTES

El servidor OPC por Lotes proporciona una manera para acceder o comunicarse a un grupo de fuentes de datos por lotes. Los tipos de fuentes disponibles están en dependencia de la implementación del servidor.

El servidor puede ser implementado para coleccionar los datos de un servidor OPC de Acceso a Datos u otra fuente de datos. También puede ser un juego de interfaces adicionales a un Servidor OPC por Lotes existente (Figura 1.10). [24]

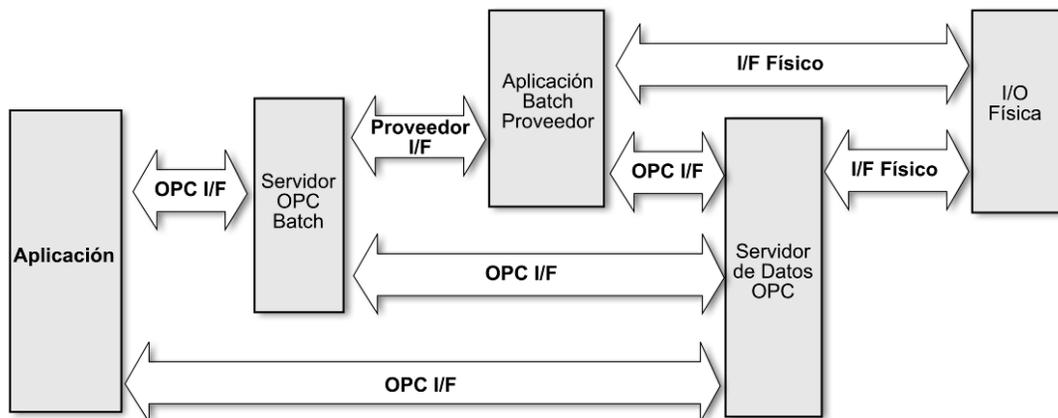


Figura 1.10 Posibles roles de un servidor OPC por lotes.

Los clientes de un servidor OPC por lotes pueden ser aplicaciones que sólo necesiten algunos valores o informes complejos que requieran datos en múltiples formatos.

Un objeto servidor OPC por lotes brinda la posibilidad de leer datos desde el servidor por lotes y de forma opcional, escribir datos para un servidor por lotes.

No existen interfaces específicas para un cliente de un servidor OPC por lotes; aunque se plantea que debe cumplir con las especificaciones de Acceso a Datos 2.0 y entender la estructura del servidor OPC por lotes.

Un modelo de clases para las interfaces automáticas se define como sigue [23] (Figura 1.11):

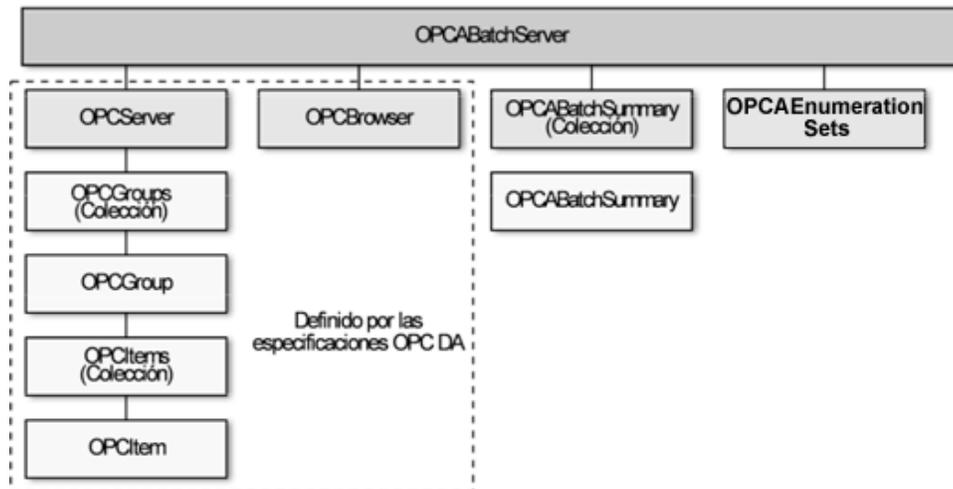


Figura 1.11 Modelo de clases para interfaz automática de OPC por Lotes.

- *OPCABatchServer*: se debe crear primero un objeto *OPCABatchServer* antes de que se puedan obtener referencias a otros objetos.
- *OPCABatchSummary* (Colección): es una colección de los objetos *OPCABatchSummary* que representan los lotes conocidos en el objeto *OPCABatchServer*.
- *OPCABatchSummary*: el propósito de este objeto es mantener información de estado respecto a un solo lote.
- *OPCAEnumerationSets*: es un objeto que permite al cliente preguntar por los conjuntos enumeración en el servidor.

Las clases: *OPCServer*, *OPCGroups* (Colección), *OPCGroup*, *OPCItems*, *OPCItem*, *OPCBrowser*, son las definidas por OPC DA.

1.8.5 SERVIDOR OPC DE ACCESO A DATOS XML

El diseño basado en XML proporciona el intercambio de información estructurada entre aplicaciones. XML es una tecnología altamente adaptable que se está desarrollando de forma precipitada en una gama amplia de plataformas. Las especificaciones OPC XML de Acceso a

Datos son la adopción de la Fundación de OPC de la tecnología XML, para facilitar el intercambio de datos de la planta, vía Internet, y dentro del dominio de la empresa.

Las interfaces que debe brindar un servidor OPC XML-DA son: [31]

- Estado: *GetStatus, GetStatusResponse*
- Lectura: *Read, ReadResponse*
- Escritura: *Write, WriteResponse*
- Suscripción: *Subscribe, SubscribeResponse*
- Refrescar las suscripciones de encuesta: *SubscriptionPolledRefresh,*
- Cancelar suscripciones: *SubscriptionCancel, SubscriptionCancelResponse*
- Navegador: *Browse, BrowseResponse*
- Lectura de propiedades: *GetProperties, GetPropertiesResponse*

1.9 DETECCIÓN DE SERVIDORES OPC INSTALADOS

Los servidores OPC actualmente se registran en el sistema a través de las categorías de componentes. Esto permite que el gestor de categorías de componentes²⁷ pueda ser usado para determinar que servidores OPC están instalados en una máquina local.

El problema surge, cuando un cliente necesita determinar los servidores OPC disponibles en un terminal remoto; debido a que esta funcionalidad solo se desarrolla en el entorno local.

Para resolver este problema La Fundación OPC proporciona el buscador de servidores: *OPCEnum.exe*, el cual puede residir en cualquier máquina. Éste accede al gestor de categorías de componentes y proporciona una interfaz, la cual puede ser usada por clientes remotos. Este buscador debe venir con la instalación de cualquier servidor OPC. El cliente solo necesita conocer el nombre del nodo al que se va conectar, para un objeto de forma remota y usar su interfaz *IOPCServerList* para determinar qué tipos de servidores están disponibles en dicha máquina.

²⁷ Funcionalidad de Windows contenida en una DLL.

1.10 SEGURIDAD EN OPC.

El modelo de seguridad OPC es consistente con el de Windows NT. Para entenderlo, hagamos un acercamiento entonces al modelo de seguridad NT, para que nos sirva de armazón conceptual (Figura 1.12). [30]

1.10.1 MODELO DE SEGURIDAD NT.

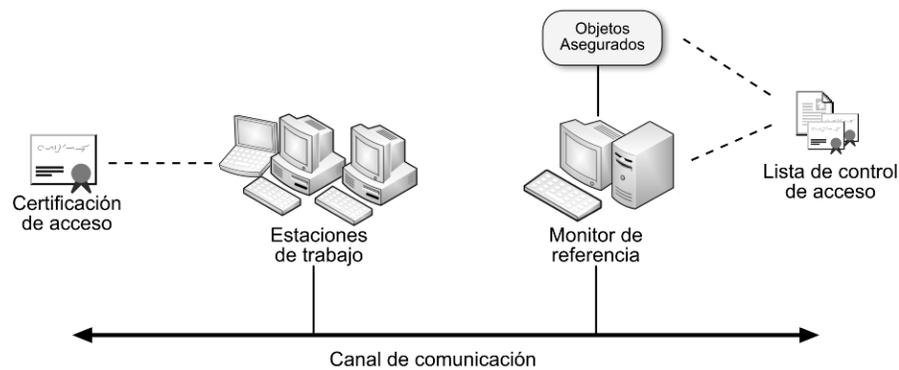


Figura 1.12 Modelo de seguridad NT.

- **Principal:** Entidad activa que requiere acceder a uno o más objetos de seguridad. Algunos ejemplos de Principal son los usuarios, los sistemas de la computadora, etc. En NT, los procesos del sistema representan a todos los Principal; que pueden ser interactivos o de servicio.
- **Objeto asegurado:** Entidad a la que se le necesita controlar el acceso. Algunos ejemplos de objetos de seguridad son los archivos, los directorios, las entradas al registro, sistemas importantes, etc.
- **Monitor de Referencias:** Entidad activa que toma las decisiones de autorización de acceso para un grupo de objetos de seguridad. Un sistema de seguridad bien implementado asegura que todo el acceso a un Objeto de Seguridad sea a través de su Controlador de Referencia asociado, es decir, ningún acceso a un Objeto de Seguridad puede concederse excepto por la aprobación de su Controlador de referencia. Un ejemplo de un controlador de referencia es el Controlador de Referencia NT para los archivos y directorios almacenados en formato NTFS.

- Lista de Control de Acceso (ACL): Estructura asociada con cada Objeto de Seguridad que especifica aquellos Principales y grupos de Principales a los que se les permite acceder al Objeto Seguro, junto con los tipos de acceso que se les permite. Ejemplo e estas son las Listas de Control de Acceso asociadas a los archivos, directorios, los servidores de DCOM, entradas de registro, etc.
- Canal: Camino de comunicación entre dos entidades activas dentro del sistema. Un canal puede ser asegurado mediante uno de estos tres niveles:
 - Verificable: El destinatario puede estar seguro que el mensaje no se ha modificado en el tránsito.
 - Privado: El remitente puede estar seguro que el mensaje sólo se leerá por los destinatarios deseados.
 - Autenticado: El destinatario puede estar seguro de la identidad de un remitente del mensaje.
- Certificado de Acceso: Credencial de confianza asociada a un Principal; que es usada por un Controlador de Referencia como base para una decisión de autorización de acceso. Un ejemplo de un certificado de acceso es la Ficha de Acceso NT asociada con cada proceso NT.

El concepto de autorización, según este modelo, es el proceso de conceder a Principal un acceso a un Objeto de Seguridad. La decisión de una autorización es tarea del Controlador de Referencia; basándose en una comparación del Certificado de Acceso del Principal con la Lista de Control de Acceso del Objeto de Seguridad.

Por su parte, la autenticación es una propiedad del Canal de comunicaciones; que se refiere a la habilidad del Controlador de Referencia de estar seguro de la validez de la identidad y credenciales de un acceso pedido por un Principal a un Objeto de Seguridad.

1.10.2 APLICACIÓN DEL MODELO DE REFERENCIA NT A OPC

Los Principales son los procesos corriendo en un servidor o un cliente. Programas o aplicaciones no son Principales; aunque se les asigna el acceso de los procesos que en ellos están corriendo. En NT, todos los procesos corren bajo una cuenta de usuario y esta especifica sus derechos de acceso.

Los Certificados de Acceso son la Ficha de Acceso NT, que debe usarse cada vez que sea posible o una credencial privada, especificada por el servidor OPC.

Los Objetos de Seguridad son el servidor y el cliente OPC (en caso de poseer interfaces de respuesta) y cualquiera que decida implementar el proveedor en su servidor OPC. Ejemplos de estos últimos objetos son las interfaces y métodos de los servidores, grupos públicos y los *items* o grupos de *items*.

Las Listas de Control de Acceso son las de NT, con la posible excepción en el caso en que se usa para la seguridad las credenciales privadas.

Los Controladores de Referencia son los de NT, que controlan el acceso a los servidores DCOM y a las aplicaciones clientes o los servidores OPC que toman decisiones de acceso a los objetos específicos, diseñados y desarrollados por los proveedores. Ellos tomarán sus decisiones de autorización basados en la Ficha de Acceso NT o en las credenciales privadas específicas en un proveedor. La Ficha de Acceso NT es la estrategia preferida.

Los niveles de los Canales de Seguridad son los soportados por DCOM.

1.10.3 NIVELES DE SEGURIDAD

Un Servidor de OPC puede llevar a cabo uno de tres niveles de seguridad:

- Sin seguridad: cualquiera puede activar y acceder a un servidor OPC y acceder a un cliente OPC. El servidor OPC no controla el acceso a ningún objeto de seguridad implementado por un proveedor.
- Seguridad Básica: sólo se implementa la seguridad de DCOM; a través del modelo de seguridad NT: se limitan la activación y el acceso al servidor OPC, así como el acceso a los clientes. Sin embargo, el servidor OPC no controla el acceso a cualquier objeto de seguridad específico del proveedor. Este es el nivel de seguridad predefinido proporcionado por DCOM.
- Seguridad Extendida: el Servidor de OPC sirve como un Controlador de Referencia para los Objetos de Seguridad específicos del proveedor que son expuestos por el servidor OPC. Un Servidor de OPC puede llevar a cabo la Seguridad OPC además de la Seguridad DCOM, o implementar la Seguridad OPC exclusivamente.

Siempre que sea posible un Servidor OPC que implemente seguridad debe basar sus decisiones de autorización de acceso en la Ficha de Acceso NT asociada con la aplicación cliente. Esta estrategia permite que la seguridad sea transparente para la aplicación cliente, ya que no tiene necesidad de tomar una acción explícita para establecer un certificado de acceso adicional. Al ser la Ficha de Acceso NT independiente del Servidor OPC, se hace más fácil escribir aplicaciones clientes portátiles.

Existen circunstancias que evitan el uso de la Ficha de Acceso NT. Algunos ejemplos de tales situaciones incluyen:

- El Servidor OPC está corriendo en un sistema operativo diferente de NT; que soporta DCOM. Tal es el caso de Windows CE Y UNIX.
- Las aplicaciones cliente pueden estar corriendo en un sistema operativo que no soporta la creación de una Ficha de Acceso NT.
- El Servidor OPC y las aplicaciones cliente son distribuidos diferentes dispositivos fuera del contexto de un Dominio de NT.

1.11 CONSIDERACIONES GENERALES

Con lo visto hasta el momento se puede inferir la integración de OPC en diferentes conceptos.

- Se habla de tecnología flexible, cuando se aprecia como un conjunto de métodos y procedimientos, basados en estándares bien definidos como son OLE, COM y DCOM; constituyendo una poderosa herramienta para la transferencia de datos, tan necesaria en la industria en general.
- Es un estándar, cuando se tiene en cuenta que la mayoría de los sistemas SCADA tienen en cuenta la interconexión usando OPC y que brinda el soporte para comunicación normalizada; donde los diferentes proveedores al incluirlo en sus dispositivos o HMI no se ven tan divorciados de otros sistemas como en los tiempos en que era imprescindible el desarrollo de *drivers* para lograr interoperabilidad.
- Esta tecnología constituye también un protocolo, al ser el medio sobre el que se materializa la interconexión como forma de comunicación entre los diferentes sistemas.

En este capítulo se ha hecho una descripción de la estructura concebida para la transferencia de datos usando el protocolo OPC. Se ha ido de lo más general a lo más específico, sin llegar a una descripción detallada de las interfaces de los objetos OPC; debido a que se considera que esto se iría de los marcos de este trabajo; ya que se aspira a una comprensión del protocolo OPC y no a desarrollar una aplicación OPC; en cuyo caso habría que consultar en detalle las especificaciones OPC.

En el siguiente capítulo, se verá la puesta en línea de un servidor y un cliente OPC. Primero se expondrá un ejemplo de configuración sencillo y seguidamente se implementará una red cliente/servidor con variables de interés industrial.

Capítulo 2 INSTALACIÓN Y CONFIGURACIÓN DE UNA RED CLIENTE/SERVIDOR OPC

Hasta ahora se ha orientado el trabajo al conocimiento del tema OPC y lo bien que ha venido a la industria y al mundo de la automatización. En lo adelante, habrá un acercamiento a la materialización de esta teoría, anteriormente expuesta.

Para todas las pruebas realizadas, se ha tenido en cuenta un esquema cliente/servidor como sigue (Figura 2.1)

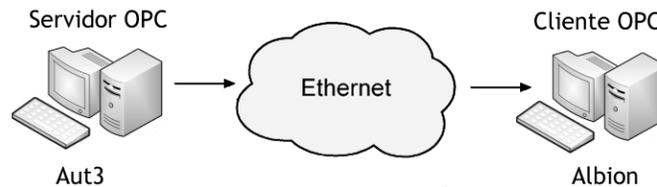


Figura 2.1 Red cliente/servidor OPC.

Nos se ha tenido en cuenta en el presente informa una configuración compleja de red, ya que como OPC se basa en estándares bien definidos, sería redundante probarlos, ya que son de uso cotidiano. Se ha supuesto que después que se ha logrado la comunicación de forma satisfactoria entre servidor y un cliente OPC, la configuración de una red más compleja, queda en manos de las necesidades que se tenga.

Primeramente se pasará a la selección de las herramientas a utilizar para la implementación de la red,

2.1 SELECCIÓN DE LAS HERRAMIENTAS A UTILIZAR

Para la configuración de una red OPC práctica, se ha decidido utilizar como fuente de datos, variables que puedan ser obtenidas en el laboratorio y que puedan brindar interés industrial. En el laboratorio radicado en el local 225, se cuenta con el equipo Servosistemas SAD-100, de la firma Alecop, el cual en este caso se utilizará para el accionamiento eléctrico de un sistema de servomotor, masa de inercia, freno y tacómetro, incluido en el equipo.

Para la adquisición del voltaje proveniente de este tacómetro, se cuenta con la tarjeta PCI-1711, de la firma ADVANTECH, cuyo CD de instalación [1] contiene varios ejemplos de comunicación con diferentes plataformas de programación, como son Delphi, Visual Basic, C++ y LabVIEW²⁸. En la red OPC a implementar se ha decidido utilizar la aplicación LabVIEW, debido a que es la más afín a la especialidad Automática.

Dado que LabVIEW soporta DDE, a través de la creación de aplicaciones que actúen como servidor o como cliente y que en el mercado se brinda servidores OPC para DDE, es que se ha concebido una configuración en que con LabVIEW se implemente una aplicación que adquiera los datos de interés y los brinde a través de DDE a otra aplicación que actúe como cliente DDE y servidor OPC.

En este punto se probaron servidores y clientes OPC de diferentes firmas: [12], [13], [14], [15], [16], [17], [35], [36]. Durante este proceso se notó que es prácticamente una generalidad que los clientes se brinden de forma libre y que a los servidores se les restrinja su funcionalidad a un determinado tiempo, después de lo cual habría que reiniciarlos. Se comprenderá que de esta manera no es factible su uso en la industria; para lo cual habría que comprar las aplicaciones OPC. Pero para un interés didáctico, como es el que corresponde, esta restricción no resulta impedimento.

Teniendo en cuenta que solo se usará DDE como fuente de datos para el servidor OPC y que Matrikon es de los proveedores que brinda un módulo servidor por cada tipo de comunicación que soporta y que es un proveedor de renombre en cuanto a tecnología y herramientas OPC, se ha seleccionado como aplicación cliente DDE y servidor OPC el *Matrikon OPC Server for DDE*, en su versión 2.1.0.65, que era la más actualizada en el momento en que se confeccionó el presente informe.

Se habla de un módulo servidor por cada tipo de conexión, porque OPC permite que un mismo servidor pueda estar usando varias conexiones a la vez (Figura 2.2), como el caso de las herramientas de Kepware, por ejemplo.

²⁸ Laboratory Virtual Instrument Engineering Workbench.

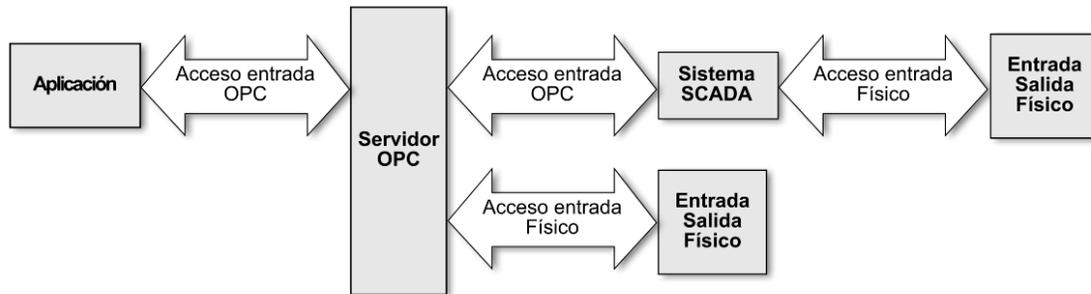


Figura 2.2 Servidor OPC usando diferentes drivers.

Para el caso de selección de la herramienta cliente OPC, no se ha hecho referencia a una selección como tal, debido a que la principal diferencia entre ellas estriba en el manejo que dan al dato recuperado a través de OPC, ya que la forma de establecer la comunicación está bien definida. Para el caso que nos ocupa, de comunicación OPC DA, un *item* OPC se referencia a través del esquema de árbol `Servidor.Grupo.Item`. De esta forma, para la implementación de la red que ha sido propuesta como objetivo, se utilizara como cliente el *Matrikon OPC Explorer*.

De cualquier forma, se he considerado que un acercamiento a OPC a través estas herramientas de Matrikon introduciría el método que serviría para cualquier otro proveedor; ya que aunque con ciertas diferencias que no son considerables al extremo, todos tienen una fuente común, que son las especificaciones OPC.

El siguiente epígrafe expone una prueba mas bien didáctica, que aunque no sigue el esquema de comunicación que se ha propuesto anteriormente, si brinda una introducción a esta, ya que se establecerá la comunicación usando OPC, lo único que la fuente de datos será una aplicación conocida por todos, que entre sus funcionalidades tiene que puede actuar como servidor y como cliente DDE. Esta aplicación es el *Microsoft Excel*.

2.2 IMPLEMENTACIÓN DE UNA RED OPC SENCILLA, A MANERA DE INTRODUCCIÓN

Pasemos a ver entonces la forma de instalar y configurar una red OPC, según un esquema sencillo (Figura 2.3), el cual posteriormente se llevará a práctico, con la transmisión de variables obtenidas en el laboratorio a través del esquema planteado en el epígrafe anterior.

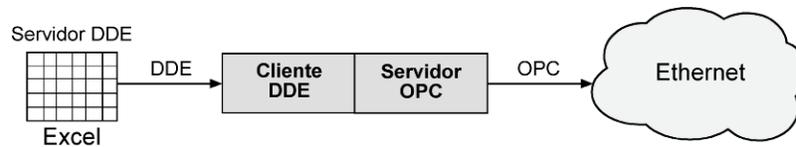


Figura 2.3 Esquema sencillo de comunicación usando OPC.

2.2.1 INSTALACIÓN DEL SERVIDOR OPC

En el proceso de instalación del Servidor OPC para DDE de Matrikon, se instalarán todos los componentes (Figura 2.4).

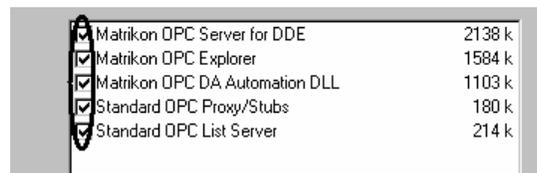


Figura 2.4 Componentes a instalar.

En lo concerniente a la opción *Standard OPC Proxy/Stubs*, es de destacar que los vendedores inicialmente debían proveer sus propias versiones. Esto causaba conflictos entre las DLL de los productos de diferentes proveedores, si estas coexistían en la misma computadora. Para resolver este problema, la Fundación OPC creó unas DLL oficiales y pone como requerimiento que estas sean usadas por todos los proveedores. De esta forma, solo las versiones antiguas de los servidores OPC siguen instalando sus propias DLL, causando los conflictos anteriormente referenciados. Si estos conflictos ocurren, lo que se debe hacer es reinstalar una versión actual de servidor, para asegurarse de que la versión de las DLL instaladas sean las oficiales.

Justo después de la instalación del servidor, se ofrecen las opciones de instalarlo como un servicio NT o como una aplicación local y la de restaurar el *ID* de antiguas versiones, en los registros (Figura 2.5). Esto último con el objetivo de guardar compatibilidad con clientes concebidos para tratar con versiones anteriores del servidor.

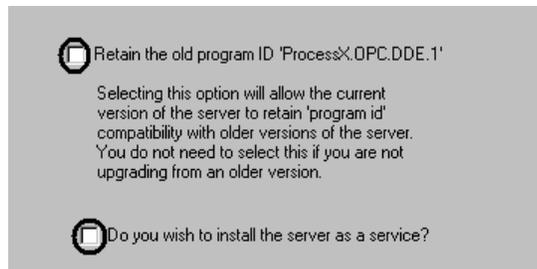


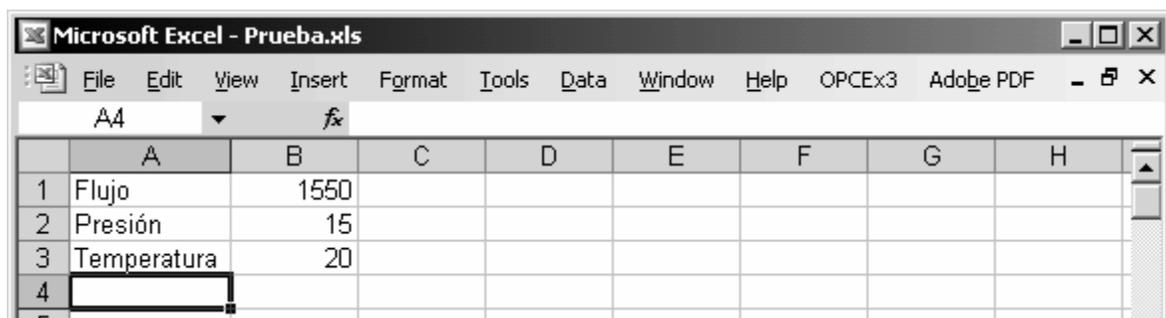
Figura 2.5 Opciones de la instalación.

Una vez que se tiene el servidor OPC instalado, lo que resta es configurarlo; pero antes, se debe ejecutar alguna aplicación que funcione como servidor de DDE. En este punto se explicará, por el momento, la aplicación *Microsoft Excel*, como se ha planteado anteriormente.

2.2.2 MICROSOFT EXCEL COMO SERVIDOR DDE

Dentro de las funcionalidades de *Microsoft Excel*, está que actúa como servidor y como cliente DDE. En este caso lo usaremos como servidor DDE; pero bien pudiera utilizarse como cliente DDE, para comunicar con una aplicación cliente OPC y servidor DDE [16] [35], de esta forma, se podrían almacenar las actualización de datos brindados por un servidor OPC.

Para usar a *Microsoft Excel* como servidor DDE, primeramente se confecciona una hoja de cálculo que podría quedar como sigue (Figura 2.6):



The image shows a screenshot of a Microsoft Excel window titled "Microsoft Excel - Prueba.xls". The window has a menu bar with "File", "Edit", "View", "Insert", "Format", "Tools", "Data", "Window", "Help", "OPCEX3", and "Adobe PDF". The spreadsheet shows a table with 4 rows and 8 columns (A-H). The data is as follows:

	A	B	C	D	E	F	G	H
1	Flujo	1550						
2	Presión	15						
3	Temperatura	20						
4								

Figura 2.6 Documento Excel que actuará como fuente del servidor OPC.

Las aplicaciones DDE utilizan un sistema de identificación que tiene una estructura de árbol. El nombre de aplicación es el nivel más alto de la jerarquía y define a la aplicación servidor. Cada nombre de aplicación controla uno o varios temas, identificados por sus nombres de

tema, nombres que también se refieren a la aplicación servidor que, a su vez contienen uno o más nombres de *items*, con los detalles de cada tema. De esta forma, DDE especifica una cadena estándar de la forma:

*[nombre_de_aplicación][nombre_de_tema]![nombre_de_item]*²⁹

El carácter que funciona como separador entre el nombre de aplicación y el de tema, no es comúnmente usado. Su código ASCII³⁰ en decimal es 124.

De esta forma, la dirección de un dato que *Microsoft Excel* brinde por DDE, según lo que esta aplicación especifica es:

Excel|[Documento.xls]Sheet#!R#C#

Como “Documento”, se pone el nombre de la hoja de cálculo que debe estar abierta para poder establecer la comunicación. El símbolo de número, es un valor que debemos especificar, para acceder a la hoja (*Sheet*), a la fila (R) y la columna (C).

De esta forma, para acceder, por ejemplo al valor del flujo, desde una aplicación cliente DDE, sería:

Excel|[Prueba.xls]Sheet1!R1C2

Esto se verá cuando se configure el cliente DDE y servidor: *OPC Matrikon OPC Server for DDE*.

Una vez que se tiene el servidor DDE y el servidor OPC para DDE, lo que resta es conectarlos, para después acceder al servidor OPC a través de un cliente remoto.

2.2.3 CONFIGURACIÓN DEL SERVIDOR OPC

Para acceder a la herramienta que permite configurar el servidor OPC, primero hay que activar el mismo, a través del menú inicio. Seguidamente, se da clic derecho en el icono que aparece en la barra de tareas (Figura 2.7).

²⁹ En ingles: AppName|TopicName!ItemName

³⁰ American Standard Code for Information Interchange.

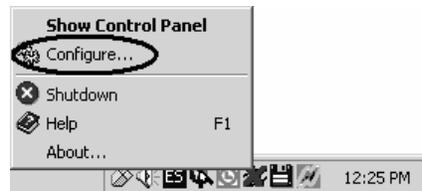


Figura 2.7 Para acceder a la herramienta de configuración

En el caso en que se hubiese instalado el servidor OPC como un servicio de Windows; esta herramienta (Figura 2.8) aparece directamente, a través del acceso directo del menú inicio. De cualquiera de las dos formas, ahora podemos configurar el servidor OPC, para que actúe como cliente DDE.

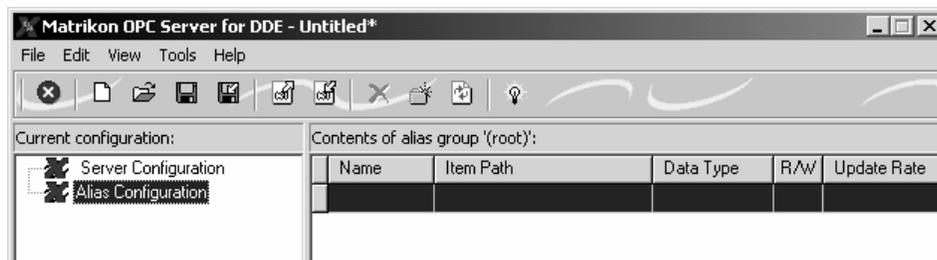


Figura 2.8 Herramienta de configuración de la aplicación Matrikon OPC Server for DDE.

Con *Alias Configuration* marcado, en la lista de configuraciones, se inserta un grupo de *alias*, a través del menú *Edit* o dando clic derecho en *Alias Configuration*. Una vez que se ha hecho esto, se adiciona un nuevo *alias*, dando clic derecho en el panel de contenido del grupo de *alias* o a través del menú inicio. A este *alias* se le especifica el nombre, la dirección de donde se va a tomar, según lo que especifica el protocolo DDE y el periodo de muestreo con que se va a actualizar el dato. Quedaría como sigue (Figura 2.9).

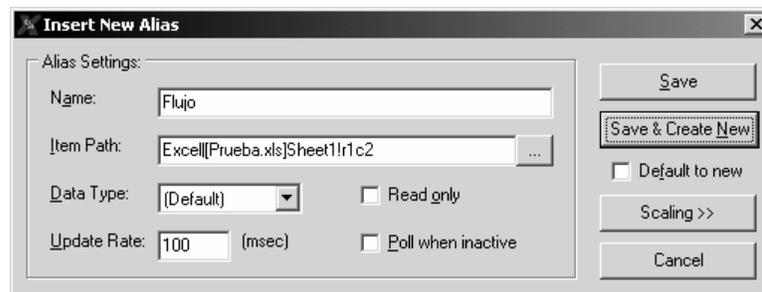


Figura 2.9 Para configurar un item.

La herramienta de configuración quedaría como sigue (Figura 2.10):

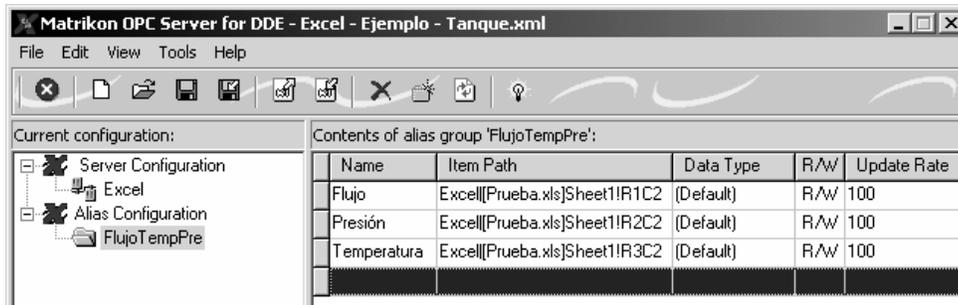


Figura 2.10 Servidor OPC configurado.

Se graba esta configuración y se pone en el menú de propiedades generales del menú *View*; que la cargue cuando se inicie (Figura 2.11).

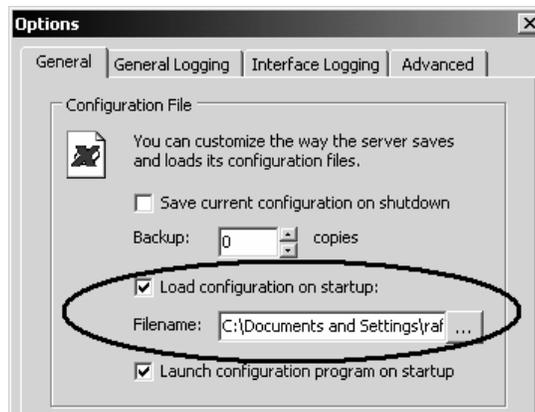


Figura 2.11 Para definir configuración por defecto.

Ya se tienen el servidor DDE y el cliente DDE y servidor OPC funcionando. Lo que resta es instalar un cliente OPC en un terminal remoto y establecer la comunicación usando OPC. Pero antes debemos configurar DCOM, para garantizar el adecuado comportamiento de las aplicaciones implicadas.

2.2.4 CONFIGURACIÓN DE LA SEGURIDAD BÁSICA PARA OPC EN W2K³¹

OPC usa la tecnología DCOM en la comunicación ente el servidor y el cliente, como se ha visto anteriormente. Para esto, es necesario que DCOM esté configurado apropiadamente en el terminal en que se va a usar OPC, en cualquiera de los casos: uso de clientes solamente, servidor; o en el caso en que coexistan ambos.

³¹ Windows 200.

Por simplicidad, se le dará acceso total al grupo *Everyone*. Después que se tengan corriendo las herramientas OPC, si es preciso, este grupo se puede cambiar, para definir niveles de acceso a los diferentes tipos de usuarios. Véanse: [3], [8], [9], [10], [11].

CONFIGURACIÓN DE LAS PROPIEDADES POR DEFECTO DE DCOM

La forma de invocar a la herramienta de Windows que permite configurar DCOM, es mediante la instrucción “dcomcnfg” (Figura 2.12).

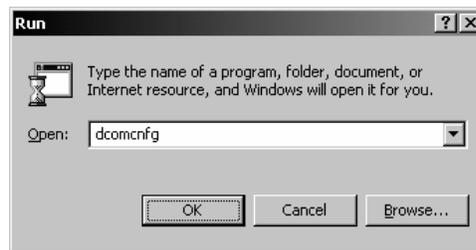


Figura 2.12 Para acceder a la herramienta de configuración de DCOM.

Al aparecer dicha herramienta, en la paleta correspondiente a las propiedades por defecto se habilita DCOM; y en el lugar correspondiente a las propiedades de la comunicación a través de DCOM, se pone el nivel de autenticación en *Connect* y el nivel de personificación en *Identify* (Figura 2.13).

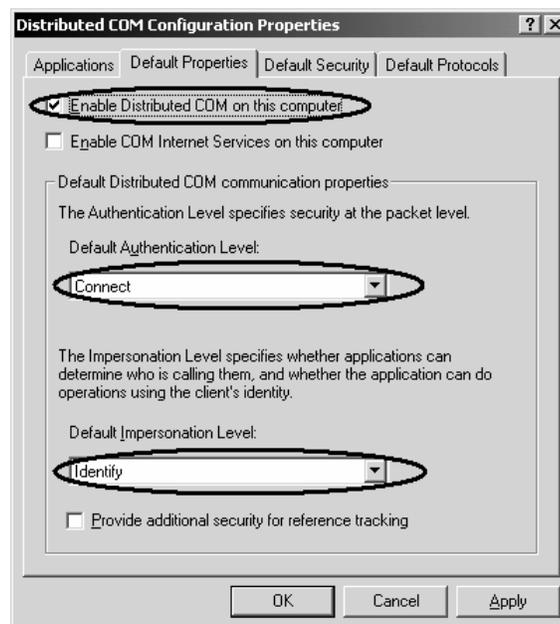


Figura 2.13 Configurando propiedades de DCOM, por defecto.

Estos son los niveles de autenticación que se pueden usar para los paquetes transmitidos:

- *None*: Ninguna seguridad se chequea durante la comunicación entre aplicaciones se usa cuando el nivel de personificación se pone a *Anonymous*.
- *Default*: Este nivel de autenticación lo define el servicio de autenticación instalado. En W2K este nivel es *connect*.
- *Connect*: El chequeo se realiza al comienzo de la comunicación.
- *Call*: El chequeo se realiza en cada llamada durante la comunicación.
- *Packet*: La identidad del emisor es encriptada, para asegurar su autenticidad.
- *Packet Integrity*: La identidad del emisor, así como su firma son encriptadas, para garantizar su autenticidad y la integridad del paquete de información.
- *Packet Privacy*: El paquete completo es encriptado, junto con la identidad y la firma del emisor.

Por su parte los niveles de personificación, que definen el nivel de acceso que le confiere la aplicación cliente al servidor, pueden ser:

- *Anonymous*: La aplicación servidor puede realizar tareas para el cliente, sin el conocimiento de esta.
- *Identity*: El servidor puede verificar la identidad de la aplicación cliente.
- *Impersonate*: La aplicación servidor puede actuar como cliente solo ejecutando tareas propias de los clientes. El servidor puede hacerse pasar por cliente solo en la computadora donde esta corriendo el servidor.
- *Delegate*: El servidor puede ejecutar tareas en otras computadoras, actuando como cliente. El servidor puede hacerse pasar por cliente en la computadora donde está corriendo y otras computadoras de al red³².

En la paleta correspondiente a la seguridad por defecto (Figura 2.14), se configuran los usuarios o grupos de estos que tendrán acceso a las aplicaciones, a activar estas y a configurar DCOM.

³² El servicio de autenticación de W2K no soporta este nivel.

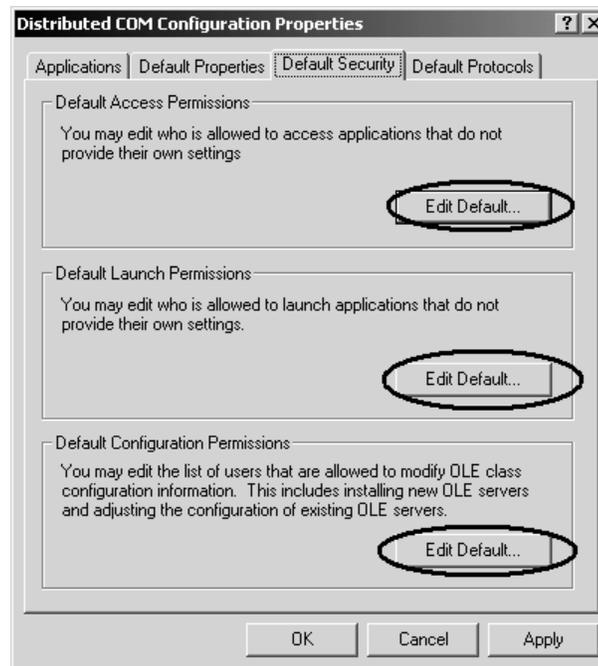


Figura 2.14 Para configurar seguridad DCOM, por defecto.

Los permisos de acceso se configurarán como sigue (Figura 2.15):

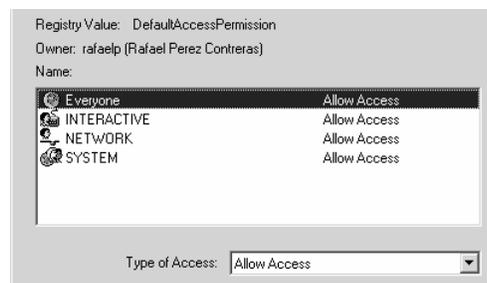


Figura 2.15 Permisos de accesos por defecto.

Los de los usuarios a los que se les permite activar las aplicaciones (Figura 2.16):

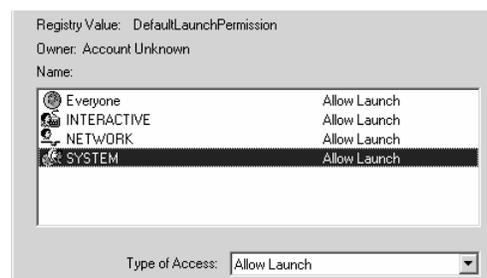


Figura 2.16 Permisos para activar DCOM, por defecto.

Y el de los usuarios a los que se les permite configurar DCOM (Figura 2.17):

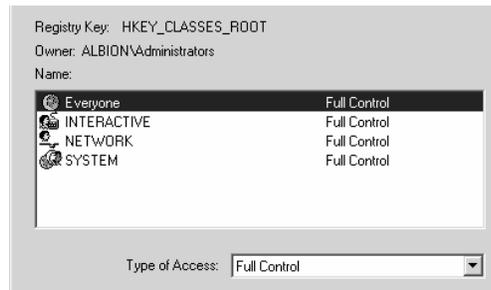


Figura 2.17 Permisos para configurar DCOM, por defecto.

Con lo hecho hasta ahora, se han configurado los permisos para todas las aplicaciones que usan DCOM. Ahora se pueden personalizar los permisos, para las diferentes aplicaciones.

CONFIGURACIÓN DE DCOM PERSONALIZADA PARA LAS DIFERENTES APLICACIONES

Esto se hace en la paleta *Application* (Figura 2.18).

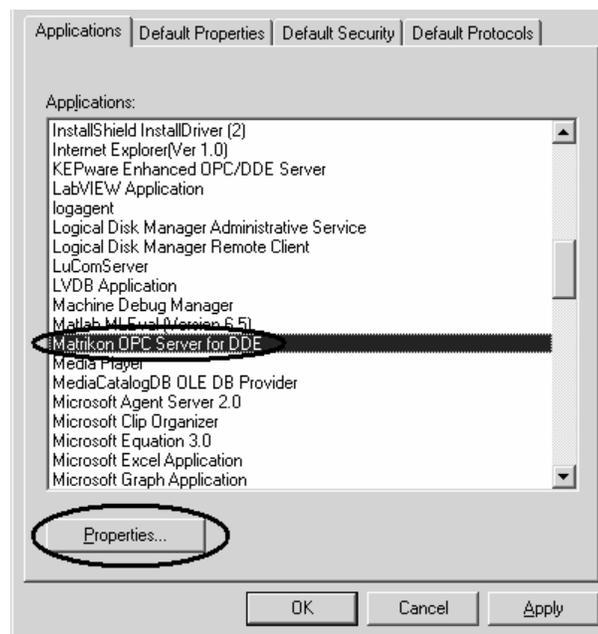


Figura 2.18 Para configurar DCOM de forma personalizada.

Seleccionando la aplicación que se quiere configurar y presionando en el botón *Properties*:

El nivel de autenticación de pone a *Connect* o a *Default*, ya que en las propiedad por defecto, ya se había configurado este punto (Figura 2.19).

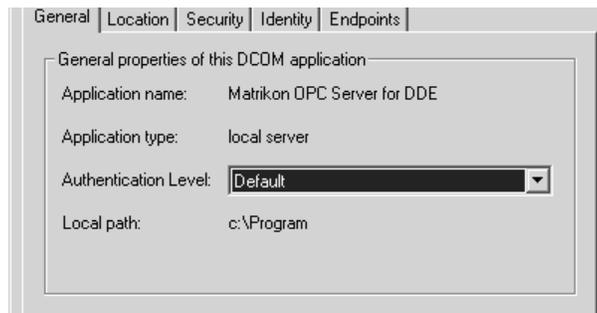


Figura 2.19 Configuración general personalizada.

En la paleta *Location*, se pone que la aplicación va a correr de forma local.

En la paleta de seguridad, se va a poner que todos son los que configuramos por defecto (Figura 2.20).

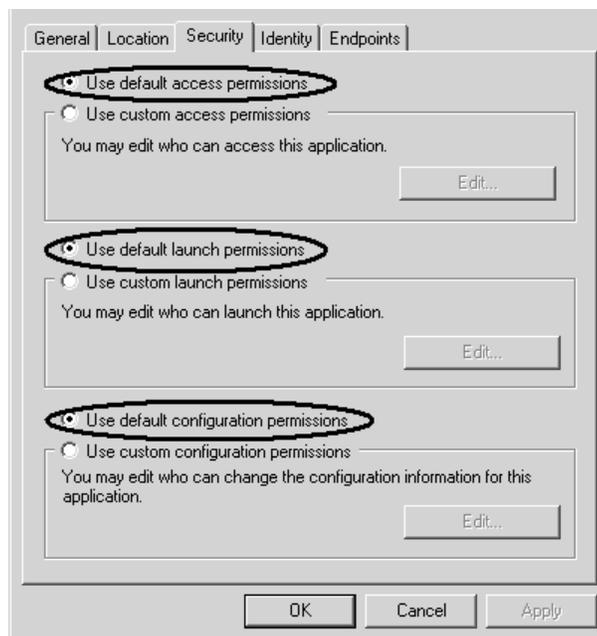


Figura 2.20 Configuración de permisos personalizados.

En la paleta correspondiente a la identificación se dan tres posibles opciones a tomar (Figura 2.21):

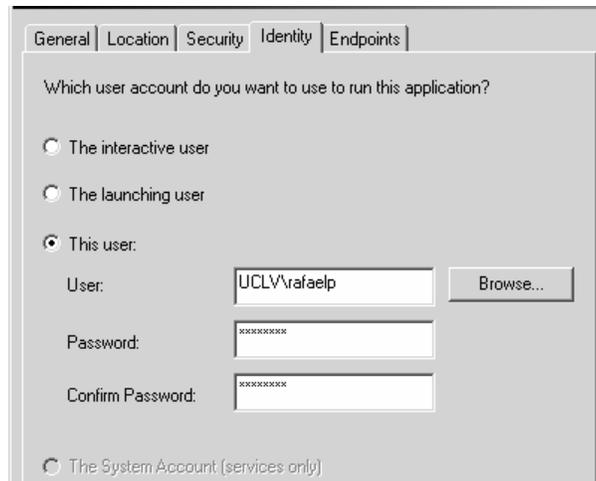


Figura 2.21 Configuración del tipo de identificación.

- *The Interactive User*: La aplicación utilizará el contexto de seguridad del usuario que esté registrado para autenticarse en el dominio. Si no hay usuario registrado, la aplicación no se activará.
- *The Launching User*: La aplicación utiliza el contexto de seguridad del usuario que la activa, para autenticarse en el dominio. El servicio de autenticación de W2K no valida una identificación que ha sido pasada por otra computadora de la red.
- *This User*: Especifica a la aplicación el usuario y la clave a utilizar para autenticarse en el dominio.

Al igual que se han personalizado los permisos para la aplicación *Matrikon OPC Server for DDE*; debe hacerse para *OPCEnum*, que como anteriormente se vio es la herramienta que permite a los clientes listar los servidores OPC disponibles en un terminal remoto.

De esta forma, ya se tiene configurado DCOM, de forma tal que no restringe ningún acceso; aunque ya se tiene la vía para configurar la Seguridad Básica OPC.

Ya se tiene todo lo necesario instalado y configurado, de parte del servidor OPC. Es hora de pasar a instalar y configurar el cliente remoto.

2.2.5 CONFIGURACIÓN DEL CLIENTE OPC

Una vez instalado el cliente *Matrikon OPC Explorer*, cuando se ejecuta desde el menú inicio, por defecto hace un listado de los servidores OPC disponibles de forma local (Figura 2.22).

Hacer la búsqueda de un servidor remoto no resulta un problema, como se ve solo se necesita navegar, o conocer el nombre de la computadora con la que se quiere establecer comunicación y el tipo de servidor; para especificarlos en el menú *Server*.

Como se aprecia a continuación existen tres menús que representa los elementos de un servidor OPC, vistos anteriormente: servidor, grupo, *item*. A través de estos menús se hace todo lo referente a la configuración de la conexión usando OPC.

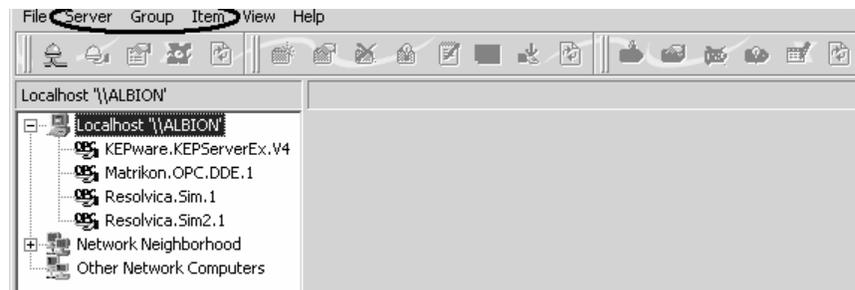


Figura 2.22 Matrikon OPC Explorer.

Las opciones que necesitamos para establecer la comunicación son: después que se tiene el servidor marcado en el navegador, *Connect* y *Add Group*, y en el caso que necesitemos especificar el nombre del terminal y el tipo de servidor, *Add/Connect Server* (Figura 2.23), en el menú *Server*.

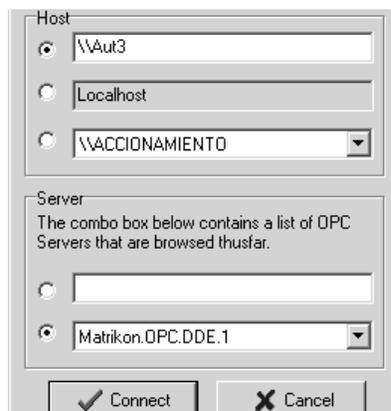


Figura 2.23 Para conectar el cliente a un servidor.

Después que se adiciona un grupo (Figura 2.24), es necesario activarlo y adicionarle *items*; esto es en el menú *Group*.

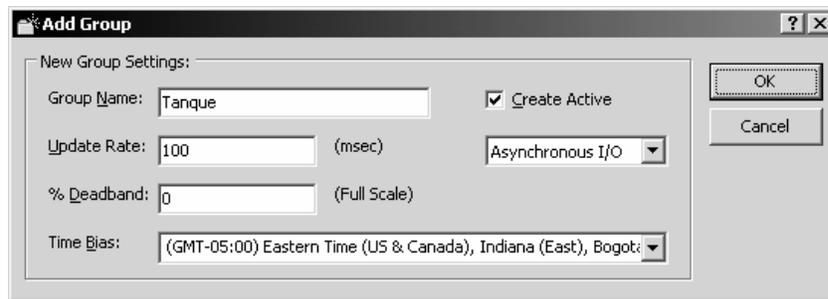


Figura 2.24 Para adicionar un grupo.

Al presionar la opción de adicionar *items*, se invoca la herramienta *Matrikon OPC Tag Estudio* (Figura 2.25), la cual es una poderosa herramienta a la hora de navegar en la configuración del servidor OPC, para adicionar los diferentes *items*. Una vez que se tienen adicionados los *items*, es preciso activarlos, si no se crean activos y opcionalmente, escribirle valores. Estas últimas opciones se encuentran en el menú *Item*.

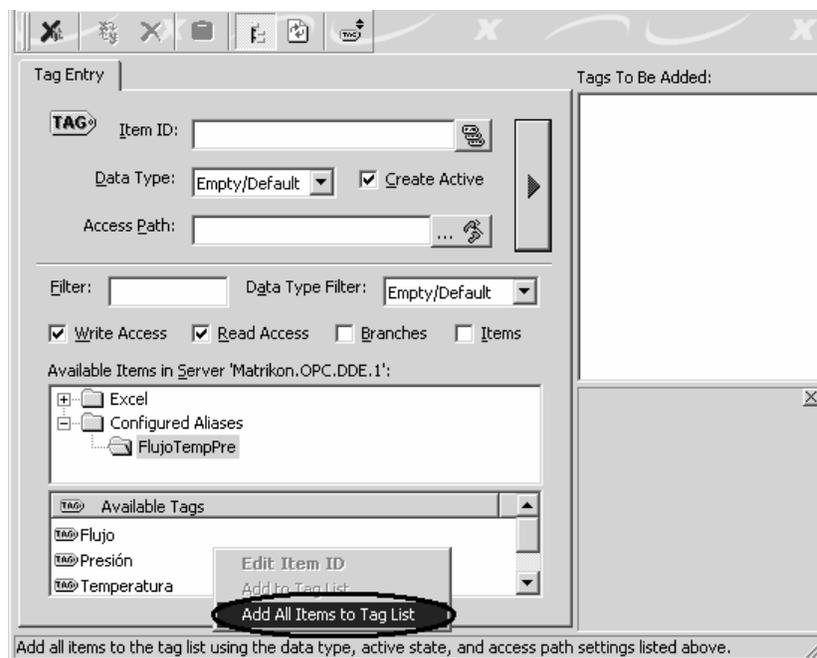


Figura 2.25 Matrikon OPC Tag Estudio.

Al adicionar los *items*, en el panel de *tags* para adicionar, estos aparecen con el símbolo “?” formando parte de su icono (Figura 2.26). Esto especifica que no ha sido validado; es decir, que no se ha establecido comunicación con el servidor.

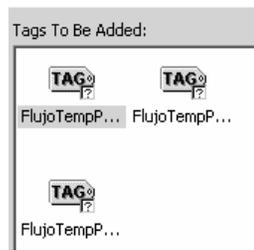


Figura 2.26 Tags para adicionar.

La forma de validar los *tags* de esta ventana, es a través de la opción *Validate Tags*, del menú *File*. Es entonces que cambia este símbolo, indicando que la comunicación se ha establecido bien (Figura 2.27), o que no se ha podido realizar (Figura 2.28); en cuyo caso hay que revisar en primer lugar que la hoja de cálculo esté abierta y la configuración de servidor.



Figura 2.27 Tags activados.



Figura 2.28 Tags con error.

Seguidamente, en el menú *File*, se activa la opción de actualizar y regresar al explorador, lugar donde ya se puede apreciar el valor de las variable que se habían definido anteriormente, en la hoja de cálculo (Figura 2.29).

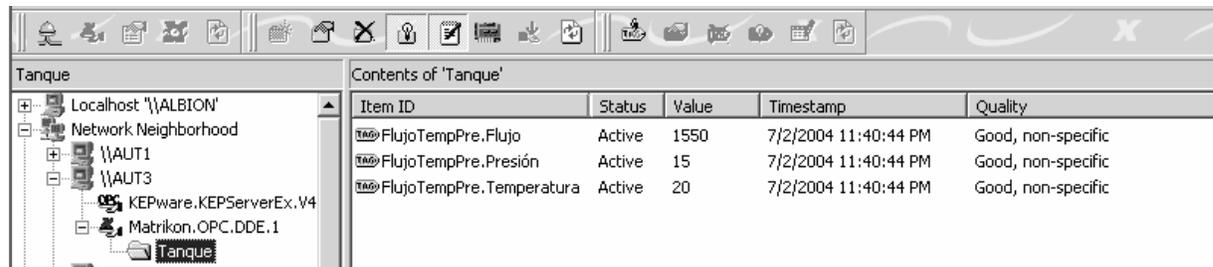


Figura 2.29 Cliente OPC mostrando los datos del documento Excel.

Dando clic derecho sobre el valor de la variable obtenido, se brinda la opción de cambiarlo, en cuyo caso, podemos verificar en la hoja de cálculo el cambio que se haya realizado.

2.3 IMPLEMENTACIÓN DE UNA RED OPC PRÁCTICA.

Hasta aquí se ha expuesto un conocimiento básico en lo referente a OPC y se ha realizado una práctica sencilla de comunicación. Ahora se vera la forma de llevar este conocimiento a la práctica mediante la transmisión de variables obtenidas en el laboratorio, a través de la tarjeta de adquisición PCI 1711.

Las variables a transmitir serán: el voltaje de un tacómetro, acoplado a un servomotor y la temperatura del local donde esta instalado el sistema. El voltaje del tacómetro se obtendrá a través del primer canal de la tarjeta de adquisición y la temperatura, a través del canal cero de dicha tarjeta, el cual toma como referencia un sensor de temperatura empotrado en esta. Estos datos serán trasmitidos mediante OPC y entonces podrán ser accedidos por cualquier cliente OPC desde cualquier maquina existente en la red,

Para la obtención de los datos desde la tarjeta de adquisición, se utilizará aplicación LabVIEW; el cual tiene, como se planteó anteriormente, la posibilidad de crear aplicaciones que funcionen como servidor DDE. Este dato brindado mediante el protocolo DDE, será tomado por la aplicación *Matrikon OPC Server for DDE*, para brindarlo a la red, a través del protocolo OPC. Los esquemas de conexión se muestran a continuación: (Figura 2.30) para el voltaje del tacómetro y (Figura 2.31) para la temperatura.

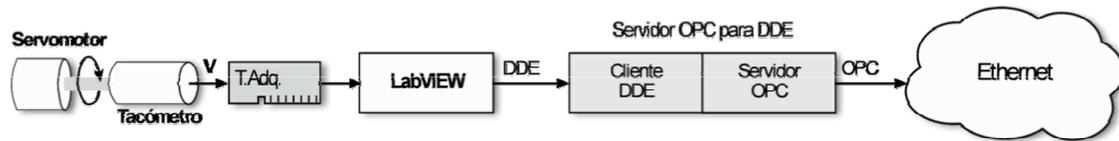


Figura 2.30 Esquema para la transmisión del voltaje del tacómetro.

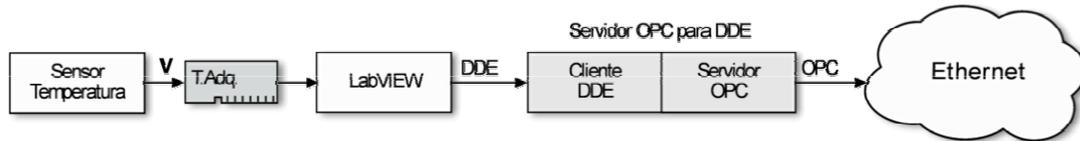


Figura 2.31 Esquema para la transmisión de la temperatura del local.

En los anexos se expondrá una síntesis de lo que representa LabVIEW y parte del trabajo que fue necesario hacer con este sistema [Anexos I y II].

2.3.1 IMPLEMENTACIÓN DE LA RED

LabVIEW incorpora gran cantidad de ejemplos que pueden ser parcialmente o completamente incorporados, de forma fácil en proyectos particulares. Por otra parte, en el disco de instalación que viene aparejado a la tarjeta de adquisición de datos [1], se brindan los VI necesarios para recuperar los datos provenientes de esta.

Con la ayuda de ambas fuentes de VI y la ayuda de LabVIEW [19], se confeccionaron los VI necesarios para la adquisición de dato y su transmisión a través de DDE.

Durante el proceso de adquisición de los datos, se obtuvieron los siguiente resultados con las aplicaciones confeccionadas para dicho propósito: (Figura 2.32) para el voltaje del tacómetro y (Figura 2.33) para la temperatura.

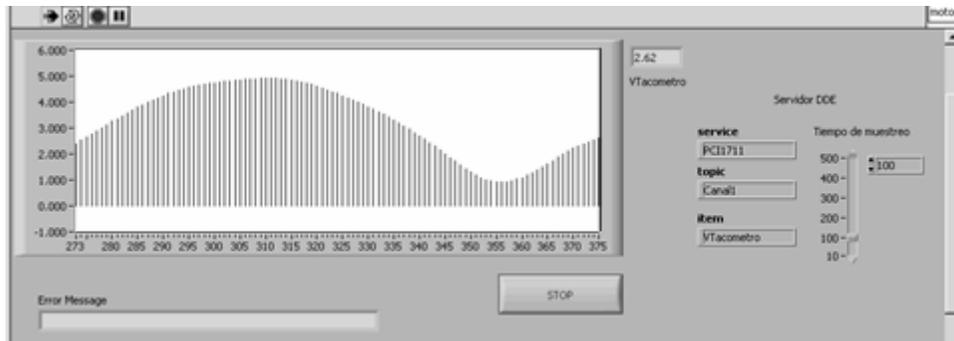


Figura 2.32 Adquisición del voltaje del tacómetro.

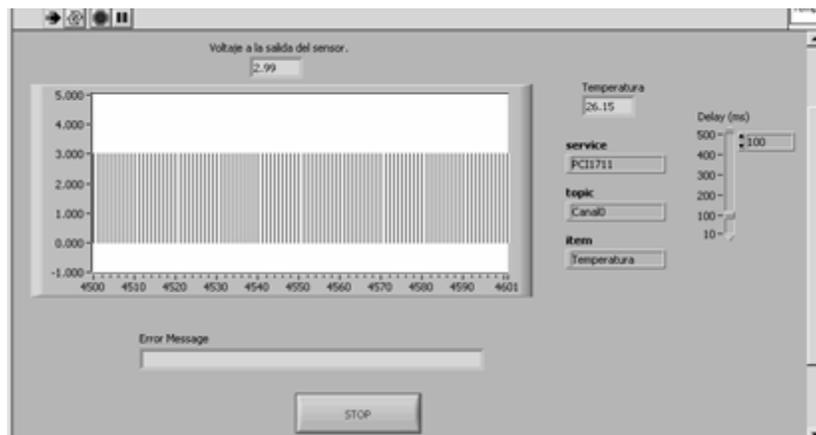


Figura 2.33 Adquisición de la temperatura.

Para la configuración del *Matrikon OPC Server for DDE*, de forma tal que se conecte con el servidor DDE, según lo que especifica DDE, y en correspondencia con las aplicaciones servidor DDE implementadas; la dirección del dato quedaría (Figura 2.34), para el voltaje del tacómetro conectado al servomotor y (Figura 2.35) para la temperatura del local.

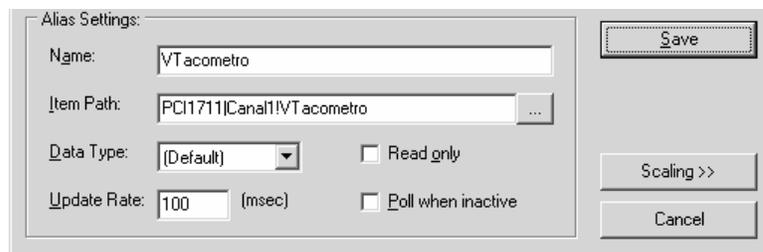


Figura 2.34 Configuración del item que referencia el voltaje del tacómetro.

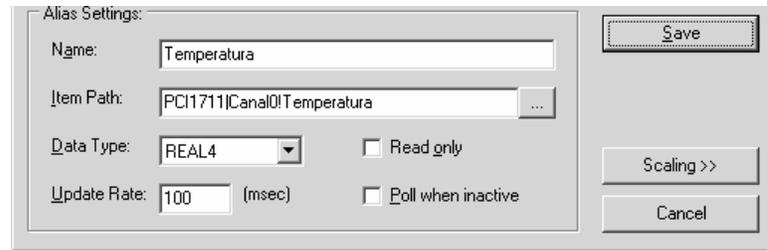


Figura 2.35 Configuración del item que referencia a la temperatura.

La aplicación *Matrikon OPC Explorer*, seleccionada en este caso, al igual que en el anterior, para recuperar los datos; muestra los mismos de la siguiente forma: (Figura 2.36) para el voltaje de tacómetro y (Figura 2.37) para la temperatura.

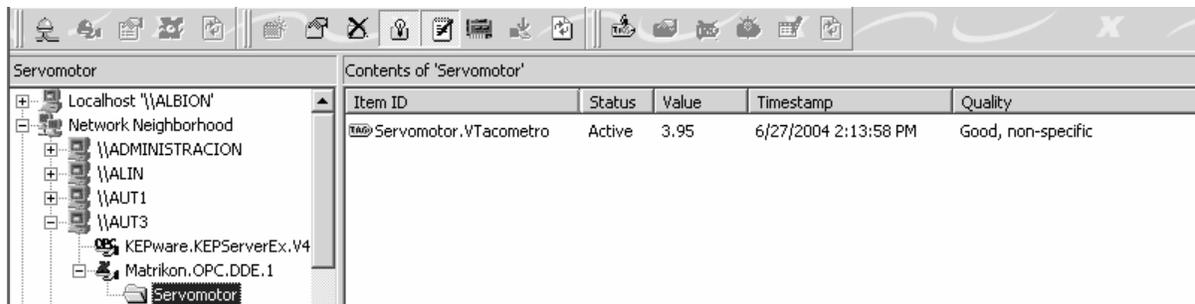


Figura 2.36 Cliente OPC mostrando el dato del voltaje del tacómetro.

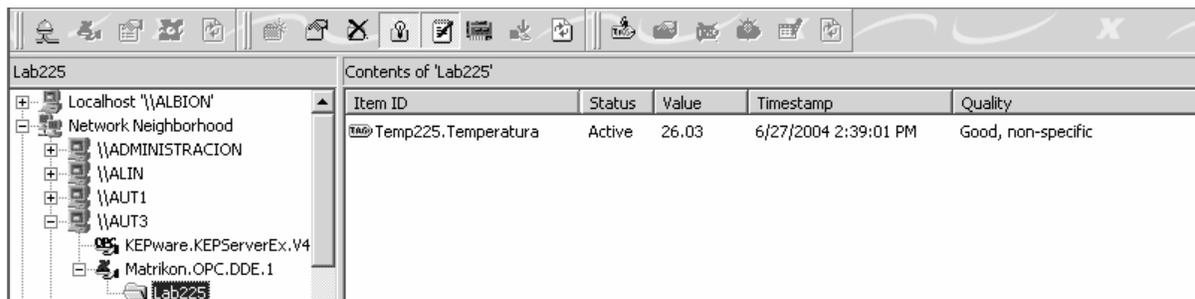


Figura 2.37 Cliente OPC mostrando el dato de la temperatura.

2.4 CONSIDERACIONES GENERALES

En este capítulo se ha expuesto la forma de instalar y configurar un cliente y un servidor OPC a través de las herramientas brindadas por el proveedor Matrikon. Se hizo referencia a la forma de configurar la seguridad, usando la seguridad básica de OPC; que es la que se basa en el modelo de Windows NT. Se explico la forma de usar como servidor DDE la aplicación

Microsoft Excel, por considerarse sencilla. Esto debido a que el enfoque que se hizo en ese caso fue con fines didácticos.

Seguidamente se enfocó el trabajo a la utilización de OPC para transmitir datos de interés industrial como son el voltaje de un tacómetro, conectado a un servomotor y la temperatura de un local. Resultando estas las pruebas realizadas como colofón de este trabajo.

De haber utilizado las herramientas de Kepware, lo único a tener en cuenta sería que; como dicha herramienta permite varios tipos de conexión, debido a que incluye varios *drivers*; se debería declarar un canal, para un tipo de conexión y seguidamente, un dispositivo, al cual se le agregan los grupos. Por lo demás toda la metodología sigue como se ha descrito. Esto para configurar el servidor; porque para recuperar datos a través de un cliente OPC, el procedimiento no difiere de lo que se ha expuesto. Parecido ocurre con las herramientas de otros proveedores; que solo es necesario un conocimiento básico sobre OPC, para poder utilizarlas. Esto representa una de las grandes virtudes de OPC.

CONCLUSIONES

Con el análisis realizado sobre OPC, la selección de las herramientas para la implementación de la red, su configuración y las pruebas realizadas, se ha hecho posible arribar a las siguientes conclusiones:

- El estudio realizado y los resultados obtenidos permiten la incorporación de la tecnología OPC, con fines docentes en la especialidad de Automática.
- El presente informe y los materiales referenciados constituyen fuente de consulta para la docencia.
- Las herramientas utilizadas no solo son aplicables a las pruebas de laboratorio realizadas, sino que se pueden integrar en diferentes asignaturas.

Una vez realizadas las tareas del trabajo de diploma, se puede plantear que se han cumplido los objetivos propuestos satisfactoriamente.

RECOMENDACIONES

- Utilizar en la docencia algunos elementos que no son solo beneficiosos para una mejor comprensión de OPC, sino para la preparación del estudiantado; como son los conceptos de COM, DCOM, ActiveX, entre otros, en algunas de las asignaturas que abordan el tema de la programación.
- Integran el estudio de OPC en la docencia, realizando prácticas en el laboratorio, para lo cual podría utilizarse el presente trabajo. Un ejemplo de donde podría insertarse este estudio, es en la asignatura “Computación Aplicada.”

REFERENCIAS BIBLIOGRÁFICA

- [1] ADVANTECH. eAutomation [CD de instalación]
- [2] Ballesteros, Marialina. Servidores/Clientes de datos para Redes Ethernet en Tiempo Real, 2003
- [3] Barillere, Renaud. OPC DCOM Settings Document [pdf], versión: 2.0, 12 de diciembre del 2003.
- [4] Brockschmidt, Kraig. Inside OLE 2, abril de 1995.
- [5] Donayre Sánchez, Javier G. Total Automatización, disponible en: <http://www.industriaaldia.com/articulos/26-5.htm>, consultado en abril del 2004.
- [6] Fundación OPC, La. Información para los miembros [pdf], julio del 2002.
- [7] Horstmann, Markus; Kirtland, Mary: DCOM Architecture, disponible en http://msdn.microsoft.com/library/en-us/dndcom/html/msdn_dcomarch.asp, julio de 1997, consultado en abril del 2004
- [8] Iconics. DCOM Quick Start [pdf], versión: 5.4, febrero de 1999.
- [9] Iconics. Distributed Component Object Model White Paper [pdf], versión: 5.2, 1999.
- [10] Intellution. OPC DCOM White Paper [pdf], 1998.
- [11] Iocomp Software Incorporated. DCOM Configuration for Windows NT4, Windows 2000, and Windows XP [pdf], impresión inicial agosto de 2003.
- [12] KEPware. Enhanced OPC/DDE Server [aplicación], versión: 4.13.287.
- [13] KEPware. Link Master [aplicación], versión: 1.18.71.
- [14] Matrikon. OPC Explorer [aplicación], versión: 3.1.0.58 y 3.2.1.150.
- [15] Matrikon. OPC Server for DDE [aplicación], versión: 2.1.0.65.
- [16] Merz. MOC OPCToDDE [aplicación], versión: 2.03.02.03.
- [17] Merz. MOS DDEToOPC [aplicacion], versión: 2.01.07.06.

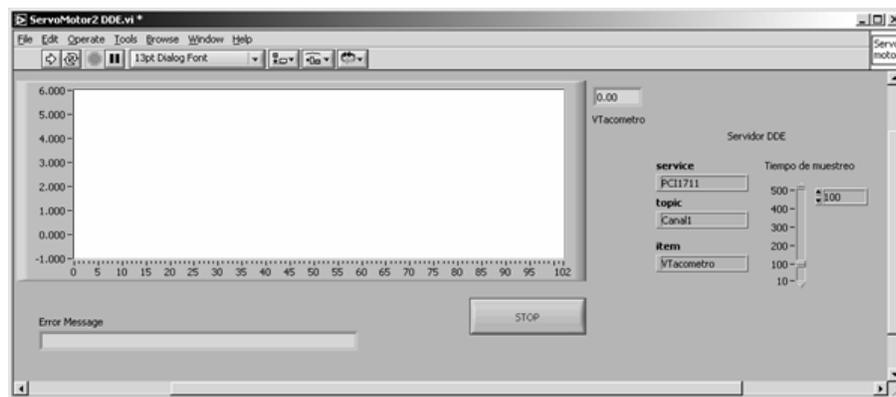
-
- [18] Microsoft Corp. COM Programmer's Reference, 1998.
- [19] National Instruments Co. LabVIEW Help [ayuda electrónica], julio del 2002.
- [20] Nuez Arencibia, Sebastián. OLE–DDE–OPC, disponible en: http://www.diea.ulpgc.es/users/aurelio/int_equipos/trab9899/ole_dde_opc/index.html, consultado en abril del 2004.
- [21] OPC Foundation, The. Data Access Custom Interface Standard [pdf], versión: 3.00, 4 de marzo del 2003.
- [22] OPC Foundation, The. OPC Alarms and Events Custom Interface Standard [pdf], versión: 1.10, 2 de octubre del 2002.
- [23] OPC Foundation, The. OPC Batch Automation Interface Specification [pdf], versión: 1.0, 19 de julio del 2001.
- [24] OPC Foundation, The. OPC Batch Custom Interface Specification [pdf], versión: 2.0, 19 de julio del 2001.
- [25] OPC Foundation, The. OPC Common Definitions and Interfaces [pdf], versión: 1.0, 27 de octubre de 1998.
- [26] OPC Foundation, The. OPC Complex Data Specification [pdf], versión: 1.00 Released December 10, 2003
- [27] OPC Foundation, The. OPC Data eXchange Specification [pdf], versión: 1.0, 5 de marzo del 2003.
- [28] OPC Foundation, The. OPC Historical Data Access Automation Interface Standard [doc], versión: 1.0, 26 de enero del 2001.
- [29] OPC Foundation, The. OPC Historical Data Access Specification [pdf], versión: 1.20, 10 de diciembre del 2003.
- [30] OPC Foundation, The. OPC Security Custom Interface [pdf], versión: 1.0, 17 de octubre del 2000.
- [31] OPC Foundation, The. OPC XML-DA Specification [pdf], versión: 1.0, 12 de julio del 2003.

-
- [32] OPC Foundation, The. What is OPC?, disponible en: http://www.opcfoundation.org/01_about/01_what_is.asp, consultado en abril del 2004.
- [33] OPC Task Force. OPC Technical Overview [pdf], extraído de un suplemento especial de la revista: Start Magazine, disponible en: <http://www.startmagazine.com/v2n1opc0.htm>, publicado en enero/febrero de 1998.
- [34] PCC, Resolución Económica del V Congreso del Partido Comunista de Cuba, disponible en: <http://www.pcc.cu/resoluci.htm>, consultado en abril del 2004.
- [35] Resolvica. OPCEx Microsoft Excel Add-In [paquete] versión: 3.0
- [36] Softing. OPC Client [aplicación], versión: 2.0.
- [37] World Wide Web Consortium: Extensible Markup Language; 10-06-2000
<http://www.w3.org/TR/REC-xml/>.

ANEXO I APLICACIONES SERVIDOR DDE

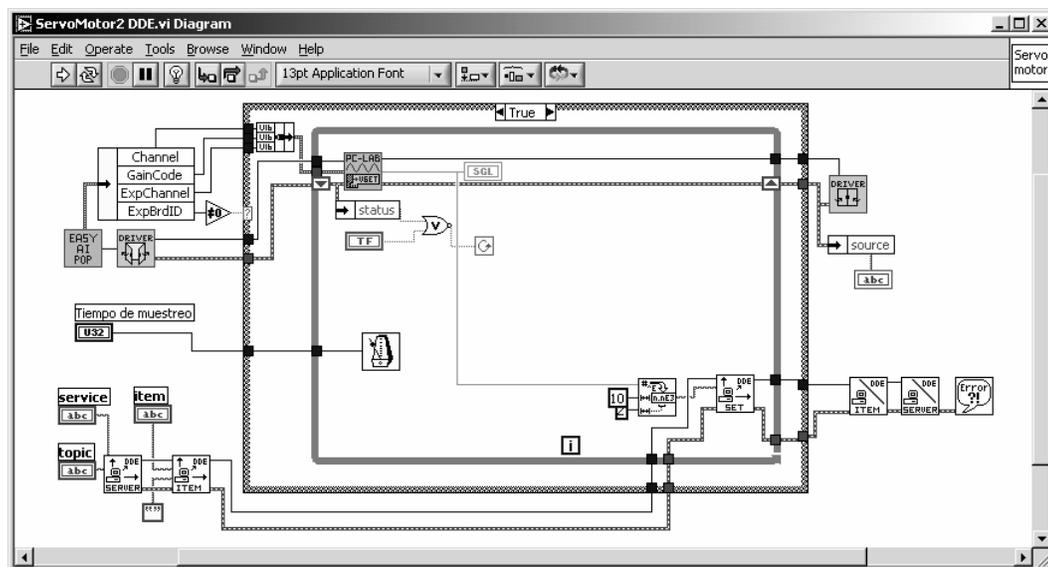
CONFECCIONADAS EN LABVIEW

El panel frontal del VI destinado a adquirir el dato del canal uno, que es el canal asignado al voltaje del tacómetro del servomotor, quedó como sigue (Anexo I - Figura 1):



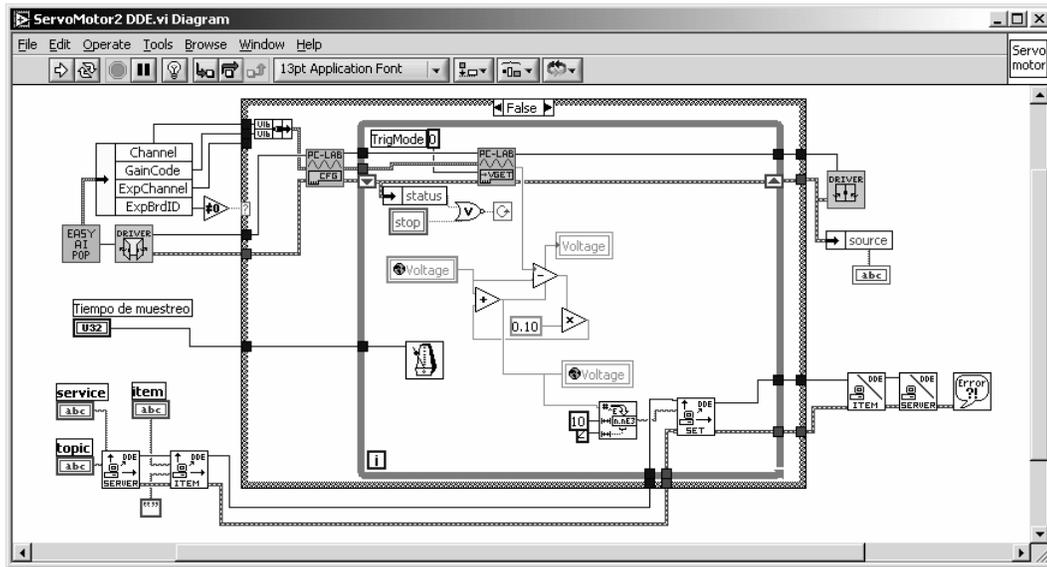
Anexo I - Figura 1: Aplicación servidor DDE para el voltaje del tacómetro.

El diagrama de bloques contiene una estructura Case de verdadero o falso. En caso de verdadero, queda como sigue (Anexo I - Figura 2):



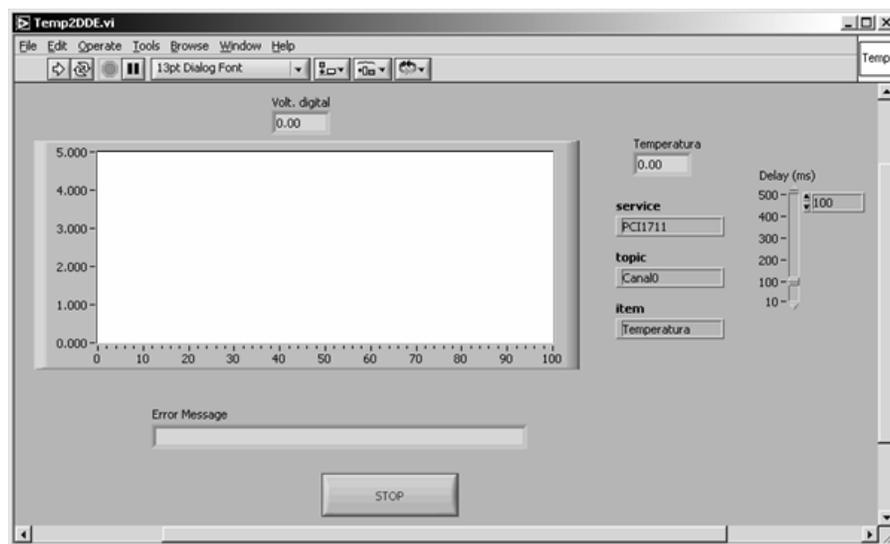
Anexo I - Figura 2: Diagrama de bloques para cuando el Case toma verdadero.

Y en caso de falso (Anexo I - Figura 3):



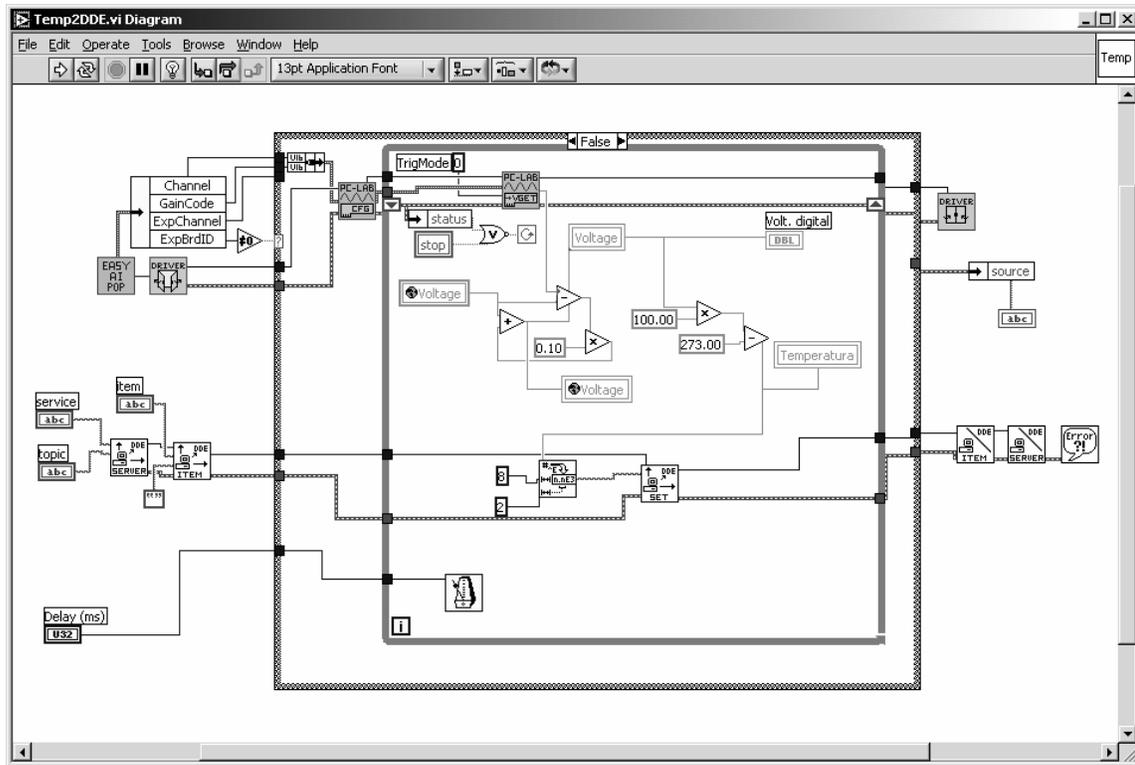
Anexo I - Figura 3: Diagrama de bloques para cuando el Case toma falso.

Para el caso en que se mide temperatura, esto es por el canal cero, fue necesario hacer la conversión de voltaje que brinda el sensor, a temperatura. El panel frontal se diseñó como sigue (Anexo I - Figura 4):



Anexo I - Figura 4: Aplicación servidor DDE para la temperatura.

Y los diagramas de bloque: para el caso en que el Case toma verdadero (Anexo I - Figura 5),



Anexo I - Figura 6: Diagrama de bloques para cuando el Case toma falso.

Un análisis no muy detallado permite diferenciar la parte de los diagramas que corresponden a la adquisición y la que corresponde a la de servidor DDE.

ANEXO II LABVIEW EN SÍNTESIS

LabVIEW es un lenguaje de programación con una marcada diferencia con los lenguajes de programación tradicionales. En primer lugar, es un compilador gráfico que usa iconos en lugar de texto y por otro lado, en LabVIEW es el flujo de datos el que determina la ejecución, en lugar de las acostumbradas instrucciones.

Con LabVIEW se construyen interfaces graficas de usuario, usando un conjunto de objetos y herramientas. Este es el panel frontal. A este se le adiciona código, en forma de representaciones gráficas de funciones, para controlar los objetos del panel frontal. Se conforma así el diagrama de bloques. El cual, si es organizado correctamente, recuerda un diagrama de flujo.

Los ficheros que son creados con LabVIEW se denominan Instrumentos Virtuales (VI). Cada VI contiene un panel frontal, un diagrama de bloques y un icono y conector. El icono y conector forman parte de la interfaz del VI, cuando este se integra como subVI en otro VI.