Aplicación para el monitoreo y control de procesos industriales basada en el estándar de comunicaciones OPC

Angel Villegas[(1)], Isaac Herrera[(1)], Gerardo Gómez[(1)], José Rodríguez[(2)], Edgar Lugo[(1)], José Pacheco[(1)]
[(1)] Centro de Procesamiento de Imágenes, Escuela de Ingeniería Eléctrica,
Facultad de Ingeniería, Valencia, Venezuela.
[(2)] Universidad Nacional Experimental de la Fuerza Armada Nacional,
Escuela de Ingeniería Electrónica, Maracay, Venezuela.
Email: avillegas@uc.edu.ve, ,ealugo@uc.edu.ve, jpacheco@uc.edu.ve, isaac.herrera@ve.nestle.com,
ing.gerardo2007@hotmail.com, jmrs@roso-control.com

Resumen

En este trabajo se desarrolló una aplicación de software tipo cliente capaz de realizar actividades de monitoreo y control de procesos industriales empleando el protocolo de comunicación "OLE para control de procesos", OPC. Se utilizó el lenguaje C# y la metodología *Extreme Programming* (XP). El software permite al usuario, interactuar con datos que pueden provenir de un proceso real o de una simulación, a través de algoritmos de control personalizados realizados en un lenguaje de fácil manejo (VBScript y JavaScript), y adicionalmente genera un registro histórico de variables OPC que puede ser recuperado en cualquier momento. Al fundamentarse en el estándar OPC, es posible agregar dispositivos de diferentes fabricantes a medida que estos son adquiridos e incorporados al proceso. El programa puede ser utilizado en aplicaciones de automatización de procesos e instrumentación virtual. La operación del sistema fue verificada de forma local y remota por medio de una red LAN. La aplicación desarrollada presenta interfaces amigables y una arquitectura abierta, capaz de crecer o adaptarse según las necesidades cambiantes de la empresa o instalación industrial.

Palabras clave: Cliente OPC, algoritmos personalizados de control, registro histórico, OPC.

Software application for industrial process monitoring and control based on the Object Linking and Embedding for Process Control (OPC) communications standard

Abstract

In this investigation a client software application has been developed which is able to carry out monitoring and control of industrial processes using the communication protocol "OLE for process control", OPC. C# language and Extreme Programming (XP) methodology were employed. Through user-defined control algorithms carried out in an easy to handle scripting language (VBScript and JavaScript), users can manipulate real process data or simulated data. The software also generates a historical data record of selected variables. This historical data can be recovered at any moment. Being based on OPC standard, it is possible to add different manufacture devices as they are acquired and incorporated into the industrial process. The software can be used for process automation or virtual instrumentation of industrial processes. System functionality was tested for local and remote operation.

Keywords: OPC client, user-defined control scripts, historical data record, OPC.

1. INTRODUCCIÓN

Una de las actividades más importantes en las industrias modernas es el monitoreo y control de las variables asociadas a sus procesos de producción. En vista que las computadoras son herramientas capaces de almacenar, procesar y presentar información en forma atractiva y confiable, la tendencia en las industrias modernas es asociar sus procesos automatizados a programas que posean un ambiente en el cual el

usuario pueda tener acceso para monitorear y modificar los distintos elementos que conforman su sistema de control [1].

Entre las dificultades de las operaciones de control industrial, resalta la de compartir información tanto entre los distintos dispositivos de campo como con el resto de la empresa o instalación industrial, ya que, es muy posible que los dispositivos transmitan sus datos usando distintos protocolos, siendo generalmente estos incompatibles entre sí [2]. Esta situación limita la operatividad entre equipos y aplicaciones de diferentes fabricantes, obligando a la utilización de medios de interfaz entre los distintos sistemas; esto además de propiciar el encarecimiento de los mismos crea una marcada dependencia, tanto con un determinado fabricante como con la tecnología utilizada en la planta o instalación [3].

Esta diversificación ha obligado a los desarrolladores de Sistemas de Control y Adquisición de Datos (SCADA) a incorporar diferentes tipos de controladores, buscando incluir a diversos fabricantes [2]. Todo esto ha evidenciado la necesidad de crear una norma de intercambio de datos a nivel de planta, desencadenando el surgimiento de la tecnología OLE (Object Linking and Embedding / Enlace e Inserción de Objetos), denominada OPC (OLE for Process Control / OLE para Control de Procesos). Progresivamente OPC ha sido incorporado en la mayoría de los equipos de control modernos por lo que el desarrollo de una aplicación software para el control de procesos industriales debe incluir soporte a dicho protocolo [3].

El desarrollo de productos basados en el estándar OPC, así como en las metodologías actuales de diseño de software, garantizarán en un futuro cercano una alta compatibilidad entre plataformas tecnológicas en el área de la automatización industrial [4], permitiendo a su vez, una independencia y generalización que liberará a las industrias de utilizar sólo la plataforma de programación propia del fabricante y supondrá un abaratamiento en la inversión, debido a que ahora, el usuario tendrá mayores alternativas para escoger los equipos con la tecnología que considere más conveniente a su proceso.

En este trabajo se desarrolló una aplicación de software tipo cliente para el monitoreo y control de procesos industriales por computadora capaz de recibir, visualizar y manipular de forma simultánea información proveniente de diferentes servidores de datos basados en la especificación de acceso a datos OPC. La aplicación aprovecha las características de interoperatividad inherentes al estándar OPC para efectuar acciones de monitoreo y control de forma local o remota en ambientes industriales heterogéneos.

2. MARCO CONCEPTUAL

2.1 Especificaciones OPC

El OPC, (OLE para el Control de Procesos) es una especificación técnica no propietaria definida por la *OPC Foundation* (http://www.opcfoundation.org) y consiste en un sistema de interfaces estándar basado en OLE/DCOM de Microsoft [5]. Con OPC es posible intercambiar información entre dispositivos industriales con sistemas de información o aplicativos de escritorio. En otras palabras, OPC permite desarrollar de manera práctica y eficiente aplicaciones que pretendan comunicarse directamente con equipos industriales controlados por controladores lógicos programables (PLC) o computadores [5]. OPC extiende el concepto original del modelo COM y de los enlaces OLE adaptados a los requerimientos de la automatización, control y monitoreo de los procesos industriales.

En cooperación con Microsoft, una fuerza constituida por cinco empresas, Intellution™, Opto 22™, Fisher-Rosemount™, Rockwell Software e Intuitiv Software, nace *OPC Foundation* en Mayo de 1995 [6]. Este grupo de empresas pretendía definir una serie de especificaciones basadas en COM/DCOM, destinadas a dar solución al problema de la poca estandarización e incompatibilidad entre plataformas en el área de la instrumentación, control y automatización industrial [7]. El primer borrador de las mismas fue completado al final de 1995, gracias a la colaboración de otras 90 compañías a lo largo del mundo, las cuales comprobaron estas especificaciones [6]. El primer conjunto oficial de especificaciones se completó en Agosto de 1996 y fue publicado bajo el nombre *OPC Specification*. Actualmente se le conoce como especificación OPC de acceso a datos [8].

La motivación principal del surgimiento del OPC se fundamentó en la necesidad de comunicar las distintas y numerosas fuentes de datos en un proceso industrial como los dispositivos de medición en campo, los dispositivos de supervisión de procesos y las bases de datos en las salas de control [3]. Para lograr

esto, inicialmente los fabricantes utilizaban una arquitectura basada en controladores, tal como se muestra en el esquema de la Figura 1.

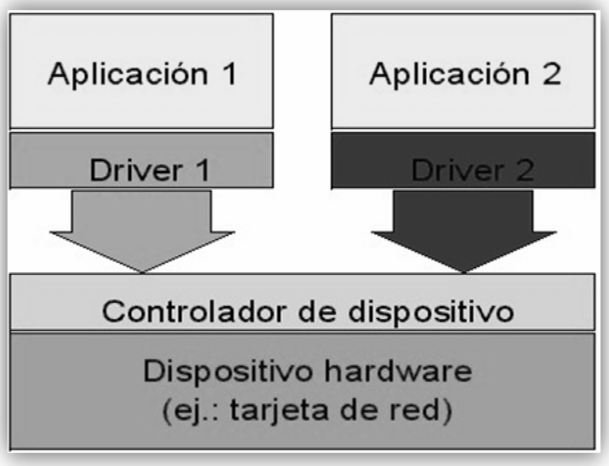

Figura 1. Modelo de arquitectura de automatización industrial basado en controladores [5].

De acuerdo a este esquema, las aplicaciones de software de control industrial acceden a los datos de los dispositivos de control utilizando controladores desarrollados específicamente para los equipos existentes [9]. Esta arquitectura, conduce a una serie de problemas como se indica en [3]:

- **Duplicación de esfuerzo:** Se deben escribir programas controladores o *drivers* específicos para el hardware de un vendedor particular.
- **Inconsistencias entre vendedores de controladores:** Las características de hardware no son soportadas por todos los controladores de dispositivos causando conflictos de compatibilidad entre estos.
- **Poco soporte para cambios en características de hardware:** Un cambio en las capacidades del hardware puede ocasionar conflictos o fallas en los controladores.
- **Conflictos de acceso:** Esta situación se presenta debido a que dos paquetes de software no pueden acceder simultáneamente al mismo dispositivo, porque cada uno contiene controladores independientes. Los fabricantes de hardware procuran resolver estos problemas desarrollando nuevos controladores, pero son obstaculizados por diferencias en los protocolos del cliente. No se puede desarrollar un controlador eficiente que pueda ser utilizado por todos los diferentes tipos de clientes.

En estas circunstancias la complejidad de realizar aplicaciones industriales era elevada, porque no existía una forma estándar de definir las conexiones sin depender del tipo de dispositivo. El OPC eliminó este problema estableciendo una interfaz de comunicación común, lo cual ha beneficiado enormemente el desarrollo de aplicaciones HMI (*Human Machine Interface*) y sistemas SCADA [5]. Así, el uso de la tecnología OPC le brinda a los fabricantes de aplicaciones (servidores y/o clientes) un mayor rango de operación al no estar limitado por la complejidad y funcionalidad de los componentes de hardware, de esta manera los componentes más diversos de diferentes fabricantes pueden trabajar juntos sin requerirse programación adicional para adaptar la interfaz entre sí. OPC emerge entonces como una tecnología conveniente y eficiente para el enlace de componentes de automatización y sus respectivos equipos de control.

La Figura 2 muestra el modelo de arquitectura simplificado de un sistema automatizado basado en OPC.

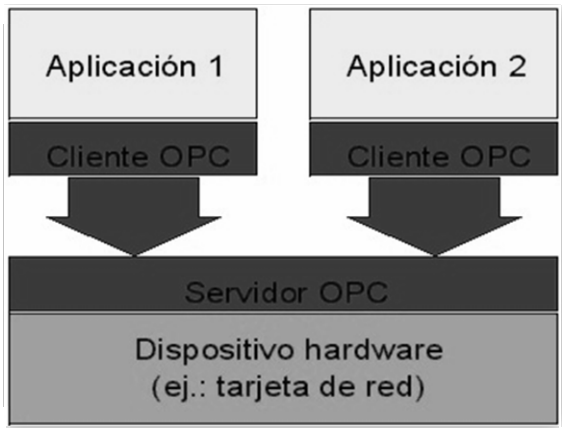

Figura 2. Modelo de arquitectura de automatización industrial basado en OPC [5].

La utilización de este modelo ofrece varias ventajas, entre estas destacan [3]:

- Se establece una línea divisoria entre los fabricantes de hardware y software dando flexibilidad a los clientes para utilizar tecnologías heterogéneas en sus aplicaciones.
- Los fabricantes de hardware tienen que hacer solamente un conjunto de componentes de software para que los clientes los utilicen en sus aplicaciones.
- Los desarrolladores de software no tienen que reescribir controladores debido a cambios en caracterís-

ticas o adiciones en un hardware.
- Los clientes tendrán más opciones con las cuales puedan desarrollar diversos sistemas de aplicación a nivel industrial.
- Los desarrolladores de servidores pueden actualizar sus productos en conformidad a los cambios en el hardware sin afectar al resto del sistema, permitiendo mantener una plataforma eficiente de adquisición de datos.

Las especificaciones OPC están disponibles en la organización OPC en forma gratuita y definen un conjunto de normas aplicables a diferentes tipos de interfaz utilizados en aplicaciones de automatización industrial. La Tabla 1, muestra las especificaciones OPC más importantes.

La arquitectura OPC es del tipo cliente – servidor como en todas las aplicaciones COM, donde el componente servidor suministra una interfaz a los objetos OPC, a la vez que administra y gerencia dichos objetos [4]. Una aplicación cliente OPC se comunica con un servidor OPC al invocar las funciones de estas interfaces OPC. La utilización de tecnología OLE (DCOM), permite que los clientes puedan tener acceso a servidores de datos locales o remotos.

Para la implementación de los clientes y servidores de datos OPC se han definido dos tipos de interfaz: la **Interfaz Personalizada** (*Custom Interface*) descrita en la especificación que lleva este mismo nombre [10] y la **Interfaz de Automatización** descrita en la *Data Access Automation Interface Standard* [11].

La mayoría de los clientes OPC se construyen utilizando la interfaz de automatización ya que esta admite ser desarrollada en lenguajes como Visual Basic 6.0, Borland Delphi y recientemente en plataforma .NET utilizando la librería COM-Interop [5]. Inclusive las herramientas como Microsoft Excel, deben usar las interfaces de automatización. Los clientes que deseen implementar la interfaz personalizada deben ser desarrollados en el lenguaje Visual C++.

2.2 Metodologías ágiles de desarrollo de software

En febrero de 2001, tras una reunión celebrada en UTA, EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software [12]. Su objetivo fue proyectar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente, respondiendo a los cambios que puedan surgir a lo largo del proyecto. Tras esta reunión se creó "*The Agile Alliance*" (http://www.agilealliance.com) una organización, sin fines de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es el Manifiesto Ágil, un documento que resume la filosofía "ágil".

2.2.1 Extreme programming XP

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el traba-

Tabla 1. Principales especificaciones OPC existentes en la actualidad.

Especificación	Contenido
OPC Introducción	Descripciones generales y aplicación de las especificaciones OPC.
OPC Definiciones comunes e Interfaces	Definiciones de términos utilizados en las especificaciones OPC.
OPC Acceso de datos	Definición de un interfaz para la lectura y escritura de datos en tiempo real.
OPC Alarmas y Eventos	Definición de un interfaz para el monitoreo de alarmas y eventos.
OPC Datos Históricos	Definición de un interfaz para el acceso de datos históricos.
OPC Seguridad	Definición de un interfaz para la utilización de políticas de seguridad.
OPC y XML	Integración del OPC y el XML para construcción de aplicaciones Web.
OPC Intercambio de Datos (DX)	Comunicaciones entre servidores – servidores de datos en los procesos.
OPC Interfaz y comandos de Ejecución	Definición de un interfaz para el intercambio de comandos y su ejecución.

jo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico [13].

Las prácticas y reglas de la Programación Extrema se agrupan en cuatro etapas [14]: Planificación, Diseño, Codificación y Pruebas.

3. METODOLOGÍA UTILIZADA

El desarrollo del software, se fundamentó en el uso de metodologías ágiles, específicamente en **Extreme Programming (Programación Extrema)**, debido a que ésta cumple con todos los requerimientos para realizar una programación de calidad, estableciendo una relación eficaz de trabajo en equipo y evaluación constante del funcionamiento óptimo del producto [14]. Por otra parte, se incorporó el uso de diagramas UML (Lenguaje Unificado de Modelado) para documentar el funcionamiento y las características que la aplicación presentará al usuario. Específicamente, fueron utilizados diagramas de Casos de Uso, ya que estos representan al sistema desde el punto de vista del usuario, es decir, describen un uso del sistema y cómo interactúa con el usuario [15].

Para describir las funcionalidades del software desde el punto de vista del usuario, se dividió el programa en cinco casos de uso los cuales representan las acciones más importantes realizadas por éste y permiten definir el escenario de funcionamiento y las características que la aplicación presenta. El Diagrama de Casos de Uso general del software se exhibe en la Figura 3.

3.1 Aplicación de la programación extrema durante el desarrollo del proyecto

Planificación

Se decidió optar por un lenguaje que sea sencillo de utilizar, orientado a objetos y con un entorno de desarrollo gratuito, siendo seleccionado el lenguaje C#, en su versión "Visual C# 2005 Express Edition" [16].

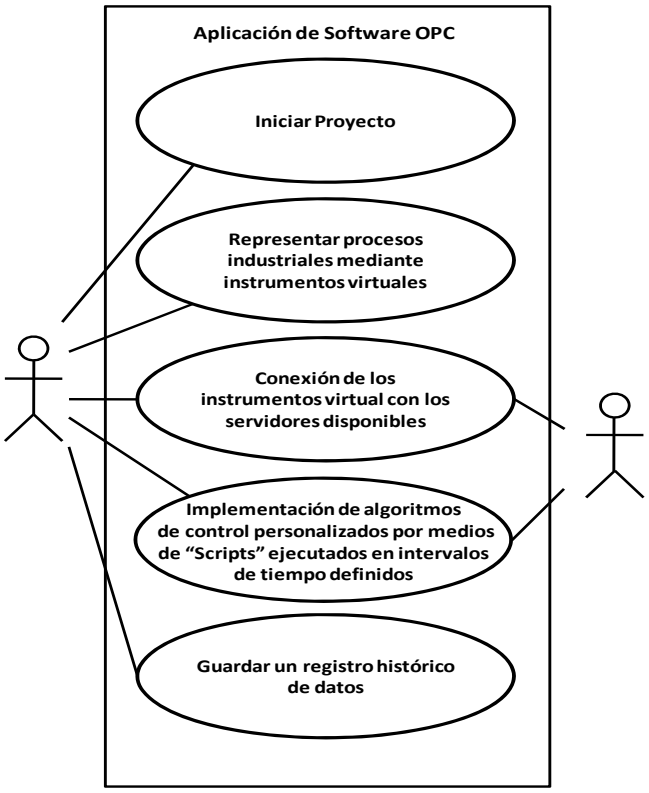

Figura 3. Diagrama de casos de uso general de la aplicación de software OPC.

Se elaboraron las historias de usuario requeridas, las cuales fueron definidas por el equipo de trabajo involucrado en el proyecto, para hacer un levantamiento de todos los requerimientos funcionales necesarios en el software. Las historias de usuarios fueron las siguientes:

- Implementar una interfaz donde el usuario tenga fácil acceso a tres áreas de trabajo: (i) procesos, (ii) guión de control (*script*) y (iii) registro histórico. El área de procesos contará con una pizarra para esquematizar procesos industriales mediante el uso de instrumentos virtuales. La sección de script incluye un entorno gráfico donde el usuario pueda abrir, crear, editar, ejecutar y detener guiones de control escritos en lenguaje JScript o VBScrip. El área de registro histórico deberá contar con un gráfico donde se puedan visualizar y almacenar el comportamiento de las variables de interés presentes en el proceso.
- Crear y eliminar los instrumentos virtuales de forma dinámica.
- Arrastrar y soltar cada instrumento en el área de procesos, de manera que puedan ser reubicados a disposición de las características de cada proceso.

- Disponer de un menú de propiedades para modificar las características de cada instrumento y establecer conexión con los servidores OPC disponibles.
- Elaborar un menú principal para la aplicación, donde el usuario pueda abrir, crear, guardar y cerrar un proyecto, incluyendo la posibilidad de guardar el proyecto al cerrar la aplicación.

Una vez definidos los requerimientos funcionales a través de las historias de usuario, se discutió el grado de dificultad de cada uno de ellos, permitiendo acordar el tiempo de entrega de cada versión parcial del software. Otros miembros del equipo de trabajo, se encargaron de validar mediante pruebas operativas el debido funcionamiento de la solución planteada para cada requerimiento funcional. En caso de cumplir con la versión correspondiente en un tiempo menor, se acordó realizar la entrega de dicha versión del software antes del momento previsto, con el fin de aprovechar el tiempo restante para cumplir con los requerimientos funcionales de las otras historias de usuarios, contribuyendo así con mejorar la velocidad del proyecto en cada iteración. El tiempo de cada iteración se fijó en dos semanas, después de las cuales se evaluó la cantidad de historias de usuario realizadas, o el avance parcial obtenido hasta el momento, para medir la velocidad del proyecto.

Diseño

Para la etapa de diseño se realizaron reuniones y debates entre los programadores en búsqueda de soluciones efectivas para los distintos requerimientos funcionales del proyecto. Se decidió listar las características necesarias que deben cumplirse, luego de esto se proponen varios modelos de solución, para después tomar una decisión según la dificultad en su elaboración y el aporte que dicha solución brinda al producto. Durante el desarrollo de la etapa de diseño, se trató en lo posible de centrarse en los requerimientos que se plantean en las historias de usuario, evitando agregar funcionalidad extra al proyecto apegados al planteamiento expuesto en la metodología XP, para evitar así un desperdicio de tiempo y recursos. Luego de determinar la solución más adecuada, según el criterio compartido de los programadores, se replantearon las ideas y se discutió la solución en busca de una simplificación o para evitar duplicación de código y mejorar su estructura.

Codificación

En la etapa de codificación se propuso como norma a seguir, la elaboración de soluciones parciales las cuales, una vez probadas se trasladaron y unificaron con el proyecto principal. La codificación se llevó a cabo en parejas para procurar así que pasen desapercibidos la menor cantidad de errores y se incremente la productividad, contribuyendo a la calidad del desarrollo. La metodología sostiene que no debe trabajarse sobre tiempos para evitar el cansancio y mejorar el rendimiento de los programadores, por lo tanto se fijó un máximo de 38 horas semanales de dedicación al proyecto y 6 horas diarias continuas de programación.

Pruebas

Las pruebas unitarias e integrales fueron llevadas a cabo por los programadores. Para las primeras, se crearon soluciones específicas para cada tarea o requerimiento del software, las cuales fueron ejecutadas, depuradas y revisadas de forma constante en búsqueda de una solución que pudiera ser llevada al proyecto principal. Se efectuaron pruebas para verificar: (i) capacidad de colocar y arrastrar instrumentos en el área de trabajo, (ii) edición de los instrumentos virtuales, (iii) conexión con los servidores y adquisición de datos, (iv) capacidad de escritura y edición de los guiones de control, (v) interacción entre los guiones de control y los instrumentos virtuales, (vi) generación, visualización y recuperación de registros históricos de variables OPC. Una vez obtenida una solución para cada requerimiento funcional, este se incorporó de forma individual al proyecto general, siendo sometido de nuevo a una serie de pruebas integrales para garantizar que su interacción con los otros componentes del programa fuese exitosa. Las pruebas de aceptación fueron conducidas por otros miembros del equipo de trabajo, implicando en algunos casos la elaboración de demostraciones a personas externas al proyecto.

4. RESULTADOS

4.1 Descripción del software

Se desarrolló una aplicación de software que permite realizar el monitoreo y control de procesos industriales por computadora. La aplicación fue desarrollada utilizando el lenguaje C# de Microsoft y presenta una instalación sencilla, pocas exigencias de hardware, interfaces amigables y una arquitectura

abierta, capaz de crecer o adaptarse según las necesidades cambiantes de la instalación industrial.

El usuario cuenta con un entorno visual donde puede insertar y manipular los instrumentos asociados a las variables proporcionadas por cualquier servidor OPC. Adicionalmente, existe la posibilidad de modificar las diferentes características visuales y el comportamiento de los instrumentos para que se adecuen al proceso, e incluso se pueden implementar algoritmos de control personalizados por medio de guiones (scripts) ejecutados en intervalos de tiempo definidos por el usuario/operador. El software genera además, un registro histórico de las variables elegidas que puede ser recuperado en cualquier momento. Al fundamentarse en el estándar OPC, es posible agregar dispositivos de diferentes fabricantes a medida que estos son adquiridos e incorporados al proceso.

La Figura 4 muestra la ventana principal del programa, la cual incluye los siguientes elementos (1) Barra de instrumentos, (2) Barra de tareas, (3) Ventana de proceso, (4) Ventana de script, (5) Ventana de registro histórico y (6) Estado de ejecución del script.

A continuación se describen las acciones más importantes que pueden ser realizadas dentro de la aplicación.

4.1.1 Inserción instrumentos

Esta acción puede llevarse a cabo desde la barra de instrumentos ubicada en la parte izquierda de la pantalla (ver Figura 4) o desde el menú "Instrumentos". Se encuentran disponibles 48 elementos o instrumentos diferentes, los cuales incluyen diversos tipos de indicadores analógicos y digitales, instrumentos de medición, actuadores, gráficos, entre otros.

4.1.2 Edición de un instrumento

Mediante un menú contextual se obtiene acceso a: (i) modificar la posición relativa entre el instrumento y los demás objetos presentes en la pantalla, (ii) cortar, copiar y eliminar el instrumento (iii) cambiar la configuración y propiedades del objeto (iv) cambiar el nombre y tamaño del instrumento.

4.1.3 Conexión con un servidor OPC de acceso a datos

Dentro de la ventana de propiedades de cada instrumento existe una pestaña identificada con el nombre "OPC" la cual permite tener acceso a la pantalla de configuración para establecer el enlace entre alguna de las propiedades del objeto y una variable

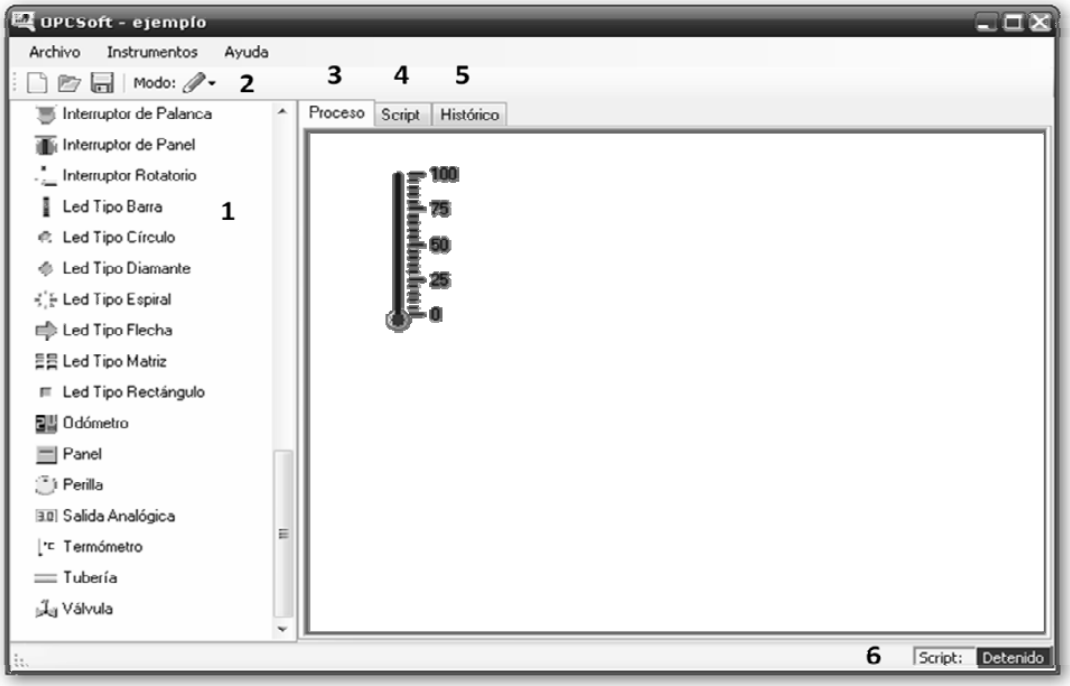

Figura 4. Ventana principal del programa.

(ítem) OPC. Para ello debe indicarse la propiedad que se desea enlazar, el computador de ubicación del servidor (se omite si es local), el nombre del servidor, el nombre del ítem OPC y la velocidad de actualización de la variable en la aplicación cliente, como se muestra en la Figura 5.

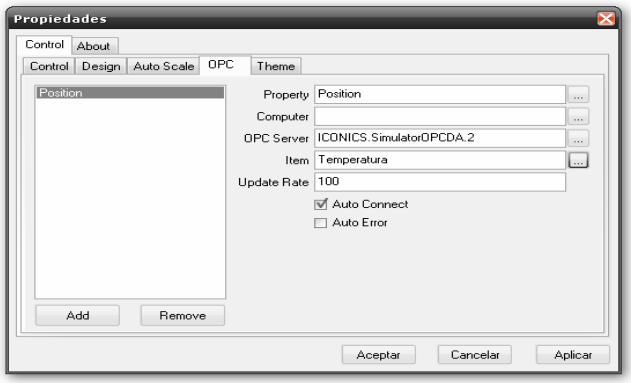

Figura 5. Conexión con una variable OPC.

4.1.4 Creación, edición y ejecución de guiones (scripts)

El área de edición de guiones, mostrada en la Figura 6, permite escribir algoritmos de control personalizados para que se ejecuten en intervalos de tiempo definidos por el usuario. Se dispone como lenguajes de programación: Visual Basic Script y Java Script.

En la parte izquierda de la ventana de edición

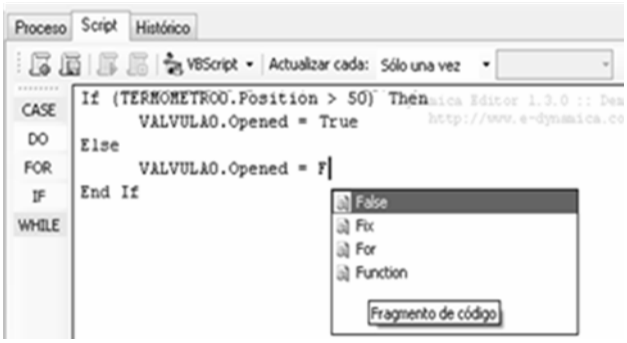

Figura 6. Ventana de edición de guiones con ayuda contextual dinámica.

de guiones, se encuentran botones de acceso rápido que insertan algunas de las estructuras de programación de uso frecuente con la sintaxis adecuada para el lenguaje que se esté utilizando. Este script se puede almacenar, cargar desde un archivo, ejecutarse y detenerse mediante las funciones disponibles en una barra de tareas. La ventana de edición muestra un cuadro de sugerencias a medida que el usuario se encuentra escribiendo su programa. Esta herramienta facilita notablemente la escritura de los guiones ya que presenta un entorno similar a los ambientes de programación utilizados por todos los lenguajes modernos.

4.1.5 Manejo de datos históricos

La aplicación incluye una región que brinda la posibilidad de llevar un registro histórico de los valores asociados a la(s) variable(s) OPC que el usuario seleccione, la cual es mostrada en la Figura 7. Los datos del histórico se almacenan en un archivo ".HistoOPCSoft" con la fecha del momento de la captura de los datos y se ubican por defecto en la subcarpeta "Historicos", dentro de la carpeta que lleva el nombre del proyecto. Los datos se guardan a intervalos de un minuto (estando el programa en modo de ejecución), y al cerrar el proyecto o la aplicación. El registro puede ser recuperado del computador y visualizado en cualquier momento.

4.2 Pruebas experimentales

Se realizaron pruebas de funcionalidad con servidores locales de datos y un servidor remoto dentro de una Red LAN, los cuales proporcionan datos de procesos reales o simulados.

4.2.1 Prueba de visualización sencilla de variables simuladas

Se realizó la simulación del proceso mostrado en la Figura 8. Los datos utilizados para simular el nivel del líquido en el interior del tanque así como el estado de las válvulas fueron obtenidos desde un simulador. El objetivo principal de la prueba consistió en observar las variaciones ocurridas en los elementos virtuales como consecuencia de los cambios en las variables. En el experimento, se emplearon simuladores de datos OPC de las empresas: ICONICS [17] y MATRIKON OPC [18].

Las variables mostradas en la Tabla 2, fueron reconocidas y asociadas a los instrumentos virtuales del software de forma satisfactoria. Se logró verificar que los ítems OPC existentes en dos servidores diferentes eran visualizadas de forma correcta en la aplicación cliente. Además se verificó la capacidad de inter operatividad con diferentes servidores de datos de forma simultánea.

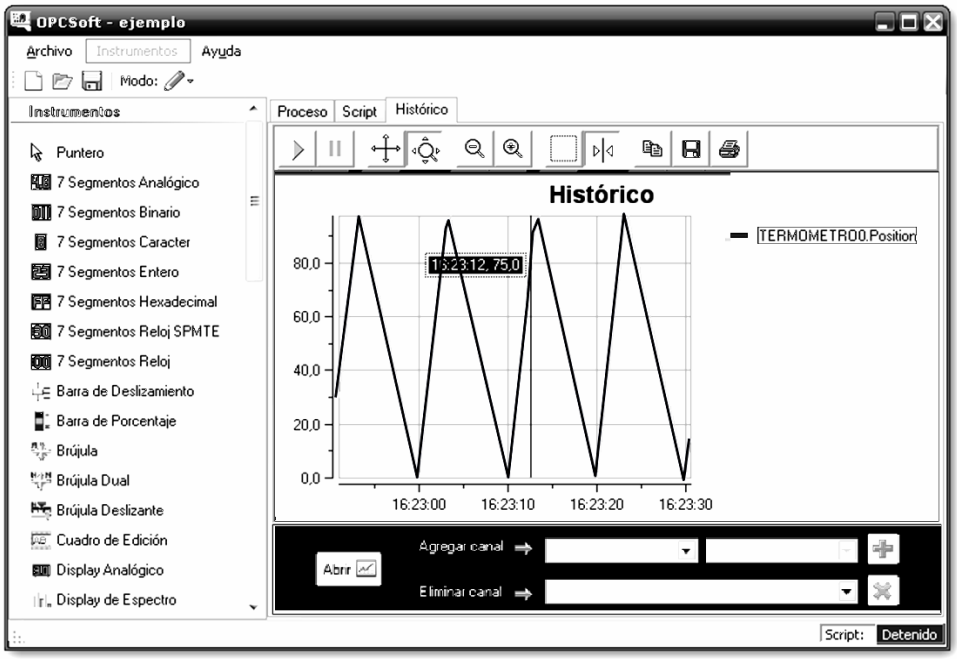

Figura 7. Ventana de datos históricos.

Tabla 2. Variables OPC asociadas a los elementos virtuales.

Simulador	Variable	Tipo de Variable	Permisos	Instrumento Virtual Asociado
ICONICS	NivelLiquido	Numérica (doble)	Lectura/Escritura	Tanque0
ICONICS	NivelLiquido	Numérica (doble)	Lectura/Escritura	Grafico0
MATRIKON	Valve0	Booleana	Lectura/Escritura	Valvula0
MATRIKON	Valve1	Booleana	Lectura/Escritura	Valvula1

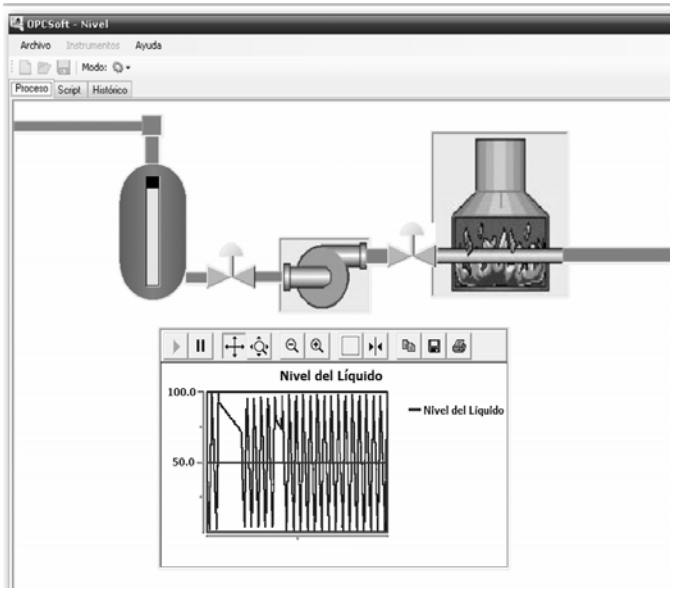

Figura 8. Visualización sencilla de variables.

4.2.2 Conexión con servidor local de datos reales

Para verificar la operatividad de la aplicación y la correcta adquisición de datos provenientes de un servidor local en tiempo real, se elaboró el diseño de la pantalla mostrada en la Figura 9.

Figura 9. Prueba de funcionamiento de *scripts* y adquisición de datos desde un servidor local en tiempo real.

En esta experiencia, la variable "Temperatura" es suministrada por un servidor llamado "VisorOPC", el cual ofrece los datos adquiridos desde un termómetro digital con tecnología 1-Wire [19]. Esta variable es asociada con dos instrumentos virtuales, en este caso un indicador numérico analógico y un termómetro. Asimismo, se optó por asociar otro instrumento virtual con la hora del sistema a través de la ejecución del *script*.

Mediante la ejecución exitosa de esta prueba, se validó la operación del módulo encargado de ejecutar el guión de control, ya que la hora mostrada en la pantalla de la aplicación estaba sincronizada con la hora del computador donde se ejecutó la aplicación. También se logró adquirir y visualizar datos de la variable temperatura provenientes de un servidor local en tiempo real .

4.2.3 Conexión con un servidor remoto dentro de una red LAN

Para verificar la funcionalidad y operatividad de la aplicación se estableció una conexión con un servidor remoto dentro de una red LAN. La prueba consistió en la simulación de un sistema de control de temperatura para un tanque de agua, en el cual la temperatura deseada es de 44 grados centígrados.

Para ello se utilizó un control ON-OFF con una histéresis de 1 grado centígrado en torno al punto de ajuste. El montaje experimental presentado en la Figura 10 utilizó dos computadores, uno empleado como servidor, el cual contiene el hardware de adquisición de datos necesario para la medición de la temperatura y el manejo (encendido/apagado) de las fuentes de calor, y el otro computador ejecutó la aplicación cliente para el control y monitoreo de procesos.

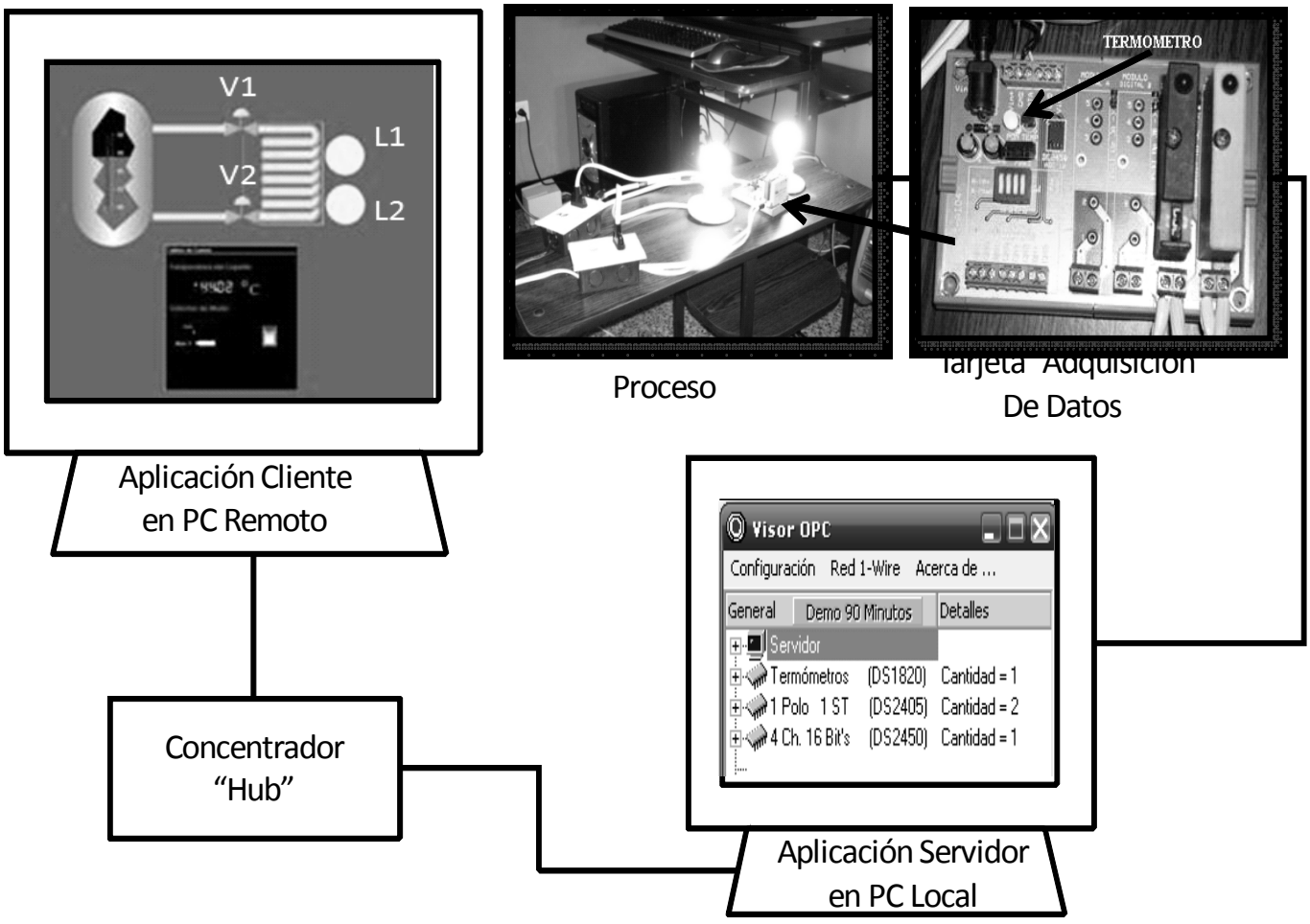

Figura 10. Montaje experimental utilizado para validar la operación remota del software.

La Tabla 3 muestra las mediciones efectuadas en diferentes condiciones de trabajo del proceso simulado. Mediante la activación y desactivación correcta de las fuentes de calor, de acuerdo a la temperatura existente, se verificó la operatividad y funcionalidad de la aplicación para realizar el control de temperatura propuesto.

Tabla 3. Resultados obtenidos al controlar un proceso de forma remota.

T0 (ºC)	Edo V1	Edo V2	Edo L1	Edo L2	Modo	Int Manual
42.36	Abierta	Abierta	On	On	Auto	-
44.58	Abierta	Abierta	On	On	Auto	-
45.91	Abierta	Abierta	Off	Off	Auto	-
44.32	Abierta	Abierta	On	On	Manual	On
43.55	Cerrada	Cerrada	Off	Off	Auto	-
44.67	Cerrada	Abierta	Off	Off	Auto	-
42.12	Abierta	Cerrada	Off	Off	Auto	-
46.04	Abierta	Abierta	Off	Off	Manual	Off

Todas las pruebas fueron realizadas en computadores equipados con sistema operativo Windows XP Service Pack 2. La aplicación fue instalada y ejecutada de forma exitosa en computadores con procesadores Intel de las series Pentium 4 , Pentium Dual Core y Pentium Core 2 Duo, de diferentes velocidades.

5. CONCLUSIONES

En el proyecto se incluyeron instrumentos de medición y control como: medidores, termómetro, indicadores, interruptores, entre otros, los cuales pueden ser asociados con cualquier variable dentro de un entorno industrial. Se adicionó la capacidad de representar tuberías e incluir imágenes de instrumentos para ampliar las capacidades del software de representar procesos industriales.

La aplicación desarrollada es una aplicación sencilla de instalar, sin excesivas exigencias de hardware, con interfaces amigables y de arquitectura abierta, capaz de crecer o adaptarse según las necesidades cambiantes de la empresa o instalación industrial.

Las experiencias y pruebas realizadas indican que la herramienta puede ser de gran valor tanto en ambientes industriales reales como en el área didáctica pudiendo ser utilizada en laboratorios relacionados con las áreas de automatización y control de procesos e instrumentación virtual entre otros.

Se verificó la capacidad de operar el software en ambientes heterogéneos, recibiendo datos desde diferentes servidores, esto es una característica necesaria de todas las aplicaciones basadas en OPC, debido a la arquitectura concebida para las automatizaciones fundamentadas en dicho conjunto de especificaciones.

6. REFERENCIAS

[1] Wang L. y Tan K. C. (2006). “Modern Industrial Automation Software Design”. John Wiley & Sons, Inc., Hoboken, New Jersey, USA.

[2] Distefano M. (1.999). “Comunicaciones en Entornos Industriales”. Facultad de Ingeniería. Universidad Nacional de Cuyo, Argentina.

[3] OPC FOUNDATION (1998). “OPC Overview”, versión 1.0. Extraído el 14 de mayo de 2007 desde: www.opcfoundation.org.

[4] Shimanuki Y. (1999). “OLE for process control (OPC) for new industrial automation systems”. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC 99)*, Vol. *6*, pp. 1048-1050. Tokio, Japón.

[5] Martínez M. (2003). “Cliente OPC realizado en Visual Basic .NET”. Extraído el 25 de Octubre de 2007 desde: www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art123.asp

[6] Pattle R. y Ramisch J. (1997). “OPC the de facto standard for real time communication”. *Proceedings of the 1997 Joint Workshop on Parallel and Distributed Real-Time Systems (WPDRTS / OORTS '97)* (pp. 289-294). Washington: IEEE Computer Society.

[7] Iwanitz F. y Lange J. (2002). “OPC Fundamentals, Implementation and Application” (2 da ed.). Heidelberg, Alemania: Húthig GMBH & Co KG Heidelberg.

[8] OPC Foundation. (s.f.) : "What is OPC?". Extraído el 10 de Enero de 2007, desde: www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC.

[9] Zheng L. & Nakagawa H. (2002): "OPC (OLE for process control) specification and its developments". *Proceedings of the 41st SICE Annual Conference (SICE 2002)*, Vol.2, pp. 917-920.

[10] OPC Foundation (2003): "Data Access Custom Interface Standard Versión 3.00".

[11] OPC Foundation (1999): "Data Access Automation Interface Standard Version 2.02".

[12] Highsmish J. (2001): "History: The Agile Manifesto". Extraído el 30 de Junio de 2007, desde: http://agilemanifesto.org/history.html.

[13] Beck K. (2000): "Extreme Programming Explained. Embrace Change", Addison Wesley-Professional.

[14] Wells D. (2001): "The Rules and Practices of Extreme Programming". Extraído el 02 de Julio de 2007 desde: www.extremeprogramming.org.

[15] Flower, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3 ra Ed. Addison Wesley. pág 99.

[16] Microsoft (2007) : "Visual C# 2005 Express Edition". Extraído el 10 de Mayo de 2007 desde: www.microsoft.com/spanish/msdn/vstudio/Express/VCS/default.mspx.

[17] ICONICS (2007): "OPC Simulator Server". Extraído el 18 de Junio de 2007 desde: www.iconics.com/support/free_tools.asp.

[18] MATRIKON OPC (2007): "MatrikonOPC Simulation Server". Extraído el 20 de Junio de 2007 desde: www.matrikonopc.com/downloads/176/index.aspx.