

Universidad Latinoamericana de Ciencia y Tecnología

Facultad de Ingeniería

Escuela de Ingeniería Informática

Trabajo final para optar al grado de Licenciatura en Informática con énfasis en
Gestión de Recursos Tecnológicos

**Administración de aplicaciones con integración de componentes en
mecánica, electrónica, robótica y domótica.**

Sustentante: Jeffrey Rojas Marín

Cédula: 1-1152-0334

Tutor: Lic. Miguel Pérez Montero

I cuatrimestre del 2009

Nota obtenida: 94

Acta: 095

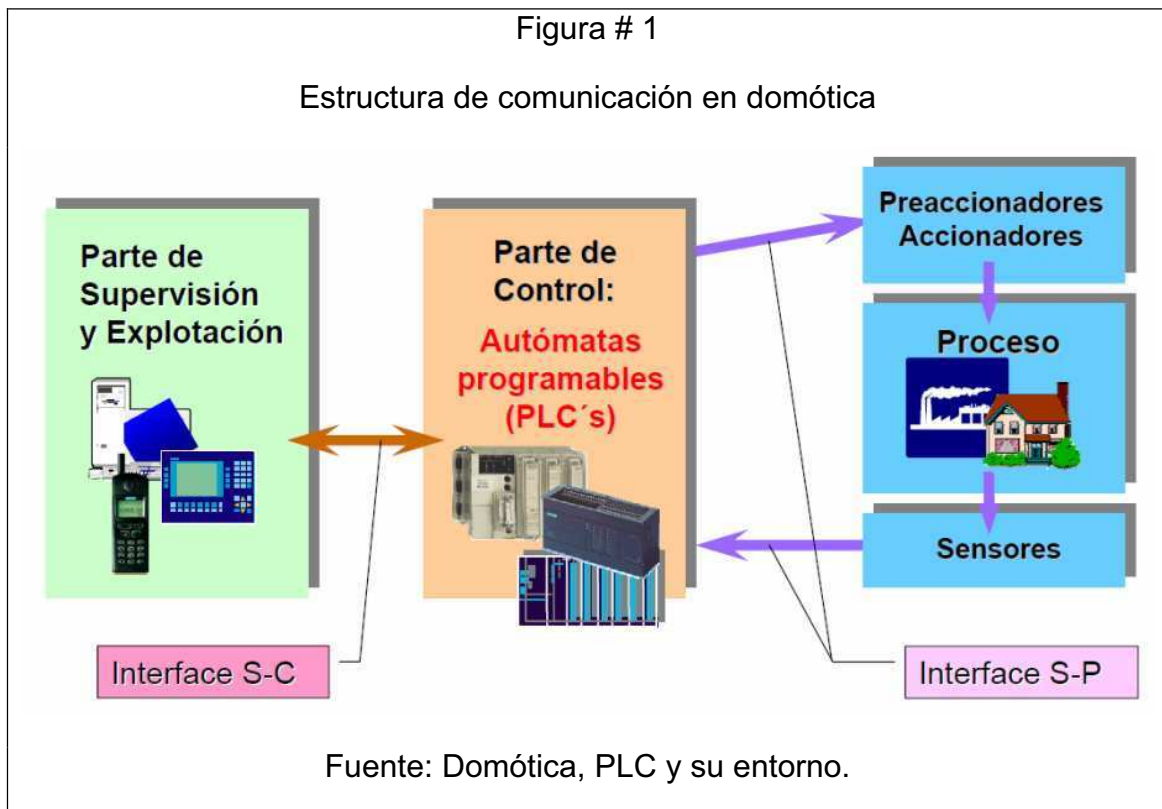
1. El mundo de las aplicaciones integradas

A partir de las necesidades de la vida cotidiana, se busca un inteligente manejo de recursos, en el cual estas necesidades son satisfechas mediante la implementación de aplicaciones integradas que hoy en día ofrecen una amplia gama de funcionalidades.

Hoy, empresas como TSX Schneider, SIEMENS y Allen-Bradley son pioneras en la implementación de componentes orientados hacia la integración entre tecnologías eléctricas y mecánicas hacia sistemas de información, por lo cual un ejemplo claro es el uso de autómatas programables (AP) para procesos de integración.

Actualmente, las funciones de las aplicaciones integradas van desde lo más sencillo, como lo es regular la temperatura de un edificio, hasta funcionalidades más complejas, como el control para el tramado de las llantas de un vehículo liviano, las cuales están ligadas en una relación precio calidad según la capacidad de sus componentes.

Desde hace algunos años, las aplicaciones en domótica sorprenden a los consumidores; estas son un buen ejemplo de las posibilidades que la integración ofrece, en lo cual los procesos normales son gestionados hacia nuevas funcionalidades y eficiente control de sus recursos. Véase figura # 1.



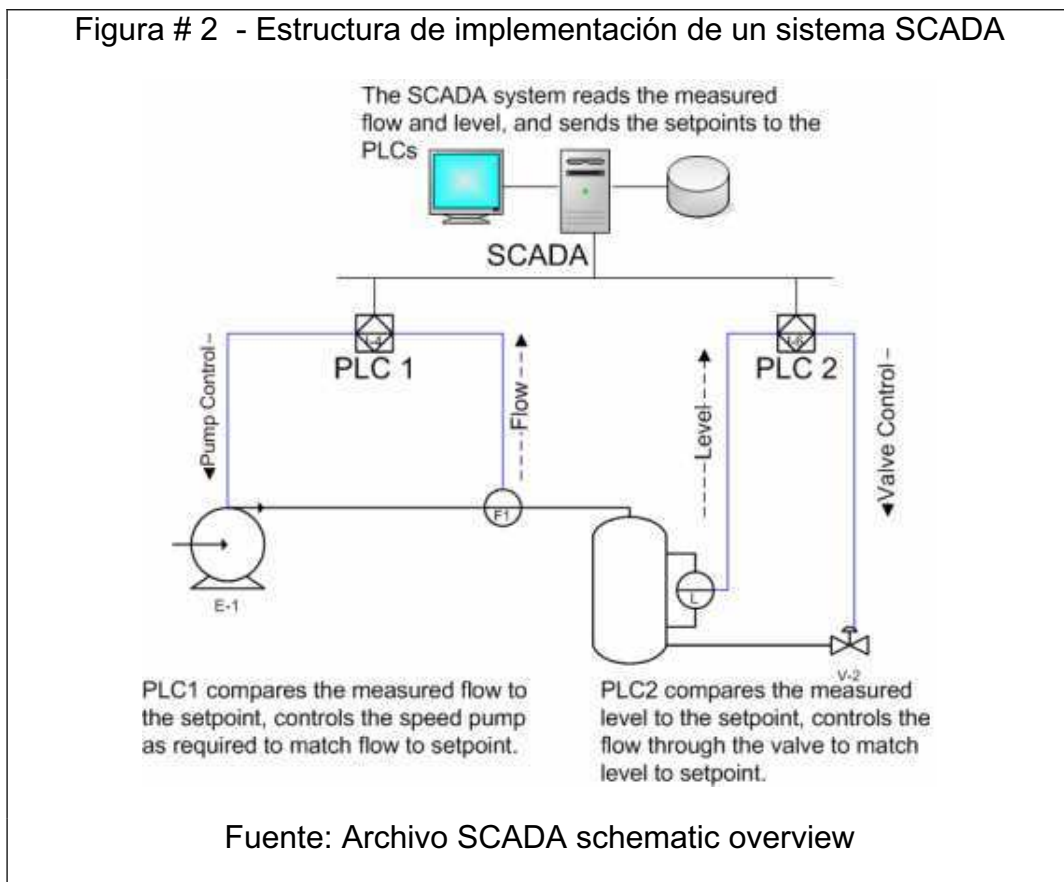
2. Comunicación y manipulación de componentes integrados en informática

La estructura básica para la implementación de un sistema integrado se logra mediante la unificación de diversos componentes, los cuales se detallan a continuación.

Ya se ha hablado sobre autómatas programables llamados AP o PLC; además de estos, se necesitan dispositivos periféricos de entradas y salidas que alimenten de información los procesos, los cuales, por medio de una aplicación de *software* se gestionan en control de la funcionalidad del proceso.

Esta aplicación será especialmente diseñada para funcionar sobre computadores, proporcionando comunicación con los dispositivos autónomos y controlando el proceso de forma automática desde la pantalla del computador por medio de la administración de sus entradas y salidas.

Un ejemplo de un sistema de control de dispositivos integrados es el caso del sistema SCADA (2006, *Supervisory Control and Data Acquisition*), el cual se comunica con controladores autónomos para gestionar el flujo de procesos. Véase figura # 2 (*SCADA_schematic_overview*).



Componentes de un sistema SCADA:

2.1. Unidades de Terminal Remota (UTR, PLC):

Estas UTR se conectan al equipo físicamente y permiten la lectura de su estado. Por ejemplo: abierto o cerrado de una válvula o un interruptor, lectura de medidas (presión, flujo, voltaje o corriente). Además, son capaces de enviar señales para su control: abrirlo, cerrarlo, configurar la velocidad de proceso, ponerla en marcha o detenerla.

Existen UTR de funcionamiento digital o analógico, en las cuales se envían comandos digitales de salida o puntos de ajuste analógicos durante su ejecución.

Según la Universidad de Oviedo (2006), uno de los componentes más importantes es el uso del PLC (*Power Line Communications*), un sistema eléctrico programable diseñado para ser utilizado en un entorno industrial.

Está formado de varios componentes, entre los cuales se pueden citar:

- CPU: cerebro de proceso, cuenta con un microprocesador y memoria de uso.
- Monitor de proceso: muestra el estado del proceso durante la ejecución.
- Fuente de alimentación: fuente eléctrica que el sistema requiere para operar.
- Módulo de entradas y salidas: comunicación por medio de señales eléctricas y código binario.

Por medio de su memoria programable y del almacenamiento interno de instrucciones lógicas (secuencia, temporización, recuento y funciones aritméticas), permite implantar soluciones específicas según las necesidades del usuario con el fin de controlar, mediante entradas y salidas, diversos tipos de máquinas o procesos.

2.2. Interfaz de usuario HMI (*Human Machine Interface*).

Este es el medio por el que el usuario puede comunicarse con una máquina o una computadora. Esta interfaz de usuario comprende todos los puntos intermedios para establecer la comunicación por medio de *hardware* y *software*. Las estaciones de trabajo HMI se comunican con una "estación

maestra", la cual consta de servidores y el *software* responsable para establecer la comunicación con el equipo del campo (UTR, PLC).

Generalmente los HMI presentan su información de forma gráfica, lo cual permite una mejor representación de la información por medio de alarmas y diagramas de representación, los cuales se componen de gráficos de líneas y símbolos esquemáticos para representar los elementos del proceso, hasta llegar a un nivel de detalle como lo es fotografías digitales de los equipos sobre los que se animan secuencias, lo cual proporciona funciones de control y supervisión sobre la ejecución de procesos.

En el pasado, plataformas abiertas como Linux no eran ampliamente usadas, pero, debido al ambiente de desarrollo altamente dinámico y flexible, permiten a clientes acomodarse en términos de *hardware* a mecanismos que hoy en día empresas como UNIX (con licencias OpenVMS) venden.

2.3. Infraestructura de comunicación

Tradicionalmente la comunicación se establece por medio de la combinación de radios y señales seriales, conexiones de módem, Ethernet, fibra óptica y Wireless.

Se busca que la comunicación se implemente según una arquitectura abierta capaz de adaptarse conforme a las necesidades de la empresa y de fácil comunicación con el usuario por medio de programas sencillos de instalar, sin excesivas exigencias de *hardware*.

3. Estandarización para *software* de aplicación

Durante la interconexión de sistemas de integración, según el mercado actual, existe un panorama bastante amplio de soluciones, en el cual hoy en día se busca mayor flexibilidad y normalización sobre sus periféricos de comunicación y lenguajes de programación, aportando apoyo a tecnologías de uso común. Con este fin, nace el estándar IEC 61131 (EIC International Electrotechnical Commission, 2003), el cual inicia el camino para lograr este objetivo.

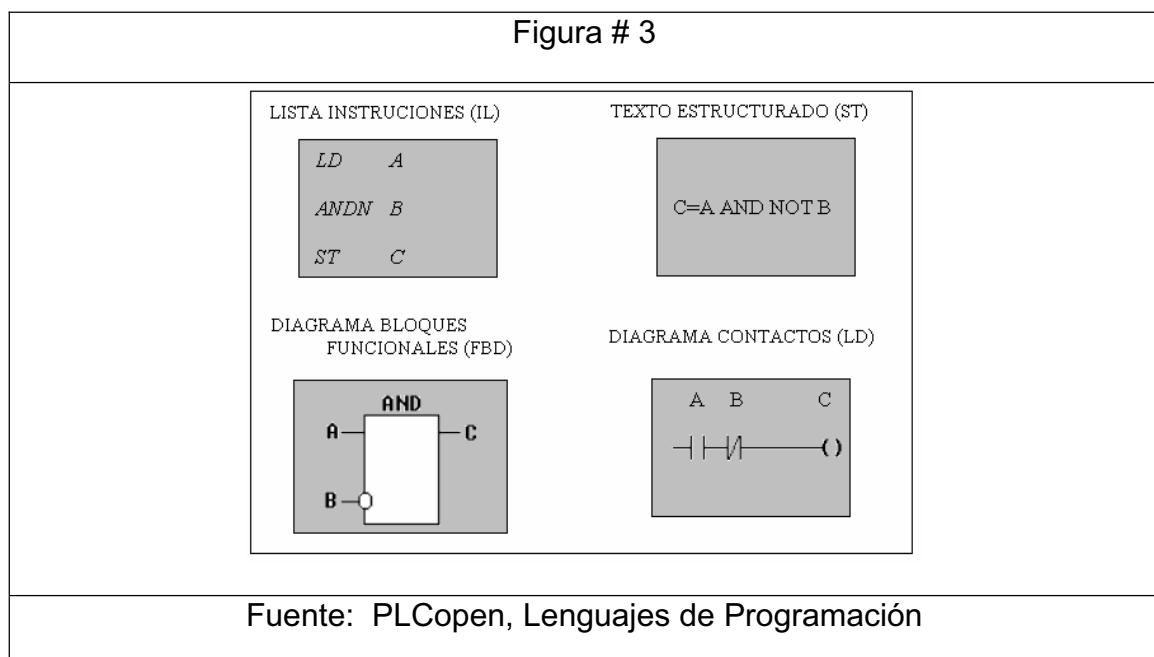
Con respecto a la normalización de aplicaciones de *software* para uso de control industrial, este se detalla en el estándar IEC 61131.3 et al. (2003) específicamente, el cual da un marco de referencia hacia recursos de programación estándar, y la definición sobre elementos comunes como tipos de datos y variables.

Basado en estos tipos de datos, el usuario puede definir sus propios tipos de datos, conocidos como tipos de datos derivados. Una variable se puede declarar como uno de los tipos de datos elementales definidos o como uno de los tipos de datos derivados. De este modo se crea un alto nivel de independencia con el *hardware*, favoreciendo la reusabilidad del *software*.

Dentro de IEC 1131-3 et al. (2003), los programas, bloques funcionales y funciones se denominan Unidades de Organización de Programas (*POU*).

Se definen 4 tipos de lenguajes de programación, literales y gráficos, los cuales se detallan a continuación:

Representación de lenguajes de programación



Sin duda alguna, el cumplir con el estándar IEC 1131-3 garantiza mayores oportunidades de integración sobre diferentes tecnologías acerca de la implementación de soluciones en *software* entre proveedores y compañías hacia un mercado que estandariza sus procesos.

4. Software de programación para componentes autómatas

⇒ Step 7

Figura # 4

STEP 7 Software **SIEMENS**

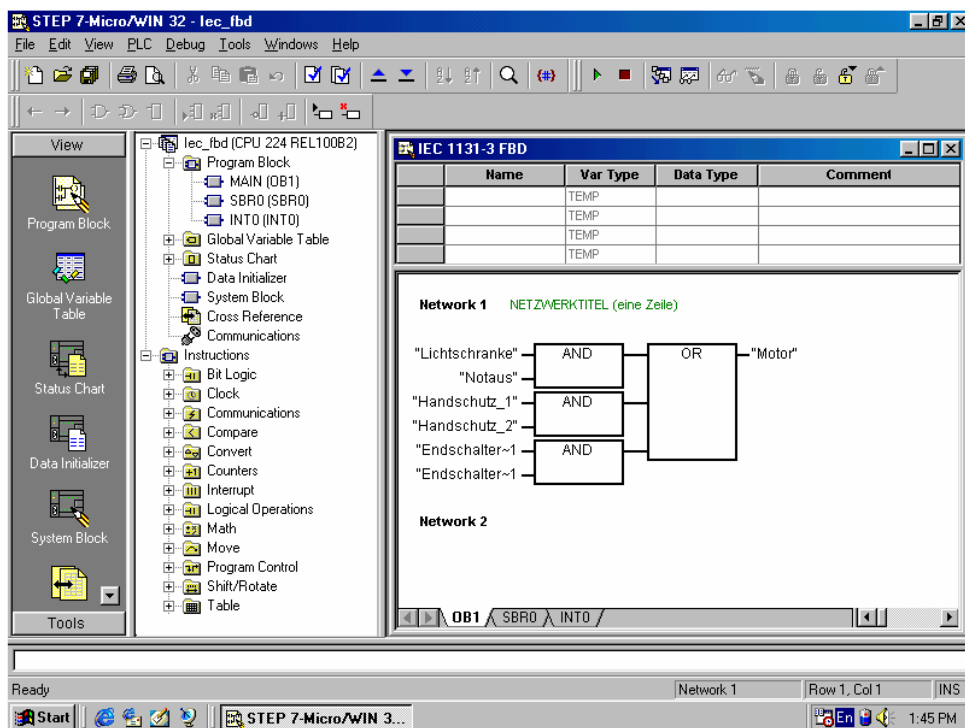


Fuente: SIMATIC STEP 7, Programming Software Siemens

Dentro de la familia de productos de programación Siemens, se encuentra esta poderosa herramienta, la cual facilita la implementación conceptual de proyectos de integración.

Posibilita realizar la simulación de componentes y garantiza el éxito de las aplicaciones antes de su instalación en producción.

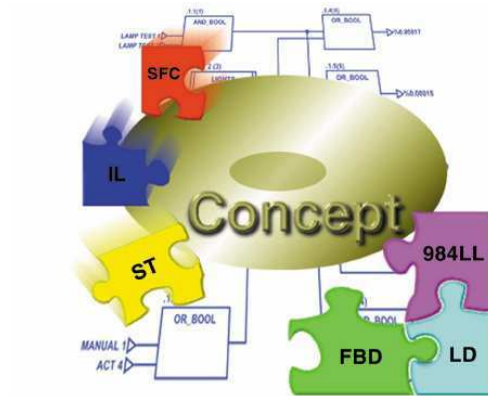
Figura # 5 - Visualización ambiente de desarrollo, Step 7



Fuente: UCR, introducción a los controladores lógicos programables (plc).

⇒ Concept

Figura # 6



Fuente: PLC Programming Software, Schneider Electric

Este producto nace para facilitar la construcción de soluciones en *software* con integración de componentes electrónicos.

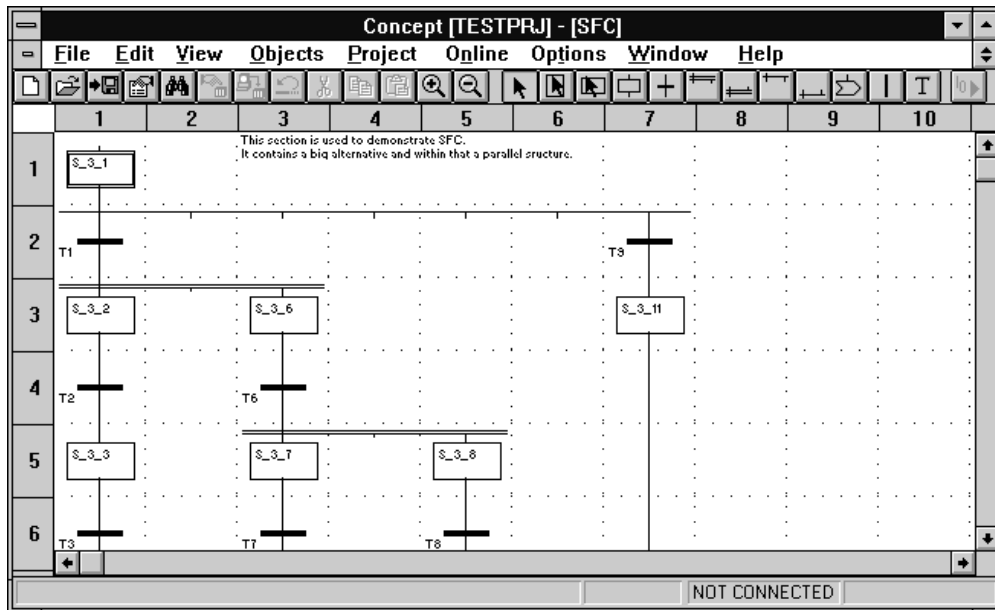
Figura # 7 - Lenguajes de programación soportados

The screenshot shows a dialog box titled 'New Program Section'. It has a blue title bar with a close button (X). The main area contains a group box labeled 'Editor type' with a list of radio buttons: FBD (selected), SFC, LD, ST, IL, and 984 LL. To the right of this list is a text input field labeled 'Section name:'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Fuente: Iniciación en CONCEPT 2.5, Universidad de los Andes.

Durante su ejecución permite la representación de sus componentes por medio de los diagramas SFC, lo cual se puede apreciar a continuación:

Figura # 8 - Visualización ambiente de desarrollo en Concept



Fuente: **Concept 2.5, Programming Software.**

Al ser diseñado para cumplir con lo estipulado según la norma IEC 61131-3, es compatible con lenguaje de programación sobre lenguaje estructurado ST y lista de instrucciones IL.

Figura # 9 - Visualización código fuente en Concept

The screenshot shows the source code editor in Concept 2.5, displaying two code windows:

```

ST1
===== Structured Text Start =====
VAR
  TIMER : TON;
END_VAR

TIMER(IN := NOT pulse,
      PT := t#1s); (* Blink timer *)
pulse := TIMER.Q;

(* Count every pulse *)
IF pulse = 1 THEN
  count := count + 1;
END_IF;
(* Animate lights according to count *)
CASE count OF
  1: out1 := TRUE;
  2: out2 := TRUE;
  3: out3 := TRUE;

```

```

IL1
===== Instruction List Start =====
VAR
  RUN_TIMER : TON; (* Blink timer *)
END_VAR

(* Default for the marker *)
LD   run_light1
ST   run_light

(* Create a 1.0 Hz. pulse *)
LD   run_pulse
STN  RUN_TIMER.IN
CAL  RUN_TIMER(PT := t#1s)
LD   RUN_TIMER.ET
ST   animatetime
LD   RUN_TIMER.Q
ST   run_pulse
JMPCN end (* No pulse y

```

Footer: Ln: 1 Col: 1 INS NUM NOT CONNECTED

Fuente: **Concept 2.5, Programming Software.**

⇒ Unity

Figura # 10



Fuente: Telemecanique, Schneider

De acuerdo con las herramientas de desarrollo más modernas para la automatización, es un poco más flexible que sus competidores ya que es compatible con lenguajes de programación tales como VB y C++, entre otros.

Figura # 10 - Ambiente de desarrollo Unity pro 500

The screenshot shows the Unity Pro 500 development environment. On the left is a project tree with folders for 'Variables e instancias FB', 'Movimiento', 'Comunicación', 'Programa', and 'Eventos'. The 'Programa' folder is expanded to show 'Tareas' and 'MAST' subfolders. The 'MAST' folder contains 'Secciones' with various program instances like 'MiProgramaEnLader', 'MiProgramaFBD', 'MiProgramaEnGrafoet', 'MiProgramaEnST', 'MiProgramaDeContaje', 'File_MGMT1', 'Control_Valvula', 'Inicializa', 'Llamada_Basic', and 'Lexium05_A_Can'. A dropdown menu is open over the 'Nueva_Variable_Elemental' row in the table, showing data types: DT, DWORD, EBOOL, INT, REAL, STRING, and string[75].

● sALIDA	2	REAL
● SD_Status	7	string[75]
● SD_Status_Bad_Card	1	string[75]
● SD_Status_No_Access	1	string[75]
● SD_Status_No_Card	1	string[75]
● SD_Status_OK	1	string[75]
● SD_Status_System_Error	1	string[75]
● SD_Status_Write_Protected	1	string[75]
● Seek_Done	2	BOOL
● Seek_Error	2	BOOL
● Seek_file	3	EBOOL
● Seek_Mode	2	INT
● Seek_Offset	2	DINT
● Seek_Status	2	WORD
● SYNC_Count_0	1	BOOL
● Valor_Count_0	1	DINT
● VAR1	2	STRING
● VAR2	2	STRING
● velocity1	4	UDINT
● velocity2	3	UDINT
● Write_Done	2	BOOL
● Write_Error	2	BOOL
● Write_file	3	EBOOL
● Write_Status	2	WORD
● Nueva_Variable_Elemental	0	BOOL

Fuente: Unity Pro 3.0, Tutorial Práctico.

⇒ CoDeSys, Smart Software Solution.

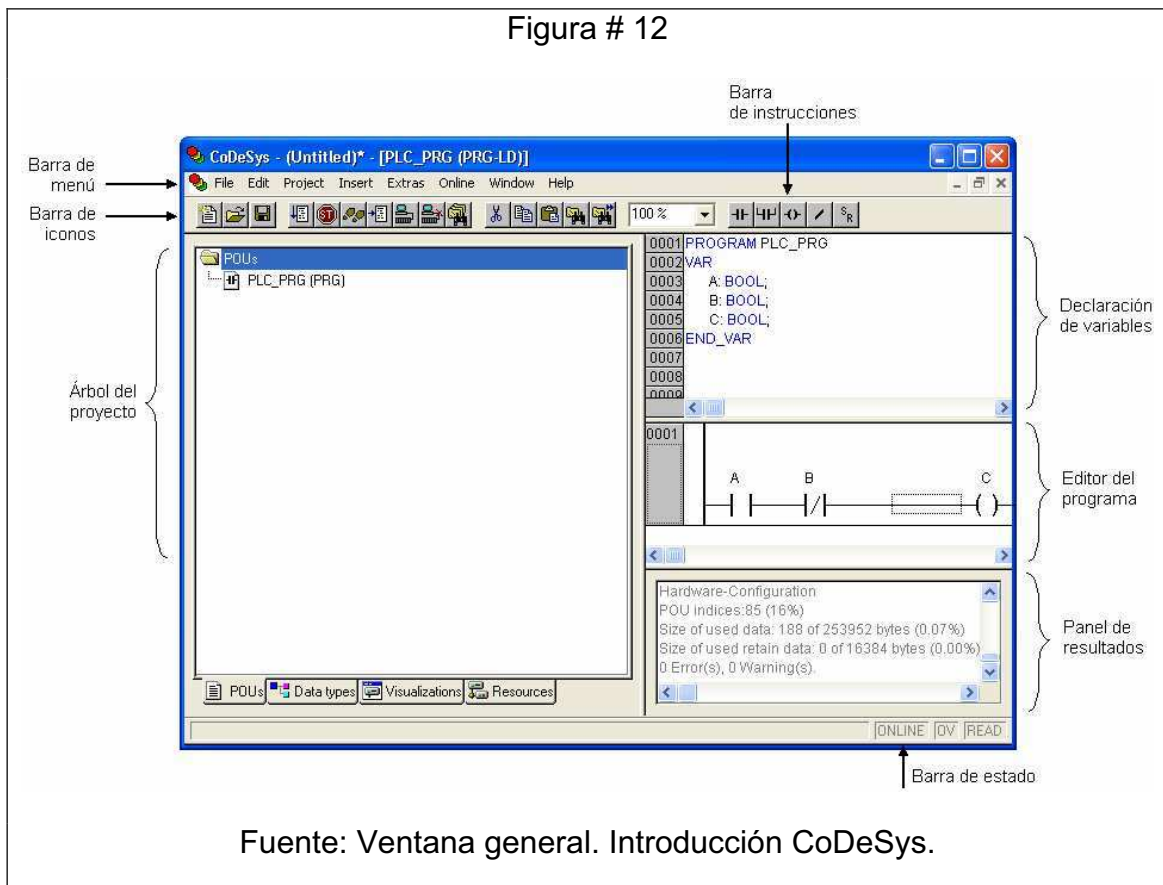
Figura # 11



Fuente: Smart Software Solutions, Automation

Esta herramienta es muy utilizada por muchas compañías en el mercado europeo, la cual nace en 1994 (Smart Software Solution, Alemania, Kempten). Su distribución es gratuita y existe gran cantidad de documentación acerca de su uso.

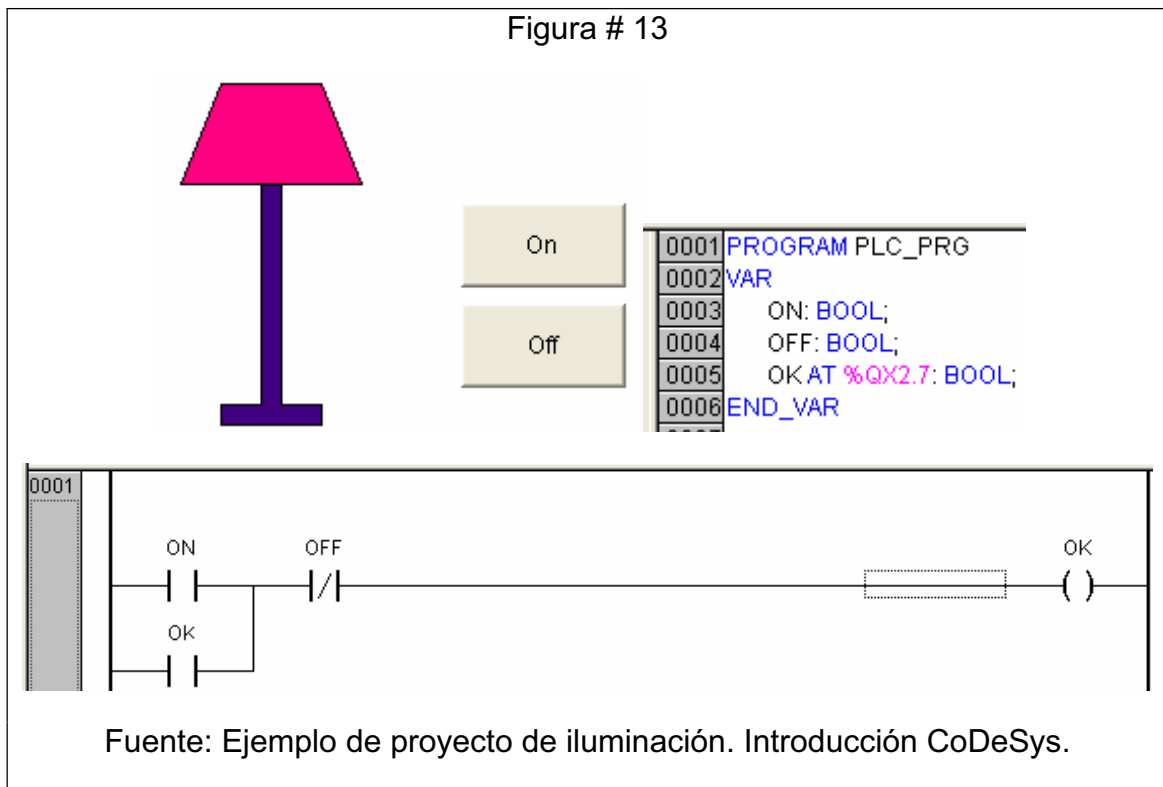
Figura # 12



Fuente: Ventana general. Introducción CoDeSys.

Según las características incluidas en esta herramienta, se destaca la preconfiguración de componentes de iluminación, motores de paso, componentes de inducción y dispositivos para regulación de temperatura.

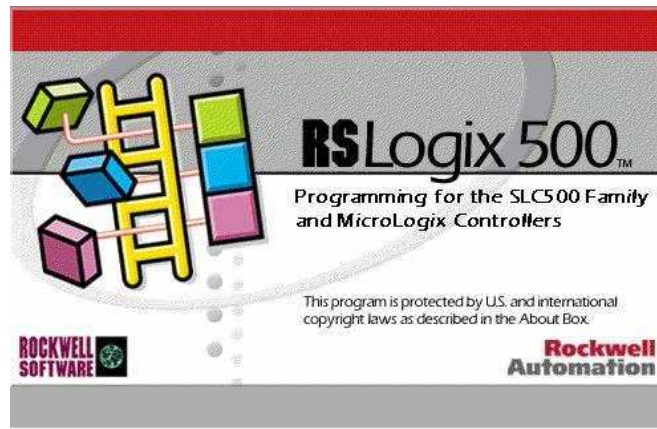
Figura # 13



Fuente: Ejemplo de proyecto de iluminación. Introducción CoDeSys.

⇒ RSLogixPro 500, Rockwell Automation.

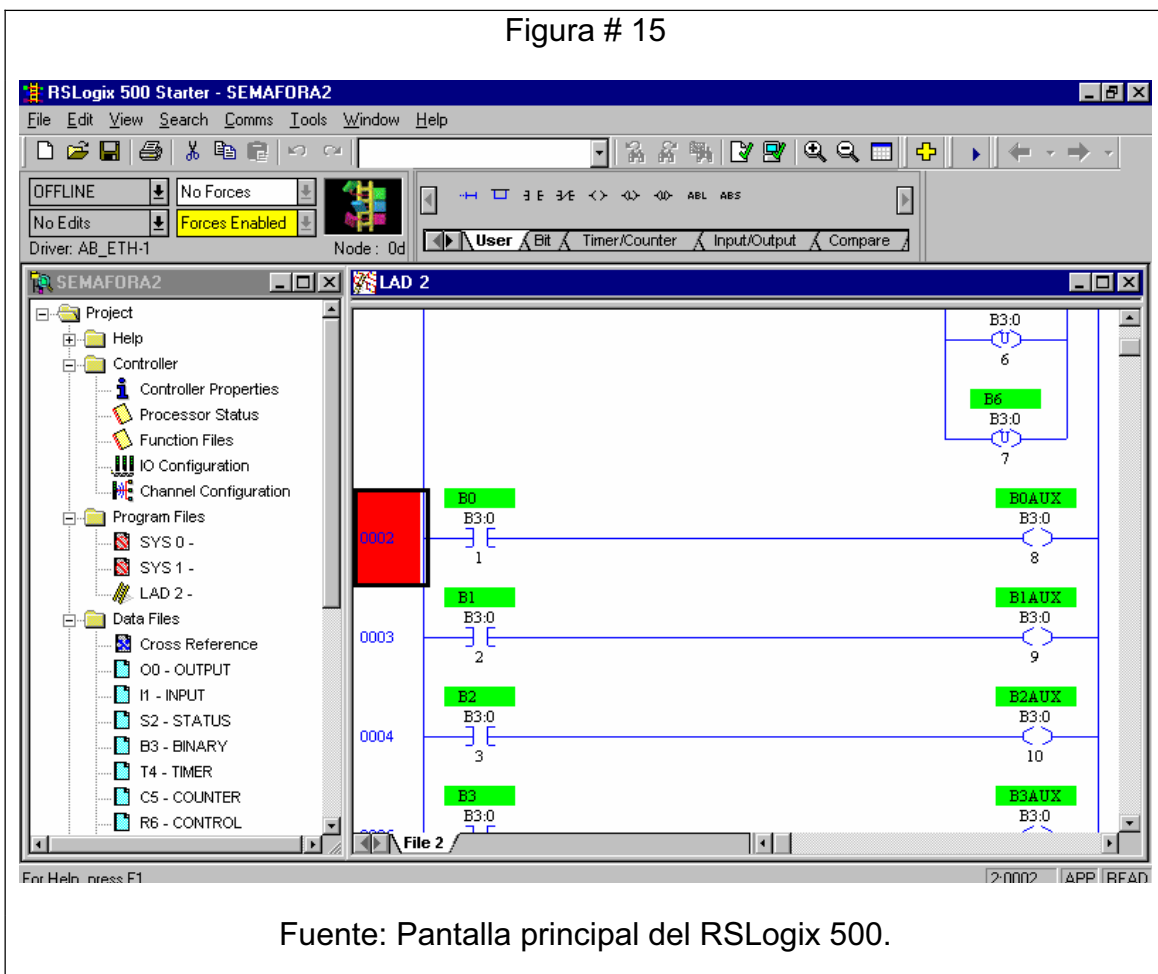
Figura # 14



Fuente: Smart Software Solutions, Automation

El LogixPro 500 (Rockwell Automation) es un *software* especializado en la construcción de lógica para componentes PLC por medio de diagramas de contactos (LD); está diseñado para ser utilizado en la plataforma de sistemas operativos Microsoft.

Figura # 15

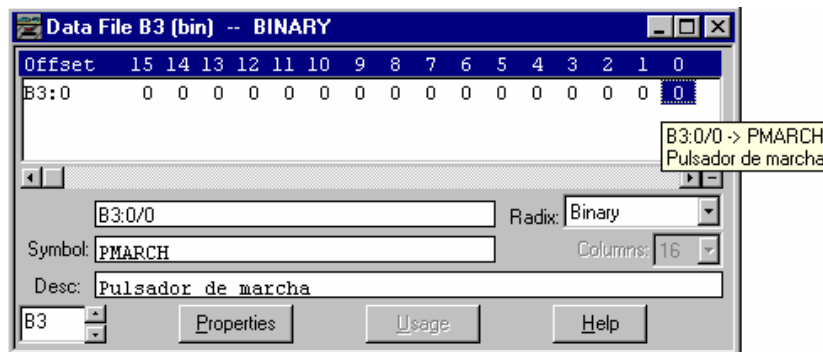


Fuente: Pantalla principal del RSLogix 500.

Dentro de sus características más importantes destacan los módulos de trabajos OffLine y Online. En el método Offline se permiten realizar modificaciones y prueba a los programas fuentes sin necesidad de estar conectado al PLC. Ellos serán reflejados cuando se realice la descarga hacia el PLC, muy diferente al nivel de independencia que permite el método Online, que necesita la conexión directa al PLC afectando directamente su funcionamiento sin necesidad de parar la producción.

Permite la administración de bajo nivel de entradas y salidas por medio de la representación de sus valores en variables binarias. Véase figura # 16.

Figura # 16



Fuente: Binary Data file. Pulsador de marcha.

Dentro de las instrucciones lógicas más importantes para configurar diagramas de contactos se encuentran las siguientes:

Figura # 17

Símbolo	Instrucción
	Procedimiento nuevo
	Instrucción Booleana Verdadero
	Instrucción Booleana Falso
	Inicio y fin de condición verdadero
	Timer de proceso para retardar salidas
	Contador numérico de proceso

Fuente: Instrucciones lógicas. Diagrama de escalera.

5. Software de simulación para componentes autómatas

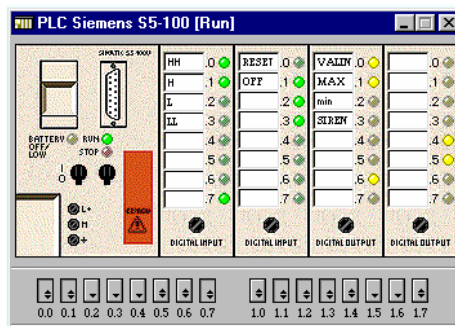
En la actualidad, se dispone de la funcionalidad de la simulación de procesos, la cual representa en forma gráfica los procesos de automatización:

Existen varios tipos de simulación, entre ellos la simulación de componentes PLC y la simulación de instalaciones. Herramientas como Logix Pro y AW-Sys utilizan este tipo de simulación en sus procesos.

⇒ PC-SIM, Allen Bradley

Por medio de esta herramienta se permite la simulación de componentes PLC, representando las características de un PLC real.

Figura # 18

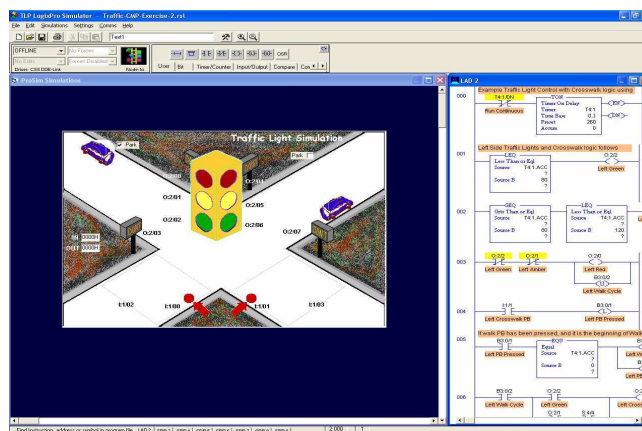


Fuente: Simulación de PLC, Siemens S5-100.

⇒ PL-SIM, Allen Bradley

PL-SIM permite la simulación de instalaciones de automatización, representando la integración de componentes en un ambiente real.

Figura # 19



Fuente: Simulación de Instalaciones. Semáforo.

6. Tipos de funcionalidades hacia un mercado integrado

En el mercado actual existen grandes innovaciones logradas desde la integración de múltiples componentes de una solución final, a partir de componentes como sensores fotoeléctricos, relés, temporizadores, disyuntores, señalizadores, transductores, interruptores y arrancadores de motores.

Algunas de las ventajas sobre la integración de procesos es la eficiencia en los tiempos de proceso, mayor producción, control de procesos de forma remota, índice de fallo muy bajo, los cuales brindan funcionalidad hacia algunas de las aplicaciones que a continuación se enumeran:

- Sistemas de protección contra incendios
- Sistemas de iluminación
- Control de acceso
- Revisión de vehículos
- Monitoreo de procesos
- Procesos de distribución
- Control de calidad
- Integración hacia sistemas ERP (Enterprise Resource Planning)

En resumen, la funcionalidad obtenida con la integración representa alcanzar los beneficios de cada técnica independiente hacia una nueva solución.

7. Clientes de un mercado de especialización

Básicamente, la especialización hacia un mercado representa ventajas que marcan una gran diferencia con respecto a sus competidores; se busca abrir el segmento de mercado hacia las necesidades de los clientes que no han sido explotadas todavía.

Ahora las empresas necesitan de procesos modernos e inteligentes, los cuales agilicen su operación y lleven sus tareas hacia nuevas funciones con el uso del tiempo mínimo de operación, en que los PLC, por su gran aplicabilidad, flexibilidad y facilidad de programación, brindan una solución para la automatización de procesos.

Un ejemplo de esto se ve representado con la implementación de la domótica, en la cual la gestión en una vivienda se automatiza, su funcionamiento se integra en controladores inteligentes que aportan servicios, seguridad y confort: “integración de la tecnología en el diseño inteligente de un recinto” (Wikipedia, 2009).

Por esto el personal de sistemas de información debe estar preparado para su participación en este ámbito, conociendo las innovaciones de campo hacia la integración de la informática a otras áreas de funcionalidad.

8. Implementación de soluciones integradas

Sobre la implementación de soluciones integradas se debe evaluar el alcance del proyecto de implementación ya que durante su conceptualización se puede encontrar gran cantidad de componentes diversificados en precio y calidad.

Es aquí donde especificaciones en modularidad, seguridad y expansión son características importantes de analizar. Generalmente proyectos extensos necesitan mayor cantidad de recursos, y su análisis debe considerar un correcto funcionamiento durante el transcurso de su vida útil. Una mayor inversión en recursos tecnológicos podría ayudar a facilitar la gestión de las piezas claves del proyecto en ejecución, sin que esto signifique el desperdicio de recursos.

Por esto, durante la etapa de planeación, se deben seguir estrictos controles hacia variables claves que garanticen el éxito y prevención de riesgos de mi proyecto. Por ellos, prácticas como análisis de costos, investigación, diseño y documentación son actividades importantes a considerar sobre el alcance de cada proyecto.

9. Complejidad para desarrollo de aplicaciones integradas

En este mercado no siempre todo está construido, por lo que generalmente muchos de los controladores de uso autónomo son construidos en casa; por esto, un error en la etapa de diseño podría significar un daño irreparable en la construcción del componente físico implementado. Es recomendable la utilización de simuladores de funcionalidad, los cuales se logran a partir de piezas de *software* de uso industrial.

Según la complejidad, para el desarrollo de aplicaciones integradas planteada desde una perspectiva de sistemas de información, el mayor problema para su implementación se centra en la poca cantidad de recurso humano especializado para realizar tareas de integración; esto es debido a la poca preparación de parte de las universidades, las cuales separan funcionalidades entre ramas de especialización.

Es necesario iniciar con procesos de formación sobre procesos de integración, los cuales permitan a nuevos profesionales el conocimiento clave y la capacidad de utilización de componentes autómatas para participar en la construcción de nuevas características de innovación.

10. Ideas generales a partir de un análisis de resultados

Después del análisis realizado sobre el proceso de integración de componentes, se puede afirmar que existen grandes avances hacia la unificación de tecnologías, las cuales representan grandes oportunidades para el desarrollo de nuevos productos de innovación.

El alcance de este tipo de proyectos ha crecido, por lo cual, día a día, muchas compañías dedican sus actividades hacia esta labor. Existe una gran diversidad de componentes y herramientas para su programación y simulación; esto disparó el inicio de un proceso de estandarización de las características que componen los procesos de integración (IEC 61131.3 et ál., 2003).

Hoy en día, existe gran demanda sobre productos de uso autónomo ya que se garantiza un mejor aprovechamiento de recursos, mejorando la producción, y se permite minimizar el riesgo pues el índice de fallo es muy bajo.

A partir de esta gran demanda de resultados, la calidad juega un papel importante sobre la implementación de procesos. Por esto, que los procesos deben pasar por etapas de análisis, según su jerarquía y puntos de control, que garanticen su correcta implementación.

En la actualidad los profesionales no cuentan con un proceso de formación adecuado para este tipo de especialidad, por lo que su labor se dificulta; pero, gracias al esfuerzo de estándares internacionales y la demanda del mercado, su interés y eficiencia crece día a día.

Un ejemplo de este crecimiento lo podemos apreciar en la presentación, por parte de Japón, de robots que hablan y caminan (robot HRP-4C, 2009) con características muy importantes de movimiento, imagen y funcionalidad.

Sin lugar a dudas, este mercado seguirá brindando grandes avances hacia proyectos de innovación, y la integración de tecnologías y técnicas en mecánica, electrónica, robótica y domótica que seguirán sorprendiendo al mundo entero.

Referencias bibliográficas

- Automatas.org, Software, recuperado el 15 de febrero del 2009.
- AutoWare, Software de simulación de PLC, recuperado el 14 de marzo del 200.
- Delta, Smalls work shop, recuperado el 14 de marzo del 2009, <http://smallsworkshop.com/content/plc.htm>
- Genia, Entornos de automatización, Introducción al Estándar IEC-61131, recuperado el 7 de marzo del 2009.
- Periódico Al Día, Nuevo robot habla y camina, recuperado el día, http://www.aldia.cr/ad_ee/2009/marzo/17/internacionales1902318.html
- PLCopen, Applications, recuperado el 15 de febrero del 2009.
- PLC Concepts, System and controls solution, recuperado el 25 de febrero del 2009, <http://www.plcconcepts.com/>
- Siemens, Autómatas programables, Customer Support, <http://olmo.pntic.mec.es/jmarti50/descarga/automatas.html>.
- Schneider Concept, Modsoft software and Industrial terminals, recuperado el 24 de febrero del 2009, http://www.schneider-electric.com.br/modulos/downloads/arquivos/BR/plc_quantum_programa_cao.pdf.
- Schneider Electric, Unity, Telemecanique, recuperado el 25 de febrero de 2009, <http://www.schneider-electric.com.ar/unity/>.
- Thelearningpit, LogixPro, Laboratorio Introductorio a Lógica de Relé, recuperado el 7 de marzo del 2009, <http://www.thelearningpit.com/lp/doc/nrl/intro-rl-span.html>.
- Universidad de los Andes (2003), iniciación en CONCEPT V2.5, Laboratorio De Control.
- Universidad de Oviedo (2009), Sistemas domóticos basados en PLC's, recuperado el 7 de febrero del 2009.
- Wikipedia (2006), SCADA, recuperado el 7 de febrero del 2009.
- Wikipedia (2009), Domótica, recuperado el 22 de febrero del 2009.