

LISTEN.
THINK.
SOLVE.SM

Arquitectura integrada y control multidisciplinar

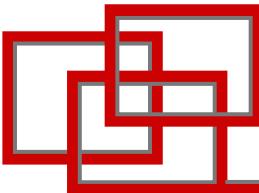


Logix

Agustín Juncal

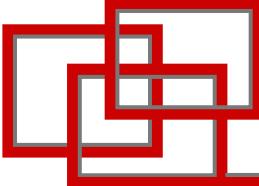
ALLEN-BRADLEY • ROCKWELL SOFTWARE • DODGE • RELIANCE ELECTRIC

**Rockwell
Automation**



Agenda

- Realidad industrial y exigencia de mercado
- El control multidisciplina, beneficios y ventajas:
 - Bases de datos de Tags. Alias
 - Estructura de datos semejante a la aplicación física
 - Multi lenguaje
 - Multitarea
 - Multiplataforma
 - ControlLogix
 - CompactLogix
 - FlexLogix
 - SoftLogix
 - DriveLogix
- La herramienta única de programación
 - Tareas
 - Rutinas
 - Programas
 - Fases (Modelo S88)
 - Seguridad (Safety)
 - RSLogix Architect
- Integración del control y la información
- Beneficios y ventajas



¿Rockwell Automation?

- Adquisición de Allen-Bradley en 1985
- Primer paso en la construcción de un “Nuevo Rockwell”
- En los siguientes 10 años, Rockwell invirtió más de 54 billones de Pts para reforzar su posición en el mercado
- Adquisición de Reliance Electric en 1995
- Seguimos creciendo: EJA, Tecsh....
- Refuerzo de su presencia mundial

Allen-Bradley

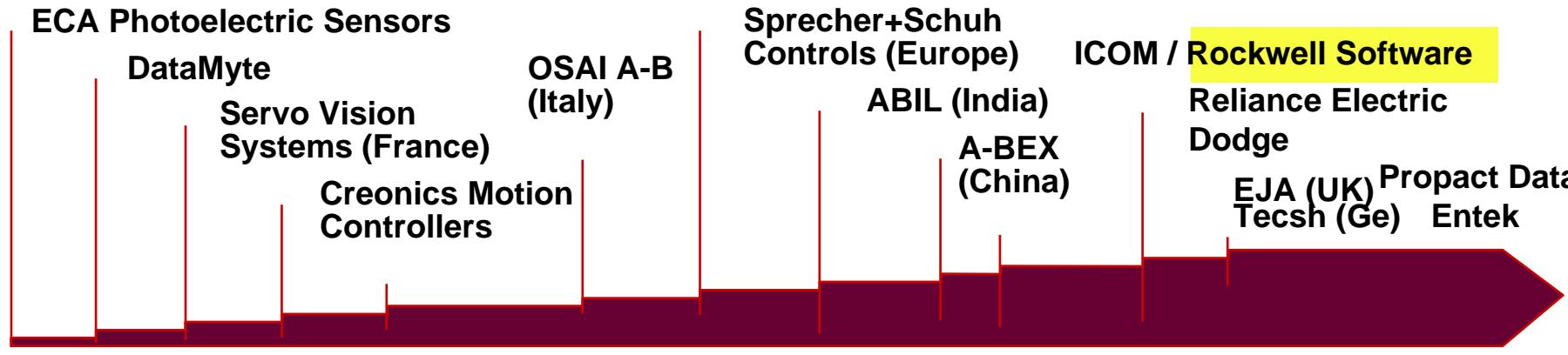
ECA Photoelectric Sensors
DataMyte
Servo Vision Systems (France)
Creonics Motion Controllers

Metal Leve Control Products (Brazil)

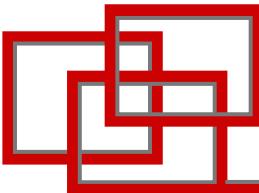
OSAI A-B (Italy)
Sprecher+Schuh Controls (Europe)
ABIL (India)
A-BEX (China)

ICOM / Rockwell Software

Reliance Electric
Dodge
EJA (UK) Propact Data
Tecsh (Ge) Entek



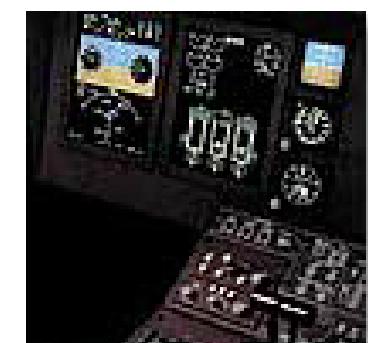
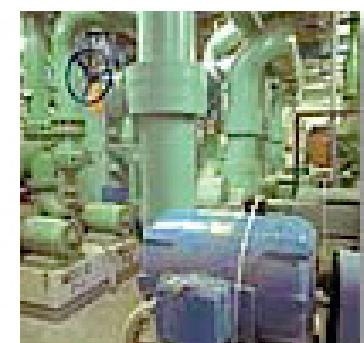
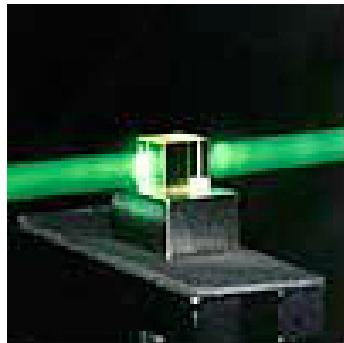
1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 2000 2003

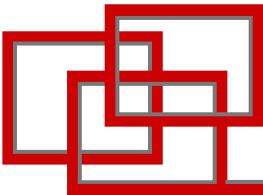


Rockwell hoy

- Sede Central : Milwaukee (Wi) USA
- Una compañía con unas ventas > de 5.000 M\$
- Cerca de 25,000 empleados
- Más de 200 fábricas en más de 80 países

**LA MAYOR COMPAÑÍA DEDICADA EXCLUSIVAMENTE A
AUTOMATIZACIÓN INDUSTRIAL**





Algunos datos históricos

- 1970 introducción del primer PLC
- 1979 Introducción de la DataHighway, la primera red de comunicaciones de planta
- Publicación de las especificaciones de DataHighway (apertura)
- DataHighway primer estandard de la industria de facto
- 1988 conjuntamente con Digital Equipment Corp., A-B desarrolló el Pyramid Integrator, el primer sistema en integrar los mundos de información y proceso productivo
- 1988-1994 Lanzamiento de diversos productos con conectividad Ethernet y TCP/IP
- 1994 introducción de DeviceNet, una red abierta de dispositivos de campo que se convirtió en un estandard de facto rápidamente

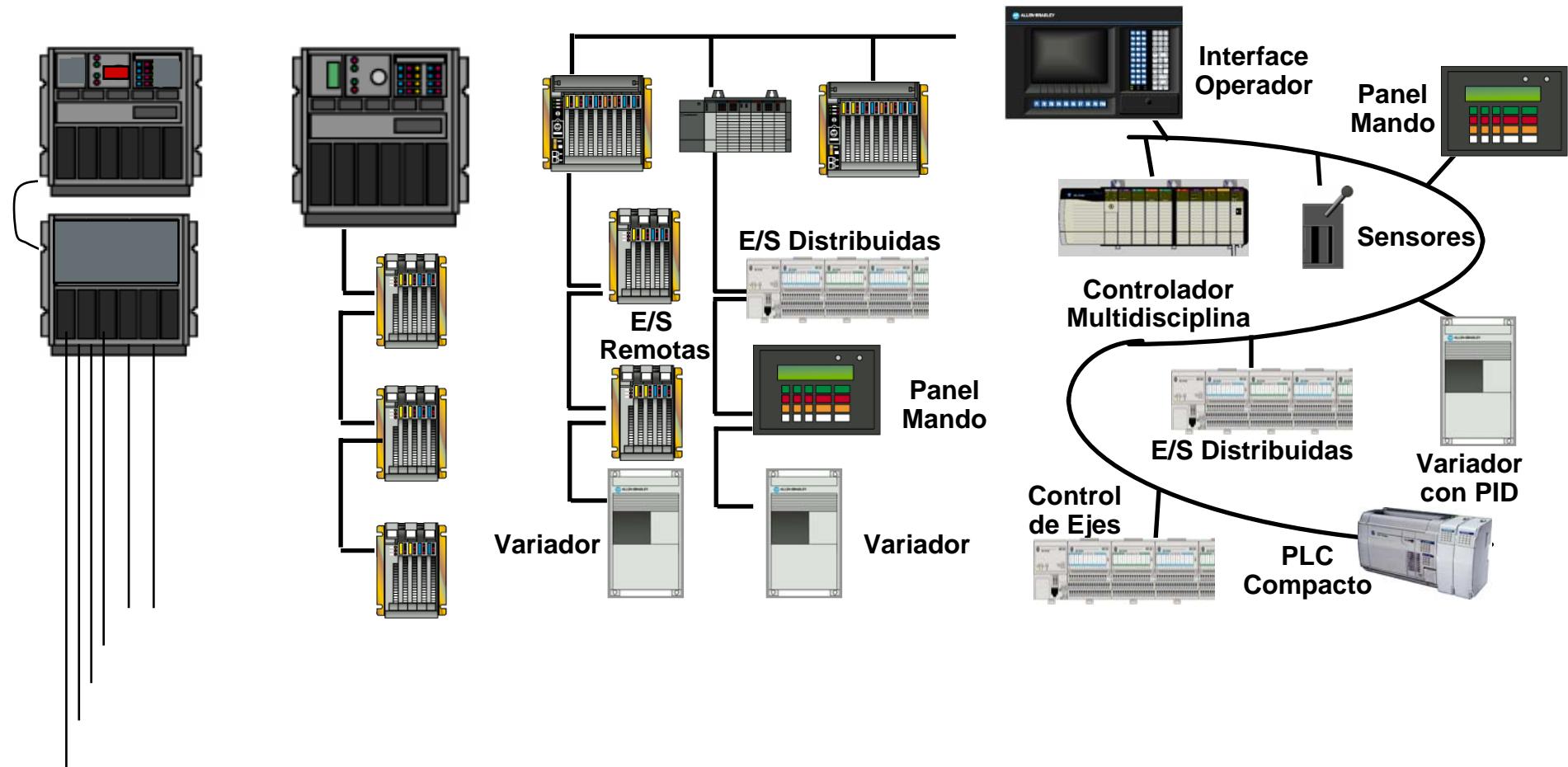
Evolución de los PLCs

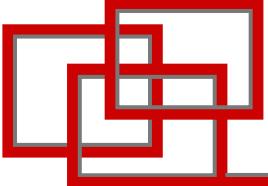
1970

1980

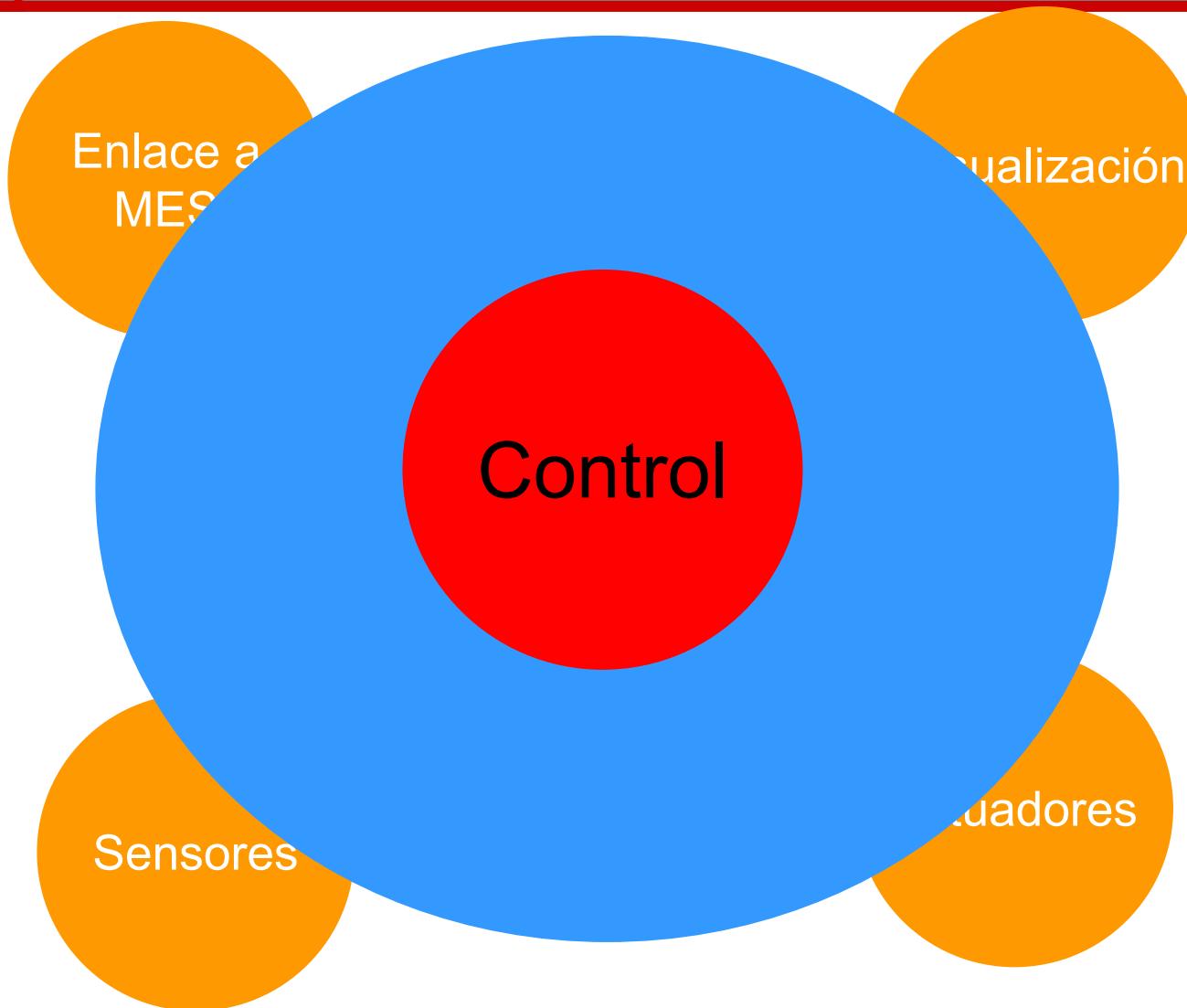
1990

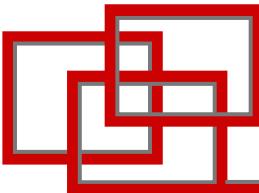
2000





Componentes de la arquitectura de automatización





La realidad del mercado

¡ Los clientes están bajo presión !

- Optimizar Fabricación
- Mejorar producción
- Agilizar respuesta
- Adaptarse a las regulaciones

Fortius
Altius
Citius

¡ Los proveedores de automatización estamos bajo presión!

Sistemas de Información

Proceso *Discreto* *Motion*

- Presión de precios
- Crecimiento en nuevas disciplinas y/o
- Crecimiento hacia la Integración

Brevius
Citius

La presión nos lleva a “Aproximaciones de sistema”

Yokogawa “Engineering for the next generation”

ABB

Emerson

Honeywell

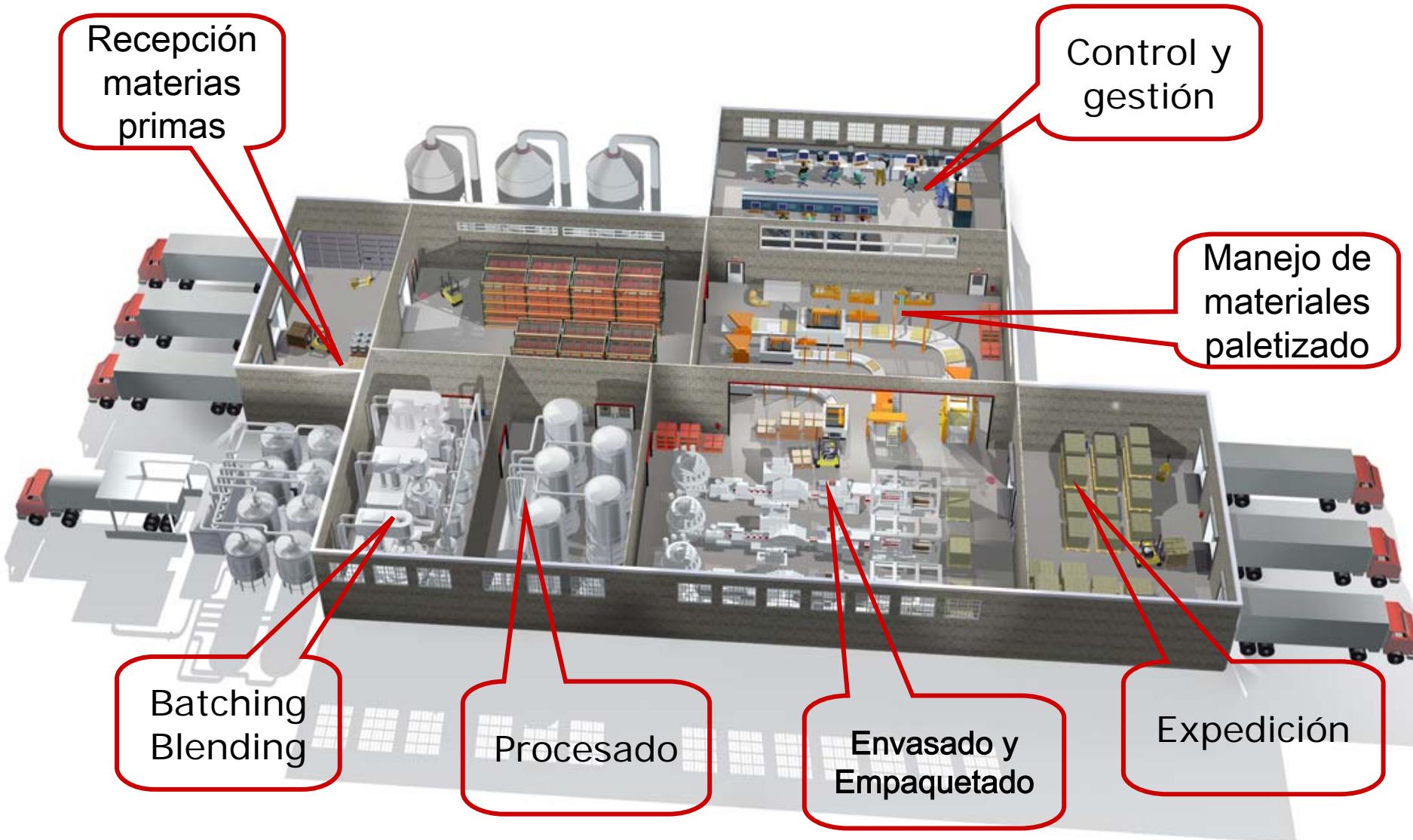
Inventec

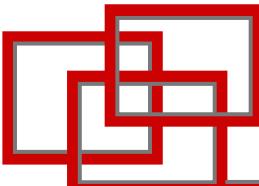
Siemens

Mientras la competencia habla...

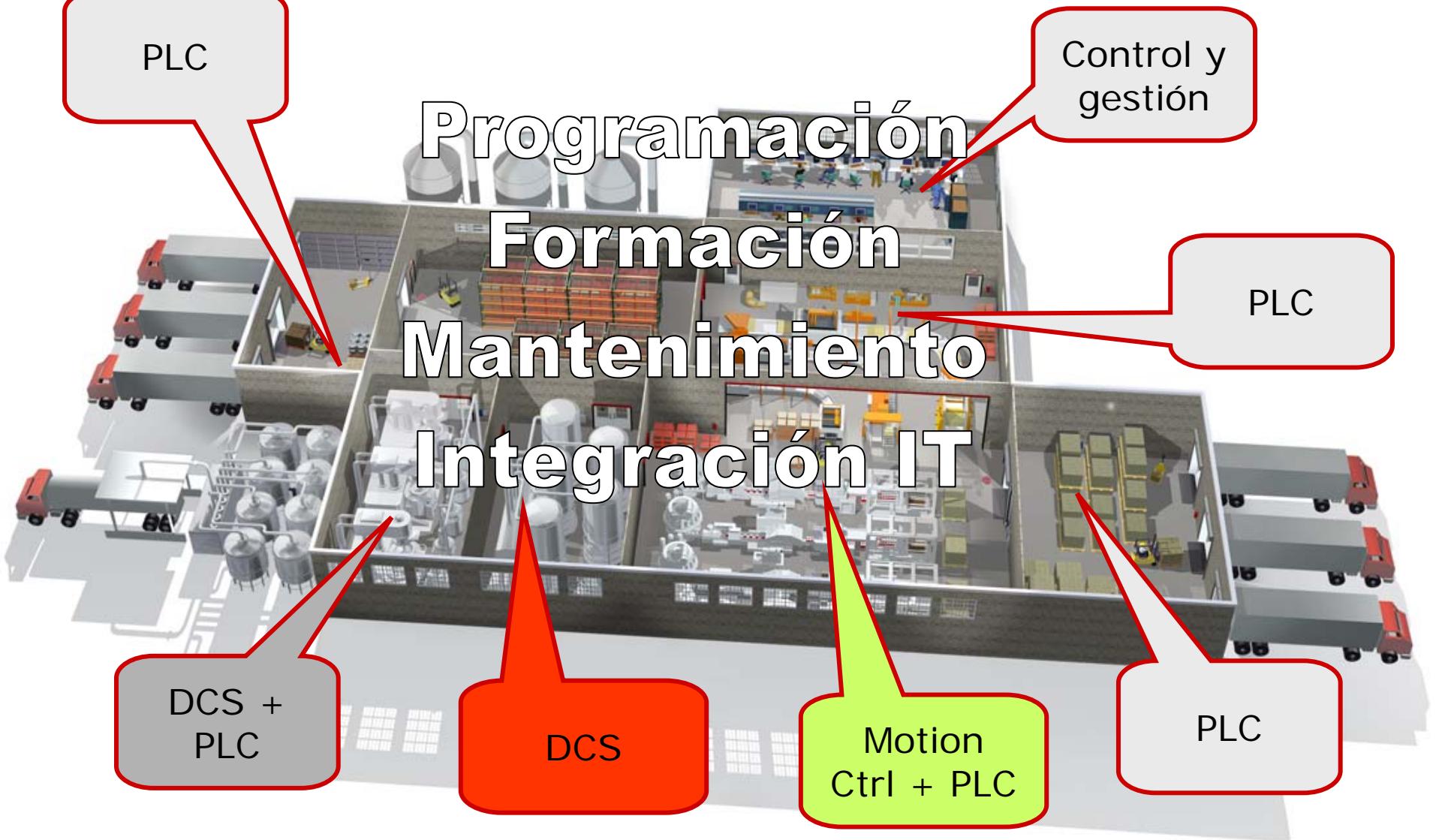
Rockwell suministra Integrated Architecture!

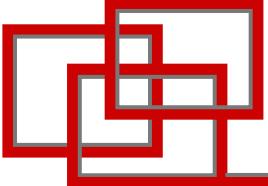
Planta típica de fabricación





Solución tradicional = €€€





Automatización convencional

Aproximación Convencional

Multiples
Procesadores

Secuencial

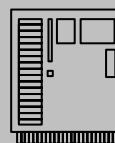


Motion



Multiples
Paquetes de
Software

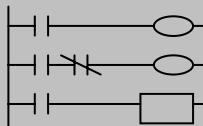
Proceso



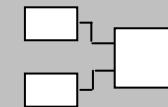
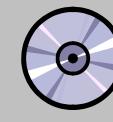
Drive



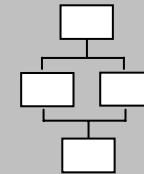
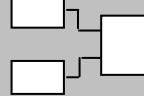
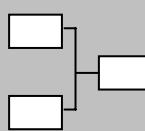
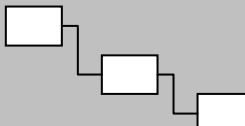
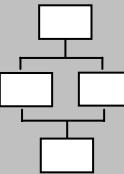
Differentes
Programas y
Lenguajes



Motion Servo

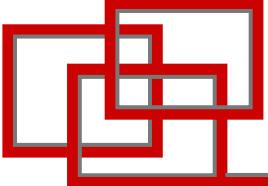


Multiple
Personal de
ingeniería y
soporte

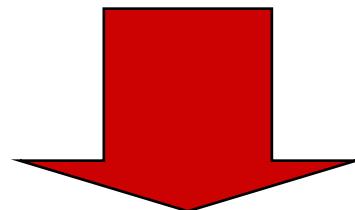


```
IF A > B THEN  
    F = B * C / D;  
END_IF;
```

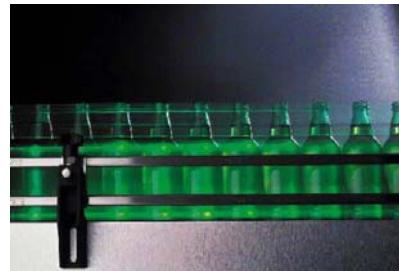




Un Control para todo.....LOGIX



sequential
control



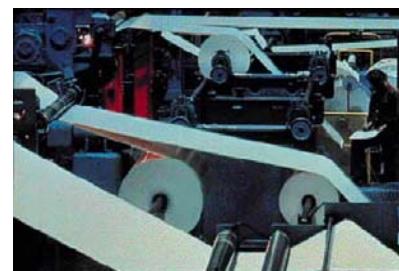
process
control

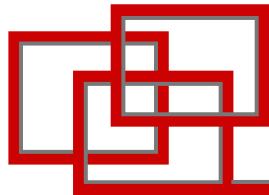


motion
control



drives
control



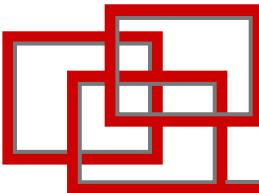


Servicios Arquitectura Automatización

*Servicios
Control*

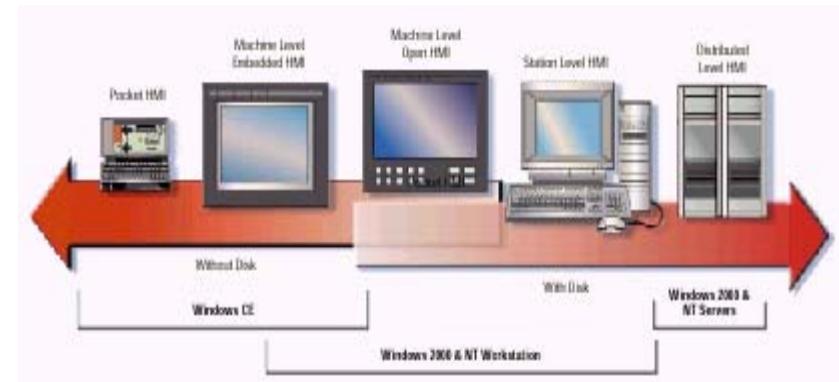
*Servicios
Comunicación*

*Servicios
HMI*

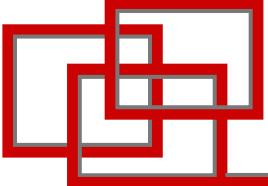


Servicios HMI

- Multidisciplina
 - Mantenimiento
 - Operación
 - Visualización
 - Adquisición de datos
- Independencia de Plataforma
 - Dispositivos Handheld
 - Paneles de operador
 - Estaciones de trabajo
 - Sistemas Cliente / Servidor

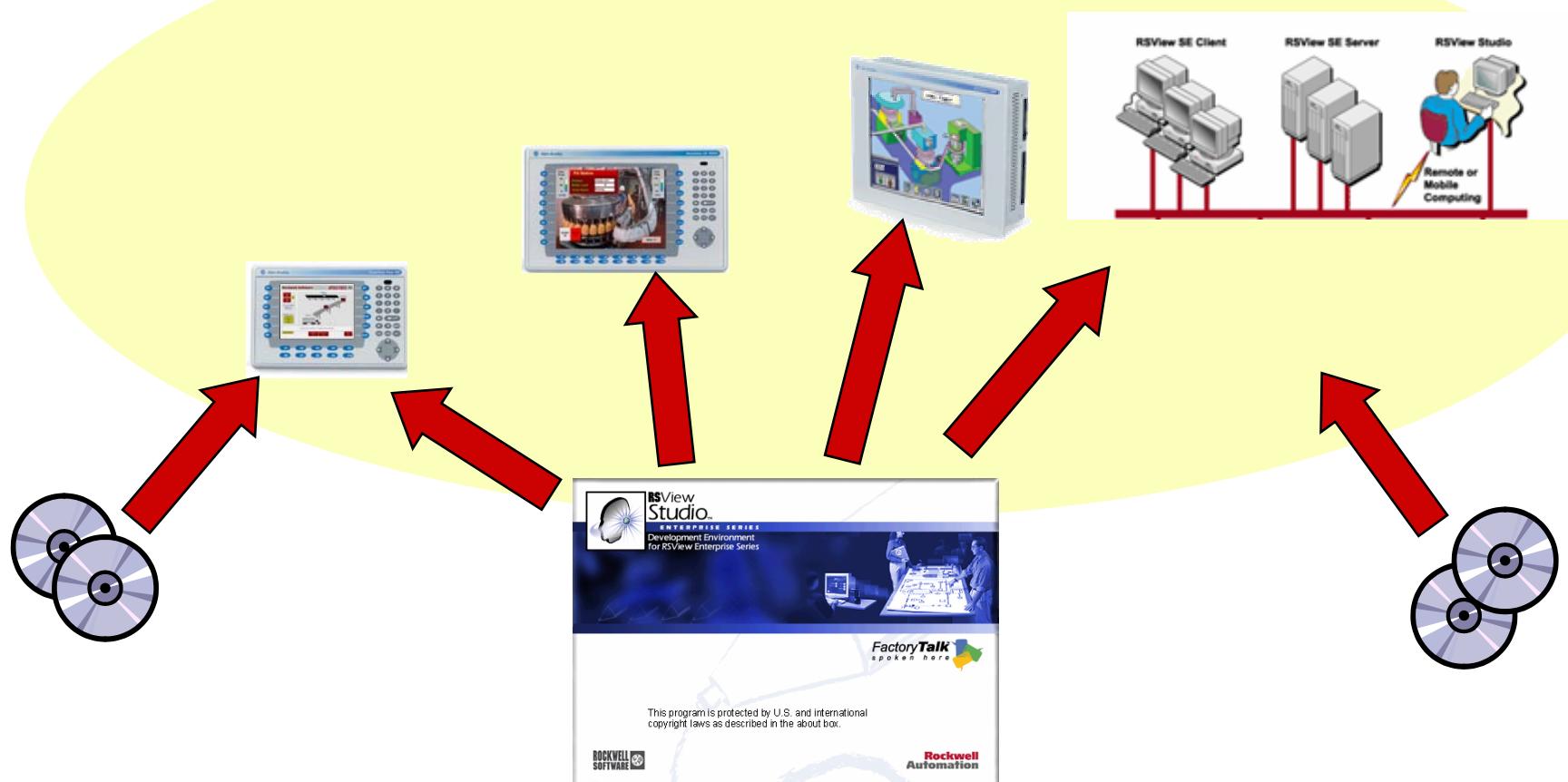


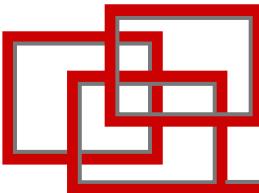
View



Beneficios de ViewAnyWare: Unico entorno de desarrollo - RSView Studio

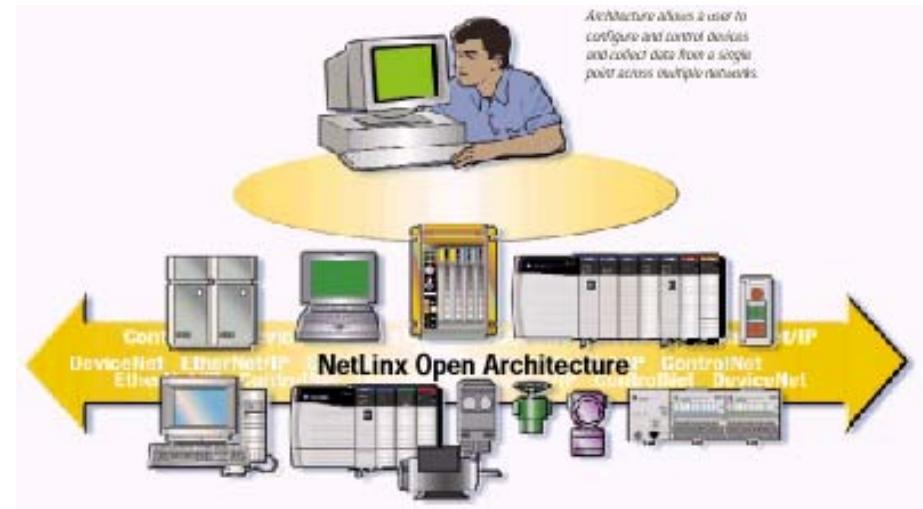
- Desde nivel de máquina al nivel de supervisión distribuida.
- Desde terminales dedicados (PV Plus) a plataformas con sistemas operativos abiertos Windows CE o Windows 2003 Server/XP.





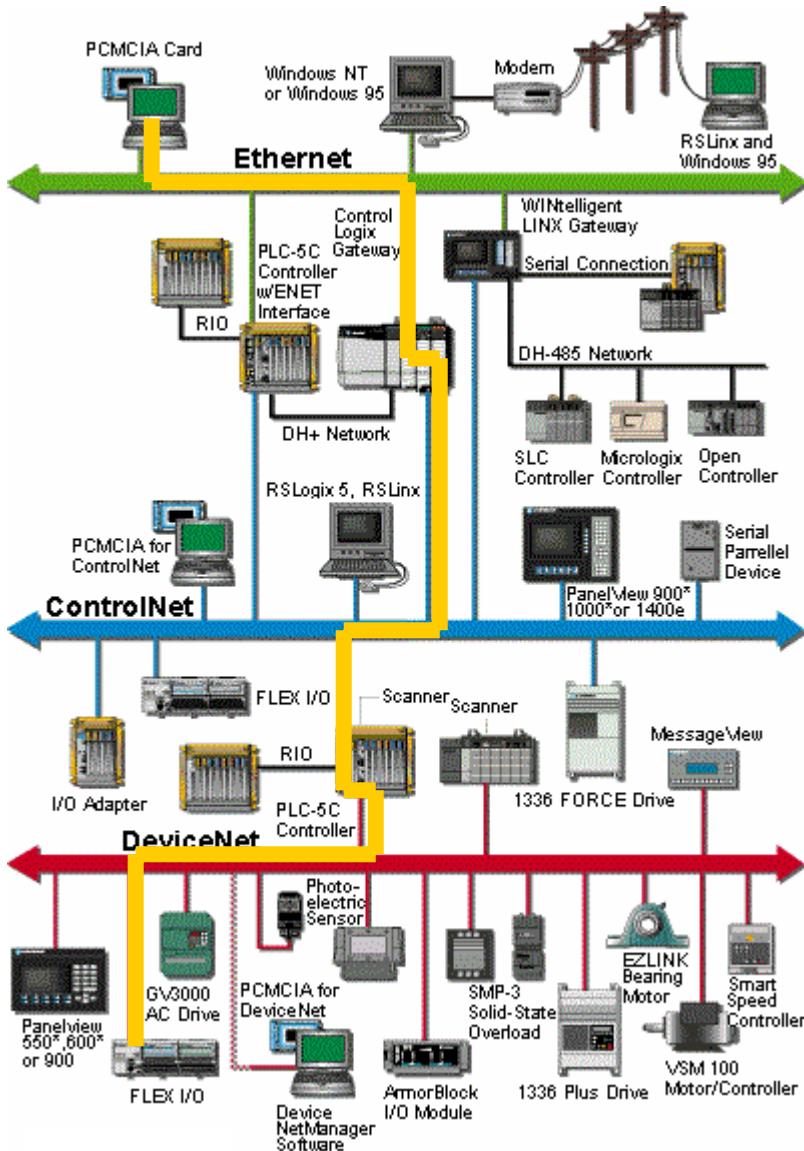
Servicios Comunicaciones

- Multidisciplina
 - Controlar
 - Configurar
 - Concentrar
- Independencia de Plataforma
 - EtheNet/IP
 - ControlNet
 - DeviceNet
 - ControlBus

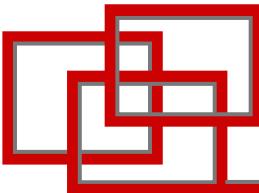


NetLinx

Información en lugar de datos con NetLinx

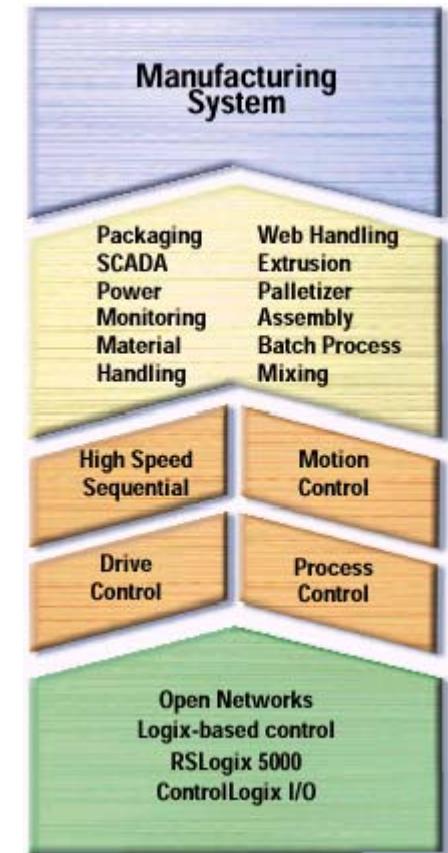


- Permite que la información fluya desde cualquier parte del sistema
 - Dispositivos en diferentes redes pueden comunicar entre ellos
- Desde una simple workstation sobre redes heterogeneas networks permiten al usuario:
 - Identificar y configurar todos los dispositivos
 - Programar todos los dispositivos de control
 - Obtener información de cualquier dispositivo
 - Monitorizar el estado de cualquier dispositivo



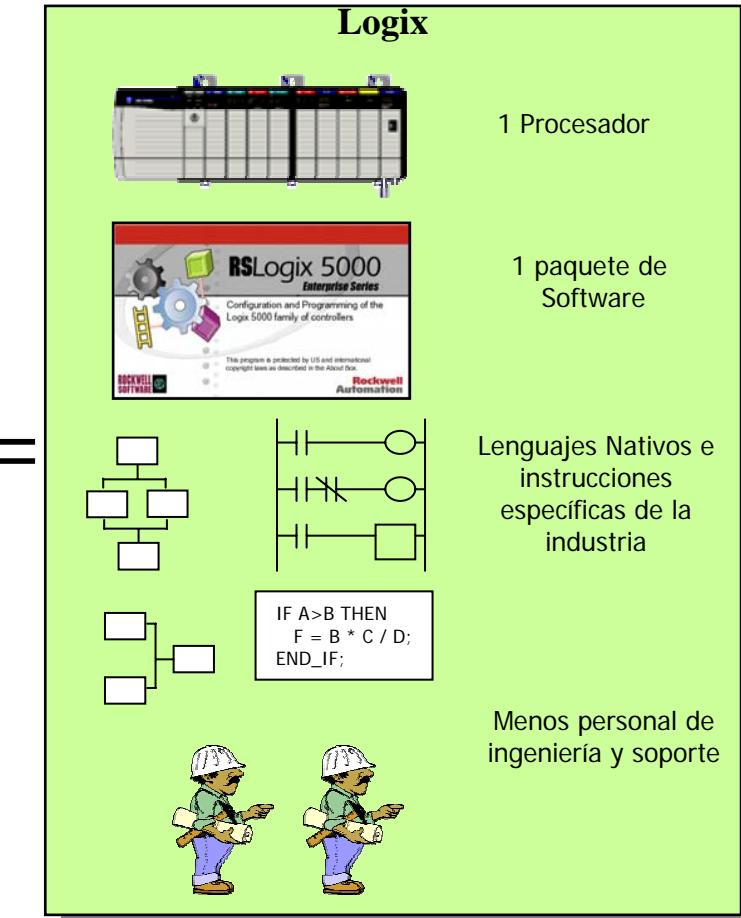
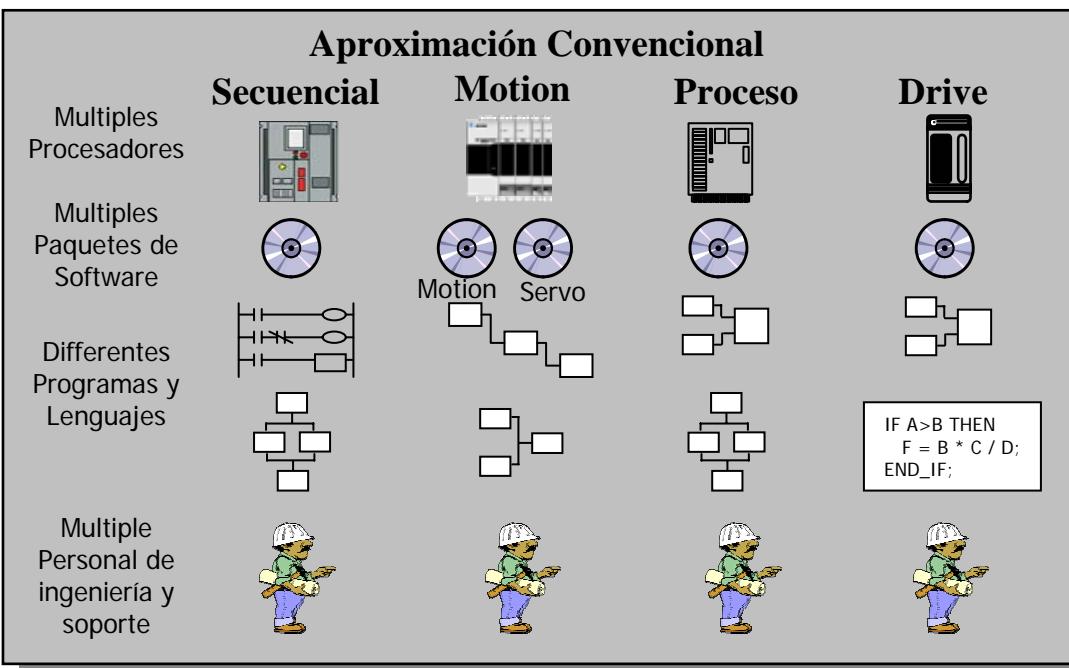
Servicios de Control

- Multidisciplina
 - Control Secuencial
 - Control de Movimiento
 - Control de Procesos
 - Control de Drives
- Independencia de Plataforma
 - Modular
 - Distribuido
 - Incrustado
 - Compacto
 - Abierto



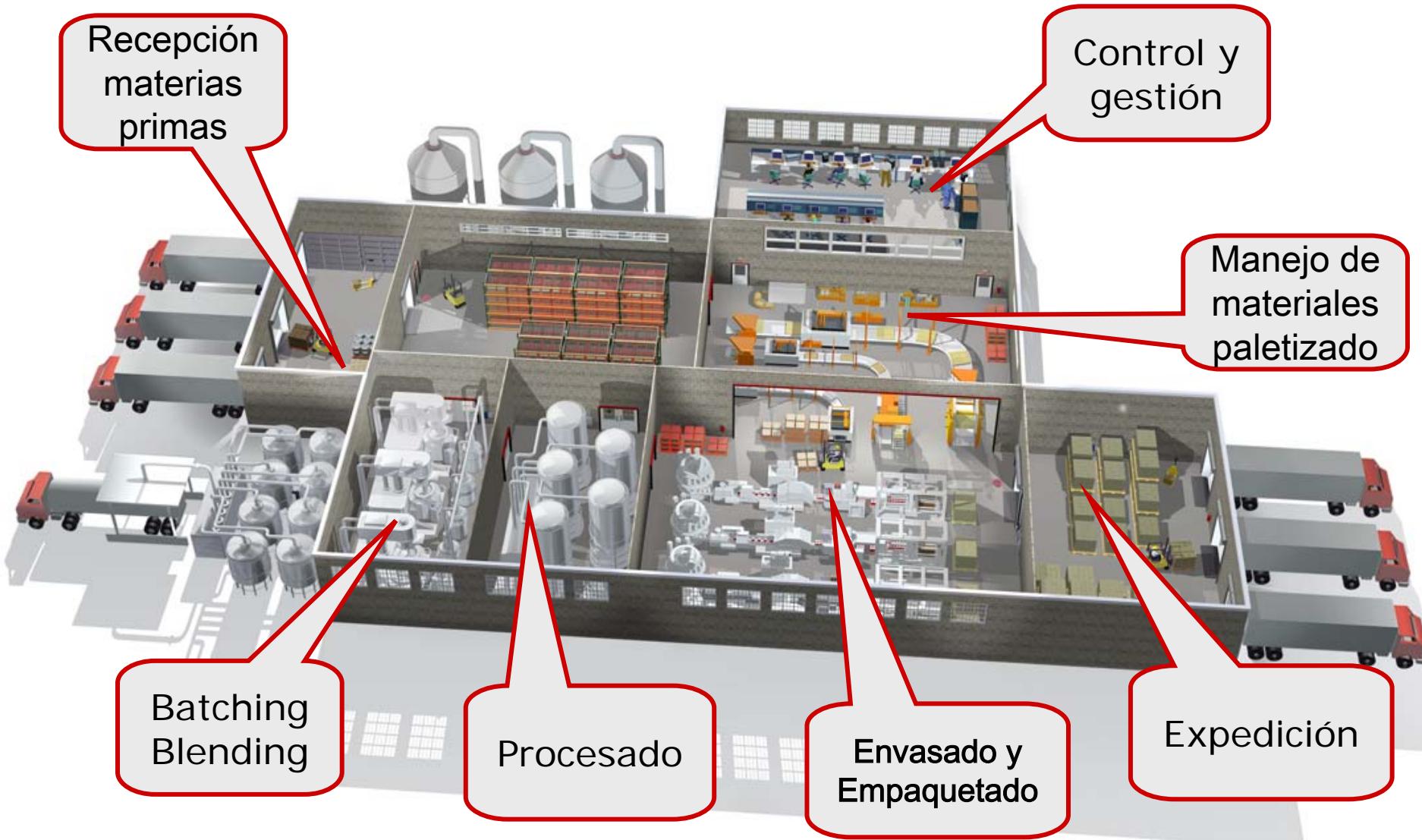
Logix

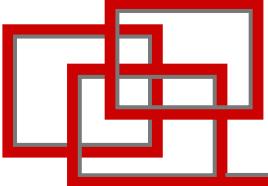
La evolución de los sistemas de control. PAC



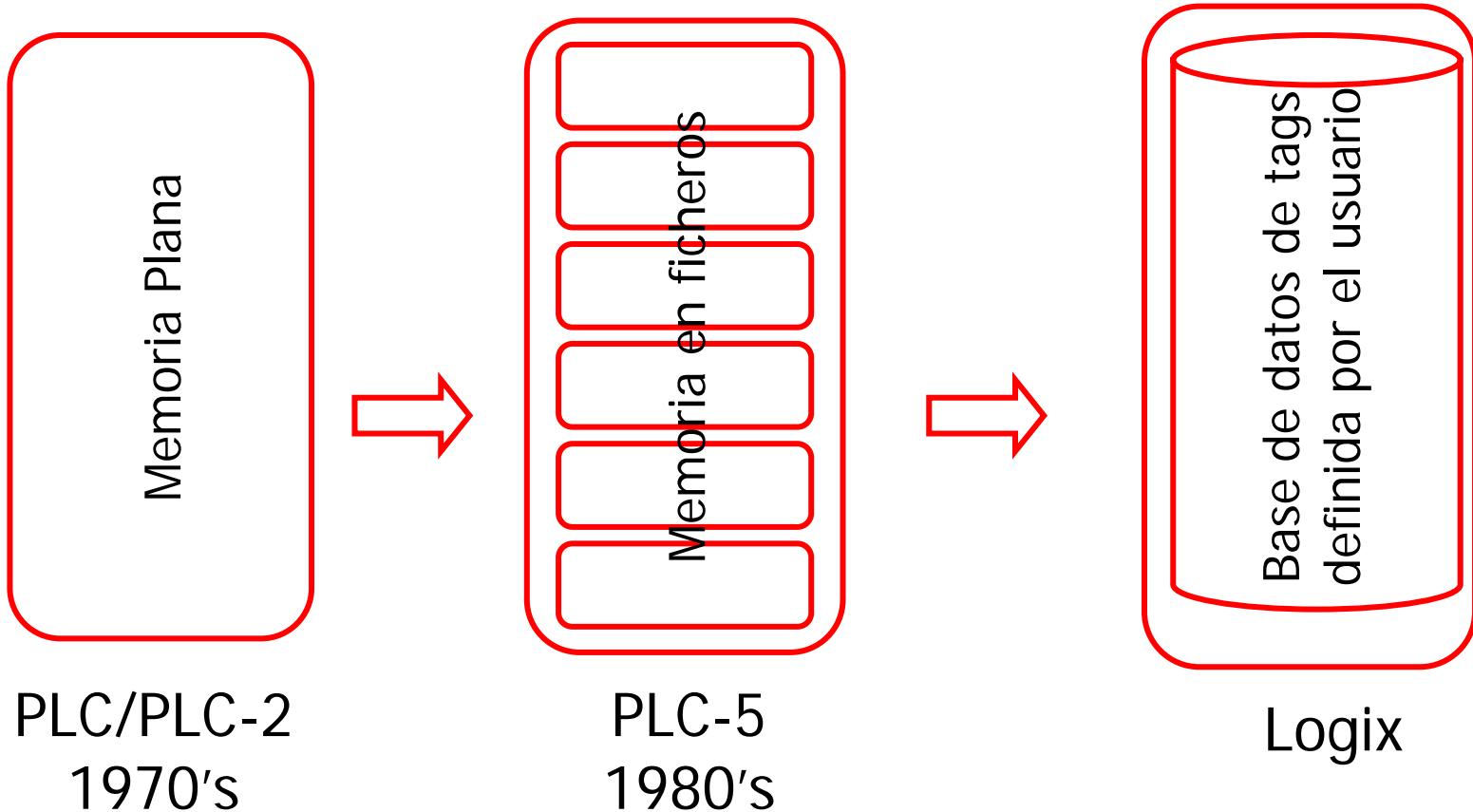
ARC Advisory Group definió ControlLogix como Programmable Automation Controller

Planta típica de fabricación

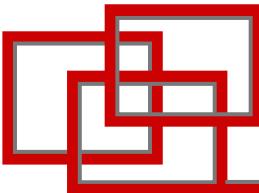




Evolución del modelo de memoria



- Un modelo avanzado de memoria simplifica la integración



Ahorro de costes de integración

BD del ordenador

Mesa10.EnPosicion
Robot5.TiempoCiclo

Memoria del PLC



210/13
375

Memoria Plana

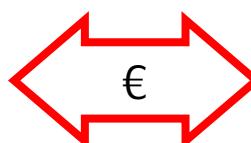
Mesa10.EnPosicion
Robot5.TiempoCiclo



B110/13
T205.ACC

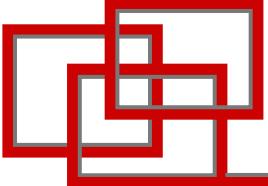
Memoria en ficheros

Mesa10.EnPosicion
Robot5.TiempoCiclo



Mesa10.EnPosicion
Robot5.TiempoCiclo

Base de datos de tags
definida por el usuario

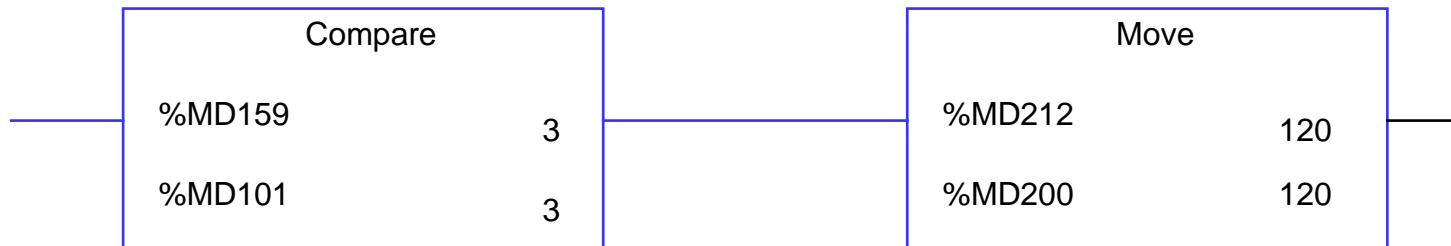


Valor de Logix: Direccionamiento basado en Tags

- Los Controladores almacenan datos usando nombres descriptivos.
- Los datos se nombran con nombres apropiados a su uso.
- Los Tags se almacenan directamente en el controlador.
- El Código queda auto-documentado

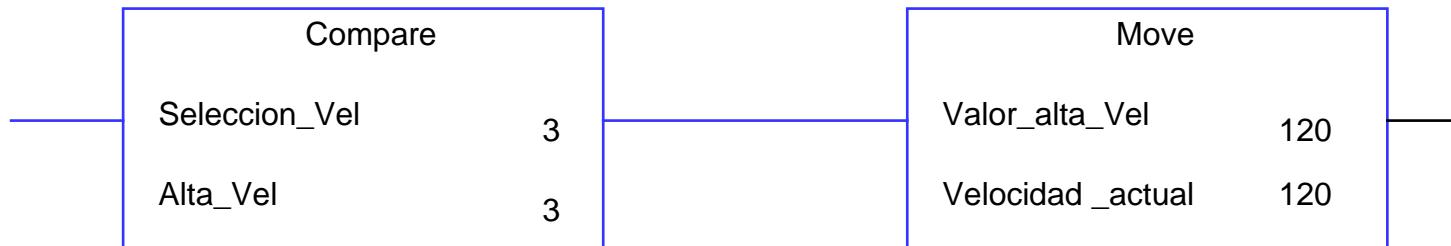
Convencional

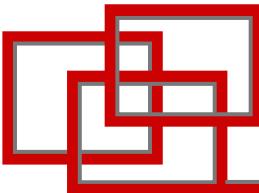
Simbolos



Logix

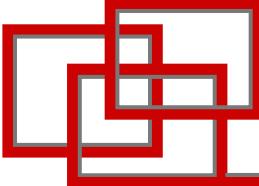
Nombre de tag





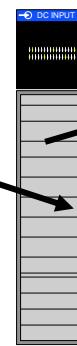
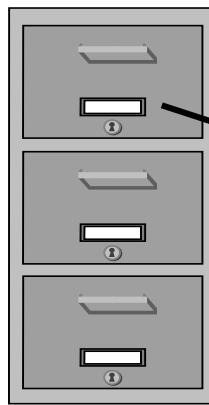
Bases de datos de Tags “Logix”

- Tipos de datos predefinidos
 - BOOL, INT, SINT, DINT, REAL
 - TIMER, COUNTER, MSG, AXIS
 - Datos Módulos E/S y perfiles de dispositivos
 - Información canal, Diagnósticos, Configuración
- Tipos de datos definidos por el Usuario
 - Estructuras de datos complejas para imitar el proceso
 - MOTOR, TANQUE, MEZCLADOR, VALVULA
- Matrices de estructuras
 - 1, 2 ó 3 dimensiones



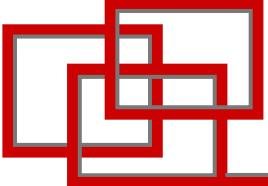
El Procesador Logix5550. Tipos de datos (Módulos)

Módulos



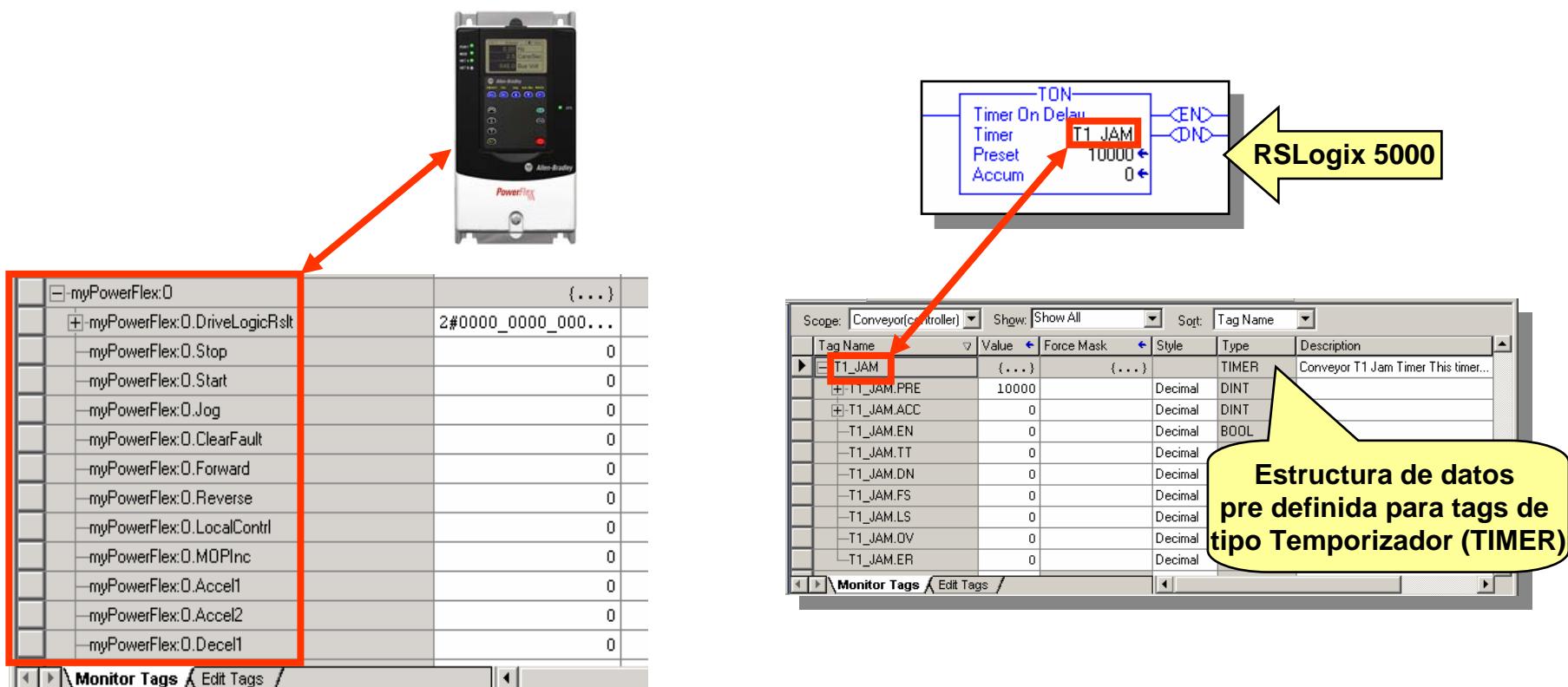
- Al añadir un módulo en la configuración de I/O, el sistema crea de forma automática la estructura de datos necesaria.

	Tag Name	Value	Force Mask
►	+Local:2:C	{ ... }	{ ... }
►	-Local:2:I	{ ... }	{ ... }
►	+Local:2:I.Fault	2#0000...	
►	-Local:2:I.Data	2#0000...	
	Local:2:I.Data.0	0	
	Local:2:I.Data.1	0	
	Local:2:I.Data.2	0	
	Local:2:I.Data.3	0	
	Local:2:I.Data.4	0	
	Local:2:I.Data.5	0	
	Local:2:I.Data.6	0	
	Local:2:I.Data.7	0	
	Local:2:I.Data.8	0	
	Local:2:I.Data.9	0	
	Local:2:I.Data.10	0	
	Local:2:I.Data.11	0	
	Local:2:I.Data.12	0	
	Local:2:I.Data.13	0	
	Local:2:I.Data.14	0	
	Local:2:I.Data.15	0	
	Local:2:I.Data.16	0	



Valor de Logix: Creación Automatica de estructuras

- RSLogix 5000 crea automáticamente tipos de datos y estructuras “enlatadas” para instrucciones, módulos de E/S y dispositivos.



The diagram illustrates the automatic creation of data structures in RSLogix 5000. It shows a PowerFlex 400 drive, the RSLogix 5000 software interface, and a tag table.

PowerFlex 400 Drive: A physical Allen-Bradley PowerFlex 400 variable frequency drive unit.

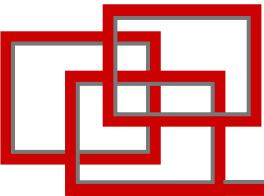
RSLogix 5000 Software: A screenshot of the RSLogix 5000 software interface. It shows a ladder logic program with a TON (Timer On Delay) instruction. The timer tag is highlighted with a red box. An arrow points from this tag to the tag table below.

Tag Table: A screenshot of the "Monitor Tags" window in RSLogix 5000. The "T1_JAM" tag is highlighted with a red box. An arrow points from this tag to the tag table below.

Tag Table Data: The tag table displays the following data for the "T1_JAM" tag:

Tag Name	Value	Type	Description
T1_JAM.PRE	10000	Decimal DINT	
T1_JAM.ACC	0	Decimal DINT	
T1_JAM.EN	0	Decimal BOOL	
T1_JAM.TT	0	Decimal	
T1_JAM.DN	0	Decimal	
T1_JAM.FS	0	Decimal	
T1_JAM.LS	0	Decimal	
T1_JAM.OV	0	Decimal	
T1_JAM.ER	0	Decimal	

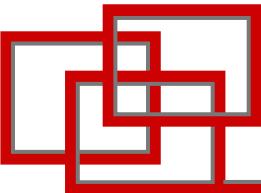
Annotation: A yellow callout bubble points to the "T1_JAM" tag in the tag table with the text: "Estructura de datos pre definida para tags de tipo Temporizador (TIMER)".



Beneficios de Logix: Creación automática de estructuras

- Reduce tiempo de programación
 - Elimina la necesidad de crear estructuras de Tags de E/S.
 - Elimina la necesidad de crear estructuras de Tags de instrucciones (timers, counters, etc.)
 - Elimina el código asociado al análisis de diagnósticos y estatus.
- Mejora el rendimiento del procesador al eliminar todo el código extra necesario para crear estructuras y acceder a valores de diagnósticos y estatus.





Valor de Logix: Datos definidos por el usuario

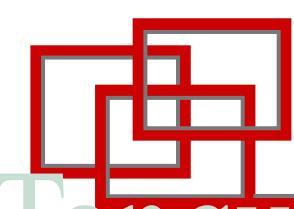
Los tipos de datos definidos por el usuario permiten reproducir una imagen de la aplicación en la base de datos del procesador Logix (Global o locales)

- Utilizar nombres comúnmente entendibles
- Agruparlos en estructuras que mejoran la organización y que reproducen la estructuración física de los datos en los dispositivos de la aplicación

Name:	Tank	Size:	16 byte(s)																									
Description:	Generic Storage Tank Data Type																											
Members:	<table border="1"><thead><tr><th>Name</th><th>Data Type</th><th>Style</th><th>Description</th></tr></thead><tbody><tr><td>Level</td><td>INT</td><td>Decimal</td><td>Stores the Level in Inches</td></tr><tr><td>Pressure</td><td>DINT</td><td>Decimal</td><td>Stores the Pressure in PSIG</td></tr><tr><td>Temp</td><td>REAL</td><td>Float</td><td>The Temperature in F</td></tr><tr><td>Agitator_Speed</td><td>DINT</td><td>Decimal</td><td>Speed in RPM</td></tr><tr><td>*</td><td></td><td></td><td></td></tr></tbody></table>				Name	Data Type	Style	Description	Level	INT	Decimal	Stores the Level in Inches	Pressure	DINT	Decimal	Stores the Pressure in PSIG	Temp	REAL	Float	The Temperature in F	Agitator_Speed	DINT	Decimal	Speed in RPM	*			
Name	Data Type	Style	Description																									
Level	INT	Decimal	Stores the Level in Inches																									
Pressure	DINT	Decimal	Stores the Pressure in PSIG																									
Temp	REAL	Float	The Temperature in F																									
Agitator_Speed	DINT	Decimal	Speed in RPM																									
*																												

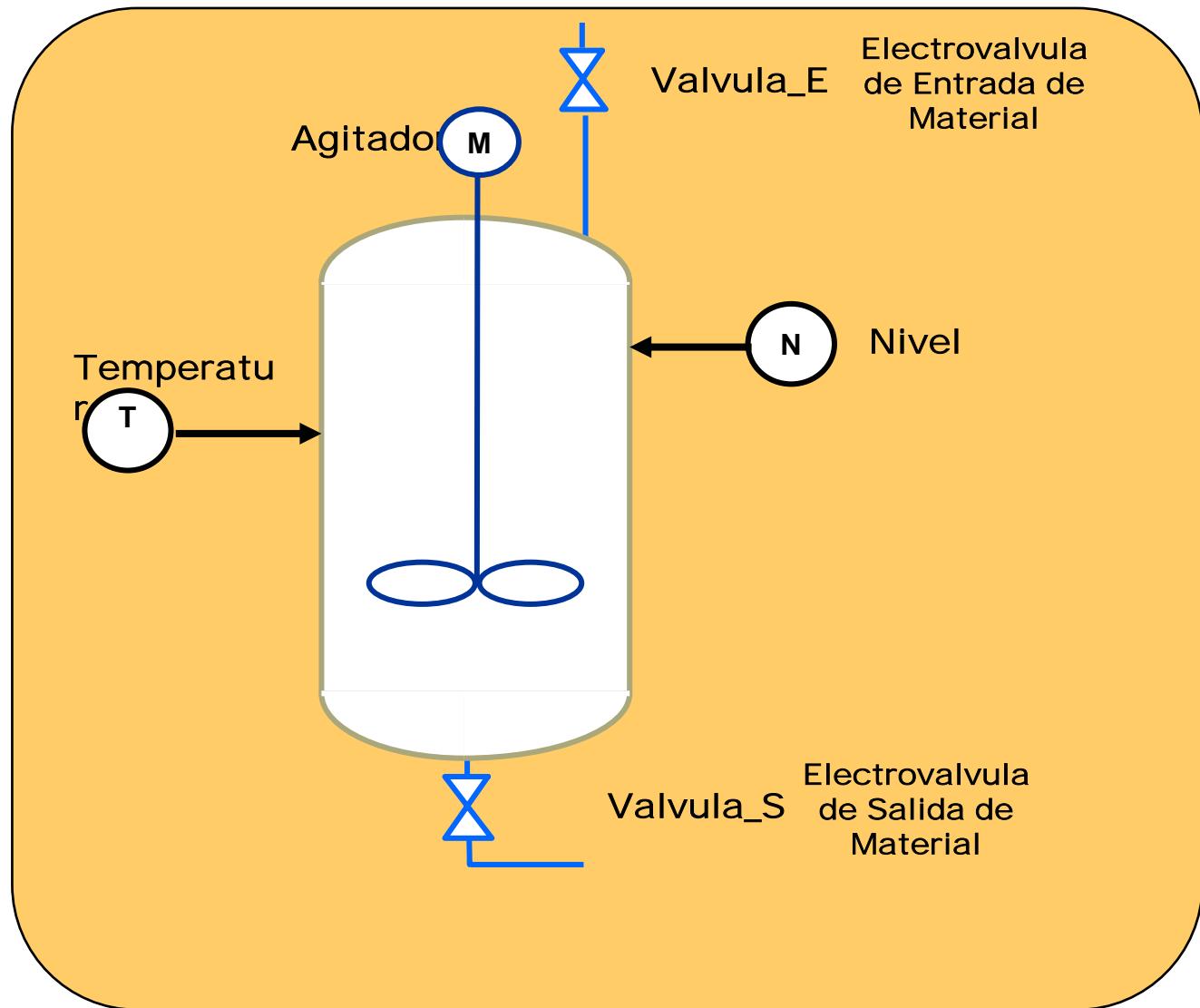


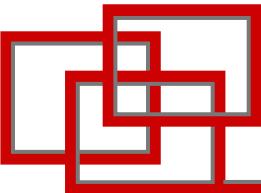
+ - Tankroom[2,0,1]	(...)
- Tankroom[2,1,0]	(...)
+ - Tankroom[2,1,0].Level	99
+ - Tankroom[2,1,0].Pressure	200
- Tankroom[2,1,0].Temp	140.0
+ - Tankroom[2,1,0].Agitator_Speed	95
+ - Tankroom[2,1,1]	(...)
+ - Tankroom[2,2,0]	(...)



Tanque
Tanque

Dispositivos del mundo real





Estructuras de datos del “Mundo Real”

Tanque

- Nivel (REAL)
- Temperatura (REAL)
- Valvula_E (VALVULA)
- Valvula_S (VALVULA)
- Agitador (MOTOR)

Válvula

- Estado (BOOL)
- Comando (BOOL)
- Com_Tiempo (TIMER)
- Alarma (BOOL)

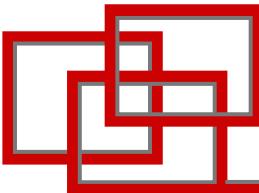
Motor

- Estado (BOOL)
- Com_Estad (BOOL)
- Velocidad (REAL)
- Con_Velocid (REAL)
- Tiempo_On (TIMER)

Tanque[20].**Valvula_E**.Estado

Tanque[5].**Agitador**.Con_Velocid

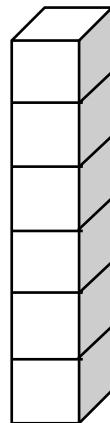
Tanque_Aceite.**Agitador**.Tiempo_On.Acc



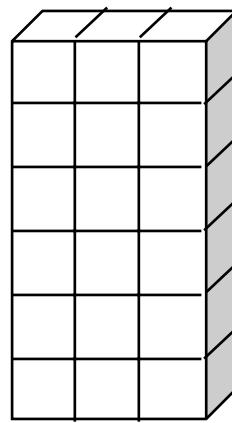
Matrices definidas por el usuario

- Creación de tablas personalizadas para almacenar información
- Las matrices se nombran con símbolos para ayudar a determinar su contenido y uso
- Matrices de 1, 2 o 3 dimensiones para representar la información que deben contener
- Pueden contener tipos de datos base (Bool, Entero, Real...) o estructuras (Temporizador, Contador, User...)

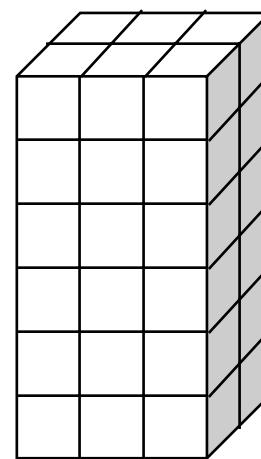
Una Dimensión



Dos Dimensiones



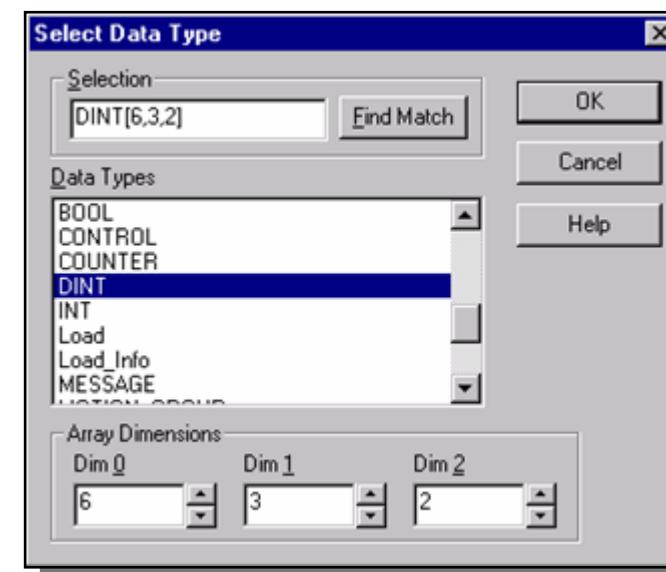
Tres Dimensiones



Tabla[6]

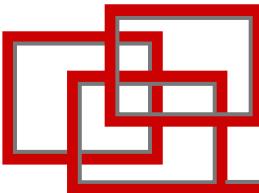
Grid[6,3]

Bloque[6,3,2]



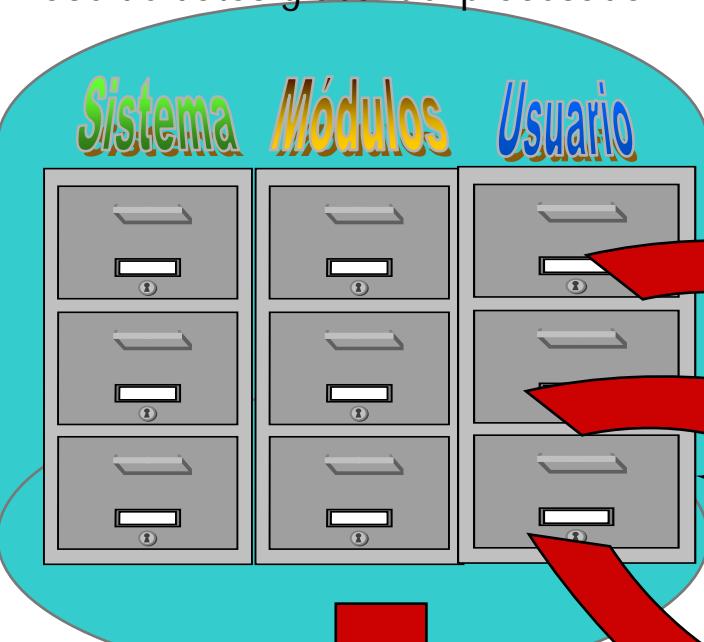
- Reduce el tiempo de ingeniería
 - Sin tener que estructurar ni predefinir las tablas de datos del procesador
 - La memoria de Logix recrea la aplicación – sin tener que forzar la aplicación para encajar con la memoria del procesador.
- Reduce el tiempo de mantenimiento – memoria fácil de leer y entender.





Procesadores Logix. Bases de datos

Base de datos global del procesador



Rutinas de Programa X

Base de datos local
del programa n+1



Rutinas de Programa Y

Base de datos local
del programa 3



Sistema Operativo Multitarea

Programas asignados a tareas

Programas están compuestos de rutinas y de una base de datos propia y exclusiva

Rutinas de Programa A

Rutinas de Programa B

Rutinas de Programa 3

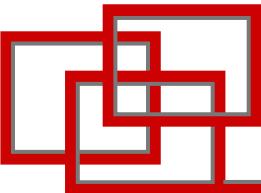
Base de datos local
del programa A



Base de datos local
del programa B

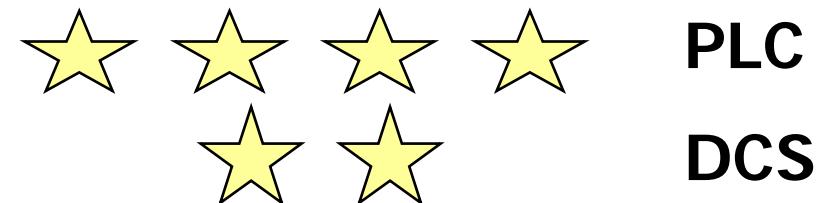


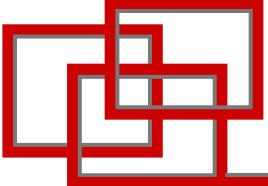
Base de datos local
del programa 3



Beneficios de Logix: Direccionamiento basado en Tags

- Reduce el tiempo de localización de averías
 - Sin necesidad de perseguir copias documentadas del programa ni preocuparse sobre bases de datos erróneas cargadas en ordenadores particulares
 - Memoria fácil de leer y entender
- Reduce tiempo de ingeniería
 - Mucha menos documentación requerida
 - Nombres de Tag reutilizados por el HMI

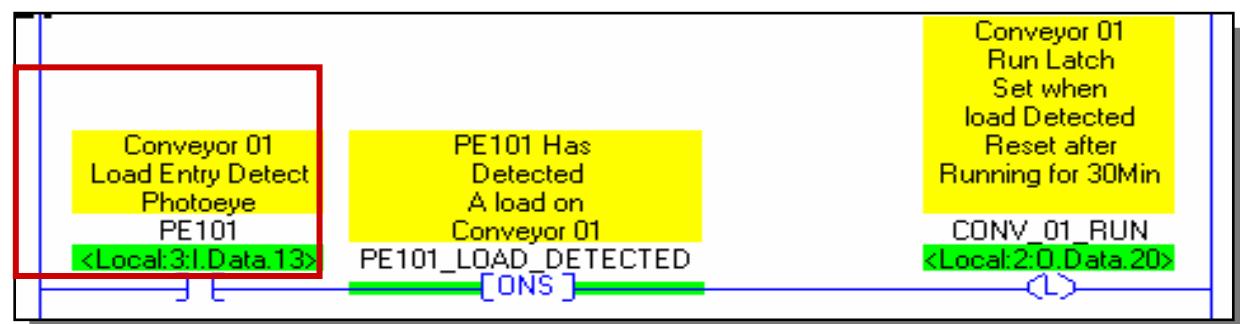


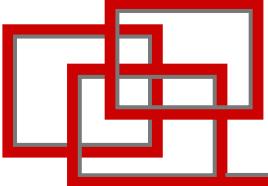


Valor de Logix: Direccionamiento por Alias y ámbito de datos

- Asociación de direcciones de tags con E/S reales
 - Elimina la relación rígida entre E/S y programa
- Permite la fácil re-utilización de código y la ingeniería concurrente
 - Evita colisiones por duplicación de nombres de tags

Tag Name	Alias For	Base Tag	Type	Style	Description
CONV_01_RUN	Local:2:0.Data.20(C)	Local:2:0.Data.20(C)	BOOL	Decimal	Conveyor 01 Run La
PE101_LOAD_DETECTED			BOOL	Decimal	PE101 Has Detecte
PE101	Local:3:I.Data.13(C)	Local:3:I.Data.13(C)	BOOL	Decimal	Conveyor 01 Load E



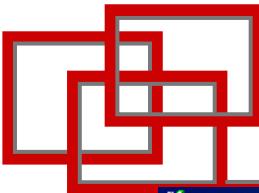


Ingeniería concurrente con Data Tag Alias

Desarrollo de un sistema
de control Típico



Ingeniería concurrente con Data Tag Alias



RSLogix 5000 - ControlLogix [1789-L60]* - [MainProgram - MainRoutine]

Archivo Edición Ver Búsqueda Lógica Comunicaciones Herramientas Ventana Ayuda

Fuera de lín. RUN
Sin forzados OK
Sin ediciones BAT
I/O

Ruta de acceso: <ninguno>

Bit Timer/counter Entrada/Salida Comparar

Controller ControlLogix

- Tag de controller
- Administrador de fallos de encendido
- Administrador de encendido

Tasks

- MainTask
- MainProgram
- Tag de program
- MainRoutine

Program no programado

Condiciones de arranque de la cinta Condicion_Inicio Presencia de caja frente a CFE Lineal, entrada derecha CelfotoL1ED

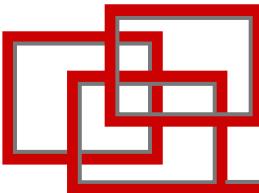
Arrancar cinta con parada temporizada a los 10 minutos sin presencia de caja MarchaL1

Condicion_Inicio CelfotoL1ED MarchaL1

Verificando routine: MainProgram - MainRoutine...
Concluido - 0 error(es), 0 advertencia(s)

RSLogix 5000 - ControlLogix

Copyright © 2005 Rockwell Automation, Inc. All rights reserved.



Ingeniería concurrente con Data Tag Alias

The screenshot shows the RSLogix 5000 software interface. On the left, the navigation tree displays the project structure, including the Controller ControlLogix, Tasks, and Data Types. A red circle highlights the 'MainProgram' tag under the 'Tasks' section. The main workspace on the right shows the 'MainProgram' tag table with three entries: 'MarchaL1', 'Condicion_Inicio', and 'CelfotoL1ED'. The 'Ruta de acceso' (Access path) is set to '<ninguno>'. Below the table, a message box displays the following text:

Verificando routine: MainProgram - MainRoutine...

Concluido - 0 error(es), 0 advertencia(s)

Ingeniería concurrente con Data Tag Alias

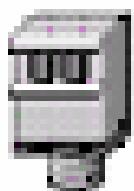
1756-IB32 1756-OB16D



4

8

CelfotoL1ED



MarchaL1

Ingeniería concurrente con Data Tag Alias

The screenshot shows the RSLogix 5000 software interface. On the left, the navigation tree displays the controller structure, with the 'Tag de controller' node circled in red. The main window shows the 'Tag de program - MainProgram' dialog, which lists tags with their aliases and descriptions. Two tags are circled in red: 'MarchaL1' (alias Local:12:0.Data.8(C)) and 'CelfotoL1ED' (alias Local:3:I.Data.4(C)). A yellow callout box points to the alias column with the text: "Con los despliegables se selecciona los tags de los que son imagen (alias) mis tags". Below the dialog, the ladder logic editor shows two contacts: 'Condicion_Inicio' (Local:3:I.Data.4) and 'CelfotoL1ED' (Local:3:I.Data.4). The coil for 'MarchaL1' is labeled 'Arrancar cinta con parada temporizada a los 10 minutos sin presencia de caja'.

La definición de los módulos de E/S crea los perfiles de dispositivos en la base de datos global del procesador

Con los despliegables se selecciona los tags de los que son imagen (alias) mis tags

Nombre de tag	Alias para	Punto de tag	Tipo	Estilo	Descripción
MarchaL1	Local:12:0.Data.8(C)	Local:12:0.Data.8(C)	BOOL	Decimal	Arrancar cinta con parada temporizada a los
Condicion_Inicio			BOOL	Decimal	Condiciones de arranque de la cinta...
CelfotoL1ED	Local:3:I.Data.4(C)	Local:3:I.Data.4(C)	BOOL	Decimal	Presencia de caja frente a CFE Linea1, entr

0

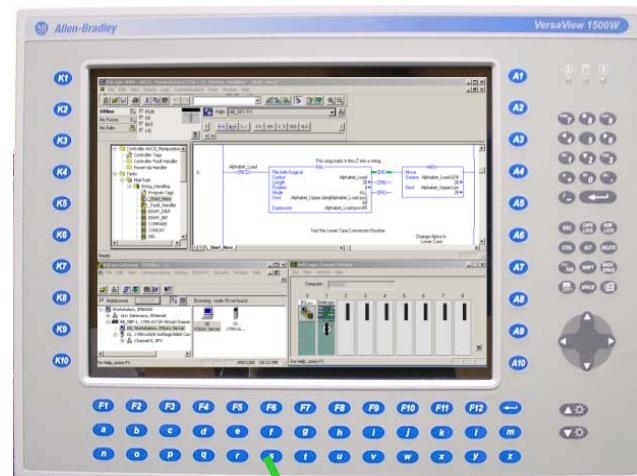
Condiciones de arranque de la cinta
Condicion_Inicio

Presencia de caja frente a CFE Linea1, entrada derecha
CelfotoL1ED
<Local:3:I.Data.4>

Arrancar cinta con parada temporizada a los 10 minutos sin presencia de caja
MarchaL1
<Local:12:0.Data.8>

Copyright © 2005 Rockwell Automation, Inc. All rights reserved.

Ingeniería concurrente con Data Tag Alias



Ordenador Industrial VersaView

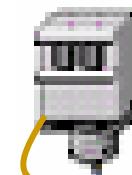
SoftLogix 5800

ES_Rem_TransL1



EtherNet/IP

CelfotoL1ED



MarchaL1

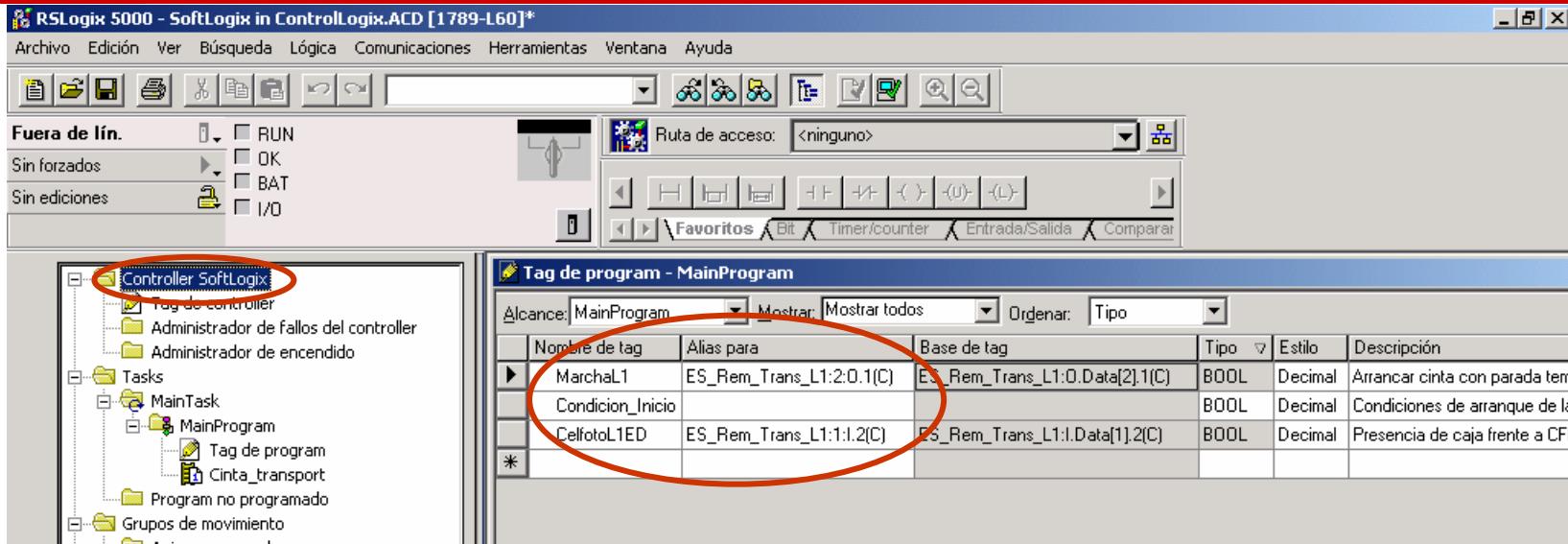
2



1



Ingeniería concurrente con Data Tag Alias



Condiciones de
arranque de la cinta
Condicion_Inicio

Presencia de caja
frente a CFE Linea1,
entrada derecha
CelfotoL1ED
<ES_Rem_Trans_L1:I.Data[1].2>

Arrancar cinta con
parada temporizada a
los 10 minutos sin
presencia de caja

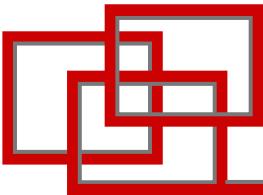
MarchaL1
<ES_Rem_Trans_L1:O.Data[2].1>
(L)

- La capacidad de creación de tags en bases de datos y asignación de Alias de Logix5000 puede reducir el tiempo y el coste de desarrollo de sus proyectos
 - Desarrollo de programas previo al diseño eléctrico completo
 - Re utilización de programas en múltiples proyectos
 - Mejora la documentación de los proyectos



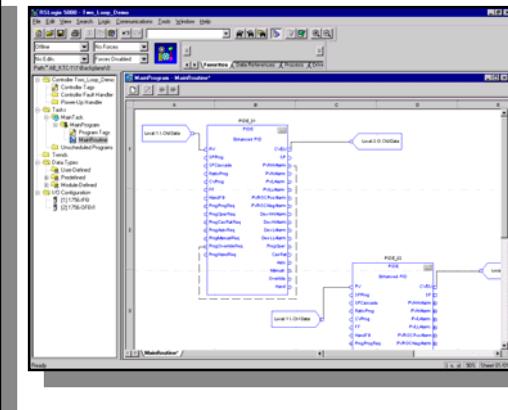
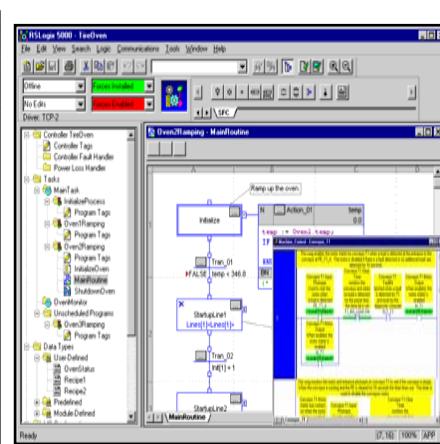
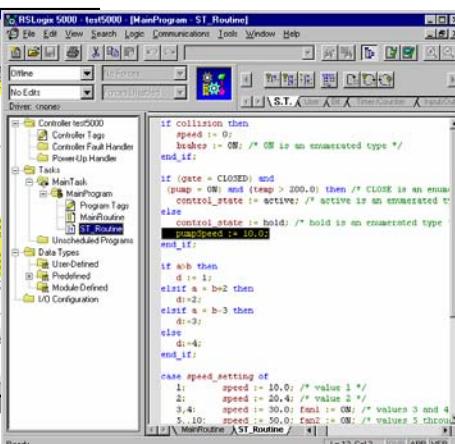
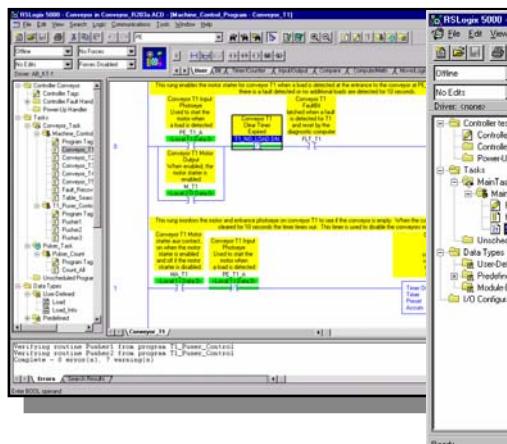
- Reduce tiempo de ingeniería –crea librerías de rutinas reutilizables
- Reduce tiempo de localización de averías – el código puede pegarse en un proyecto sin considerar nombres de tags duplicados ni conflictos de memoria
- Reduce tiempo al mercado
 - Los Programas pueden escribirse simultáneamente sin conflictos.
 - Los Programas pueden escribirse mientras se monta el hardware



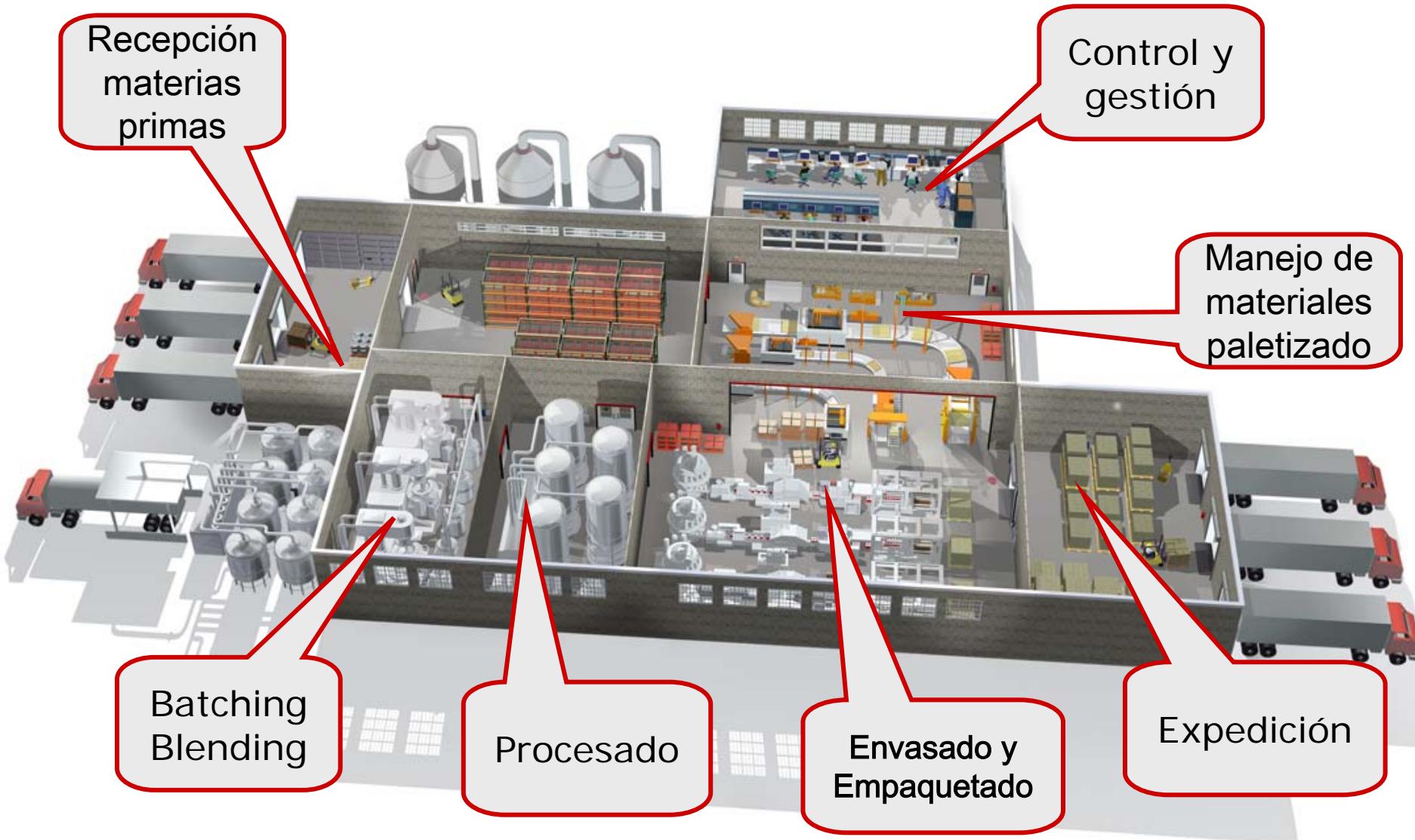


Valor de Logix: Lenguaje apropiado para el proyecto

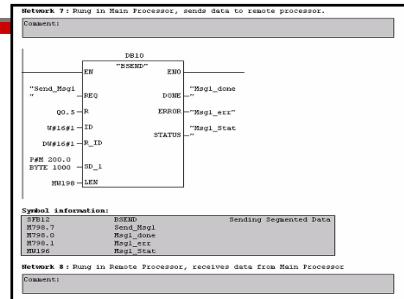
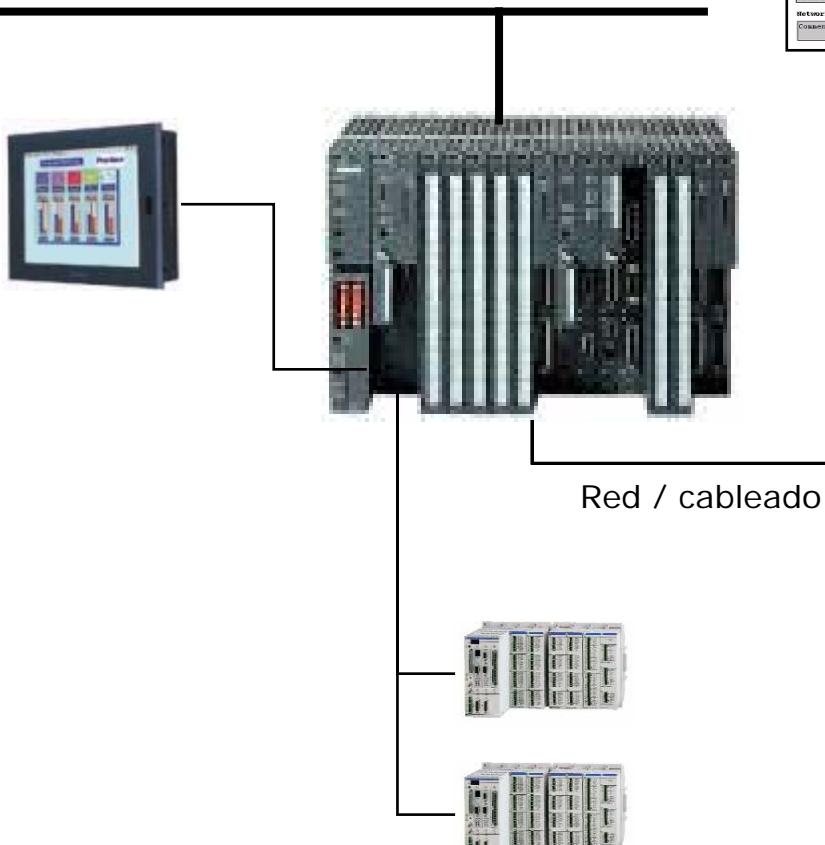
- RSLogix 5000 permite 4 lenguajes de programación diferentes
 - Ladder Diagram
 - Function Block
 - Structured Text
 - Sequential Function Chart (SFC)
 - Los ingenieros pueden programar en el lenguaje que les resulte más cómodo o en el que mejor encaje con la resolución de la aplicación



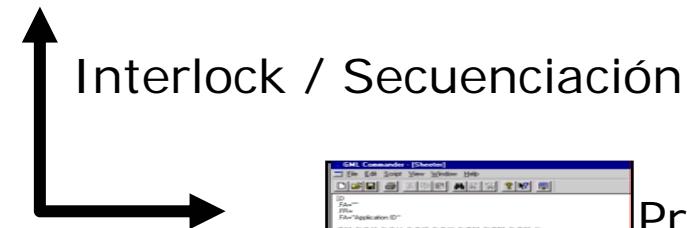
Planta típica de fabricación



Solución tradicional PLC + Motion Controller



Programa
del PLC



```
GML Commander [Elementary]
File Edit Help Application ID: 00000000000000000000000000000000
G01 F1000
G01 F1000
G01 F1000
G01 F1000
```

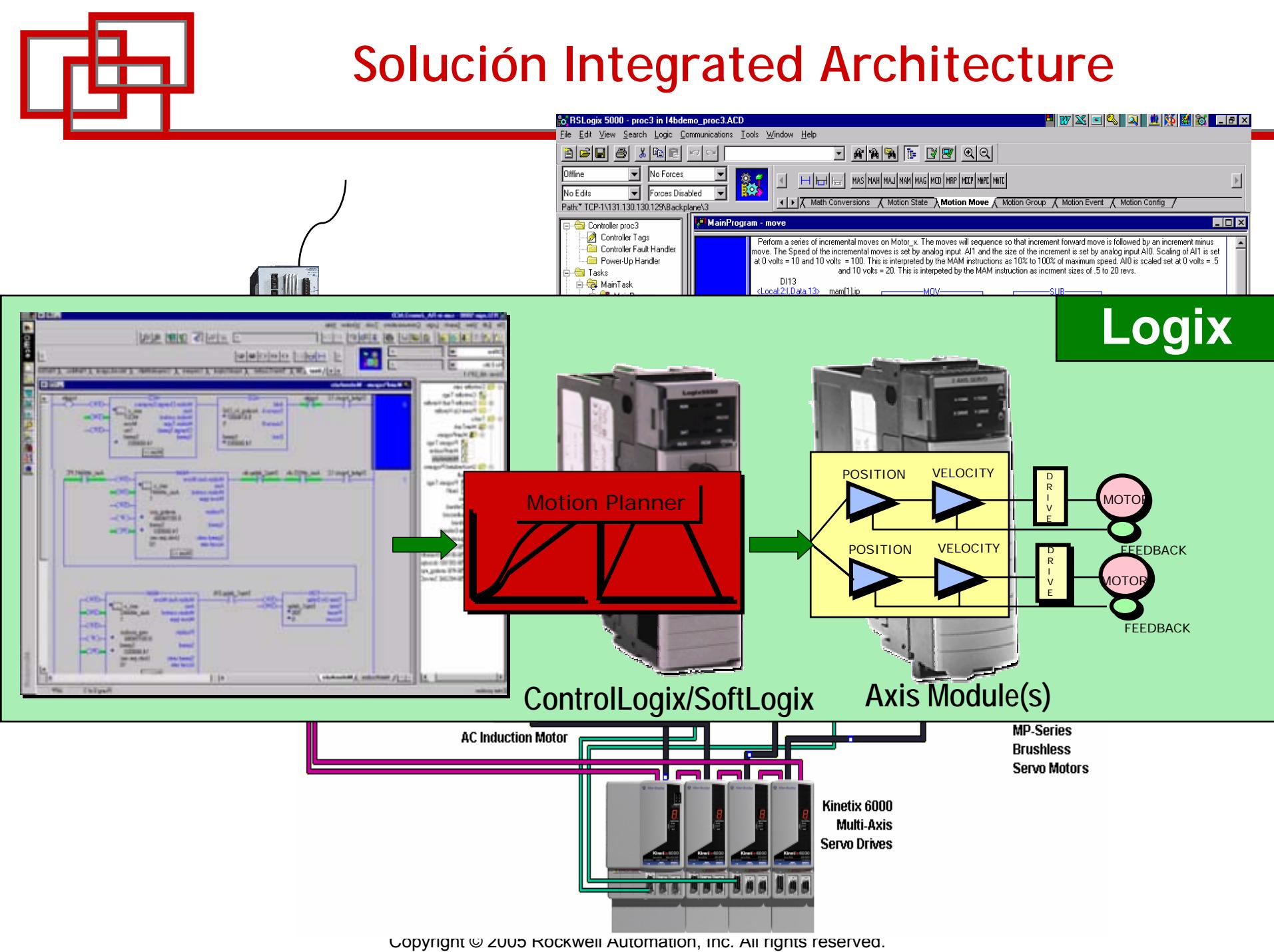
Programa(s)
de servos



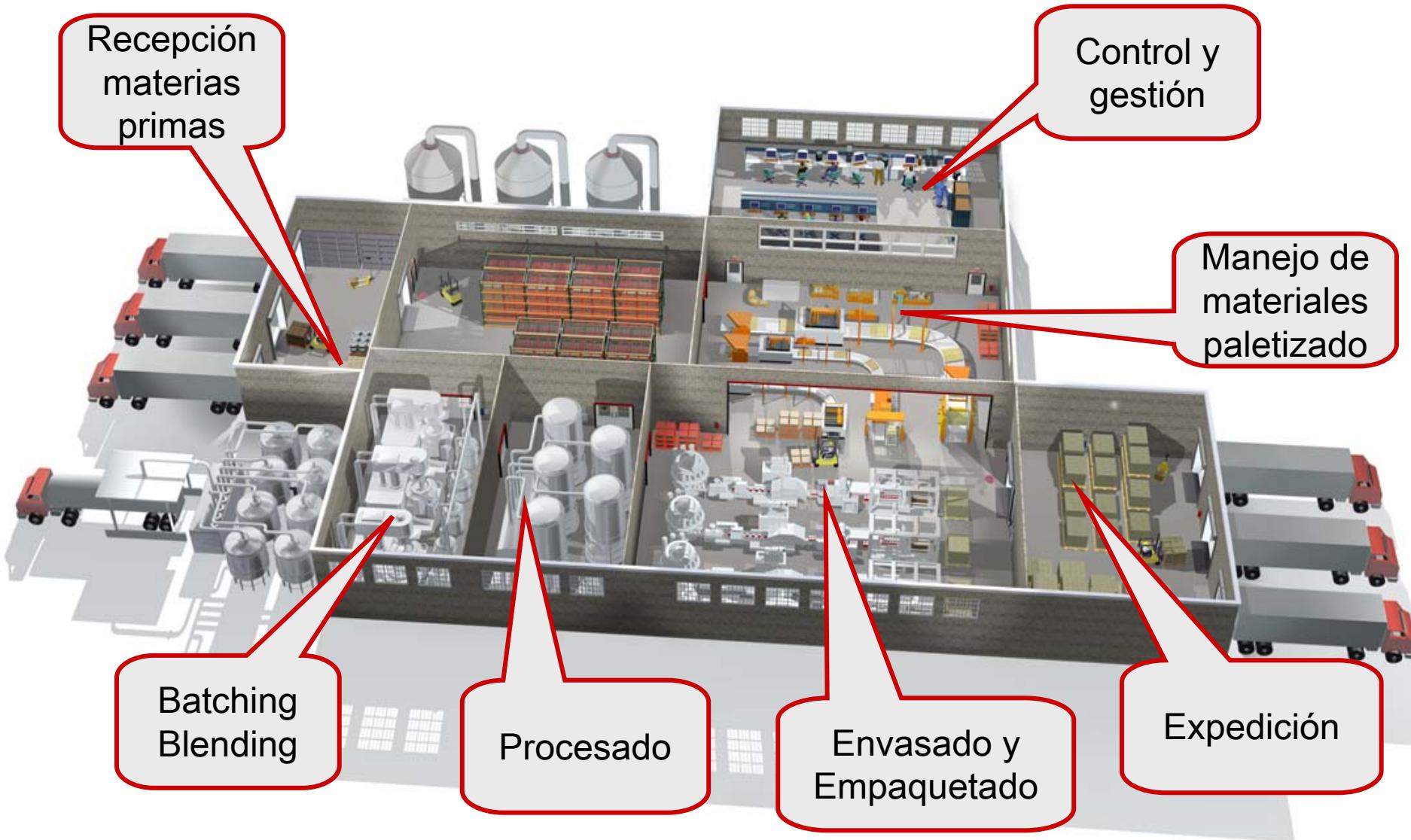
Servo
Tuning

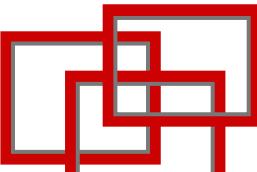


Solución Integrated Architecture



Planta típica de fabricación





Lenguaje e instrucciones para proceso

RSLinx 5000 - MyApplication

File Edit View Search Logic Communications Tools Window Help

Offline No Forces Forces Disabled

Path: AB_PCC-1\1\Backplane\0

MainProgram - MyFBDRoutine

Sheet 1 of 2 Temperature_Zone_1_Control

A B C

1 Local:1:I.Ch0Data

2 AdaptivePGain

3

PIDE Autotune - FlowLoop

Start Abort

Execution State: Complete
Autotune Status: OK

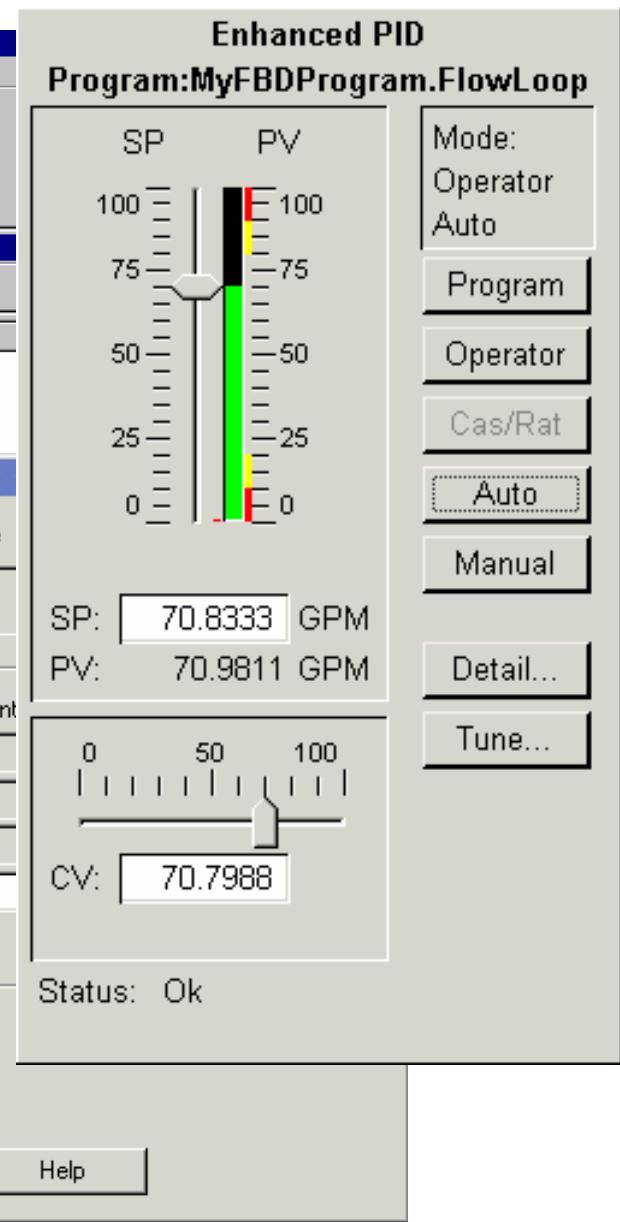
Autotune Gains

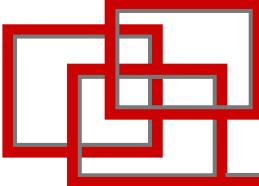
	Proportional	Integral
Slow Response	0.5857508	
Medium Response	1.1715016	
Fast Response	1.7572525	
Current	1.7975489	

Set Gains in PIDE

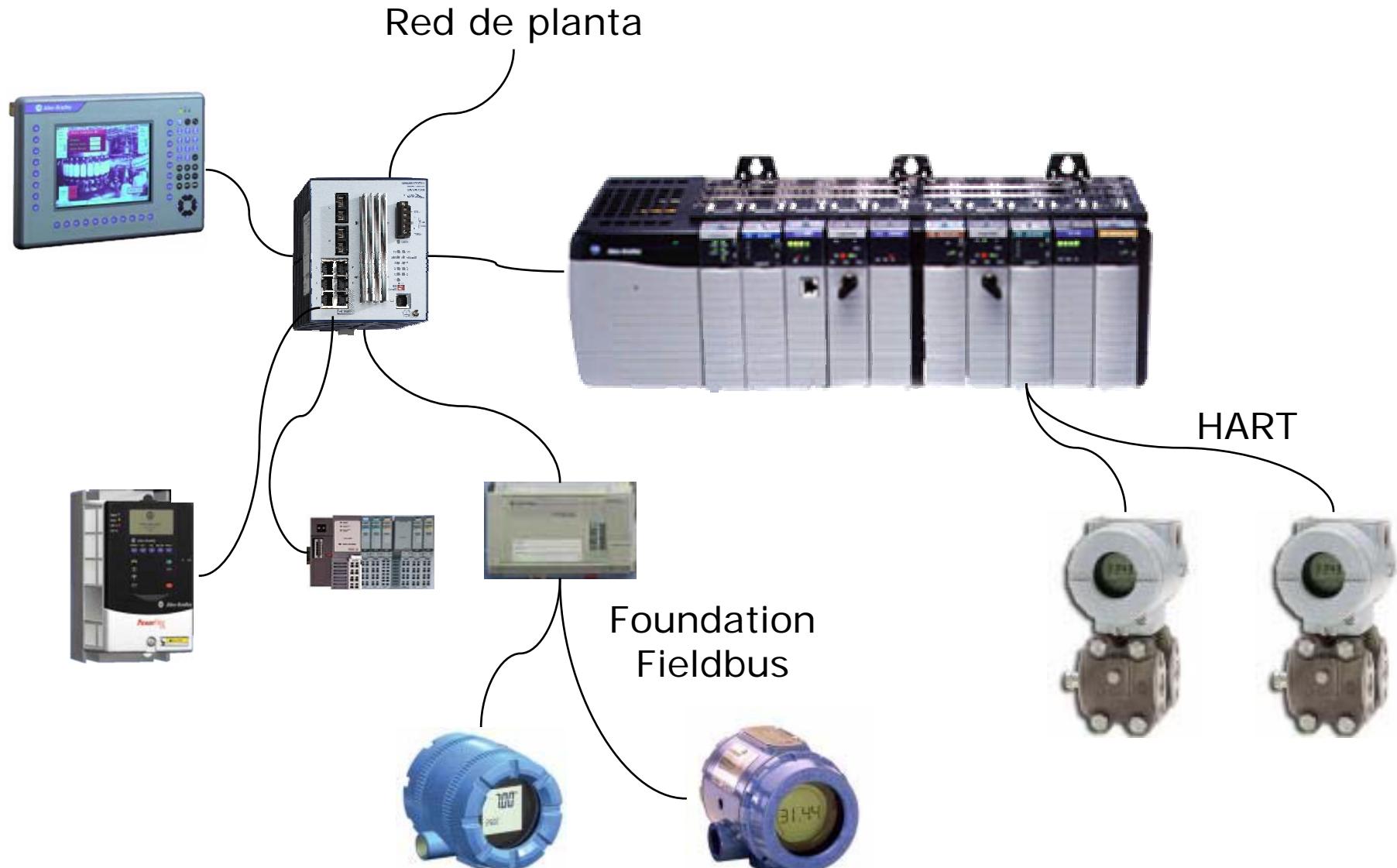
Time Constant: 5.1 sec
Deadline: 2.3 sec
Gain: 1.0505912
Based on Non-integrating model

Close Help

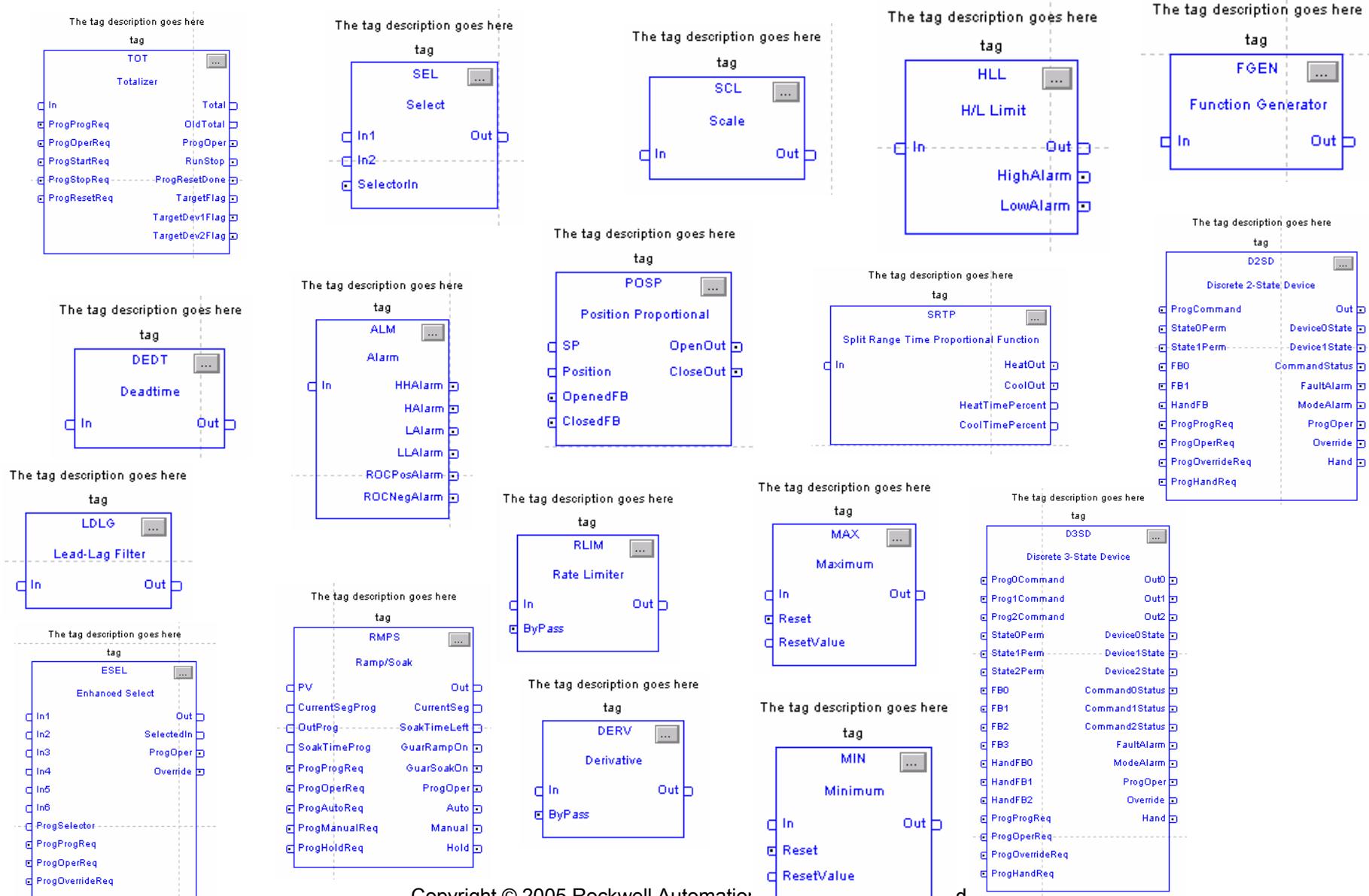




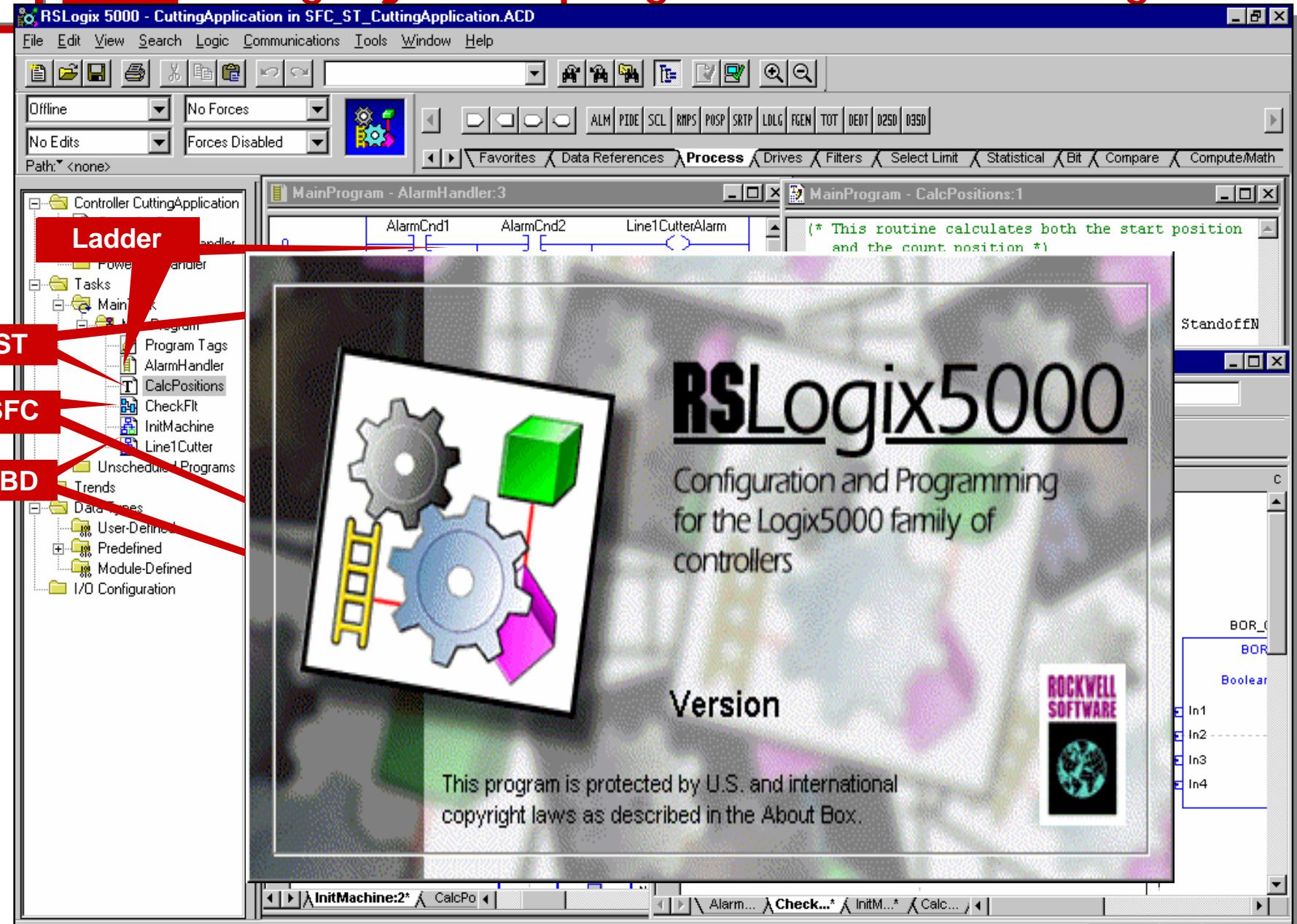
Comunicaciones para proceso



Set de instrucciones para proceso



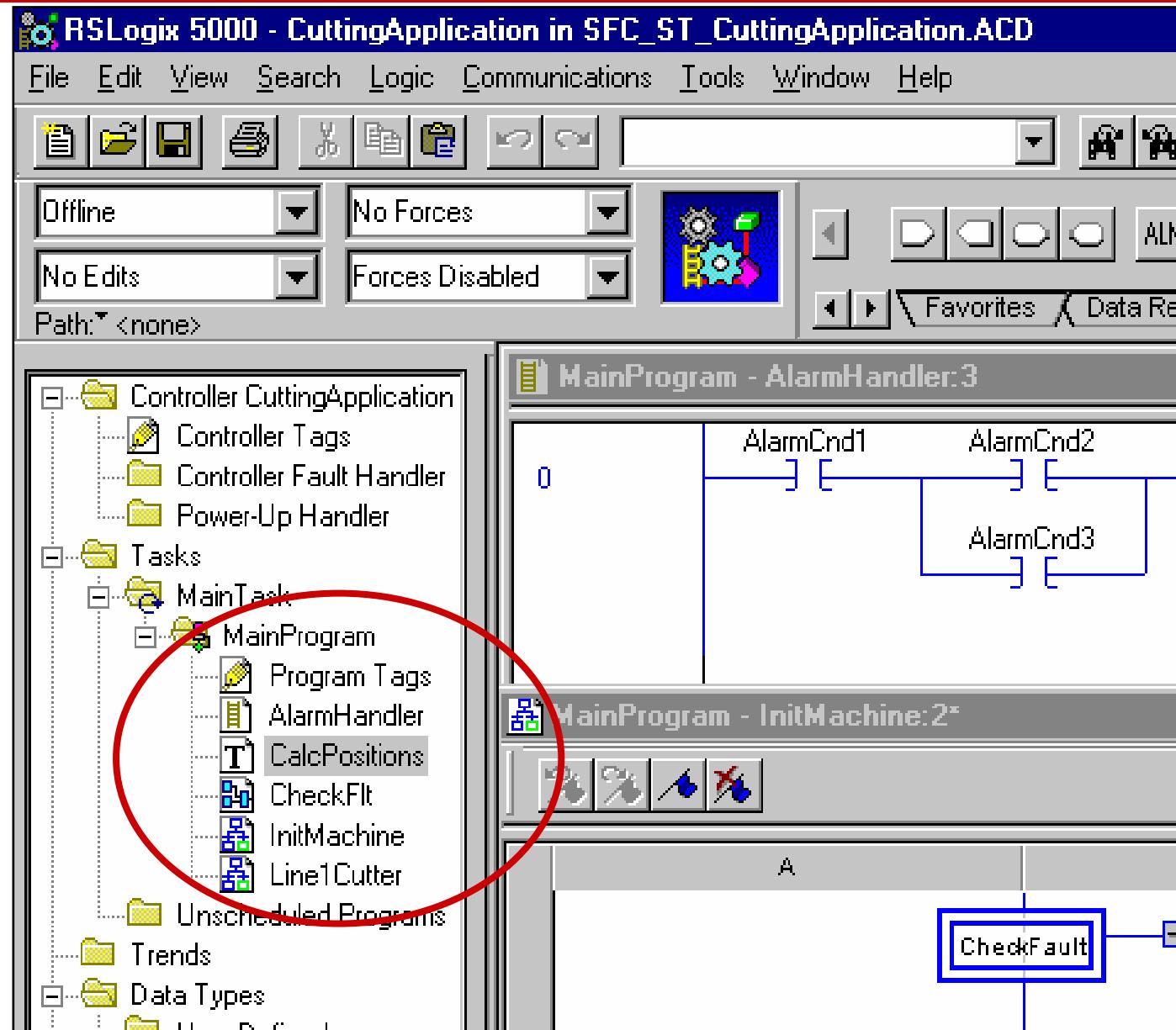
Lenguajes de programación en RSLogix5000

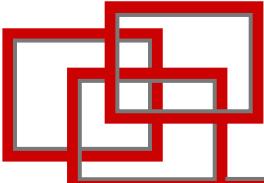


Lenguajes de programación en RSLogix5000

Selección de lenguaje de programación a nivel de Rutina.

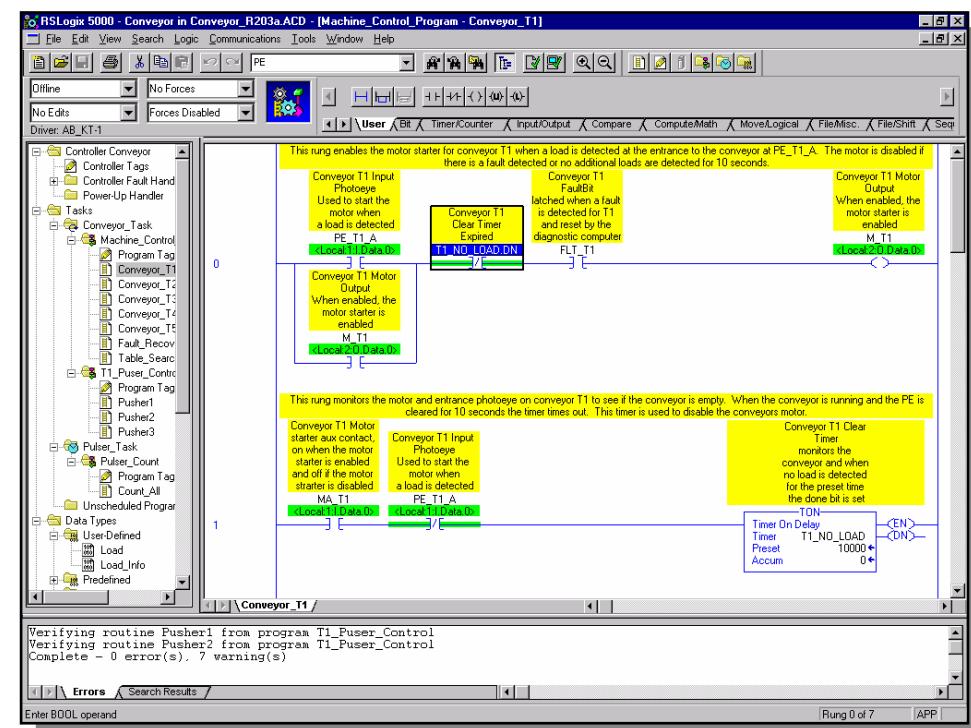
En un programa, cada rutina puede ser programada en el lenguaje idóneo para la función





Ladder Diagram (LD)

- El entorno de edición permite ediciones múltiples simultáneas
 - Las modificaciones son chequeadas cuando el usuario realiza una descarga o selecciona la operación de verificación
 - El usuario puede seleccionar cualquiera de los dos modos de verificación, manual o automático
 - Interfase común con RSLogix 5 / 500 reduciendo el tiempo de adaptación
 - El set de instrucciones LD más intuitivo de la industria!
 - Evitando la necesidad de otro lenguaje
 - Basado en PLC/SLC
 - Motion control blocks
 - Manejo de tablas de datos
 - Diagnóstico
 - Puerto serie y mensajería
 - Manipulación ASCII
 - Edición off-line y on-line para segmentos individuales
 - Permite cambios en RUN
 - Mínimo impacto en el controlador
 - Cumple la norma IEC1131



Ladder Diagram (LD)

RSLogix 5000 - Conveyor in Conveyor_R203a.ACD - [Machine_Control_Program - Conveyor_T1]

File Edit View Search Logic Communications Tools Window Help

PE

Offline No Edits Driver: AB_KT-1

User Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical File/Misc. File/Shift Seq

Controller Conveyor Controller Tags Controller Fault Hand Power-Up Handler Tasks Conveyor_Task Machine_Control Program Tag Conveyor_T1 Conveyor_T2 Conveyor_T3 Conveyor_T4 Conveyor_T5 Fault_Recover Table_Search T1_Puser_Control Program Tag Pusher1 Pusher2 Pusher3 Pulser_Task Pulser_Count Program Tag Count_All Unscheduled Program Data Types User-Defined Load Load_Info Predefined

This rung enables the motor starter for conveyor T1 when a load is detected at the entrance to the conveyor at PE_T1_A. The motor is disabled if there is a fault detected or no additional loads are detected for 10 seconds.

Conveyor T1 Input Photoeye Used to start the motor when a load is detected PE_T1_A <Local[1].Data.0>

Conveyor T1 Clear Timer Expired T1_NO_LOAD.DN 1/E

Conveyor T1 FaultBit latched when a fault is detected for T1 and reset by the diagnostic computer FLT_T1

Conveyor T1 Motor Output When enabled, the motor starter is enabled M_T1 <Local[20].Data.0>

Conveyor T1 Input Photoeye Used to start the motor when a load is detected PE_T1_A <Local[1].Data.0>

Conveyor T1 Motor Output When enabled, the motor starter is enabled M_T1 <Local[20].Data.0>

This rung monitors the motor and entrance photoeye on conveyor T1 to see if the conveyor is empty. When the conveyor is running and the PE is cleared for 10 seconds the timer times out. This timer is used to disable the conveyors motor.

Conveyor T1 Motor starter aux contact, on when the motor starter is enabled and off if the motor starter is disabled MA_T1 <Local[1].Data.0>

Conveyor T1 Input Photoeye Used to start the motor when a load is detected PE_T1_A <Local[1].Data.0>

Conveyor T1 Clear Timer monitors the conveyor and when no load is detected for the preset time the done bit is set TON Timer On Delay Timer T1_NO_LOAD Preset 10000 Accum 0

Verifying routine Pusher1 from program T1_Puser_Control
Verifying routine Pusher2 from program T1_Puser_Control
Complete - 0 error(s), 7 warning(s)

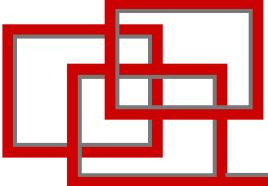
Errors Search Results

Enter BOOL operand

Rung 0 of 7 APP

```
graph TD
    MA_T1[MA_T1] --- C1(( ))
    PE_T1_A[PE_T1_A] --- C1
    C1 --- T1_NO_LOAD_DN[T1_NO_LOAD.DN]
    T1_NO_LOAD_DN --- FLT_T1[FLT_T1]
    FLT_T1 --- T1_NO_LOAD_DN

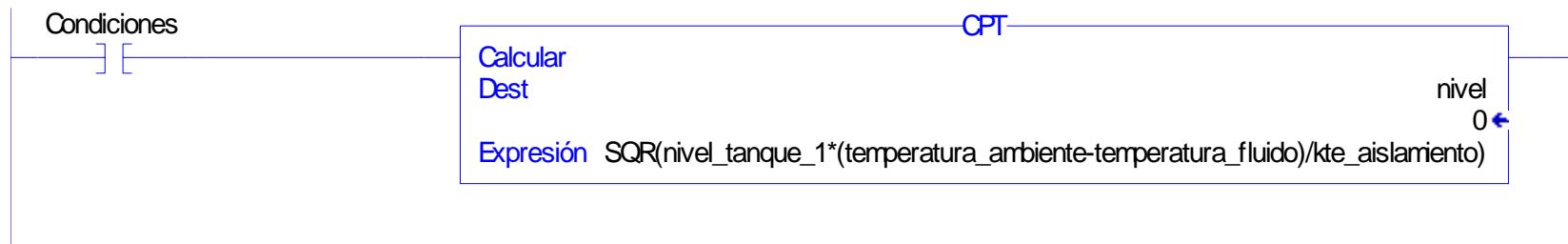
    MA_T1 --- C2(( ))
    PE_T1_A --- C2
    C2 --- T1_NO_LOAD[T1_NO_LOAD]
    T1_NO_LOAD --- TON[TON Timer On Delay]
    TON --- Preset[Preset 10000]
    TON --- Accum[Accum 0]
```



Valor de Logix: La potencia y simplicidad de cálculos

- Disponemos de instrucciones tremendamente potentes

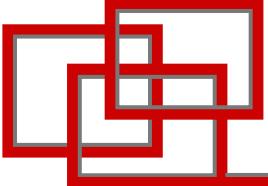
3



Expresion: SQR(**nivel_tanque_1***(**temperatura_ambiente** – **temperatura_fluido**)/**Kte_aislamiento**)
INT (Ent. 16 bits) REAL (Float 32 bits) REAL (Float 32 bits) REAL (Float 32 bits)

No hace falta convertir (unificar) los operandos de las instrucciones de cálculo. Los procesadores saben multiplicar enteros por decimales

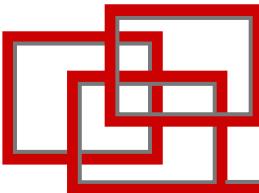
Nombre de tag	Valor	Máscara de forza	Estilo	Tipo	Descripción
temperatura_fluido	47.2		Float	REAL	
temperatura_ambiente	21.7		Float	REAL	
kte_aislamiento	0.245		Float	REAL	
+nivel_tanque_1	1234		Decimal	INT	
+nivel	0		Decimal	DINT	



Aplicaciones para Ladder Diagram

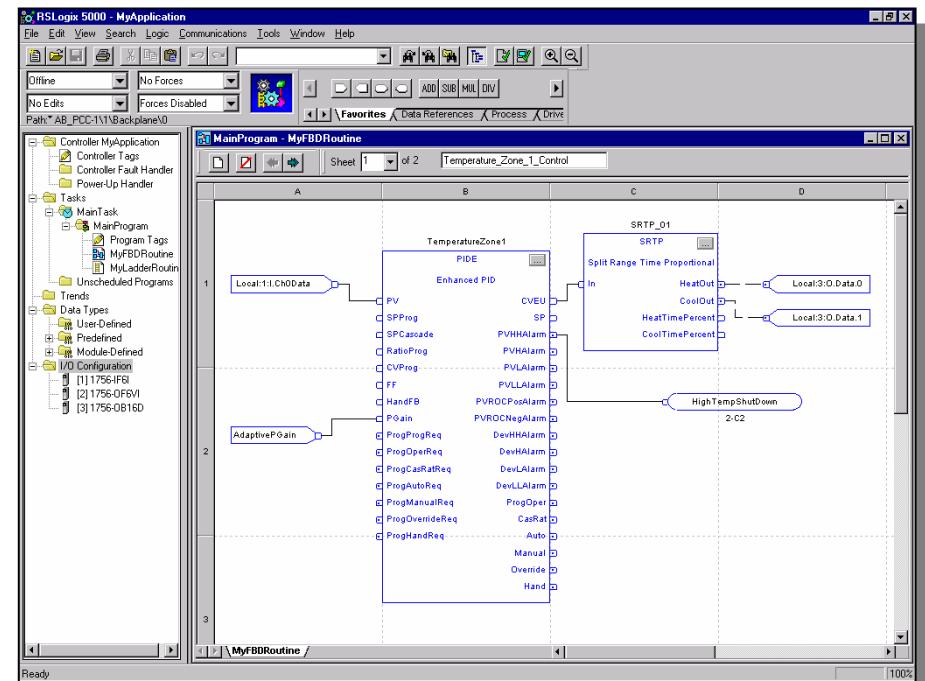
- Aplicaciones de control que requieren decisiones complejas y procesamiento paralelo, enclavamiento de máquina, operaciones continuas
 - Manipulación de materiales - Manejo de paquetes, ordenación, paletizado...
 - Automóvil – prensa, ensamblado, pintura...
 - Entretenimiento – telesquí, parques de atracciones...
 - Mensajería y comunicación serie
- Aplicaciones de control de movimiento general o robótica con servos analógicos o sercos
 - Productos de consumo – Packaging...
 - Líneas Transfer- Motores, transmisión...
 - Equipos de movimiento – puente-grúa, carros de transferencia...
 - Entretenimiento – teatro, animaciones, simuladores...



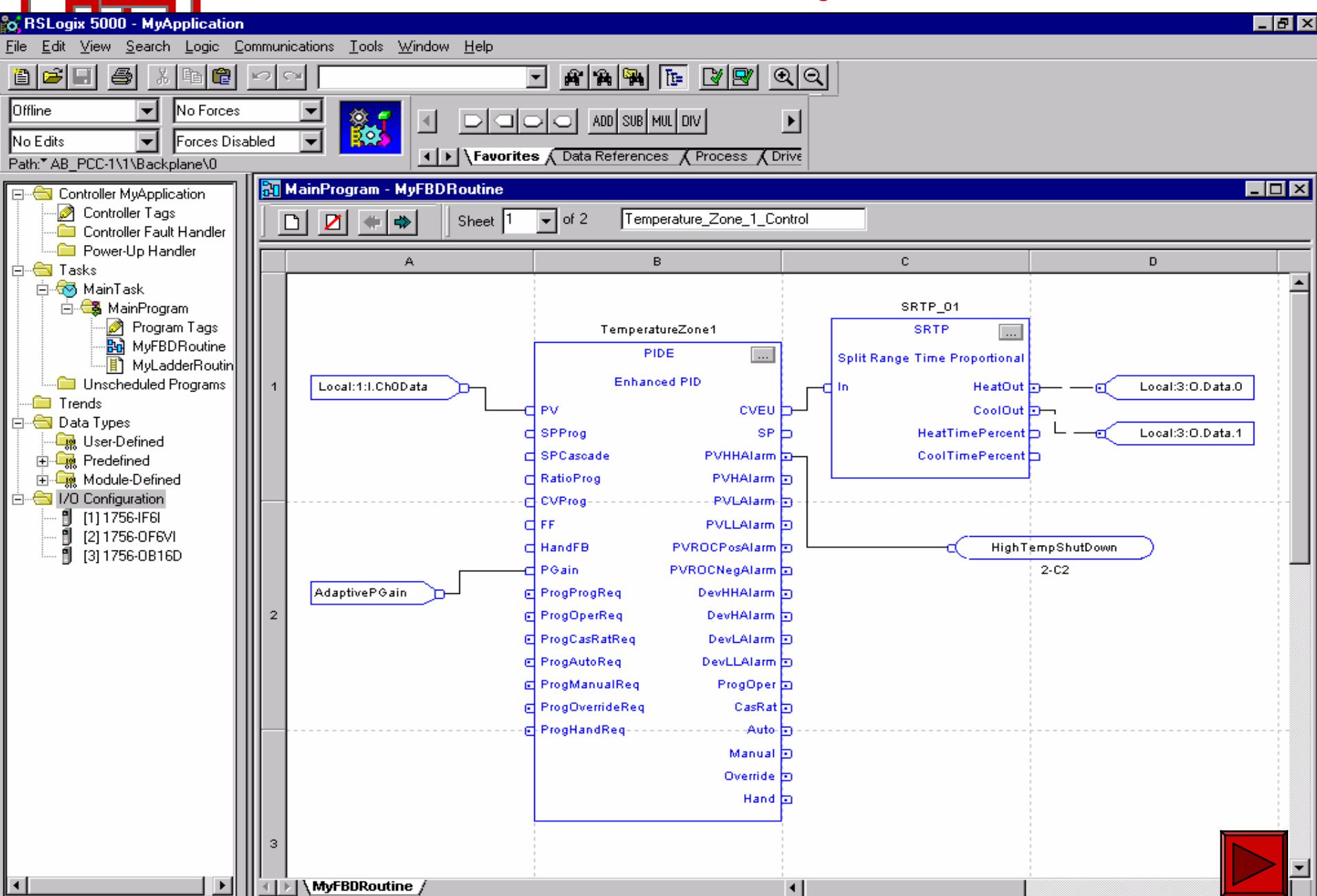


Function Block Diagram (FBD)

- **Medio de programación gráfico**
 - Situar instrucciones y dibujo de conexiones para paso de parámetros
 - Posicionamiento y organización de los bloques basados en favorecer la lectura de la aplicación
 - Cajas de texto flotante para la documentación de la aplicación
- **El orden de ejecución determinado automáticamente por las interconexiones**
 - Elimina el proceso de creación de programa
 - Simplifica las modificaciones
- **Edición Off-line y estado on-line**
 - Cargas y descargas completas
 - Edición On-line para rutinas FBD
 - Las funciones FBD son nativas para Logix5000
 - Bloques comunes con LD
 - 42 funciones nuevas de proceso
- **Cumple la norma IEC1131-3**

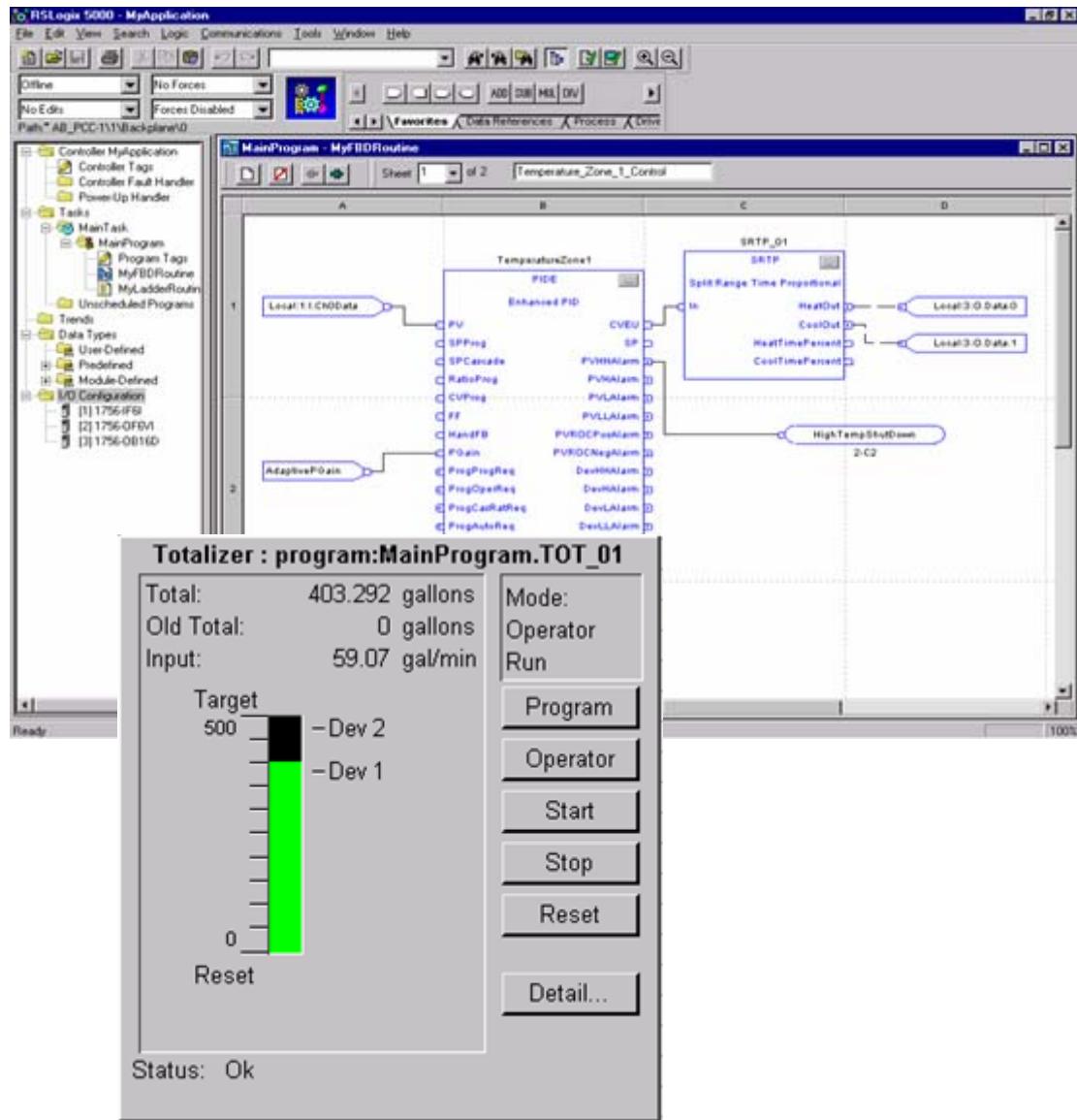


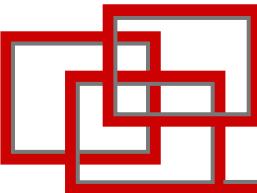
Function Block Diagram (FBD)



Editor de Function Block de RSLogix 5000

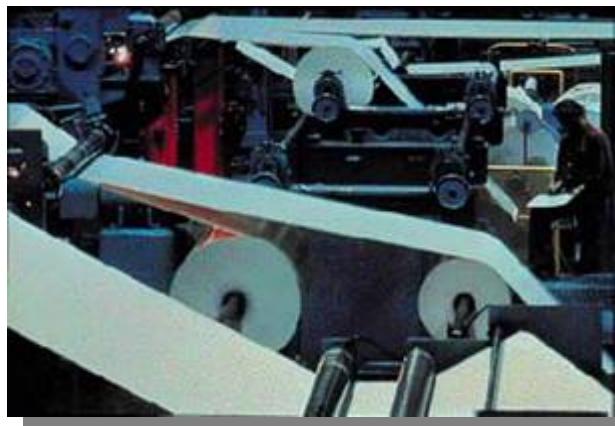
- Lenguaje y editor de Function Block totalmente integrados que rivalizan con los configuradores de los DCS's
 - Facil configuración arrastrar y soltar
 - Faceplates ActiveX permiten diseñar HMI
- Las rutinas Function Block se almacenan en el controlador
 - Pueden ser descargadas sin el código fuente
- Los tags de la base de datos almacenados en el controlador ofrecen más flexibilidad y funcionalidad comparable a los DCS

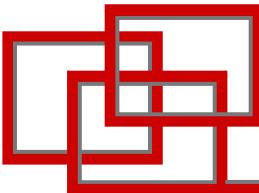




Aplicaciones para Function Block Diagram

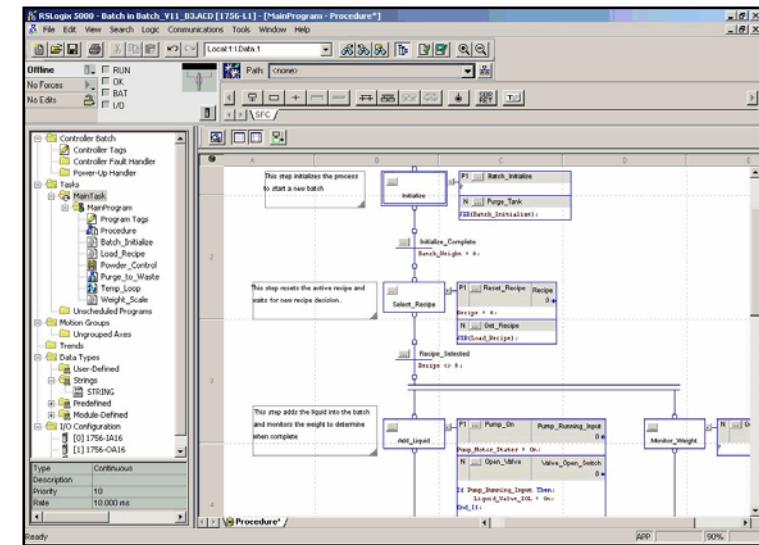
- Ideal para algoritmos de control con lazos analógicos
 - Representación gráfica de los bucles de control
 - Lenguaje altamente visual y fácil de entender
 - Parecido a los diagramas de circuitos
 - Utilizado comúnmente para sistemas de control distribuido (DCS)
- Lazos de control continuo o Batch
- Lazos de control de motores
 - Regulación de velocidad, posición y tensión ...
 - Control de bobinado, tensión y basculación...
 - Proporciona una base para aplicaciones que previamente han sido usadas en Reliance Automax



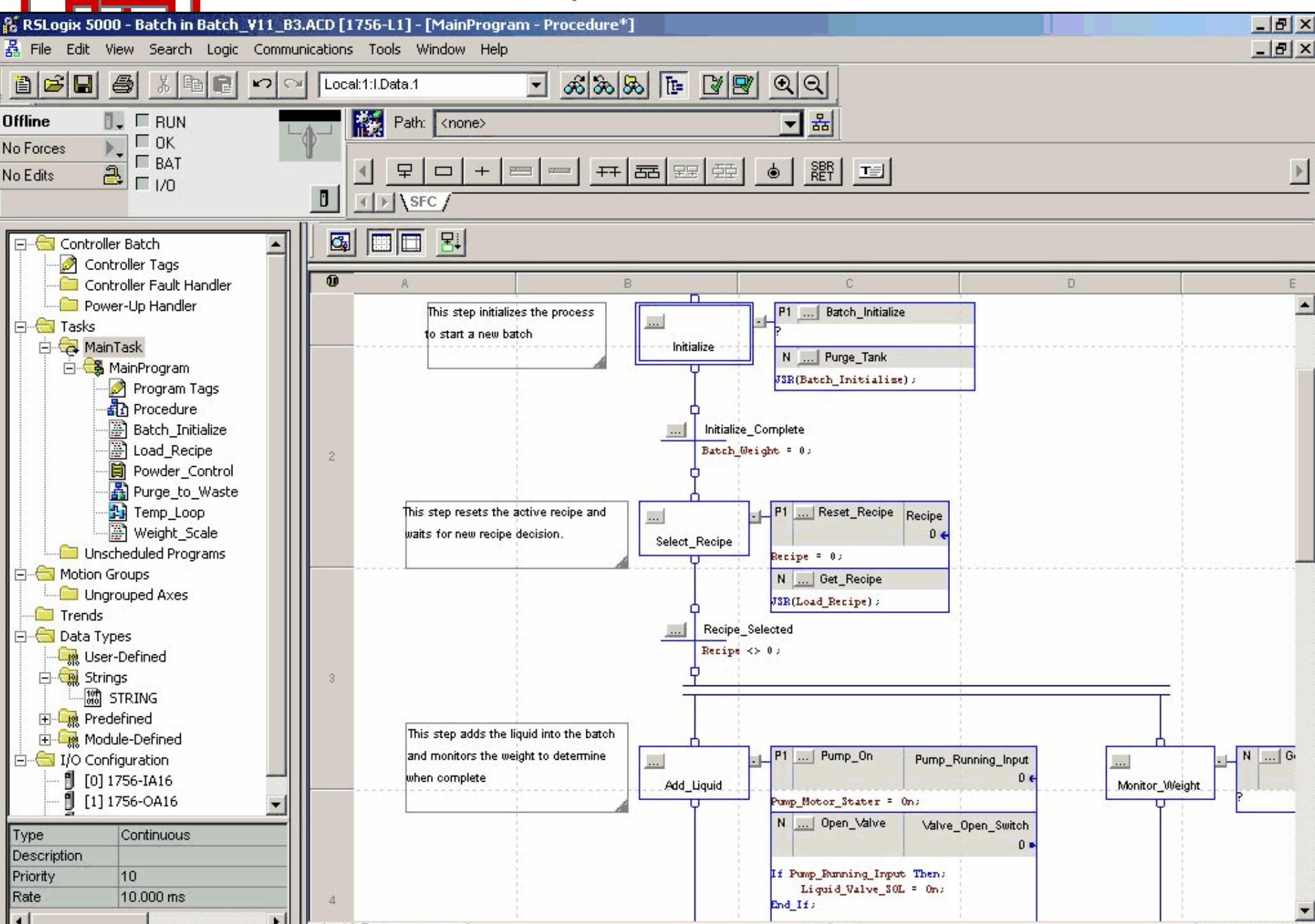


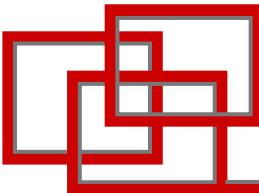
Sequential Function Chart (SFC)

- Medio de programación gráfico
 - Situar steps, transiciones, ramificaciones y dibujar las conexiones para determinar el flujo de ejecución
 - Posicionamiento y organización de los bloques basandonos en favorecer la lectura de la aplicación
 - Cajas de texto flotante para la documentación de la aplicación
- Integración directa del texto en las transiciones y acciones
 - Mejor lectura y mantenimiento
 - Elimina rutinas extrañas
 - Uso de ST para llamar a rutinas FBD, LD, SFC o ST cuando lo necesite
- Edición Off-line con monitorización on-line
 - Visualiza con auto-desplazamiento de los steps activos simplificando mantenimiento
 - Trabaja con el motor Logix5000 con soporte upload/download para rutinas SFC desde el procesador
 - Edición On-line
- Cumple la norma IEC1131-3



Sequential Function Chart (SFC)

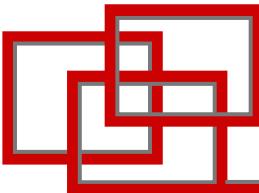




Aplicaciones para SFC

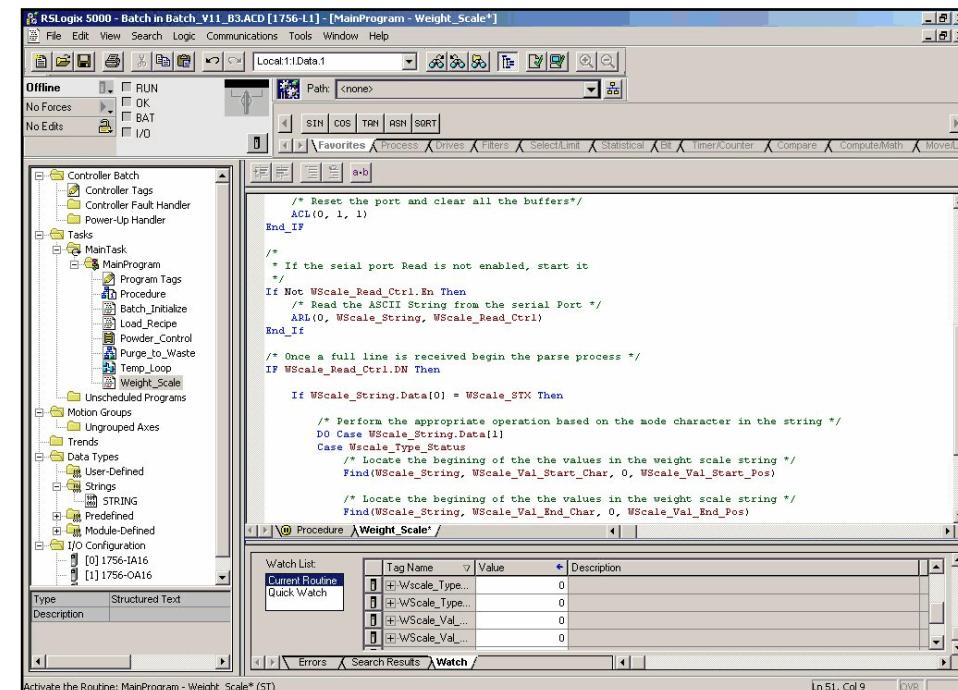
- Secuenciado de estados de máquina
 - Gestión en alto nivel de la ejecución de programas y rutinas
 - Abordar el desarrollo de secuenciadores muy flexibles
 - Lenguaje altamente visual y fácil de entender
 - Ideal para máquinas con operaciones repetitivas
- Aplicaciones con procesos de ejecución por lotes
- Proporciona una base para aplicaciones con motion o robotica
 - La integración de comandos Structured Text Motion proporcionando un entorno de desarrollo racionalizado





RSLogix5000 ST

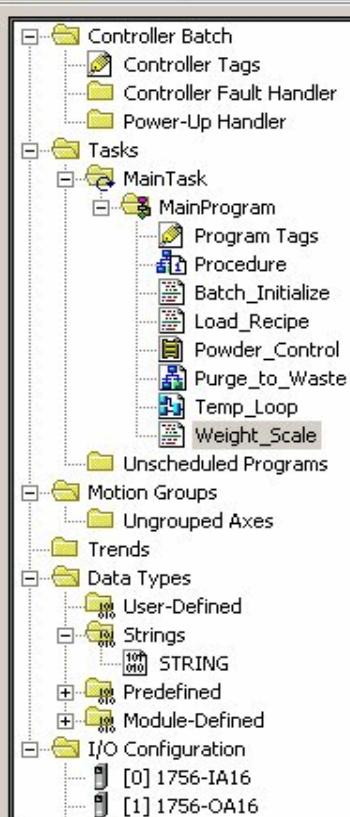
- Editor completamente funcional
 - Indica mediante subrayado en rojo los tag no definidos o errores de programa
 - Sintaxis coloreada para comandos, tags y comentarios facilitando la lectura
 - El formato de las instrucciones está asistido simplificando el desarrollo del código
 - Cut/Copy/Paste desde otra herramienta de edición de texto
- Programación en lenguaje de alto nivel similar a Basic, Fortran, Pascal o “C”
 - Construcciones If/Then, Case, Do/While, Do/Until, y For/Next
 - Soporta instrucciones LD y FBD, incluyendo Motion y Process
 - Llamada a rutinas en FBD, LD, SFC o ST cuando se necesite
- Desarrollo Off-Line y On-line
 - Trabaja con el motor Logix5000 con soporte upload/download para rutinas SFC desde el procesador
 - Lista automática de tags, y visualización de los valores reales
 - Edición On-line
- Cumple la norma IEC1131-3



RSLogix 5000 ST

RSLogix 5000 - Batch in Batch_V11_B3.ACD [1756-L1] - [MainProgram - Weight_Scale*]

File Edit View Search Logic Communications Tools Window Help



```
/* Reset the port and clear all the buffers*/
ACL(0, 1, 1)
End_IF

/*
 * If the serial port Read is not enabled, start it
 */
If Not WScale_Read_Ctrl.En Then
    /* Read the ASCII String from the serial Port */
    ARL(0, WScale_String, WScale_Read_Ctrl)
End_If

/* Once a full line is received begin the parse process */
IF WScale_Read_Ctrl.DN Then

    If WScale_String.Data[0] = WScale_STX Then

        /* Perform the appropriate operation based on the mode character in the string */
        DO Case WScale_String.Data[1]
        Case Wscale_Type_Status
            /* Locate the beginning of the values in the weight scale string */
            Find(WScale_String, WScale_Val_Start_Char, 0, WScale_Val_Start_Pos)

            /* Locate the beginning of the values in the weight scale string */
            Find(WScale_String, WScale_Val_End_Char, 0, WScale_Val_End_Pos)
```

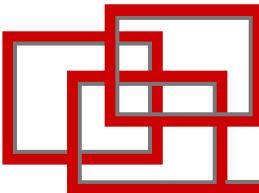
Procedure \Weight_Scale* /

Watch List:

Current Routine
Quick Watch

Tag Name	Value	Description
+Wscale_Type...	0	
+WScale_Type...	0	
+WScale_Val...	0	
+WScale_Val...	0	

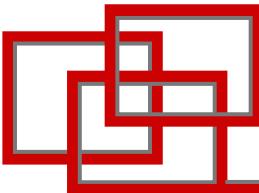
Errors Search Results Watch



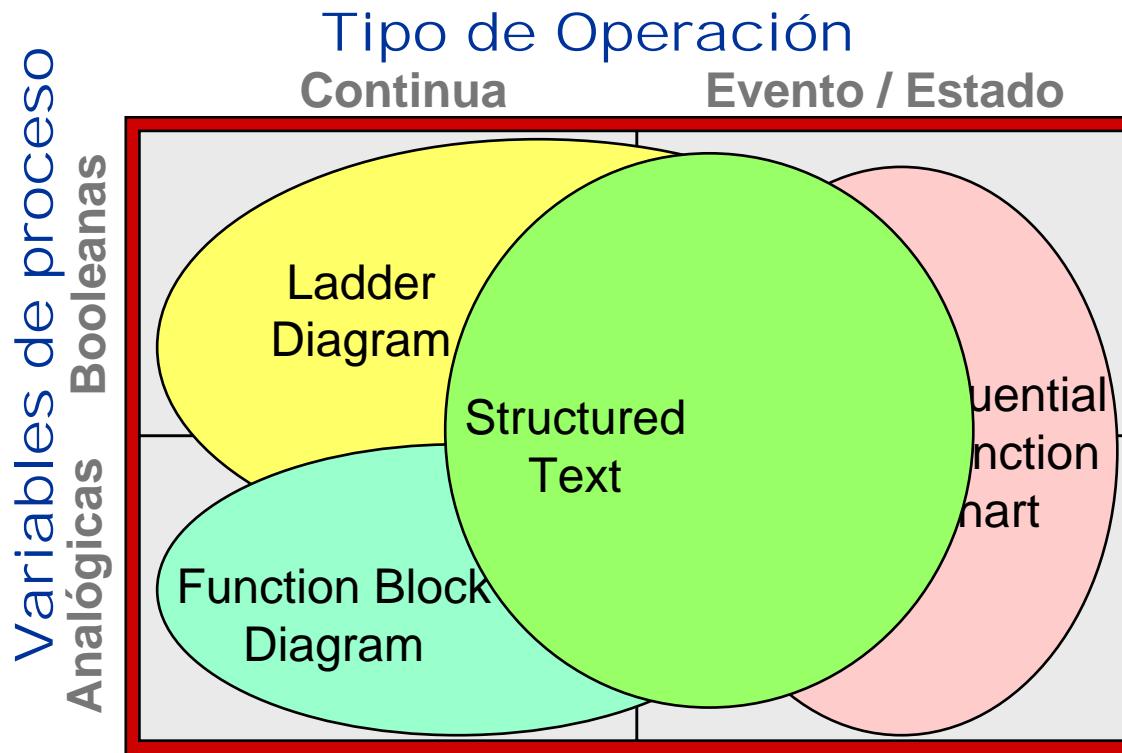
Aplicaciones para Structured Text

- Fácil representación de complejos cálculos matemáticos
- Desarrollo de procesos con manejo de matrices de datos
- Creación de procesos con protocolos de cadenas ASCII
- Integración de funciones motion control
- Inclusión de funciones Process / Drive control permitiendo algoritmos especiales
- Proporciona una base para aplicaciones que utilizan el lenguaje básico de Reliance Automax
- Fácil conversión del texto basado en código Basic, “C”, Pascal y Fortran para funcionar en un controlador Logix5000





Seleccionar el lenguaje adecuado

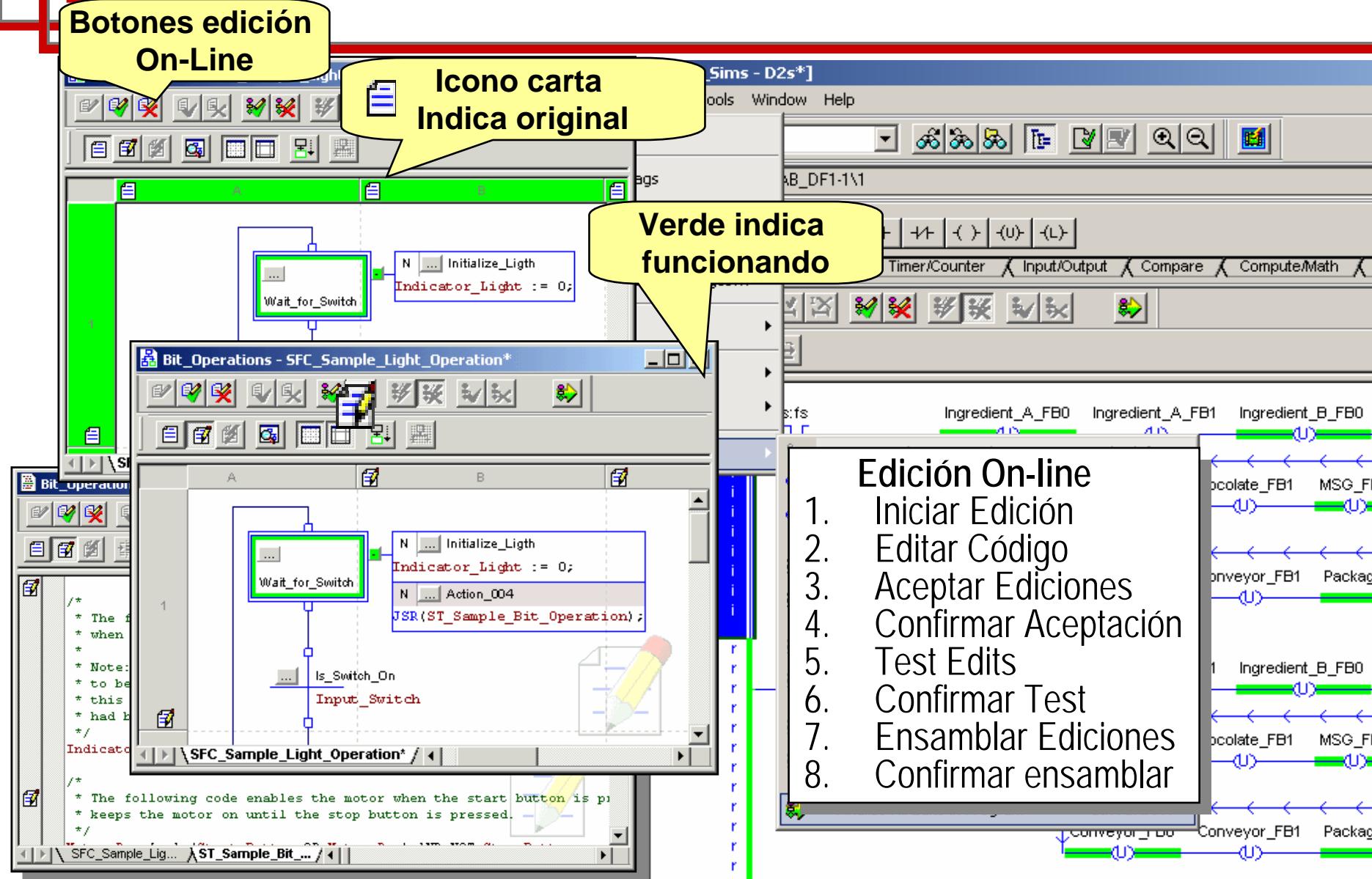


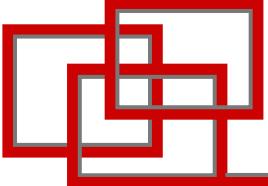
- Hay muchos factores a considerar cuando seleccionamos un lenguaje de programación:
 - Examinar toda las partes de la aplicación y dividirla en pequeñas secciones
 - Seleccionar el lenguaje apropiado a las necesidades de cada sección
 - Forzar la realización de la aplicación en un lenguaje no apropiado nos puede dar como resultado un código más complejo y mas dificultades en el soporte
 - Cuando desarrollamos en el lenguaje adecuado mejoraremos el desarrollo del proceso y favorecemos el mantenimiento de la aplicación resultante

- Reduce el tiempo de desarrollo – “Ahorro del 10 al 20% del tiempo de desarrollo y puesta en marcha.”
- Reduce el coste de formación y mantenimiento – los 4 lenguajes en un paquete de software, que pueden mezclarse en un proyecto.
- Reduce el tiempo de desarrollo y documentación – usar el lenguaje SFC para documentar el proceso durante la fase de diseño y utilizar estos diagramas funcionales directamente, en lugar de desarrollar otros.



Valor de Logix: Edición On-Line múltiple

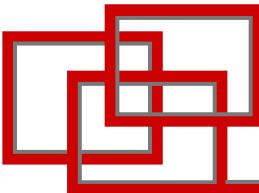




Beneficios de la edición OnLine

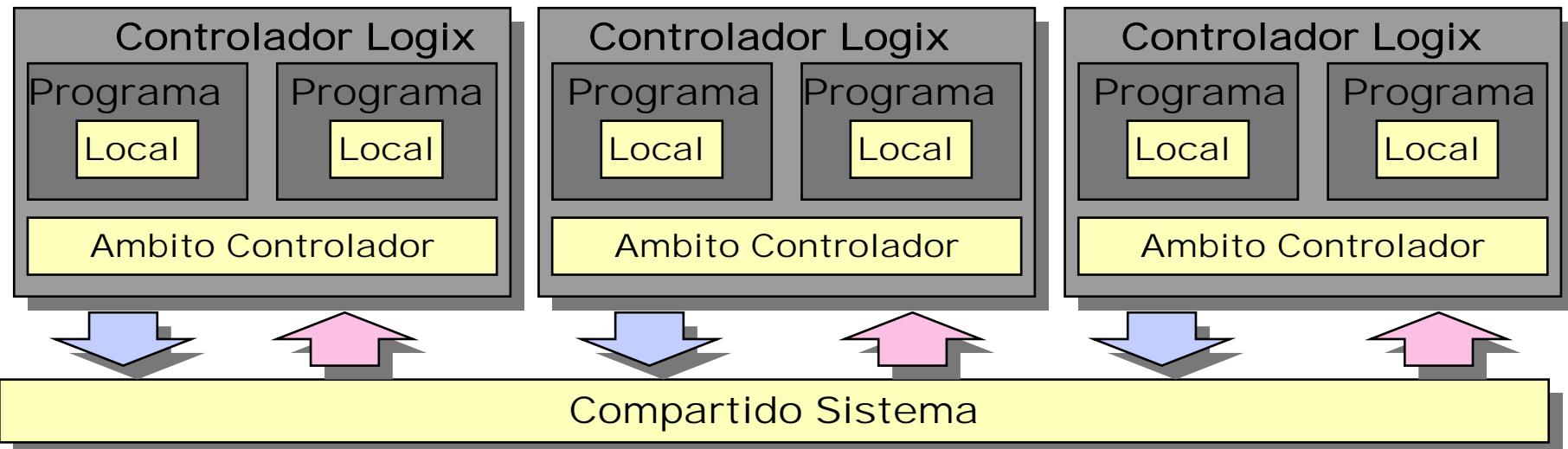
- Se pueden realizar modificaciones sobre la lógica existente y
- Se puede añadir código nuevo al programa, mientras la instalación está funcionando.
- En el caso de modificaciones, permite verificar el programa alternativo, sin perder el original.
- La vuelta a la situación anterior (más segura) se realiza con un click.
- Se puede editar OnLine en todos los lenguajes.
- Varios ingenieros pueden editar programas simultáneamente, incluso pueden editar la misma rutina.
- Mucho más efectivo y seguro que la descarga parcial de programas.

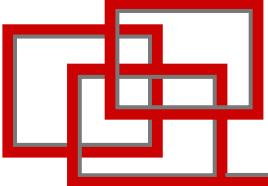




Datos Compartidos por el Sistema

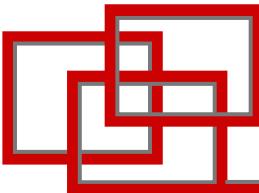
- Con el uso del modelo Productor/Consumidor múltiples controladores bien en el chasis local o conectados vía ControlNet, pueden compartir la información
- La información de ambito el procesador puede ser programa periódicamente entre procesadores como E/S para generar la información en el backplane
 - Periódicamente desde un 1msec
 - Por cambio de estado
- El usuario puede seleccionar la información disponible a consumir





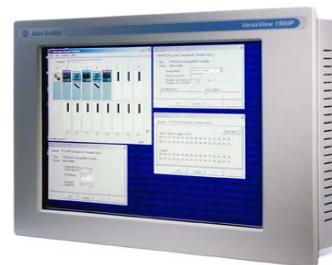
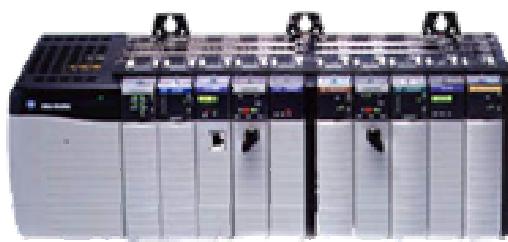
Integración con las Comunicaciones

- El modelo de memoria de Logix basado en Tags está estrechamente integrado con los servicios de comunicaciones NetLinx
 - Más fácil integración de la información en los controladores
 - Con dispositivos de planta
 - Con otros controladores en la planta
 - Con sistemas HMI (Human Machine Interface)
 - Con sistemas ERP (Enterprise Resource Planning)
 - Costes de Integración más reducidos



¿ Que es Logix?

Control multi-disciplina, Superior...

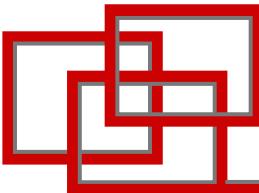


... en una variedad de plataformas...



...con un entorno de programación único...

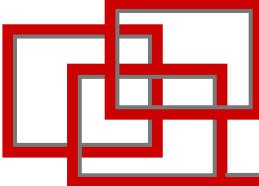
¡ Esto es Logix !



Las plataformas de la familia de controladores Logix

- ControlLogix para aplicaciones de altas prestaciones
- FlexLogix para distribuir el control
- CompactLogix para control de máquinas/estaciones
- SoftLogix para soluciones basadas en PC
- DriveLogix para control distribuído en Drives
- Emulador



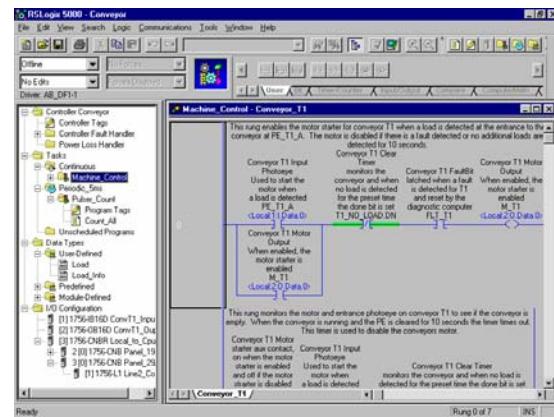


ControlLogix - Del PLC al controlador multidisciplina

- Diseñado para alto rendimiento y escalabilidad
- Multiprocesador, Multitarea, Multi-red, Multi-lenguaje
- Amplio rango de módulos de E/S analógicos, digitales y especiales
- Máxima conectividad desde planta a sistemas de información
- Avanzada integración de control secuencial, de proceso, de drives y de motion

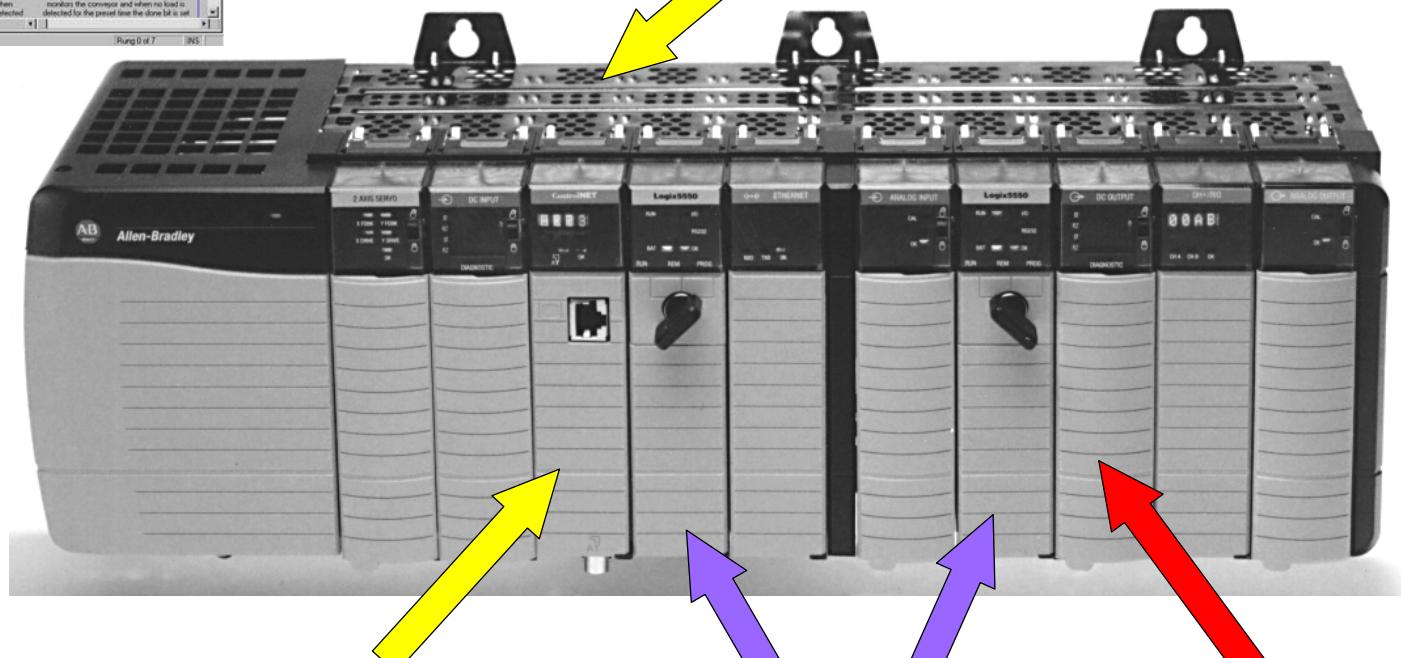


Componentes del Sistema



RSLogix5000

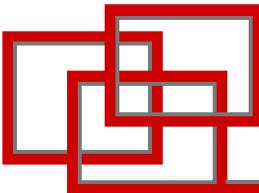
ControlBus



Comunicaciones

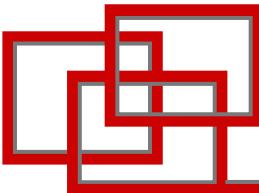
Logix550

E/S 1756



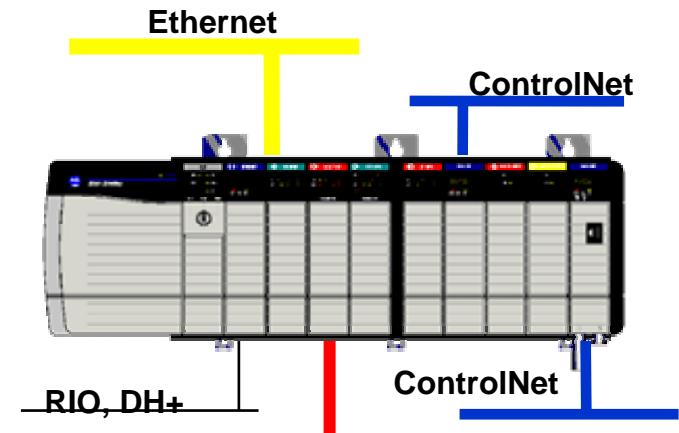
ControlBus

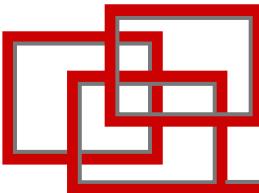
- Backplane Productor / Consumidor
 - Red paralela de alta velocidad basada en el modelo ControlNet
 - No requiere procesador para controlar el tráfico
 - E/S y comunicaciones disponibles para cualquier procesador
- Inserción y extracción de módulos bajo tensión (RIUP)
 - cualquier módulo, incluyendo procesadores y interfaces de comunicación
- Sin dependencias del número de slot
 - cualquier número de modulos de comunicación
 - cualquier número de procesadores



Comunicaciones

- Modelo Productor / Consumidor
 - el chasis ControlBus es una extensión de la red
- Comunicaciones modulares
 - cualquier número y combinación de módulos de comunicación
 - Ethernet, ControlNet, DeviceNet
 - DH+ / RIO
 - Otras
- Bridging y Routing
 - No se requiere procesador para gestionar las comunicaciones

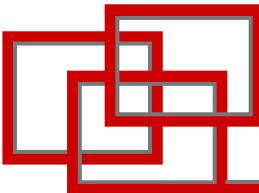




E/S 1756

- **Modelo Productor / Consumidor**
 - Producción cíclica o por Cambio de Estado
 - Información E/S disponible para cualquier procesador del sistema
 - chasis local o en red
- **Altas prestaciones**
 - 0 - 60 °C
 - Canales E/S aislados
 - Diagnósticos avanzados
 - Marcaje tiempo (Datado)
- **RIUP**
 - Tanto parte chasis como parte campo
 - Reconfiguración automática del módulo sustituido
- **Configurables por software**
 - sin interruptores
 - asistentes configuración RSLogix5000

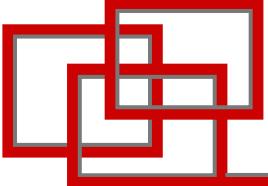




Características E/S Digitales Estandard

- Elección aisladas y no aisladas
- Densidad de 8, 16, y 32 puntos
- Tiempo de filtro configurables por software
 - DC: 0 a 2mS Off a On / 0 a 18mS On a Off
 - AC: 1 a 2 mS Off a On / 9 a 18 mS On a Off
- Definible estado Salida para modo programa y fallo comunicación (Mantener, Activada o Desactivada)
- Dos métodos multicast de enviar valor entradas
- Eco de salidas a todo el sistema
- Activación salidas en base al reloj del sistema
- Fusible electrónico en algunos modelos de salidas

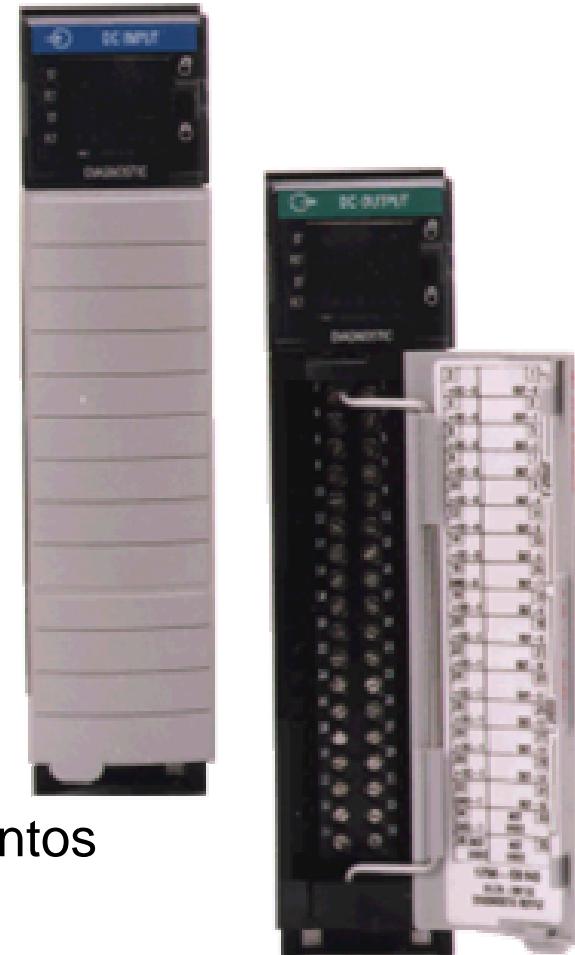


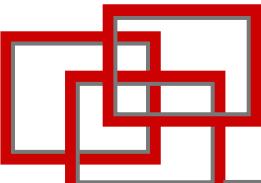


Diagnosticos y características E/S digitales

Entrada Salida

- S Detección Cable abierto
- S Detección falta carga
- S Verificación salida a campo
- S Fusible electrónico por punto
- S Pulso verificación salida
- S S Marca tiempo en diagnóstico
- S S Report fallo a nivel de punto
- S S Memorización fallo intermitente
- S S Densidades 8 Puntos CA y 16 y 32 Puntos CC

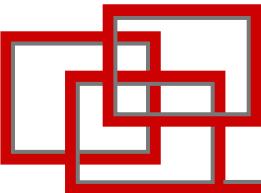




Características módulos E/S analógicos

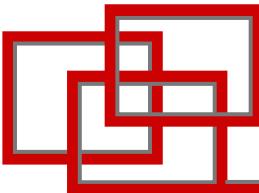
- Resolución 13-16 bit - dependiente del rango
- Formato datos Entero o coma flotante
- Rápida actualización de canales:
 - Temperatura: 50 mS (C.F)
 - Analógica aislada: 25 mS (C.F) / 10 mS (Ent)
 - No aislada: 18 mS (C.F.) / 5 mS (Ent)
- Límite de gradiente de salida
- Estado configurable frente a estado de fallo modo de programa por punto de salida:
 - Mantener último estado
 - Ir a un valor seguro especificado
- Enclavamiento de salida a un valor bajo-alto especificado
- Característica de congelar salida al inicio de aplicaciones PID





Características módulos analógicos de entrada

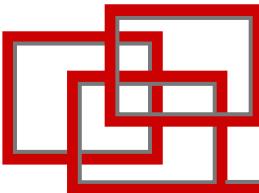
- Escalado a unidades de ingeniería en tarjetas
- Alarmas en tarjetas:
 - Alarmas High, High-High, Low, Low-Low
 - Opción memoria para alarmas intermitentes
 - Opción banda muerta para control de sensibilidad
- Detección sobre/bajo rango y sensor abierto
- Datado de lectura de canal de entrada
- Almacenamiento de datos y fecha de calibración en memoria no volatil
- Selección de filtros digital, noch, pasa bajos...



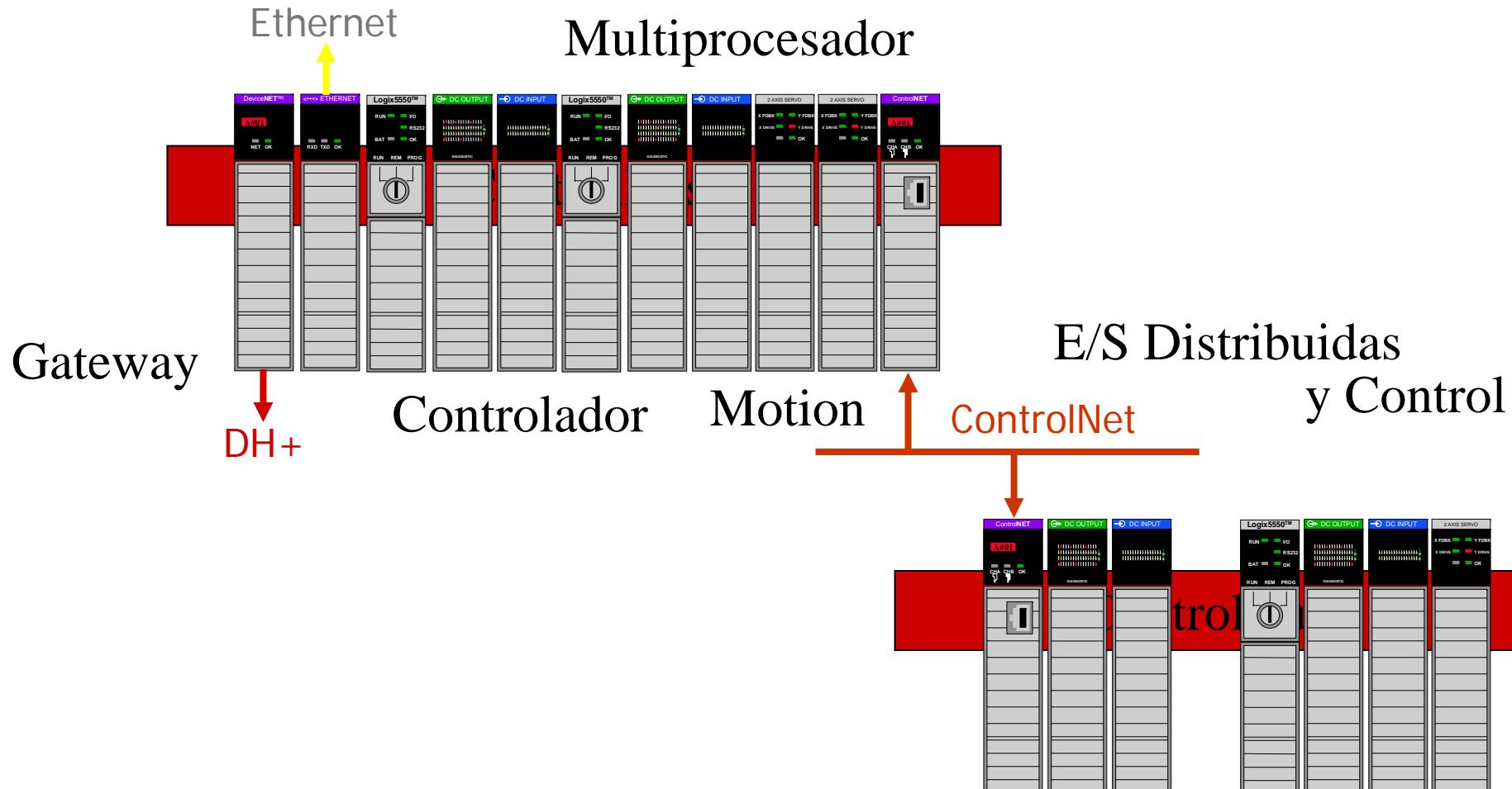
Los procesadores Logix55xx

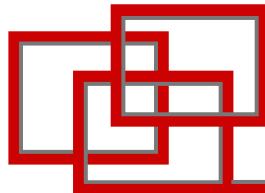


- Logix5555
 - 750Kb, 1.5Mb, 3.5Mb, 7.5Mb
 - Ejecución coma flotante más rápida
 - Soporte de Redundancia
 - Versiones con Memoria Retentiva
 - Versión LogixOS v6.0 y posterior
- LogixL6x (61,62,63)
 - 8Mb
 - Optimizados para Motion Control
- Múltiples procesadores por sistema

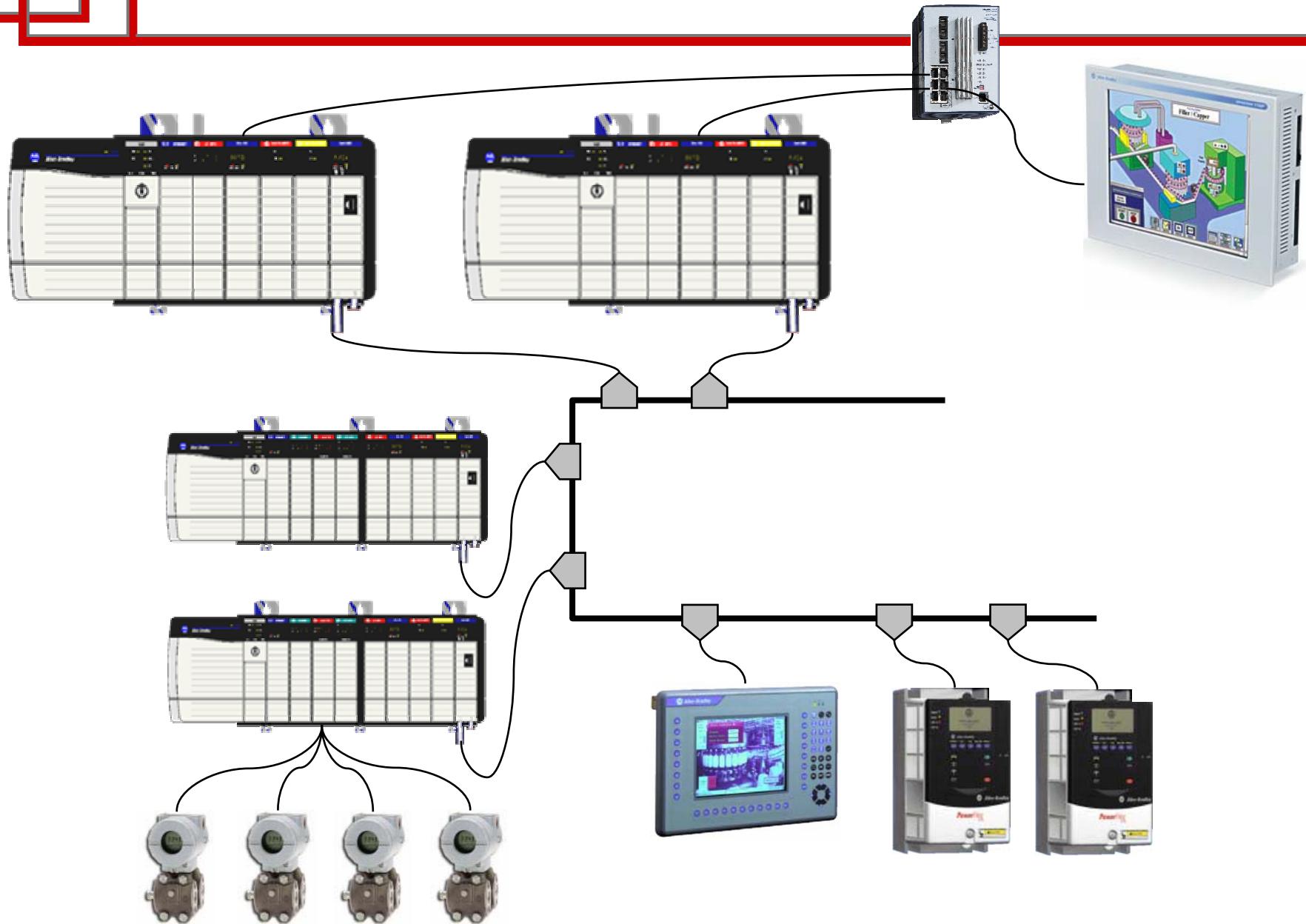


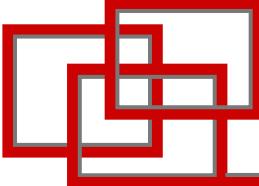
Una arquitectura ControlLogix





Redundancia de ControlLogix / SIL2



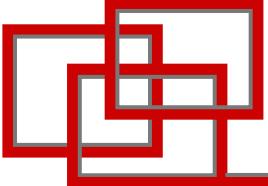


Control
Distribuido

FlexLogix

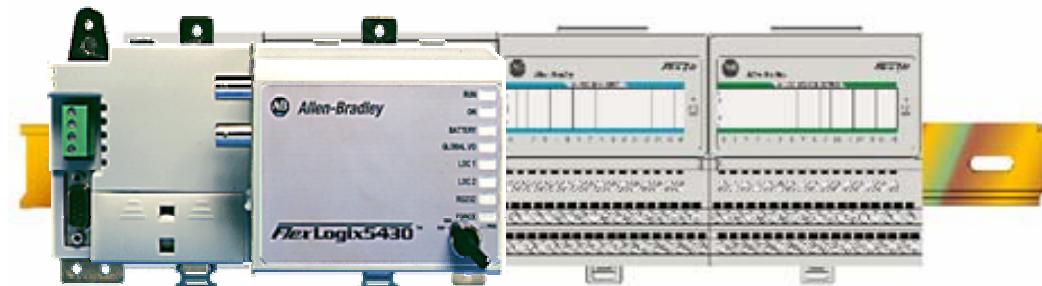


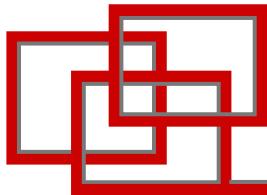
Flex I/O



FlexLogix - Distribución del Control con FLEX I/O

- FlexLogix 5434
 - 512K de memoria
 - Multiples tareas
 - 1 o 2 railes FLEX I/O
- Comunicaciones
 - 2 slots para tarjetas de comunicaciones soportando ControlNet, DeviceNet, y EtherNet/IP
 - Puerto RS-232 incorporado
- Capacidad de E/S
 - Hasta 16 módulos locales



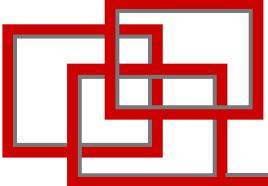


Control aislado

CompactLogix

1769 I/O

Bajo coste



CompactLogix - Solución económica para control a nivel de máquina o estación

CompactLogix L31, L32x y L35x

- 512K, 750K o 1,5 MBytes memoria
- Multiples tareas
- Diseño sin chasis con Compact I/O

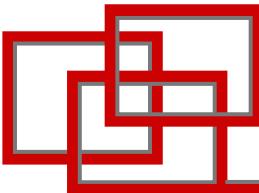
- Comunicaciones

- Uno o dos puertos RS-232
 - DF1/DH-485/ASCII
 - Con Conectividad via módulos 1761-NET
 - DH-485, DeviceNet y EtherNet/IP
- Puerto EtherNet/IP nativo en modelos “E”
- Puerto ControlNet nativo en modelos “C”
 - Capacidad para admitir multiples escaners de DeviceNet por procesador.

- Capacidad de E/S

- Hasta 16 o 30 módulos locales
CompactI/O + E/S y dispositivos remotos vía DeviceNet, ControlNet o EtherNet/IP



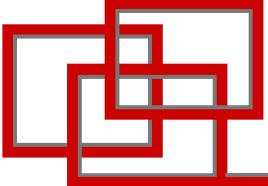


Drives con Control
de altas prestaciones

DriveLogix

Control distribuido

Control incrustado



DriveLogix - Control embebido en Drive y Drive System

Controlador Logix incorporado

Redes abiertas NetLinx :
ControlNet, DeviceNet y

EtherNet/IP

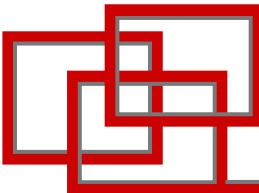
SynchLink para alta sincronización

Usa el entorno de programación RSLogix 5000



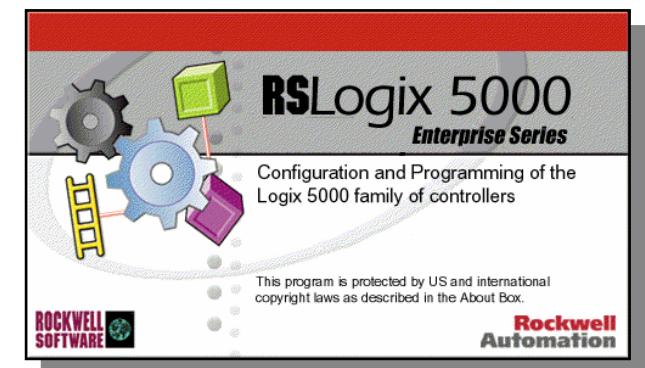
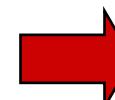
Estructuras PowerFlex , Interfaces de operador, Coms y E/S

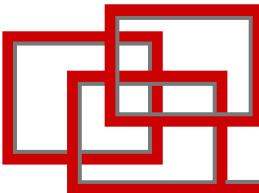
Control de motor de altas prestaciones con V/Hz, lazo cerrado y abierto y algoritmo Brushless



Valor de Logix: Control Escalable

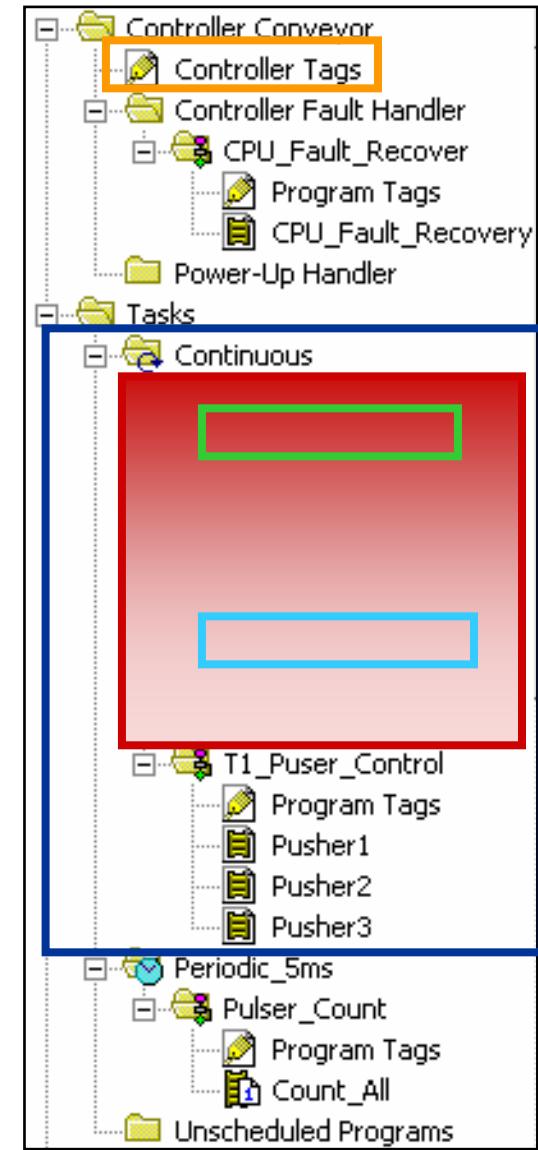
- Controladores para aplicaciones pequeñas , medianas o grandes
- Procesadores especializados (incrustado en variadores y soft PLC)
- Programados todos ellos con RSLogix 5000



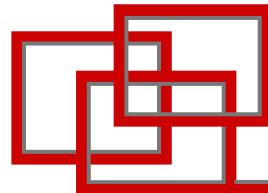


Tareas / Programas / Rutinas

- Un programa está “asociado” o “integrado” en una Tarea.
(TASK) Dependiendo de la CPU hay hasta 32 tareas, típicamente una continua y el resto periódicas.
- Cada tarea puede tener hasta 32 programas (100 en CLX)
- Un **programa (PROGRAM)** consta de:
 - una **base de datos** particular de este programa y
 - un número de **rutinas (ROUTINE)**
- Las rutinas se “escriben” o se “dibujan” en diferentes lenguajes de programación. Contienen el código exe
- Una rutina sólo puede ser creada, editada y vista en un lenguaje determinado
- Las instrucciones (de las rutinas) en cualquier lenguaje pueden acceder a las variables de su **base de datos** particular del **programa (Program Tags)** o a la **base de datos** del controlador (**Controller Tags**)
- En un programa puede haber un número “ilimitado” (solo limitado por tamaño de memoria) de rutinas en todos los lenguajes



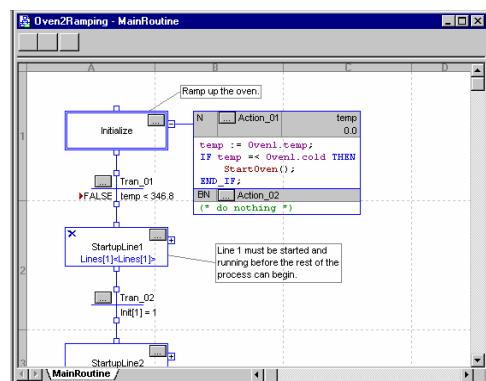
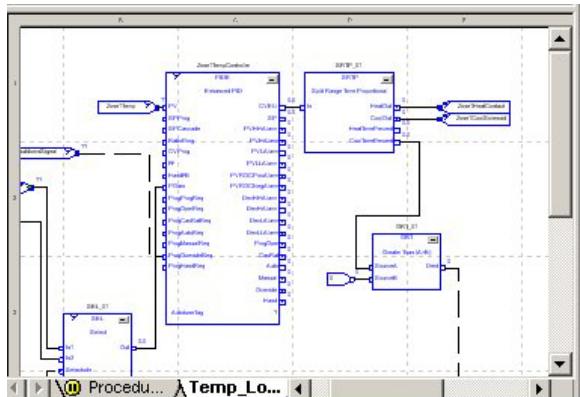
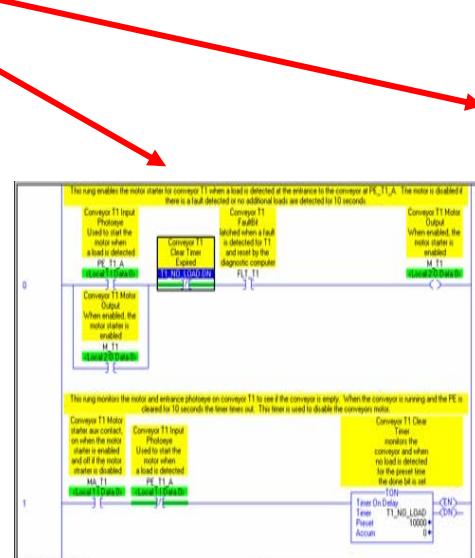
Los objetos de programa Logix

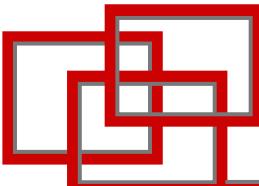


Mesa elevadora

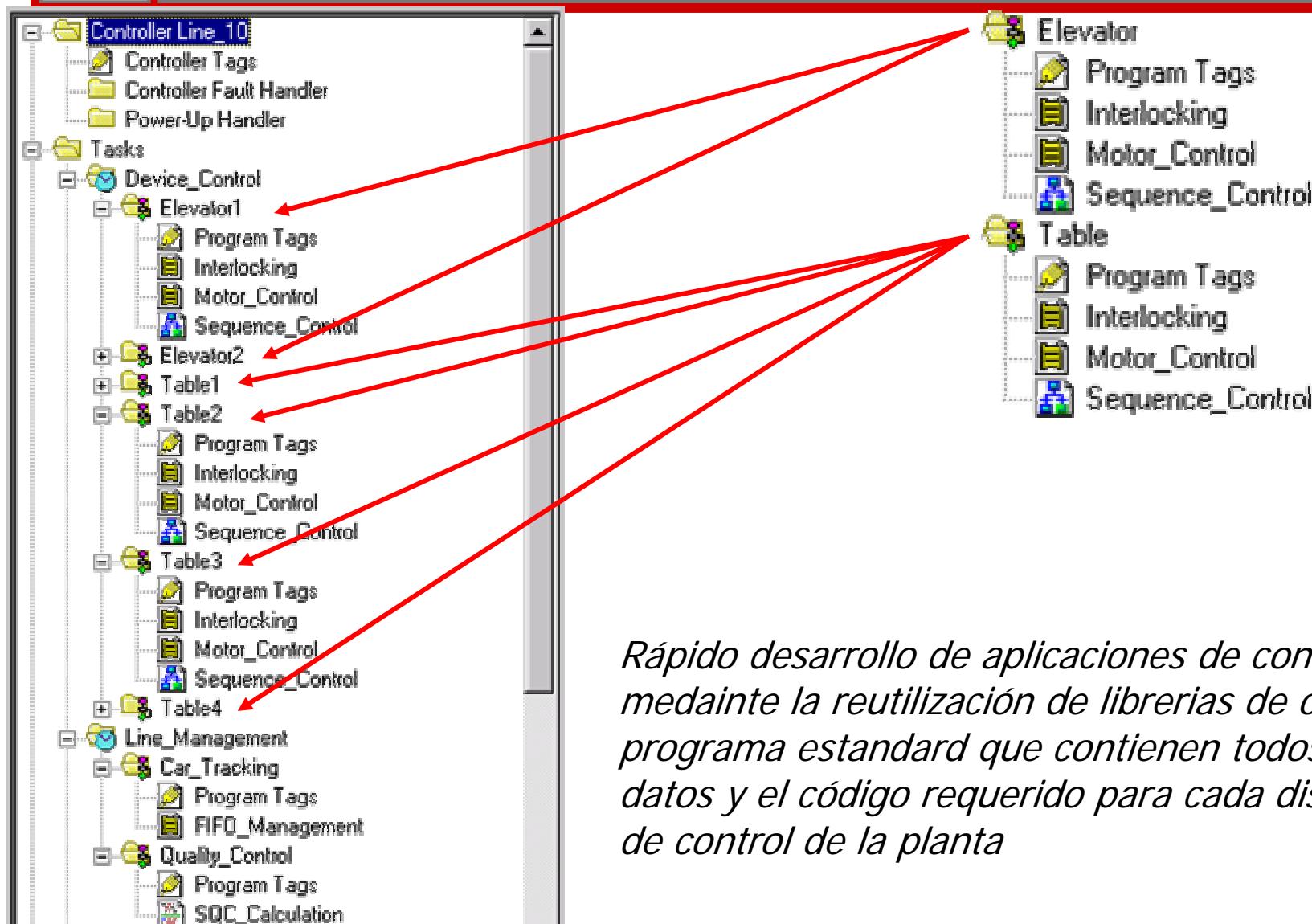


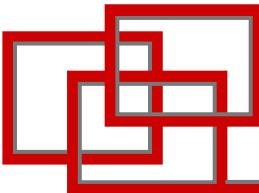
Tag Name	Type
► This_Table	Table_Data
This_Table.Loaded	BOOL
This_Table.Unloaded	BOOL
This_Table.Load_CMD	BOOL
This_Table.Unload_CMD	BOOL
+This_Table.Car_Code	STRING
'ST304BF567'	STRING





Rápido desarrollo de aplicaciones





Independiente del Hardware



Mesa giratoria



Program_Tags



Interlocking

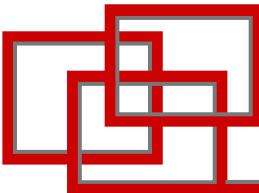


Motor_Control



Sequence_Control

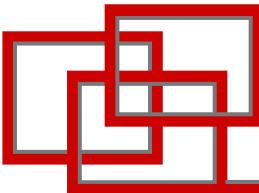




Beneficios de Logix: Control escalable

- Reduce tiempo de ingeniería
 - Puede re-utilizarse el código
 - Elimina formación adicional
- Reduce el coste del hardware – compre solo lo que necesite y amplie las capacidades de su equipo de control cuando lo necesite
- Reduce formación – elimina la curva de aprendizaje



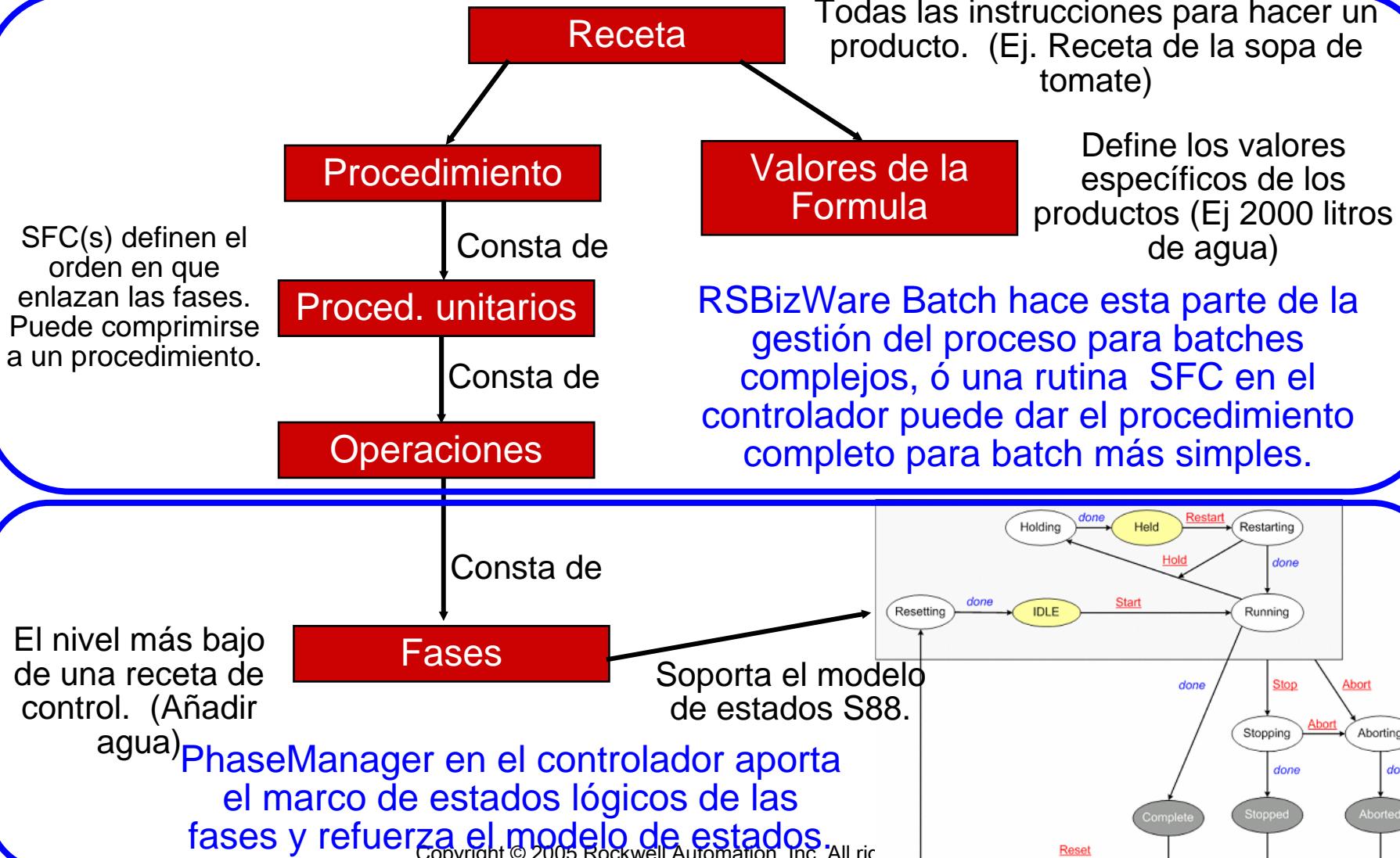


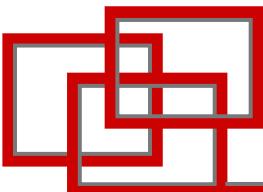
PhaseManager™

- Incorpora el modelo S88 de estado de fases en los procesadores Logix
- Proporciona el marco para seguimiento completo de estados de las fases
 - Refuerza el modelo de estado de fases S88
- Elimina código Phase Logic Interface (PLI) con RSBizWare Batch
- Proporciona una fácil integración con RSBizWare Batch
- Las fases pueden secuenciarse por:
 - RSBizWare Batch
 - Una rutina SFC en el propio controlador
- RSLogix 5000 proporciona capacidad de control manual de fases útil para puestas en marcha y localización de averías
- Potencia el uso de estado de máquina “S88” en aplicaciones NO-Batch

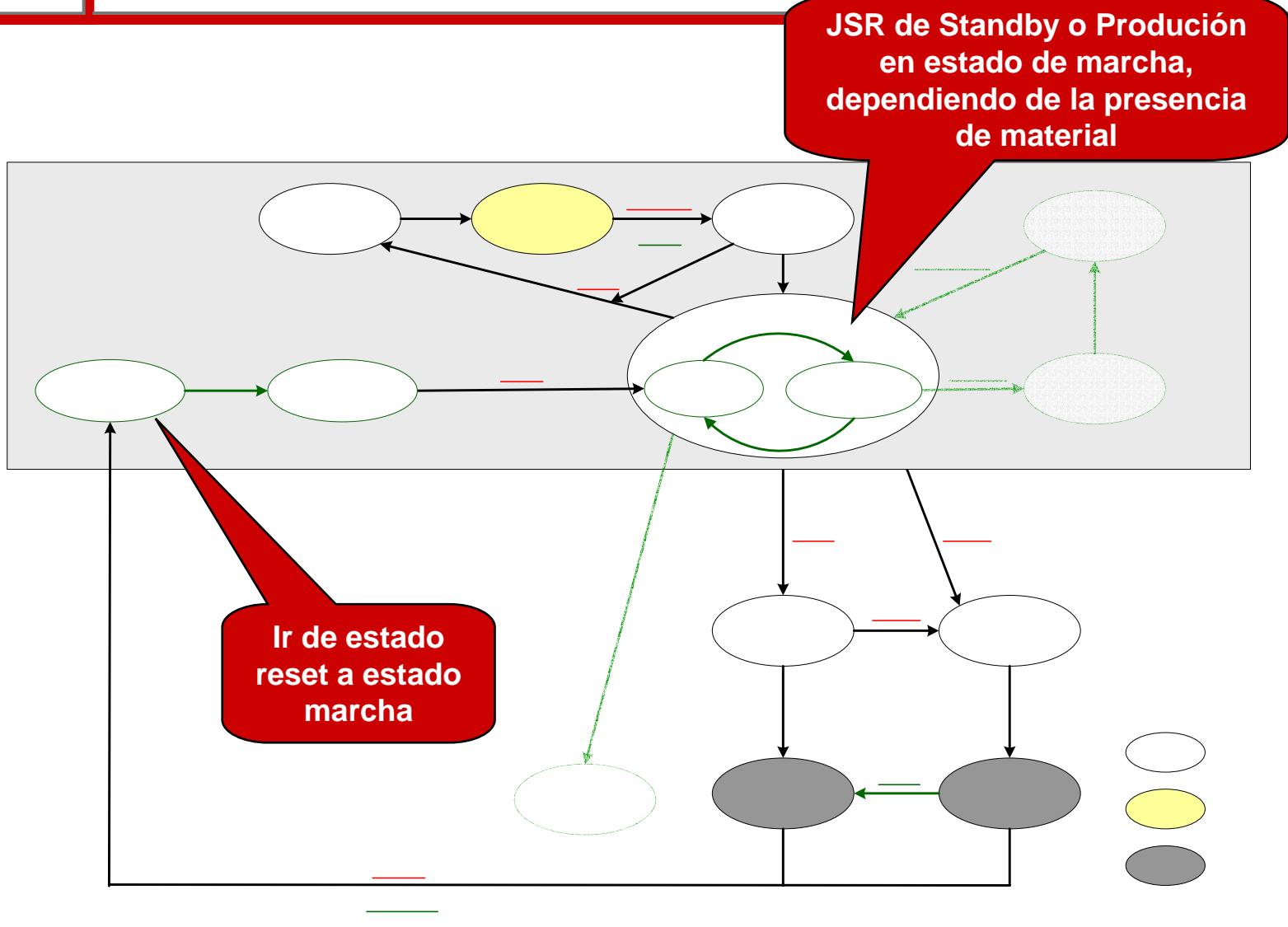
¿Como encaja PhaseManager en S88 ?

- El estandard S88 define terminología y metodología de los procesos batch.



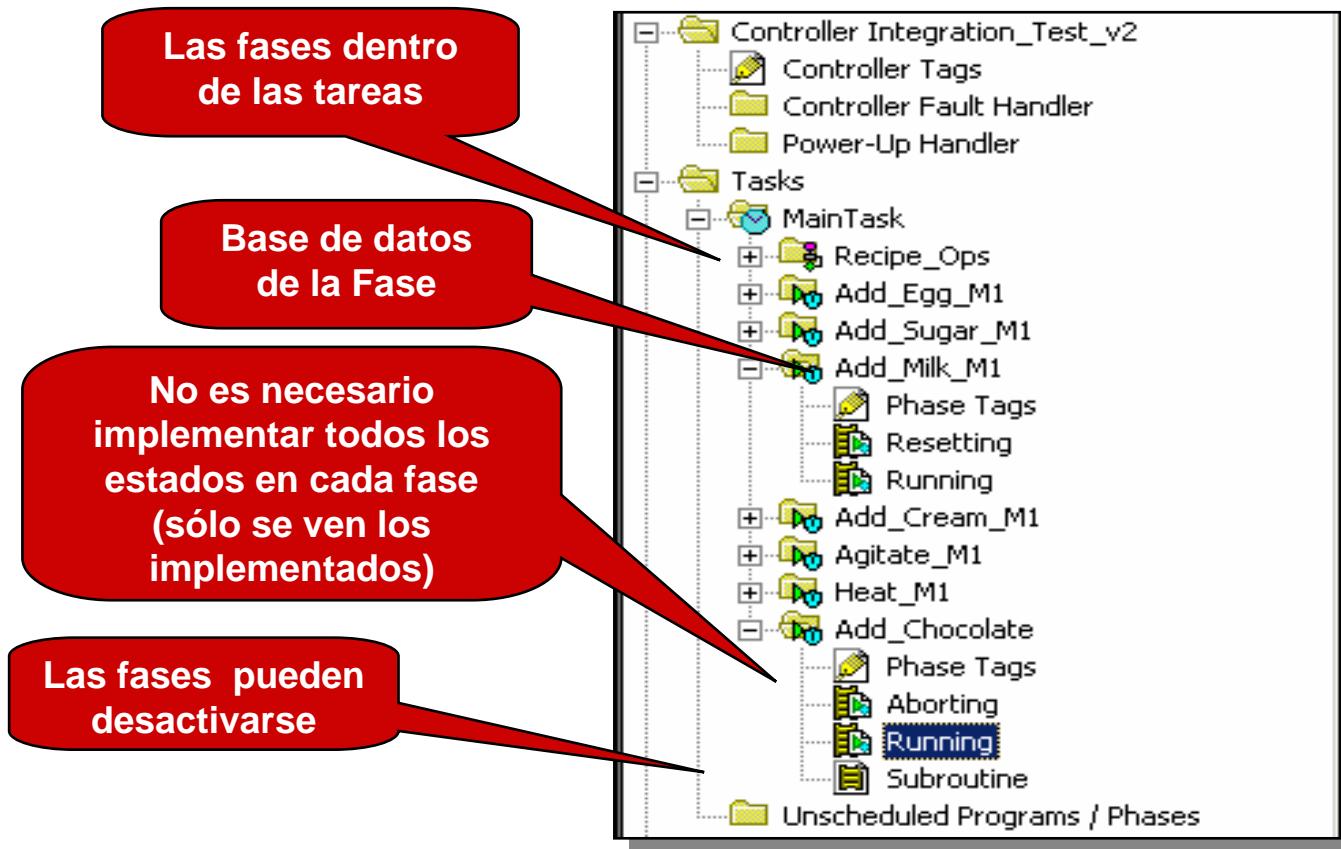


OMAC/PackML- También encaja en el modelo S88



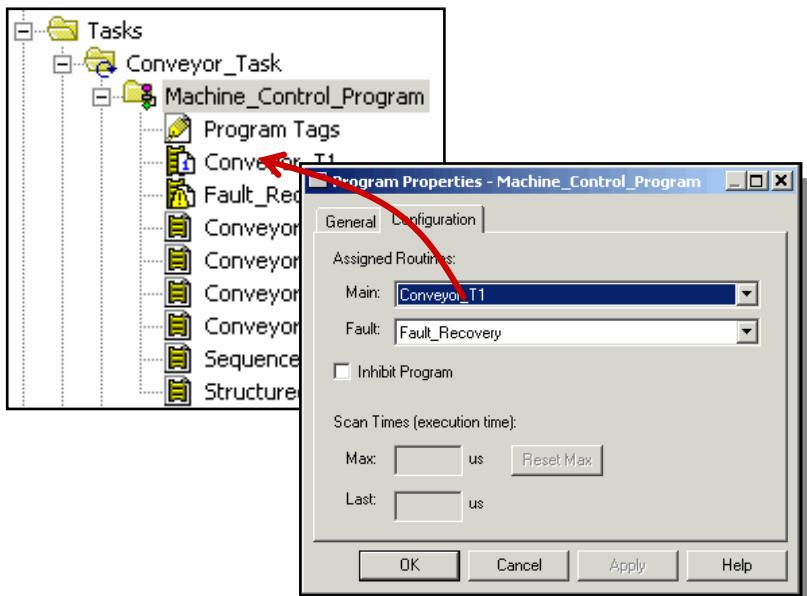
Vista del arbol organizador del controlador

- Las fases se crean y planifican dentro de las tareas, similar a los programas
- Las rutinas de estado de las fases se crean dentro de las fases
- Pueden escribirse en cualquier lenguaje de programación de Logix

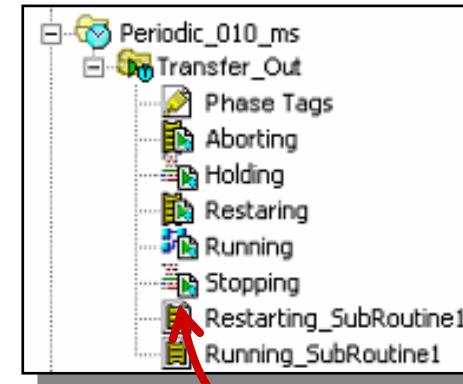


Programas Vs Fases

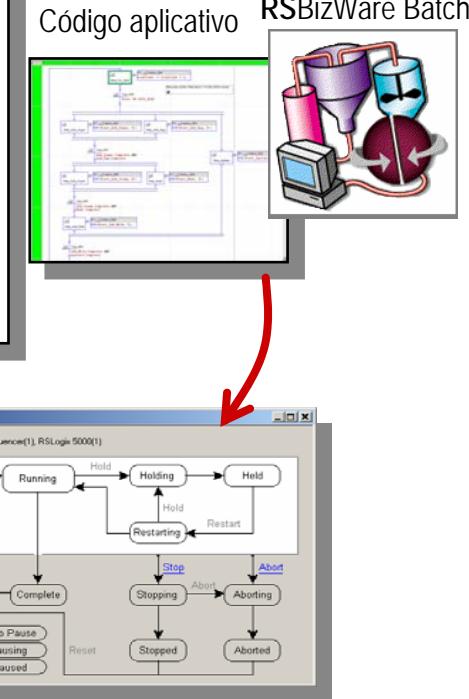
Programa



Fase

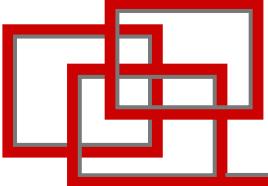


Secuenciador Batch



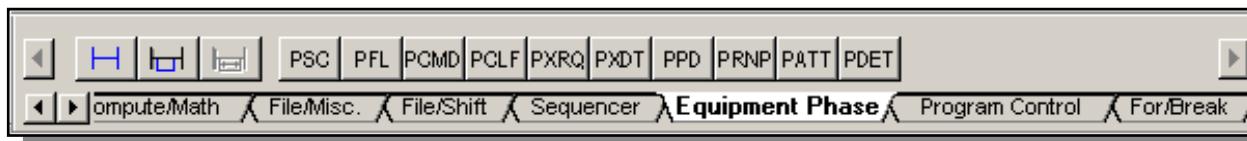
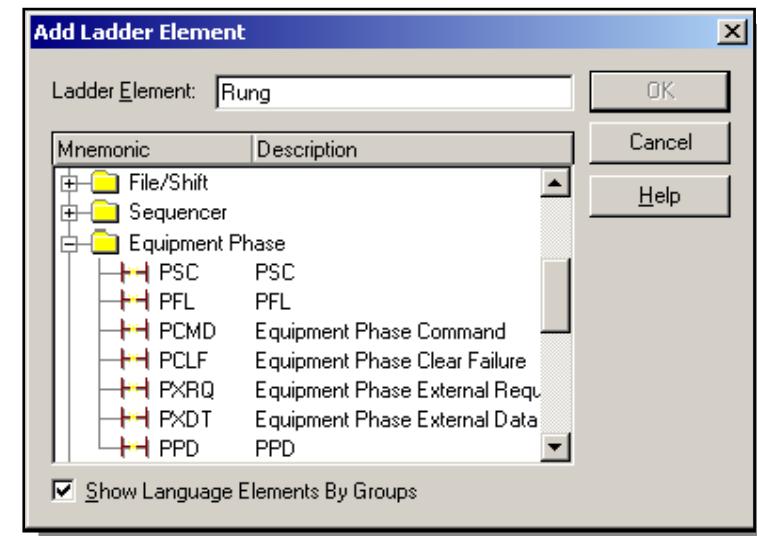
- 1 Rutina principal por programa
- La Rutina principal se ejecuta cada vez que entra la tarea y llama a las otras rutinas mediante la lógica del usuario
- Hay que analizar el código para saber que se está ejecutando

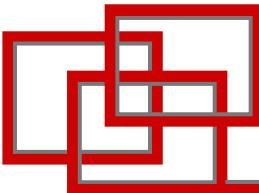
- Las rutinas "Ing" actúan como rutinas principales
 - Aborting, Holding, Resetting, Restarting, Running y Stopping
- La ejecución es determinada por el secuenciador de fases (RSBizWare Batch o código de la aplicación)
- Existe la herramienta "Phase monitor" para ver que rutina se está ejecutando



Nuevas Instrucciones

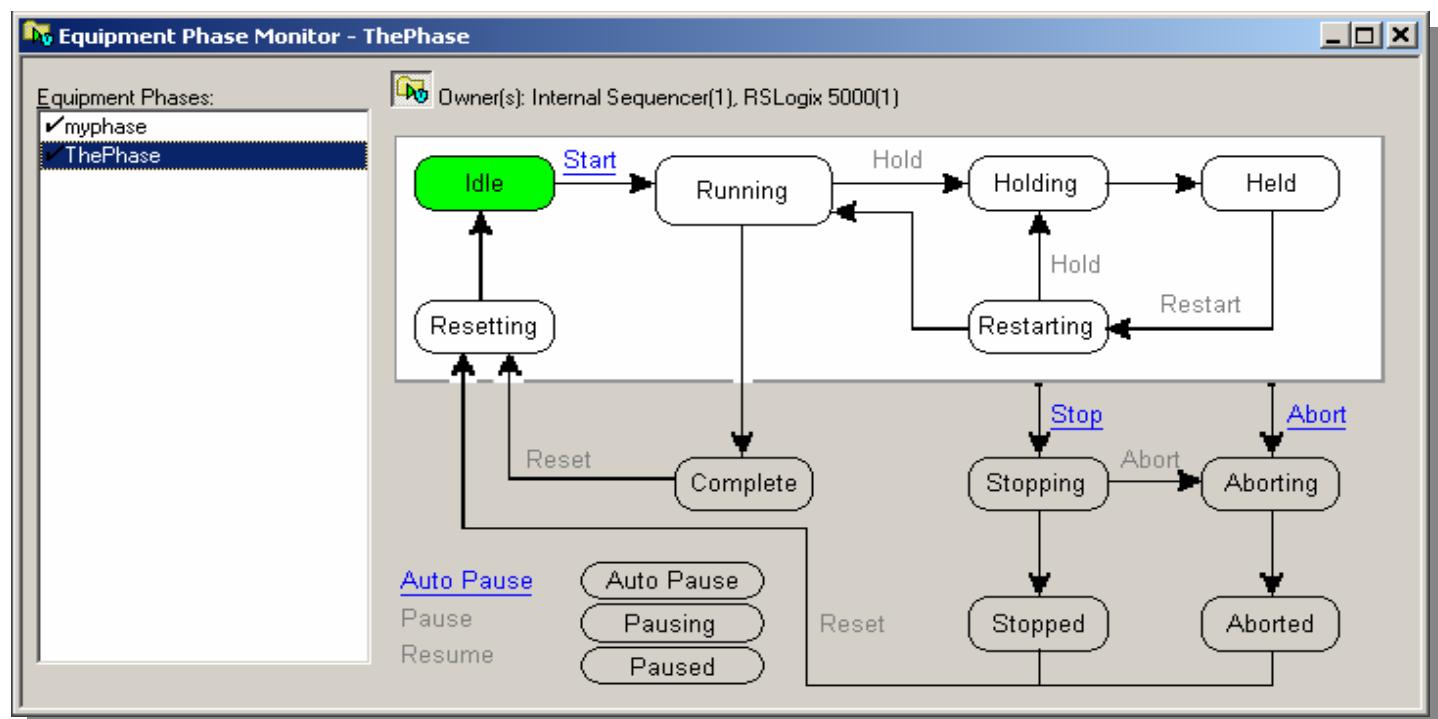
- Se ha añadido un set de nuevas instrucciones para permitir a las fases interactuar con los procedimientos RSBizWare Batch o con las rutinas SFC del controlador :
 - PSC – State Complete
 - PCMD – Equipment Phase Command
 - PATT – Phase Attach
 - PDET – Phase Detach
 - PFL, PCLF – Set and Clear Phase Failures
 - PPD – Pause Equipment Phase
 - PRNP – Reset New Parameters
 - PXRQ – Phase Request to External Sequencer
- Todas las instrucciones disponibles en lenguajes Ladder (RLL) y Structured Text (STX)

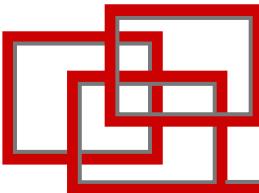




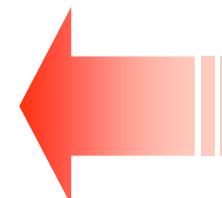
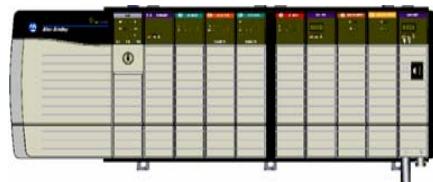
Phase Monitoring

- Phase Monitor le da a RSLogix 5000 la capacidad de monitorizar y gobernar las fases y los comandos de estado de las fases.
 - Util para puesta en marcha y localización de fallos o averías.





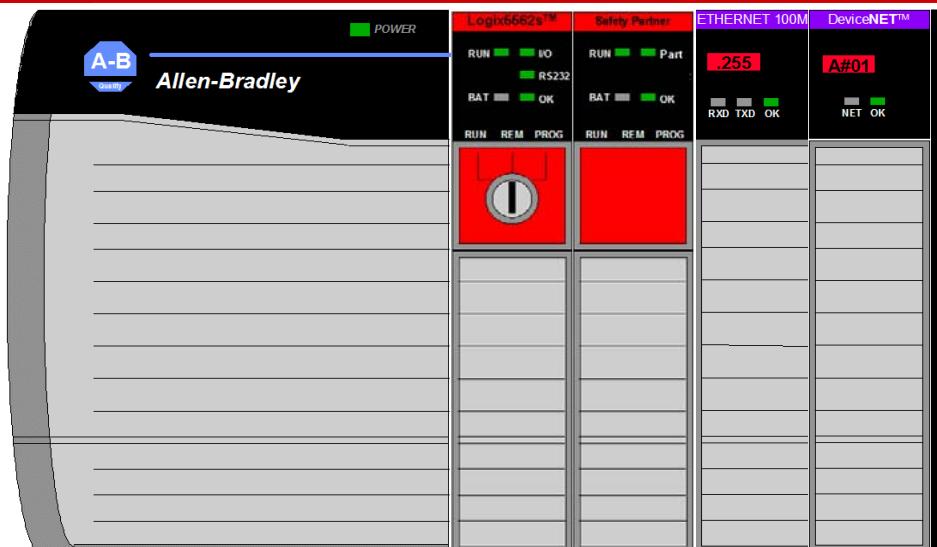
Integración ControlLogix Safety



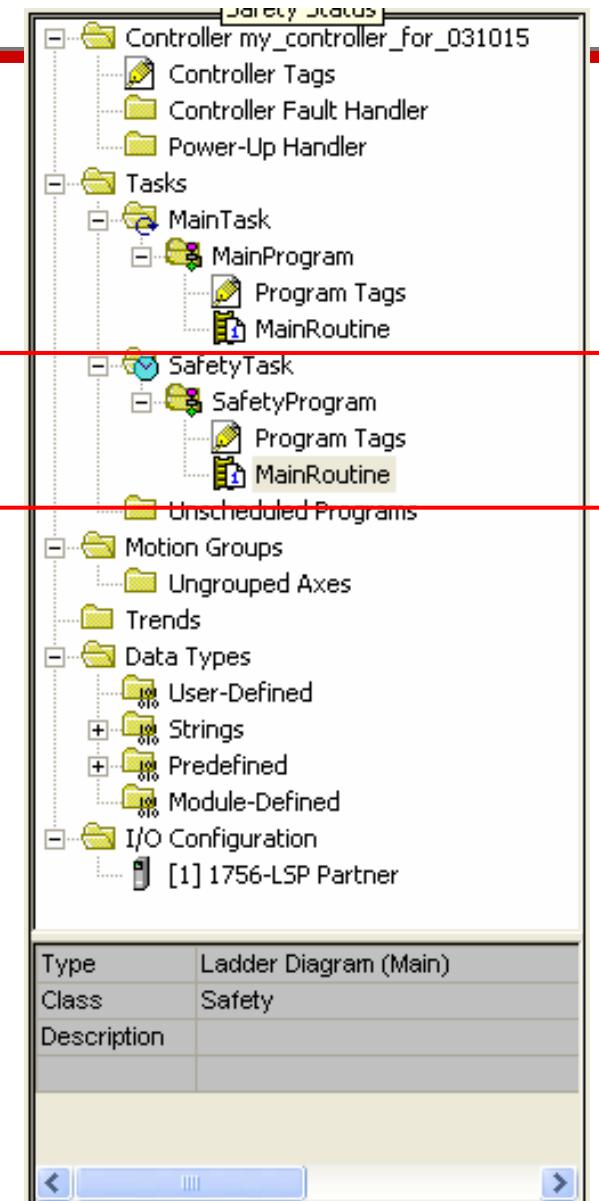
GuardLogix

- Seguridad (Safety) integrada en ControlLogix
 - Solución de procesador dual (Arquitectura 1oo2D)
 - Certificación SIL-3 según IEC 61508
 - Categoría 4 según EN 954-1
 - Programación con RSLogix5000
 - Instrucciones certificadas coincidentes con las de los productos GuardPLC
 - Conectividad con DeviceNet Safety

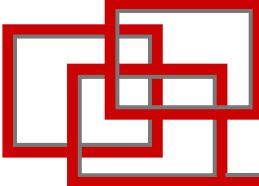
Safety Integrada en Logix - GuardLogix



- Procesador ControlLogix Integrado con procesador de Safety para crear un procesador certificado SIL 3
- RSLogix5000
 - Safety Task creada con herramientas restrictivas a la implementación de funciones de Safety
 - Árbol del proyecto único (Single controller view)
- Programa de control estandar
 - Todas las prestaciones de un controlador Logix. Sin restricciones.
- Tareas de Safety: Safety Control
 - Set de instrucciones, características y funciones restringido a tareas de Safety
 - Instrucciones específicas para Safety
- La tarea de Safety, direcciona E/S de Safety

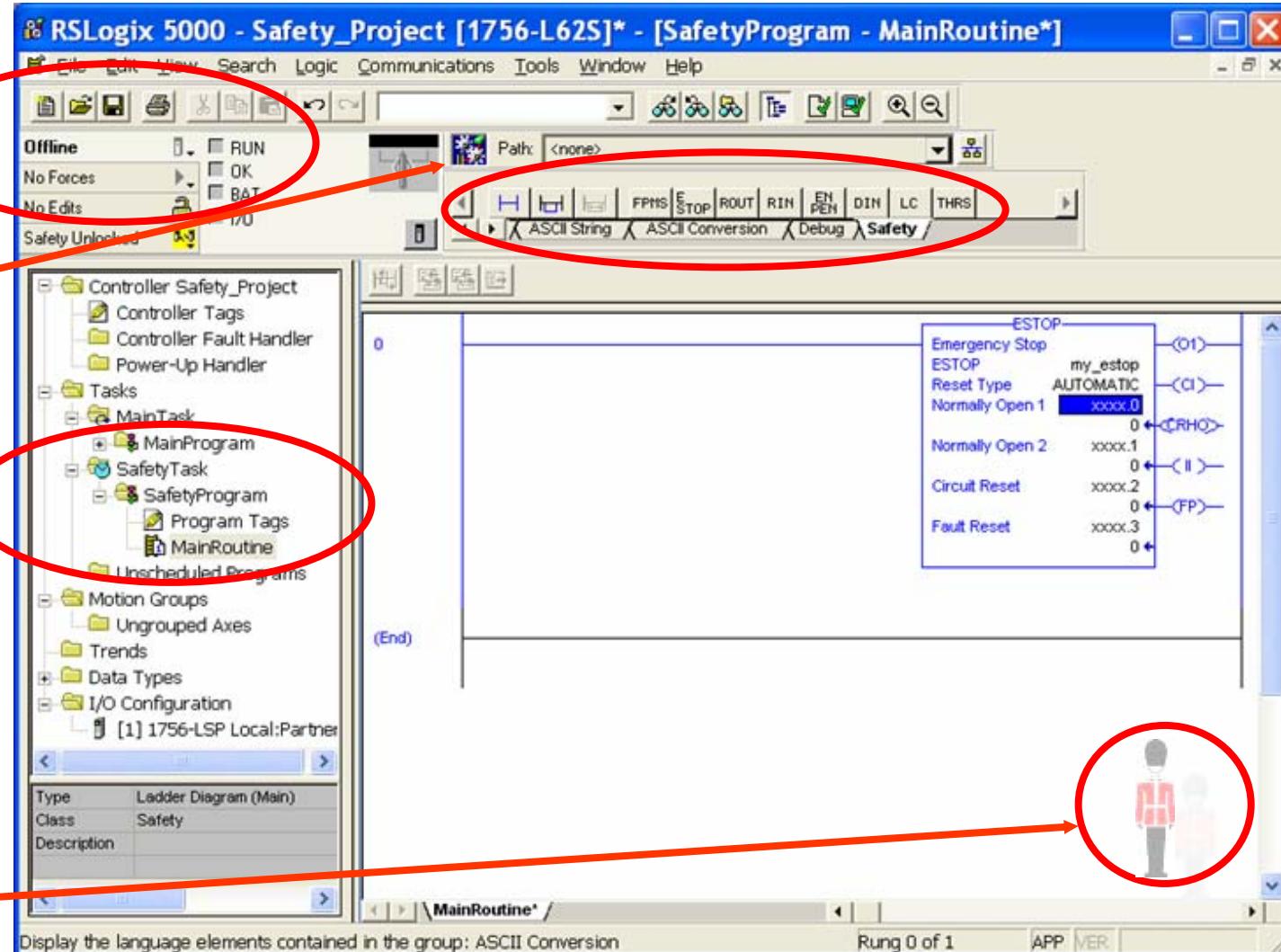


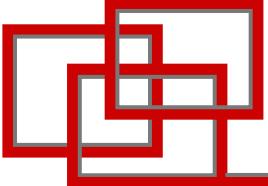
Type	Ladder Diagram (Main)
Class	Safety
Description	



Entorno explícito de Safety

- Estado del controlador de Safety
- Paleta de instrucciones Safety
- Tarea periódica de Safety
- Marca de agua de pantalla de edición de Safety

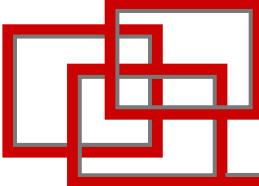




Instrucciones para aplicaciones de Safety

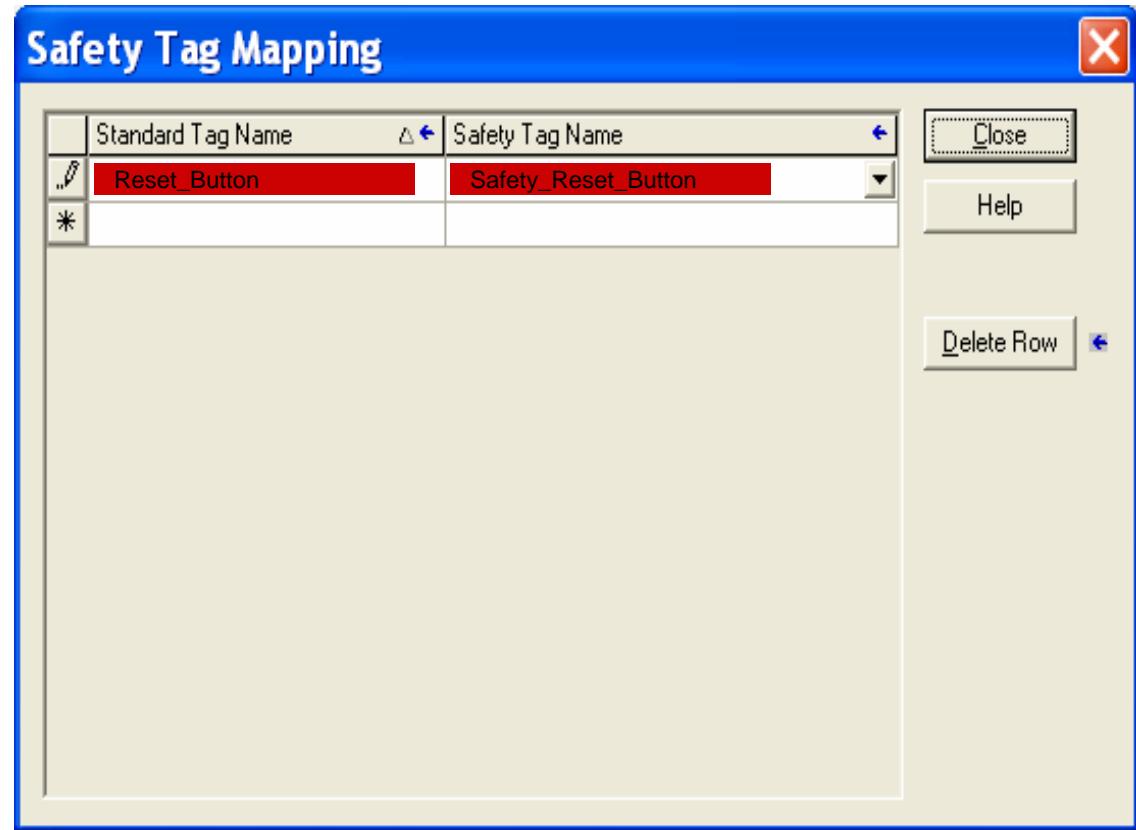
La lista de instrucciones:

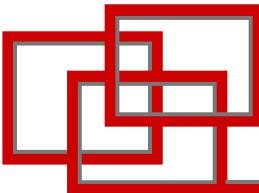
1. Diverse Input with Auto Reset
2. Diverse Input with Manual Reset
3. Emergency Stop with Auto Reset
4. Emergency Stop with Manual Reset
5. Redundant Input with Auto Reset
6. Redundant Input with Manual Reset
7. Redundant Output
8. Enable Pendant with Auto Reset
9. Enable Pendant with Manual Reset
10. Two Hand Run Station with Active Pin
11. Two Hand Run Station without Active Pin
12. Light Curtain with Auto Reset
13. Light Curtain with Manual Reset
14. Five Position Mode Selector



Mapeado de Tags estandard con Tags de Safety

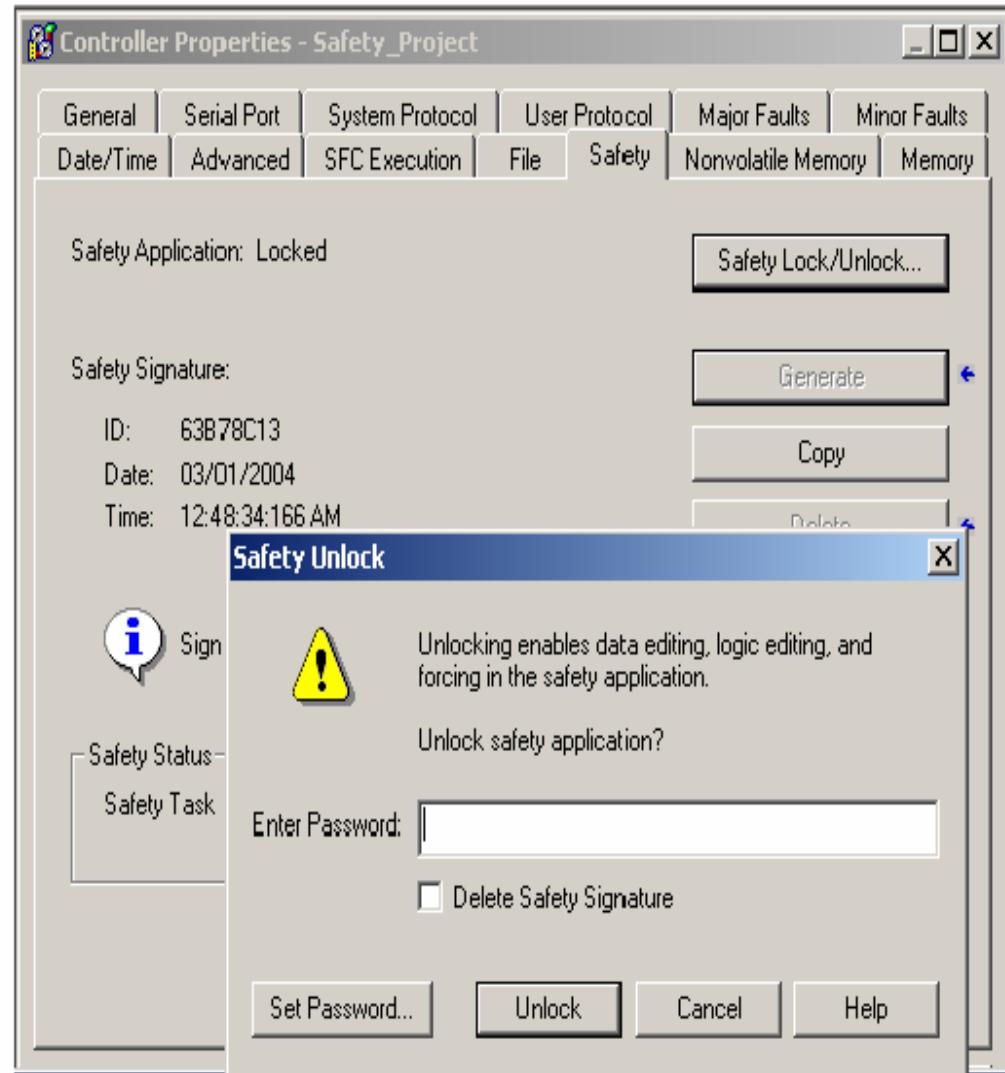
- Los Tags de Safety pueden leerse con la lógica estandard, pero no escribirse
- Proporcionan entrada de estado de tarea de Safety al control estandard
- Escaneados al principio de la tarea de Safety
- Utilizables en entradas HMI, reset de circuitos, etc.
- Deben ser cualificados para usarlos en la tarea de Safety





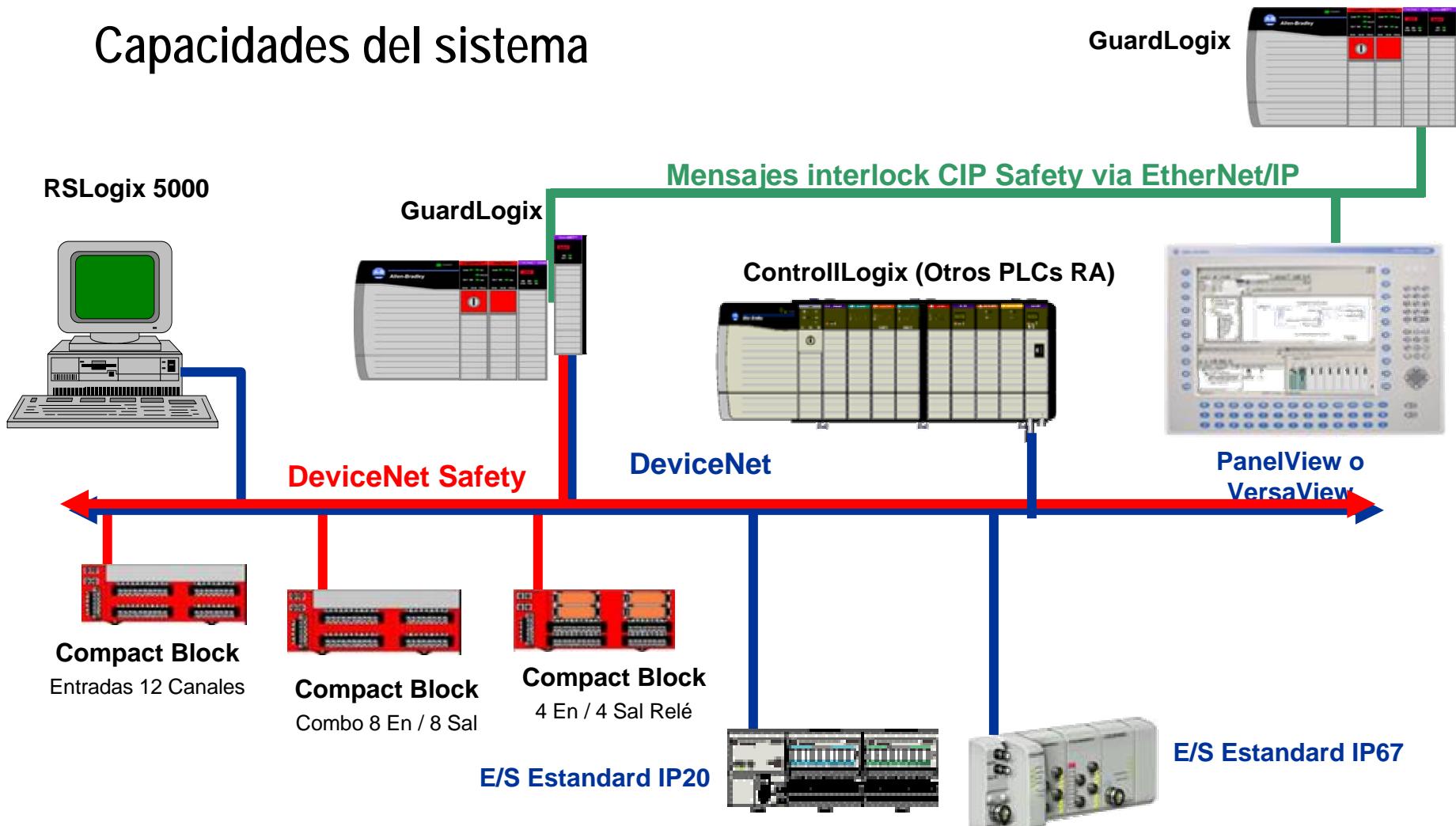
Security / Control de acceso

- Bloqueo del acceso al proyecto y al controlador para prevenir cambios
 - Protección por Password
 - Diferente Password usado para desbloquear
 - Prevención edición de datos, programa y forzados
- Safety Signature
 - Usado para Identificar rápidamente si la tarea de Safety ha sido cambiada
 - Generado usando una combinación de Checksum de la lógica, fecha y hora

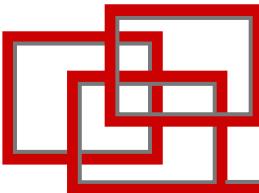


Safety integrada en DeviceNet Safety

Capacidades del sistema



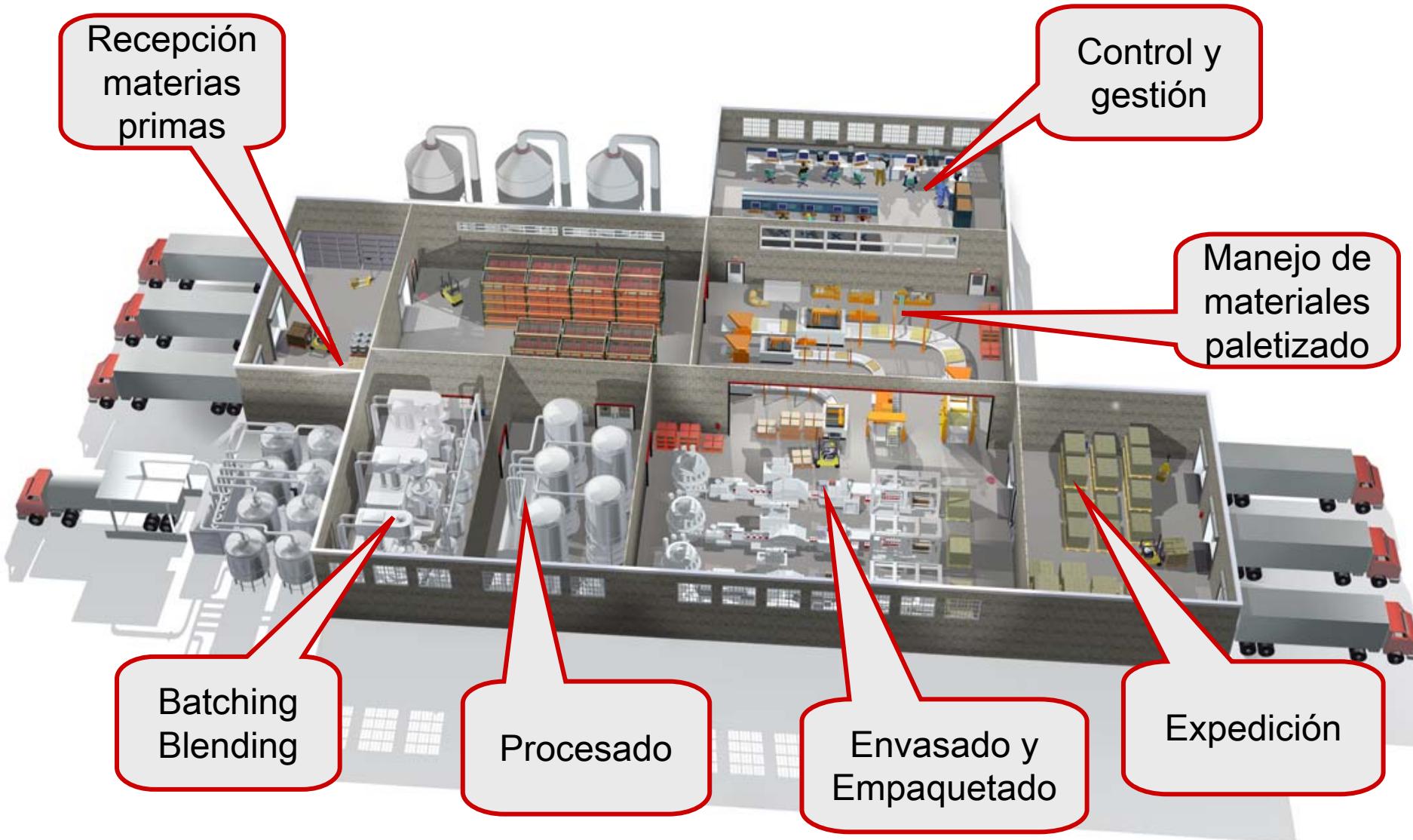
Expansión de funcionalidad a EtherNet/IP Safety (Safe IO en EtherNet/IP)

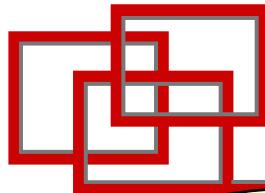


Resumen GuardLogix

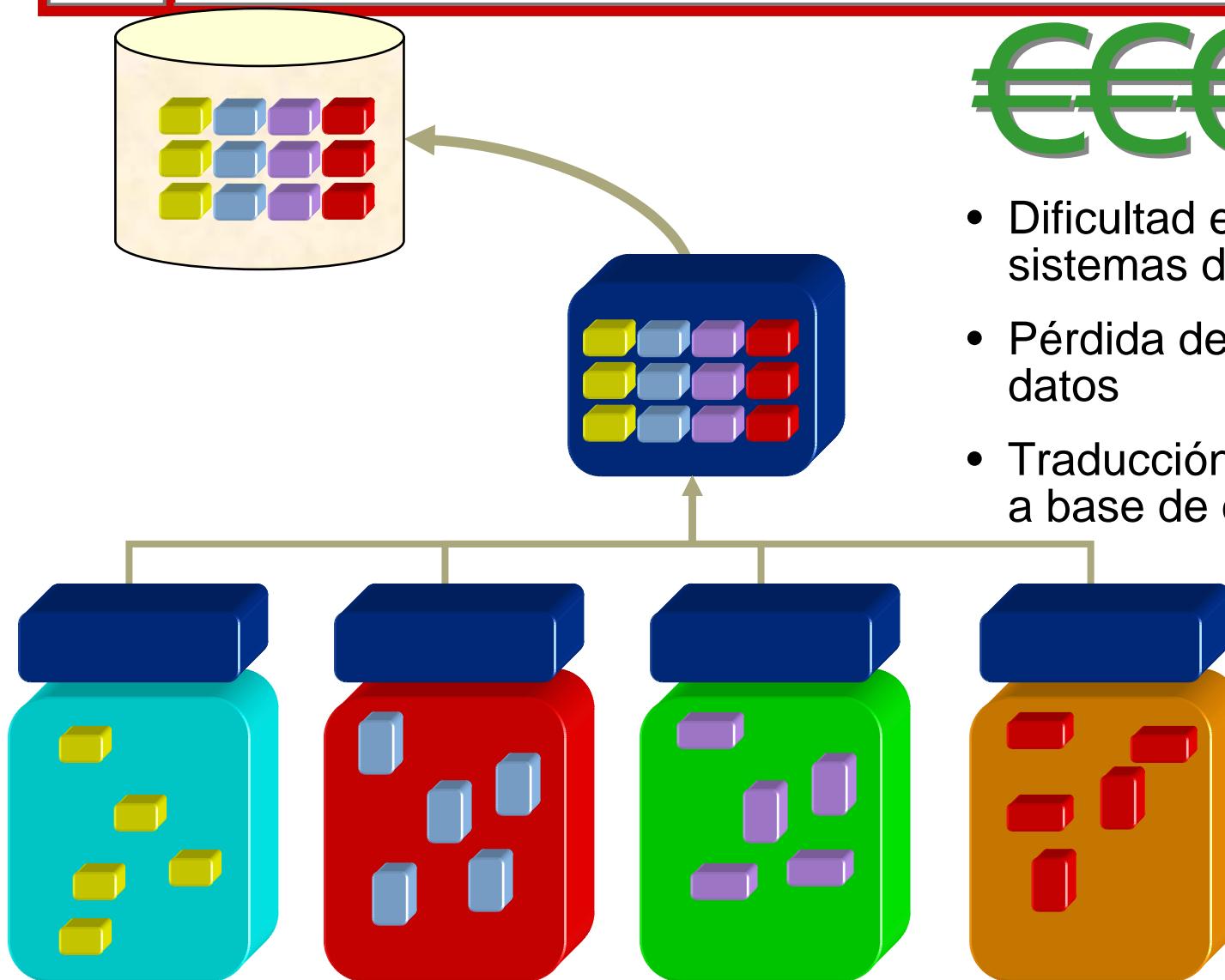
- GuardLogix - Safety Integrated Controller
 - Unica visión del proyecto de control estandard y Safety
 - GuardLogix basado en tecnología Logix estandard
- Herramienta común de Programa / Diseño / Configuración
 - RSLogix 5000+ Extensiones para Safety
 - Control estandard via RLL, FBD, SFC, STL y Control Safety via RLL
- Utilización de hardware estandard ControlLogix
 - Chasis, Fuentes, Comunicaciones...
- Instrucciones certificadas para Safety
 - Simplifica la creación de aplicaciones del usuario
 - Biblioteca básica disponible al lanzamiento
- Security Environment attached to safety
- Entorno de Seguridad (Security) ligado a la Seguridad (Safety)

Planta típica de fabricación

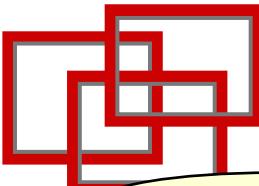




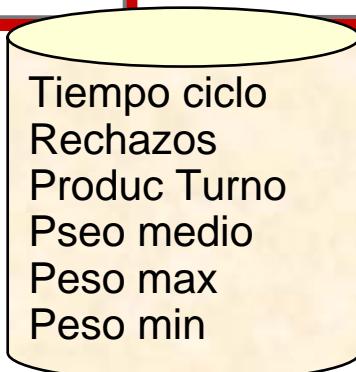
Integración tradicional de planta



- Dificultad en integrar sistemas diferentes
- Pérdida de fluidez de datos
- Traducción de datos tabla a base de datos

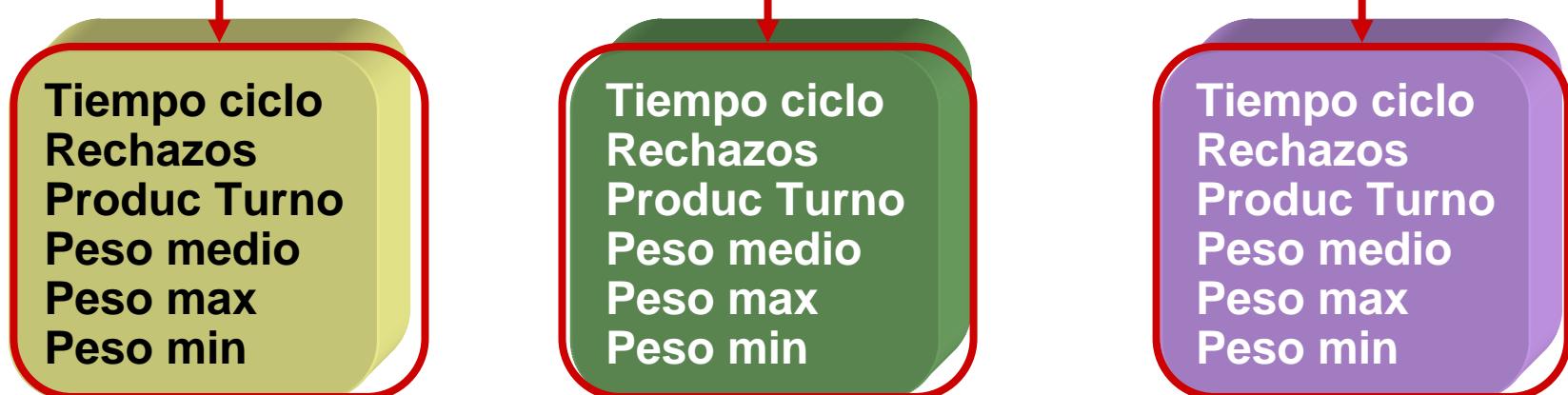


La solución Logix / NetLinx



- La convención de nombres de Logix resuelve los problemas de traducción
- El protocolo Common Industrial Protocol (CIP) en NetLinx asegura disponibilidad de datos

CIP



Ejemplo: Adquisición de Datos



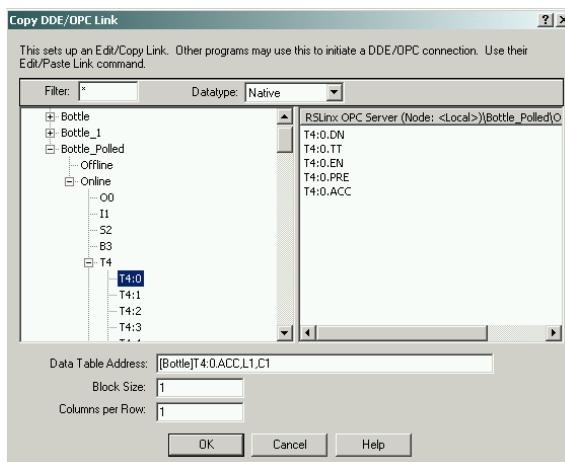
IT Ethernet

Factory Talk
RSSQL

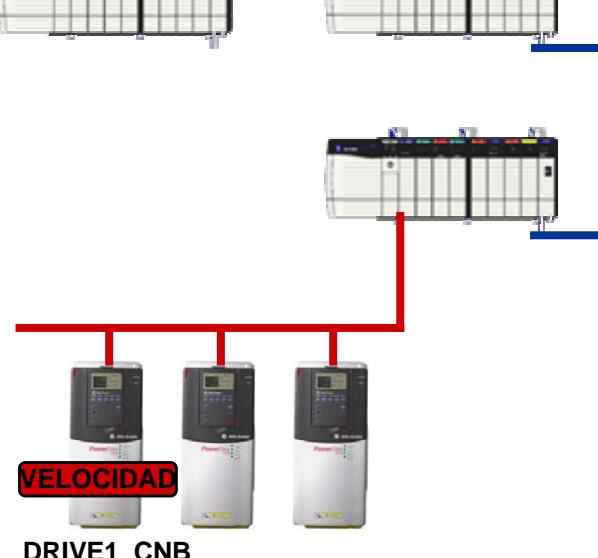
EtherNet Planta

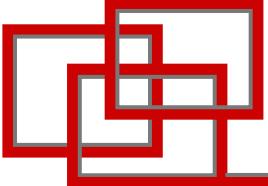


Supervisión



OPC
OLE for Process Control

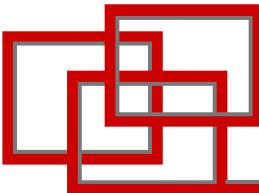




Beneficios de Logix: Control Multi-Disciplina

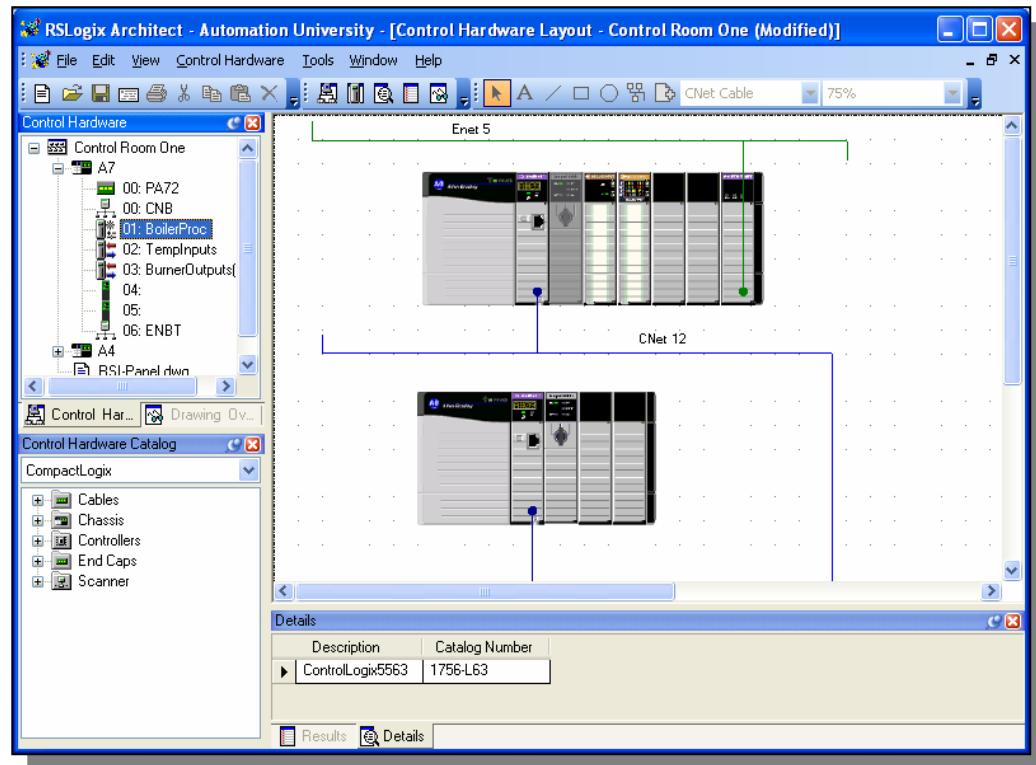
- Reduce tiempo de parada – encontrar elemento de fallo en un programa
- Reduce tiempo de ingeniería (elimina extenso código para tener las partes secuencial, de procesos, de ejes y variadores comunicando con las otras partes)
- Mejora la eficacia (una persona de mantenimiento soporta todo)
- Adquisición de un único paquete de software de programación
- No hay extra H/W o S/W que comprar para establecer gateways
- Reduce inventario de repuestos usando un conjunto de recambios para todo
- Reduce formación de personal de mantenimiento al usar el mismo hardware y software para todas las disciplinas





RSLogix Architect™

- Punto central de gestión de proyectos
- Gestión de comunicaciones y relaciones de tags producidos/consumidos
- Proporciona mejoras en configuración y mantenimiento del sistema
- Incluído en RSLogix 5000 Professional Edition o vendido separado
- Planes de migración para usuarios de RSLogix 5000 Professional bajo acuerdo de ampliación de garantía



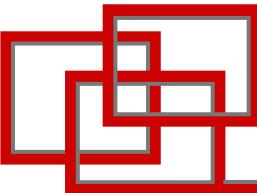
LISTEN.
THINK.
SOLVE.SM

La solución Logix es mucho más
que un PLC
(*Programmable Logix Controller*)
tradicional

Los analistas
internacionales de ARC
Advisory Group ya lo
llamaron PAC
*Programmable
Automation Controller*
cuando estaba dando los
primeros pasos

ALLEN-BRADLEY • ROCKWELL SOFTWARE • DODGE • RELIANCE ELECTRIC

**Rockwell
Automation**



La Arquitectura Integrada de Rockwell

