

DESARROLLO DE GUIAS DE TRABAJO PARA EL ÁREA DE AUTOMATIZACIÓN
INDUSTRIAL, HACIENDO USO DEL SOFTWARE FACTORY I/O CON LAS
TARJETAS USB-4704 DE ADVANTECH, ARDUINO UNO Y EL PLC SIEMENS S7-
1200

JUAN DAVID RODRIGUEZ FERRER

UNIVERSIDAD SANTO TOMAS DE TUNJA
FACULTAD DE INGENIERIA ELECTRONICA
TUNJA
2015

DESARROLLO DE GUIAS DE TRABAJO PARA EL ÁREA DE AUTOMATIZACIÓN
INDUSTRIAL, HACIENDO USO DEL SOFTWARE FACTORY I/O CON LAS
TARJETAS USB-4704 DE ADVANTECH, ARDUINO UNO Y EL PLC SIEMENS S7-
1200

JUAN DAVID RODRIGUEZ FERRER

TRABAJO DE GRADO PARA OBTENER EL TITULO DE INGENIERO
ELECTRONICO

ASESOR

CARLOS ALBERTO CARDONA COY. INGENIERO ELECTRONICO

UNIVERSIDAD SANTO TOMAS DE TUNJA
FACULTAD DE INGENIERIA ELECTRONICA

TUNJA

2015

EXONERACION DE RESPONSABILIDADES

El autor de este trabajo de grado es el único responsable, por lo tanto la facultad de Ingeniería Electrónica de la Universidad Santo Tomas de Tunja, no se hacen responsables por comentarios, afirmaciones y consecuencias consumados en este.

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Tunja, 09 de Octubre de 2015

DEDICATORIA

A mi abuela Flor Helena Leguizamón que por medio de sacrificios, amor, comprensión y paciencia me dio la oportunidad de crecer como persona brindándome su mano en todo momento sin perder la esperanza en mí y siendo un gran ejemplo en mi vida, donde me enseñó que las cosas se logran con esfuerzo y que no importa las razones siempre hay que sonreír y creer.

A mi madre Gabby Ferrer Leguizamón que me brindo su sabiduría para afrontar todo problema y obstáculo en la vida, gracias por ser esa gran mujer que siempre me ha escuchado y me ha amado sobre todas las cosas, agradezco que me has enseñado a ser fuerte y sabio, sin eso mi camino no sería el mismo.

A mis hermanos Felipe Camacho y Tania Camacho por brindarme su amor y darme tanta felicidad y ser las personas con las que quiero compartir siempre.

A mi familia porque sin ellos no podría encontrar mi identidad y que cada uno de ellos apporto en mí el valor de la vida; agradezco a mi primo y hermano German David López por cada momento vivido y por darme todo su apoyo incondicional, enseñándome el camino a la felicidad en un mundo destruido por la avaricia.

AGRADECIMIENTO

Al ingeniero Carlos Alberto Cardona Coy por su apoyo en este proyecto y su compromiso para llevar a cabalidad el desarrollo del mismo, con la asesoría y búsqueda de soluciones que se plantearon en el desarrollo de este proyecto.

A mis maestros del Colegio Andino de Tunja y la universidad Santo Tomas de Tunja por brindarme su conocimiento y dedicación en todo mi desarrollo de aprendizaje.

TABLA DE CONTENIDO

LISTA DE FIGURAS	9
LISTA DE TABLAS	10
GLOSARIO	11
RESUMEN	12
PROLOGO	13
INTRODUCCION	14
JUSTIFICACION	15
PLANTEAMIENTO DEL PROBLEMA	16
OBJETIVOS	17
1. MARCO TEORICO	18
1.1. AUTOMATIZACION INDUSTRIAL	18
1.2. PROCESOS INDUSTRIALES	19
1.2.1. OPERACIONES CONTINUAS	19
1.2.2. OPERACIONES DISCONTINUAS	19
1.2.3. OPERACIONES POR LOTES	19
1.3. AUTOMATAS PROGRAMABLES	20
1.3.1. VARIABLES DEL AUTOMATA	20
1.3.1.1. VARIABLES EXTERNAS	20
1.3.1.1.1. VARIABLES TODO/NADA	20
1.3.1.1.2. VARIABLES ANÁLOGAS	20
1.3.1.2. VARIABLES INTERNAS	20
1.4. LENGUAJE DE PROGRAMACION	21
1.4.1. COMPUERTAS LOGICAS Y ALGEBRA BOOLEANA	21
1.4.1.1. DISEÑO DE CIRCUITOS LOGICOS COMBINATORIOS	22
1.4.2. LENGUAJE LADDER	23
1.4.2.1. RELES INTERNOS O MARCAS	23
1.4.2.2. TEMPORIZADORES	24
1.4.2.2.1. TIPOS DE TEMPORIZADORES	24
1.4.2.3. CONTADORES	24
1.4.3. GRAFCET	24
1.4.3.1. ETAPAS	25
1.4.3.2. CONDICIONES DE TRANSICION	25
1.4.3.3. REGLAS DE EVOLUCION	26

1.5. FACTORY I/O	26
1.6. CONNECT I/O	28
1.7. ADVANTECH USB-4704	29
1.7.1. ENTRADAS ANALOGAS	29
1.7.2. SALIDAS ANALOGAS	30
1.7.3. ENTRADAS DIGITALES	30
1.7.4. SALIDAS DIGITALES	30
1.7.5. ADVANTECH	30
1.8. ARDUINO	31
1.8.1. ARDUINO UNO	32
1.9. SIEMENS S7-1200	33
1.9.1. INTERFAZ PROFINET INTEGRADA	34
2. DISEÑO METODOLOGICO	35
3. RESULTADOS	37
3.1. FASE I: INSTALACIÓN DEL SOFTWARE FACTORY I/O	37
3.2. FASE II: ANÁLISIS DEL SOFTWARE FACTORY I/O	38
3.3. FASE III: INSTALACIÓN DEL SOFTWARE CONNECT I/O	38
3.4. FASE IV: ANÁLISIS DEL SOFTWARE CONNECT I/O	40
3.5. FASE V: INSTALACIÓN DRIVER TARJETA USB 4704	40
3.6. FASE VI: CONEXIÓN TARJETA USB 4704 Y ARDUINO UNO	41
3.7. FASE VII: COMUNICACIÓN (PLC) S7-1200 Y FACTORY I/O	43
3.8. FASE VIII: DESARROLLO DEL EJEMPLO 1	45
3.9. FASE IX: DESARROLLO DEL EJEMPLO 2	48
3.10. FASE X: DESARROLLO DEL EJEMPLO 3	52
3.11. FASE XI: DESARROLLO DEL EJEMPLO 4	56
4. CRONOGRAMA	61
5. PRESUPUESTO	62
6. CONCLUSIONES	63
7. RECOMENDACIONES	64
8. BIBLIOGRAFIA	65

LISTAS DE FIGURAS

FIGURA 1. CONTROL LAZO ABIERTO	19
FIGURA 2. CONTROL LAZO CERRADO	19
FIGURA 3. SISTEMA FORMADO POR EL PROCESO Y EL AUTÓMATA	20
FIGURA 4. TABLA DE VERDAD Y COMPUERTA OR.	21
FIGURA 5. TABLA DE VERDAD Y COMPUERTA AND.	21
FIGURA 6. TABLA DE VERDAD Y COMPUERTA OR.	21
FIGURA 7. TABLA DE VERDAD Y COMPUERTA NOR.	22
FIGURA 8. TABLA DE VERDAD Y COMPUERTA NAND.	22
FIGURA 9. SÍMBOLOS BÁSICOS DEL LENGUAJE LADDER	23
FIGURA 10. EJEMPLO LADDER.	23
FIGURA 11. RELÉ INTERNO.	24
FIGURA 12. ETAPAS.	25
FIGURA 13. CONDICION DE TRANSICION CT	26
FIGURA 14. ENTORNO FACTORY I/O	27
FIGURA 15. ENTORNO CONNECT I/O	28
FIGURA 16. TARJETA USB-4704	29
FIGURA 17. ARDUINO UNO	32
FIGURA 18. PARTES (PLC) S7-1200	34
FIGURA 19. DESCARGA DE FACTORY I/O	37
FIGURA 20. INSTALACION DE FACTORY I/O	37
FIGURA 21. RECONOCIMIENTO DE FACTORY I/O	38
FIGURA 22. DESCARGA DE CONNECT I/O	39
FIGURA 23. INSTALACION DE CONNECT I/O	39
FIGURA 24. RECONOCIMIENTO DE CONNECT I/O	40
FIGURA 25. TRAJETA USB 4704 Y DRIVERS	41
FIGURA 26. INSTALACION DE USB 4704	41
FIGURA 27. CONEXIÓN USB 4704 Y ARDUINO UNO	42
FIGURA 28. INVERTIR LAS SALIDAS EN FACTORY I/O	42
FIGURA 29. CONEXIÓN DEL PLC S7-1200	43
FIGURA 30. DIRECCIÓN IP DEL PLC S7-1200	43
FIGURA 31. CONFIGURACIÓN PLC S7-1200 EN FACTORY I/O	44
FIGURA 32. EJEMPLO DE TIA PORTAL ENTRADAS	44
FIGURA 33. EJEMPLO 1 FACTORY I/O	45
FIGURA 34. EJEMPLO 1 CONNECT I/O	45
FIGURA 35. EJEMPLO 1 TIA PORTAL	47
FIGURA 36. EJEMPLO 2 FACTORY I/O	48
FIGURA 37. EJEMPLO 2 CONNECT I/O	48
FIGURA 38. EJEMPLO 2 TIA PORTAL	50
FIGURA 39. EJEMPLO 3 FACTORY I/O	52
FIGURA 40. EJEMPLO 3 CONNECT I/O	52
FIGURA 41. EJEMPLO 3 TIA PORTAL	54

FIGURA 42. EJEMPLO 4 FACTORY I/O	56
FIGURA 43. EJEMPLO 4 CONNECT I/O	57
FIGURA 44. DIAGRAMA DE GANTT	62

LISTAS DE TABLAS

TABLA 1. EJEMPLO TABLA DE VERDAD	22
TABLA 2. RANGO DE ENTRADA	30
TABLA 3. CARACTERÍSTICAS ARDUINO UNO	33
TABLA 4. CARACTERÍSTICAS (PLC) S7-1200	34
TABLA 5. CRONOGRAMA DE ACTIVIDADES	61
TABLA 6. PRESUPUESTO	62

GLOSARIO

Automatización Aplicación de procedimientos automáticos en un proceso o industria.

Factory I/O Es un entorno de pruebas virtual de automatización en tiempo real, donde se puede construir y simular sistemas industriales en 3D de una forma fácil y rápida a través de un enfoque de arrastrar y soltar.

Arduino: Es una plataforma basada en una placa con un microcontrolador, su diseño facilita la creación de proyectos.

PLC: Controlador lógico programable, es una computadora utilizada en la ingeniería automática, el cual automatiza procesos electromecánicos.

S7-1200: Es un autómata o PLC, desarrollado por Siemens.

Simulador: Aparato que permite la reproducción de un sistema, produciendo sensaciones y experiencias similares a las reales.

Artificial: Es algo no natural, hecha con intención y propósito.

Procesos Industriales: Conjunto de operaciones unitarias necesarias para modificar las características de las materias primas.

Autómatas Programables: Es un sistema secuencial, programable diseñado para controlar procesos secuenciales.

Ethernet: Es un estándar de redes de área local IEEE 802.3.

RESUMEN

TITULO: DESARROLLO DE GUIAS DE TRABAJO PARA EL ÁREA DE AUTOMATIZACIÓN INDUSTRIAL, HACIENDO USO DEL SOFTWARE FACTORY I/O CON LAS TARJETAS USB-4704 DE ADVANTECH, ARDUINO UNO Y EL PLC SIEMENS S7-1200¹

AUTOR: JUAN DAVID RODRIGUEZ FERRER²

PALABRAS CLAVES: Automatización industrial, Autómatas programables, control, procesos industriales, guías de trabajo, Arduino UNO, Siemens S7-1200, USB-4704, Factory I/O, Connect I/O.

La automatización de sistemas industriales incluye como todo desarrollo metódico, una serie de pasos ordenados y/o organizados, su propósito es llegar a un resultado preciso, como la identificación, la detección y el transporte de uno o varios objetos, así como el diseño y control del mismo. A lo largo de la historia la automatización a avanzado significativamente, desde la época de la revolución industrial a la digital actual, dando así un gran paso a grandes inventos y descubrimientos para llegar a lo que es hoy en día la capacidad de controlar la producción con máquinas de control computarizado, es indispensable para muchas organizaciones automatizar los procesos, de tal forma que puedan controlarse, incidiendo directamente en el mejorando de la producción y la calidad, para esto es necesario establecer un diseño organizado en cada uno de los equipos y determinar con exactitud su función, para realizar las tareas eficientemente y que aseguren el éxito de la operación.

Estos procesos son realizados a través de acciones autónomas mediante programación, haciéndolo un proceso artificial, reduciendo al mínimo la participación humana, mejorando la productividad y la velocidad de ejecución de estas tareas repetitivas.

Los sistemas autómatas poseen tres partes fundamentales: Operativa, control y supervisión, los cuales hay que tener en cuenta al momento de diseñar la tarea y el objetivo del proceso.

Este proyecto tiene como propósito el desarrollo de nueve guías de trabajo, enfatizando el aprendizaje en sistemas autónomos industriales, con la herramienta Factory I/O, este es un software que simula en tiempo real sistemas industriales en 3D, teniendo interconexión con diferentes dispositivos, los cuales se utilizaran la tarjeta USB-4704, Arduino UNO y PLC Siemens S7-1200. Mediante el uso de esta herramienta se complementaran los conocimientos adquiridos durante el programa de ingeniería electrónica en la Universidad Santo Tomas de Tunja en el área de la automatización industrial.

¹ Proyecto de Grado

² Universidad Santo Tomas de Tunja, Facultad de Ingeniería Electrónica; Ing. Carlos Cardona, director de proyecto.

PROLOGO

Es de gran agrado para mí presentar esta monografía, cuyo autor no es únicamente un compañero de pregrado sino un excelente amigo. Considero importante resaltar que la elaboración del presente trabajo de grado, expone la dedicación de una profunda investigación, en la que el autor adquiere una gran responsabilidad para llegar a obtener el diseño apropiado de unas guías de aprendizaje relacionadas con el uso de la valiosa herramienta Factory I/O, utilizada en procesos de automatización industrial.

Es así como la facultad de Ingeniería Electrónica de la Universidad Santo Tomas de Tunja, le confía el diseño de las primeras guías, permitiéndole demostrar los conocimientos y principios éticos y morales adquiridos a lo largo de su vida. Cabe destacar que la labor de su familia y demás personas que han hecho parte de su proceso, son el reflejo de la esencia que caracteriza al autor.

El libro muestra información sobre Factory I/O, enfatizando en el aprendizaje hacia la simulación de sistemas autónomos industriales en 3D, haciendo uso de la comunicación con la tarjeta USB-4704, Arduino UNO y el PLC Siemens S7-1200. La información presentada aquí permite a estudiantes, ingenieros y demás personas interesadas en la herramienta, conocer y comprender el funcionamiento y usos que esta brinda.

Felicito al autor por el gran trabajo de investigación realizado y por su interés en compartir diferentes aspectos sobre Factory I/O.

Cindy Vanessa Vergara Tibocho
Ingeniera Electrónica

INTRODUCCIÓN

El control en procesos industriales, es un conjunto de métodos y procedimientos computarizados y electromecánicos, los cuales sustituyen un operario en tareas físicas previamente programadas. En la actualidad es indispensable para muchas organizaciones automatizar los procesos, de tal forma que puedan controlarse, incidiendo directamente en el mejorando de la producción y de la calidad en la consecución de un producto o la prestación de un servicio, para esto es necesario establecer un diseño organizado en cada uno de los equipos y determinar con exactitud su función, para realizar las tareas eficientemente y que aseguren el éxito de la operación.

Estos procesos son realizados a través de acciones autónomas mediante programación, haciéndolo un proceso artificial, reduciendo al mínimo la participación humana, mejorando la productividad y la velocidad de ejecución de los procesos repetitivos.

Los sistemas autómatas poseen tres partes fundamentales: Operativa, control y supervisión, los cuales hay que tener en cuenta al momento de diseñar la tarea y el objetivo del proceso. La parte operativa es la materialización de la actividad o meta deseada, el control es la parte lógica que logra la operatividad, y la supervisión la interfaz hombre-máquina.

Para lograr el control es necesario tener en cuenta componentes necesarios, como los sensores que convierten una señal de magnitud física en una señal eléctrica, así la parte de control tomara decisiones de acuerdo a estas señales y poner en funcionamiento los actuadores.

La facultad de ingeniería electrónica de la Universidad Santo Tomas, está fuertemente interesada en el aprendizaje de la automatización industrial, por ende esta investigación va encaminada a la utilización de una herramienta que acerque al estudiante en diseño y programación, por medio del simulador en tiempo real Factory I/O y así aprovechando sus interconexiones con diferentes procesadores gracias a la tarjeta USB 4704, que lo permite; en este proceso se demostrara la conexión de este con la tarjeta Arduino UNO, como también por medio de comunicación Ethernet con el autómata programable S7-1200 de Siemens.

En este trabajo de investigación se Elaboran guías de aprendizaje relacionando los conceptos y pasos necesarios en la utilización del software y los dispositivos mencionados anteriormente, ayudando al estudiante de automatización industrial en el aprendizaje del área.

JUSTIFICACIÓN

La automatización de sistemas industriales incluye como todo desarrollo metódico, una serie de pasos ordenados y/o organizados, su propósito es llegar a un resultado preciso, como la identificación, la detección y el transporte de uno o varios objetos, así como el diseño y control del mismo. Hoy en día el campo de la industria estará asociado a un equipo de control automático, y a una constante evolución, que implica un continuo aprendizaje en este campo.

Los procesos autónomos de productividad en las empresas, involucran la calidad, efectividad e innovación de la producción, comprometiéndose cada vez con la automatización industrial permitiendo incrementar la producción y reducir los costos. A lo largo de la historia la automatización a avanzado significativamente, desde la época de la revolución industrial a la digital actual, dando así un gran paso a grandes inventos y descubrimientos para llegar a lo que es hoy en día la capacidad de controlar la producción con máquinas de control computarizado.

Este proyecto tiene como propósito el desarrollo de guías de trabajo, enfatizando el aprendizaje en sistemas autónomos industriales, con la herramienta Factory I/O, este es un software que simula en tiempo real sistemas industriales en 3D, teniendo interconexión con diferentes dispositivos, los cuales se utilizaran la tarjeta USB-4704, Arduino UNO y PLC Siemens S7-1200. Mediante el uso de esta herramienta se complementaran los conocimientos adquiridos durante el programa de ingeniería electrónica en la Universidad Santo Tomas de Tunja en el área de la automatización industrial.

PLANTEAMIENTO DEL PROBLEMA

El propósito de este estudio exploratorio es desarrollar guías de trabajo encaminadas hacia el aprendizaje de la automatización industrial en estudiantes de ingeniería electrónica de la universidad Santo Tomas, con herramientas adecuadas, como los simuladores, ellos permiten acercarse a la realidad, permitiendo explorar y desarrollar la lógica de programación en sistemas autónomos, y generan habilidades necesarias en la formación del estudiante de ingeniería electrónica, mediante un enfoque académico, metodológico y sistemático, que permita solucionar problemas en la automatización de procesos industriales, mediante la implementación del software Factory I/O y la utilización del hardware USB-4704 de Advantech, Arduino y PLC Siemens S7-1200.

La necesidad de suplir tareas en el sector industrial, en procesos repetitivos, como el transporte y la selección de elementos de diseño en sistemas autónomos, permiten incrementar la efectividad de la industria, reduciendo los tiempo de producción, y obteniendo productos y/o servicios de mayor calidad; además de disminuir el riesgo humano en la participación de la producción, donde los sistemas autónomos realizaran las tareas más difíciles.

Por esta razón es importante que el ingeniero electrónico tenga los conocimientos necesarios en control y automatización para suplir las necesidades en estos sistemas, como el diseño e implementación de estos, así este trabajo brindara al estudiante un aprendizaje con el software Factory I/O.

Debido a que el proceso de aprendizaje es realizado en un simulador (Factory I/O), hay que tener en cuenta que en un ámbito real, existen factores físicos que influyen en el funcionamiento de actuadores y sensores.

Con la utilización de este software se simulan panoramas reales de producción, identificando dificultades y ventajas, por esta razón es indispensable preguntarse, ¿Es necesario el aprendizaje en situaciones simuladas? ¿La biblioteca de piezas del Factory I/O que limitantes tiene respecto a los reales? Sabiendo esto se formula la siguiente pregunta, ¿Qué hay que tener en cuenta en las guías para la retención del conocimiento?

OBJETIVO GENERAL

- Diseñar nueve guías de trabajo con un enfoque académico, metodológico y sistemático, encaminadas hacia el aprendizaje del estudiante de ingeniería electrónica en el área de automatización industrial a través del manejo del software Factory I/O como también la tarjeta USB-4704, Arduino y PLC Siemens S7-1200.

OBJETIVOS ESPECÍFICOS

- Verificar el funcionamiento y configuración del software Factory I/O y Connect I/O, como también su interconexión entre las tarjetas USB 4704 y Arduino UNO, y el PLC siemens S7-1200.
- Identificar y aplicar conceptos básicos de ingeniería electrónica para el desarrollo y ejecución del proyecto, que hagan parte de sistemas de control y automatización industrial.
- Implementar estrategias en los lenguajes de programación utilizados, aprovechando las herramientas necesarias para el entendimiento en la lógica.
- Diseñar guías de trabajo, con un enfoque metodológico el cual se evidencie los pasos necesarios para el entendimiento del software.

1. MARCO TEORTICO

1.1. AUTOMATIZACIÓN INDUSTRIAL

Desde el inicio de la era industrial hasta la actualidad, la automatización se ha convertido en una herramienta esencial para competir en un mercado globalizado, permitiendo reducir los tiempos de producción, realizar tareas complejas, mayor rentabilidad y aumento en la calidad.

La real academia de ciencias exactas físicas y naturales define la automática como el estudio de los métodos y procedimientos cuya finalidad es la sustitución del operador humano por un ordenador artificial en la generación de una tarea física o mental previamente programada³.

En todo proceso industrial son muchas las variables a controlar, siendo muy difícil monitorear de forma constante por el ser humano; se vio necesario la creación de equipos, capaces de procesar y memorizar variables físicas. Un proceso industrial se define como un conjunto de acciones, realizadas por una o más maquinas coordinadas que dan como resultado la fabricación de un producto⁴.

La mayoría de los procesos industriales pueden automatizarse, este proceso puede dividirse en dos partes: una operativa, la cual comprende las acciones que realizan determinados elementos sobre el proceso, así como motores, válvulas, actuadores, etc. Otra parte es el control, el cual programa las secuencias de la parte operativa con ayuda de la lectura de sensores, y la utilización de autómatas programables, los cuales son sistemas basados en un microcontrolador.

Existen dos formas básicas de realizar el control de un proceso industrial las cuales están definidas como control en lazo abierto y cerrado (ver gráfico 1 y 2).

³ Romero, j., Lorite, J., & Montoro, S. (1994). *Automatizacion problemas resuletos con autómatas Programables*. Paraninfo sa.

⁴(s.f.).[http://www.marcombo.com/Descargas/9788426715753/SENSORES/TEMAS/SA%20Tema%2016%20Interfaces%20\(1\).pdf](http://www.marcombo.com/Descargas/9788426715753/SENSORES/TEMAS/SA%20Tema%2016%20Interfaces%20(1).pdf)

Figura 1. Control lazo abierto



Basado Problemas Resueltos con autómatas programables. Romera. J. Pedro

Figura 2. Control lazo cerrado



Basado Problemas Resueltos con autómatas programables. Romera. J. Pedro

1.2. PROCESOS INDUSTRIALES:

1.2.1. Operaciones Continuas:

Las operaciones continuas se caracterizan por que la materia prima es procesada de manera continua, dando un producto final constante.

Este proceso se realiza por periodos de tiempo largo, donde la variable empleada en el proceso y sistema de control es de tipo analogico¹.

1.2.2. Operaciones discontinuas:

El producto de salida se obtiene a través de una serie de operaciones, muchas de ellas con gran similitud entre sí. La materia prima sobre la que se trabaja es habitualmente un elemento discreto que se trabaja de forma individual¹.

1.2.3. Operaciones por lotes:

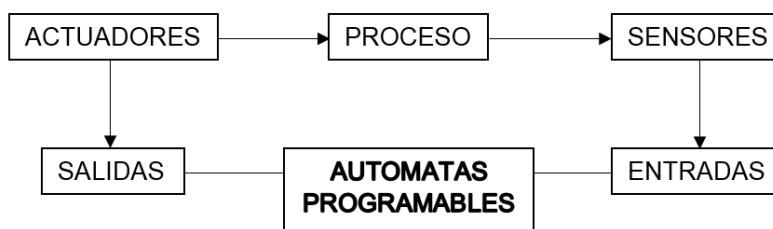
Este es un proceso discontinuo, su diferencia es que requiere de diferentes piezas discretas para realizar el proceso siguiendo una secuencia especificada¹.

1.3. AUTÓMATA PROGRAMABLE:

El autómata programable es la unidad de control, basado en un microcontrolador, que posee una configuración modular, diseñado para controlar en tiempo real procesos que presentan una evolución secuencial, incluye todas las interfaces con las señales de proceso, el cual es un sistema con capacidad de conexión directa a las señales de campo y programable para el usuario¹.

Al conjunto de señales de consigna y de realimentación que entran en el autómata se les denomina genéricamente entradas y al conjunto de señales de control obtenidas se denominan salidas⁵.

Figura 3. Sistema formado por el proceso y el autómata



Basado Problemas Resueltos con autómatas programables. Romera. J. Pedro

1.3.1. VARIABLES DEL AUTÓMATA:

1.3.1.1. Variables externas:

1.3.1.1.1. **Variables todo/nada:** los autómatas programables trabajan con lógica positiva, es decir la señal entregada será un 1 o un 0.

1.3.1.1.2. **Variables análogas:** son variaciones de tensión entregadas al autómata suministrado por un variador.

1.3.1.2. **Variables internas:** estas variables tienen asignada su propia zona de memoria que puede ser direccionada en función del dato almacenado, el cual guarda resultados parciales de operaciones lógicas o aritméticas que se producen en el programa de usuario¹.

⁵Balcells, J., & Romeral, J. L. (s.f.). *Autómatas Programables*. marcombo.

1.4. LENGUAJE DE PROGRAMACIÓN:

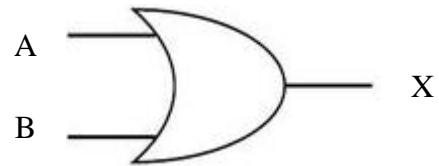
El lenguaje de programación es una estructura la cual da instrucciones detalladas a un programa de computadora, esto se le conoce como algoritmo, a un conjunto de instrucciones o reglas definidas, que permiten realizar una acción mediante pasos ordenados sucesivos.

1.4.1. COMPUERTAS LÓGICAS Y ALGEBRA BOOLEANA:

El álgebra booleana nos permite analizar y diseñar sistemas digitales, por esta razón es necesario el estudio de compuertas lógicas, los cuales son los circuitos lógicos más fundamentales, y observamos cómo puede describirse su operación mediante el uso del álgebra booleana⁶.

Figura 4. Tabla de verdad y compuerta or.

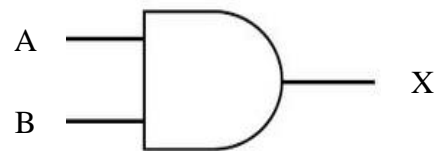
A	B	$X=A+B$
0	0	0
0	1	1
1	0	1
1	1	1



Basado Sistemas Digitales Principios y aplicaciones. Tocci, R.

Figura 5. Tabla de verdad y compuerta and.

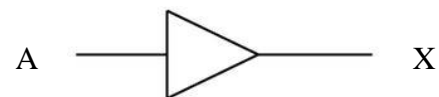
A	B	$X=AB$
0	0	0
0	1	0
1	0	0
1	1	1



Basado Sistemas Digitales Principios y aplicaciones. Tocci, R.

Figura 6. Tabla de verdad y compuerta Not.

A	$X=A$
0	1
1	0

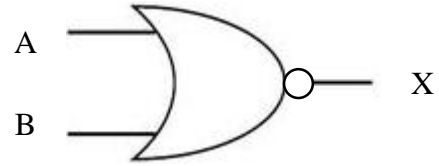


Basado Sistemas Digitales Principios y aplicaciones. Tocci, R.

⁶Tocci, R., Widmer, N., & Moss, G. (s.f.). *Sistemas Digitales Principios y Aplicaciones*. Prentice Hall.

Figura 7. Tabla de verdad y compuerta Nor.

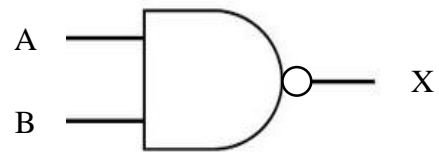
A	B	$X=A+B$
0	0	1
0	1	0
1	0	0
1	1	0



Basado Sistemas Digitales Principios y aplicaciones. Tocci, R.

Figura 8. Tabla de verdad y compuerta Nand.

A	B	$X=AB$
0	0	1
0	1	1
1	0	1
1	1	0



Basado Sistemas Digitales Principios y aplicaciones. Tocci, R.

1.4.1.1. DISEÑO DE CIRCUITOS LÓGICOS COMBINATORIOS:

Cuando se da el nivel de salida deseado de un circuito lógico en todas las posibles condiciones de entrada, los resultados se pueden mostrar adecuadamente en una tabla de verdad.

Tabla 1. Ejemplo tabla de verdad

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Fuente: Sistemas Digitales Principios y aplicaciones. Tocci, R.

Seleccionar las salidas donde sea 1 y escribir la expresión en términos de AND, teniendo en cuenta que las variables de entrada si es 0 serán invertida y si es 0 será no invertida.

$$\bar{A}BC, A\bar{B}C, AB\bar{C}, ABC$$

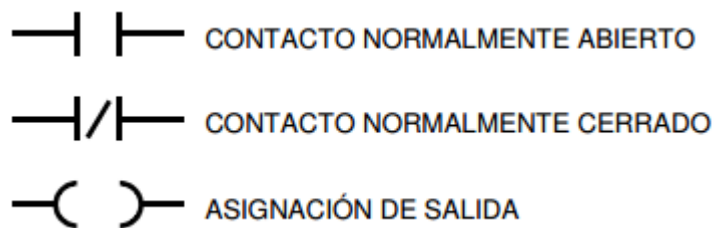
Se escribe la expresión como suma de productos.

$$x = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

1.4.2. LENGUAJE LADDER:

Este lenguaje gráfico, derivado del lenguaje de relés, es utilizado mediante símbolos que representan contactos y boninas, es muy utilizado en autómatas programables basado en esquemas eléctricos de control clásicos, es un lenguaje estandarizado IEC 61131-3 (norma por la comisión electrotécnica internacional, estandariza los autómatas programables).⁵

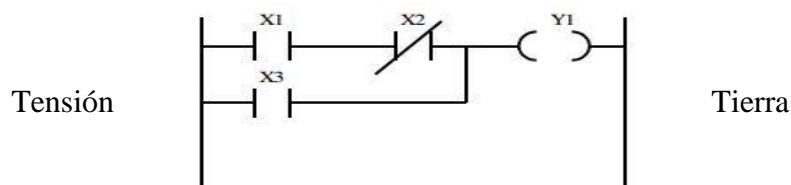
Figura 9. Símbolos básicos del lenguaje ladder



Fuente: <http://www.ing.unlp.edu.ar/electrotecnia/procesos/apuntes/Diagrama%20Escalera.pdf>

Estos diagramas tienen líneas verticales, la izquierda representa un conductor con tensión, y la línea vertical a la derecha tierra.⁵

Figura 10. Ejemplo ladder.

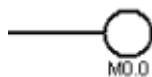


Fuente: <http://www.ing.unlp.edu.ar/electrotecnia/procesos/apuntes/Diagrama%20Escalera.pdf>

1.4.2.1. RELÉS INTERNOS O MARCAS:

Son variables lógicas que se pueden usar para memorizar estados o acumuladores de resultados que se utilizan posteriormente en el programa, se les identifica con la letra “M”⁵

Figura 11. Relé interno.



Fuente: <http://www.ing.unlp.edu.ar/electrotecnia/procesos/apuntes/Diagrama%20Escalera.pdf>

1.4.2.2. TEMPORIZADORES:

Al alcanzar un valor de tiempo activa un contacto interno denominado PRESET, es necesario dar indicación una entrada de inicio denominada START, al contactar en su entrada una condición de arranque.⁵

1.4.2.2.1. TIPOS DE TEMPORIZADOR:

SE- Con retardo a la conexión

SS- Con retardo a la conexión activado por impulso en START.

SI- Mientras se mantenga conectado START, la salida se activara durante un tiempo.

SV- Mantiene la salida activa durante un tiempo.

1.4.2.3. CONTADORES:

Definidos como posiciones de memoria que almacenan un valor numérico, mismo que se incrementa o decremento según su configuración dada a dicho contador.⁷

1.4.3. GRAFCET (GRÁFICO DE MANDO ETAPA TRANSICIÓN):

Es un diagrama funcional que describe la evolución del proceso que se pretende automatizar, indicando la acción que hay que realizar sobre el proceso y que informaciones las provocan; partiendo de él se pueden obtener las secuencias que ha de realizar el autómata programable, su empleo para resolver tareas de automatización facilita el dialogo entre personas con niveles de formación técnicas diferente, tanto en el momento del análisis del proceso a automatizar, como posteriormente en el mantenimiento y reparación de averías.¹

⁷ <http://www.ing.unlp.edu.ar/electrotecnia/procesos/apuntes/Diagrama%20Escalera.pdf>

1.4.3.1. ETAPAS:

Para representar la evolución de un proceso con GRAFCET, se considera que el proceso a automatizar y el autómata que se emplea como controlador forma un solo sistema; el nexo de unión entre las actuaciones que hay que hacer sobre el proceso y el programa de usuario, cargado en el autómata, que da origen a aquellas es la etapa.

Por tanto, la representación gráfica de la evolución de un proceso con GRAFCET estará formada por una serie de etapas, y cada una de ellas llevara asociada una o varias acciones a realizar sobre el proceso.¹

Figura 12. Etapas.

REPRESENTACION DE ETAPAS:



REPRESENTACION DE ETAPAS INICIALES:



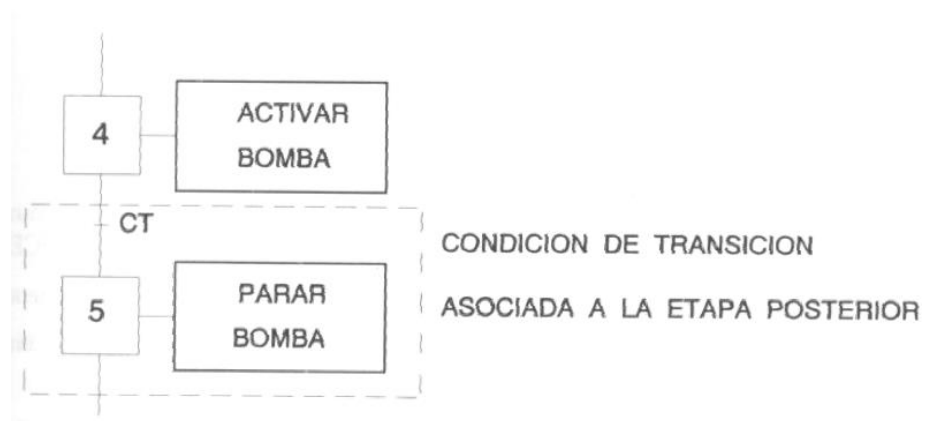
Fuente: Romero, j., Lorite, J., & Montoro, S. (1994). Automatizacion problemas resuletos con autómatas Programables. Paraninfo sa.

1.4.3.2. CONDICIONES DE TRANSICIÓN:

Un proceso secuencial se caracteriza porque una acción determinada se realiza en función del resultado de la acción anterior.¹

En GRAFCET, el proceso se descompone en una serie de etapas que son activadas una tras otra. Por tanto, tendra que existir una condición que se ha de cumplir para pasar de una a otra etapa.¹

Figura 13. Condicion de transicion ct



Fuente: Romero, j., Lorite, J., & Montoro, S. (1994). Automatizacion problemas resuletos con autómatas Programables. Paraninfo sa.

1.4.3.3. REGLAS DE EVOLUCIÓN:

2. El proceso se descompone en etapas, que serán activadas de forma secuencial.
3. Una o varias acciones se asocian a cada etapa. Estas acciones solo están activas cuando la etapa este activa.
4. Una etapa se hace activa cuando la precedente lo está y la condición de transición entre ambas etapas ha sido activada.
5. La activación de una condición de transición implica la activación de la etapa siguiente y la desactivación de precedente.
6. La etapa inicial E0 tiene que ser activada antes de que se inicie el ciclo GRAFCET, un ciclo está formado por todas las etapas posteriores a la etapa inicial.¹

1.5. FACTORY I/O

Es un entorno de pruebas virtual de automatización en tiempo real, donde se puede construir y simular sistemas industriales en 3D de una forma fácil y rápida a través de un enfoque de arrastrar y soltar.⁸

Contiene una biblioteca de piezas completamente funcionales que representan equipos industriales, contiene escenarios predeterminados los cuales son muy comunes en plantas

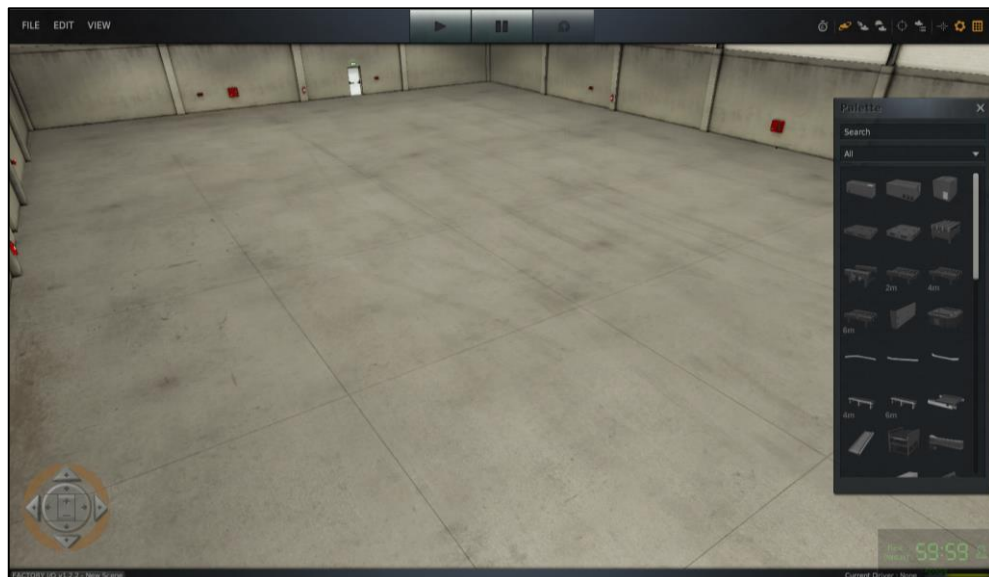
⁸ http://www.realgames.pt/wiki/FACTORY_I/O

industriales, estas son totalmente editables. Podemos interconectar diferentes tecnologías de automatización más comunes gracias a sus drivers, estos son:

- Open (SDK)
- Advantech USB-4704
- Advantech USB-4750
- Automgen TCP/IP Server
- MHJ
- Modbus TCP/IP Client
- Modbus TCP/IP Server
- Siemens LOGO! 0BA7 TCP/IP
- Siemens LOGO! 0BA8 TCP/IP
- Siemens S7-300 TCP/IP
- Siemens S7-400 TCP/IP
- Siemens S7-1200 TCP/IP
- Siemens S7-1500 TCP/IP
- Siemens S7-PLCSIM

Este es diseñado y distribuido por la empresa *real games*; han desarrollado más plataformas virtuales, útiles en la automatización como lo es: *Home I/O* el cual presenta la simulación en domótica en tiempo real interactuando en primera persona, *ITS PLC* posee una simulación virtual de sistemas industriales incluyendo sensores y actuadores virtuales.

Figura 14. Entorno FACTORY i/o



Fuente: Autor

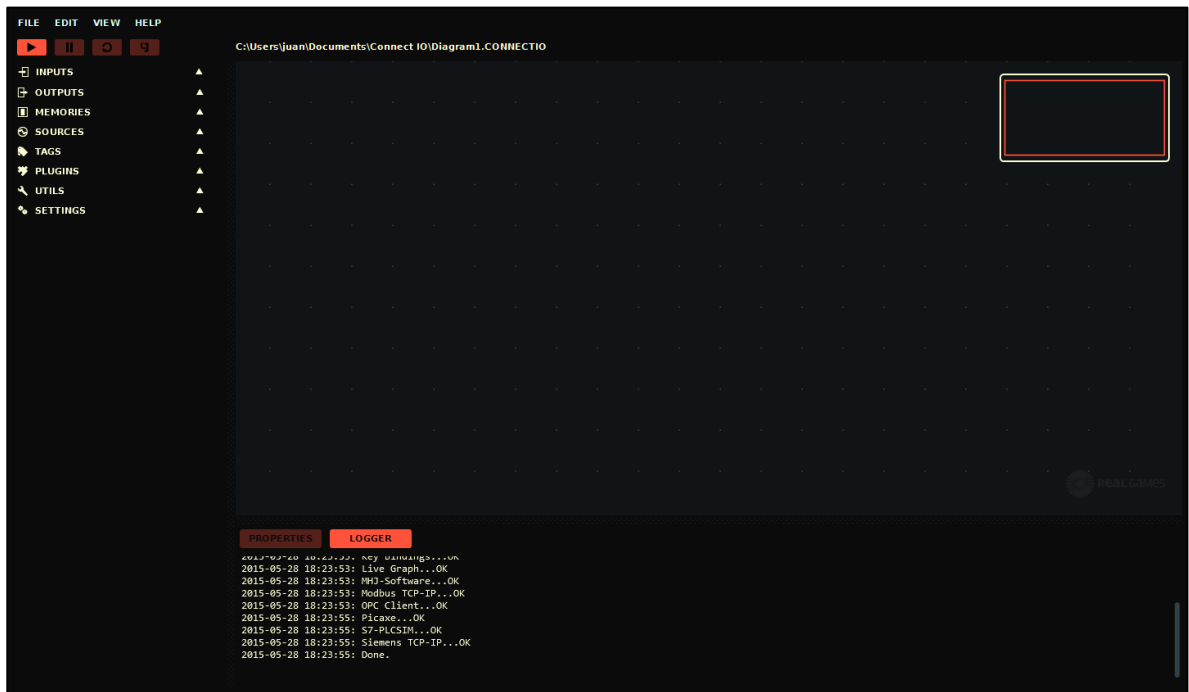
1.6. CONNECT I/O

Es una herramienta de programación visual que proporciona una interfaz intuitiva, en el que un diagrama se dibuja con nodos y enlaces, posee una interconexión directa con el software Factory I/O. Este es diseñado y distribuido por la empresa *real games*.⁹

Su programación está basada en los controladores lógicos programables y en lógica de circuitos combinatorios, posee las herramientas necesarias, como lo es compuertas lógicas, temporizadores, contadores, bits de memoria, etc.

Tiene una conexión directa con el software Factory I/O permitiendo el desarrollo y programación de proyectos.

Figura 15. Entorno CONNECT i/o



Fuente: Autor

⁹http://www.realgames.pt/wiki/CONNECT_I/O

1.7. ADVANTECH USB-4704:

Es un módulo de adquisición de datos, el cual se puede utilizar como una interfaz entre *Factory I/O* y microcontroladores, PLC, etc.

Figura 16. Tarjeta usb-4704



Fuente: http://www2.advantech.com/products/1-2MLKNO/USB-4704/mod_4D0800CC-F6FD-402A-9782-24CD0FFDAF42.aspx

Es un módulo de adquisición de datos por USB de bajo costo, fácil y eficiente en su manejo, posee una alimentación de energía recibida del puerto USB, cuenta con las siguientes características:

- Soporta USB 2.0
- Portátil
- Sin fuente de alimentación
- 8 canales de entrada analógicos
- 14 bits de resolución
- Frecuencia de muestreo de hasta 48 Ks/s
- 8 canales de entrada digitales
- 8 canales de salida digitales
- Contador de 32 bits

1.7.1. ENTRADAS ANÁLOGAS:

- protección contra sobretensiones de 30 Vp-p
- impedancia de entrada 127 K Ω

Tabla 2. Rango de entrada

Single Ended		±10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Differential		N/A	±1	±1.25	±2	±2.5	±4	±5	±10	±20
Absolute Accuracy (% of FSR)*	USB-4702	0.2	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
	USB-4704	0.15	0.1	0.1	0.1	0.1	0.1	0.1	0.15	0.15

Fuente: Datasheet Usb-4702 / Usb-4704

1.7.2. SALIDAS ANÁLOGAS:

- 2 canales
- Resolución 12 bits
- Rango de 0 a 5 V
- Capacidad de corriente 5mA
- Impedancia de salida 51 Ω

1.7.3. ENTRADAS DIGITALES:

- 8 canales
- Compatibilidad 3.3 V/5 V/TTL

1.7.4. SALIDAS DIGITALES:

- 8 canales
- Compatibilidad TTL

1.7.5. ADVANTECH:

Fundada en 1983, Advantech es líder en el suministro de productos innovadores, integrados, productos de automatización y soluciones. Advantech ofrece una integración completa del sistema, hardware, software, servicios de diseño centrada en el cliente. Advantech siempre ha sido un innovador en el desarrollo y fabricación de alta calidad, plataformas de computación de alto rendimiento.¹⁰

¹⁰ <http://www.advantech.com/about/missionandfocus>

1.8. ARDUINO:

Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia). En ese tiempo, los estudiantes usaban el microcontrolador BASIC Stamp, cuyo coste era de 100 dólares, lo que se consideraba demasiado costoso para ellos. Por aquella época, uno de los fundadores de Arduino, Massimo Banzi, daba clases en Ivrea.¹¹

El nombre del proyecto viene del nombre del *Bar di Re Arduino* (Bar del Rey Arduino) donde Massimo Banzi pasaba algunas horas. En su creación, contribuyó el estudiante colombiano Hernando Barragán, quien desarrolló la tarjeta electrónica Wiring, el lenguaje de programación y la plataforma de desarrollo. Una vez concluida dicha plataforma, los investigadores trabajaron para hacerlo más ligero, más económico y disponible para la comunidad de código abierto (hardware y código abierto). El instituto finalmente cerró sus puertas, así que los investigadores, entre ellos el español David Cuartielles, promovieron la idea. Banzi afirmaría años más tarde, que el proyecto nunca surgió como una idea de negocio, sino como una necesidad de subsistir ante el inminente cierre del Instituto de diseño Interactivo IVREA. Es decir, que al crear un producto de hardware abierto, éste no podría ser embargado.⁹

Posteriormente, Google colaboró en el desarrollo del Kit Android ADK (Accessory Development Kit), una placa Arduino capaz de comunicarse directamente con teléfonos móviles inteligentes bajo el sistema operativo Android para que el teléfono controle luces, motores y sensores conectados de Arduino.⁹

Para la producción en serie de la primera versión se tomó en cuenta que el coste no fuera mayor de 30 euros, que fuera ensamblado en una placa de color azul, debía ser Plug and Play y que trabajara con todas las plataformas informáticas tales como MacOSX, Windows y GNU/Linux. Las primeras 300 unidades se las dieron a los alumnos del Instituto IVREA, con el fin de que las probaran y empezaran a diseñar sus primeros prototipos.⁹

En el año 2005, se incorporó al equipo el profesor Tom Igoe, que había trabajado en computación física, después de que se enterara del mismo a través de Internet. Él ofreció su apoyo para desarrollar el proyecto a gran escala y hacer los contactos para distribuir las tarjetas en territorio estadounidense. En la feria Maker Fair de 2011 se presentó la primera placa Arduino 32 bit para trabajar tareas más pesadas.⁹

Los modelos en venta de Arduino se categorizan en 4 diferentes productos: placas, escudos, kits y accesorios.

¹¹ <http://kaiwatechnology.com.ar/tag/que-es/>

1.8.1. ARDUINO UNO:

Posee un microcontrolador de Atmel, dispone de 14 pines que pueden configurarse como entrada o salida y a los que facilita la conexión con cualquier otro dispositivo, recibe señales digitales de 0 a 5 v, dispone de entradas y salidas análogas las cuales pueden ser utilizadas en forma de señales PWM.

Un oscilador de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP (In-Circuit Serial Programming), y un botón de reinicio, contiene todo lo necesario para apoyar el microcontrolador.

Posee una memoria de 32KB con 0.5 KB para el arranque, también 2 KB de SRAM y 1 KB de EEPROM.¹²

Figura 17. Arduino uno



Fuente: <https://www.arduino.cc/en/Main/arduinoBoardUno>

Tiene 14 pines los cuales pueden configurarse como entradas o salidas, funcionan a 5v y pueden suministrar hasta 40mA, cada pin posee una resistencia pull-up interna entre 20 k Ω y 50 k Ω .¹⁰

¹² <http://www3.gobiernodecanarias.org/medusa/ecoblog/ralvgon/files/2013/05/Caracter%C3%ADsticas-Arduino.pdf>

Tabla 3. Características Arduino UNO

Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Limite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Fuente:

<http://www3.gobiernodecanarias.org/medusa/ecoblog/ralvgon/files/2013/05/Caracter%C3%ADsticas-Arduino.pdf>

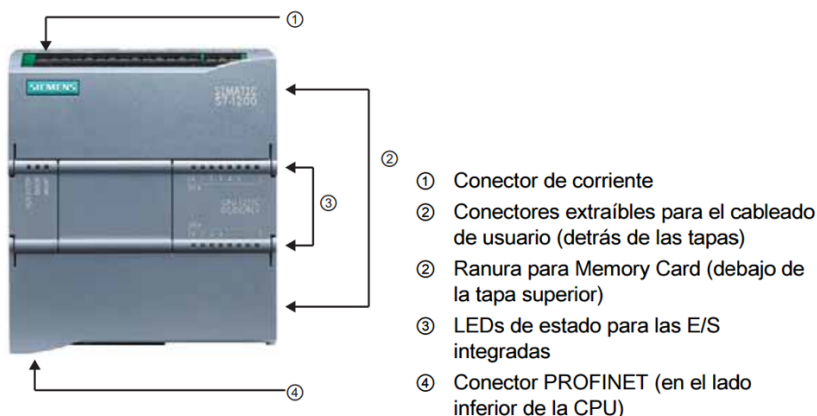
1.9. SIEMENS S7-1200:

La gama S7-1200 abarca distintos controladores lógicos programables (PLCs) que pueden utilizarse para numerosas tareas. Gracias a su diseño compacto, bajo costo y amplio juego de instrucciones, los PLCs S7-1200 son idóneos para controlar una gran variedad de aplicaciones. Los modelos S7-1200 y el software de programación basado en Windows ofrecen la flexibilidad necesaria para solucionar las tareas de automatización.

El controlador lógico programable (PLC) S7-1200 ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización. Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones. La CPU incorpora un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y salida en una carcasa compacta, conformando así un potente PLC. Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de conteo y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.¹³

¹³ <http://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S71200-MANUAL%20DEL%20SISTEMA.PDF>

Figura 18. Partes (PLC) S7-1200



Fuente: <http://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S71200-MANUAL%20DEL%20SISTEMA.PDF>

Tabla 4. Características (PLC) S7-1200

Función	CPU 1211C	CPU 1212C	CPU 1214C
Dimensiones físicas (mm)	90 x 100 x 75		110 x 100 x 75
Memoria de usuario	<ul style="list-style-type: none"> • Memoria de trabajo • Memoria de carga • Memoria remanente 		<ul style="list-style-type: none"> • 50 KB • 2 MB • 2 KB
E/S integradas locales	<ul style="list-style-type: none"> • Digitales • Analógicas 	<ul style="list-style-type: none"> • 6 entradas/4 salidas • 2 entradas 	<ul style="list-style-type: none"> • 8 entradas/6 salidas • 2 entradas
Tamaño de la memoria imagen de proceso	1024 bytes para entradas (I) y 1024 bytes para salidas (Q)		
Área de marcas (M)	4096 bytes		8192 bytes
Ampliación con módulos de señales	Ninguna	2	8
Signal Board	1		
Módulos de comunicación	3 (ampliación en el lado izquierdo)		
Contadores rápidos	3	4	6
• Fase simple	• 3 a 100 kHz	• 3 a 100 kHz 1 a 30 kHz	• 3 a 100 kHz 3 a 30 kHz
• Fase en cuadratura	• 3 a 80 kHz	• 3 a 80 kHz 1 a 20 kHz	• 3 a 80 kHz 3 a 20 kHz
Salidas de impulsos	2		
Memory Card	SIMATIC Memory Card (opcional)		
Tiempo de respaldo del reloj de tiempo real	Típico: 10 días / Mínimo: 6 días a 40 °C		
PROFINET	1 puerto de comunicación Ethernet		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Velocidad de ejecución booleana	0,1 µs/instrucción		

Fuente: <http://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S71200-MANUAL%20DEL%20SISTEMA.PDF>

1.9.1. INTERFAZ PROFINET INTEGRADA:

S7-1200 dispone de una interfaz PROFINET que garantiza una comunicación con el sistema, permitiendo la programación y comunicación, y de equipos de otros fabricantes mediante un protocolo abierto de Ethernet.¹⁴

¹⁴<http://www.catedu.es/elechomon/s71200/CARACTERISTICAS%20Y%20MONTAJE%20DEL%20SIMULADOR%20S7-1200.pdf>

2. DISEÑO METODOLOGICO

Este trabajo está dirigido a la comunidad estudiantil de la Universidad Santo Tomas de Tunja, comprende el desarrollo de guías de trabajo, en el área de Automatización industrial, enfocado en la utilización del software Factory I/O, como también la interconexión de dispositivos compatibles a este, los cuales se utilizaran la tarjeta USB 4704 con Arduino uno, y el autómatas siemens S7-1200, de esta manera se desarrollaran una serie de ejemplos, encaminados hacia la comunidad estudiantil.

De acuerdo al desarrollo de este trabajo se presenta una investigación cualitativa, ya que se consideran técnicas como observación y en base al diseño lingüístico, estudiando las cualidades y características del software y el hardware a utilizar, con un enfoque pedagógico con el objeto de mejorar la calidad del aprendizaje.

Este proyecto es desarrollado de acuerdo a las siguientes fases.

Fase I: Instalación del software Factory I/O

De acuerdo al sitio web de real games (<http://www.realgames.pt/download/>, s.f.) Se descargara el software Factory I/O, siguiendo los pasos de instalación y teniendo en cuenta los drivers que este requiere.

Fase II: Análisis del software Factory I/O

Se comprueba la funcionalidad y estudio a partir de los manuales de usuario (http://www.realgames.pt/downloads/factoryio/software_files/factoryio_user_guide_en.pdf, s.f.) (http://www.realgames.pt/downloads/factoryio/software_files/factoryio_parts_essentials.pdf, s.f.) Explorando sus herramientas de trabajo, así como cada elemento en el espacio de trabajo del software.

Fase III: Instalación del software Connect I/O

De acuerdo al sitio web de real games (<http://www.realgames.pt/download/>, s.f.) Se descargara el software Connect I/O, siguiendo los pasos de instalación

Fase IV: Análisis del software Connect I/O

Se comprueba la funcionalidad y estudio a partir de los manuales de usuario (http://www.realgames.pt/downloads/connectio/software_files/connectio_user_guide_en.pdf, s.f.) Explorando sus herramientas de trabajo, teniendo presente el tipo de lenguaje que se utiliza, así como la comunicación con Factory I/O.

Fase V: Instalación driver tarjeta USB 4704

Se utilizara el CD de instalación de la tarjeta USB 4704, el cual el software Factory I/O la detectara, se comprobara la comunicación para asegurarse de que la instalación se allá realizado correctamente.

Fase VI: Conexión tarjeta USB 4704 y arduino uno

Se realizara la conexión y comunicación entre las tarjetas, comprobando su funcionamiento con el software Factory I/O.

Fase VII: Comunicación (PLC) S7-1200 y Factory I/O.

Se realizara la comunicación y configuración entre (PLC) S7-1200 y Factory I/O, por medio del estándar de red de área local (ETHERNET), comprobando su funcionamiento.

Fase VIII: Desarrollo del ejemplo 1 Buffer Station

En esta fase se desarrollara el ejemplo predeterminado de Factory I/O Buffer Station, implementando la tarjeta USB 4704 con arduino UNO y (PLC) S7-1200

Fase IX: Desarrollo del ejemplo 2 Sorting Station

En esta fase se desarrollara el ejemplo predeterminado de Factory I/O Sorting Station, implementando la tarjeta USB 4704 con arduino UNO y (PLC) S7-1200

Fase X: Desarrollo del ejemplo 3 Converge Station

En esta fase se desarrollara el ejemplo predeterminado de Factory I/O Converge Station, implementando la tarjeta USB 4704 con arduino UNO y (PLC) S7-1200

Fase XI: Desarrollo del ejemplo 4 Pick & Place

En esta fase se desarrollara el ejemplo predeterminado de Factory I/O Pick & Place, implementando la tarjeta USB 4704 con arduino UNO

Fase XII: Realización de guías de trabajo

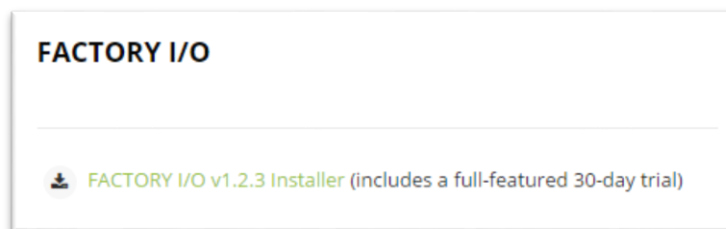
Se procede a la elaboración de nueve guías, las cuales comprenderán desde la instalación, tutoriales de manejo de los softwares, interconexiones software y hardware y la elaboración de cuatro ejemplos paso a paso.

3. RESULTADOS

3.1. FASE I: INSTALACIÓN DEL SOFTWARE FACTORY I/O (Anexo I)

En la página web de Real Games (<http://www.realgames.pt/download/>, s.f.) Tenemos acceso a la descarga del software.

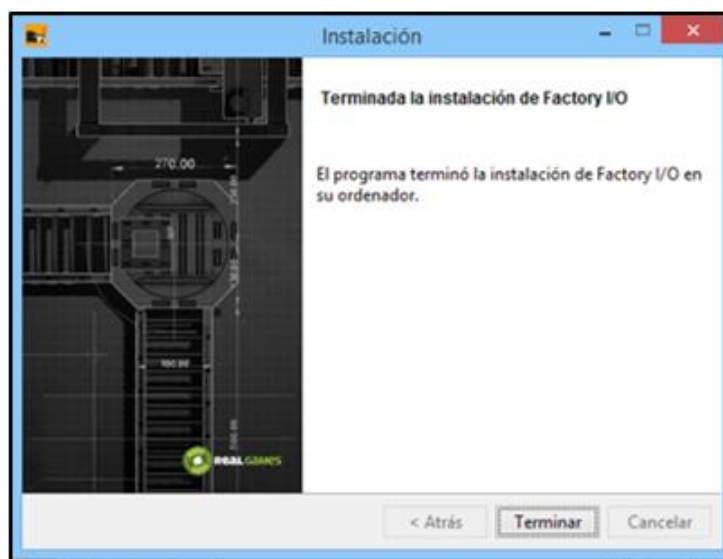
Figura 19. Descarga de factory i/o



Fuente: Autor

Previo a la instalación se siguen los pasos que sugiere este proceso.

Figura 20. Instalación de FACTORY I/O



Fuente: Autor

3.2. FASE II: ANÁLISIS DEL SOFTWARE FACTORY I/O (Anexo II)

Reconocer el espacio de trabajo con sus herramientas.

Figura 21. Reconocimiento de FACTORY I/O



Fuente: Autor

- **A:** Barra de herramientas.
- **B:** Modo Run y Edit.
- **C:** Tipo de cámara, tags, palette.
- **D:** Palette.
- **E:** Navegación.

3.3. FASE III: INSTALACIÓN DEL SOFTWARE CONNECT I/O (Anexo I)

En la página web de Real Games (<http://www.realgames.pt/download/>, s.f.) Tenemos acceso a la descarga del software.

Figura 22. Descarga de CONNECT I/O



Fuente: Autor

Previo a la instalación se siguen los pasos que sugiere este proceso.

Figura 23. Instalación de CONNECT I/O

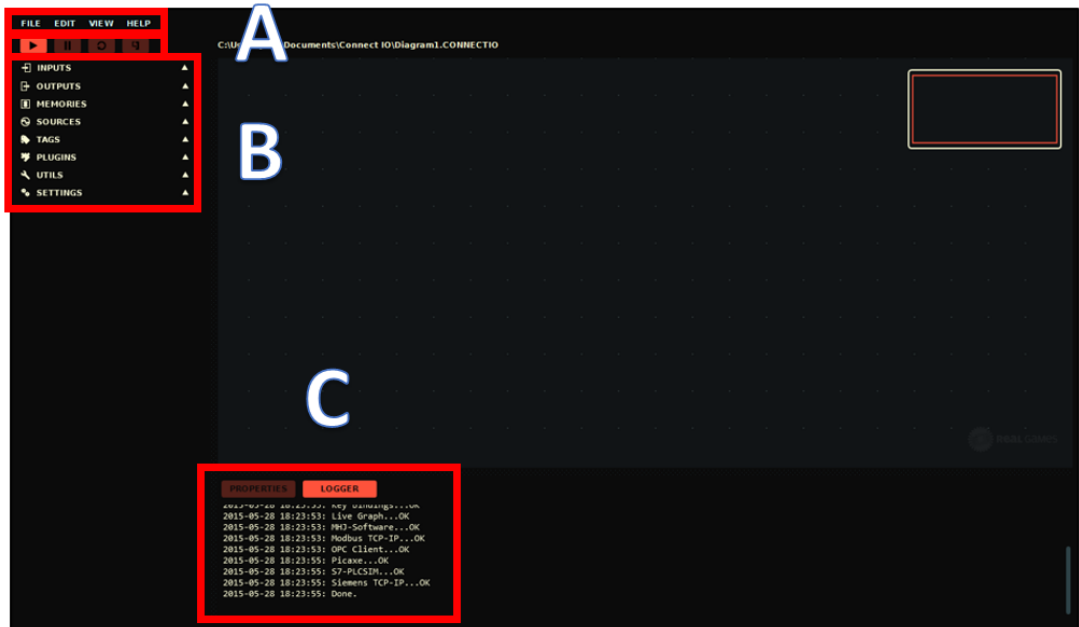


Fuente: Autor

3.4. FASE IV: ANÁLISIS DEL SOFTWARE CONNECT I/O (Anexo III)

Reconocer el espacio de trabajo con sus herramientas.

Figura 24. Reconocimiento de CONNECT I/O



Fuente: Autor

- **A:** Barra de herramientas.
- **B:** Panel Principal.
- **C:** panel de propiedades.

3.5. FASE V: INSTALACIÓN DRIVER TARJETA USB 4704 (Anexo I)

Se conecta la tarjeta a la computadora y se inserta el CD de drivers, para instalarla.

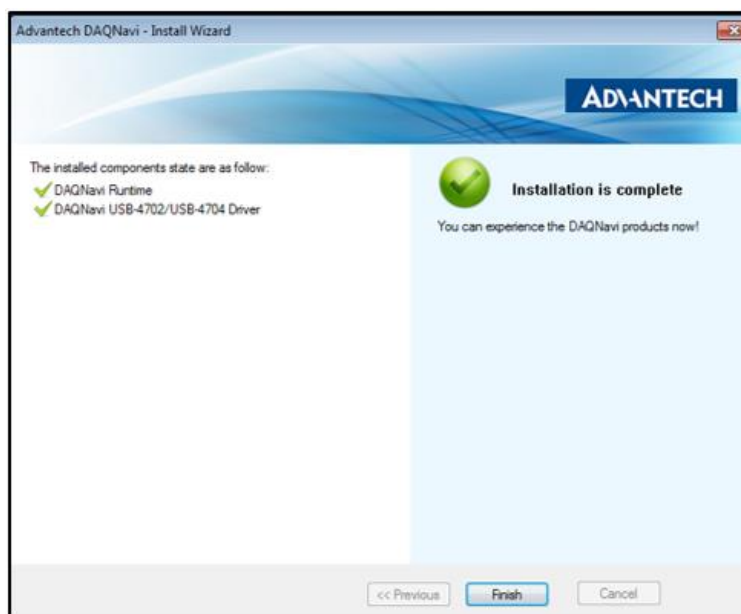
Figura 25. Tarjeta USB 4704 y drivers



Fuente: Autor

Previo a la instalación se siguen los pasos que sugiere este proceso.

Figura 26. Instalación de USB 4704



Fuente: Autor

3.6. FASE VI: CONEXIÓN TARJETA USB 4704 Y ARDUINO UNO (Anexo IV)

Se conecta las entradas del arduino UNO con las de la tarjeta USB 4704, de igual forma con sus salidas.

Figura 27. Conexión USB 4704 y Arduino UNO



Fuente: Autor

En el software Factory I/O es necesario invertir las salidas seleccionando invert.

Figura 28. Invertir las salidas en FACTORY I/O



Fuente: Autor

3.7. FASE VII: COMUNICACIÓN (PLC) S7-1200 Y FACTORY I/O (ANEXO V)

Se realiza la conexión del PLC S7-1200 al computador por medio del cable ETHERNET.

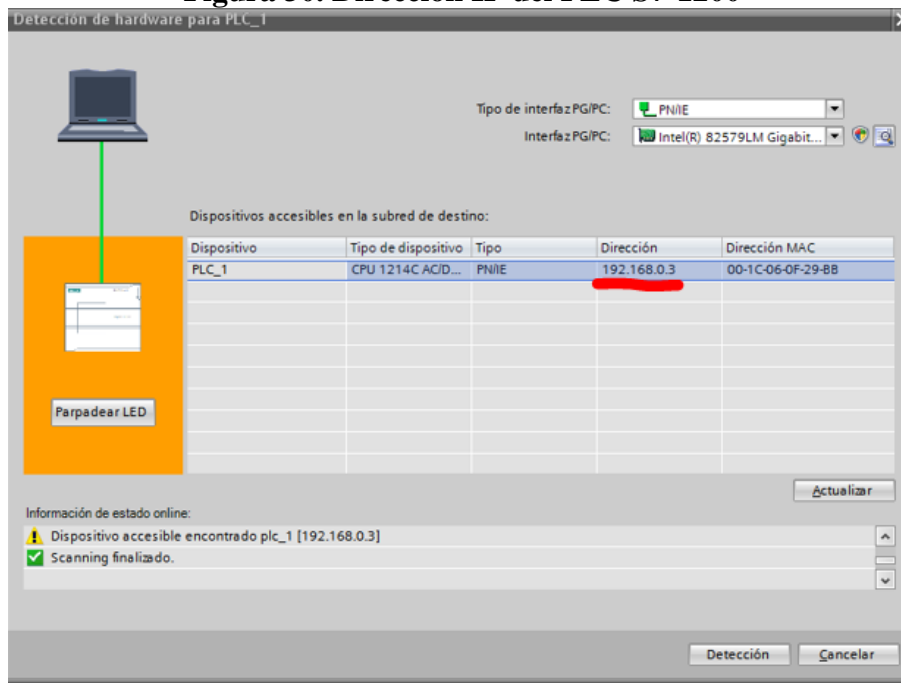
Figura 29. Conexión del PLC S7-1200



Fuente: Autor

En tía portal se crea un nuevo proyecto y se busca la dirección IP del PLC.

Figura 30. Dirección IP del PLC S7-1200



Fuente: Autor

Al ingresar al software Factory I/O se configura la dirección IP y se agrega un offset de 10 en las entradas.

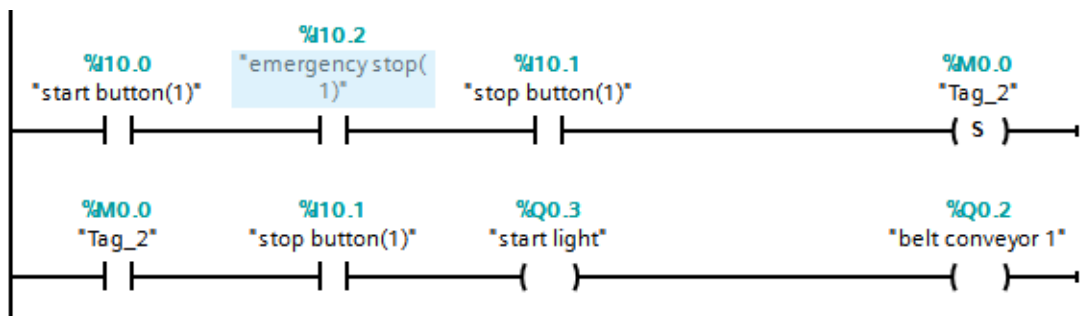
Figura 31. Configuración PLC S7-1200 En FACTORY I/O



Fuente: Autor

Al momento de realizar la programación en TIA portal, hay que tener en cuenta que en las entradas se debe aumentar en 10 la dirección de cada conector, ya que si no se realiza no lo tomara como entradas internas, por ejemplo %I10.2

Figura 32. Ejemplo de TIA portal entradas



Fuente: Autor

3.8. FASE VIII: DESARROLLO DEL EJEMPLO 1 BUFFER STATION (Anexo VI)

El objetivo de este ejemplo es acumular 5 cajas y luego dejarlas pasar una a la vez.

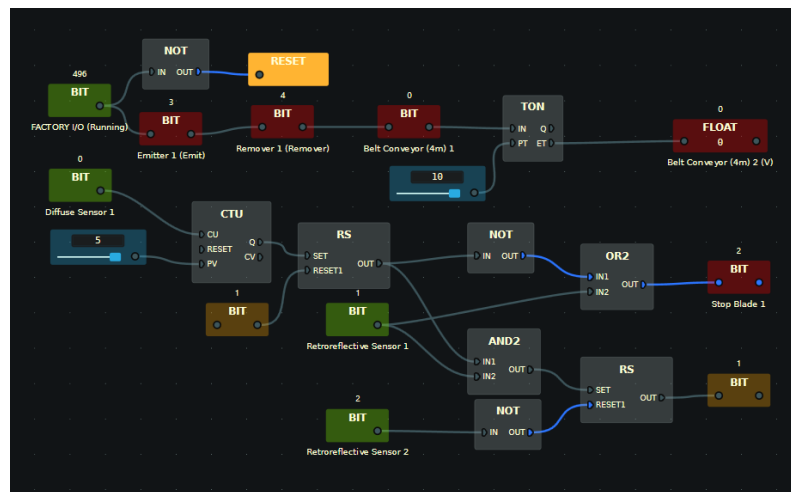
Figura 33. Ejemplo 1 FACTORY I/O



Fuente: Factory I/O

CONNECT I/O

Figura 34. Ejemplo 1 CONNECT I/O



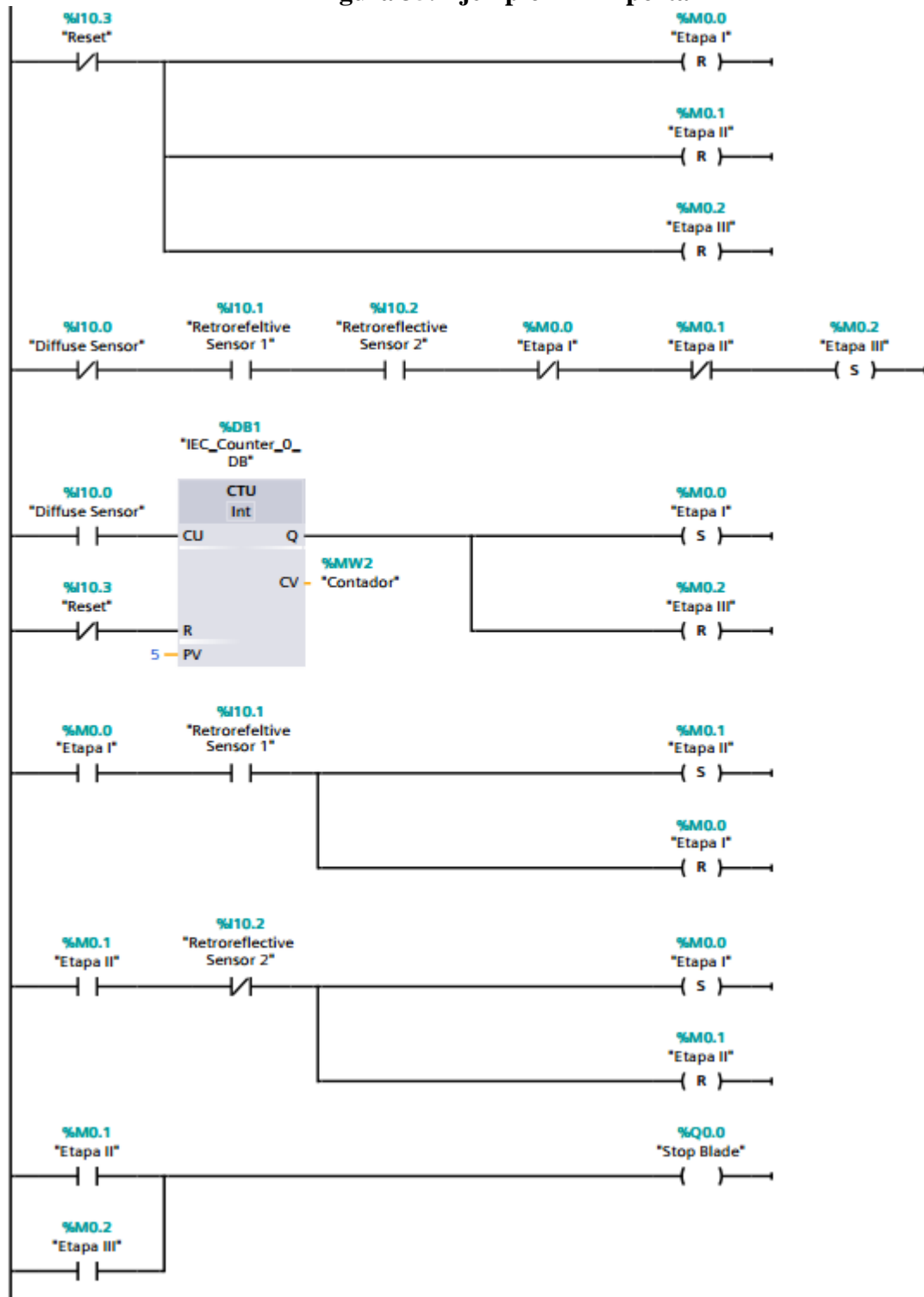
Fuente: Autor

ARDUINO UNO

```
const int DFS = 13; //diffuse sensor
const int RS1 = 12; //retroreflective
sensor 1
const int RS2 = 11; //retroreflective
sensor 2
const int STOP = 7; //stop blade 1
const int BC1 = 6; //belt conveyor 1
int a=0;
int b=0;
void setup() {
  pinMode(DFS, INPUT);
  pinMode(RS1, INPUT);
  pinMode(RS2, INPUT);
  pinMode(STOP, OUTPUT);
  Serial.begin(9600);
} //fin setup
void loop() {
  int Start = digitalRead(DFS);
  if(Start!=b){
    if(Start==HIGH){
      a++;
      Serial.println(a);
    }
  }
  b=Start;
  if (a<5){
    digitalWrite(STOP, HIGH);
  }else{
    uno:
    int StartA = digitalRead(RS1);
    if (StartA==LOW){
      digitalWrite(STOP, LOW);
    }else{
      dos:
      digitalWrite(STOP, HIGH);
      int StartB = digitalRead(RS2);
      if (StartB==LOW){
        goto uno;
      }else{
        goto dos;
      }
    }
  }
} //fin loop
```

TIA PORTAL

Figura 35. Ejemplo 1 TIA portal

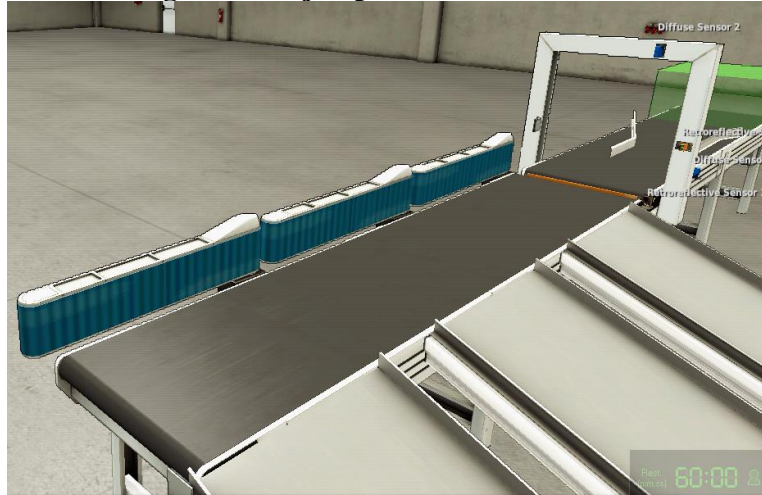


Fuente: Autor

3.9. FASE IX: DESARROLLO DEL EJEMPLO 2 SORTING STATION (Anexo VII)

El objetivo de este ejemplo es clasificar los tres tipos de cajas S, M y L

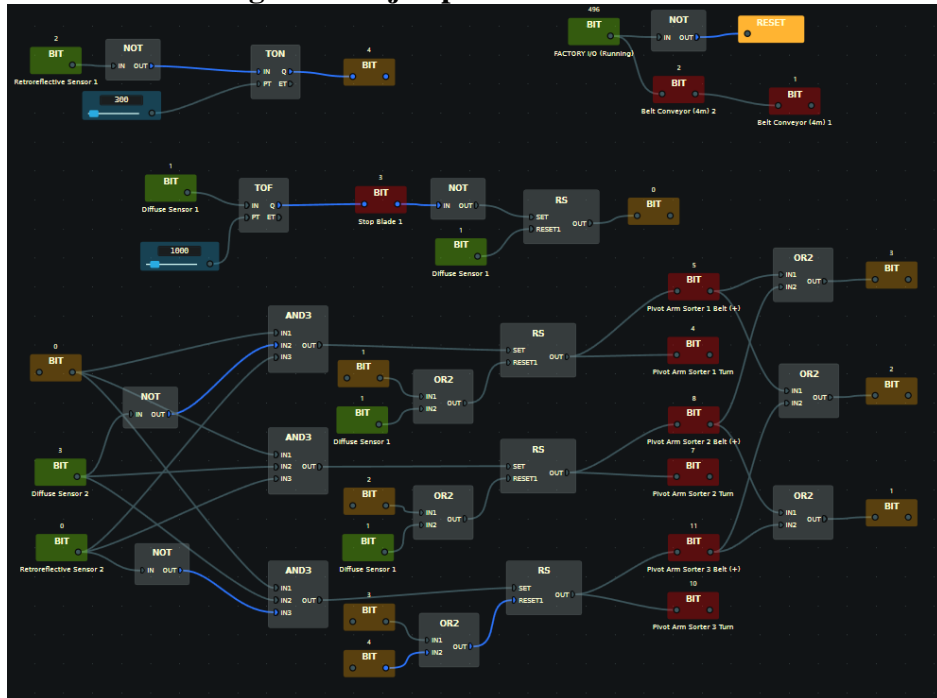
Figura 36. Ejemplo 2 FACTORY I/O



Fuente: Factory I/O

CONNECT I/O

Figura 37. Ejemplo 2 CONNECT I/O



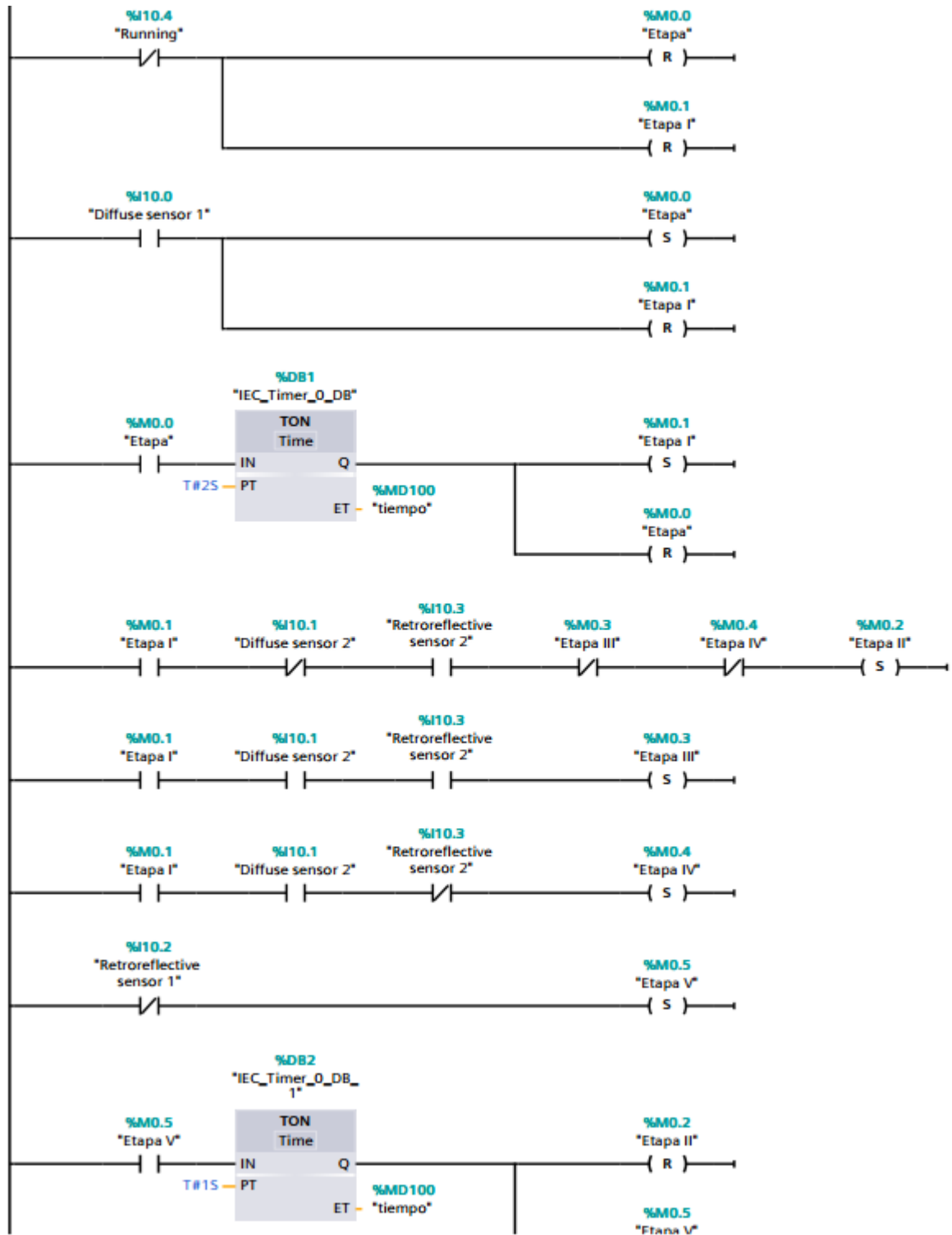
Fuente: Autor

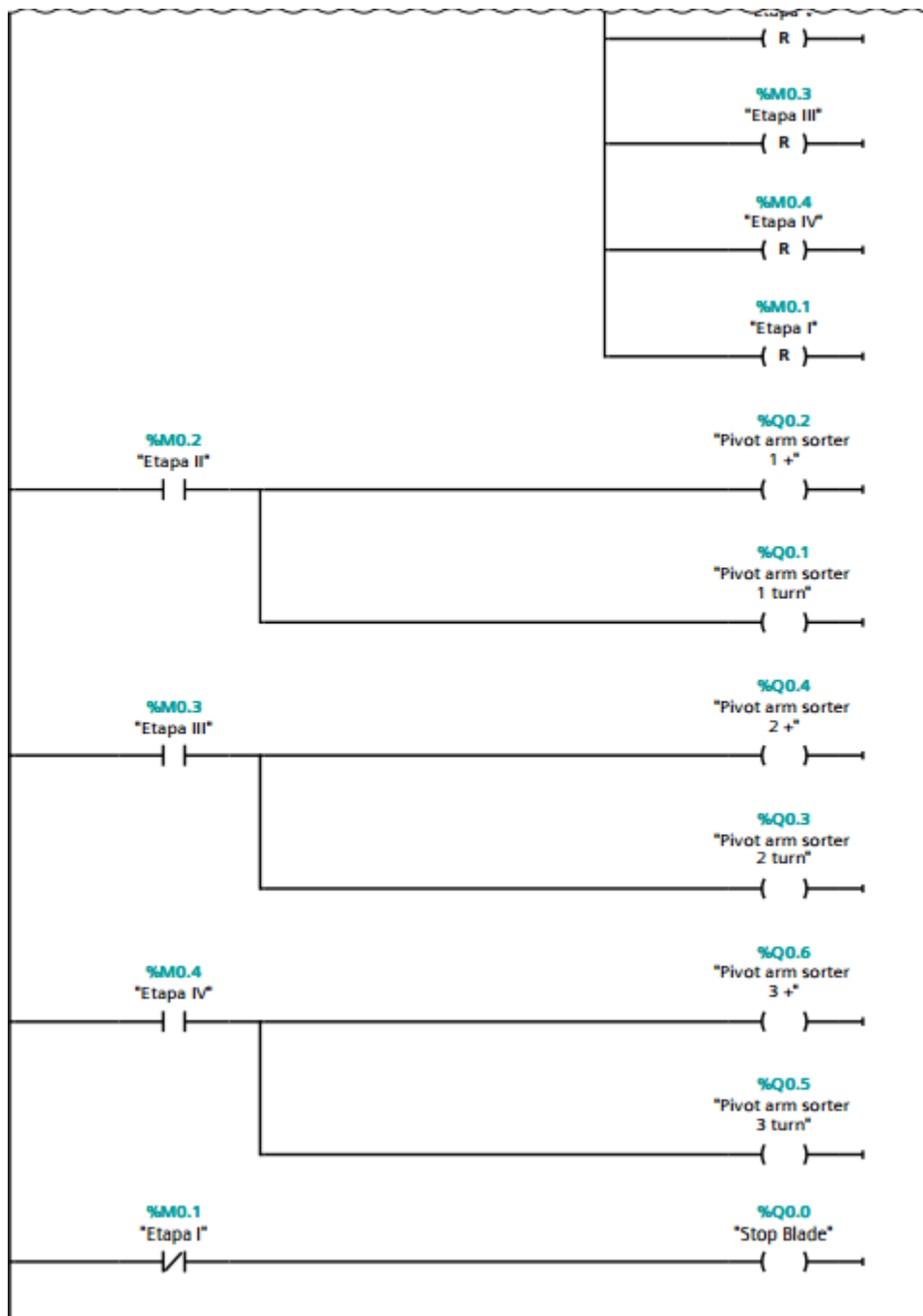
ARDUINO UNO:

```
const int DS1 = 13;
const int DS2 = 12;
const int RS1 = 11;
const int RS2 = 10;
const int STOP = 8;
const int PAS1T = 7;
const int PAS1S = 6;
const int PAS2T = 5;
const int PAS2S = 4;
const int PAS3T = 3;
const int PAS3S = 2;
int a = 0;
int b = 0;
int c = 0;
int d = 0;
void setup() {
  pinMode(DS1, INPUT);
  pinMode(DS2, INPUT);
  pinMode(RS1, INPUT);
  pinMode(RS2, INPUT);
  pinMode(STOP, OUTPUT);
  pinMode(PAS1T, OUTPUT);
  pinMode(PAS1S, OUTPUT);
  pinMode(PAS2T, OUTPUT);
  pinMode(PAS2S, OUTPUT);
  pinMode(PAS3T, OUTPUT);
  pinMode(PAS3S, OUTPUT);
}
void loop() {
  if (digitalRead(DS1)==HIGH){
    a=1;
  }
  if (a==1){
    uno:
    b=0;
    digitalWrite(STOP, HIGH);
    delay(1500);
    if (digitalRead(DS2)==LOW){
      if (digitalRead(RS2)==HIGH){
        b=1;
      }else{
        b=0;
        goto uno;
      }
    }else{
      if (digitalRead(RS2)==HIGH){
        b=2;
      }else{
        b=3;
      }
    }
    switch (b){
    case 1:
      digitalWrite(PAS1T, HIGH);
      digitalWrite(PAS1S, HIGH);
      digitalWrite(PAS2T, LOW);
      digitalWrite(PAS2S, LOW);
      digitalWrite(PAS3T, LOW);
      digitalWrite(PAS3S, LOW);
      break;
    case 2:
      digitalWrite(PAS1T, LOW);
      digitalWrite(PAS1S, LOW);
      digitalWrite(PAS2T, HIGH);
      digitalWrite(PAS2S, HIGH);
      digitalWrite(PAS3T, LOW);
      digitalWrite(PAS3S, LOW);
      break;
    case 3:
      digitalWrite(PAS1T, LOW);
      digitalWrite(PAS1S, LOW);
      digitalWrite(PAS2T, LOW);
      digitalWrite(PAS2S, LOW);
      digitalWrite(PAS3T, HIGH);
      digitalWrite(PAS3S, HIGH);
      break;
    }
    digitalWrite(STOP, LOW);
    delay(1500);
    a=0;
  }else{
    digitalWrite(STOP, HIGH);
  }
}
} //fin
```

TIA PORTAL

Figura 38. Ejemplo 2 TIA PORTAL



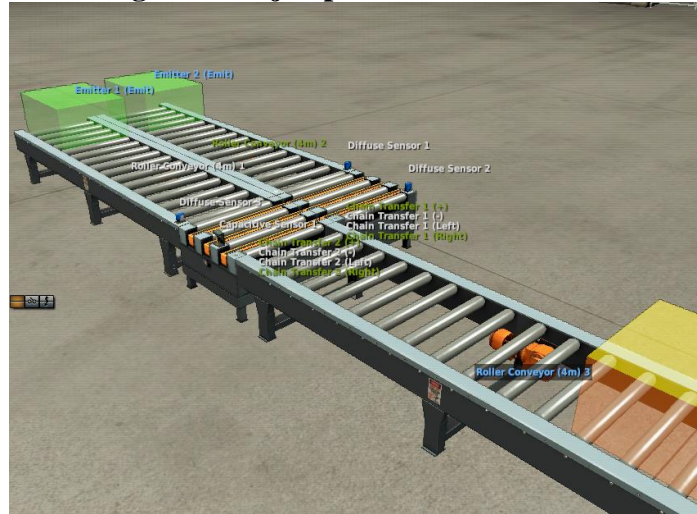


Fuente: Autor

3.10. FASE X: DESARROLLO DEL EJEMPLO 3 CONVERGE STATION (Anexo VIII)

El objetivo de este ejemplo es organizar las cajas que circulan paralelamente, de tal forma que puedan ir las dos por un solo camino.

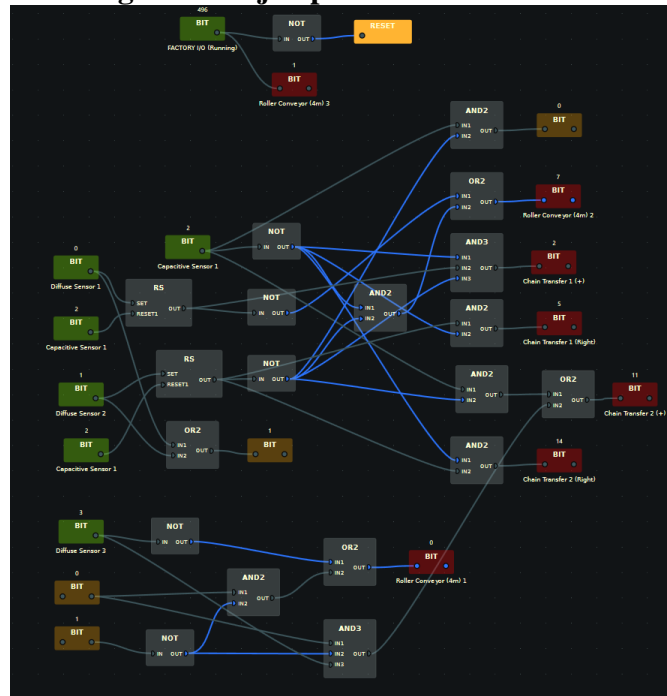
Figura 39. Ejemplo 3 FACTORY I/O



Fuente: Factory I/O

CONNECT I/O

Figura 40. Ejemplo 3 CONNECT I/O



Fuente: Autor

ARDUINO UNO

```
const int DS1 = 13;
const int DS2 = 12;
const int DS3 = 11;
const int CS1 = 10;
const int RC1 = 7;
const int RC2 = 6;
const int CT1S = 5;
const int CT1R = 4;
const int CT2S = 3;
const int CT2R = 2;
int a = 0;
int b = 0;
int c = 0;
void setup() {
  pinMode(DS1, INPUT);
  pinMode(DS2, INPUT);
  pinMode(DS3, INPUT);
  pinMode(CS1, INPUT);
  pinMode(RC1, OUTPUT);
  pinMode(RC2, OUTPUT);
  pinMode(CT1S, OUTPUT);
  pinMode(CT1R, OUTPUT);
  pinMode(CT2S, OUTPUT);
  pinMode(CT2R, OUTPUT);
}
void loop() {
  if (digitalRead(DS1)==HIGH){
    a=1;
  }
  if (a==1){
    if (digitalRead(DS3)==HIGH){
      if (digitalRead(DS2)==HIGH){
        b=1;
      }
      if (b==1){
        if (digitalRead(CS1)==HIGH){
          c=1;
        }
        if (c==1){
          if (digitalRead(CS1)==LOW){
            uno:
            digitalWrite(RC1, HIGH);
            digitalWrite(RC2, LOW);
            digitalWrite(CT1S, LOW);
            digitalWrite(CT1R, LOW);
            digitalWrite(CT2S, HIGH);
            digitalWrite(CT2R, LOW);
            digitalWrite(CT2S, HIGH);
          }
        }else{
          digitalWrite(RC1, LOW);
          digitalWrite(RC2, LOW);
          digitalWrite(CT1S, LOW);
          digitalWrite(CT1R, HIGH);
          digitalWrite(CT2R, HIGH);
        }
      }else{
        digitalWrite(RC1, LOW);
        digitalWrite(RC2, HIGH);
        digitalWrite(CT1S, HIGH);
      }
    }else{
      digitalWrite(RC1, HIGH);
      digitalWrite(RC2, LOW);
    }
  }else{
    if (digitalRead(DS3)==HIGH){
      digitalWrite(RC1, LOW);
      digitalWrite(RC2, HIGH);
    }
  }
}
```

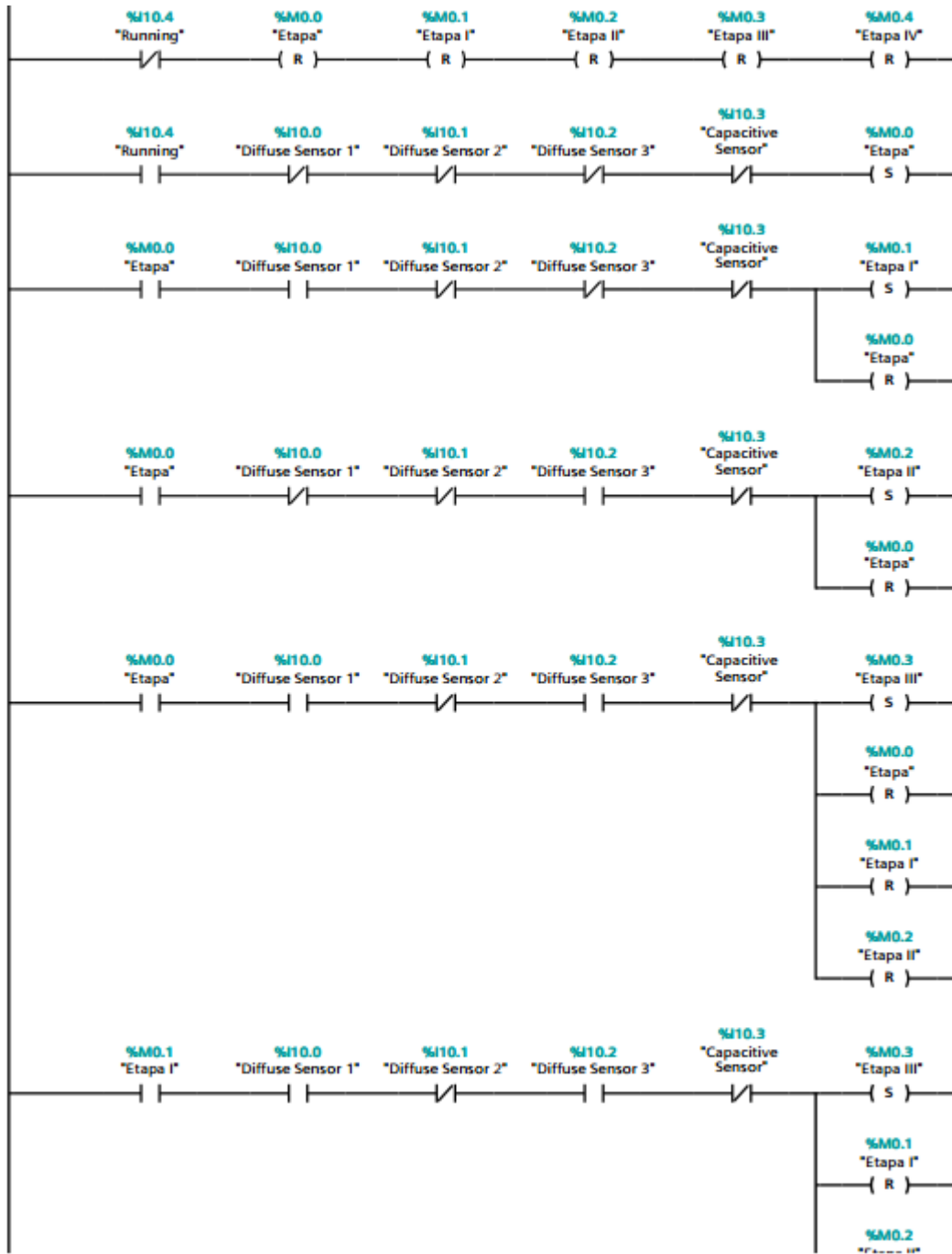
```

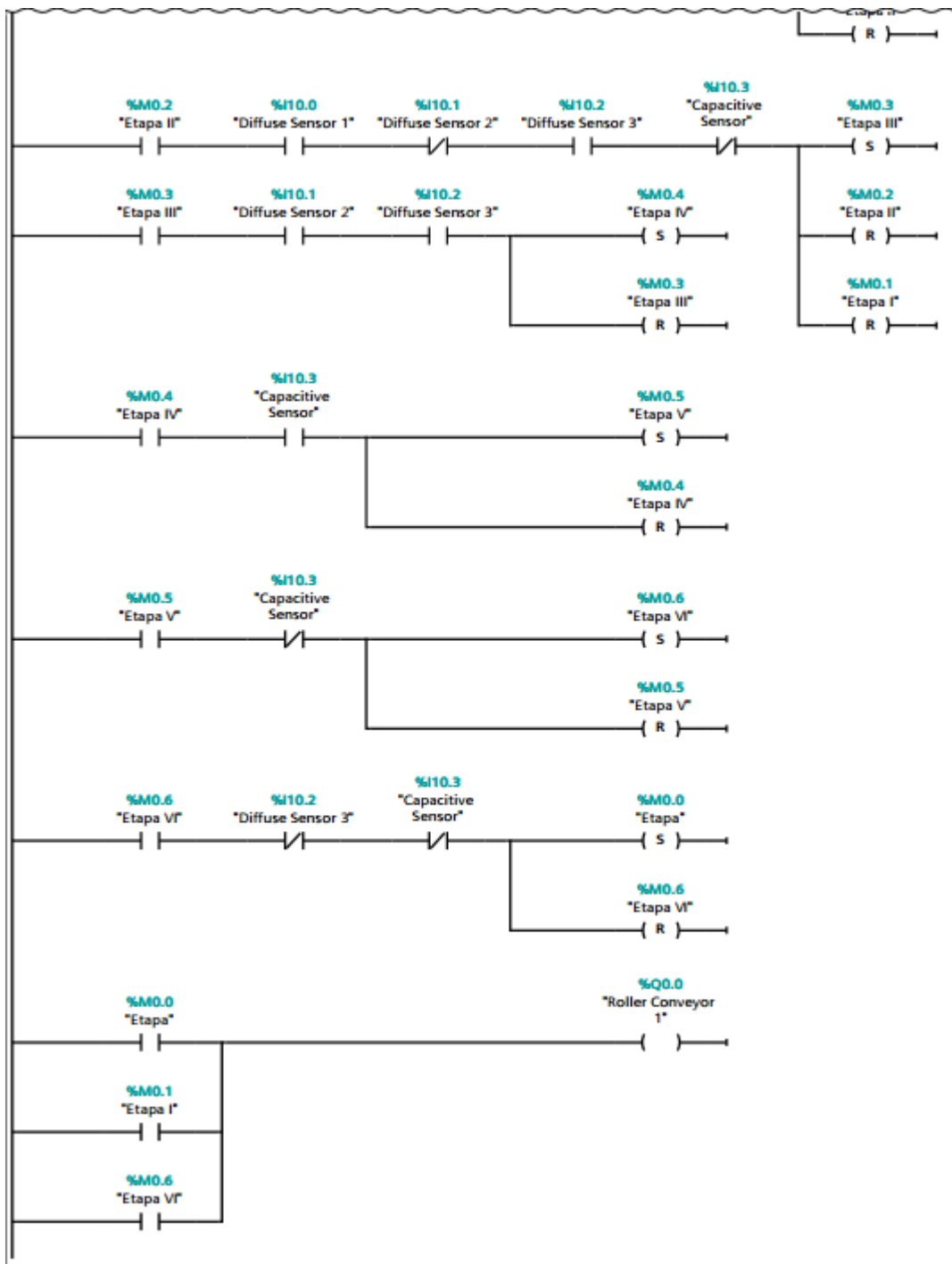
}else{
digitalWrite(RC1, HIGH);
digitalWrite(RC2, HIGH);
}
}
} //FIN LOOP

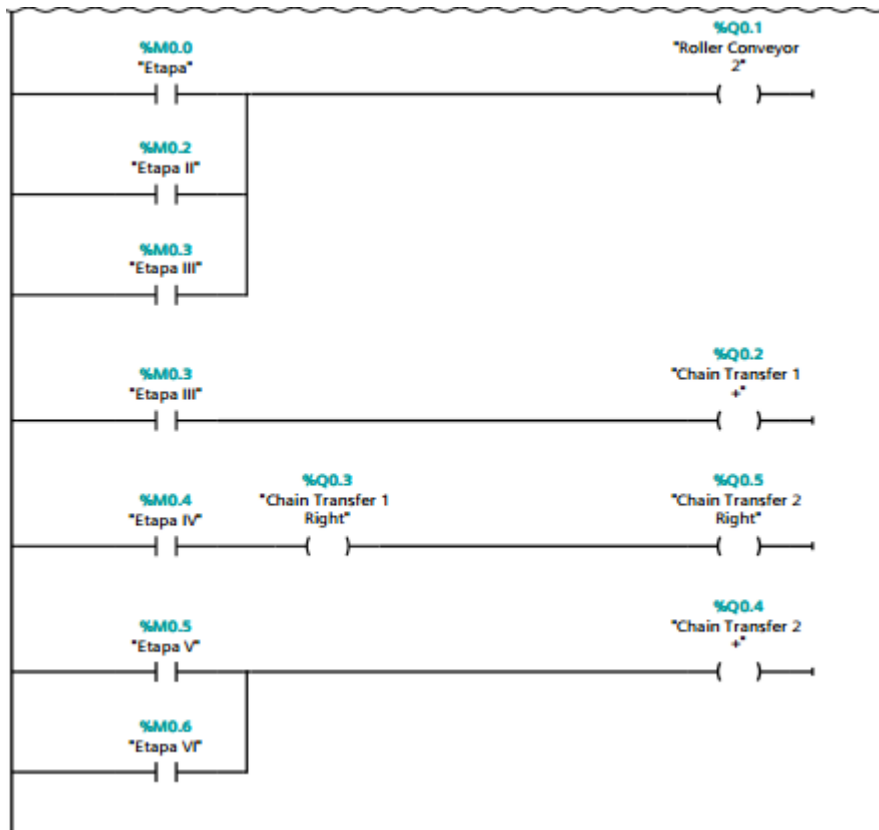
```

TIA PORTAL

Figura 41. Ejemplo 3 TIA Portal





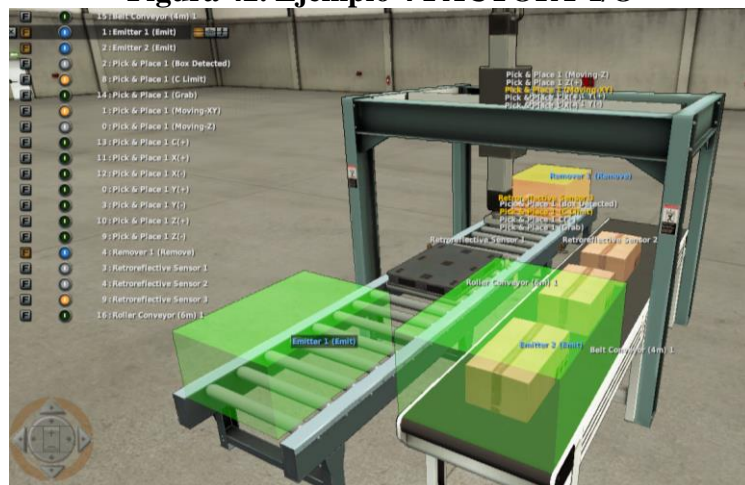


Fuente: Autor

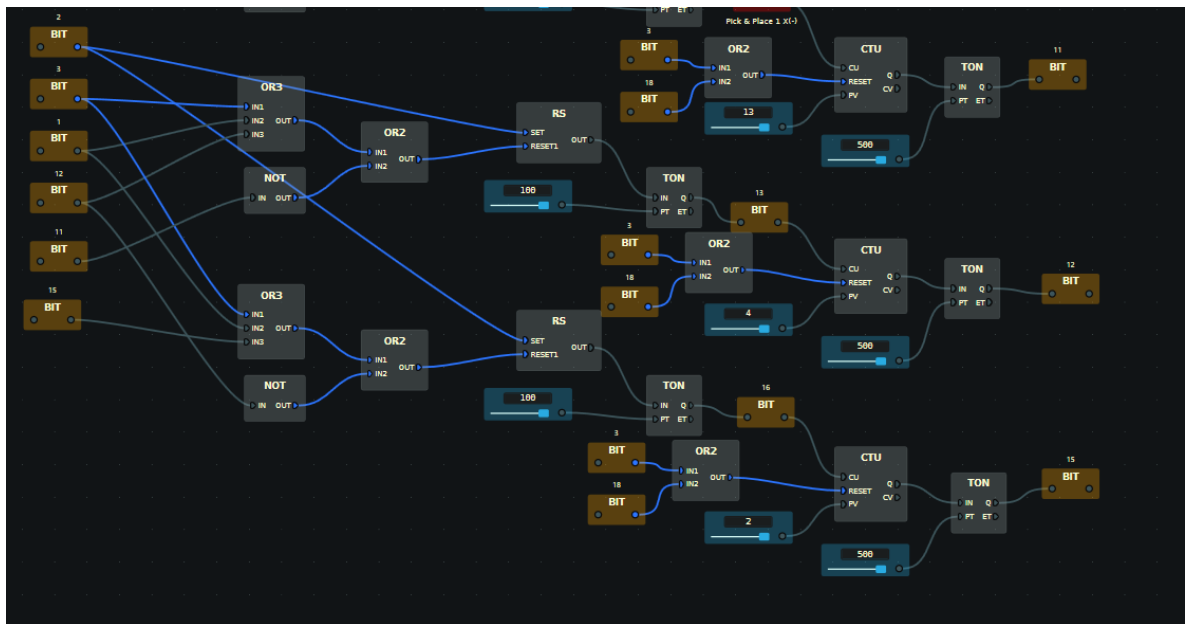
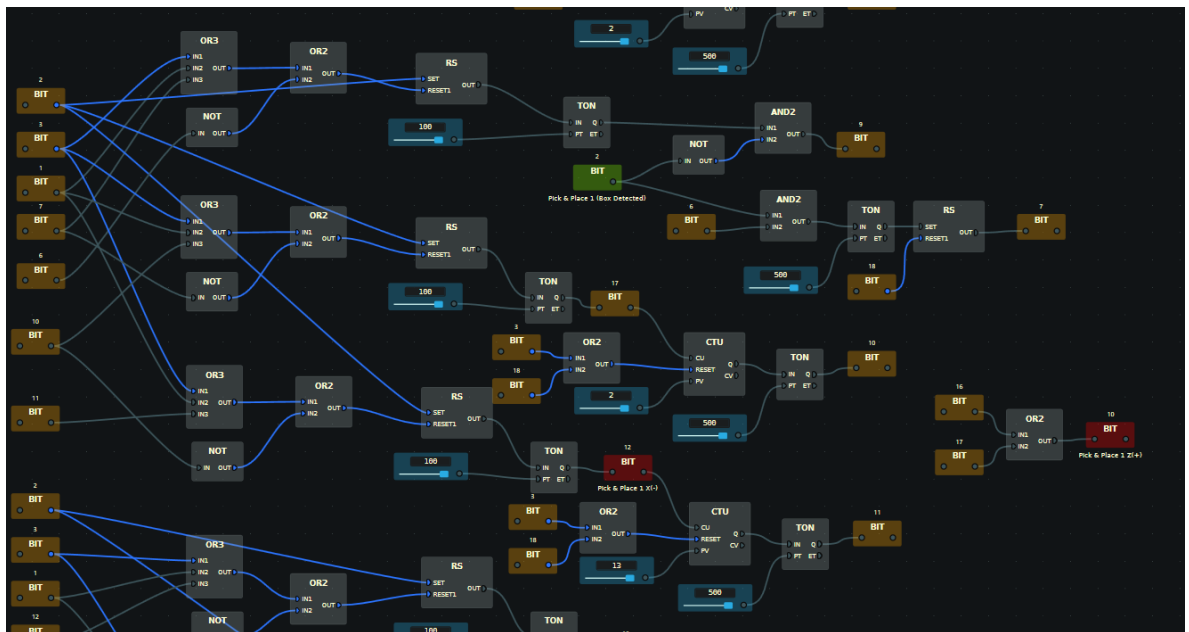
FASE XI: DESARROLLO DEL EJEMPLO 4 PICK & PLACE (Anexo IV)

El objetivo de este ejemplo es recoger por medio de un robot cartesiano una caja y acomodarla en una pallet.

Figura 42. Ejemplo 4 FACTORY I/O



Fuente: Factory I/O



Fuente: Autor

ARDUINO UNO

```
const int PPB = 13; //box detected
const int RS1 = 12; //retroreflective
sensor 1
const int RS2 = 11; //retroreflective
sensor 2
const int RS3 = 10; //retroreflective
sensor 3
const int BC1 = 9; //belt conveyor 1
const int PPG = 8; //grab
const int PPXS = 7; //X +
const int PPXR = 6; //X -
const int PPYS = 5; //Y +
const int PPZS = 4; //Z +
const int PPZR = 3; //Z -
const int RC1 = 2; //roller conveyor
int a = 0;
void setup() {
  pinMode(PPB, INPUT);
  pinMode(RS1, INPUT);
  pinMode(RS2, INPUT);
  pinMode(RS3, INPUT);
  pinMode(BC1, OUTPUT);
  pinMode(PPG, OUTPUT);
  pinMode(PPXS, OUTPUT);
  pinMode(PPXR, OUTPUT);
  pinMode(PPYS, OUTPUT);
  pinMode(PPZS, OUTPUT);
  pinMode(PPZR, OUTPUT);
  pinMode(RC1, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  if (digitalRead(RS2)==HIGH){
    if (digitalRead(RS1)==HIGH){
      digitalWrite(BC1, HIGH);
      digitalWrite(RC1, HIGH);
    }else{
      digitalWrite(BC1, HIGH);
      digitalWrite(RC1, LOW);
    }
  }else{
    if (digitalRead(RS1)==HIGH){
      digitalWrite(BC1, LOW);
      digitalWrite(RC1, HIGH);
    }else{
      digitalWrite(BC1, LOW);
      digitalWrite(RC1, LOW);
    }
  }
  while (a < 2) {
    digitalWrite(PPYS, HIGH);
    delay (500);
    digitalWrite(PPYS, LOW);
    delay (100);
    a++;
  }
  a=0;
  while (a < 2){
    digitalWrite(PPZR, HIGH);
    delay (500);
    digitalWrite(PPZR, LOW);
    delay (100);
    a++;
  }
  a=0;
  while (a < 3){
    digitalWrite(PPXS, HIGH);
    delay (500);
    digitalWrite(PPXS, LOW);
    delay (100);
    a++;
  }
  uno:
  a=0;
  while (a < 12){
    digitalWrite(PPXS, HIGH);
    delay (500);
    digitalWrite(PPXS, LOW);
    delay (100);
    a++;
  }
  digitalWrite(PPG, HIGH);
  a=0;
  while (a < 2){
    digitalWrite(PPZS, HIGH);
    delay (500);
    digitalWrite(PPZS, LOW);
    delay (100);
  }
}
```

```

    a++;
}
a=0;
//
while (a < 12){
    digitalWrite(PPXR, HIGH);
    delay (500);
    digitalWrite(PPXR, LOW);
    delay (100);
    a++;
}
a=0;
while (a < 4){
    digitalWrite(PPZR, HIGH);
    delay (500);
    digitalWrite(PPZR, LOW);
    delay (100);
    a++;
}
    digitalWrite(PPG, LOW);
a=0;
while (a < 2){
    digitalWrite(PPZS, HIGH);
    delay (500);
    digitalWrite(PPZS, LOW);
    delay (100);
    a++;
}
a=0;
dos:

```

```

if (digitalRead(RS2)==HIGH){
    if (digitalRead(RS3)==HIGH){
        digitalWrite(RC1, HIGH);
        digitalWrite(BC1, HIGH);
        goto dos;
    }else{
        digitalWrite(RC1, LOW);
        digitalWrite(BC1, HIGH);
        goto dos;
    }
}
}else{
    if (digitalRead(RS3)==HIGH){
        digitalWrite(RC1, HIGH);
        digitalWrite(BC1, LOW);
        goto dos;
    }else{
        if (digitalRead(RS1)==HIGH){
            digitalWrite(RC1, HIGH);
            digitalWrite(BC1, LOW);
            goto dos;
        }else{
            digitalWrite(RC1, LOW);
            digitalWrite(BC1, LOW);
        }
    }
}
}
goto uno;
}
}

```

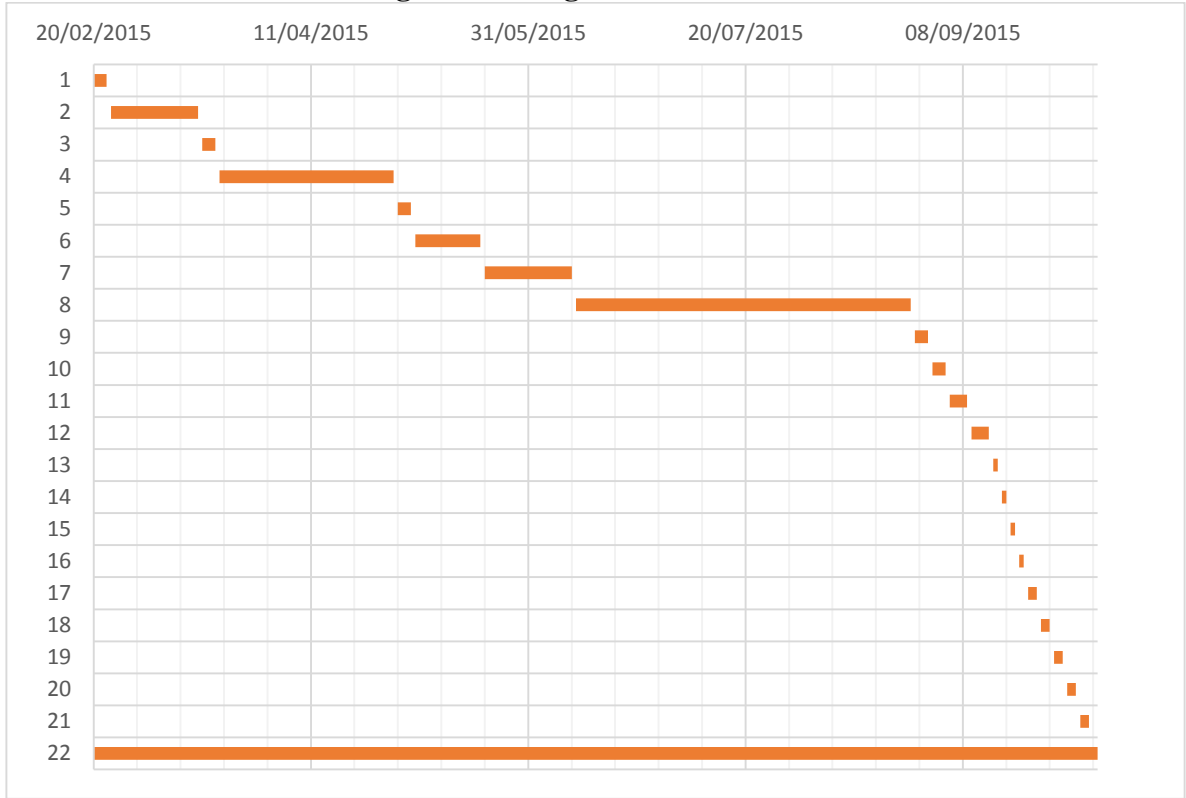
Cronograma:

Tabla 5. Cronograma De Actividades

#	Actividad	Fecha de inicio	duración	Fecha de terminación
1	<i>Instalación del software Factory I/O</i>	20/02/2015	3	23/02/2015
2	<i>Análisis del software Factory I/O</i>	24/02/2015	20	16/03/2015
3	<i>Instalación del software Connect I/O</i>	17/03/2015	3	20/03/2015
4	<i>Análisis del software Connect I/O</i>	21/03/2015	40	30/04/2015
5	<i>Instalación driver tarjeta USB 4704</i>	01/05/2015	3	04/05/2015
6	<i>Conexión tarjeta USB 4704 y arduino uno</i>	05/05/2015	15	20/05/2015
7	<i>Comunicación tarjeta USB 4704 y Factory I/O</i>	21/05/2015	20	10/06/2015
8	<i>Comunicación (PLC) S7-1200 y Factory I/O</i>	11/06/2015	77	27/08/2015
9	<i>Desarrollo del ejemplo 1 Buffer Station</i>	28/08/2015	3	31/08/2015
10	<i>Desarrollo del ejemplo 2 Sorting Station</i>	01/09/2015	3	04/09/2015
11	<i>Desarrollo del ejemplo 3 Converge Station</i>	05/09/2015	4	09/09/2015
12	<i>Desarrollo del ejemplo 4 Pick & Place</i>	10/09/2015	4	14/09/2015
13	<i>Elaboración Guía I</i>	15/09/2015	1	16/09/2015
14	<i>Elaboración Guía II</i>	17/09/2015	1	18/09/2015
15	<i>Elaboración Guía III</i>	19/09/2015	1	20/09/2015
16	<i>Elaboración Guía IV</i>	21/09/2015	1	22/09/2015
17	<i>Elaboración Guía V</i>	23/09/2015	2	25/09/2015
18	<i>Elaboración Guía VI</i>	26/09/2015	2	28/09/2015
19	<i>Elaboración Guía VII</i>	29/09/2015	2	01/10/2015
20	<i>Elaboración Guía VIII</i>	02/10/2015	2	04/10/2015
21	<i>Elaboración Guía IX</i>	05/10/2015	2	07/10/2015
22	<i>Documento Final</i>	20/02/2015	231	09/10/2015

Fuente: Autor

Figura 44. Diagrama de GANTT.



Fuente: Autor

PRESUPUESTO:

Tabla 6. Presupuesto

CONCEPTOS	VALOR
tarjeta USB-4704	\$ 616.040
Software Factory I/O	\$ 7.625.000
Fuente PLC	\$ 341.472
PLC Siemens S7-1200	\$ 787.905
Arduino Uno	\$ 30.000
TOTAL	\$ 9.400.417

Fuente: Autor

4. CONCLUSIONES

- Para elaborar las guías de trabajo fue necesario entender el funcionamiento y conexión entre las herramientas que se utilizaron, tanto el software como las tarjetas.
- La comunicación entre el PLC S7-1200 y el software Factory I/O, presenta una respuesta rápida y cómoda al momento de simular un sistema, ya que solo requiere un cable Ethernet, hay que tener en cuenta una debida configuración y el agregar los offset en software.
- Para el desarrollo de una programación, es indispensable ya tener una estructura previa del algoritmo, es decir tener presentes conceptos básicos, como lo es el uso del algebra booleana y diagramas de flujo, estas herramientas son útiles al momento de encontrar errores o pre visualizar un algoritmo.
- Se presentó un desarrollo progresivo en la dificultad, en las guías de trabajo, y una explicación clara paso a paso de cómo realizar el manejo de estas herramientas, con el fin de que el lector las entienda.
- El rendimiento del software Factory I/O, depende mucho de las características mínimas del ordenador para que este lo soporte, por esta razón es necesario bajar la calidad de gráficos para no presentar errores al momento de dar inicio a la simulación.

5. RECOMENDACIONES

- Depende del rendimiento del ordenador depende la correcta simulación en el programa Factory I/O, tener en cuenta modificar el nivel de gráficas.
- Ya que Factory I/O es un simulador, se recomienda dar algunos retardos en las programaciones, ya que en ocasiones no toma lectura.
- Al conectar el PLC S7-1200 con el software Factory I/O no olvidar que las entradas van a partir del dígito 10 es decir por ejemplo %I10.0, y de igual forma modificar el offset en Factory I/O a partir de 10, si no se realiza esto el PLC tomara las entradas Físicas y no las internas.
- Si se utiliza la tarjeta USB 4704 y Arduino Uno, es indispensable conectar las tierras de las dos, así se realiza la lectura, de igual forma en Factory I/O invertir las salidas.

6. BIBLIOGRAFIA

- [1] Factory I/O user Guide.
(http://www.realgames.pt/downloads/factoryio/software_files/factoryio_user_guide_en.pdf)
- [2] USB-4704 DATASHEET (http://downloadt.advantech.com/ProductFile/PIS/USB-4704/Product%20-%20Datasheet/USB-4704_DS20131212114147.pdf)
- [3] Automatización problemas resueltos con autómatas programables. J. Pedro Romera, J. Antonio Lorite, Sebastián Montoro. Editorial Paraninfo SA.
- [4] Autómatas programables y sistemas de automatización. Mandado Pérez, Enrique. Editorial Alfaomega.
- [5] Ingeniería de la automatización industrial. Piedrafita Moreno, Ramón. Editorial Alfaomega.
- [6] Sistemas digitales principios y aplicaciones, Tocci, Ronald J. editorial Prentice Hall.
- [7] ARDUINO curso práctico de formación. Torrente Artero, Óscar. Editorial Alfaomega.
- [8] Automating with simatic S7-1200 hardware components, programming with STEP 7 basic in LAD and FBD, visualization with HMI basic panels. Berger, Hans. Publicis Publishing.
- [9] Romero, j., Lorite, J., & Montoro, S. (1994). *Automatizacion problemas resuletos con autómatas Programables*. Paraninfo sa.
- [10] [http://www.marcombo.com/Descargas/9788426715753/SENSORES/TEMAS/SA%20Tema%2016%20Interfaces%20\(1\).pdf](http://www.marcombo.com/Descargas/9788426715753/SENSORES/TEMAS/SA%20Tema%2016%20Interfaces%20(1).pdf)
- [11] Balcells, J., & Romeral, J. L. (s.f.). *Autómatas Programables*. marcombo.