

RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de INGENIERO MECATRÓNICO.
2. **TÍTULO:** INTEGRACIÓN Y PROGRAMACIÓN DE UN SISTEMA DIDÁCTICO DE INDUSTRIA 4.0 PARA LA UNIVERSIDAD DE SAN BUENAVENTURA, SEDE BOGOTÁ.
3. **AUTOR:** Daniel Eduardo Rincón Ruiz.
4. **LUGAR:** Bogotá, D.C.
5. **FECHA:** Marzo de 2018.
6. **PALABRAS CLAVE:** ingeniería, Industria 4.0, Sistemas Ciberfísicos, módulos de trabajo, PLC.
7. **DESCRIPCIÓN DEL TRABAJO:** El objetivo principal de este proyecto es el diseño e implementación de un sistema didáctico de industria 4.0 en la universidad de San Buenaventura, haciendo uso de los elementos presentes en el laboratorio de automatización del plantel. Se conforma una secuencia básica de funcionamiento usando los dispositivos interconectados y aplicando los puestos de trabajo de Distribución, Manipulación, Procesamiento y Clasificación además de la implementación de un Robotino.
En el siguiente documento se encontrarán documentados cada uno de los protocolos de comunicación utilizados, los elementos de investigación y el funcionamiento básico del sistema.
8. **LÍNEAS DE INVESTIGACIÓN:** Línea de investigación de la USB: Tecnologías actuales y Sociedad. Sub línea de Facultad de Ingeniería es: Instrumentación y Control de Procesos. Campo Temático del Programa: Automatización de Procesos.
9. **METODOLOGÍA:** Es de carácter empírico-analítico, con un enfoque metodológico con base en el estudio y diseño de un sistema didáctico de industria 4.0.
10. **CONCLUSIONES:** Se diseñó e implementó una Interfaz Humano Máquina desde la pantalla táctil FED-550, la cual tiene la intención de permitirle al operador dar órdenes a cada una de las 4 estaciones de manera independiente, y así mismo adquirir información fundamental para el funcionamiento de la aplicación desde cada una de las mismas y mostrarla eficientemente. Se realiza un manual que explique el funcionamiento del sistema, basado en las pruebas sobre el sistema ya en funcionamiento. La realización del programa demo en el cual se pueden evidenciar los desarrollos explicados anteriormente. Finalmente, con base a lo anterior se concluye que se cumple el objetivo general de esta tesis al haberse podido integrar y programar un sistema que aplique varios de los conceptos de la industria 4.0, a través de una metodología y un desarrollo ingenieril previamente propuestos utilizando equipos existentes en la universidad San Buenaventura, Sede Bogotá.

**INTEGRACIÓN Y PROGRAMACIÓN DE UN SISTEMA DIDÁCTICO DE
INDUSTRIA 4.0 PARA LA UNIVERSIDAD DE SAN BUENAVENTURA, SEDE
BOGOTÁ.**

ii

Daniel Eduardo Rincón Ruiz.
Febrero 2018.

Tutor: Ing. Rafael Castro

Universidad de San Buenaventura.
Facultad de Ingeniería.
Programa de Ing. Mecatrónica.
Bogotá D.C.
Proyecto de Grado.

**INTEGRACIÓN Y PROGRAMACIÓN DE UN SISTEMA DIDÁCTICO DE
INDUSTRIA 4.0 PARA LA UNIVERSIDAD DE SAN BUENAVENTURA, SEDE
BOGOTÁ.**

iii

Daniel Eduardo Rincón Ruiz.
Febrero 2018.

Trabajo presentado como requisito parcial para optar por el título de profesional
en Ingeniería Mecatrónica.
Asesor: Rafael Castro

Universidad de San Buenaventura.
Facultad de Ingeniería.
Programa de Ing. Mecatrónica.
Bogotá D.C.
Proyecto de Grado.

Through this graduation project is proposed the implementation of a didactic system of industry 4.0 from the elements present in the automation laboratory, both modules and PLC (Festo / Siemens). Achieving create a system of recognition of parts capable of identifying if any of its parts is defective and if so replace it with another.

Being the main challenge and objective to achieve communication between each of the participating devices of the system, for which it was necessary to use three types of communication such as the EasyIP, UDP and S7 protocols (the latter exclusively for Siemens). After achieving this, the demo sequence mentioned above was programmed, linking it to a remote control with a control screen.

La presente investigación tiene como finalidad el llevar a cabo una integración de sistemas destinados a la educación en el ámbito de la automatización industrial, utilizando como referente a la industria 4.0, la cual suele ser aplicada a distintos procesos que pretenden ser ágiles y eficientes en sus procesos de automatización, posibles por el uso masivo de tecnologías de la información y comunicaciones como lo son: Internet, comunicación tiempo real entre máquinas y la fabricación aditiva.

Por su parte los sistemas ciberfísicos están diseñados como un sistema de red de elementos que interactúan entre sí con entradas y salidas físicas en lugar de comunicarse por medio de dispositivos aislados y constituyen el máximo nivel en la formación de la industria 4.0, reproduciendo las estaciones de una planta de producción real en forma de modelos integrales de tecnología mecatrónica y automatización.

La característica principal de la industria 4.0 es permitir que personal, equipos y sistemas se encuentren conectados a tiempo completo a lo largo de la cadena de valor, por lo cual toda la información relevante se encuentra disponible en tiempo real además de estar al alcance de proveedores, fabricantes y clientes.

La intención principal de este proyecto es académica, al permitir a los estudiantes de Ingeniería Mecatrónica de la Universidad de San Buenaventura, estar al tanto de estos nuevos conceptos y tener la capacidad de aplicarlos a futuro en su vida profesional, por lo cual para darle cumplimiento se implementa un sistema didáctico de industria 4.0 que utiliza las estaciones de trabajo presentes dentro del laboratorio de automatización.

En principio, se planteó el diseño de la ubicación de las estaciones como el de su^{vi} interfaz, luego se realizó la comunicación entre ellas y por último se programó una secuencia demo dispuesta para evidenciar el funcionamiento completo de la industria 4.0 aplicada.

Capítulo 1 Planteamiento del problema.....	1
1.1 Antecedentes (Estado del Arte)	1
1.2 Descripción y Formulación del Problema.....	5
1.3 Justificación.	6
1.4 Objetivos de la investigación.	7
1.5 Alcances y Limitaciones.	7
Capítulo 2 Marco Teórico.....	9
3.1 Industria 4.0.	9
3.2 ToolBox Industria 4.0	9
3.3 Protocolo EasyIP.....	12
3.4 UDP OpenService.....	13
3.5 UDP OpenSend.....	13
Capítulo 3 Diseño Ingenieril.....	15
3.1 Diseño del sistema didáctico industria 4.0.....	15
3.1.1 Ubicación de módulos de trabajo.....	16
3.1.2 Interfaz del sistema didáctico de industria 4.0.....	19
3.1.3 Diseño Ingenieril.	22
3.2. Protocolos de comunicación y programación interfaz.....	24
3.2.1 Comunicación entre estaciones de Distribución y Procesamiento.	24
3.2.2 Comunicación entre estación Procesamiento y Robotino.....	25
3.2.3 Comunicación entre estaciones Distribución y Manipulación.....	25
3.2.4 Comunicación S7 entre estación Manipulación y Clasifiación.	26
3.2.5 Programación interfaz.....	28
Capítulo 4 Programación secuencia demo.....	29
4.1 Pantalla de Inicio.....	29
4.2 Estación de Distribución.....	29
4.3 Estación de Procesamiento.	30
4.4 Robotino.....	30
4.5 Estación Manipulación.....	30
4.6 Estación Clasificación.....	31
Capítulo 5. Análisis de Datos.....	33
5.1 Flujo de datos entre estaciones Distributing y Handling por medio de UDP.....	33
Capítulo 6 Manual de Usuario.....	38
6.1 Inicialización del sistema.....	49
6.2 Posición inicial del Robotino.....	56
Capítulo 7 Conclusiones.....	59
Lista de referencias	62
Anexo.....	64
Anexo 1. Diagrama de Flujo estación Distribución.....	64
Anexo 2. Diagrama de Flujo estación Procesamiento.....	67
Anexo 3. Diagrama de Flujo estación Manipulación.....	71
Anexo 4. Diagrama de Flujo estación Clasificación.....	75
Anexo 5. Planos estructura diseño del Robotino.	78
Anexo 6. CD con programas adjuntos.....	81

Figura 1.1. Cuadro de herramientas Industria 4.0 – Productos. (VDMA German Engineering Federation, 2016)	11
Figura 2. Cuadro de herramientas industria 4.0 – Producción. (VDMA German Engineering Federation, 2016)	14
Figura 3. Diseño ubicación de estaciones y robotino a implementar para el Sistema didáctico de industria 4.0.....	17
Figura 4. Ubicación real del sistema didáctico de industria 4.0.	18
Figura 5. Ventana principal estación Distribución.	20
Figura 6. Ventana de Registro de Piezas.	21
Figura 7. Bases de datos dentro de estación Handling para comunicación con Distributing.	34
Figura 8. Bases de datos dentro de estación Distributing para comunicación con Handling.	35
Figura 9. Forzado de una variable en estación Handling.....	36
Figura 10. Respuesta al forzado de la variable en estación Distributing.....	37
Figura 11. Bases de datos dentro de estaciones Distributing y Processing para comunicación con Easy_IP.	39
Figura 12. Bases de datos dentro de estaciones Distributing y Processing con una variable forzada.....	39
Figura 13. Base de datos dentro de estación Processing para comunicación con Robotino.....	41
Figura 14. Muestra de envío de datos desde la estación hacia el Robotino.....	42
Figura 15. Respuesta del Robotino y asignación de nuevo trabajo.	43
Figura 16. Bases de datos de estaciones Handling y Sorting para comunicación por S7_Connection.....	44
Figura 17. Demostración de variable forzada desde estación Handling hasta Sorting.....	45
Figura 18. Demostración de Reset forzado desde estación Handling hasta Sorting.....	46
Figura 19. Envío de información diagnostica desde estación Sorting a estación Handling.	47
Figura 20. Forzado de variables de estación Sorting reflejado en base de datos de estación Handling.....	48
Figura 21. Puntos de energización del Sistema didáctico de Industria 4.0.....	50
Figura 22. Encendido de dispositivos.....	52
Figura 23. Interruptor para encender la estación Distribucion.	53
Figura 24. Fuente alimentación PLC.	54
Figura 25. Perillas neumáticas a abrir.....	55
Figura 26. Posición Inicial Robotino.	56
Figura 27. Pantalla Principal del Robotino.	57
Figura 28. Diagrama de Flujo Distribución. Parte 1.....	64
Figura 29. Diagrama de Flujo Distribución. Parte 2.....	65
Figura 30. Diagrama de Flujo Distribución. Parte 3.....	66
Figura 31. Diagrama de Flujo Procesamiento. Parte 1.	67
Figura 32. Diagrama de Flujo Procesamiento. Parte 2.	68
Figura 33. Diagrama de Flujo Procesamiento. Parte 3.	69
Figura 34. Diagrama de Flujo Procesamiento. Parte 4.	70
Figura 35. Diagrama de Flujo Manipulación.Parte 1.....	71
Figura 36. Diagrama de Flujo Manipulación.Parte 2.....	72
Figura 37. Diagrama de Flujo Manipulación.Parte 3.....	73

Figura 38. Diagrama de Flujo Manipulación.Parte 4.....	74
Figura 39. Diagrama de Flujo Clasificación.Parte 1.....	75
Figura 40. Diagrama de Flujo Clasificación.Parte 2.....	76
Figura 41 Diagrama de Flujo Clasificación.Parte 3.....	77
Figura 42. Plano de torre para gripper de Robotino.....	78
Figura 43. Plano de soporte para gripper de Robotino	79
Figura 44. Plano soporte Pinza derecha del gripper.	80
Figura 45. Plano soporte extensor de gripper para Robotino.....	81

Capítulo 1

Planteamiento del problema

1.1 Antecedentes (Estado del Arte)

Entre los desarrollos más significativos de los sistemas electrónicos y su integración con las recientes tendencias en la tecnología de información y comunicación, se encuentra la implementación de sistemas ciberfísicos. Muchos definen a la industria 4.0 como Cudri et al. (2015) lo hace relacionándola con la integración de sistemas de información, con lo cual se busca otorgar acceso en tiempo real a la información y datos del proceso a controlar, los cuales son almacenados en una nube informática, siendo posible así controlar la respuesta del sistema, su configuración y el funcionamiento del proceso. Por su parte, Cudri se encarga de aplicarla a procesos de control numérico, en este caso, una CNC para la fabricación de placas de circuitos impresos y su integración con la simulación, así como con sus características físicas cibernéticas. Todo esto a partir de un control electrónico automatizado, donde se propuso visualizar el comportamiento de los distintos sistemas desde sensores a actuadores conectados a una tarjeta de adquisición de datos, la cual interactúa con una plataforma de visualización tridimensional.

Teniendo en cuenta lo dicho por García, Irisarri y Pérez (2017) los sistemas automatizados industriales se encuentran enfrentando los desafíos que actualmente surgen al tratar de solventar las necesidades generadas por el mercado competitivo actual, por lo cual recurrir a sistemas ciberfísicos de producción se ha convertido en la mejor opción. En virtud de lo anterior se ha iniciado la creación de aplicaciones para el

intercambio de datos entre el nivel de planta y las capas más altas empleando un estándar industrial, todo ello utilizando el estándar industrial OPC UA (Arquitectura unificada). La comunicación OPC UA otorga mecanismos de suscripción permitiendo una integración eficiente y sencilla de los recursos existentes en distintos dispositivos. Haciendo uso de este protocolo proponen entonces aplicar mecanismos de suscripción para obtención de datos de una planta industrial por medio de la aplicación de arquitecturas de bajo costo e independiente al funcionamiento de la planta.

Igualmente, puede concluirse de lo dicho por García, Irisarri y Perez (2017) que todos estos desafíos conducen a una nueva generación de procesos de automatización, basados en los sistemas de producción conectados globalmente para generar un sistema flexible de componentes cibernéticos de producción físicos, los cuales requieren la adquisición de datos del sistema y procesamiento inteligente de información para mejorar el rendimiento general del sistema. Uno de los proyectos de inclusión de estos sistemas consiste en utilizar un bloque de funciones para intercambiar datos entre el suelo de la planta y las capas de estándar industria, este ofrece mecanismos de suscripción integrando varios recursos residentes en la planta. Por lo cual se adquiere información de ella a bajo costo.

La industria 4.0 tiene múltiples aplicaciones una de ellas por ejemplo es un prototipo de sistema de vigilancia como bien lo referencia el prototipo formulado por Betancourt y Gómez (2015), donde el objetivo principal es capturar imágenes del lugar y permitir el acceso a ellas en tiempo real y de forma remota estableciendo una comunicación medio-objeto y objeto-objeto para lograrlo. Por su parte la comunicación

medio-objeto del sistema permitió monitorear diferentes variables del entorno y sus cambios de estado a través de imágenes, para lo cual fueron utilizados cuatro sensores encargados de determinar el flujo de personas en el recinto junto con la presencia de gases en la atmósfera, estabilidad en la superficie y la existencia de contacto entre las personas y el prototipo. La lectura de los datos de los sensores se realizó por medio de un microcomputador Raspberry Pi, donde se estableció una comunicación entre el microcomputador y un dispositivo móvil, cuyo puente de conexión en la comunicación fue establecido por el servidor de almacenamiento. Por lo cual a través de la comunicación objeto-objeto, se realiza la sincronización de las imágenes, enviando así los datos multimedia a la nube informática (Dropbox, Google Drive y Amazon para este caso en específico) permitiendo ser consultada por el usuario en su SmartPhone o Tablet en tiempo real.

Además, parafraseando a Neves et al. (sf) suele proponerse la implementación de sistemas ciberfísicos como herramienta clave en la aplicación de la Industria 4.0, por lo cual suele proponerse también la creación de entornos de trabajo inteligentes basados en una arquitectura de conjunto donde los mecanismos sean capaces de realizar las interacciones entre el software y los agentes físicos, en una alta producción personalizada, otorgando así niveles de autonomía.

La inclusión de sistemas ciberfísicos también se hace presente en el desarrollo de sistemas de transporte modular y auto organizados de componentes cibernéticos como lo es presentado por Leitão y Barbosa (2015) en su proyecto, el cual consistió en la creación de mecanismos auto organizadores a través de reglas simples, las cuales permiten operar

el sistema dinámicamente, incluyendo propiedad, la posibilidad de añadir o retirar transportadoras y la posibilidad de aplicar algún cambio a la secuencia. De forma tal que los mecanismos auto-organizadores mantiene la funcionalidad y dinamismo del sistema automáticamente sin necesidad de detener, reprogramar o incluso reiniciar el sistema.

Concordando con Oliveir (2017) en cuanto al contexto de la fabricación, el seguimiento de los cambios en las estrategias de producción enfocadas en el uso de tecnologías de la información ha desarrollado el concepto de fabricación remota y de su misma distribución a través de la red. Por lo cual es posible vincular los estudios relacionados con la fabricación electrónica con los primeros movimientos hacia el desarrollo de industrias inteligentes, orientadas principalmente a la industria 4.0. Uno de los mayores retos es entonces la adquisición de datos en una planta industrial, la cual se encuentre en su mayoría compuesta por maquinaria CNC con diferentes tecnologías además de fabricantes, pues es necesario desarrollar un sistema de comunicación aplicable a todas ellas. Por ende, han sido implementados sistemas cliente-servidor basado en Internet para realizar monitoreo y tele operación máquina-herramienta, donde se muestran los atributos de acuerdo con la estrategia de la industria 4.0.

Para finalizar y tomando lo dicho por Torre (2016) la Industria 4.0 ha afectado principalmente a los procesos de automatización por eso es más recurrente su implementación en este sector, por ejemplo, se realizó un prototipo cuyo fin principal es automatizar todas las líneas de transporte de materia prima y la apertura de las raseras. Se implementa entonces un sistema SCADA basado en tecnologías web, con lo cual el

operario podrá monitorear en cualquier momento y lugar las distintas operaciones que se estén realizando en la fábrica.

1.2 Descripción y Formulación del Problema.

Son muchos los docentes conscientes de la importancia existente en las prácticas de materias de corte científico-técnico en la formación de estudiantes, los cuales demandan nuevas formas de acceso a los elementos que brindan los laboratorios de experimentación como resalta Vázquez (2016), en su artículo.

Las mayores limitantes para la mantener una alta variedad de espacios para prácticas estudiantiles suelen ser los factores: económicos, espaciales y temporales. El factor económico, dado que, estos espacios de práctica representan un elevado costo para los centros educativos tanto para su adquisición como para el mantenimiento del equipo, haciendo difícil para las instituciones mantenerse renovando constantemente sus elementos de trabajo. En cuanto al factor espacial, es complicado ofrecer la cantidad de puestos de trabajo en los laboratorios estudiantiles que satisfagan la necesidad de otorgar a cada uno de los estudiantes un módulo de trabajo individual para que pueda familiarizarse por sí mismo, y el factor temporal, ya que el acceso a laboratorios es restringido a la presencia de un docente acompañante en el aula, por lo cual suelen permanecer la mayor parte del tiempo cerrados al acceso del público.

En la actualidad, la industria se encuentra experimentando grandes cambios por lo cual el estudiante de ingeniería debe estar capacitados para hacerles frente y ser un profesional competente en la aplicación de sistemas ciberfísicos y la implementación de

espacios de trabajo en los ámbitos universitarios enfocados a la creciente industria 4.0. Por lo cual, se genera el siguiente interrogante: ¿Cómo implementar un sistema didáctico de industria 4.0 utilizando los módulos de trabajo del laboratorio de la Universidad de San Buenaventura, Sede Bogotá?

1.3 Justificación.

El uso de módulos PLC se identifica como una de las principales herramientas en automatización, pues otorga al mercado herramientas que dan soluciones para los requerimientos más sencillos como para aquellos con un mayor nivel de competitividad. El sector de la automatización es un ámbito tremendamente exigente el cual se mantiene en constante desarrollo, como lo es la aplicación de la industria 4.0.

Fortalecer el conocimiento de los profesionales en las instituciones de educación superior, es un factor clave para aumentar los niveles de competitividad internacional y acceder a un mayor nivel de expansión en el mercado a largo plazo. Estratégicamente, los programas de ingeniería necesitan que sus futuros profesionales identifiquen, conozcan y desarrollen diferentes prácticas tecnológicas entre los distintos modelos, y referencias existentes, generando así una visión lo suficientemente amplia para consolidar conocimientos competentes para el futuro en un ámbito laboral, como la implementación de sistemas didácticos de industria 4.0.

Las demandas del alumnado van creciendo día tras día y la necesidad del campo laboral donde los estudiantes se encuentren familiarizados con sus elementos de trabajo también crece exponencialmente, la posibilidad de obtener conocimientos prácticos sobre

la manipulación acerca de módulos PLC de control aplicados a industria 4.0 por medio de modelos de comunicación inalámbrica de redes, puede marcar la diferencia en los profesionales que surjan en un futuro del plantel.

1.4 Objetivos de la investigación.

1.4.1 Objetivo General

Integrar un sistema didáctico de Industria 4.0 utilizando los módulos PLC de la Universidad de San Buenaventura, Sede Bogotá.

1.4.2 Objetivos Específicos

- Diseñar una interfaz interactiva que permita conexiones rápidas como seguras entre el sistema didáctico y el operador.
- Elaborar un manual de operaciones para estudiantes como docentes sobre el modo de uso del sistema didáctico.
- Realizar un programa demo que demuestre la funcionalidad del sistema didáctico utilizando todos los módulos PLC.

1.5 Alcances y Limitaciones.

- Integrar módulos Siemens y Festo al sistema didáctico de Industria 4.0.
- La interfaz diseñada será de sencilla implementación.

- Los módulos ensamblados serán puestos a disposición del plantel universitario quienes decidirán si serán implementados en sus planes de trabajo o no.
- Se hará uso de las estaciones de trabajo existentes. (Procesamiento, Distribución, Manipulación y Clasificación)
- El sistema didáctico de Industria 4.0 a implementar cumplirá con 3 de los 5 niveles de aplicación: Comunicación entre dispositivos, almacenamiento como exportación de información y monitoreo. Los siguientes dos niveles no se aplicarán puesto que los implementos presentes en el laboratorio de automatización no contienen la tecnología necesaria para conectar continuamente a nube de trabajo por sus limitaciones de comunicación.
- Se realizará la construcción de un sistema didáctico de Industria 4.0 para ser modelos de prueba en la implementación dentro del laboratorio de automatización de la universidad.

Capítulo 2

Marco Teórico.

3.1 Industria 4.0.

La competitividad hoy que hoy por hoy enfrentan muchas de las compañías en pro de dar un servicio más rápido y participativo a sus clientes les ha incitado a indagar en nuevos protocolos tecnológicos, los cuales permitan mejorar la falta de competencia en muchos sistemas de manufactura para manejar grandes cantidades de información, quedándose atrás frente a la cuarta generación tecnológica, la industria 4.0.

Por lo que la industria 4.0 se convierte entonces en la implementación de sistemas de telecomunicación con procesos industriales para controlar en tiempo real la flota de maquinarias, supervisar el proceso como la calidad del producto por medio de lazos de retroalimentación y el control de posibles fallas en alguno de los componentes.

3.2 ToolBox Industria 4.0

Existe un elemento clave para comprender con mayor facilidad la industria 4.0, el cual es el cuadro de herramientas presentado por la Anderl y Fleischer (2016) en su Guía de Industria 4.0 y que puede visualizarse en las siguientes figuras. (Figura 1 y Figura 2) El toolbox permite diferenciar los distintos niveles de aplicación de la industria teniendo en cuenta la relevancia del producto y su proceso de producción. En ella pueden visualizarse cinco niveles de aplicación y el desarrollo secuencial de cada uno de ellos, convirtiéndose en el punto clave para clasificar los campos que se ofrecerán con respecto a esta industria.

Es necesario aclarar que para que exista la industria 4.0 no debe cumplirse con las 5 etapas, sin embargo, los niveles aplicativos a incluir deben tener un desarrollo acorde entre ellos. Ejemplo, un proyecto sin etapa de comunicación no puede tener una etapa de control a distancia u otorgar servicios relacionados con la producción vía red pero un proyecto con comunicación no debe estar supeditado al uso de una nube de información para controlar los datos para ser catalogado de industria 4.0.

La primera sección del cuadro de herramientas corresponde al producto referenciando el desarrollo de nuevas ideas innovadoras entre la industria. Con lo cual son fácilmente identificable las herramientas necesarias dentro de la industria 4.0.

Products					
Integration of sensors / actuators					
	No use of sensors/actuators	Sensors / actuators are integrated	Sensor readings are processed by the product	Data is evaluated for analyses by the product	The product independently responds based on the gained data
Communication / Connectivity					
	The product has no interfaces	The product sends or receives I/O signals	The product has field bus interfaces	The product has Industrial Ethernet interfaces	The product has access to the internet
Functionalities for data storage and information exchange					
	No functionalities	Possibility of individual identification	Product has a passive data store	Product with data storage for autonomous information exchange	Data and information exchange as integral part
Monitoring					
	No monitoring by the product	Detection of failures	Recording of operating condition for diagnostic purposes	Prognosis of its own functional condition	Independently adopted control measures
Product-related IT services					
	No services	Services via online portals	Service execution directly via the product	Independently performed services	Complete integration into an infrastructure of IT services
Business models around the product					
	Gaining profits from selling standardized products	Sales and consulting regarding the product	Sales, consulting and adaption of the product to meet customer specifications	Additional sale of product-related services	Sale of product functions

Figura 1.1. Cuadro de herramientas Industria 4.0 – Productos. (VDMA German Engineering Federation, 2016)

La Figura 1 presenta 6 niveles de aplicación con base a los productos a utilizar durante la implementación de la industria. Ordenándolos de la siguiente manera.

- Integración de sensores y actuadores.

- Comunicación y conectividad.
- Funcionalidad de almacenamiento de datos e intercambio de información.
- Monitoreo.
- Servicios de TI relacionados con los productos.
- Modelos de negocio alrededor del producto.

Mientras que por su parte el cuadro de herramientas (Figura 2) referente a la producción resalta los siguientes niveles de aplicación de procesos a aplicar durante la industria 4.0.

- Procesamiento de datos en producción.
- Comunicación máquina a máquina.
- Redes de conexión con la producción.
- Infraestructura TI para la producción.
- Interfaces humano-máquina.
- Eficiencia para lotes pequeños.

3.3 Protocolo EasyIP.

Es un protocolo exclusivo de la empresa Festo, permite la conexión de dispositivos por medio de puertos Ethernet, y es programable a partir de las bibliotecas existentes en el software de programación CodeSys 2.3, identificado con el nombre Festo_EasyIp.Lib. Este protocolo de información cuenta con un bloque de programación que cuenta con las siguientes variables:

- TableIndex = Indica sí el número de IP del esclavo ha sido guardado.
- OpType = Identifica el tipo de datos a comunicar.

- NumToSend = Equivale al número de datos enviados.
- OpOffLoc = Desfase de lectura de los datos.
- OpOffRun = Desfase de envío de los datos.
- Status =Indica sí la comunicación fue exitosa. Es necesario que su valor sea diferente a -1.
- Easy_S = Lee y diagnostica el estado actual de operación en modo online a través de un ordenador.

3.4 UDP OpenService.

Se define con la finalidad de abrir el canal de comunicación por medio del puerto destinado para la recepción de datos.

3.5 UDP OpenSend.

Se define para abrir el canal de comunicación por medio del puerto determinado para el envío de datos.

Production					
Data processing in the production					
	No processing of data	Storage of data for documentation	Analyzing data for process monitoring	Evaluation for process planning / control	Automatic process planning / control
Machine-to-machine Communication (M2M)					
	No communication	Field bus interfaces	Industrial ethernet interfaces	Machines have access to internet	Web services (M2M software)
Company-wide networking with the production					
	No networking of production with other business units	Information exchange via mail / telecommunication	Uniform data formats and rules for data exchange	Uniform Data formats and inter-divisionally linked data servers	Inter-divisional, fully networked IT solutions
ICT infrastructure in production					
	Information exchange via mail / telecommunication	Central data servers in production	Internet-based portals with data sharing	Automated information exchange (e.g. order tracking)	Suppliers / customers are fully integrated into the process design
Man-machine interfaces					
	No information exchange between user and machine	Use of local user interfaces	Centralized / decentralized production monitoring / control	Use of mobile user interfaces	Augmented and assisted reality
Efficiency with small batches					
	Rigid production systems and a small proportion of identical parts	Use of flexible production systems and identical parts	Flexible production systems and modular designs for the products	Component-driven, flexible production of modular products within the company	Component-driven, modular production in value-adding networks

Figura 2. Cuadro de herramientas industria 4.0 – Producción. (VDMA German Engineering Federation, 2016)

Capítulo 3

Diseño Ingenieril.

3.1 Diseño del sistema didáctico industria 4.0.

El modelo comprende un conjunto de tecnologías en las cuales se propondrá la implementación de un sistema didáctico de industria 4.0, desde su identificación y diseño, hasta su automatización, monitorización y mejora continua. Constituye un sistema de trabajo enfocado a perseguir la mejora continua del funcionamiento de las actividades del dispositivo mecatrónico, mediante la identificación, selección y mejora de las distintas etapas de proceso. Se pueden diferenciar 4 fases en este modelo:

- A. Análisis de Procesos:** Analizar los procesos actuales para obtener los conocimientos necesarios, con el fin de definir los distintos requerimientos (definición de tareas, como ejecutar dichas tareas, quien realiza las tareas, donde se realizan, que datos utiliza, que reglas de manipulación de alimentos deben cumplirse).
- B. Diseño de Procesos:** Diseñar las fases del proceso que permitan realizar una tarea específica, con el fin de hacer la integración completa de todos los equipos.
- C. Ejecutar los procesos:** Construcción las fases del proceso que cumplan el objetivo del proyecto
- D. Monitorear las actividades:** Toma y registro de datos de las diferentes fases del proceso, para realizar una comparación con el proceso industrial existente, con el objetivo de encontrar las fallas y las ventajas que tendría el sistema mecatrónico sobre el proceso actual.

3.1.1 Ubicación de módulos de trabajo.

El laboratorio de la universidad de San Buenaventura, sede Bogotá cuenta en su totalidad con 5 módulos MPS (Sistema de Producción Modular) sin embargo, para el sistema didáctico de industria 4.0 a proponer se escogieron solamente 4 estaciones de trabajo: distribución, procesamiento, manipulación y clasificación.

La estación de verificación decide no utilizarse por tres factores principales. El primer factor es la limitación de integración de dicha estación, pues para poder implementarse debe ser precedida por la estación de distribución por especificación Festo. Como segundo factor, las estaciones de verificación y clasificación están planeadas para ser implementadas como estaciones finales en los sistemas didácticos, siendo necesario escoger tan sólo a una de ellas para su aplicación en el sistema didáctico, la estación de clasificación por su parte permite conocer una mayor cantidad de datos acerca de la pieza a transportar como su color y material, mientras la estación de verificación solo permite saber si la pieza se encuentra defectuosa. El tercer y último factor tiene relación con la estructura de la estación de verificación pues en la parte final del tobogán de entrega existe un freno mecánico el cual no permite al Robotino tener acceso para tomar la pieza y lograr transportarla a la siguiente estación del sistema.

Se decide por lo tanto que el sistema didáctico tendrá el siguiente orden: Distribución, procesamiento, Robotino, manipulación y clasificación. (Figura 3)

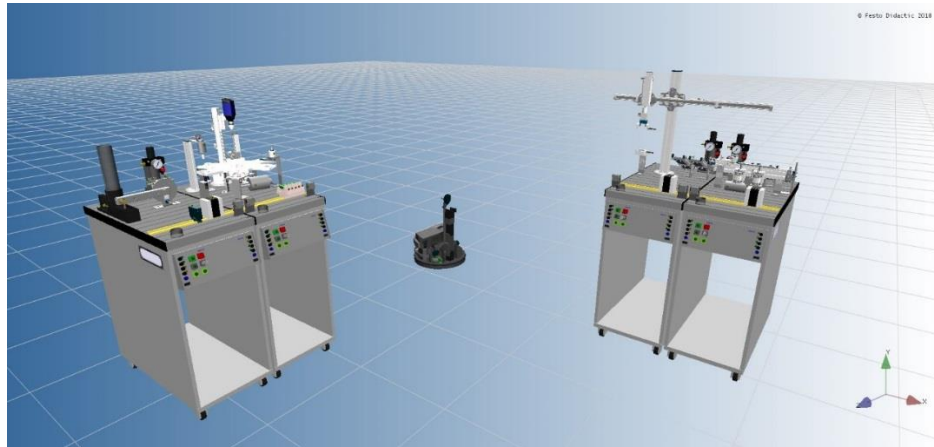


Figura 3. Diseño ubicación de estaciones y robotino a implementar para el Sistema didáctico de industria 4.0.

La estación de distribución es seleccionada como estación de inicio pues no cuenta con un emisor el cual informe de su estado, sin embargo, si cuenta con un receptor el cual le permite reconocer si la siguiente estación se encuentra ocupada. Igualmente, ninguna de las demás estaciones cuenta con una manera de distribución para colocar las piezas dentro del magazine.

Se decide que la segunda estación de trabajo sea procesamiento ya que el transporte de la pieza conserva una altura uniforme durante todo el proceso, siendo mucho más sencillo para el robotino lograr aferrar las piezas.

Por otra parte, la implementación del robotino corresponde a un servidor logístico para lograr una integración mucho más completa entre estaciones, además, se visualiza con mayor facilidad la conexión de sistemas referidos dentro de la industria 4.0, integrando en este caso también un sistema de robótica móvil. Es importante resaltar que el robotino funcionará como un esclavo de la segunda estación, por lo cual depende de

sus órdenes para movilizarse. Cuenta también con dos sensores inductivos, encargados de permitir su alineación con las estaciones de trabajo y un sensor óptico, que identificará sus puntos de parada.

Mientras la estación de manipulación funciona como un primer filtro de clasificación aplicando así la toma de decisiones a partir del análisis de datos; uno de los componentes principales de la industria 4.0. Simultáneamente, incluye la comunicación máquina a máquina puesto que se le pone en contacto con la primera estación por sí es necesario solicitar una pieza extra durante el proceso.

Como se mencionó con anterioridad, la estación de clasificación está pensada desde un inicio como la estación final además tienen la posibilidad de generar bases de datos gracias a su arquitectura, haciendo posible tener un historial de información el cual pueda visualizarse en la interfaz usuario-máquina.

El resultado real es el siguiente:



Figura 4. Ubicación real del sistema didáctico de industria 4.0.

3.1.2 Interfaz del sistema didáctico de industria 4.0.

La intención del sistema didáctico es que sea de fácil operación, amigable para el usuario y de acceso rápido para el usuario, por lo que se implementa una interfaz interactiva haciendo uso de la pantalla Fed-550.

En el momento de ser encendida desplegará una pantalla inicial, donde se ubicarán el nombre del sistema, el logotipo de la universidad y un botón de inicio el cual arrojará al usuario a un menú principal que le permitirá acceder a cualquiera de las estaciones del sistema e ingresar a cada una de ellas según lo desee. Las 4 estaciones cuentan con las siguientes opciones:

- **START:** Arrancar el programa principal de la estación. El indicador que se encuentra en la esquina superior izquierda del botón, se pone amarillo cuando la estación solicita que se oprima el botón START. Usualmente este indicador se iluminará después de que la estación se ha reseteado de manera correcta o de que la estación se encuentre en modo manual y haya culminado un ciclo, ya que, en este caso, se requerirá volver a oprimir el botón START para comenzar un nuevo ciclo.
- **RESET:** Devolver tanto a el programa principal al inicio y a los actuadores de la estación a condiciones iniciales. El indicador que se encuentra en la esquina superior izquierda del botón, se pone amarillo cuando la estación solicita que se oprima el botón RESET. Usualmente este indicador se iluminará después de que se ha detenido el programa usando el botón STOP o apenas la estación se encienda, ya que, se debe

asegurar que la estación arranque en condiciones iniciales cada vez que se va a correr el programa principal.

- **STOP:** Detener el programa principal de la estación independientemente del paso que se esté ejecutando en ese momento.
- **AUTO/MAN:** Selector que determina el modo en el que está la estación. La estación por defecto, siempre empieza en modo automático. En el modo automático, la estación ejecuta ciclos de producción de manera indefinida sin la necesidad de otra instrucción aparte de producir órdenes pendientes. En el modo manual, se requiere oprimir el botón START cada vez que la estación llevo a cabo un ciclo de producción.
- **Botones en referencia a las demás estaciones:** Cada una de las ventanas principales de las estaciones contará con un listado de botones con el nombre de las demás, el cual llevará al usuario a la ventana principal de cada una de ellas. La Figura 4, evidencia un ejemplo de ello.



Figura 5. Ventana principal estación Distribución.

- **I/O:** Lleva a la ventana de visualización de entradas y salidas digitales de la respectiva estación.
- **INICIO:** Lleva a la ventana de selección de estación.
- **Entradas y Salidas según la estación:** En esta ventana podrá visualizar en tiempo real el estado de las entradas y salidas digitales de la estación, en tiempo real.

La única ventana de control que contará con un botón adicional será la estación de Manipulación, el cual será **CLASIFICAR**, cuyo impacto se verá evidenciado en la secuencia demo a implementar en el sistema didáctico. (Ver capítulo 5)

La interfaz permitirá también tener acceso a una ventana de registro de piezas donde se visualizará el número de piezas producidas y descartadas, su color, total de piezas, detalle de cada registro y el reinicio del conteo.

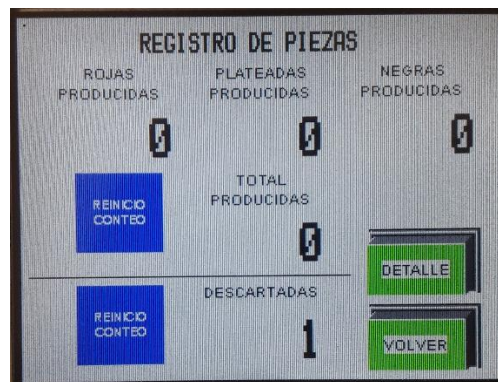


Figura 6. Ventana de Registro de Piezas.

3.1.3 Diseño Ingeniril.

Para el correcto funcionamiento de lo anteriormente mencionado, fue necesario implementar al sistema un router principal, el cual, tiene como función principal crear una red interna para uso exclusivo del sistema, debido a que a lo largo de las pruebas y los desarrollos realizados usando la red universitaria, fue posible evidenciar que dicha red era insuficiente al ser inestable y ser de uso libre. Dicho percance se identificó cuando el desarrollo de las pruebas de comunicación entre los PLC coincidió con una clase, donde debían asignar direcciones IP aleatorias a otros sistemas PLC, resultando en un daño en la comunicación ya existente entre el sistema a prueba. Con esto dicho, la forma de crear la red privada fue la siguiente: primero se adquirió un router el cual genera la red inalámbrica utilizada por el sistema didáctico de industria 4.0, posteriormente se agregaron dos **Access Points**, cada uno configurado como clientes del router principal. A su vez, a cada uno de estos, se le agregó un Switch con la finalidad de aumentar la cantidad de puertos Ethernet de cada Access Point y haciendo posible conectar todos los dispositivos.

Este paquete de dispositivos se configuró dos veces, ya que cada uno de estos grupos fue ubicado justo debajo de las placas perfiladas de cada uno de los dos conjuntos de estaciones MPS, dentro de los vagones donde son sostenidas las mismas estaciones. Con esta configuración de dispositivos, quedó asegurado que cada uno de los equipos a comunicar en este sistema, tuviesen una conexión con la red principal, generada por el router. Igualmente, el Robotino cuenta de fábrica con un dispositivo de red propio el cual puede ser configurado ya sea como Access Point o como Cliente de otra red, en este caso,

se ha configurado al igual que los otros dos Access Points, como un cliente de la red principal para asegurar que el Robotino también tenga una dirección asignada dentro de la misma y así pueda comunicarse libremente con el resto de controladores.

El siguiente inconveniente que surgió fue lograr que el robotino interactuara de manera cooperativa con las estaciones y con la posible existencia de un obstáculo en su camino (una persona probablemente). Para esto, se requirió diseñar un soporte el cual le diera la altura suficiente a una pinza o “gripper” para alcanzar las piezas en la salida de la estación de Procesamiento y llevarlas a la entrada de la estación de Manipulación; una vez la parte mecánica fue cubierta, con la ayuda de un pequeño sistema de control para el servomotor de la pinza y un relé que convirtiera una señal de 24VDC provenientes de una salida digital del Robotino (DO1 en este caso según el programa del Robotino) la cual se puede identificar en las fichas técnicas brindadas por Festo o en la sección de ayuda dentro del software de programación de RobotinoView2, a los 5VDC que utiliza tanto el servomotor como su respectivo sistema de control como entrada digital.

Por el otro lado, para lograr la cooperación entre el Robotino y un obstáculo, en su programa se agregaron 9 sensores infrarrojos de proximidad con los que el Robotino cuenta de fábrica para detenerse momentáneamente en caso de que algún objeto sea detectado a medida que se está efectuando alguno de sus recorridos.

Finalmente, para poder comunicar la pantalla táctil al resto del sistema, se instaló justo al lado del Router principal con la intención de poderla conectar directamente con un cable RJ45 o cable Ethernet de configuración directa, y se le asignó una dirección IP perteneciente a la misma de la del resto del sistema.

3.2. Protocolos de comunicación y programación interfaz.

3.2.1 Comunicación entre estaciones de Distribución y Procesamiento.

El protocolo de comunicación utilizado en este caso es el EasyIP, el cual permite la comunicación entre dispositivos por medio del puerto Ethernet (LAN/WLAN). Se genera un programa con un llamado a la función Easy_SetIP, el cual se encargará de almacenar el número IP de la estación esclava, en este caso la estación de procesamiento, identificada como 192.168.106.155.

Igualmente se implementa una variable booleana definida como una función Blink, cuyo tiempo de encendido y apagado corresponderá a 100 ms, la cual se encargará de actualizar el envío de datos de envío al estar activa. Es muy importante que el protocolo se mantenga activo de forma cíclica.

Las especificaciones internas del bloque de comunicación EasyIP corresponderán entonces a las siguientes:

- TableIndex = 1
- Optype = Word.
- NumtoSend = 5
- OpOffloc = 1006, elegido para que no presente interferencia de datos.
- OpOffRun = 1006
- Status = 2000, así no interferirá con ninguna variable.
- EasyS = Word.

El valor de estas variables debe ser el mismo en ambas estaciones, para que el sistema pueda funcionar sin interferencias.

3.2.2 Comunicación entre estación Procesamiento y Robotino.

Para este caso se utiliza el protocolo de comunicación UDP (User Datagram Protocol), el cual consiste en enviar datagramas de datos a través de la red LAN/WLAN a un dispositivo en permanente lectura. Primero, es necesario crear un programa inicial con el objetivo de reiniciar la información almacenada en ambos dispositivos cada vez que sean encendidos, posteriormente, se crea una estructura de datos nombrada en este caso **“Param_Robotino”**, en donde se actualizarán los datos de comunicación, envío y recepción.

Se implementan después dos subprogramas: el primero UDPConsumer, en donde se identifica el número de puerto a utilizar, 5050 en esta ocasión y el valor del número IP del Robotino 192.168.106.99, el cual es el dispositivo esclavo y el segundo, es el subprograma UDPProducer, donde se habilita la variable SendBit encargada de recibir órdenes.

En cuanto a la programación del protocolo en el robotino, desde el software RobotinoView2 de Festo, se agrega la función **“Intercambio de datos UDP”** y se configuran de nuevo las propiedades como el número de puerto y número de IP el cual en este caso corresponde al valor de la estación de procesamiento. (Ver numeral 4.1)

3.2.3 Comunicación entre estaciones Distribución y Manipulación.

De nuevo se utiliza el protocolo de comunicación UDP, en este caso la variable Station N equivale a 20.

Se implementan los dos subprogramas mencionados con anterioridad. En el subprograma UDPProducer se define el número IP de la estación de Manipulación 192.168.106.150 junto con el número de puerto asignado, 2044; mientras en el subprograma UDPConsumer se configuran las propiedades de forma tal sea capaz de recibir los datos de las demás estaciones, además de agregar una variable llamada “**Pedir_Pieza**” con la intención de hacer control de calidad. (Ver numeral 5.4)

En cuanto a la programación del PLC Siemens de la estación de Manipulación, se crea un proyecto principal donde se implementa el bloque **TCON**, con el cual es posible crear una base de datos donde se almacenen de forma permanente los datos que lleguen a la estación de Distribución, y se crea también una segunda base de datos incluyendo la columna parámetros pre configurada por Siemens, en donde se encuentra la información requerida para el correcto funcionamiento del protocolo de comunicación.

Igualmente, es necesario implementar el bloque **TUSEND**, la cual generará una nueva base de datos encargada de enviar la información recolectada a la estación de Distribución, la cual se encargará de enviarlos a la interfaz del sistema didáctico.

3.2.4 Comunicación S7 entre estación Manipulación y Clasificación.

A continuación, se muestra la respectiva comunicación entre ambas estaciones para poder enviar las variables necesarias a la estación Clasificación y así mismo, lograr devolver la información diagnóstico entregado por la estación, para su posterior visualización en la interfaz del sistema didáctico.

Se crea como punto inicial una nueva conexión entre las estaciones, seleccionando el tipo de conexión **S7_Connection** entre los puertos ProfiNet (Ethernet) de cada uno de los PLC.

Para este protocolo de comunicación es necesario utilizar los bloques “*Get*” y “*Put*” los cuales son los encargados del envío y recepción de datos desde otra CPU como sucede en este caso. La estación manipulación actúa como comunicador, representando así una ventaja puesto que primero se crea la red de comunicación y luego si se enciende la estación con la cual comunica.

Por su parte el bloque “*Get*” tiene como función buscar dentro de los registros de la estación Clasificación los datos que hayan sido configurados en el bloque. Para este caso en específico se han configurado la dirección DBX5.0 en adelante, los siguientes 60 Bytes de información de la base de datos DB8 en el PLC remitente y sean almacenados desde la dirección DBX2.0 los siguientes 60 Bytes de información de la base de datos DB1 del PLC local, cada vez que la variable M0.0 (la cual es similar a la variable Blink programada en los PLC Festo) vuelve a tomar el valor TRUE.

En el caso del bloque “*Put*”, básicamente es la misma configuración con diferencia que ahora son tomados los datos del PLC local y son inyectados a través de la red en la base de datos configurada en el PLC Receptor (Estación Clasificación en este caso).

3.2.5 Programación interfaz.

Por último, tras haber hecho todas las anteriores configuraciones de comunicación se procede a conectar todas las variables almacenadas hasta el momento y distribuidas en las demás estaciones gracias a la estación Distribución, a la pantalla FED-550 para poder modificarlas y leer en tiempo real.

En el proyecto a cargar a la pantalla en el software Designer, se procede a configurar el protocolo de comunicación con el PLC de la estación Distribución, insertando el archivo Symbol proveniente del PLC en los “*tags*” del sistema.

Posteriormente se configura el controlador a comunicar la pantalla, seleccionando el protocolo de “*CoDeSys ETH*” (CoDeSys ETH), configurando entonces que la elección TCP/IP Level 2 Route se encuentre habilitada para que el límite de dispositivos conectados no corresponda únicamente a 2. Se ingresa igualmente la dirección IP de la estación Distribución 192.168.106.141 y se asegura que el puerto por conexión equivalga a 1200, el cual es el puerto por defecto de los PLC para este protocolo de comunicación.

Con el controlador configurado se procede entonces a cargar los tags creados anteriormente al proyecto, importando el archivo *GLOBAL* y el diccionario *LOCAL* creados para el proyecto.

Ya con todas las configuraciones previas realizadas se procede a configurar cada uno de los elementos que van a hacer parte del proyecto visual de la pantalla, y posteriormente en su configuración se asigna el tag correspondiente.

3.3 Programación secuencia demo.

Para evidenciar el correcto funcionamiento entre la comunicación de las estaciones y comprender mejor el concepto de industria 4.0 en el sistema didáctico, se programa una secuencia demo que involucre tanto a las estaciones como al Robotino, en este caso el sistema didáctico se encargará de la verificación de piezas por color y material, se usarán tres tipos de piezas: negras, rojas y metálicas.

3.3.1 Pantalla de Inicio.

Funcionará como la interfaz de control del sistema didáctico. A partir de ella, el usuario podrá visualizar el funcionamiento de cada una de las estaciones, reconocer las entradas y salidas, además, será la interfaz a través de la cual se indicarán la cantidad piezas a trabajar durante cada proceso; para lo cual el número seleccionado se cargará en la variable **CARGAR_PIEZAS** y será enviado a la estación de distribución.

3.3.2 Estación de Distribución.

Es la estación maestra del sistema didáctico, se encarga de distribuir (como indica su nombre) las piezas aleatoriamente desde el magazine y las transporta hasta la estación de procesamiento. (Ver Anexo 1)

3.3.3 Estación de Procesamiento.

Se encarga de censar la pieza y se indica un estado ocupado para la estación maestro, luego procede a transportarla a un proceso de verificación, donde se comprueba que la posición en la cual llega la pieza sea la correcta, es decir, que la cara agujerada de la pieza sea su cara superior, si es así entonces se procederá a taladrar la pieza y finalmente se trasladará al punto final de la estación. Al ubicarse allí, la estación enviará una señal al Robotino dando aviso que la pieza se encuentra lista para recogerse y ser transportada a la estación de manipulación. (Ver Anexo 2)

La estación de procesamiento funcionará como maestro para el Robotino, por lo tanto, será la única estación que podrá ordenarle.

3.3.4 Robotino.

El robotino se ubicará teniendo en cuenta los caminos delimitados para él en el área de trabajo por medio de odometría. Al llegar a la pieza, la asirá por medio del gripper instalado y procederá entonces a transportarla a la estación de manipulación.

3.3.5 Estación Manipulación.

El funcionamiento de esta estación manipulación se define según la selección del usuario por medio de la interfaz del sistema didáctico. Si el usuario selecciono en la interfaz la opción **Clasificación**, la estación de manipulación discriminará las piezas por medio del sensor óptico ubicado en el gripper de la estación, gracias a la reflexión, si la

pieza es negra no existirá reflexión de luz al sensor óptico. En el momento en que la estación identifica una pieza de este color, la relacionará con una pieza defectuosa y la enviará al tobogán de descarte además enviará una señal a la estación de distribución solicitando una pieza más para así completar el lote, mientras que sí se identifica una pieza roja o metálica será entregada a la banda transportadora de la estación de clasificación.

Sí la opción de **Clasificación** no se encuentra activa en la interfaz, la estación simplemente actuará como una estación de transporte entre el Robotino y la última estación. (Ver Anexo 3)

3.3.6 Estación Clasificación.

Al momento de la pieza ser despachada en la banda transportadora, la estación enviará una señal de estado ocupado a la estación maestra, posteriormente transportará la pieza al punto de verificación de material y color, donde por medio de un sensor óptico y un sensor inductivo identificará cada una de las piezas.

Sí ninguno de los sensores se activa, la pieza es negra, será enviada al tercer tobogán de la estación. Si el sensor óptico se activa, pero el sensor inductivo permanece inactivo, la pieza es roja, será enviada por el primer tobogán. Si tanto el sensor óptico como el inductivo se activan, la pieza será metálica, será enviada por el segundo tobogán.

Su mayor importancia en el sistema didáctico yace en la recolección de información que hace la estación, estos datos los enviará primero a la estación de manipulación, quien los enviará a la estación maestra de distribución y esta última a la

pantalla para almacenarla y permitir al usuario visualizar el historial de piezas despachadas, así como antiguos registros y el conteo de piezas. (Ver anexo 4)

Capítulo 4. Análisis de Datos

A continuación, se presentan los resultados de las comunicaciones planteadas para la utilización del sistema, habiéndose cargado el programa demo del mismo. La intención de este capítulo es evidenciar detalladamente como los datos son enviados desde alguno de los controladores y como se logra la recepción en otro, dependiendo de los dispositivos incluidos en cada una de las comunicaciones.

4.1 Flujo de datos entre estaciones Distributing y Handling por medio de UDP

En la Figura 7 a continuación se muestra en los recuadros rojos que dentro del programa de la estación Handling, que existen dos bases de datos, una con el nombre de Received_Data la cual identifica los datos que están siendo recibidos por medio de UDP provenientes de la estación de Distributing, y otra denominada Send_Data la cual muestra los datos que serán enviados a dicha estación.

The image shows two side-by-side windows from the Siemens SIMATIC Manager. The left window is titled 'Received_Data [D85]' and the right window is titled 'Send_Data [D88]'. Both windows display a table of data points with columns for Name, Data type, Offset, Start value, and Monitor value.

Received_Data [D85] Table:

Name	Data type	Offset	Start value	Monitor value
Static				
Recibido	Array[1..10] of Byte	0.0		
Recibido[1]	Byte	0.0	16#0	16#00
Recibido[2]	Byte	1.0	16#0	16#00
Recibido[3]	Byte	2.0	16#0	16#00
Recibido[4]	Byte	3.0	16#0	16#00
Recibido[5]	Byte	4.0	16#0	16#00
Recibido[6]	Byte	5.0	16#0	16#00
Recibido[7]	Byte	6.0	16#0	16#00
Recibido[8]	Byte	7.0	16#0	16#00
Recibido[9]	Byte	8.0	16#0	16#00
Recibido[10]	Byte	9.0	16#0	16#00

Send_Data [D88] Table:

Name	Data type	Offset	Start value	Monitor value
Static				
Enviados	Array[1..66] of Byte	0.0		
Enviados[1]	Byte	0.0	16#0	16#A2
Enviados[2]	Byte	1.0	16#0	16#80
Enviados[3]	Byte	2.0	16#0	16#01
Enviados[4]	Byte	3.0	16#0	16#00
Enviados[5]	Byte	4.0	16#0	16#01
Enviados[6]	Byte	5.0	16#0	16#50
Enviados[7]	Byte	6.0	16#0	16#00
Enviados[8]	Byte	7.0	16#0	16#00
Enviados[9]	Byte	8.0	16#0	16#00
Enviados[10]	Byte	9.0	16#0	16#00
Enviados[11]	Byte	10.0	16#0	16#00
Enviados[12]	Byte	11.0	16#0	16#00
Enviados[13]	Byte	12.0	16#0	16#00
Enviados[14]	Byte	13.0	16#0	16#01
Enviados[15]	Byte	14.0	16#0	16#00
Enviados[16]	Byte	15.0	16#0	16#00
Enviados[17]	Byte	16.0	16#0	16#00
Enviados[18]	Byte	17.0	16#0	16#00
Enviados[19]	Byte	18.0	16#0	16#00
Enviados[20]	Byte	19.0	16#0	16#00
Enviados[21]	Byte	20.0	16#0	16#00
Enviados[22]	Byte	21.0	16#0	16#00
Enviados[23]	Byte	22.0	16#0	16#00
Enviados[24]	Byte	23.0	16#0	16#00

Figura 7. Bases de datos dentro de estacion Handling para comunicación con Distributing.

Por el otro lado, en la Figura 8, se evidencian el correspondiente de las mismas dos bases de datos para el caso de la estacion Distributing, aclarando que en este caso los subprogramas que las contienen en CoDeSys 2.3 se denominan UDPProducer (“Produce” o envía datos con este protocolo) y UDPConsumer (“Consume” o recibe datos).

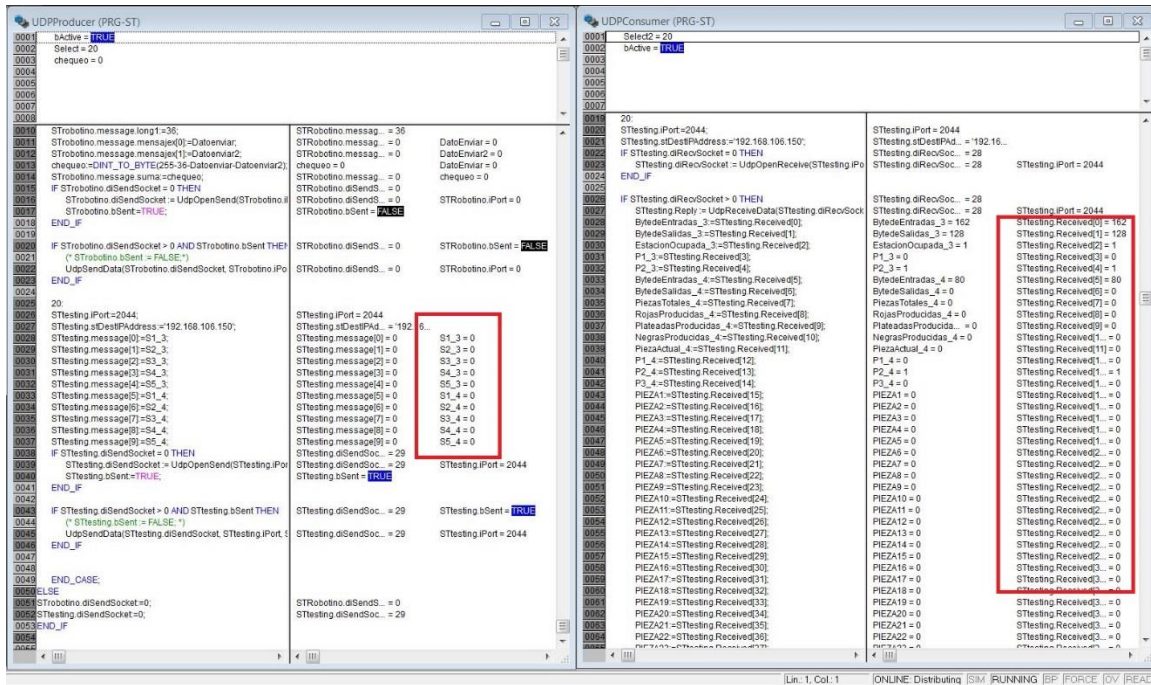


Figura 8. Bases de datos dentro de estación Distributing para comunicación con Handling.

Ahora bien, para la identificación de los cambios de datos y el análisis de la comunicación, lo que se hace es comunicarse a cada uno de los dispositivos en tiempo real a través de “debugging” o monitoreo de variables internas y tomar capturas de pantalla simultaneas con la intencion de mostrar como el cambio de valores de unas variables a un lado, se debe ver reflejado de la misma manera en el otro lado. Tomando lo encerrado en los recuadros rojos de la Figura 8, se ve como hay unos valores en un momento determinado en el recuadro de la derecha, por ejemplo, que empiezan con el valor 162 y seguido por el valor 128. Estos valores, junto con todos los demás que siguen en su respectivo orden hacia abajo, están siendo recibidos por medio de la red desde la estacion de Handling. A su vez, en la Figura 7, los valores dentro del tercer recuadro rojo comienzan con el valor A2 seguidos por el valor 80, los cuales corresponden a los

primeros dos valores mencionados, pero en notación Hexadecimal. En otras palabras, el valor 162 en notación decimal (como la muestra CoDeSys 2.3) es equivalente al valor A2 en notación hexadecimal (como es representado en TIA Portal en el caso de Siemens), y de igual manera, 128 en decimal, se representa como 80 en notación hexadecimal. Cabe aclarar que la representación no influye en el valor, ya que todos los valores realmente se están enviando y recibiendo en tipos de dato BYTE, sencillamente cada uno de los softwares los representan de la manera que el desarrollador lo haya determinado.

Name	Data type	Offset	Start value	Monitor value
Static	Array[1..10] of Byte	0.0		
Recibido				
Recibido[1]	Byte	0.0	16#0	16#00
Recibido[2]	Byte	1.0	16#0	16#00
Recibido[3]	Byte	2.0	16#0	16#00
Recibido[4]	Byte	3.0	16#0	16#00
Recibido[5]	Byte	4.0	16#0	16#00
Recibido[6]	Byte	5.0	16#0	16#00
Recibido[7]	Byte	6.0	16#0	16#00
Recibido[8]	Byte	7.0	16#0	16#00
Recibido[9]	Byte	8.0	16#0	16#00
Recibido[10]	Byte	9.0	16#0	16#00

Name	Data type	Offset	Start value	Monitor value	Retain
Static	Array[1..66] of Byte	0.0			
Enviados					
Enviados[1]	Byte	0.0	16#0	16#A2	
Enviados[2]	Byte	1.0	16#0	16#81	
Enviados[3]	Byte	2.0	16#0	16#01	
Enviados[4]	Byte	3.0	16#0	16#01	
Enviados[5]	Byte	4.0	16#0	16#00	
Enviados[6]	Byte	5.0	16#0	16#50	
Enviados[7]	Byte	6.0	16#0	16#00	
Enviados[8]	Byte	7.0	16#0	16#00	
Enviados[9]	Byte	8.0	16#0	16#02	
Enviados[10]	Byte	9.0	16#0	16#02	
Enviados[11]	Byte	10.0	16#0	16#09	
Enviados[12]	Byte	11.0	16#0	16#01	
Enviados[13]	Byte	12.0	16#0	16#00	
Enviados[14]	Byte	13.0	16#0	16#00	
Enviados[15]	Byte	14.0	16#0	16#00	
Enviados[16]	Byte	15.0	16#0	16#03	
Enviados[17]	Byte	16.0	16#0	16#02	
Enviados[18]	Byte	17.0	16#0	16#03	
Enviados[19]	Byte	18.0	16#0	16#03	
Enviados[20]	Byte	19.0	16#0	16#03	
Enviados[21]	Byte	20.0	16#0	16#02	
Enviados[22]	Byte	21.0	16#0	16#03	
Enviados[23]	Byte	22.0	16#0	16#03	
Enviados[24]	Byte	23.0	16#0	16#03	

Figura 9. Forzado de una variable en estacion Handling.

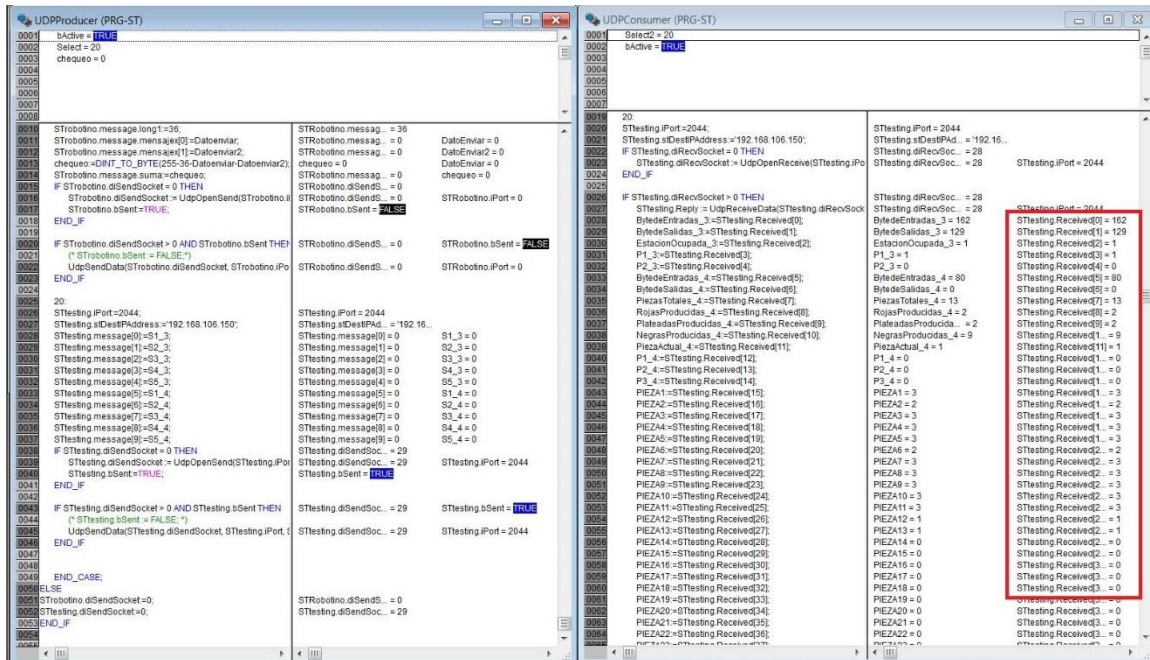


Figura 10. Respuesta al forzado de la variable en estación Distributing.

Para la segunda parte de la prueba, se hizo necesario ubicar una pieza en el buffer de entrada de la estacion Handling, con la intencion de que cambiara uno de los datos en la base de cada uno de los dos dispositivos. Razon por la cual, el segundo dato de la lista, en la Figura 9, el cual originalmente se encontraba con el valor 80 en Hexadecimal, ahora se ha convertido a 81, mostrando que el primer BIT de la variable, ha sido forzado del valor original 0 a un valor de 1 lógico, incrementando la variable en una unidad. En respuesta a lo anterior, en la Figura 10, se ve como el valor correspondiente almacenado en la variable con nombre Bytedesalidas_3, también ha aumentado su valor en una unidad, denotando su nuevo valor como 129.

Está claro que, para efectos de simplicidad en el documento, se ha decidido hacer el forzado de una sola variable, pero a lo largo de la ejecución del programa demo, se utilizan todas las demás variables que se encuentran en cada una de las bases de datos

para algún punto específico del mismo. Así que se puede comprobar la correlación de cualquiera de las variables que se muestran en las bases de datos, las cuales, según la respuesta de la estación, pueden haber cambiado al recibir el estímulo inicial, e igual habría coincidencia.

4.2 Flujo de datos entre estaciones Distributing y Processing por medio de Easy_IP

A continuación, se presenta la misma correspondencia de flujo de datos de una estación a otra, pero en el caso de las estaciones Distributing y Processing las cuales mantienen una comunicación constante entre ellos por medio del protocolo de comunicación interno de FESTO denominado Easy_IP. Por eso, en la Figura 11, se muestran las bases de datos de cada uno de los dos programas, en las que se encuentran las variables implicadas en este flujo de datos, las cuales están encerradas en recuadros rojos. En esta ocasión, no hay variación en la representación de los valores, gracias a que ambos programas están hechos en CoDeSys 2.3.

Finalmente, en la Figura 12, al igual que en el anterior caso, se ha forzado una de las entradas en la estación Processing para identificar como se modifica el valor de las variables en ambos dispositivos a partir de un cambio físico en una de las estaciones. Es importante resaltar que todos los datos que están siendo enviados entre estaciones, son posteriormente comunicados a la pantalla táctil para cumplir con la intención

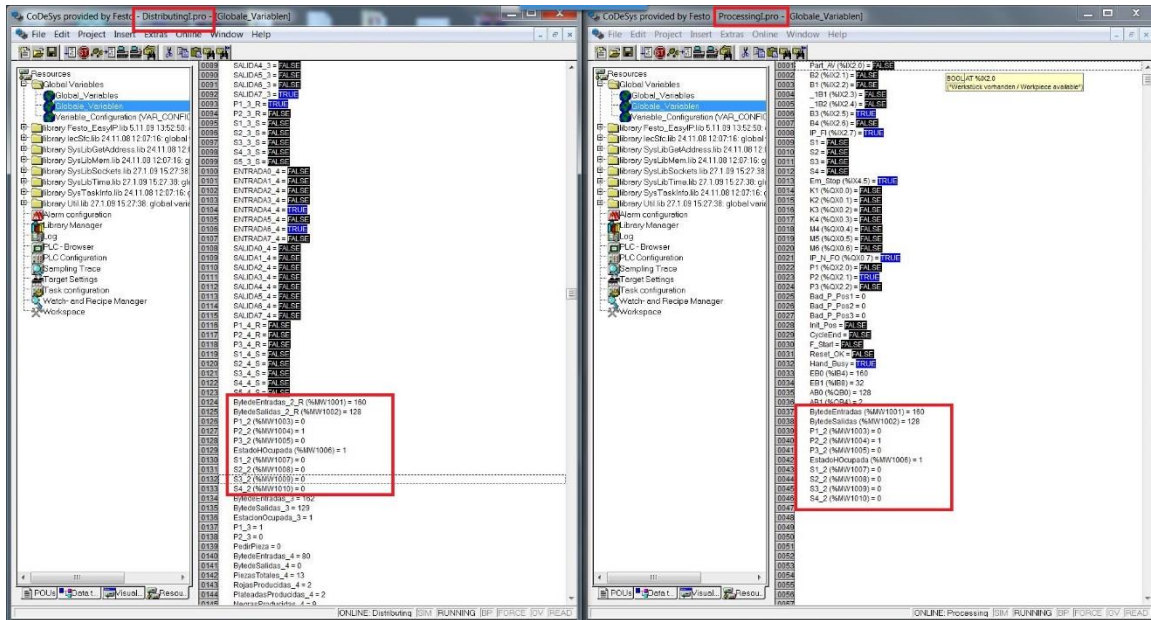


Figura 11. Bases de datos dentro de estaciones Distributing y Processing para comunicación con Easy_IP.

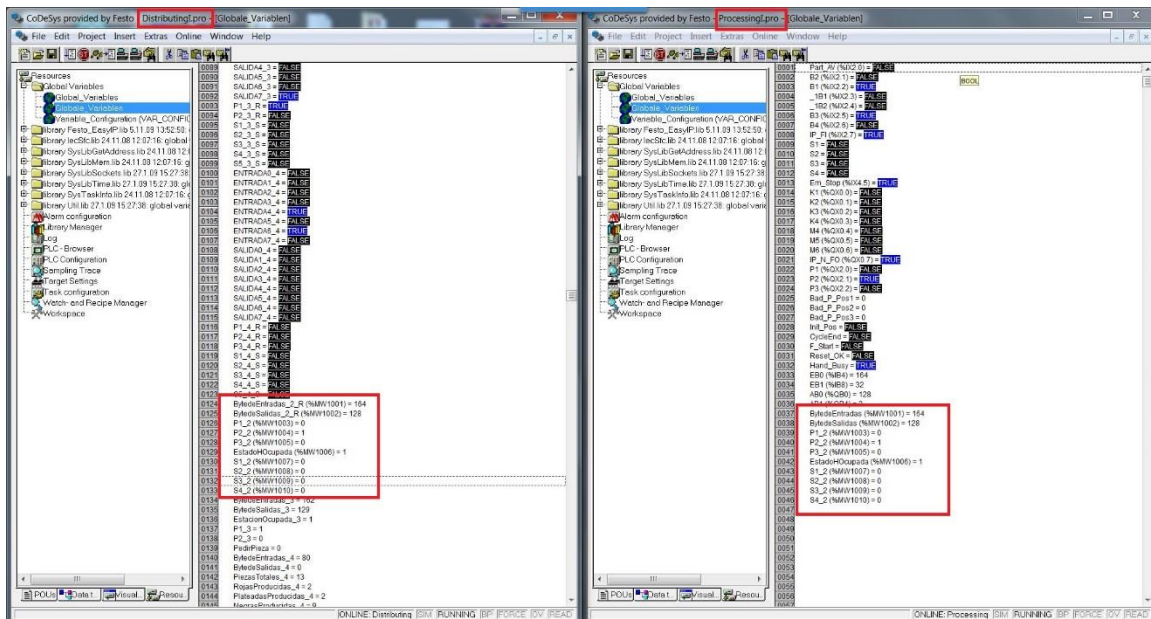


Figura 12. Bases de datos dentro de estaciones Distributing y Processing con una variable forzada.

4.3 Flujo de datos entre estación Processing y Robotino por medio de UDP

Para el caso de la comunicación entre la estación Processing y el Robotino, lo primero que se tiene en consideración para el análisis de resultados, es que internamente el Robotino no tiene un sistema de monitoreo de variables en tiempo real, por lo que para hacer el análisis de resultados, lo que se hace es mostrar cómo se establece la comunicación, y posteriormente como al enviar un dato interpretado como una orden, el Robotino a su vez responde a través de otras variables.

Lo primero entonces es determinar que al hacerse la inicialización de manera correcta según las instrucciones del manual de usuario, al monitorear la base de datos dentro de la estación Processing, se debe ver como los “sockets” o puntos de comunicación en los dos cuadros rojos superiores en la Figura 13, deben tener cualquier valor que no sea ni 0 ni -1, los cuales determinarían que el socket no ha sido creado aun y por ende no hay comunicación establecida. En esta ocasión, los sockets creados tienen el valor de 28 en el programa de recepción y de 29 en el de envió por parte de la estación. Adicionalmente, se identifica que, por ahora, los valores de las variables DatoEnviar y Datorecibido, se encuentran en 0 como punto inicial.

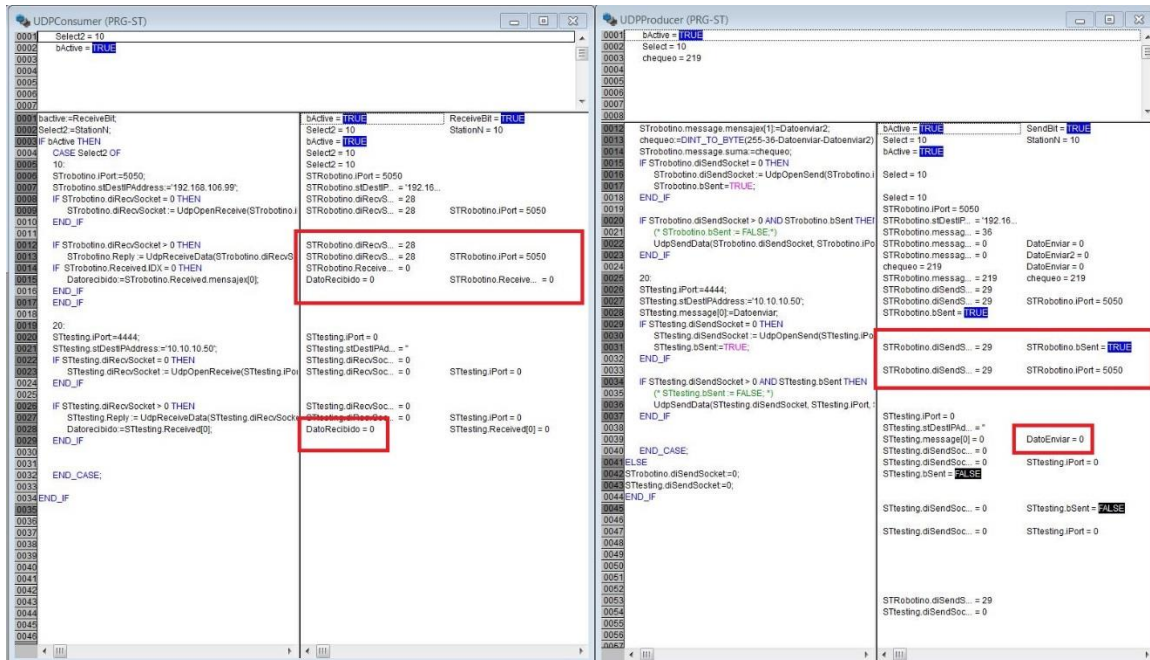


Figura 13. Base de datos dentro de estación Processing para comunicación con Robotino.

Ahora, en la Figura 14, se identifica como en un punto determinado, cuando la estación ha procesado una pieza y la tiene lista en el buffer de salida donde el Robotino la tomará, la estación convierte el valor de la variable DatoEnviar a un 1, con la intención de que el Robotino reciba esto como una orden y comience su desplazamiento para tomar la pieza. Sobra agregar que, para esto, el Robotino ya debe haber iniciado su programa y haber logrado un “homing” satisfactorio.

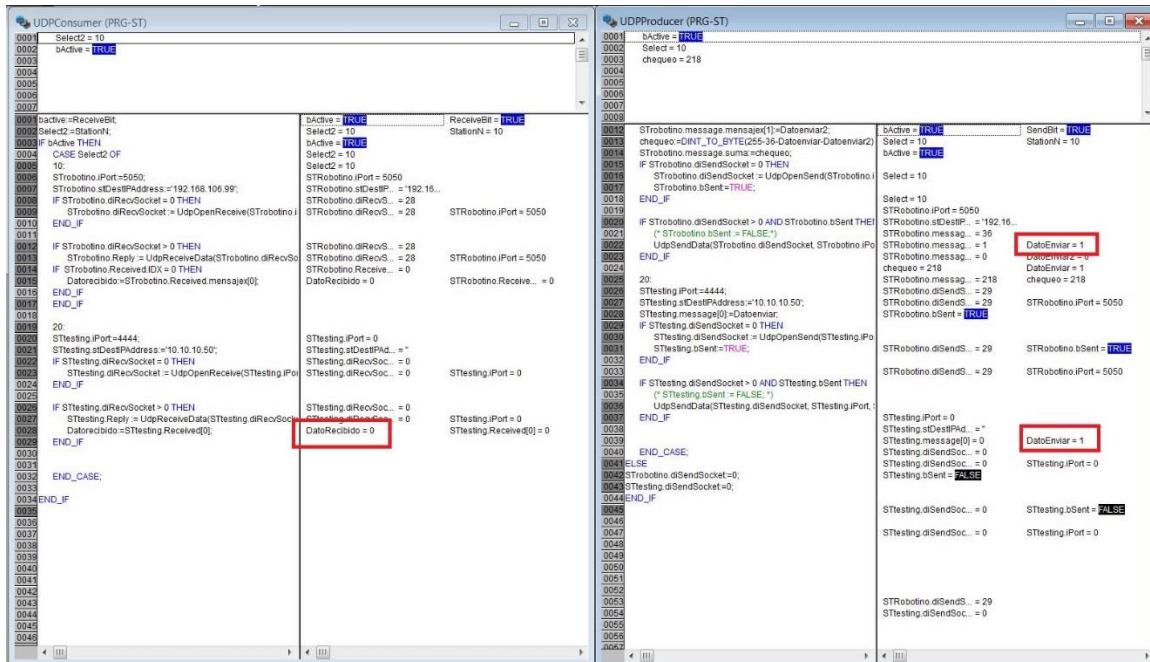


Figura 14. Muestra de envío de datos desde la estación hacia el Robotino.

Finalmente, como se puede ver en la Figura 15, cuando finalmente el Robotino lleva a cabo la primera orden que estaba determinada por el valor de DatoEnviar=1, envía una respuesta por medio de DatoRecibido=2, la cual quiere decir que se encuentra esperando la orden número dos por parte de la estación, y al recibir esta confirmación, la estación a su vez pasa a estar nuevamente disponible para recibir la siguiente pieza por parte de la estación Distributing y a su vez le entrega la siguiente orden al Robotino por medio de la variable DatoEnviar en valor de 2.

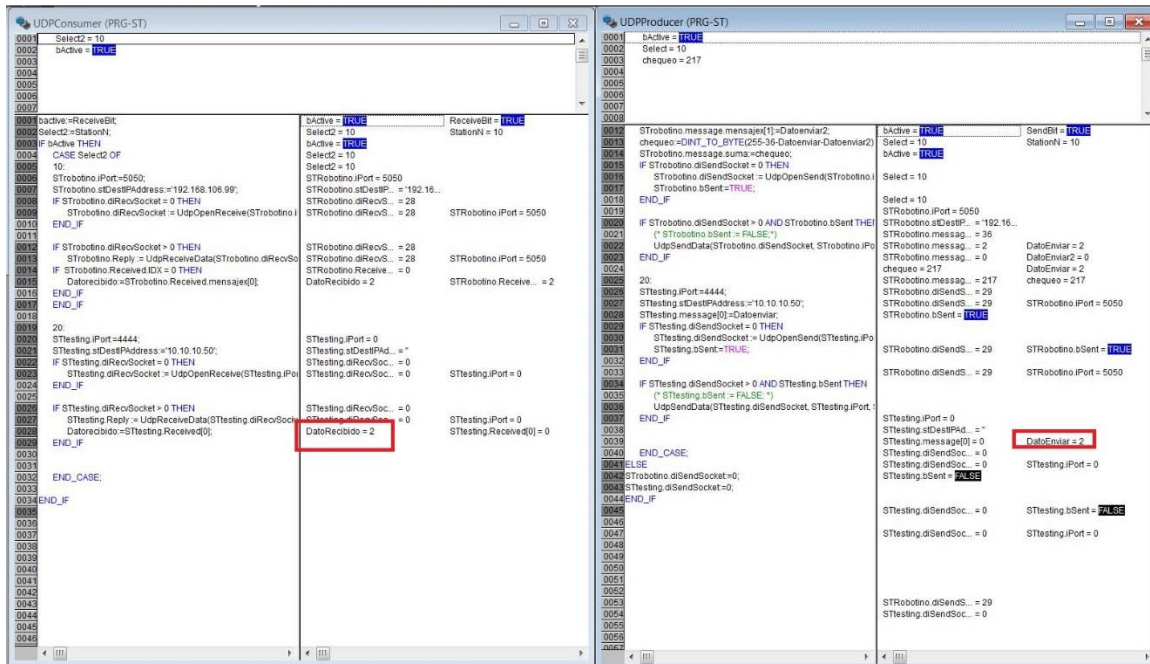


Figura 15. Respuesta del Robotino y asignación de nuevo trabajo.

4.4 Flujo de datos entre estaciones Handling y Sorting por medio de S7_Connection

Por último, al revisar la comunicación entre la estación de Handling y la de Sorting (las dos estaciones que cuentan con PLCs Siemens S7-300), la cual se hace por medio de protocolo S7_Connection, se ve en la Figura 16 como al lado izquierdo de la misma se tiene la base de datos de la estación de Handling y al lado derecho la de Sorting, encerrando en un recuadro rojo la correlacion de variables en cada una de ellas. En el primer caso, todas sus variables se encuentran en 0.

Name	Data type	Offset	Start value	Monitor value
Static	Array[1..10] of Byte	0.0		
Recibido				
Recibido[1]	Byte	0.0	16#0	16#00
Recibido[2]	Byte	1.0	16#0	16#00
Recibido[3]	Byte	2.0	16#0	16#00
Recibido[4]	Byte	3.0	16#0	16#00
Recibido[5]	Byte	4.0	16#0	16#00
Recibido[6]	Byte	5.0	16#0	16#00
Recibido[7]	Byte	6.0	16#0	16#00
Recibido[8]	Byte	7.0	16#0	16#00
Recibido[9]	Byte	8.0	16#0	16#00
Recibido[10]	Byte	9.0	16#0	16#00

Name	Data type	Offset	Start value	Monitor value
Static				
S1_4	Byte	0.0	16#0	16#00
S2_4	Byte	1.0	16#0	16#00
S3_4	Byte	2.0	16#0	16#00
S4_4	Byte	3.0	16#0	16#00
S5_4	Byte	4.0	16#0	16#00

Figura 16. Bases de datos de estaciones Handling y Sorting para comunicación por S7_Connection.

Por consiguiente, en la Figura 17, se demuestra como al activar la señal de STOP desde la pantalla táctil, esta información luego de ser redireccionada hasta la estación de Handling, es posteriormente reenviada como variable Booleana hasta la estación de Sorting. Detalladamente este cambio se ve cuando la segunda variable dentro de cada uno de los recuadros rojos es modificada de su valor inicial 0 (Figura 16) a un valor forzado 1 (Figura 17). La comunicación puede ser constatada en el momento en el que este cambio no solamente se ve en la estación de Handling, sino que simultáneamente se puede evidenciar al hacer monitoreo de las variables internas de la base de datos de la estación de Sorting.

Name	Data type	Offset	Start value	Monitor value
Static				
Recibido	Array[1..10] of Byte	0.0		
Recibido[1]	Byte	0.0	16#0	16#00
Recibido[2]	Byte	1.0	16#0	16#00
Recibido[3]	Byte	2.0	16#0	16#00
Recibido[4]	Byte	3.0	16#0	16#00
Recibido[5]	Byte	4.0	16#0	16#00
Recibido[6]	Byte	5.0	16#0	16#00
Recibido[7]	Byte	6.0	16#0	16#01
Recibido[8]	Byte	7.0	16#0	16#00
Recibido[9]	Byte	8.0	16#0	16#00
Recibido[10]	Byte	9.0	16#0	16#00

Name	Data type	Offset	Start value	Monitor value
Static				
S1_4	Byte	0.0	16#0	16#00
S2_4	Byte	1.0	16#0	16#01
S3_4	Byte	2.0	16#0	16#00
S4_4	Byte	3.0	16#0	16#00
S5_4	Byte	4.0	16#0	16#00

Figura 17. Demostración de variable forzada desde estación Handling hasta Sorting.

En la Figura 18 se puede evidenciar el mismo procedimiento anterior pero enviando la señal de Reset desde la pantalla táctil, la cual modifica el valor de la cuarta variable de las 5 encerradas en los recuadros.

Name	Data type	Offset	Start value	Monitor value	Ret
Static	Array[1..10] of Byte	0.0			
Recibido	Byte	0.0	16#0	16#00	
Recibido[1]	Byte	1.0	16#0	16#00	
Recibido[2]	Byte	2.0	16#0	16#00	
Recibido[3]	Byte	3.0	16#0	16#00	
Recibido[4]	Byte	4.0	16#0	16#00	
Recibido[5]	Byte	5.0	16#0	16#00	
Recibido[6]	Byte	6.0	16#0	16#00	
Recibido[7]	Byte	7.0	16#0	16#00	
Recibido[8]	Byte	8.0	16#0	16#01	
Recibido[9]	Byte	9.0	16#0	16#00	

Name	Data type	Offset	Start value	Monitor value	Ret
Static	Byte	0.0	16#0	16#00	
S1_4	Byte	1.0	16#0	16#00	
S2_4	Byte	2.0	16#0	16#00	
S3_4	Byte	3.0	16#0	16#01	
S4_4	Byte	4.0	16#0	16#00	

Figura 18. Demostración de Reset forzado desde estación Handling hasta Sorting.

Para finalizar, en las figuras 19 y 20, se prueba la el envío de información desde la estación de vuelta a la estación de Handling, para así completar todas las comunicaciones dentro de la red que se tiene entre las estaciones. Para esto, se ve como los primeros dos datos de la estación Sorting (mitad derecha de la captura de pantalla) se ven reflejados en la base de datos de la estación Handling (mitad izquierda) en la Figura 19, y a continuación son modificados en la Figura 20, al conmutar una de las entradas de la estación y así mismo las salidas que la estación activa en respuesta a la orden dada.

The image shows two side-by-side screenshots of the Siemens STEP 7 LAD editor. The left window is titled 'Enviado_Desde_Sorting' and the right window is titled 'Send_Data'. Both windows display a table of diagnostic data points.

Enviado_Desde_Sorting Table:

Name	Data type	Offset	Start value	Monitor value
Static				
CONTADOR	Int	0.0	0	0
ENTRADAS	Byte	2.0	16#0	16#50
SALIDAS	Byte	3.0	16#0	16#80
PiezasTotales	Byte	4.0	16#0	16#00
RojasProducidas	Byte	5.0	16#0	16#00
PlateadasProducidas	Byte	6.0	16#0	16#00
NegrasProducidas	Byte	7.0	16#0	16#00
PiezaActual	Byte	8.0	16#0	16#01
P1_4	Byte	9.0	16#0	16#01
P2_4	Byte	10.0	16#0	16#00
P3_4	Byte	11.0	16#0	16#00
PIEZA	Array[1..50]	12.0		
PIEZA[1]	Byte	0.0	16#0	16#03
PIEZA[2]	Byte	1.0	16#0	16#02
PIEZA[3]	Byte	2.0	16#0	16#03
PIEZA[4]	Byte	3.0	16#0	16#03
PIEZA[5]	Byte	4.0	16#0	16#03
PIEZA[6]	Byte	5.0	16#0	16#02
PIEZA[7]	Byte	6.0	16#0	16#03
PIEZA[8]	Byte	7.0	16#0	16#03
PIEZA[9]	Byte	8.0	16#0	16#03
PIEZA[10]	Byte	9.0	16#0	16#03
PIEZA[11]	Byte	10.0	16#0	16#03
PIEZA[12]	Byte	11.0	16#0	16#01
PIEZA[13]	Byte	12.0	16#0	16#01
PIEZA[14]	Byte	13.0	16#0	16#00
PIEZA[15]	Byte	14.0	16#0	16#00
PIEZA[16]	Byte	15.0	16#0	16#00
PIEZA[17]	Byte	16.0	16#0	16#00

Send_Data Table:

Name	Data type	Offset	Start value	Monitor value	Retain
Static					
Enviados	Array[1..66]	0.0			
Enviados[1]	Byte	0.0	16#0	16#22	
Enviados[2]	Byte	1.0	16#0	16#80	
Enviados[3]	Byte	2.0	16#0	16#01	
Enviados[4]	Byte	3.0	16#0	16#00	
Enviados[5]	Byte	4.0	16#0	16#01	
Enviados[6]	Byte	5.0	16#0	16#50	
Enviados[7]	Byte	6.0	16#0	16#80	
Enviados[8]	Byte	7.0	16#0	16#00	
Enviados[9]	Byte	8.0	16#0	16#00	
Enviados[10]	Byte	9.0	16#0	16#00	
Enviados[11]	Byte	10.0	16#0	16#00	
Enviados[12]	Byte	11.0	16#0	16#01	
Enviados[13]	Byte	12.0	16#0	16#01	
Enviados[14]	Byte	13.0	16#0	16#00	
Enviados[15]	Byte	14.0	16#0	16#00	
Enviados[16]	Byte	15.0	16#0	16#03	
Enviados[17]	Byte	16.0	16#0	16#02	
Enviados[18]	Byte	17.0	16#0	16#03	
Enviados[19]	Byte	18.0	16#0	16#03	
Enviados[20]	Byte	19.0	16#0	16#03	
Enviados[21]	Byte	20.0	16#0	16#02	
Enviados[22]	Byte	21.0	16#0	16#03	
Enviados[23]	Byte	22.0	16#0	16#03	
Enviados[24]	Byte	23.0	16#0	16#03	
Enviados[25]	Byte	24.0	16#0	16#03	
Enviados[26]	Byte	25.0	16#0	16#03	
Enviados[27]	Byte	26.0	16#0	16#01	
Enviados[28]	Byte	27.0	16#0	16#01	
Enviados[29]	Byte	28.0	16#0	16#00	
Enviados[30]	Byte	29.0	16#0	16#00	

Figura 19. Envío de información diagnostica desde estación Sorting a estación Handling.

Name	Data type	Offset	Start value	Monitor value
1	Static			
2	CONTADOR	Int	0.0	0
3	ENTRADAS	Byte	2.0	16#50
4	SALIDAS	Byte	3.0	16#89
5	PiezasTotales	Byte	4.0	16#02
6	RojasProducidas	Byte	5.0	16#00
7	PlateadasProducidas	Byte	6.0	16#01
8	NegrasProducidas	Byte	7.0	16#00
9	PiezaActual	Byte	8.0	16#02
10	P1_4	Byte	9.0	16#00
11	P2_4	Byte	10.0	16#00
12	P3_4	Byte	11.0	16#00
13	PIEZA	Array[1..50]	12.0	
14	PIEZA[1]	Byte	0.0	16#02
15	PIEZA[2]	Byte	1.0	16#02
16	PIEZA[3]	Byte	2.0	16#03
17	PIEZA[4]	Byte	3.0	16#03
18	PIEZA[5]	Byte	4.0	16#03
19	PIEZA[6]	Byte	5.0	16#02
20	PIEZA[7]	Byte	6.0	16#03
21	PIEZA[8]	Byte	7.0	16#03
22	PIEZA[9]	Byte	8.0	16#03
23	PIEZA[10]	Byte	9.0	16#03
24	PIEZA[11]	Byte	10.0	16#03
25	PIEZA[12]	Byte	11.0	16#01
26	PIEZA[13]	Byte	12.0	16#01
27	PIEZA[14]	Byte	13.0	16#00
28	PIEZA[15]	Byte	14.0	16#00
29	PIEZA[16]	Byte	15.0	16#00
30	PIEZA[17]	Byte	16.0	16#00

Name	Data type	Offset	Start value	Monitor value	Retain
1	Static				
2	Enviados	Array[1..66]	0.0		
3	Enviados[1]	Byte	0.0	16#22	
4	Enviados[2]	Byte	1.0	16#80	
5	Enviados[3]	Byte	2.0	16#01	
6	Enviados[4]	Byte	3.0	16#00	
7	Enviados[5]	Byte	4.0	16#01	
8	Enviados[6]	Byte	5.0	16#50	
9	Enviados[7]	Byte	6.0	16#89	
10	Enviados[8]	Byte	7.0	16#02	
11	Enviados[9]	Byte	8.0	16#00	
12	Enviados[10]	Byte	9.0	16#01	
13	Enviados[11]	Byte	10.0	16#00	
14	Enviados[12]	Byte	11.0	16#02	
15	Enviados[13]	Byte	12.0	16#00	
16	Enviados[14]	Byte	13.0	16#00	
17	Enviados[15]	Byte	14.0	16#00	
18	Enviados[16]	Byte	15.0	16#02	
19	Enviados[17]	Byte	16.0	16#02	
20	Enviados[18]	Byte	17.0	16#03	
21	Enviados[19]	Byte	18.0	16#03	
22	Enviados[20]	Byte	19.0	16#03	
23	Enviados[21]	Byte	20.0	16#02	
24	Enviados[22]	Byte	21.0	16#03	
25	Enviados[23]	Byte	22.0	16#03	
26	Enviados[24]	Byte	23.0	16#03	
27	Enviados[25]	Byte	24.0	16#03	
28	Enviados[26]	Byte	25.0	16#03	
29	Enviados[27]	Byte	26.0	16#01	
30	Enviados[28]	Byte	27.0	16#01	
31	Enviados[29]	Byte	28.0	16#00	
32	Enviados[30]	Byte	29.0	16#00	

Figura 20. Forzado de variables de estación Sorting reflejado en base de datos de estación Handling.

Capítulo 5

Manual de Usuario.

Con el fin de dar una mayor comprensión del funcionamiento del sistema didáctico aquí presentado se adjunta un manual de usuario detallando los procesos a trabajar que no hayan sido explicados ya en el documento presente con la intención de no repetir información previamente explicada.

5.1 Inicialización del sistema.

En esta primera parte se explica el orden correcto con el cual deben ser encendidos cada uno de los elementos que conforman el sistema didáctico de industria 4.0 para la Universidad San Buenaventura, sede Bogotá. El no seguir las instrucciones dadas, podría influir en el mal funcionamiento de alguno de los elementos implicados ya que fueron programados para seguir un orden establecido de inicialización dentro de cada uno de los programas.

Antes de iniciar, la primera verificación a realizarse es que tanto las estaciones como el sistema de robótica móvil (Robotino), estén apagados. Para el caso de las estaciones, las cuales deben empezar sin estar conectadas a la energía, también es importante verificar que su interruptor principal se encuentre en modo apagado.

La red bajo la cual corre el sistema, es una red privada la cual consta de un router principal y de 3 Access Points (AP) usados como clientes secundarios los cuales tienen la intención de extender la comunicación de manera inalámbrica a los demás dispositivos, ayudados de un Switch cada uno para aumentar la cantidad de puertos Ethernet que

pueden llegar al AP. En este orden de ideas, se hace indispensable encender el Router principal el cual se encuentra ubicado justo debajo de la pantalla táctil FED-550 y acto seguido, encender tanto los Access Points como los Switches que se encuentran ubicados detrás de cada uno de los dos bloques de estaciones. Para esta finalidad, es suficiente con energizar cada uno de los elementos anteriormente mencionados.

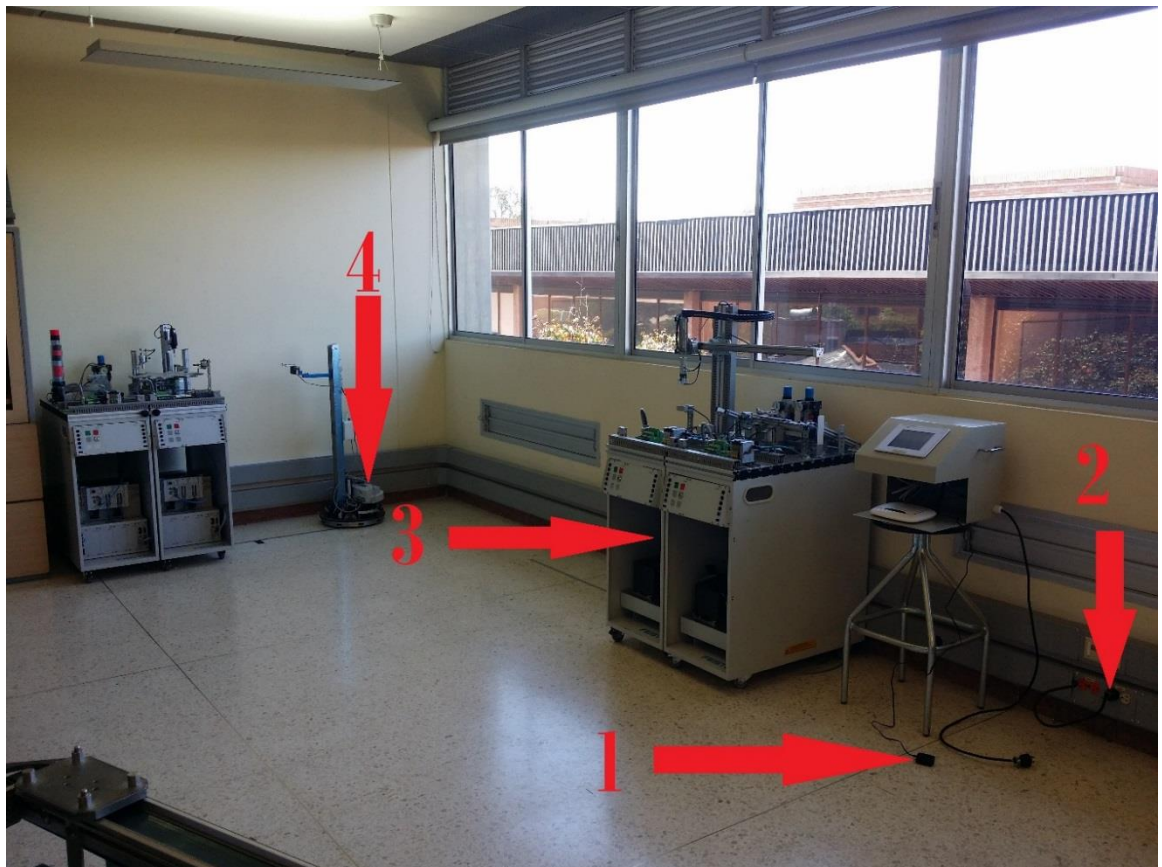


Figura 21. Puntos de energización del Sistema didáctico de Industria 4.0.

Los numerales en la Figura 21 indican los distintos puntos de energización del sistema de forma visual además de indicar el orden de pasos en los cuales deben ser encendidos, son los siguientes:

1. Punto de energización del Router Principal.
2. Punto de energización de todo el bloque de estación Distribucion, estación Procesamiento, AP y Switch respectivo. Su ubicación se encuentra retirada de la estación debido a que originalmente en este punto del salón no había puntos de corriente alterna, entonces se pasó una extensión dedicada solamente para este bloque, sin tocar la acometida eléctrica ya existente del salón.
3. Puntos de energización de estación Manipulación, estación Clasificación, AP y Switch respectivo, detrás de las estaciones alcanzable a través del módulo donde se encuentra ubicado el PLC de la estación Manipulación.
4. Botón de encendido del sistema Robotino(Φ). En este caso se requiere mantener oprimido el botón por lo menos durante 2 segundos con la intención de que se inicie. Esto es necesario ya que el AP que se encarga de conectar al Robotino con el resto del sistema, también se energiza en el momento en éste se enciende. Es importante aclarar que a pesar de que el Robotino demora aproximadamente un minuto y medio en inicializar por completo, el AP demora apenas alrededor de 15 segundos en estar conectado dentro de la red y listo para ser usado.

Una vez inicializada toda la red, lo siguiente corresponde a encender los dispositivos en el orden visualizado en la siguiente figura.

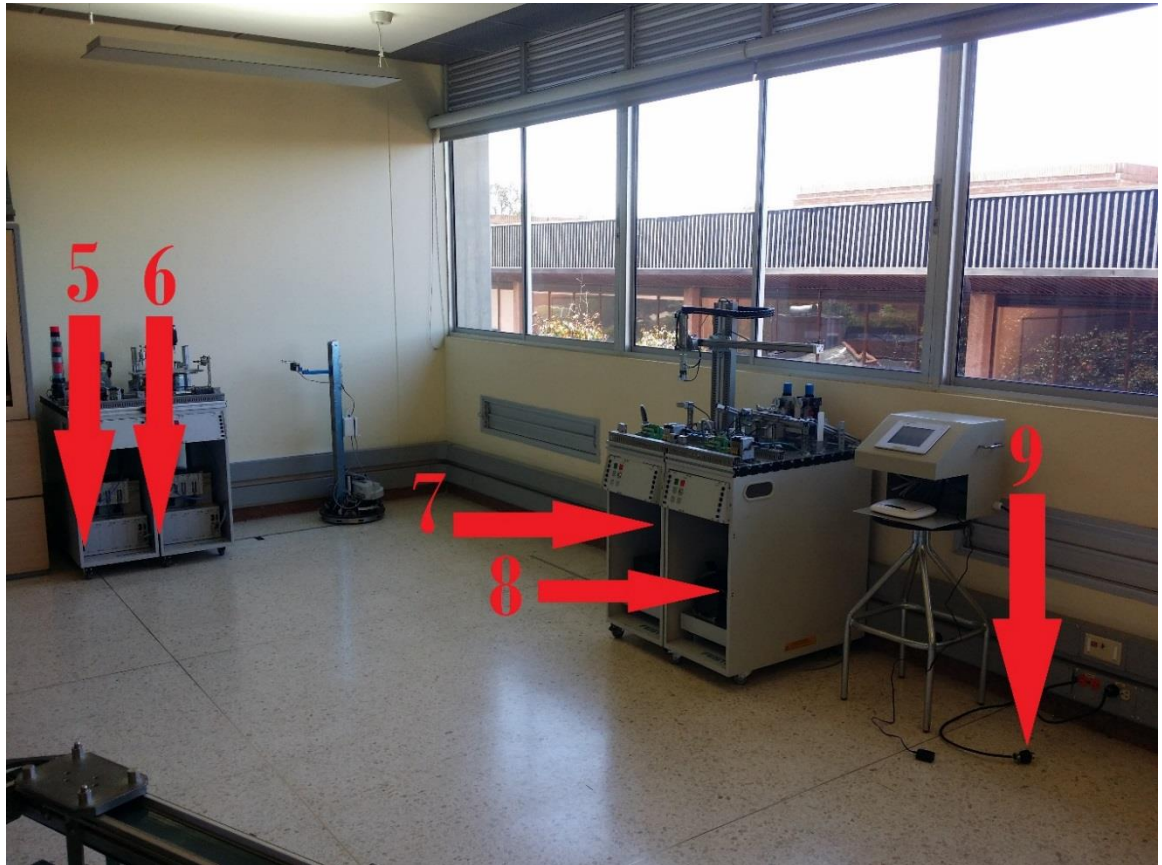


Figura 22. Encendido de dispositivos.

Como quinto paso se enciende la estación Distribucion (Primera estación de izquierda a derecha), debido a que es la cual recibe la información que llega a través de la red proveniente de las otras estaciones. Puesto que esta estación ya se encuentra energizada en el momento en el que se conectó el punto de energización de todo el bloque, no hace falta nada más que encender el interruptor que se ubica en frente del

EduTrainer del PLC en la parte inferior izquierda del mismo, donde muestra la flecha.

(Ver Figura 22)



Figura 23. Interruptor para encender la estación Distribucion.

Antes de encender la estación Procesamiento (segunda estación de izquierda a derecha) es necesario asegurar que el Robotino se encuentra inicializado mostrando por lo menos su respectiva dirección IP en la pantalla LCD superior. La razón de lo anterior, es porque al encenderse la estación, el programa del PLC lo primero que hace es crear los sockets de comunicación al Robotino a partir de la presencia del mismo en la red, motivo por el cual, si el este no está encendido y completamente inicializado, la estación no lo encontrará en la red cuando se encienda y por ende no generará la comunicación de manera correcta. Si el Robotino ha sido completamente inicializado, se enciende la estación de la misma manera que se encendió la primera. Convirtiéndose así en el paso 6.

El paso 7 corresponde a encenderse la estación de Manipulación (tercera estación de izquierda a derecha), para esto se hace necesario prender el módulo completo del PLC.

Es importante revisar que tanto la fuente de alimentación del PLC como el interruptor estén en posición de encendido. (Ver Figura 23)



Figura 24. Fuente alimentación PLC.

Se enciende la estación de Procesamiento (cuarta estación de izquierda a derecha). El procedimiento para encenderla es el mismo que para la tercera estación, siendo así el paso número 8.

El paso número 9 corresponde a encender la pantalla táctil FED-550. Para esto, se conecta la clavija monofásica de 110VAC a cualquier toma de corriente con este voltaje correspondiente. Se aclara que no es obligatorio conectar ni el Router principal ni la pantalla a una de las tomas de corriente que se ven en las imágenes, sino que preferiblemente puede ser desde cualquier punto dentro del salón, ya que la red que el Router genera y mediante la cual se comunican con las demás estaciones, tiene un

alcance de 10 metros sin obstáculos de por medio sin afectar la eficiencia de las comunicaciones ni generando pérdidas de datos en la red.

El paso número 10 es abrir las perillas rojas de las unidades de mantenimiento neumático de cada una de las 3 estaciones, señaladas en la Figura 25. Esto con la intención de hacer llegar la presión de aire deseada a las 3 estaciones que requieren aire comprimido para su correcto funcionamiento. Se resalta la importancia de que la presión que muestra el manómetro de cada una de las 3 unidades de mantenimiento no debe ser inferior a 4Bar ni superar los 6Bar. En caso de que el valor de la presión se salga de estos rangos, se debe levantar ligeramente la perilla azul que tienen las unidades en la parte superior y luego graduar la presión girando la perilla hasta que la presión se encuentre entre en los valores establecidos.



Figura 25. Perillas neumáticas a abrir.

El último paso es posicionar en su punto inicial al Robotino y arrancar el programa el cual está encargado de su comunicación con las estaciones y reciba las órdenes que estas le den. Este procedimiento se menciona a continuación.

5.2 Posición inicial del Robotino.

Para posicionar el Robotino, se toma la dirección donde se encuentra la pinza como su parte frontal, se apunta la pinza en dirección de la estación Procesamiento como lo muestra la Figura 26. Adicionalmente, desde una vista superior del sistema, se debe asegurar que el brazo azul esté ubicado a la derecha de la cinta metálica que se encuentra pegada en el suelo. Esto se hace porque cuando el Robotino hace “Homing”, a partir de la línea metálica de derecha a izquierda para alinearse y establecer su punto cero.



Figura 26. Posición Inicial Robotino.

Al estar en la pantalla principal donde se muestra la dirección IP del Robotino, se debe pulsar el botón “ENTER” ubicado debajo de la pantalla LCD. En cuanto aparezca el Menú principal, es necesario utilizar la flecha hacia abajo (↓) para llegar a la opción “Programs”, es allí donde están guardados los programas que se han cargado al Robotino

desde su restauración. Luego se pulsa la tecla “ENTER” nuevamente para ingresar al submenú de programas.



Figura 27. Pantalla Principal del Robotino.

Estando en el submenú “Programs”, con la ayuda de las flechas se selecciona el programa “Sistema_Industy” el cual hace referencia al programa “Sistema_Industry_4.0.rvw2” (programa de funcionamiento del Robotino). Para comenzar a correr el programa, se pulsa la tecla “ENTER”.

Si todo fue hecho de manera correcta, el Robotino debería empezar a moverse a su izquierda buscando con los dos sensores inductivos que tiene por debajo la cinta metálica para alinearse y hacer el respectivo “Homing” que es establecer su posición cero dentro del plano en el que se moverá en el sistema. Una vez haya cumplido con este procedimiento, este queda quieto sobre la cinta metálica hasta que la estación de Procesamiento haya “procesado” una pieza y la ubique en su buffer de salida y posteriormente de la orden al Robotino de recogerla y llevarla a la estación de

Manipulación. Y así, ahora el sistema está listo para ser operado desde la pantalla táctil FED-550.

Capítulo 6

Conclusiones.

Después de haber llevado a cabo el desarrollo del proyecto descrito en este documento, utilizando la metodología planteada, se llegaron a las siguientes conclusiones:

El primer objetivo específico fue cumplido gracias a que se diseñó e implementó una Interfaz Humano Maquina desde la pantalla táctil FED-550, la cual tiene la intención de permitirle al operador dar órdenes a cada una de las 4 estaciones de manera independiente, y así mismo adquirir información fundamental para el funcionamiento de la aplicación desde cada una de las mismas y mostrarla eficientemente. A lo largo de este desarrollo, la principal complicación que se tuvo fue la de lograr que la información enviada llegara a todas las estaciones. Esto se debe a que, si la inicialización de las 4 estaciones no se hace en el orden correcto, existía la posibilidad de que se cortara la cadena de comunicación desde la pantalla hasta la estación, principalmente en la última estación que es en la cual se requiere una cadena con más puentes.

Con respecto al segundo objetivo específico, se cumple al haberse hecho un manual que explique el funcionamiento del sistema, basado en las pruebas sobre el sistema ya en funcionamiento.

En referencia al tercer objetivo específico, se logró la realización del programa demo en el cual se pueden evidenciar los desarrollos explicados anteriormente. Para el cumplimiento de este objetivo, el principal problema que se tuvo fue el desarrollo de los programas lo suficientemente fuertes como para mantener la comunicación entre

estaciones durante un tiempo indefinido. Lo anterior debido a que los ciclos de actualización de datos entre los PLC dentro de una misma red deben ser lo suficientemente rápidos como para no tener demoras en el envío de los datos, pero no tanto como para afectar el protocolo y por ende perder la comunicación ya establecida. El segundo problema al cual se llegó, fue que al estar utilizando los equipos existentes en la universidad, las limitaciones de los PLC para concretar una conexión a través de internet y así mismo con la nube virtual, hicieron que el desarrollo tuviese un tope máximo en integración con la industria 4.0, debido a que se logra la intercomunicación de las estaciones por medio de una red local con el concepto de comunicación Maquina a Maquina (M2M), el diagnostico en tiempo real de sistemas y el flujo de información que hay desde una base de datos distribuida entre las diferentes maquinas, hasta el sistema de visualización y control que enlaza al operario con el equipo. Una vez se alcanzó la estabilidad propuesta, los pormenores enfrentados fueron de alineación, calibración de sensores, ajustes mecánicos y de detección de fallas ya existentes en las estaciones de la universidad, las cuales habían sido manipuladas y des configuradas anteriormente por estudiantes y profesores.

Finalmente, con base a lo anterior se concluye que se cumple el objetivo general de esta tesis al haberse podido integrar y programar un sistema que aplique varios de los conceptos de la industria 4.0, a través de una metodología y un desarrollo ingenieril previamente propuestos utilizando equipos existentes en la universidad San Buenaventura, Sede Bogotá.

Lista de referencias

- CUDRI, K.J.S; TARAZONA, F.P; ARBOLEDA, A.F; ALDANA, D.J.M; SALAZAR, L.A.C. (2015) *Sistema Ciber-Físico de una CNC para la producción de circuitos impresos*, Memorias.
- GARCÍA, M.V; IRISARRI, E; PEREZ, F. (2017) *Integración Vertical en plantas industriales utilizando OPC UA e IEC-61499*, Enfoque UTE 8.1, pp. 287.
- BETANCOURT, D.Y; GÓMEZ, G.A. (2015) *Prototipo de sistema de vigilancia Basado en la internet de las Cosas con Aplicativo para Dispositivos Móviles*, Facultad de Ingeniería Industrial, Universidad Distrital Francisco José de Caldas, Bogotá.
- NEVES DOS REIS, A.; VIERO, C.F; SILVA,T; BEN, R. (sf) *Ambiente inteligente para Indústria 4.0: uma proposta baseada em agentes*, Centro de Tecnologias Digitais, Universidade fevale, Brasil.
- LEITÃO, P. BARBOSA,J. (2015) *Desenvolvimento de um sistema de transporte modular com auto-organização de componentes ciber-físicos*, Instituto politécnico de Bragança, Brasil.
- OLIVEIR, L.E. (2017) *Concepção de un framewok para monitoramento e teleoperação de máquinas-ferramenta CNC via Internet aderente à Indústria 4.0*, Faculdade de tecnologia, Departamento de Engenharia mecânica, Universidad de Brasilia, Brasil.
- TORRE, J.A. (2016) *Diseño e Implementación de prototipo de Hardware/Software para la automatización de procesos industriales en la Harinera Castellana*, Escuela Superior de Informática, Universidad de Castilla-La Mancha, España.
- Escuela Superior de Informática, Universidad de Castilla-La Mancha, España.
- DE LA FUENTE, E; MAZAEDA, R. (sf) *Industria 4.0*, Informática Industrial, Escuela de Ingenierías Industriales, Universidad de Valladolid, España.
- VAZQUEZ, F. (2016) *Implementación de los protocolos de comunicación industrial mediante Simatic S7-1200 para el laboratorio de automatización*, Universidad del Azuay, Ecuador.
- EQUIPO DE TRABAJO SENA. (2013) *Los PLC en los sistemas SCADA*, 1ra Unidad, PLC Modulares, Programa de Formación SENA.
- POZO, D. (2016) *La Empresa Digital:camino hacia Industria*, Siemens, S.A.

- NAVARRO,M.; SABALZA, X. (2016) *Reflexiones sobre la Industria 4.0 desde el caso vasco*, EKONOMIAZ, Revista vasca de Economía, No. 89, Ed. 1, pp. 142-173, España. 2016.
- FELIX, C.C; BLESÁ, A. (2015) *Industria 4.0 en el grado de Ingeniería Electrónica y Automática*, Actas de las XXI Jornadas de la Enseñanza, Universitaria de la Informática, Andorra, La vella, 2015.
- LEE,J.; BAGHERI,B; KAO,H. (2015) *A cyber-physical Systems architecture for Industry 4.0- based manufacturing systems*, Manufacturing Letters. Vol. 3, pp. 18-23.
- FESTO. (sf) *Entorno didáctico para industria 4.0*, Recuperado de <http://www.festo-didactic.com/int-es/highlights/soluciones-de-formacion-integrales/entornos-didacticos/industria-4.0/?fbid=aW50LmVzLjU1Ny4xNC4xMC43ODAyLjQzNjU>. el 10 de octubre 2017.

Anexo

Anexo 1. Diagrama de Flujo estación Distribución.

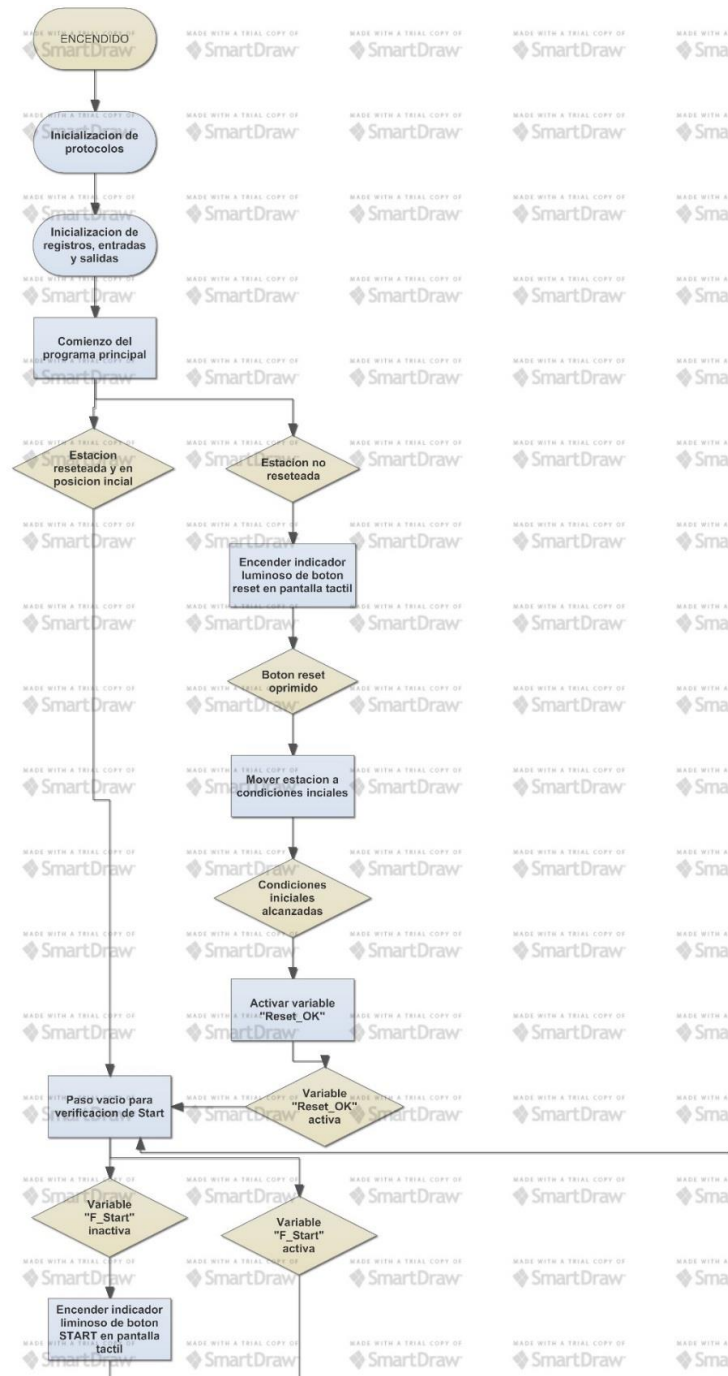


Figura 28. Diagrama de Flujo Distribución. Parte 1.

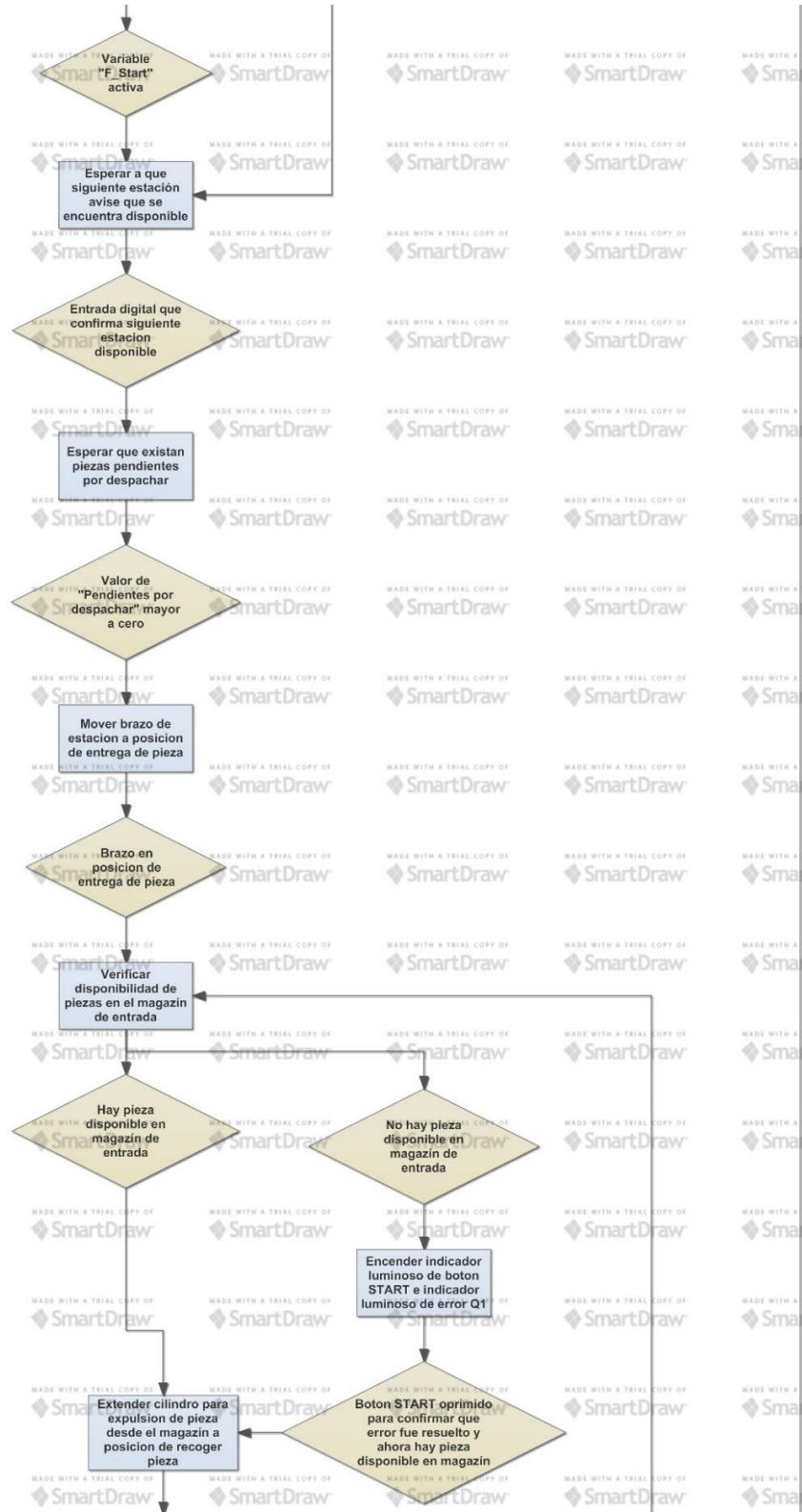


Figura 29. Diagrama de Flujo Distribución. Parte 2.

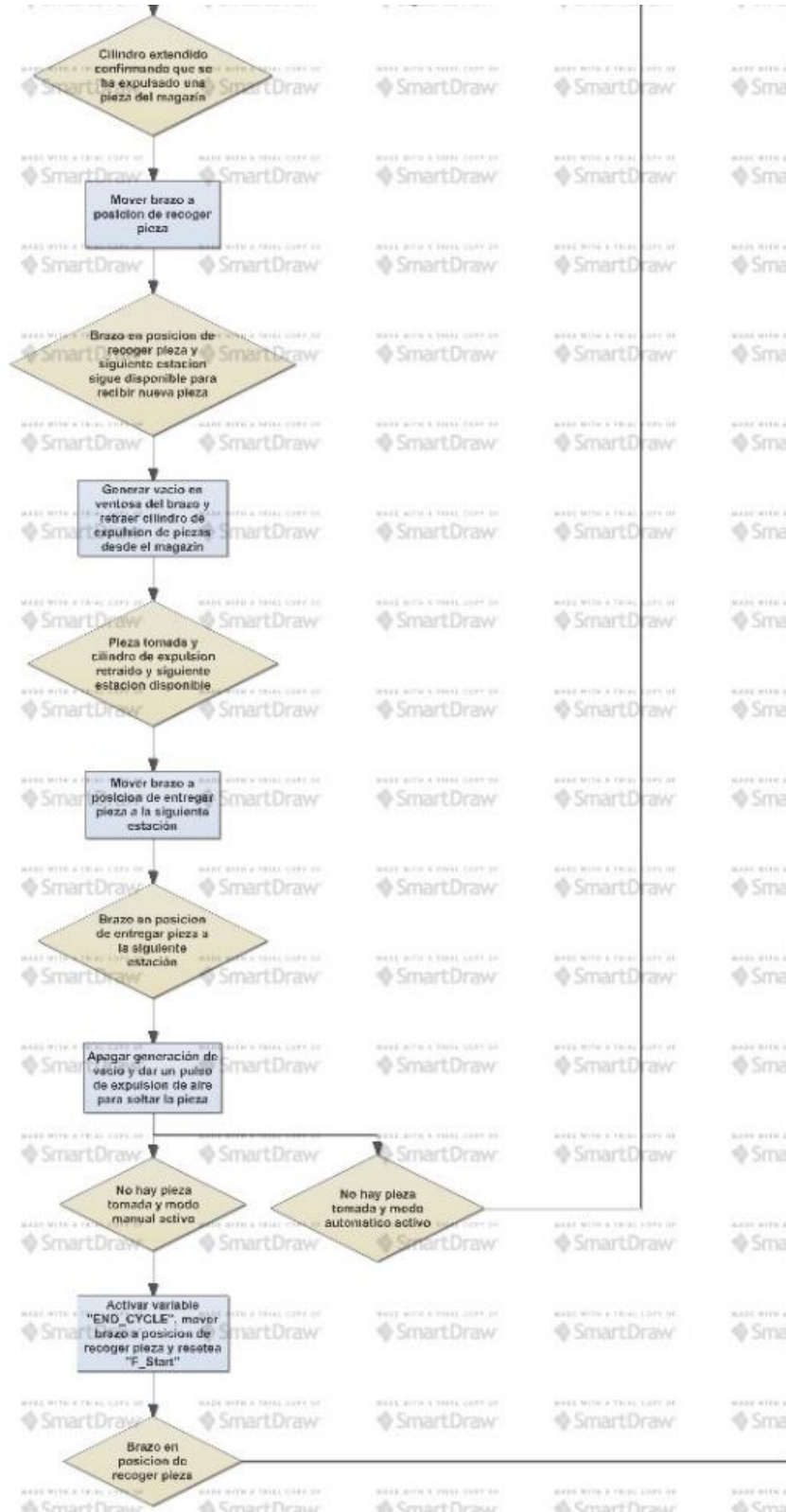


Figura 30. Diagrama de Flujo Distribución. Parte 3.

Anexo 2. Diagrama de Flujo estación Procesamiento.

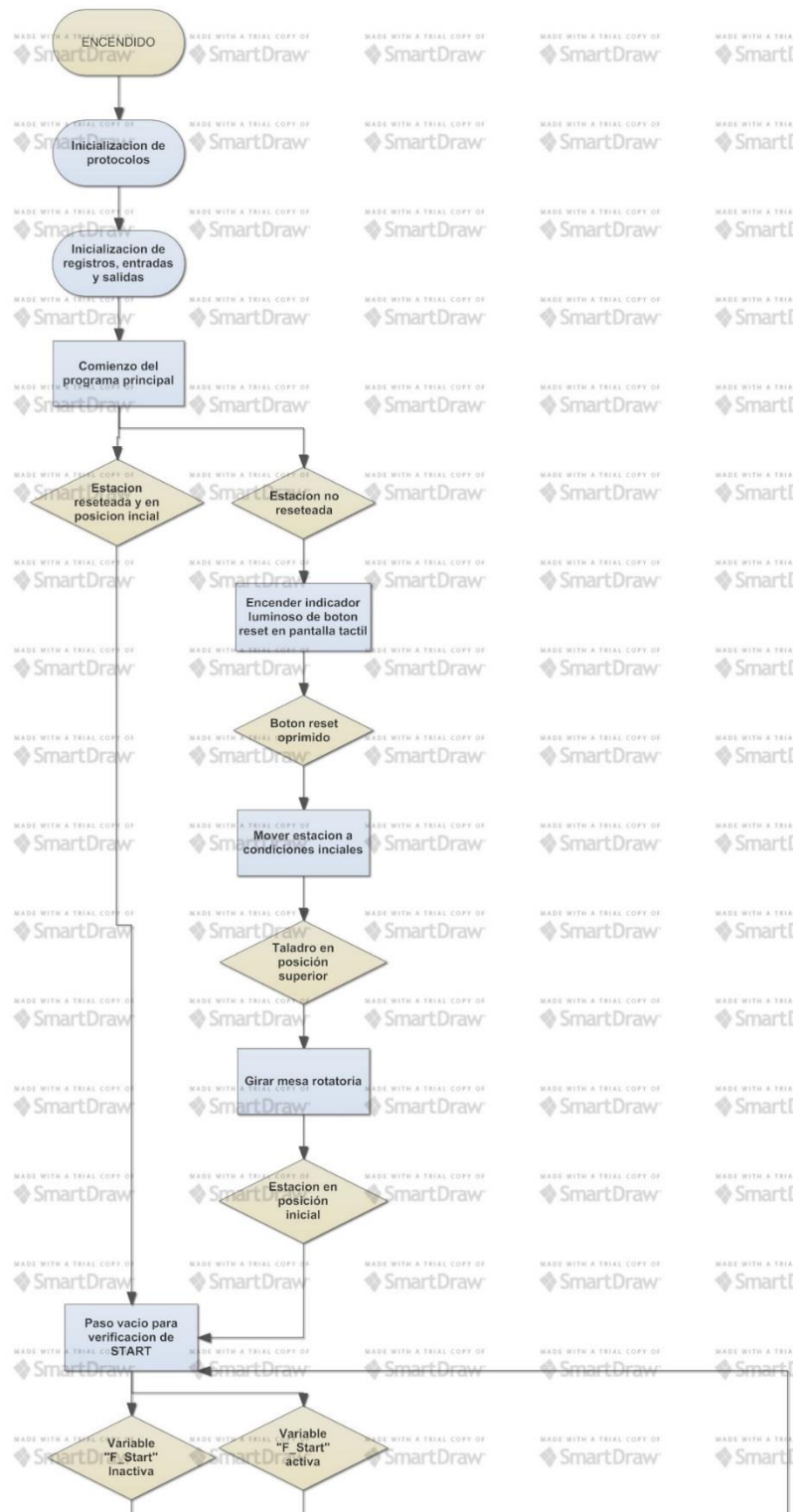


Figura 31. Diagrama de Flujo Procesamiento. Parte 1.

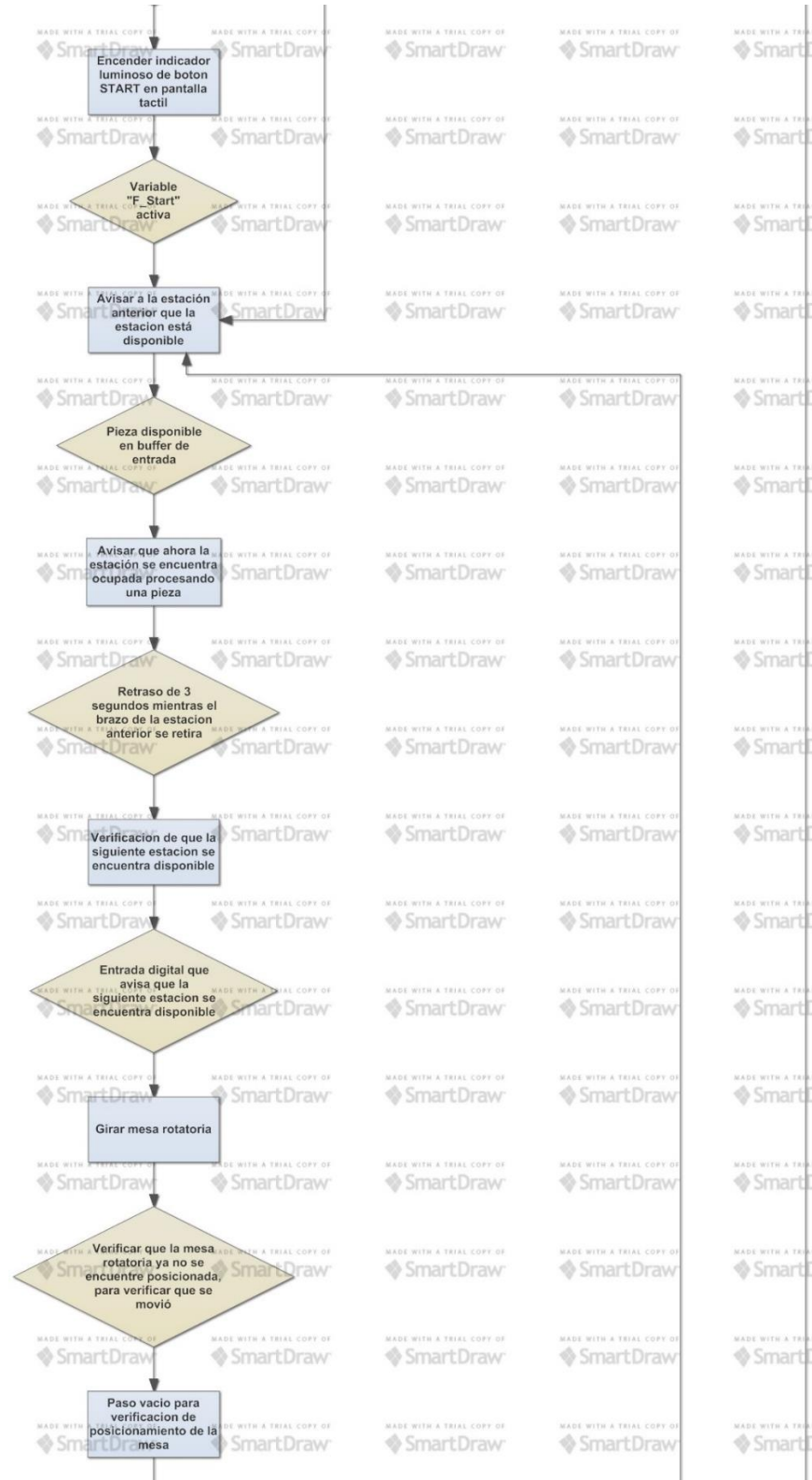


Figura 32. Diagrama de Flujo Procesamiento. Parte 2.

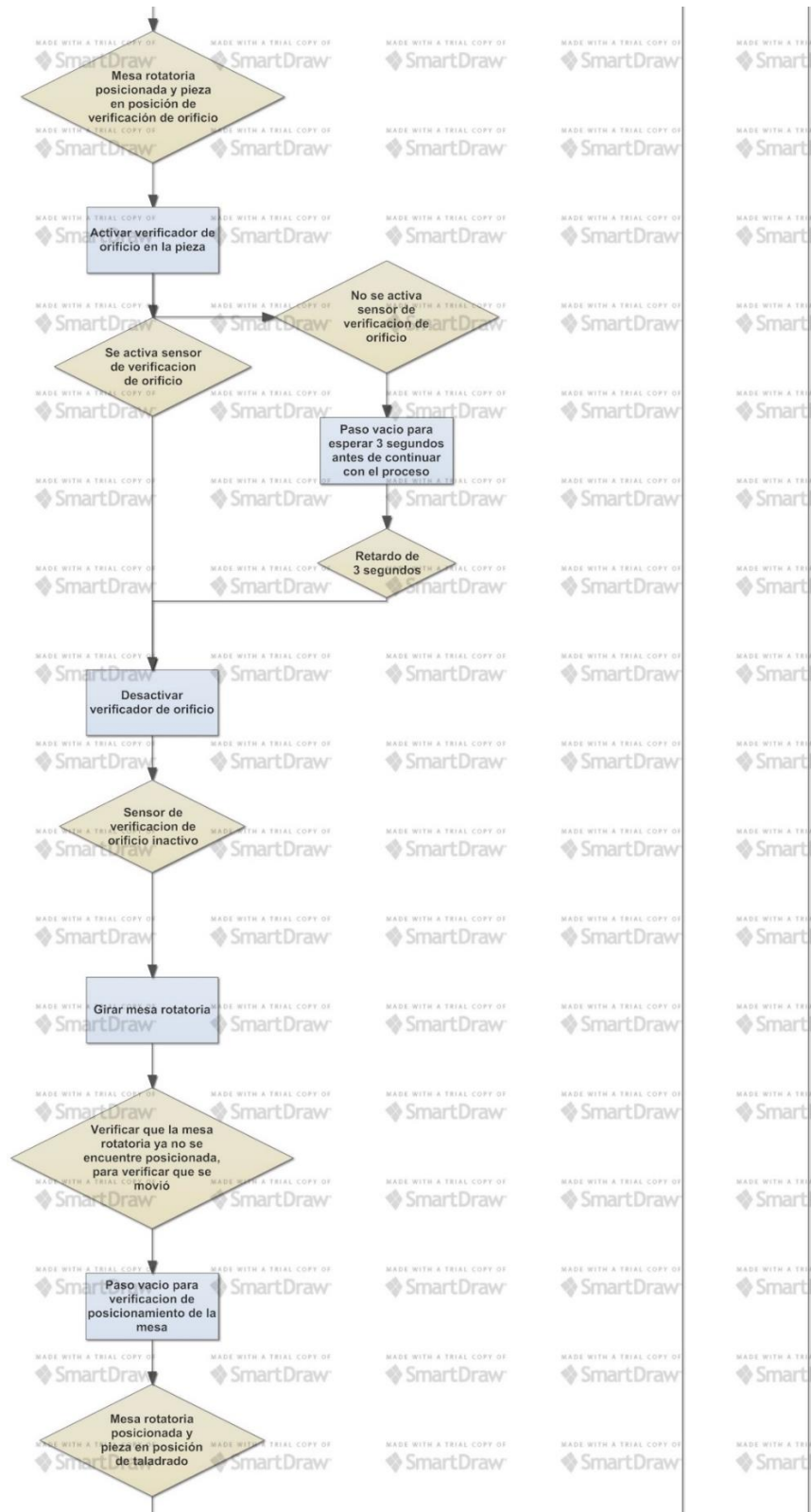


Figura 33. Diagrama de Flujo Procesamiento. Parte 3.

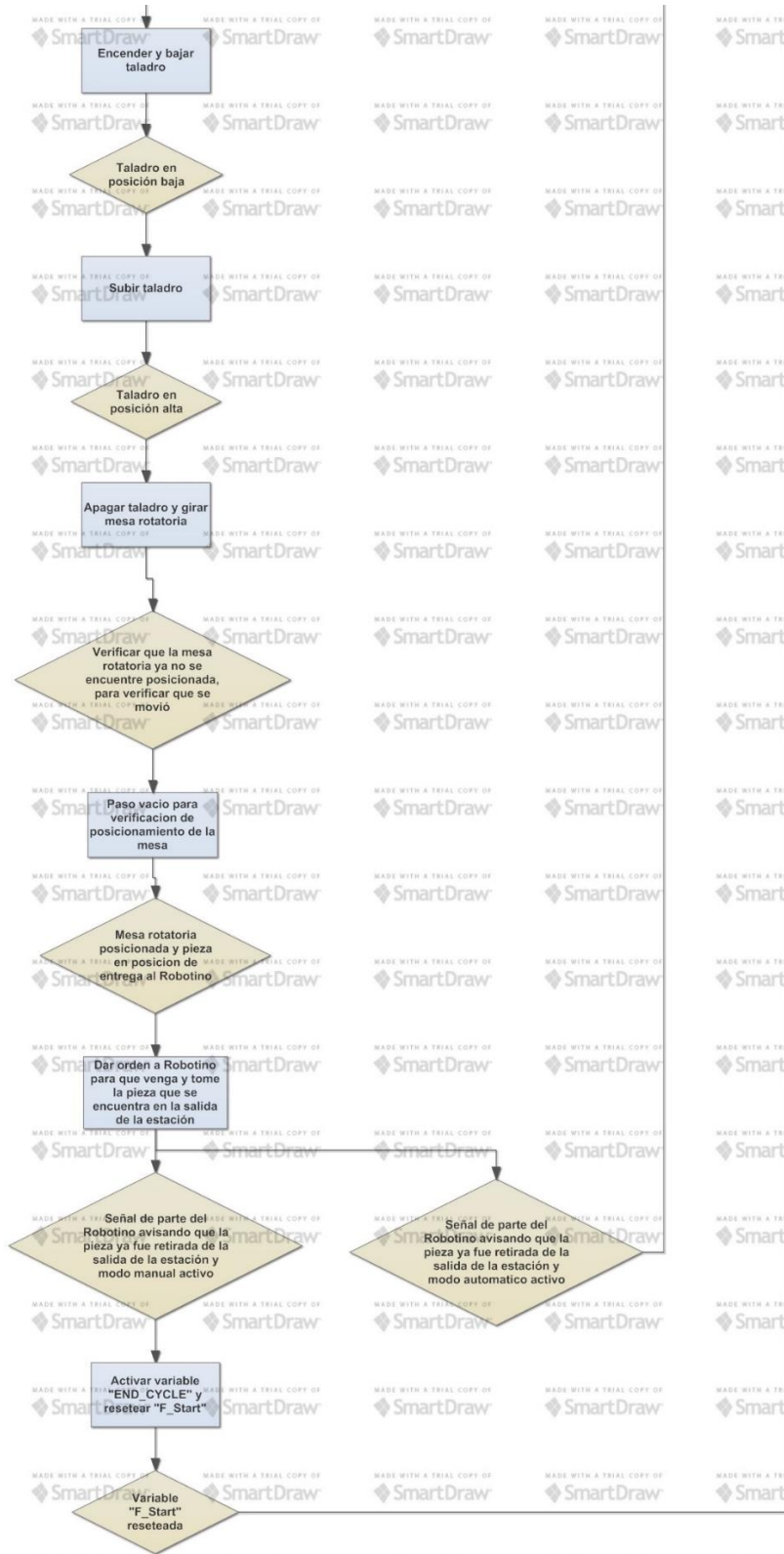


Figura 34. Diagrama de Flujo Procesamiento. Parte 4.

Anexo 3. Diagrama de Flujo estación Manipulación.

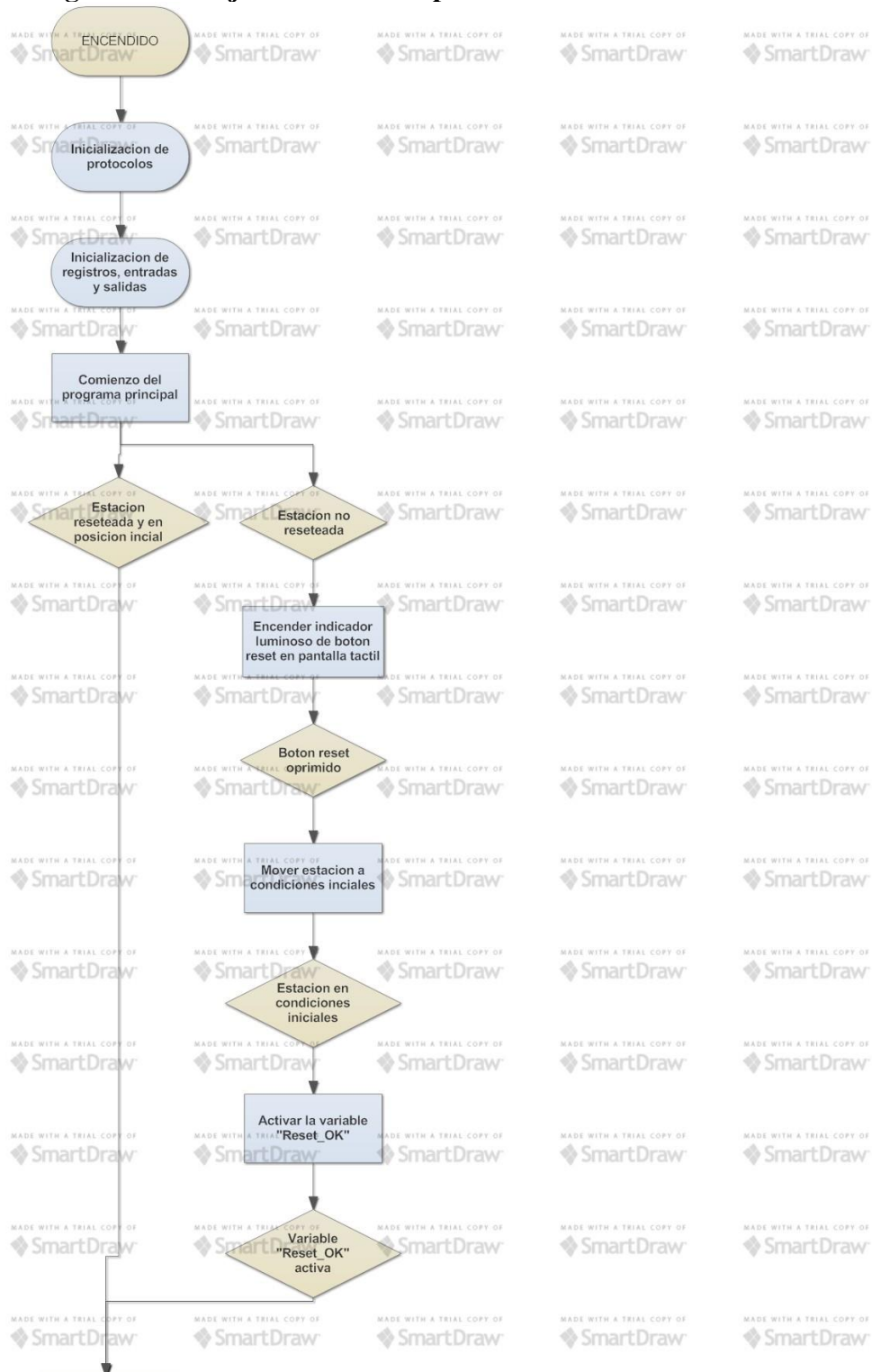


Figura 35. Diagrama de Flujo Manipulación. Parte 1.

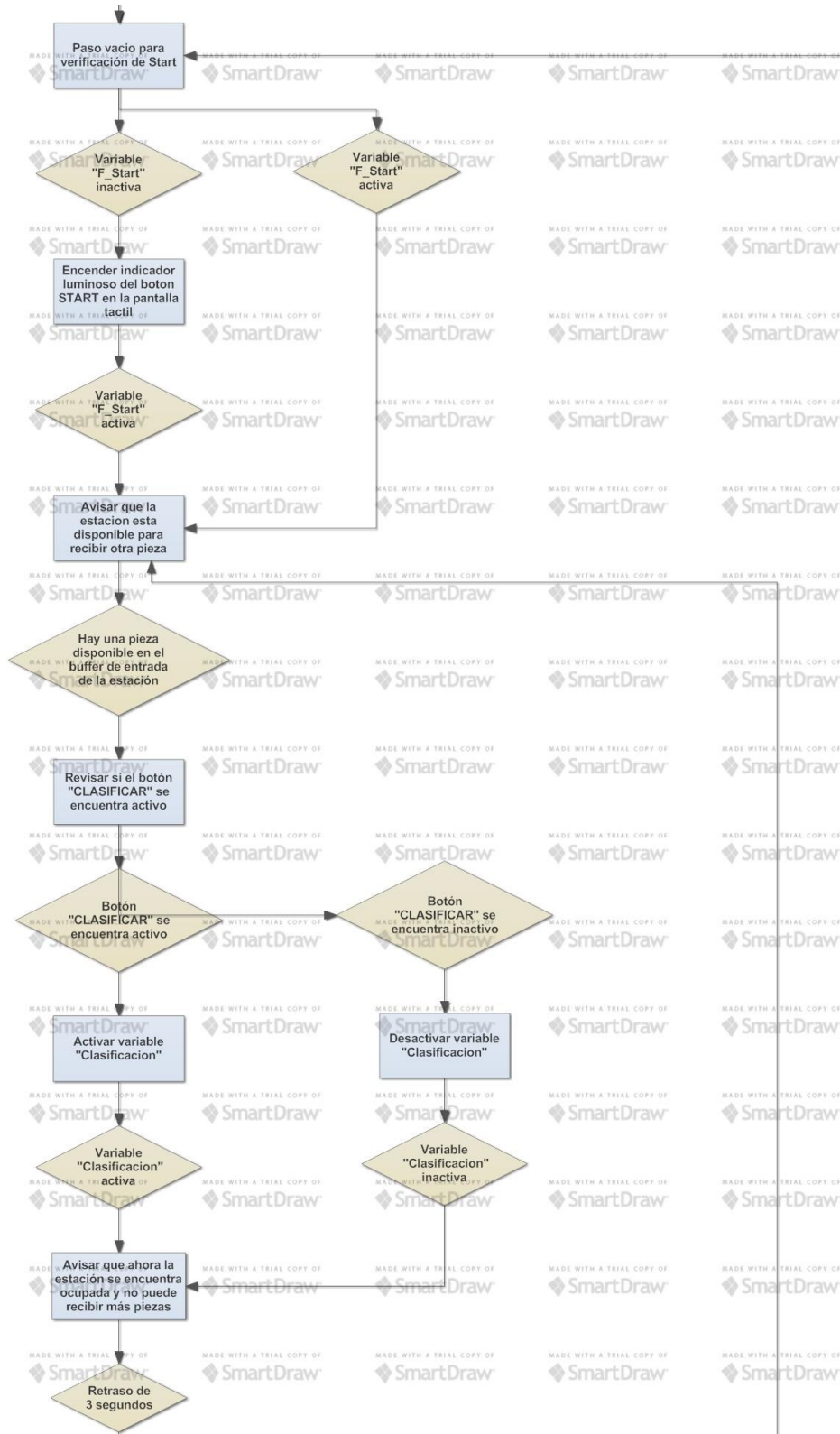


Figura 36. Diagrama de Flujo Manipulación. Parte 2.

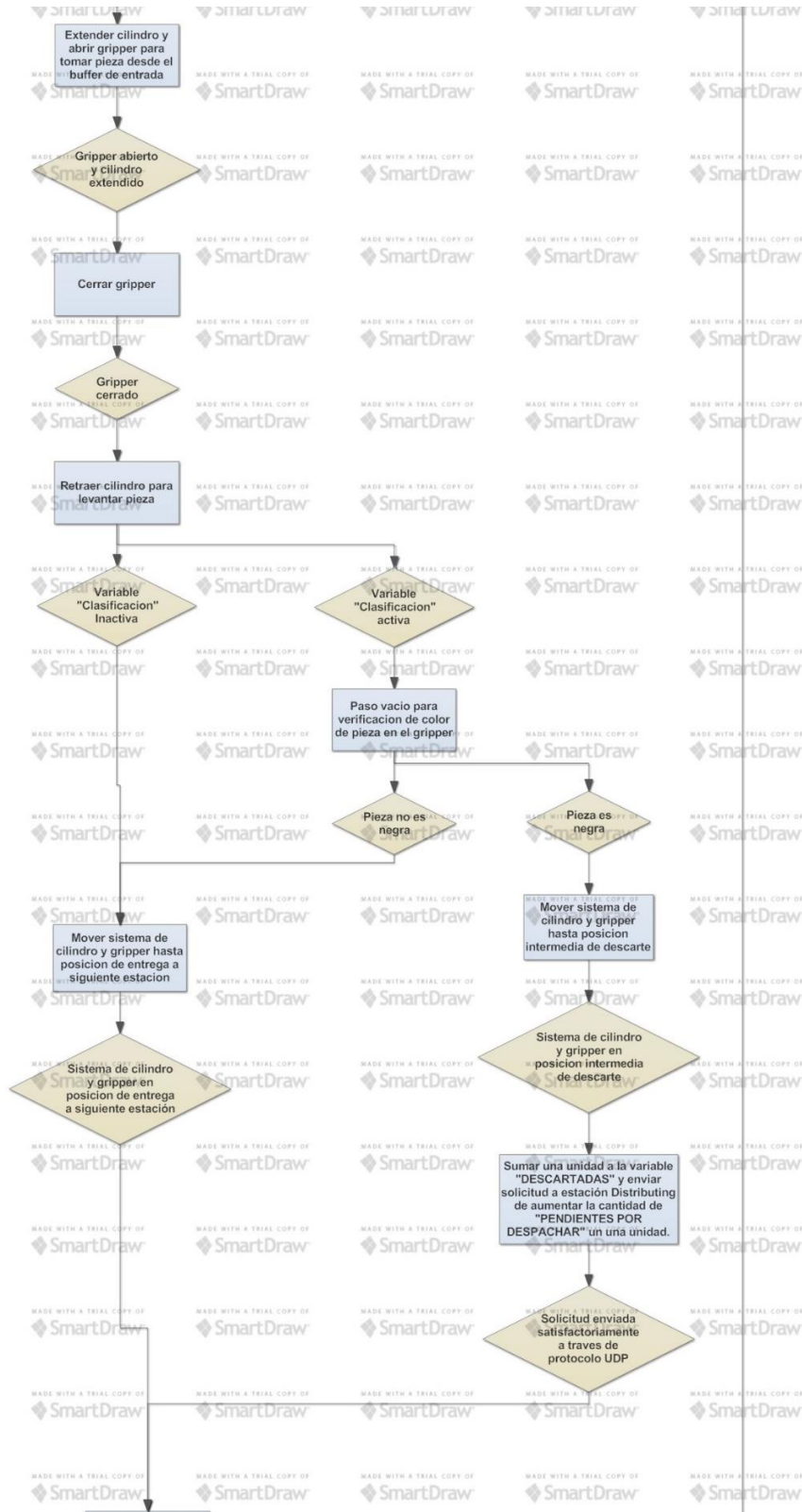


Figura 37. Diagrama de Flujo Manipulación. Parte 3.

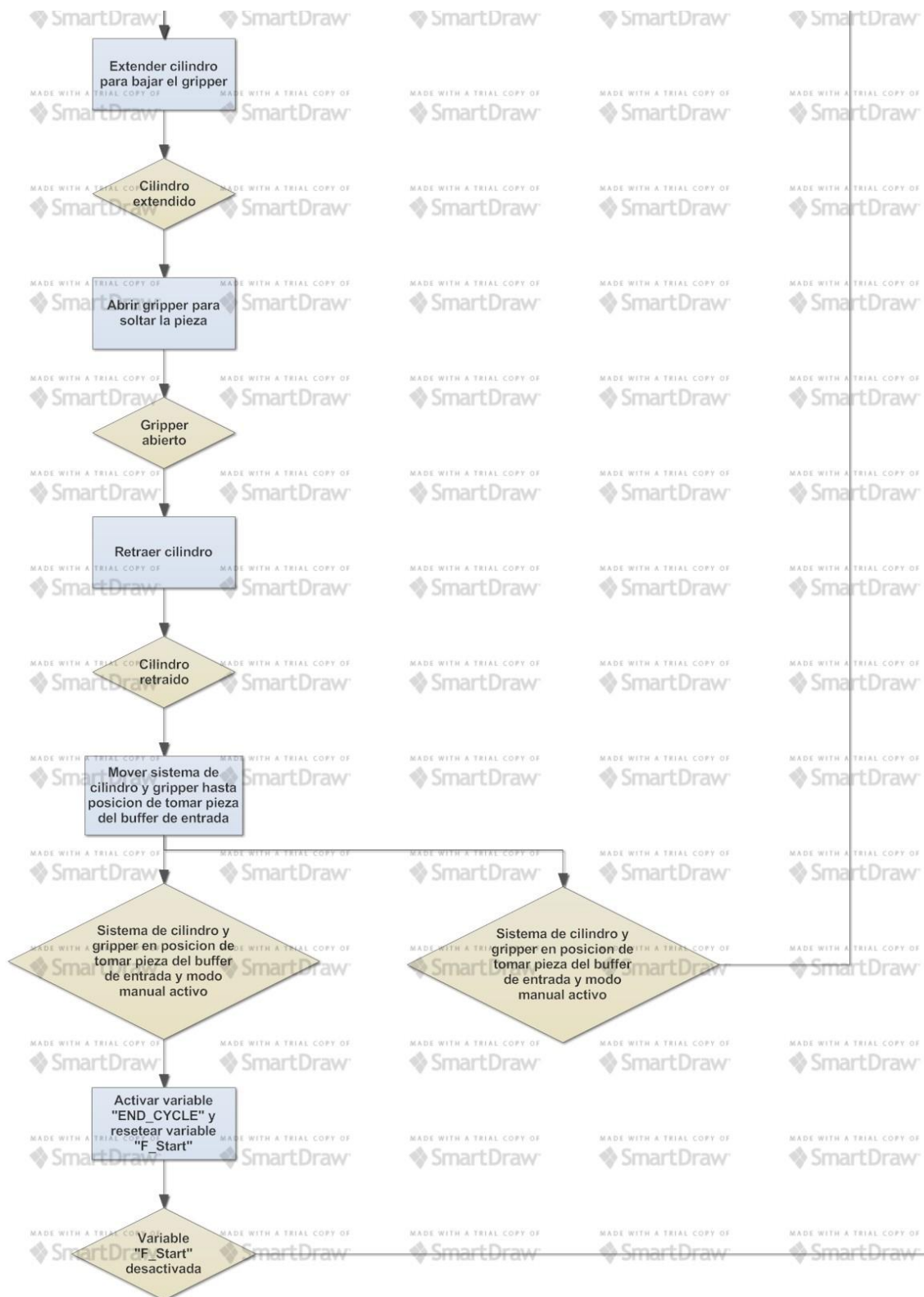


Figura 38. Diagrama de Flujo Manipulación.Parte 4.

Anexo 4. Diagrama de Flujo estación Clasificación

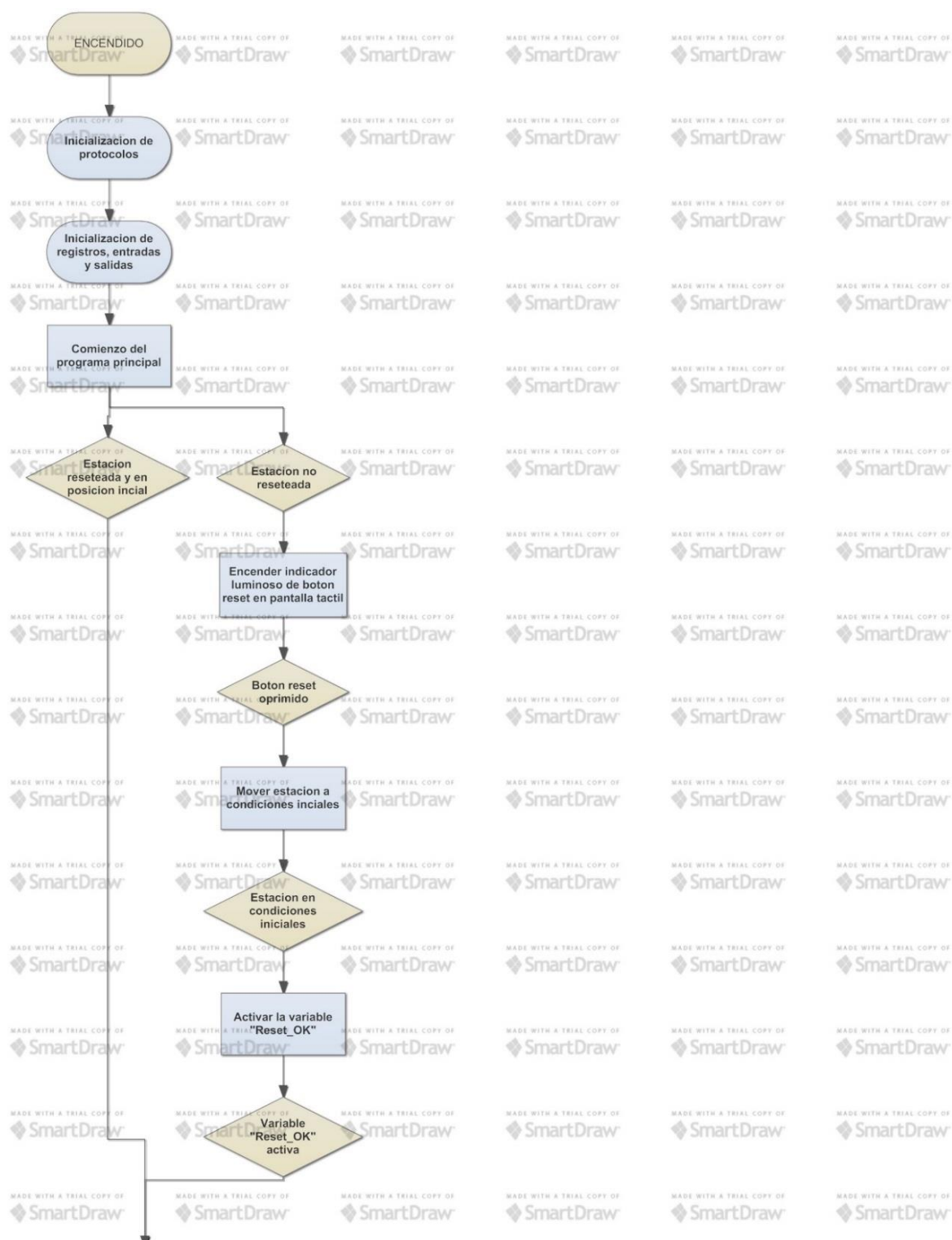


Figura 39. Diagrama de Flujo Clasificación. Parte 1.

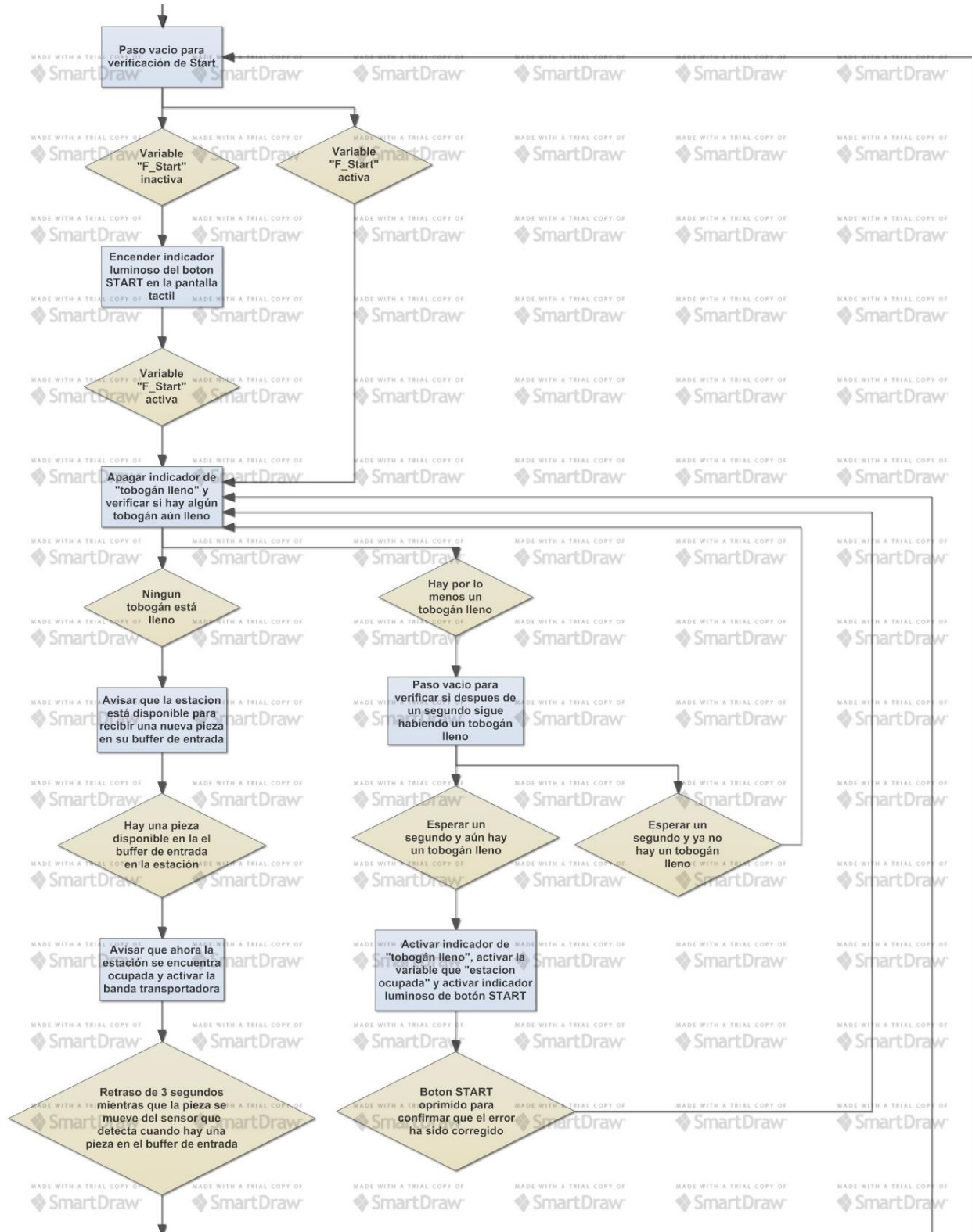


Figura 40. Diagrama de Flujo Clasificación.Parte 2.

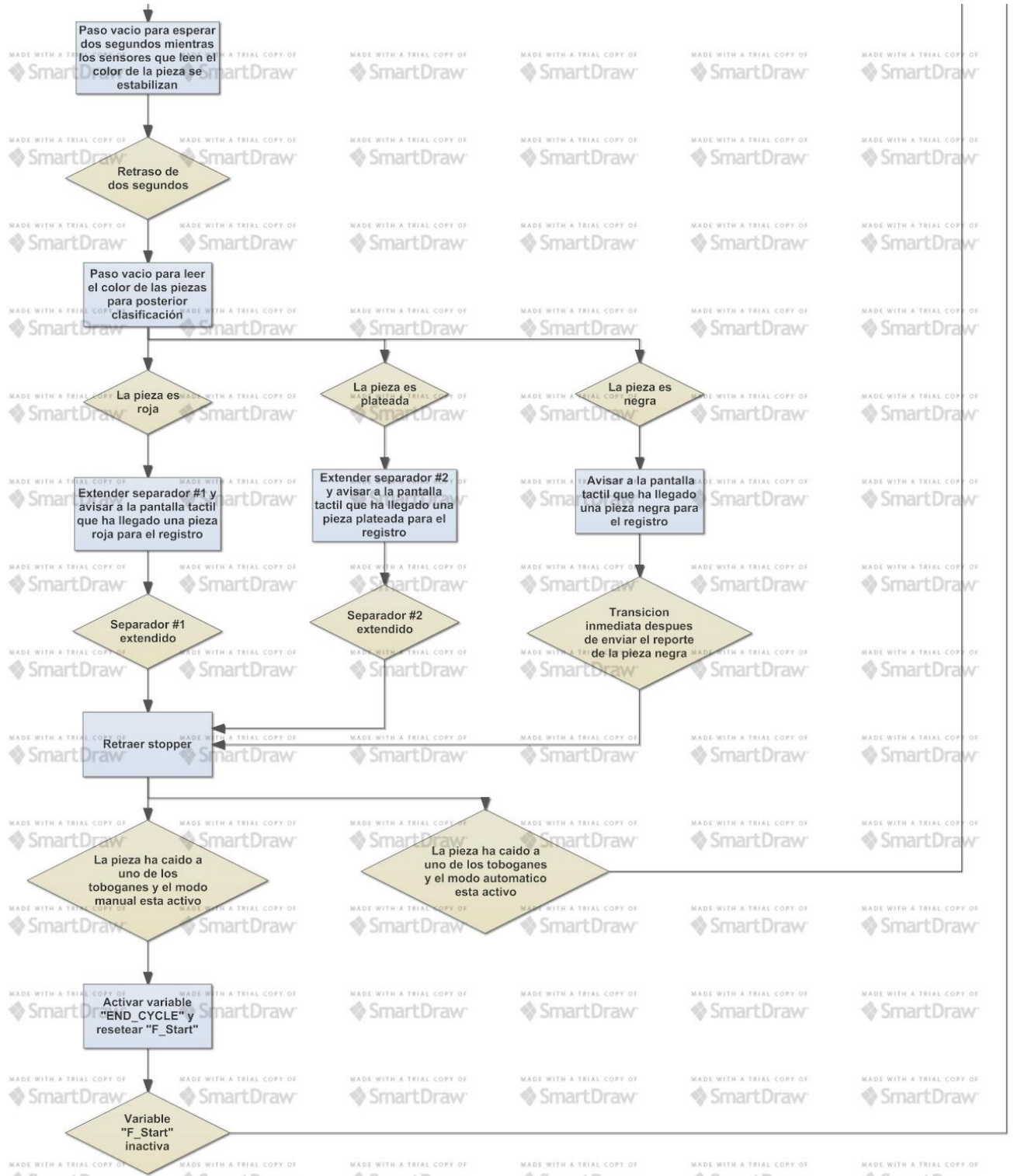


Figura 41 Diagrama de Flujo Clasificación.Parte 3.

Anexo 5. Planos estructura diseñá del Robotino.

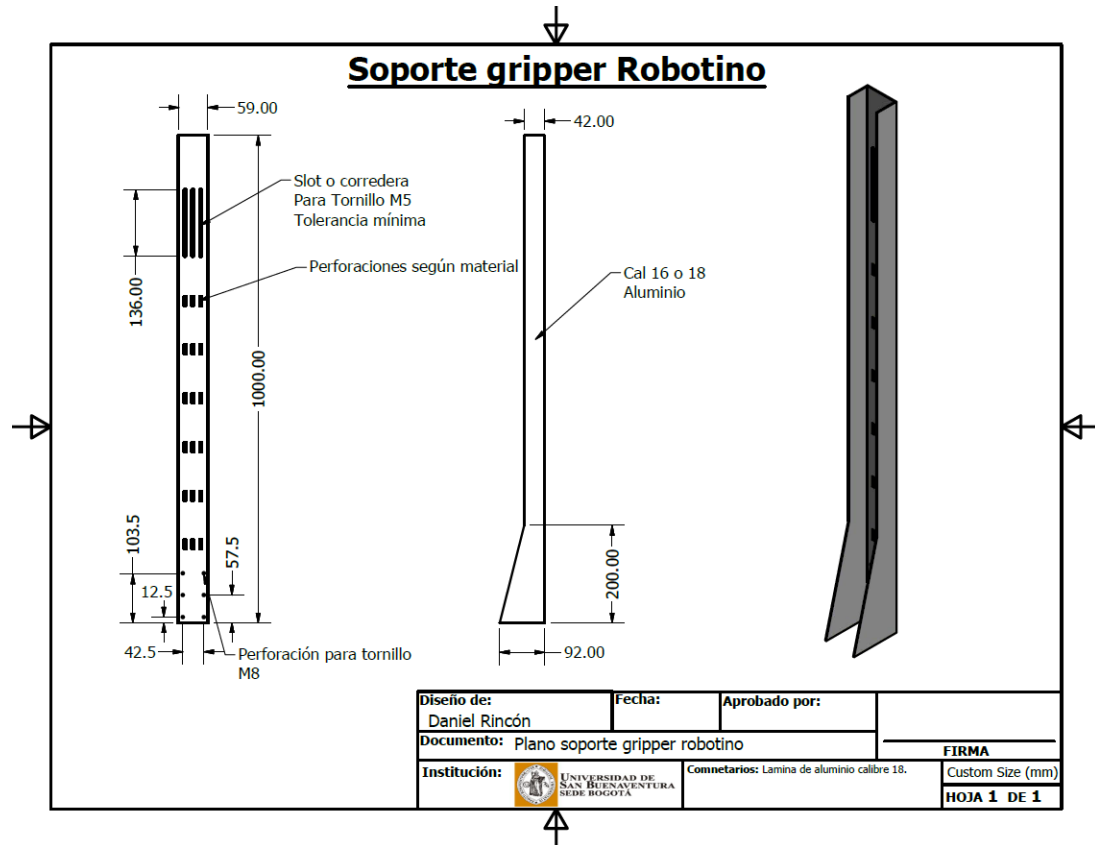


Figura 42. Plano de torre para gripper de Robotino.

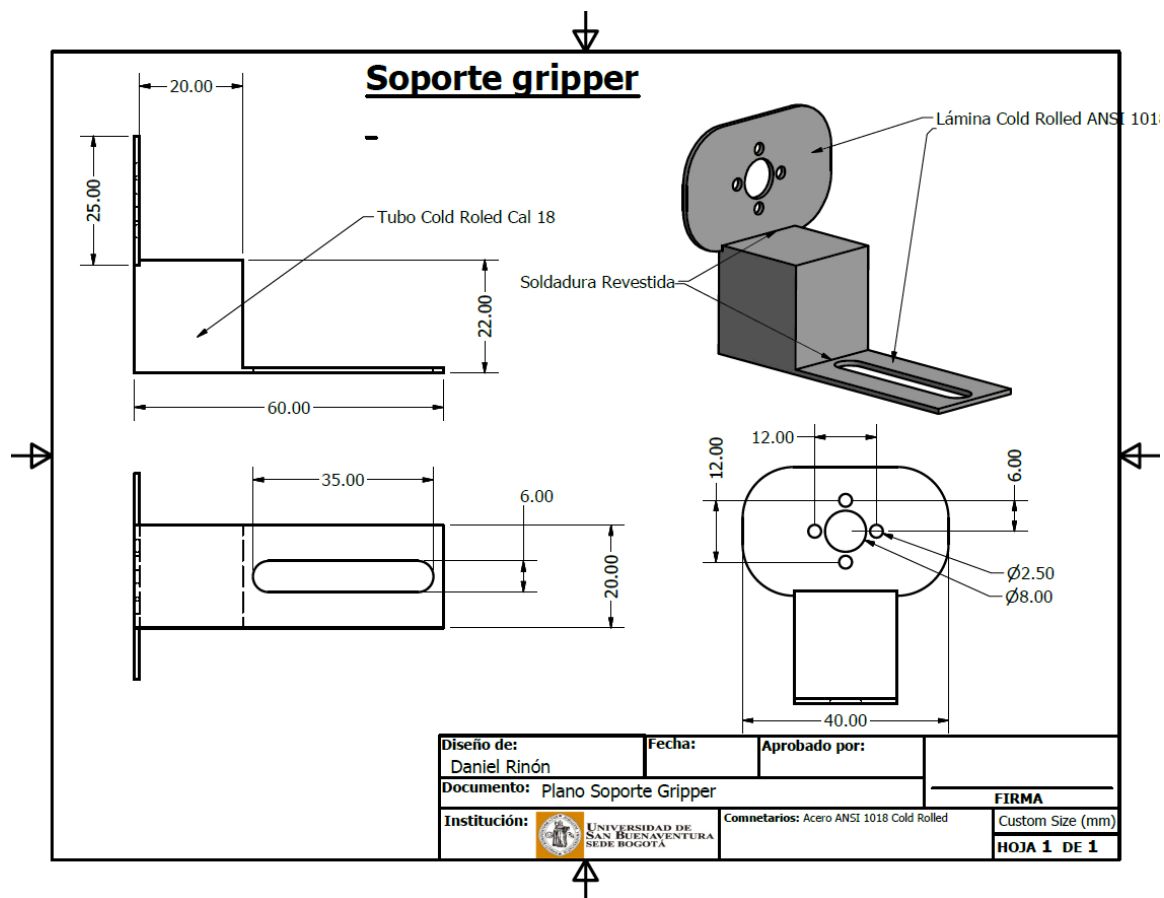


Figura 43. Plano de soporte para gripper de Robotino

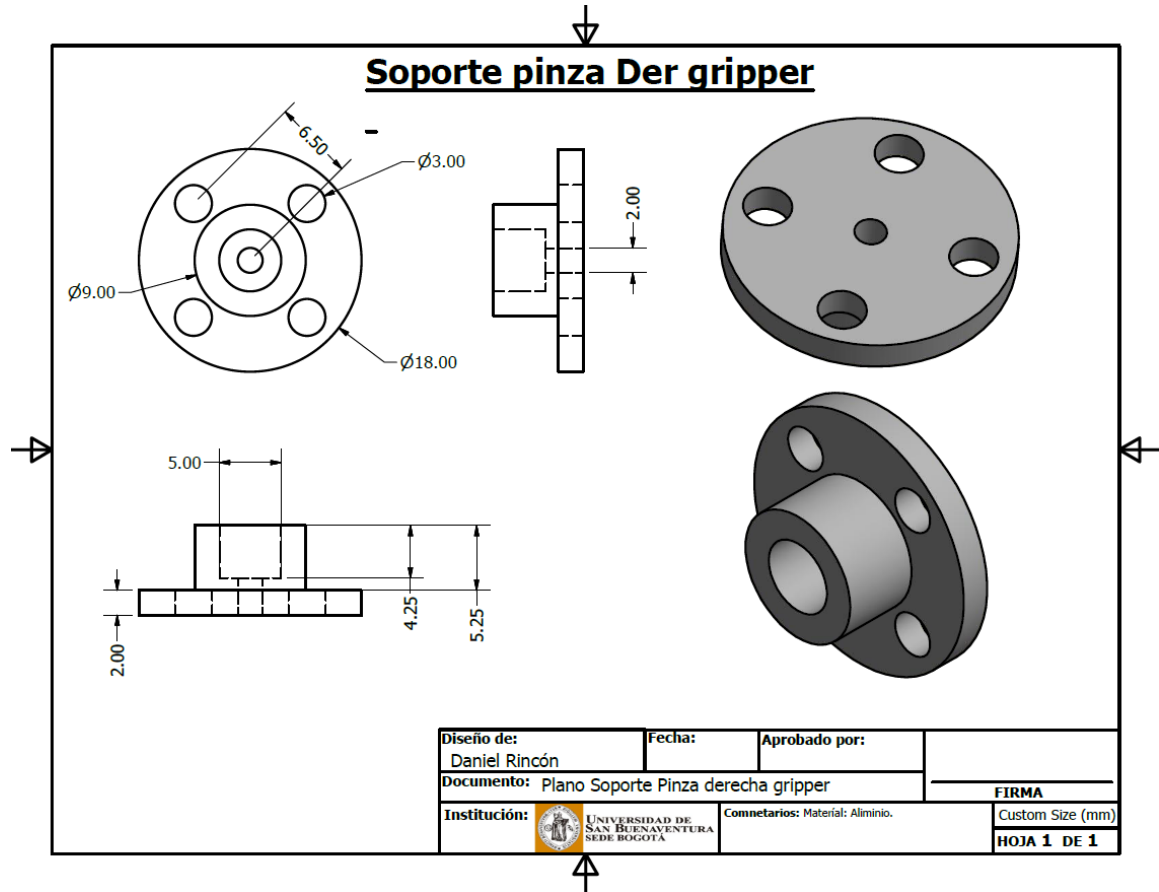


Figura 44. Plano soporte Pinza derecha del gripper.

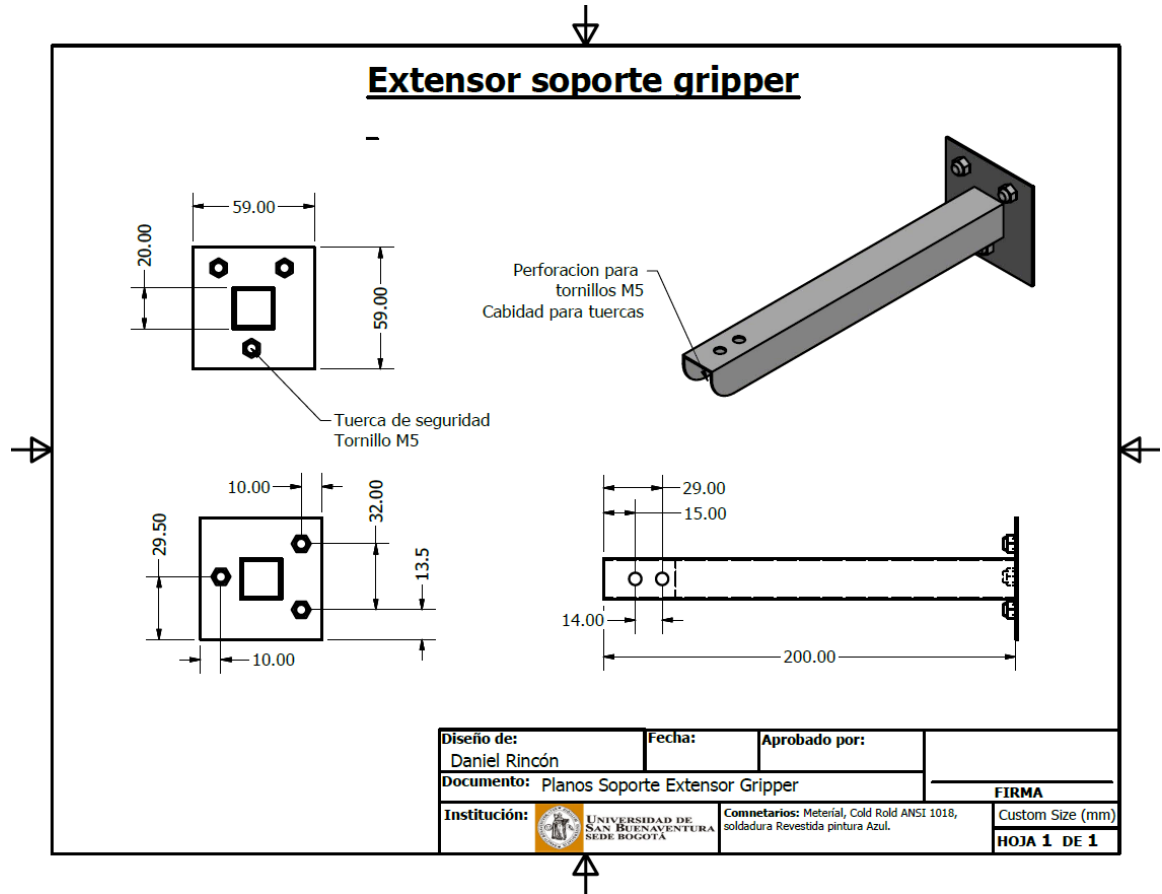


Figura 45. Plano soporte extensor de gripper para Robotino

Anexo 6. CD con programas adjuntos.

Se adjunta un CD con los respectivos programas utilizados para la realización de este proyecto.