



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES**

**Tema:**

---

**IMPLEMENTACIÓN DE MULTIAGENTES BAJO LA NORMA IEC-61499  
PARA RECONFIGURACIÓN DINÁMICA DE AUTOMATIZACIÓN DE  
PROCESOS EN LA ESTACIÓN FESTO MPS 500.**

---

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Electrónica y Comunicaciones.

**ÁREA:** Electrónica

**LÍNEA DE INVESTIGACIÓN:** Tecnología de la Información y Sistemas de Control

**AUTORES:** Willian David Morales Chicaiza

Jéssica Cristina Tituaña Gallardo

**TUTOR:** Ing. Paulina Elizabeth Ayala Baño

Ambato - Ecuador

marzo – 2023

## **APROBACIÓN DEL TUTOR**

En calidad de tutora del Trabajo de Titulación con el tema: IMPLEMENTACIÓN DE MULTIAGENTES BAJO LA NORMA IEC-61499 PARA RECONFIGURACIÓN DINÁMICA DE AUTOMATIZACIÓN DE PROCESOS EN LA ESTACIÓN FESTO MPS 500, desarrollado bajo la modalidad de Proyecto de Investigación por el señor Willian David Morales Chicaiza y por la señorita Jéssica Cristina Tituaña Gallardo, estudiantes de la carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que los estudiantes han sido tutorados durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2023.

-----  
Ing. Paulina Elizabeth Ayala Baño

TUTORA

## AUTORÍA

El presente Proyecto de Investigación titulado IMPLEMENTACIÓN DE MULTIAGENTES BAJO LA NORMA IEC-61499 PARA RECONFIGURACIÓN DINÁMICA DE AUTOMATIZACIÓN DE PROCESOS EN LA ESTACIÓN FESTO MPS 500, es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad de los autores.

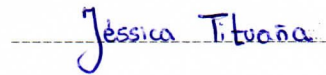
Ambato, marzo 2023



Willian David Morales Chicaiza

CI: 1805065461

AUTOR



Jéssica Cristina Tituaña Gallardo

CI: 0502926876

AUTORA

## DERECHOS DE AUTOR

Se autoriza a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta de procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

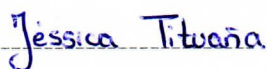
Ambato, marzo 2023



Willian David Morales Chicaiza

CI: 1805065461

AUTOR



Jéssica Cristina Tituaña Gallardo

CI: 0502926876

AUTORA

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Morales Chicaiza Willian David y por la señorita Tituaña Gallardo Jéssica Cristina, estudiantes de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Titulación, titulado IMPLEMENTACIÓN DE MULTIAGENTES BAJO LA NORMA IEC-61499 PARA RECONFIGURACIÓN DINÁMICA DE AUTOMATIZACIÓN DE PROCESOS EN LA ESTACIÓN FESTO MPS 500, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, marzo 2023.

-----

Ing. Pilar Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----

Ing. Ana Pamela Castro Martín, Mg  
PROFESOR CALIFICADOR

-----

Ing. Marcelo Vladimir García Sánchez, Mg  
PROFESOR CALIFICADOR

## **DEDICATORIA**

A mi padre quien en vida demostró su amor incondicional mediante su apoyo, consejos, disciplina y fuerza, dejando así un legado de vida que ni la fuerza del tiempo podrá borrar. A mi madre quien con sus sonrisas y afectos es capaz de borrar cualquier tristeza en mi alma; a mis hermanos quienes con sus consejos, confianza, apoyo y alegrías logran ser el complemento perfecto en este viaje llamado vida; a mi abuelita quien me provee fuerza desde las promesas que nos hicimos y que aún faltan cumplir; finalmente a mi compañera de carrera, de ella aprendí el verdadero significado de la palabra amigo.

Willian Morales

Este proyecto de investigación lo dedico a mi madre que con tanto sacrificio ha sacado adelante a sus dos hijas, gracias por comprender y por demostrar con su ejemplo que el estudio es importante y que con esfuerzo se logran los objetivos, a mi segunda madre “Mami Ubita” la persona que ha sabido expresar todo su amor y su apoyo en el momento preciso, porque nunca me faltó sus deliciosas comidas cuando tenía que viajar, un abrazo, un beso y la bendición, a mi hermana que siempre seguí sus pasos, mi ejemplo y mi admiración total para ella, finalmente a mis grandes amigos Cynthia Endara que desde el prepo hasta el día de hoy me ha demostrado su lealtad y su amistad sincera, y Willian Morales quien ha demostrado ser una persona fuerte y valiente, y que con sus palabras ha sabido aconsejarme en el momento indicado.

Jéssica Tituaña

## ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR.....	iv
APROBACIÓN DEL TRIBUNAL DE GRADO .....	v
DEDICATORIA .....	vi
ÍNDICE GENERAL DE CONTENIDOS.....	vii
ÍNDICE DE TABLAS .....	x
ÍNDICE DE FIGURAS.....	x
RESUMEN EJECUTIVO .....	xiii
ABSTRACT.....	xiv
CAPITULO I.....	6
MARCO TEÓRICO.....	6
1.1. Antecedentes Investigativos .....	6
1.1.1. Contextualización del problema .....	7
1.1.2. Fundamentación Teórica.....	9
Agente inteligente de software.....	9
Sistema multiagente.....	10
Arquitectura reconfigurable .....	10
Reconfiguración dinámica .....	10
Norma IEC-61499.....	11
Bloques funcionales (FBs) .....	12
Hardware .....	13
ET 200SP Open Controller CPU 1515S PC .....	13
Módulos de entradas y salidas (E/S) .....	15

Sistema de producción modular (MPS) .....	16
FESTO MPS 500.....	16
Software .....	17
4DIAC-FORTE .....	17
ODK 1500S V2.5 .....	18
SPADE .....	19
Microsoft Visual Studio 2017 .....	19
Openfire.....	20
STEP 7 TIA PORTAL V16 .....	21
FACTORY IO IO 2.4.3.....	21
1.3. Objetivos.....	22
1.3.1. Objetivo General.....	22
1.3.2. Objetivos Específicos .....	22
CAPITULO II.- METODOLOGÍA .....	23
2.1. Materiales .....	23
2.2. Métodos .....	23
2.2.1. Modalidad de la Investigación.....	23
Investigación Aplicada.....	23
Investigación Bibliográfica .....	24
Investigación Campo.....	24
2.2.2. Recolección de Información .....	24
2.2.3. Procesamiento y Análisis de Datos.....	24
2.2.4. Desarrollo del proyecto.....	25
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN .....	26
3.1. Análisis y discusión de los resultados .....	26
3.1.1 Módulo Físico .....	32
4DIAC en el Módulo Físico .....	33



Archivos generados en 4DIAC FORTE .....	36
Archivos generados ODK 1500S V2.5 .....	39
3.1.2. Módulo Digital .....	47
Multiagentes .....	47
3.1.3. Comunicación entre el Módulo Físico y el Módulo Digital .....	50
3.1.4. Pruebas del sistema implementado .....	53
<b>CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>58</b>
4.1 Conclusiones.....	58
4.2 Recomendaciones .....	59
<b>C. MATERIALES DE REFERENCIA.....</b>	<b>60</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>60</b>
<b>ANEXOS.....</b>	<b>65</b>

## ÍNDICE DE TABLAS

Tabla 1 Características técnicas del CPU 1515S PC .....	65
Tabla 3 Datos técnicos del DI 8×24VDC HF .....	66
Tabla 4. Datos técnicos del DQ 8x24VDC/0.5A HF .....	68
Tabla 5. Datos técnicos del AQ 2×U/I HS .....	69
Tabla 5. Registro del entradas y salidas en el IPC y 4DIAC .....	76

## ÍNDICE DE FIGURAS

Figura 1 Esquema básico de funcionamiento de un agente. ....	9
Figura 2. Modelo IEC-61499 .....	11
Figura 3. Estructura de un bloque funcional (FB).....	12
Figura 4. ET 200SP Open Controller CPU 1515S PC .....	13
Figura 5. Vista frontal de CPU 1515SP .....	14
Figura 6. Configuración de la CPU 1515sp .....	14
Figura 7. Tarjetas de E/S en el ET 200SP .....	15
Figura 8. Vista del módulo DQ .....	16
Figura 9. Interfaz 4DIAC .....	17
Figura 10 Esquema general de 4DIAC-FORTE .....	18
Figura 11. Entornos de ODK .....	18
Figura 12. Comportamientos básicos de un agente SPADE. ....	19
Figura 13 Área de trabajo de Visual Studio Code. ....	20
Figura 14 Interfaz de Openfire.....	20
Figura 15. TIA PORTAL V16 .....	21
Figura 16 Escenas predeterminadas en Factory IO.....	22
Figura 17. Simulación en Factory IO de la estación de clasificación Festo .....	27
Figura 18. Diagrama funcional del módulo simulado.....	28
Figura 19. Esquema de Proyecto MF y MD .....	29
Figura 20. Arquitectura proyecto multiagente .....	31
Figura 21. Esquema de conexión física: controlador ATMEGA328 .....	32
con ET 200SP principal y respaldo.....	32
Figura 22. Modelo del sistema 1 .....	33
Figura 23. Esquema general de conexión entre 4DIAC y ODK 1500s v2.5 .....	35

Figura 24. Estructura general de 4DIAC-FORTE.....	36
Figura 25. Direccionamiento hacia DLL de Windows en archivo .cpp.....	37
Figura 26. Sistema de destino de los archivos .h y .cpp de cada bloque funcional ..	38
Figura 27. Empaquetamiento de archivos a través de CMAKE .....	38
Figura 28. Arquitectura ODK .....	39
Figura 29. Funciones del ODK del módulo de salida digital y analógica.....	40
Figura 30. Bloques funcionales generados por ODK a través del archivo SCL.....	41
Figura 31. Conexión entre 4DIAC-FORTE a través de los archivos DLLs .....	41
Figura 32. MF Aplicación desarrollada en 4Diac para el control de salidas y entradas de los IPCs.....	43
Figura 33. Bloques Funcionales entradas y salidas del IPC.....	45
Figura 34. Evento de arranque .....	45
Figura 35. BF publish y subscribe.....	46
Figura 36. Sub-aplicación para envío de datos al publish.....	47
Figura 37. Lógica de funcionamiento de los agentes.....	49
Figura 38. Módulo Digital.....	50
Figura 39 Conexión de los módulos.....	51
Figura 40. Esquema Final de control de proceso .....	52
Figura 41. Conexión hardware.....	53
Figura 42. Puntos de prueba.....	54
Figura 43. Tiempo registrado por Wireshark .....	55
Figura 44. Gráfica comparativa de tiempos .....	56
Figura 45. Gráfica de retardo Whirespark.....	56
Figura 46. ET 200SP Open Controller CPU 1515S PC en TÍA PORTAL.....	70
Figura 47 Proceso de clasificación en Factory IO .....	70
Figura 48 Comunicación con el IPC 1 .....	71
Figura 49. Comunicación con el IPC 2 .....	71
Figura 50. Agente de Monitoreo 1 (AM1).....	72
Figura 51. Agente de Monitoreo 2 (AM2).....	72
Figura 52 Agente de Análisis (AA) .....	73
Figura 53. Agente de ejecución (AE).....	73
Figura 54. Ejecución del 4DIAC FORTE.....	74

Figura 55 Diagrama de flujo para la decodificación de las señales virtuales en señales físicas en ATMEGA328 Shield Ethernet.....	75
--	----

## **RESUMEN EJECUTIVO**

La industria 4.0 incorpora sistemas de automatización descentralizados para reducir recursos y gastos permitiendo sistemas más rápidos y eficientes a diferencia de los sistemas centralizados que tras una paralización en dicho sistema provoca daños en los equipos y genera pérdidas económicas, es por esto que en el presente trabajo de investigación se expone la implementación de multiagentes en los equipos ET 200SP CPU 1515SP creados bajo la norma IEC-61499 capaces de ejecutar una reconfiguración dinámica en casos de fallos por desconexión en la estación de clasificación FESTO MPS 500; logrando que el proceso de clasificación no sea paralizado, sino que continúe desde el punto en el que presentó la falla sin que sea necesario un paro prolongado y un reinicio manual del proceso. El control creado incluye agentes embebidos dentro de bloques funcionales programados para funcionar en conjunto con un sistema de respaldo. Una vez implementado el proyecto se realiza pruebas de funcionamiento estableciendo su tiempo de reconexión al momento de que el equipo principal deja de funcionar comprobando así funcionalidad y eficiencia.

**Palabras clave:** sistema multiagente, IEC-61499, computadoras industriales (IPCs), reconfiguración dinámica.

## ABSTRACT

Industry 4.0 incorporates decentralized automation systems to reduce resources and expenses allowing faster and more efficient systems unlike centralized systems that after a stoppage in the system causes damage to equipment and generates economic losses, which is why in this project presents the implementation of multi-agents in the ET 200SP CPU 1515SP equipment created under the IEC-61499 standard capable of executing a dynamic reconfiguration in cases of disconnection failures in the FESTO MPS 500 classification station; achieving that the classification process is not paralyzed but continues from the point where it presented the failure without requiring a prolonged stoppage and a manual restart of the process. The control created includes embedded agents within functional blocks programmed to work in conjunction with a backup system. Once the project is implemented, functional tests are carried out to establish the reconnection time when the main equipment stops working, thus verifying functionality and efficiency.

**Keywords:** multi-agent system, IEC-61499, industrial computers (IPCs), dynamic reconfiguration.

## CAPITULO I

### MARCO TEÓRICO

#### 1.1. Antecedentes Investigativos

En la investigación de artículos científicos y proyectos de investigación referentes a sistemas multiagentes bajo la norma IEC-61499 para la reconfiguración dinámica, se analizaron los siguientes trabajos:

El estándar IEC-61499 mediante la incorporación de software con especificaciones abiertas ha sido desarrollado con tal esfuerzo de obtener características tales como: la interoperabilidad, configurabilidad y portabilidad [1], permitiendo la conexión entre la automatización y los avances de la cuarta revolución industrial. Investigaciones tales como las desarrolladas por Caiza et al.,[2], Garcia et al.,[3], Pang et al., [4], ; Lepuschitz et al, [5], mencionan propuestas de innovación de modelos para desarrollar software de control de procesos aplicando la norma IEC-61499, además en los últimos años la tecnología Industrial Internet of Things (IIoT) está teniendo una gran acogida al tratarse de automatizar la industria ya que incorpora el uso de sensores, actuadores y unidades de control en base a IoT.

Drozdov et al, [1] analiza la arquitectura IEC-61499 a través de estándares abiertos permitiendo el progreso de aplicaciones de control con lógica descentralizada mediante la arquitectura de los bloques funcionales (FB) basados en el estándar IEC-61499, logrando aumentar el rendimiento del sistema y reducir el tiempo de respuesta en eventos de entrada y salida, así mismo mediante la comunicación de los bloques funcionales se obtiene modelos complejos y ejecución periódica [6].

Los sistemas multiagentes y holónicas tienen la facultad de ser flexible, al ser sistemas autónomas cada elemento persigue su propia finalidad [7], al verse limitados por sus capacidades locales el sistema no es capaz de cumplir con objetivos complejos, pero si estas entidades colaboran de manera sincronizada tienen la capacidad necesaria para realizar actividades complejas [8] [9], corroborando esta idea Avila et al., [10] menciona que una arquitectura holónica posee propiedades de autonomía y cooperación, cada entidad toma decisiones con el fin de cumplir sus objetivos

individuales y en el caso de no lograrlo, otra entidad aporta con más funcionalidades permitiendo alcanzar los objetivos en su totalidad.

Los agentes de software analizan los datos y el estado del sistema de fabricación, además, un sistema multiagente (MAS) utiliza relaciones organizacionales compuestas por varias entidades en un solo entorno permitiendo reconfigurar y controlar las aplicaciones según la necesidad de la planta obteniendo un alto nivel de autoorganización y solución a los problemas de los sistemas de control.[11]

En base a estas dos tecnologías se puede dar paso a su cooperación conjunta facilitando la mejora en varios aspectos, que incluyen: robustez, flexibilidad, reconfigurabilidad, redistribución, costo y escalabilidad. El planteamiento de esta idea ha sido estudiada a través del tiempo; por ejemplo Marick y et al,[12] plantean el uso de agentes para la reconfiguración en tiempo real de la arquitectura de automatización distribuida IEC-61499, además de una lógica de programación que se compone de un lenguaje de bajo nivel empleando bloques de función conjuntamente con agentes de software.

El objetivo de este proyecto es desarrollar una arquitectura que facilite la autoadaptación y la evolución dinámica según el avance de la industria. Por lo cual es necesario, al contrario de implementaciones realizadas anteriormente, que la lógica de control emplee los equipos de planta para que se ejecuten de manera local y estén en comunicación constante. Los dispositivos utilizados aparte de los IPC de la marca Siemens serán tarjetas de control de fácil acceso.

### **1.1.1. Contextualización del problema**

La industria 4.0 incorpora técnicas avanzadas en los sistemas de automatización para control industrial, combinando los procesos de producción con tecnologías inteligentes tales como: robótica, inteligencia artificial, internet de las cosas (IOT), entre otros. Lo que ha permitido el desarrollo de sistemas más complejos y robustos que se adapten a diversos ambientes industriales con el fin de mejorar y optimizar una gran cantidad de procesos de automatización [13].

Dichos procesos se enfrentan a varios cambios que se generan en la industria, por lo cual se debe considerar la personalización y demanda, así como reducir el tiempo de elaboración del producto. Además, la adaptación rápida del producto en respuesta a la demanda es uno de los factores claves para que las industrias sean competitivas [14].



La modularización de la planta ha sido un aspecto clave para tener tiempos de producción más cortos, dado que la evolución constante de la tecnología ha permitido la incorporación de software con especificaciones abiertas en el campo industrial y con ello aspectos como la portabilidad, interoperabilidad y configurabilidad de los procesos; características que han provocado un interés constante por parte de las empresas de automatización de la industria [15].

Una de las mayores dificultades que afronta los sistemas centralizados es la paralización de un proceso provocado por desperfectos o desconexiones del hardware que contiene la lógica de control. Cuando se produce un defecto de esta magnitud es necesario detener completamente el sistema, teniendo como consecuencias: pérdidas de datos, pérdidas económicas, tiempos muertos de producción y reinicio total del sistema [16].

La reconfiguración dinámica se presenta como una solución que facilita: el manejo de datos de información, ciclos de vida más cortos, flexibilidad en las configuraciones y arreglos de la planta. Para esto se han desarrollado estándares de arquitectura abierta con sistemas centralizados que agilicen la operabilidad, comunicación y sustitución que se utiliza en los esfuerzos de modularización de la planta. Los estándares más importantes considerados en la digitalización industrial son: IEC-61499 y Open Process Automation Standard (OPAS) impulsado por OPAF.[17]

Por tanto, el presente proyecto busca dar solución a este problema mediante el desarrollo de una plataforma abierta de reconfiguración dinámica ante posibles fallos empleando computadoras industriales IPC marca Siemens con el objetivo de que un segundo dispositivo hardware tome el control en el caso de presentarse un fallo por desconexión del sistema principal, sin la necesidad de parar con el proceso de producción reduciendo así tiempos muertos y tomando el control desde el punto antes de la falla. Se propone desarrollar una plataforma con multiagentes bajo la norma IEC 61499 en la estación FESTO MPS 500 a través de los criterios de arquitectura abierta dado por OPAS.

Con la tecnología actual a disposición en los laboratorios de la facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato es viable la construcción y simulación de la plataforma multiagente empleando los computadores industriales IPC sobre el módulo de clasificación de la estación FESTO MPS 500.

El trabajo de titulación servirá como aporte para la investigación: “Plataforma de reconfiguración dinámica de control automático de un proceso industrial basado en IEC-61499 y multiagentes” aprobado por el Consejo de Investigación e Innovación CONIN, mediante resolución CONIN-P-227-2020 de fecha 24 de septiembre de 2021.

### 1.1.2. Fundamentación Teórica

#### Agente inteligente de software

Son entidades que poseen un propio entendimiento para realizar acciones conjuntas destinadas a resolver los problemas de un usuario del software en ejecución, sea por decisión propia o porque algún programa lo requiera. Todos los agentes inteligentes son considerados como entidades individuales, es decir, forman parte de un programa, pero poseen control sobre sus propias acciones y movimientos, normalmente están desarrollando procesos que les muestra qué hacer y cómo. Desde el punto de vista de la inteligencia artificial un agente tiene: autonomía, capacidad de reacción, iniciativa y racionalidad. Además, puede analizar su entorno, procesar los datos adquiridos, reaccionar y dar respuesta de forma racional maximizando la eficiencia del sistema, su esquema básico de funcionamiento se refleja en la figura 1. [18]

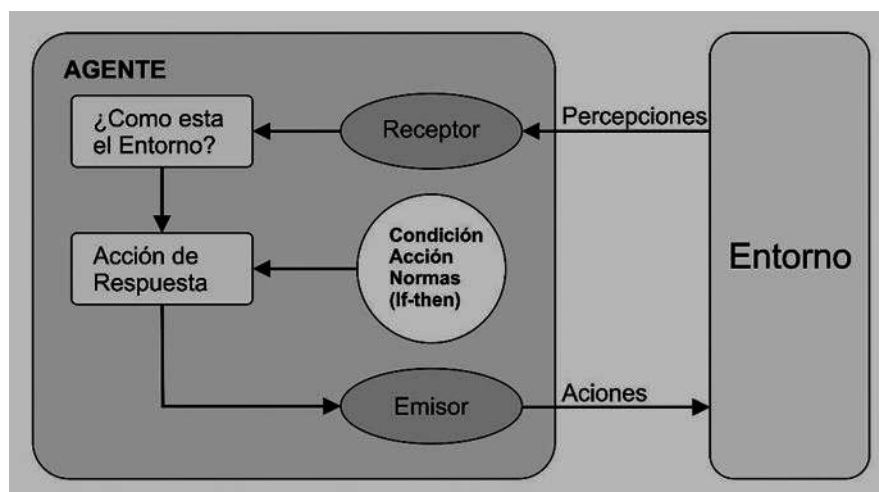


Figura 1 Esquema básico de funcionamiento de un agente. [18]

### **Sistema multiagente.**

Los sistemas multiagentes (SMA) son sistemas conformados por varios agentes inteligentes que se relacionan, cooperan y se coordinan entre cada uno de ellos. Se pueden utilizar para resolver conflictos que son grandes para un agente individual, ofreciendo la ventaja de dar respuesta a un desastre o colisión en un sistema industrial. Dichos sistemas forman procedimientos ordenados donde los nodos son sistemas de inteligencia artificial, o incluso procedimientos distribuidos donde el comportamiento coordinado de cada una de las partes da como resultado una entidad o conjunto inteligente [12].

### **Arquitectura reconfigurable**

Una estructura reconfigurable se define como aquella capaz de manipular funcionalidades del sistema, componentes y la red que los conecta mediante la variación de parámetros o estados iniciales permitiendo realizar más operaciones por ciclo de máquina; cuando la reconfiguración se la realiza de manera rápida sin ninguna sobrecarga se dice que es dinámica.

### **Reconfiguración dinámica**

La reconfiguración dinámica (DR) es la capacidad de volver a configurar un sistema mientras se encuentra en ejecución sin afectar de alguna manera las operaciones que se están realizando.

Además, la reconfiguración tiene la capacidad de variar parámetros, datos o estados iniciales para que el proceso que se realice sea idóneo.

La reconfiguración dinámica se lo presenta también como un coprocesador dedicado, ofreciendo velocidades más rápidas que los enfoques convencionales, estableciendo un equilibrio entre velocidad y eficiencia aprovechando al máximo el hardware del sistema mediante la adaptación a la computadora. Por ejemplo, diversas aplicaciones de imágenes y video, jeroglífico y redes utilizan estas características [19]

### **IEC (Comisión Electrotécnica Internacional)**

La norma IEC (Comisión Electrotécnica Internacional) es un estándar creado con un amplio alcance geográfico siendo implementado alrededor del mundo con el objetivo de brindar información técnica según diferentes puntos de vista nacionales e

internacionales. La implementación de la norma IEC es utilizada en los sistemas de automatización industrial entre ellos se incluyen: gestión de la organización, especificaciones del producto, servicios, personal entre otros según el área a ser aplicada [20]

### Norma IEC-61499

La norma IEC 61499 es una arquitectura que permite la realización de algoritmos de control aplicando una lógica descentralizada basada en la interconexión de bloques funcionales (FB) permitiendo modelar aplicaciones, además tiene la capacidad de implementar multiagentes logrando así sistemas más flexibles, brindando gran importancia a la reutilización de código. La arquitectura IEC 61499 se basa en tres aspectos importantes: portabilidad, configurabilidad, e interoperabilidad de las aplicaciones distribuidas teniendo un sistema más completo y robusto [21]

La arquitectura se basa en un modelo organizado como se muestra en la figura 2 compuesto por: [22]

- Sistema
- Dispositivo
- Recurso - aplicación
- Bloque de función.

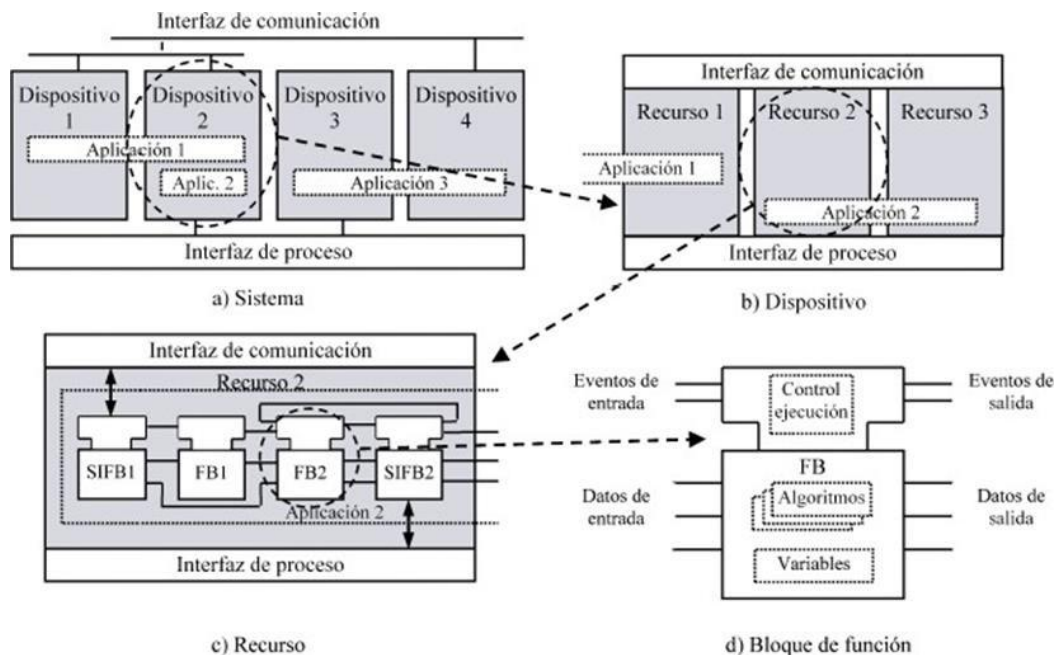
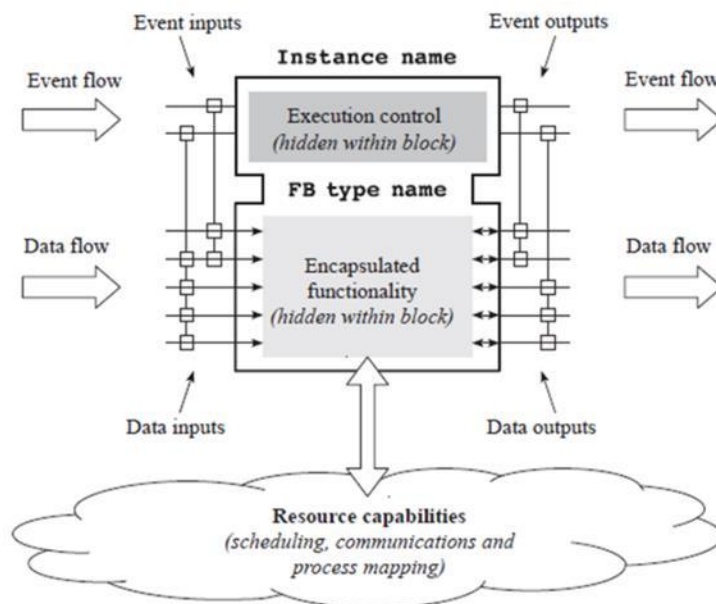


Figura 2. Modelo IEC-61499 [22]

## Bloques funcionales (FBs)

Los Bloques Funcionales denominados FB (nombrada así por sus siglas en inglés: Function Block) de la norma IEC 61499 unen dos tipos de parámetros, los eventos y los datos, en la Figura 3 se describe la estructura de un FB. La funcionalidad de un FB viene comandado por la recepción de eventos, simultáneamente también dependen de los datos de entrada, salida y de las condiciones internas del mismo. Un FB es capaz de tomar y enviar ambos tipos de parámetros, por así decirlo, la toma o envío de un evento se coordina con la actualización de los datos de entrada o salida asociadas con dicho evento. [23]



**Figura 3.** Estructura de un bloque funcional (FB)

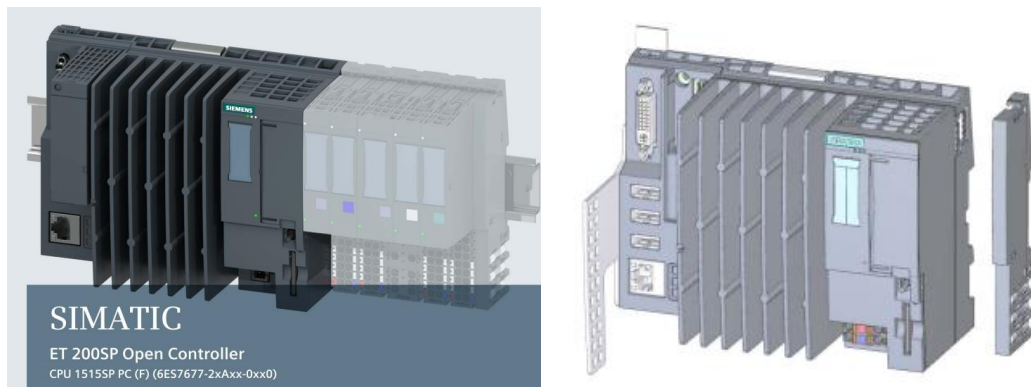
### Existen tres tipos de FB:

- Los BFB están definidos por poseer una estructura y procedimiento que están definidos por medio de un gráfico de control de ejecución.
- Los bloques de funciones compuestos CFB son simplemente una red internada conectada de BFB.
- Los bloques de funciones de interfaz de servicio SIFB. Son necesarios para acceder al hardware, la aplicación de estos tipos de FB necesariamente debe acceder a entradas y salidas, se activan no solamente por condiciones internas sino también por hardware. [23]

## Hardware

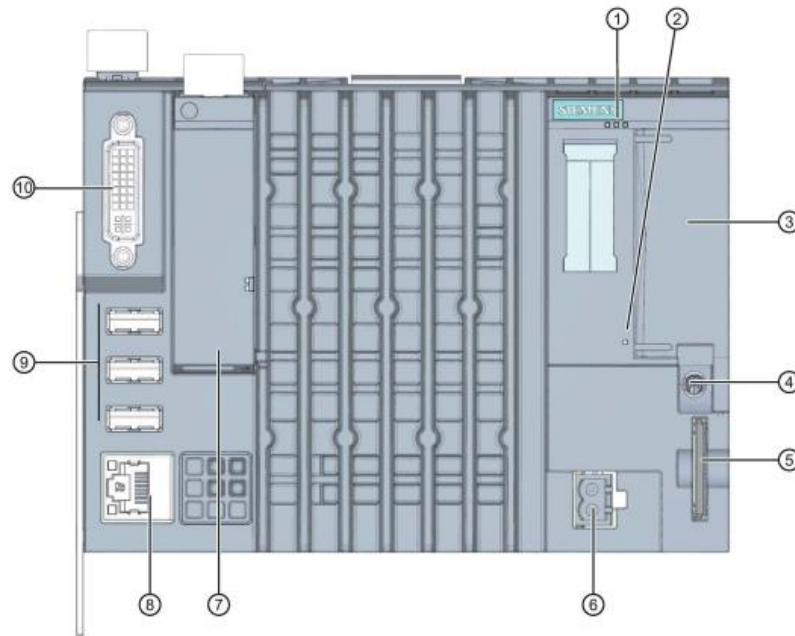
### ET 200SP Open Controller CPU 1515S PC

SIMATIC ET 200SP Open Controller está compuesto por un CPU 1515SP PC2 que tiene como soporte de memoria una tarjeta CFast (memoria Flash) que contiene el sistema operativo "Windows 10 IoT Enterprise" además de sus módulos de periferia. Cabe recalcar que el CPU 1515SP es un autómata programable basado en PC con el diseño ET 200SP capaz de realizar diferentes tareas de control y visualización. En la figura 4 se muestra el equipo físico ET 200SP Open Controller CPU 1515S PC. [24]



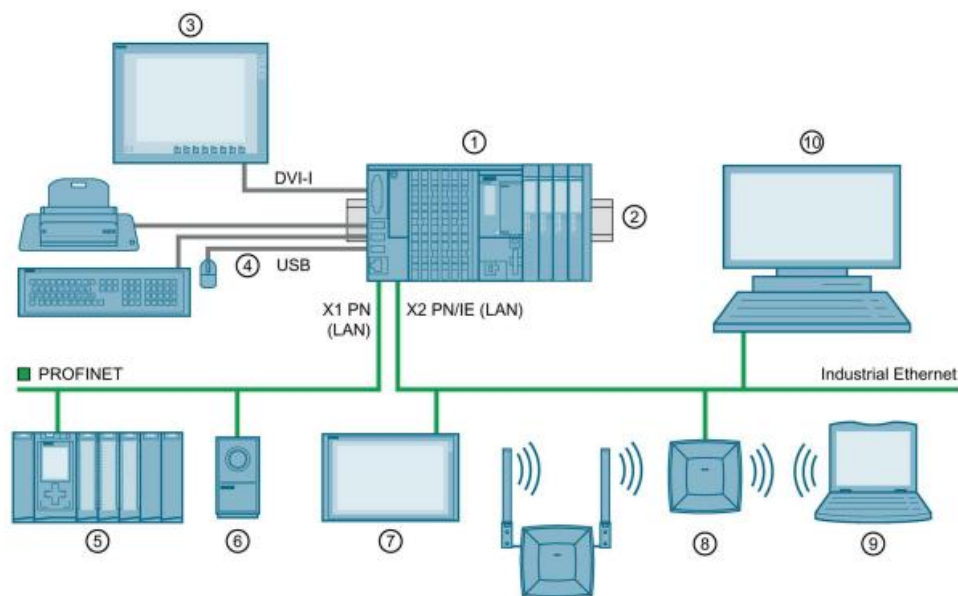
**Figura 4.** ET 200SP Open Controller CPU 1515S PC [24]

El equipo posee diferentes interfaces con las que se puede conectar con varios dispositivos como se muestra en la figura 5 posee: (1) Luz LED indicadora de estado de la CPU, (2) Luz LED indicadora de alimentación, (3) X50: Ranura para la tarjeta CFast (memoria Flash), (4) Selector de modo, (5) X51: Ranura para una tarjeta SD/MMC opcional, (6) X80: Tensión de alimentación 24V DC, (7) X1 PROFINET (LAN): Ranura para BusAdapter para la conexión de PROFINET IO, (8) X2 PN/IE (LAN): Puerto Ethernet, (9) X60, X61, X62: 3 puertos USB, (10) X70 DVI-I: Puerto para monitor. [24]



**Figura 5.** Vista frontal de CPU 1515SP [24]

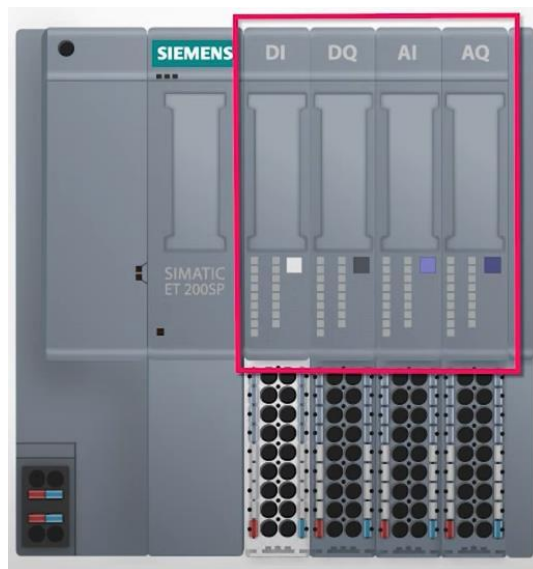
El dispositivo posee distintas interfaces que permiten la conexión con varios dispositivos de periferia, como se puede ver en la figura 6 a continuación se muestra (1) CPU 1515S PC con sus módulos digitales y analógicos, (2) Soporte, (3) Pantalla panorámica, (4) Dispositivos Periféricos USB: teclado, impresora, ratón , teclado, (5) Interfaz X1 PN (LAN) para dispositivo PROFINET IO, (6) Interfaz X1 PN (LAN) - cámara, (7) X2 PN/IE (LAN) - Industrial Thin Client ITC, (8) SCALANCE W786, (9) Portátil industrial Field PG y (10) Computador personal. [24]



**Figura 6.** Configuración de la CPU 1515sp [24]

## Módulos de entradas y salidas (E/S)

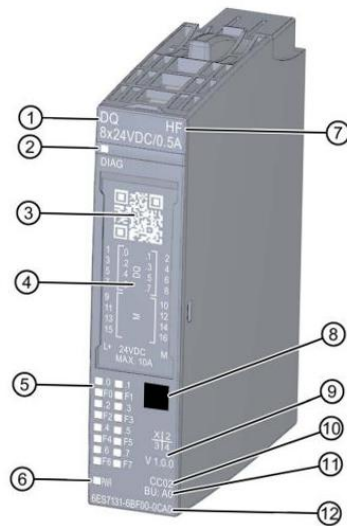
La unidad base ET 200SP de SIEMENS posee un módulo principal y módulos de E/S digitales y analógicas que al ser configuradas a través de un software de implementación permiten la conexión con sensores y actuadores de un proceso industrial, estos equipos terminales envían todas las señales a los módulos de E/S los cuales son enviados al control lógico programable para realizar las funciones según la programación establecida, en su hardware posee Leds indicadores de estado, esquema de conexiones y referencias. Cabe recalcar que la tensión de alimentación es de 24V proveniente del módulo principal. Como se observa en la figura 7 consta de cuatro tipos de módulos: un módulo de 8 entradas digitales (DI), un módulo de 8 salidas digitales (DQ), un módulo de 2 entradas analógicas (AI) y un módulo 2 de salidas digitales (AQ). [25]



**Figura 7.** Tarjetas de E/S en el ET 200SP [26]

En la figura 8 se muestra una vista del módulo de salidas analógica (DQ) con sus etiquetas respectivas: (1) tipo y designación del módulo, (2) luz indicadora de diagnóstico, (3) código de barras bidimensional, (4) esquema de conexiones, (5) luz LED indicadora de estado de canal, (6) luz LED indicadora de tensión de alimentación, (7) clase de función (HF-diseñado con funciones amplias), (8) Identificación por color del tipo de módulo, (9) versión funcional y de firmware, (10) código de color, (11) Tipo de BU, (12) Referencia. Cada módulo tiene sus características propias según su tipo. [27]





**Figura 8.** Vista del módulo DQ [28]

### **Sistema de producción modular (MPS)**

Un sistema de producción modular (MPS) es el conjunto de estaciones constituidas de una parte mecánica, electrónica, electrotécnica, neumática e incluso la implementación de tecnologías de la información, dicho sistema permite emular con hardware real lo procesos industriales de diferentes niveles de complejidad [29]

### **FESTO MPS 500**

Es una planta de entrenamiento que analiza piezas cilíndricas según su material (plástico, aluminio), color (rojo, gris, negro) y altura, el sistema está constituido por actuadores los cuales constan de accionamientos neumáticos y eléctricos, además de sensores de presencia, inductivos y capacitivos. [30]

El sistema de producción modular se divide en cinco estaciones: [29]

- Estación de distribución
- Estación de prueba
- Estación de procesamiento
- Estación de manipulación
- Estación de clasificación

## Software

Los diferentes softwares permiten la programación de datos y procedimientos según requiera el usuario a continuación se describen los paquetes de desarrollo que se utilizarán en el presente proyecto de investigación.

### 4DIAC

Una herramienta de software que permite el modelado y programación de bloques funcionales que cumple con IEC 61499 para aplicaciones de control distribuido, permitiendo simular el hardware de control y sus interconexiones a través de redes. 4diac es ejecutado por 4diac-forte el cual permite ser compilado en dispositivos sencillos de 16 y 32 bits, permitiendo la reconfiguración dinámica en línea, sin desconexión, y en tiempo real. En la figura 9 se muestra la interfaz de 4DIAC con bloques funcionales de ejemplo [23]

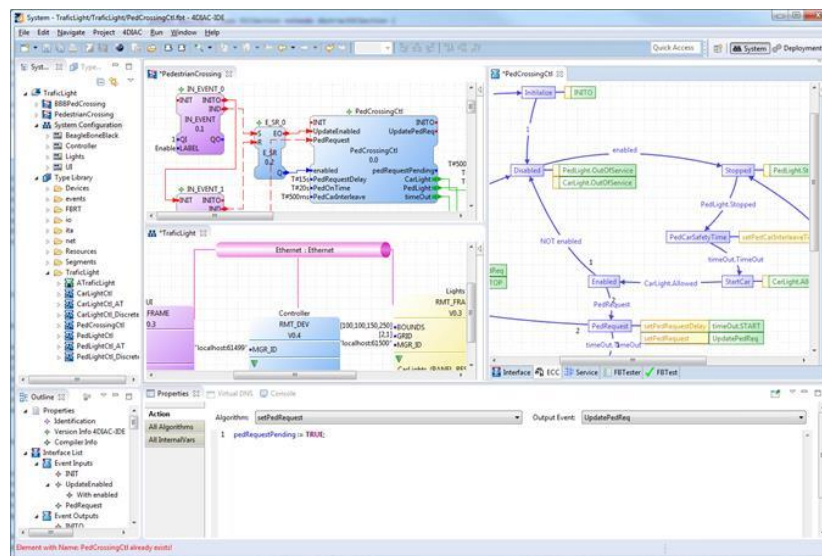
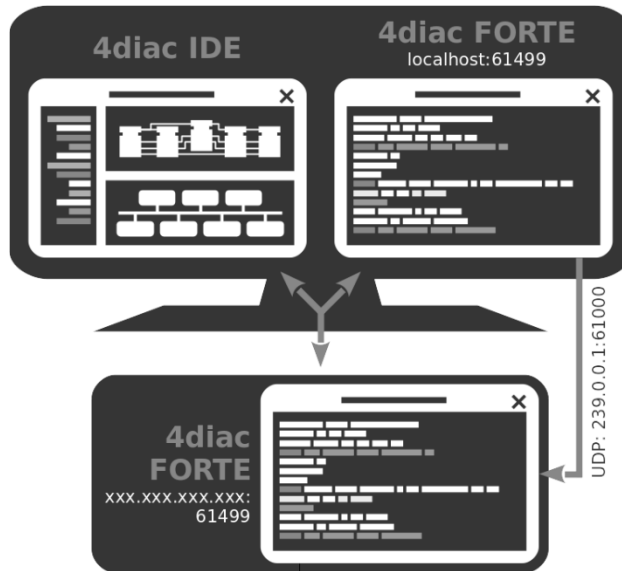


Figura 9. Interfaz 4DIAC [23]

### 4DIAC-FORTE

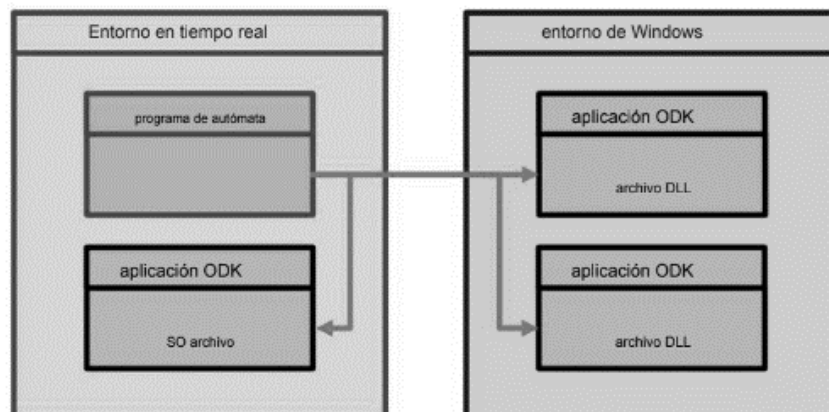
Es una plataforma de ejecución basada en la norma I IEC-61499 que es implementada en C++ en sistemas embebidos en tiempo real, fue desarrollada por 4DIAC para trabajar de forma independiente e decir que tiene la capacidad de ejecutarse en diferentes hardwares y sistemas operativos. 4diac FORTE proporciona una infraestructura de comunicación flexible a través de las llamadas capas de comunicación. En la figura 10 se muestra un esquema general de 4DIAC-FORTE. [23]



**Figura 10** Esquema general de 4DIAC-FORTE [23]

### ODK 1500S V2.5

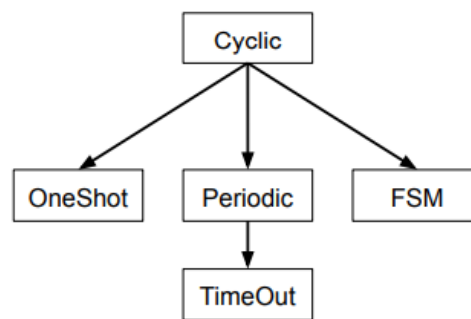
Es un paquete de desarrollo que permite programar aplicaciones en lenguaje de alto nivel y cargarlos en el controlador de planta, creando sus propias funciones y generando archivos que puedan ser llamados directamente desde TIA PORTAL hacia otros programas que requieran sus recursos. Posee interfaz de entorno para Windows posibilitando llamar varias aplicaciones a la vez en tiempo real. [31]. ODK 1500S V2.5 permite programar sus propias funciones y generar archivos como: el archivo DLL (Dynamic Link Library) que es programado en Microsoft Visual Studio para el entorno de Windows y el archivo SO (Share objects – Objetos compartidos) programado en Eclipse para ejecutarse en un entorno en tiempo real. En la figura 11 se muestra una estructura general de los archivos DLL y SO [31]



**Figura 11.** Entornos de ODK [31]

## SPADE

SPADE (Smart Python multi-agent Development Environment) es una plataforma para la creación de agentes que posee la capacidad de ser exportada como librería. Posee un canal de mensajería instantánea entre los agentes basado en XMPP, lo que permite transporte cifrado de información; además soporta HTTP. El código que se ejecuta por cada agente SPADE se estructura a través de comportamientos de manera que se implementa en varias partes de la lógica del agente por separado. Uno de los comportamientos más usados es Cyclic, es decir que trabaja de forma cíclica al poner en marcha todo el proceso iniciando así a cada agente y manteniéndolo activado en espera de información, es útil para realizar tareas repetitivas. En la figura 12 se visualiza los tipos de comportamientos que puede ejecutar. [32]



**Figura 12.** Comportamientos básicos de un agente SPADE. [32]

## Microsoft Visual Studio 2017

Es un editor de código fuente que permite el desarrollo de librerías de funciones el cual trabaja en diferentes multiplataformas una de sus mayores ventajas es que puede ejecutar cualquier tipo de lenguaje de programación como Python, C++, JavaScript entre otros, debido a que utiliza varias extensiones, además es de fácil instalación y se ejecuta en diferentes sistemas operativos como Windows y Linux, además es compatible con GitHub que permite manejar el código desde internet, en la figura 13 se puede observar el área de trabajo Visual Studio. [33]

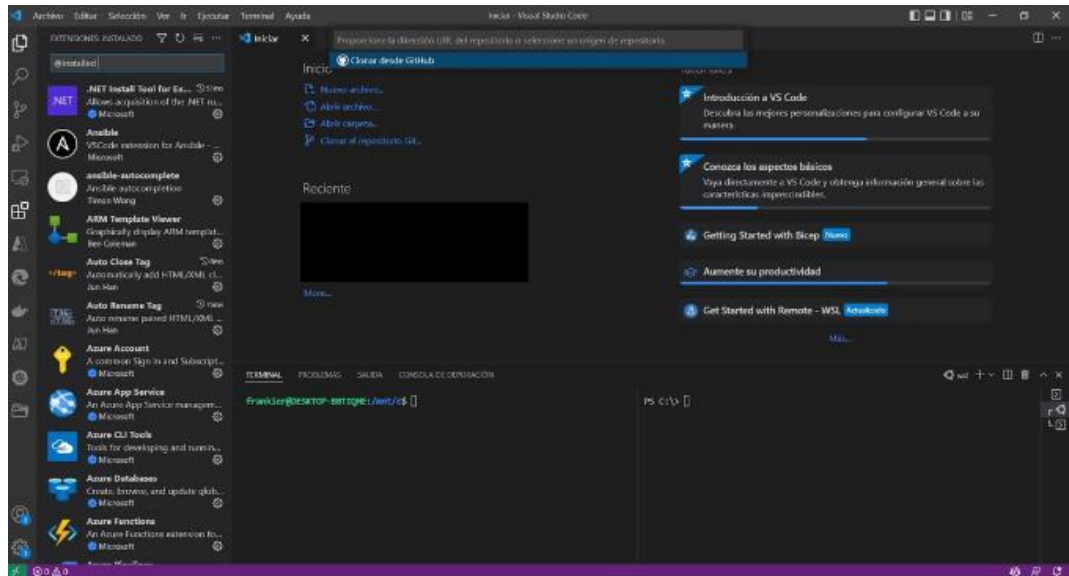


Figura 13 Área de trabajo de Visual Studio Code. [33]

## Openfire

Openfire es un servidor de mensajería instantánea que utiliza el protocolo XMPP para el envío y recepción de mensajes debido a que es sencillo de implementar, es de software libre y posee una interfaz amigable con el usuario, en este es posible monitorear los mensajes desde cualquier lugar. En la figura 14 se muestra un esquema básico de cómo se conecta a través del protocolo de mensajería instantánea. En su plataforma se puede crear varios usuarios según lo requiera el sistema. [33]

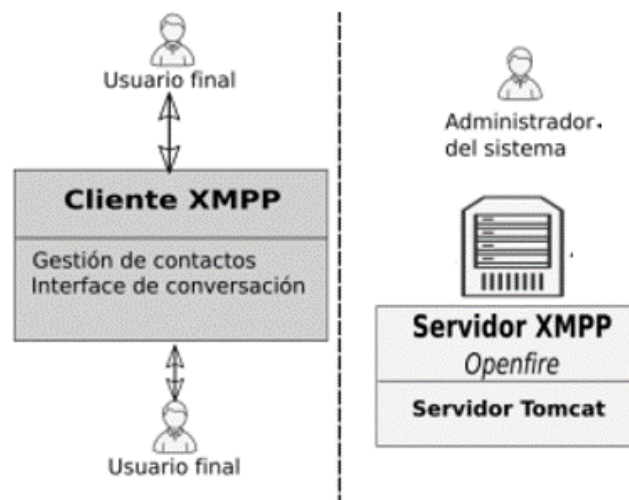


Figura 14 Interfaz de Openfire. [33]

## STEP 7 TIA PORTAL V16

Es un software que permite integrar todos los varios componentes para controlar procesos industriales, ofrece un entorno de ingeniería capaz de unificar las tareas de control, visualización y accionamiento a través del tipo de programación que se realice, estos pueden ser: diagrama de contactos, bloques de funcionamiento, texto, y lista de instrucciones, dichos programas son cargados en el control lógico programable. En la figura 15 se muestra el área de trabajo de TIA PORTAL V16

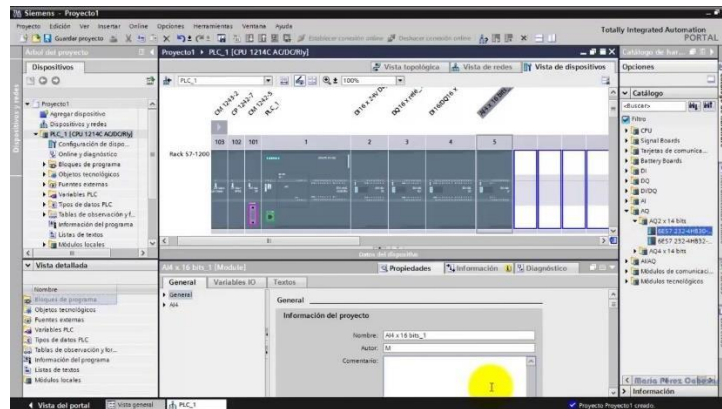


Figura 15. TIA PORTAL V16 [34]

## FACTORY IO IO 2.4.3

Factory IO es una plataforma 3D que permite diseñar escenas industriales utilizando una selección de piezas cumpliendo con su mecanismo de funcionamiento, además de controlar el proceso en tiempo real, consta de varios elementos como estaciones, operadores, carga ligera, sensores, actuadores, diversos tipos de pulsadores, motores, pistones, entre otros. Posee una lista de drivers que hace posible la conexión con diferentes tipos de controladores como PLC's e IPC's SIEMENS, realizando una configuración adecuada de las entradas y salidas tanto digitales como analógicas de dichos controladores, Factory IO realiza la lectura y escritura según la programación realizada. En la figura 16 se muestra varias escenas predefinidas en FACTORY IO [35]

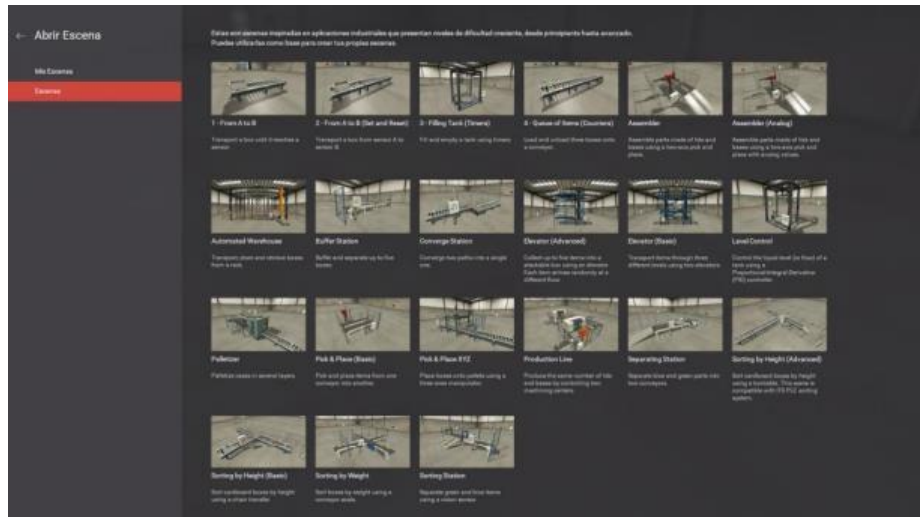


Figura 16 Escenas predeterminadas en Factory IO. [35]

### 1.3. Objetivos

#### 1.3.1. Objetivo General

Implementar una plataforma multiagente bajo la norma IEC-61499 para la reconfiguración dinámica de automatización de procesos en la estación FESTO MPS 500 que permita un proceso continuo ante fallas.

#### 1.3.2. Objetivos Específicos

- Desarrollar una arquitectura en base a agentes bajo la norma IEC-61499 para adquirir los datos enviados desde la estación FESTO MPS 500 hacia el agente de control.
- Implementar un sistema multiagente basado en bloques funcionales (FB) según la norma IEC-61499 para controlar el funcionamiento de la estación FESTO MPS 500 en el caso de que se generen errores en el proceso.
- Comprobar el tiempo de reacción de la reconfiguración dinámica aplicada a través de la medición de tiempos de recuperación para que se garantice el proceso de producción continuo.

## **CAPITULO II.- METODOLOGÍA**

### **2.1. Materiales**

Para el desarrollo del presente trabajo de investigación se utiliza diferentes fuentes bibliográficas como artículos científicos, libros y repositorios de universidades para recolectar información, analizar los procedimientos realizados y sus resultados aplicados tanto de la Norma IEC-61499 como del sistema multiagente. Además, para la implementación del sistema se requiere de distintos dispositivos y herramientas tanto para el hardware como para el software. En el hardware se utiliza dos computadores industriales IPC's SIMATIC ET 200SP Open Controller CPU 1515SP PC2, DAC ATMEGA328 Shield Ethernet (decodificar las señales virtuales en señales físicas), cables de conexión Ethernet, computadoras y periféricos de entradas y de salidas mientras que para el desarrollo del software se utiliza el sistema operativo Windows 10 IoT Enterprise, paquete de desarrollo ODK 1500S V2.5, 4DIAC-FORTE basado en el estándar IEC-61499, editor de texto Microsoft Visual Studio 2017, lenguaje de Python 3.8.10\_x64, servidor de mensajería instantánea basado en protocolo XMPP Openfire y el software de simulación FACTORY IO 2.4.3. Cabe mencionar que se utiliza los manuales de SIEMENS de SIMATIC ET 200SP Open Controller. CPU 1515SP PC2 y ODK 1500S V2.5.

### **2.2. Métodos**

#### **2.2.1. Modalidad de la Investigación**

##### **Investigación Aplicada**

La presente investigación fue de carácter aplicativo debido a que se empleó los conocimientos adquiridos durante la carrera de Ingeniería en Electrónica y Comunicaciones en el área de sistemas industriales con el objetivo de implementar de manera práctica un sistema multiagente para la reconfiguración dinámica.



## **Investigación Bibliográfica**

Uno de los enfoques empleados fue la investigación bibliográfica ya que se utilizó varias fuentes como artículos científicos indexados, libros y repositorios de las diferentes universidades para recolectar información y analizar los procedimientos y resultados aplicados en los diferentes estudios que permitieron afianzar las bases de la presente investigación, logrando la autocalibración del sistema multiagente en base a las investigaciones anteriores.

## **Investigación Campo**

El desarrollo del proyecto se realizó en un entorno real enfrentándose a obstáculos como perturbaciones en el ambiente no controladas, por lo cual un algoritmo multiagente debe ser completamente funcional bajo estas circunstancias respondiendo de manera eficaz ante los cambios realizados.

## **Investigación Experimental**

Se empleó un enfoque de prueba y error en el sistema, cuando se configuró el algoritmo del sistema multiagente se controló las variables utilizadas, incrementando o disminuyendo sus magnitudes con el fin de conocer su comportamiento en la reconfiguración dinámica y así disminuir el tiempo muerto de dicho sistema en el caso de que se genere algún tipo de fallo.

### **2.2.2. Recolección de Información**

La recolección de información del presente trabajo de titulación se llevó a cabo mediante artículos académicos, libros, proyectos de investigación previos que se obtuvieron en las diversas bases de datos de las universidades.

### **2.2.3. Procesamiento y Análisis de Datos**

Se recopiló la información requerida para la investigación, desarrollando los siguientes pasos:

- Análisis de la información recolectada mediante la metodología PRISMA.
- Delimitación de temas de investigación como: Sistema multiagente, estándar IEC-61499, industria 4.0, entre otros.
- Desarrollo del sistema encargado de controlar los datos del proceso en ejecución.

- Análisis de los dispositivos hardware a utilizarse en la implementación del algoritmo multiagente.
- Identificación de software utilizado para la implementación del sistema multiagente.
- Definir la información de gran importancia que aporte a la elaboración de la propuesta de solución.

#### **2.2.4. Desarrollo del proyecto**

Para cumplir con los objetivos planteados en el desarrollo del proyecto se llevó a cabo los siguientes pasos:

- Análisis del estado del arte de sistemas multigentes y el estándar IEC-61499.
- Diseño de la arquitectura hardware para el control de la estación de clasificación FESTO MPS 500.
- Diseño de bloques funcionales (FB) bajo la norma IEC-61499 para adquirir datos de sensores y recepción de comandos en los actuadores.
- Análisis de los algoritmos de programación para la implementación en los sistemas multiagentes.
- Programación de los multiagentes bajo el enfoque del estándar Open Process Automation Standard (OPAS)
- Compilación y ejecución del sistema multiagente dentro del runtime existente en la norma IEC-61499.
- Comprobación del sistema de monitorización bajo la norma IEC-61499 a través de pruebas de validación.
- Comprobación del sistema multiagente bajo la norma IEC-61499 a través de pruebas de validación.

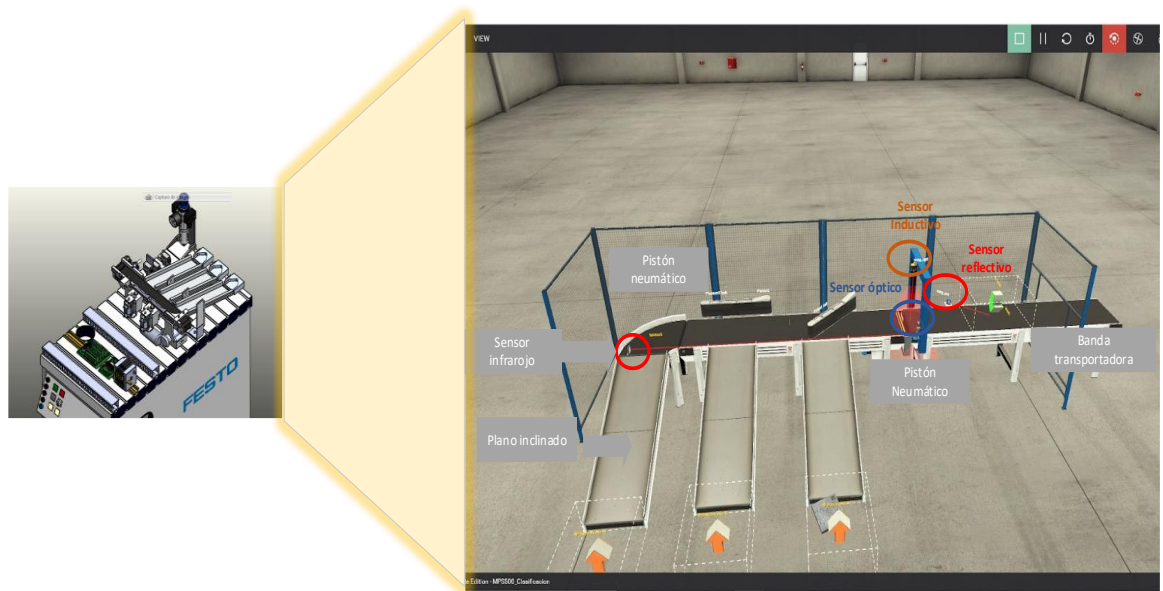
## **CAPÍTULO III.- RESULTADOS Y DISCUSIÓN**

### **3.1. Análisis y discusión de los resultados**

En la actualidad el requerimiento de la automatización de los diferentes procesos industriales se ha visto en la necesidad de la aplicación de sistemas de control distribuido para una mejor respuesta ante posibles fallas en el sistema con el fin de obtener una mayor eficiencia en la producción o servicio que brinde el proceso, siendo esta la razón del porqué de este proyecto. Se realizó un sistema multiagente bajo la norma IEC-61499, la cual permite la reconfiguración dinámica de un proceso de clasificación ante posibles fallas por desconexión utilizando dos equipos IPC's ET 200SP Open Controller CPU 1515SP PC2.

El proyecto sería implementado en la estación de clasificación FESTO MPS 500 de los laboratorios de la Facultad de Sistemas Electrónica y Comunicación de la Universidad Técnica de Ambato, debido a que la estación se encontraba en mantenimiento y/o ocupada, mientras se desarrollaba la lógica del proyecto, no fue posible el uso de la estación; para avanzar de una mejor manera se optó por un entorno de simulación 3D mediante el software Factory IO el cual nos permite realizar y controlar procesos industriales en tiempo real, en este caso se simuló una estación con similares características a la estación de clasificación FESTO MPS 500.

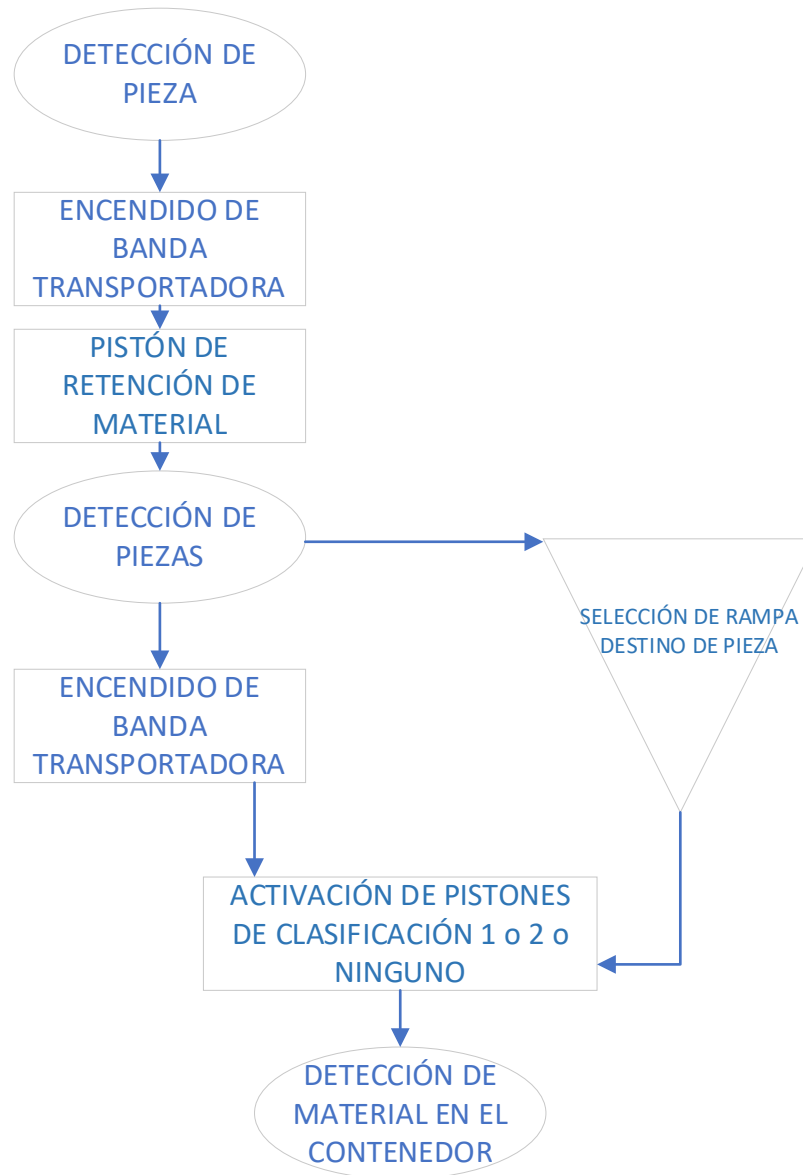
La estación de clasificación Festo 500 consta de la detección de materiales con sensores clasificándolos en tres planos inclinados de material, posee un sensor de reflexión directa el cual detecta la pieza y activa una banda transportadora hacia el punto de clasificación, un pistón neumático detiene la pieza sobre la banda transportadora, los sensores ópticos y los sensores inductivos detectan las propiedades de color y material de las piezas y finalmente un pistón es activado para clasificar las piezas en el plano inclinado correspondiente, la implantación realizada en el software Factory.io tiene exactamente las mismas características antes descritas, además se adiciona un sensor infra rojo para la detección del depósito final de las piezas, como se puede observar en la figura 17.



**Figura 17.** Simulación en Factory IO de la estación de clasificación Festo

**Elaborado por:** Investigadores

Un diagrama funcional del módulo simulado se diseña en la figura 18, los elementos que intervienen en la monitorización del proceso son tres sensores, el primer sensor es el encargado de detectar la presencia de material, el segundo sensor óptico permite reconocer que tipo de material es y el tercer sensor reflectivo confirma el depósito de las piezas. Así mismo contiene una serie de actuadores que permite la movilización de las piezas, al inicio del proceso una banda se encarga de llevar las piezas hacia un pistón de retención que detiene las piezas para que sean reconocidas, enseguida la banda transportadora desplaza las piezas hasta los pistones de clasificación, existen dos pistones que se activan de manera individual dependiendo del tipo de pieza, al final del proceso los pistones desvían las piezas hacia una de las tres rampas que dan al depósito final.

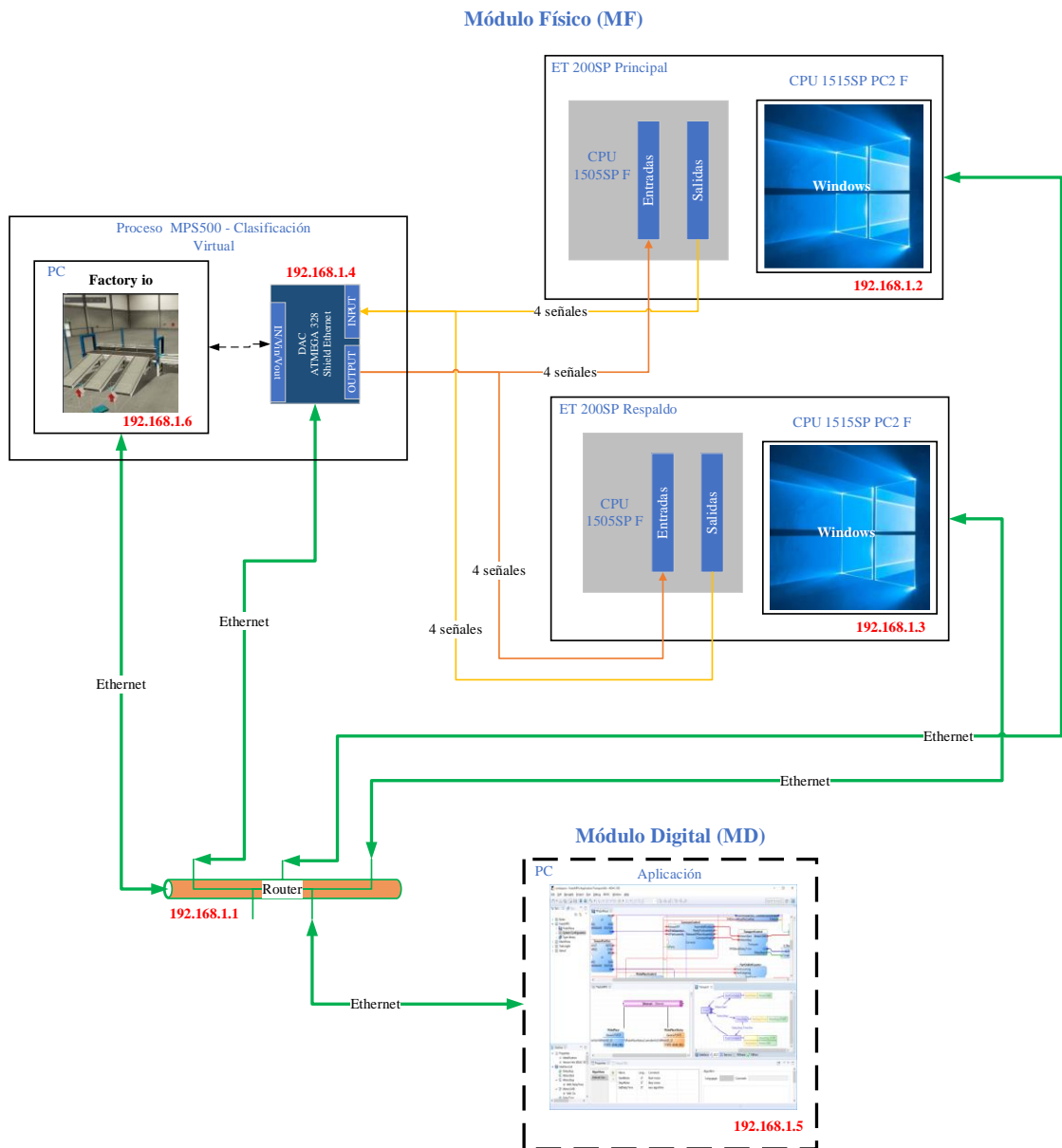


**Figura 18.** Diagrama funcional del módulo simulado

**Elaborado por:** Investigadores

El proceso de clasificación antes descrito se controla mediante el sistema programado bajo la norma IEC 61499, que es básicamente un control condicional que activa diferentes actuadores dependiendo de los valores de entrada de los sensores, además se realiza una inspección de todas las lecturas de los sensores y estados de los actuadores para almacenar estos datos en tiempo real, para así, en el caso de existir alguna falla por desconexión la información almacenada sea dirigida hacia el IPC de respaldo.

La distribución del sistema de control se conforma por dos módulos uno Físico (MF) y uno Digital (MD) como se observa en la figura 19.



**Figura 19.** Esquema de Proyecto MF y MD

**Elaborado por:** Investigadores

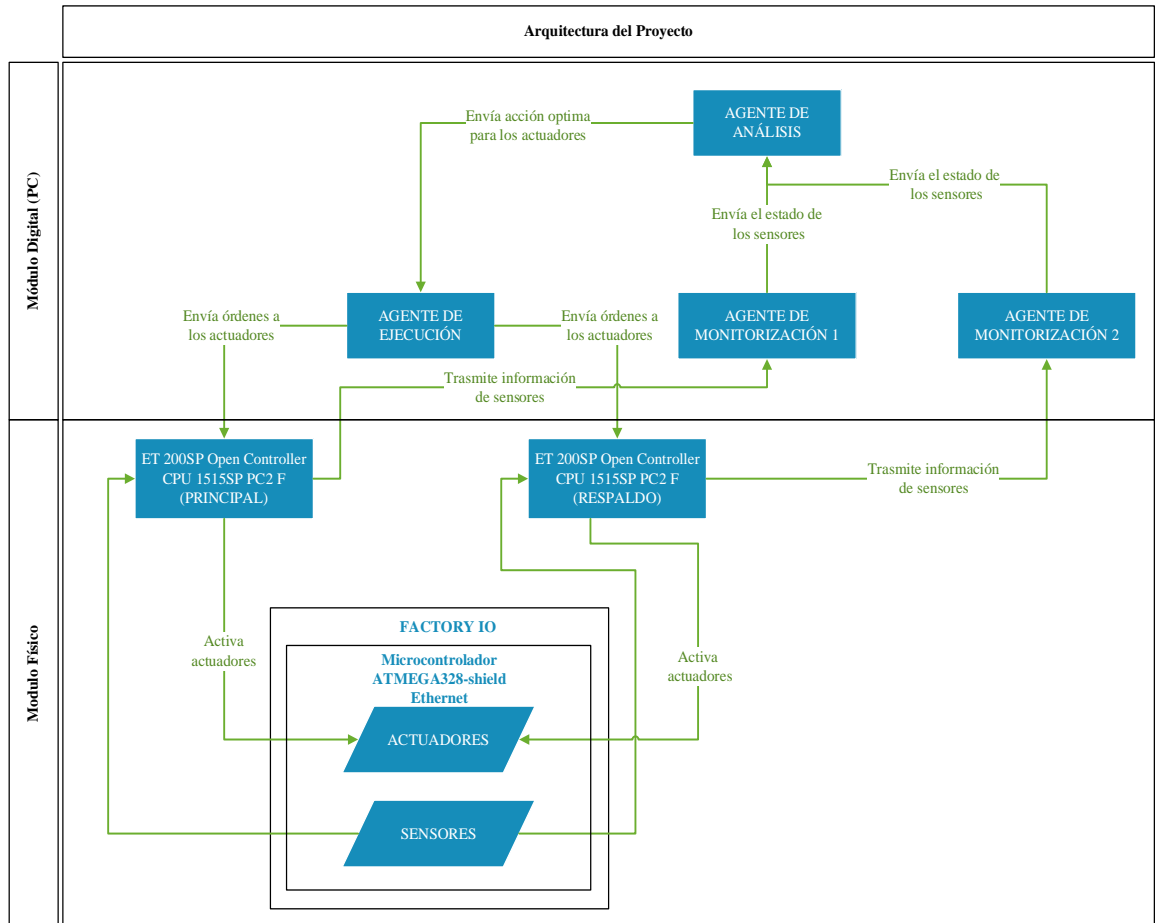
El Módulo Físico (MF) está compuesto por los sensores y actuadores de la estación de clasificación simulada en un ordenador que se comunica con el microcontrolador ATMEGA328-shield Ethernet (trabaja de intermediario para decodificar las señales virtuales en señales físicas permitiendo la conexión hacia los dos equipos de control), el cual está conectado en paralelo a dos IPC's ET 200SP Open Controller CPU 1515SP PC2 que no funcionan de manera simultánea ya que un equipo trabaja solamente como

respaldo del equipo principal, cada IPC posee un sistema operativo en conjunto con un Control Lógico Programable que funcionan individualmente, para lograr la integración de ambas partes se utiliza la interfaz ODK 1500 v2.5 que permite enviar datos desde las entradas del IPC - CPU 1515SP hacia una DLL de Windows que se comunica con el sistema operativo del mismo equipo, utilizando espacios de memoria, y así poder procesar los datos de los sensores y el control de actuadores utilizando librerías y programación en lenguaje de alto nivel bajo la norma IEC 61499, una vez llegado a este punto el módulo Físico (MF) se comunica con el Módulo Digital (MD) a través de los bloques funcionales de la misma norma.

El Módulo Digital contiene el control condicional de todo el sistema, está formado por tres tipos de agentes:

- 1. Agente de Monitorización (AM):** Monitorea en tiempo real el estado de los sensores mediante los datos recibidos del Módulo Físico y envía los datos al Agente de análisis. Existen dos Agentes de Monitorización, uno perteneciente al equipo principal y otro al equipo de respaldo.
- 2. Agente de Análisis (AA):** Capta los datos enviados desde el Agente de Monitorización, luego interpreta la información y la procesa, el Agente de Análisis es el que abarca toda la programación del comportamiento del sistema enviando ordenes de ejecución que deben seguir los actuadores al Agente de Ejecución de acuerdo a los datos leídos por los sensores. De existir alguna falla por parte del IPC principal el Agente de Análisis de forma inmediata retoma el proceso utilizando el IPC de respaldo.
- 3. Agente de Ejecución (AE):** Al recibir las ordenes de ejecución del Agente de Ejecución las envía al Módulo Físico en funcionamiento.

En la figura 20 se presenta la arquitectura del proyecto y el funcionamiento de cada elemento que la compone.



**Figura 20.** Arquitectura proyecto multiagente

**Elaborado por:** Investigadores

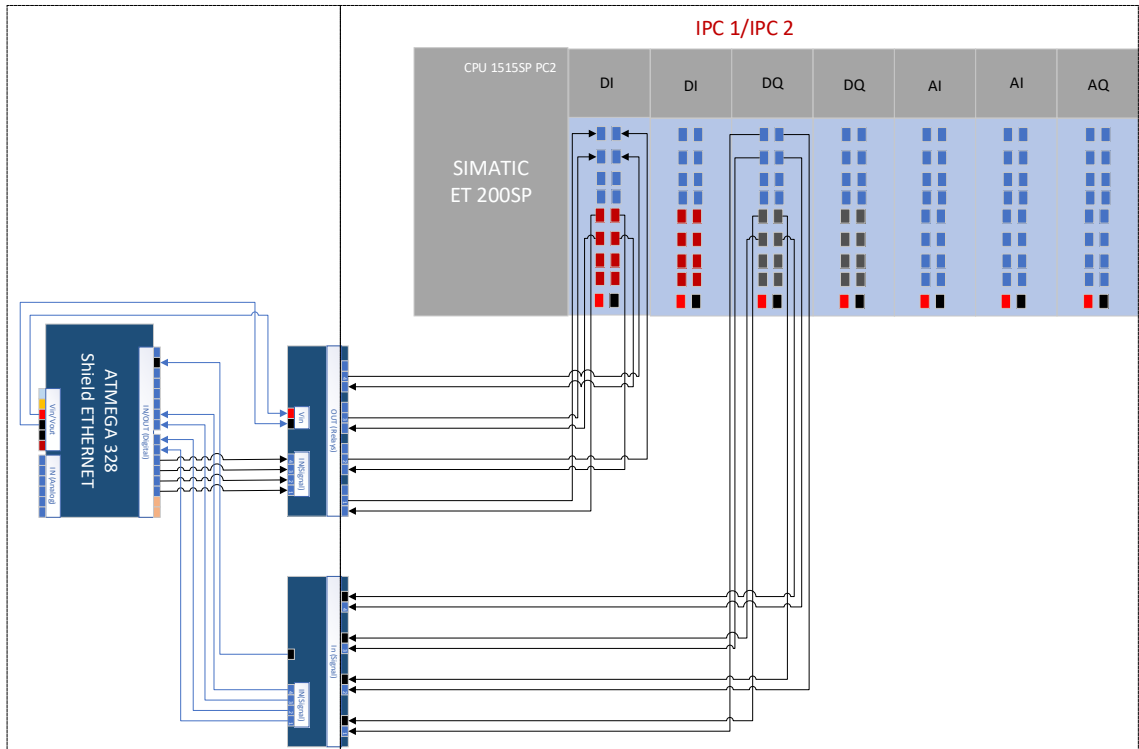
### Desarrollo de la propuesta

Para la ejecución del proyecto se necesita una visión clara de las conexiones y protocolos a utilizar el cual permite la transmisión y flujo de datos generados.

Debido a que la estación FESTO500 no se encontraba disponible (por motivos varios), se utiliza el software Factory IO el cual permite el desarrollo de un entorno similar o idéntico a la Estación de clasificación Festo500, comenzando por la conexión virtual de los sensores y actuadores hacia un módulo de entrada/salida DAC ATMEGA328 Shield Ethernet (trabaja de intermediario para decodificar las señales virtuales en señales físicas) permitiendo la conexión de las señales de manera paralela hacia cada IPC S7-1500 los pines del microcontrolador designados como salidas de los sensores son conectados directamente a las entradas de los módulos ET 200SP, mientras que las salidas de los módulos del ET 200SP son conectados a un divisor de tensión, para



luego enlazar a los pines de entrada del microcontrolador que trabajan como estados de los actuadores, ver figura 21.



**Figura 21.** Esquema de conexión física: controlador ATMEGA328 con ET 200SP principal y respaldo

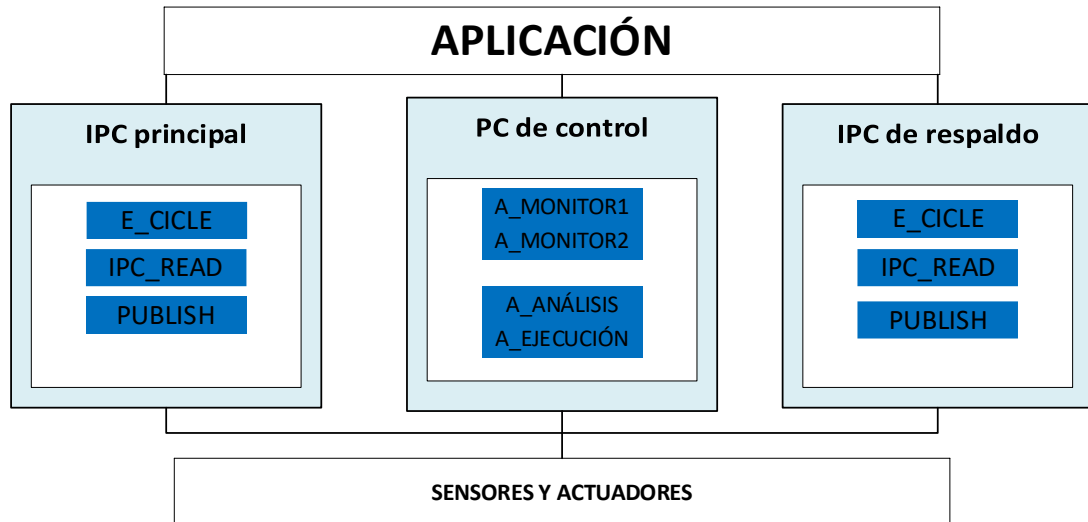
**Elaborado por:** Investigadores

La distribución del sistema de control se conforma por dos módulos uno Físico (MF) y uno Digital (MD). El módulo Físico MF

### 3.1.1 Módulo Físico

Está conformado por los sensores y actuadores de la estación de clasificación simulada que se encuentran conectados de manera paralela a dos IPC Siemens 1515SP a su vez los IPC se conectan de manera remota a la PC de control. La lógica de la estructura de aplicación que admite la comunicación de datos a través de estos dispositivos está basada en bloques funciones (FB por sus siglas en inglés) creados en 4DIAC y ejecutados en un entorno de ejecución 4DIAC-FORTE de acuerdo al modelo de sistema que se muestra en la figura 22. El controlador S7-1500 del ET 200SP a su vez se conectan por medio de una DLL (Dynamic Link Library por sus siglas en inglés) a cada uno del sistema operativo Windows con la ayuda de la herramienta ODK 1500S por medio de Bloques Funcionales creados que leen y escriben directamente la

información de las entradas y salidas del S7-1500 del ET 200SP. Para enlazar los datos entre el sistema operativo Windows con Forte, se logró por medio de memorias compartidas (Archivos DLL, bibliotecas de enlace dinámico), ver Figura 23.



**Figura 22.** Modelo del sistema 1

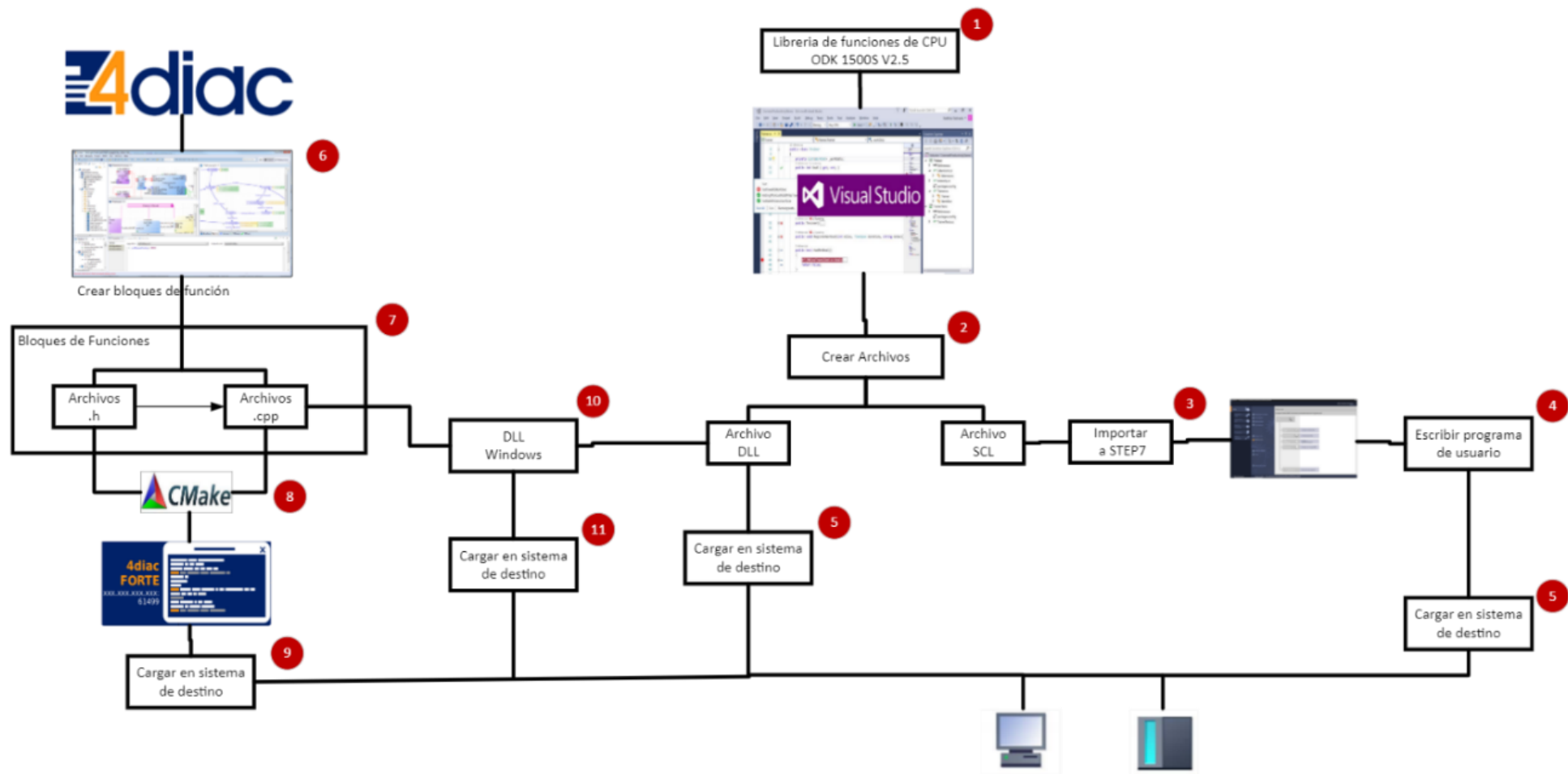
**Elaborado por:** Investigadores

#### **4DIAC en el Módulo Físico**

Se utiliza el sistema 4DIAC debido a que encapsula los datos y código de forma independiente a través de bloques funcionales (FBs), dichos bloques se conectan entre sí formando una sola aplicación brindando un control en base al estándar IEC 61499, para que 4DIAC pueda ejecutarse correctamente hace uso de Forte una plataforma de ejecución implementada en C++. Los bloques de función permiten el manejo de las entradas y salidas tanto digitales como analógicas desde el IPC - CPU 1515SP, tomando en cuenta que los IPC's utilizados están conformados por el CPU 1515SP PC2 y los módulos de periferia se necesita una transmisión bidireccional de los datos a través de memorias compartidas, para lograr esta comunicación se utiliza el paquete de desarrollo ODK 1500S V2.5 el cual permite programar sus propias funciones y generar archivos DLL que se comunica con el sistema operativo Windows del IPC S7-1500, es decir que, la implementación de dichos archivos permiten la comunicación entre todo el sistema y son cargados en la ubicación de destino correspondiente.

En la figura 23 se muestra un esquema general de la conexión entre el ODK 1500 y 4DIAC-FORTE: (1) Implementación de la función para las librerías de funciones de

CPU en Visual Studio 2017 (creación de los bloques de funciones para DLL y SCL y creación de bloques de funciones para comunicar el DLL con FORTE), (2) generación del archivo DLL y el archivo SCL, (3) Importación de archivo SCL a STEP 7-TIA PORTAL V16, (4) Programación en TIA PORTAL V16, (5) Carga del programa de usuario en la CPU 1505s y el archivo DLL en el sistema de destino, (6) creación de los bloques funcionales en 4DIAC, (7) Generación de los archivos .h y .cpp (8) empaqueta los archivos para crear un ejecutable en FORTE, (9) ubicación de los archivos de FORTE en el sistema de destino.

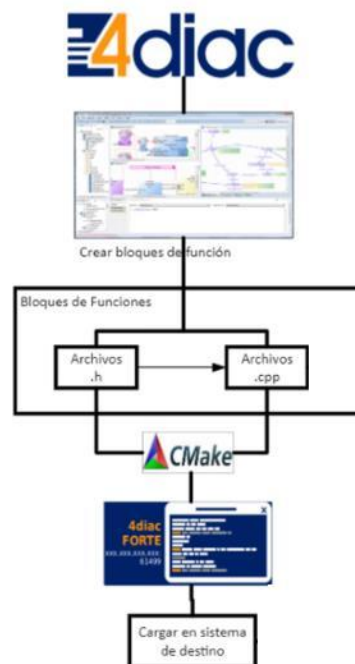


**Figura 23.** Esquema general de conexión entre 4DIAC y ODK 1500s v2.5

**Elaborado por:** Investigadores

## Archivos generados en 4DIAC FORTE

La primera rama del esquema general, ver figura 24, se realiza mediante la implementación de los bloques funcionales FBs configurando los eventos de entrada y salida así como sus datos, una vez culminado se exportan generando dos tipos de archivos .h que corresponde al encabezado y .cpp al programa de usuario, es necesario empaquetar los archivos para que puedan trabajar con 4DIAC-FORTE que es multiplataforma permitiendo ser ejecutado en diferentes hardwares y sistemas operativos, dicho empaquetamiento se genera desde un código fuente dando lugar a varios archivos específicos y finalmente cargando en el sistema de destino.



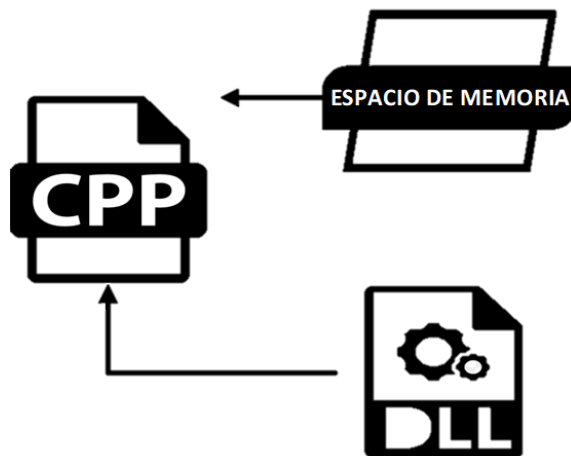
**Figura 24.** Estructura general de 4DIAC-FORTE

**Elaborado por:** Investigadores

El diseño de los bloques funcionales se realiza de forma individual por cada tipo de entrada y salida, dichos FBs contienen la programación necesaria para el funcionamiento de la aplicación y obtención de la información desde los IPCs. Los FBs creados encapsulan algoritmos mediante el desarrollo de código de usuario y librerías dentro de su codificación en C++, 4DIAC al ser una aplicación robusta permite la implementación de varios tipos de bloques funcionales FBs, para este caso de estudio se utilizan los bloques de interfaz de servicio (SIFB) debido a que gestiona

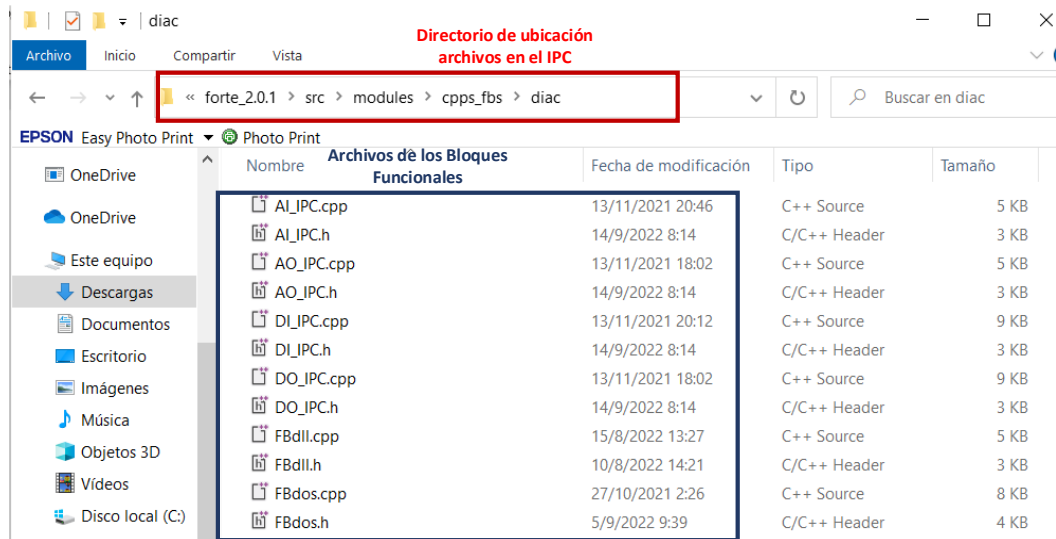
la comunicación entre bloques funcionales y el mundo exterior, accede a partes específicas del hardware (entradas o salidas) a diferencia de los bloques funcionales básicos BFB y los bloques funcionales compuestos CFB que no lo pueden hacer, estos bloques se activan a través de un evento o también por el hardware.

Los bloques funcionales FBs deben ser fáciles de manipular es por esto que se diseña un bloque funcional por cada tipo de entrada y salida (entrada digital DI, entrada analógica AI, salida digital DO y salida analógica AO), además de tener la misma estructura. Para que los FBs puedan ejecutarse correctamente desde el 4DIAC se utiliza la plataforma FORTE, una vez exportado cada bloque funcional se genera el archivo .h que corresponde al archivo encabezado, del cual no se modifica su código base permitiendo el funcionamiento directo sin la necesidad de volver a compilar el FORTE; también, se genera el archivo .cpp que corresponde al programa de usuario, en donde se direcciona las memorias y el DLL de Windows, ver figura 25, cabe recalcar que existen DLL's propios del Sistema Operativo Windows denominados smClient64.dll el cuál se encuentra en el directorio de FORTE. Es importante direccionar al directorio “\forte\_2.0.1\src\modules\cpps\_fbs” los archivos .h y .cpp de cada bloque funcional, ver figura 26. Estos códigos se utilizan en los dos IPC's.



**Figura 25.** Direccionamiento hacia DLL de Windows en archivo .cpp.

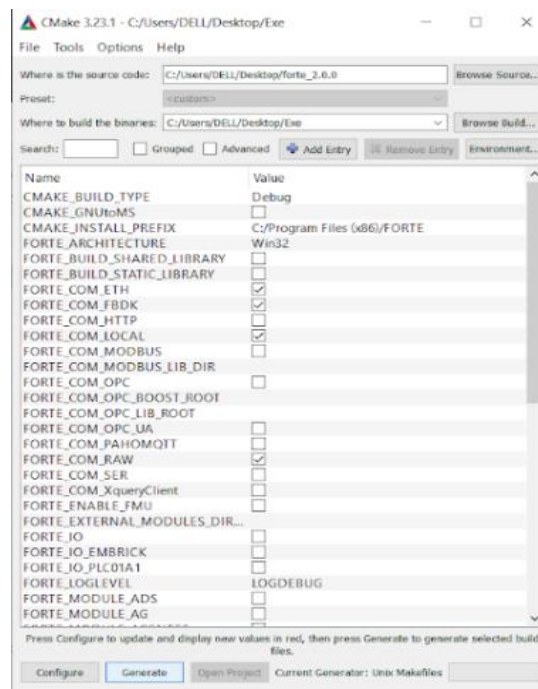
**Elaborado por:** Investigadores.



**Figura 26.** Sistema de destino de los archivos .h y .cpp de cada bloque funcional

**Elaborado por:** Investigadores

Es necesario añadir un directorio específico “Cmakelist” (ubicado en el directorio Forte 2.0.1\scr\modules\cpps\_fbs) para llamar a los bloques funcionales exportados mediante un bloc de notas .txt. Finalmente, todos los archivos son empaquetados a través del software Cmake que permite contener el Runtime Forte como se observa en la figura 27.



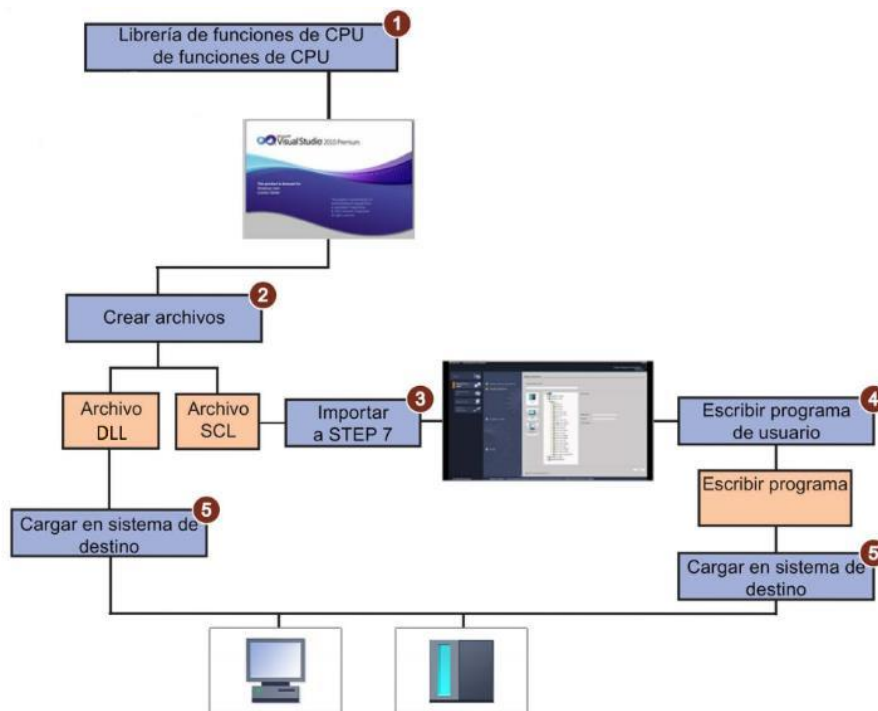
**Figura 27.** Empaquetamiento de archivos a través de CMAKE

**Elaborado por:** Investigadores

Después del proceso de empaquetamiento, comienza un proceso de construcción de un archivo ejecutable Forte en la pantalla CMD de modo que se ejecuta el runtime y abre la conexión TCP-Sever estableciendo una conexión con el cliente, a continuación se envía los datos de las entradas y salidas del sistema manejadas a través de los bloques funcionales los cuales se comunican con el entorno de Runtime FORTE. Cuando los bloques funcionales se detienen en el IDE de 4Diac la conexión se cierra en los runtimes de FORTE de los IPCs.

### Archivos generados ODK 1500S V2.5

La rama que comprende la herramienta ODK 1500S V2.5 del esquema general, funciona como una interfaz para un entorno Windows (DLL) es decir que las funciones de Windows están disponibles en la aplicación ODK 1500, mientras que, si se trabaja en un entorno de tiempo real (SO) dichas funciones no se pueden utilizar. En la figura 28 se muestra la arquitectura correspondiente a los pasos desarrollados en ODK 1500 con los archivos que son generados en Microsoft Visual Studio, una vez compilado el programa se genera un archivo DLL y SCL que serán cargados en el sistema destino.

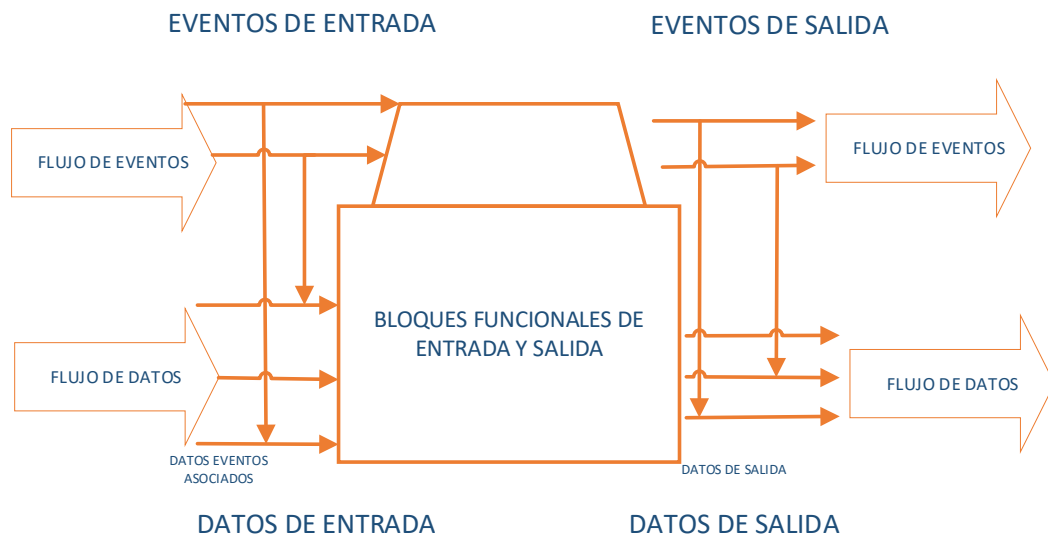


**Figura 28.** Arquitectura ODK

**Elaborado por:** Investigadores



Se utiliza Visual Studio 2017 ya que este software incluye plantillas de proyectos que facilitan el desarrollo de las librerías de funciones de CPU, para crear una librería se genera el ProyectoDIAC.odk en donde se habilita todas las direcciones tanto entradas como salidas, además existe un archivo.cpp en donde se direcciona el DLL de Windows (smClient64.dll), cabe recalcar que los archivos DLL del ODK 1500 y de FORTE apuntan hacia este directorio con el objetivo de que exista un flujo de datos bidireccional y que todo el sistema esté conectado entre sí, además se crean funciones del ODK 1500 correspondientes a los módulos de las entradas y salidas tanto digitales como analógicas para generar comunicación con el controlador lógico programable del IPC - CPU 1515SP y 4DIAC-FORTE, en la figura 29 se muestra un ejemplo de la configuración de los de los módulos de la entrada digital 1 y salida digital 1.

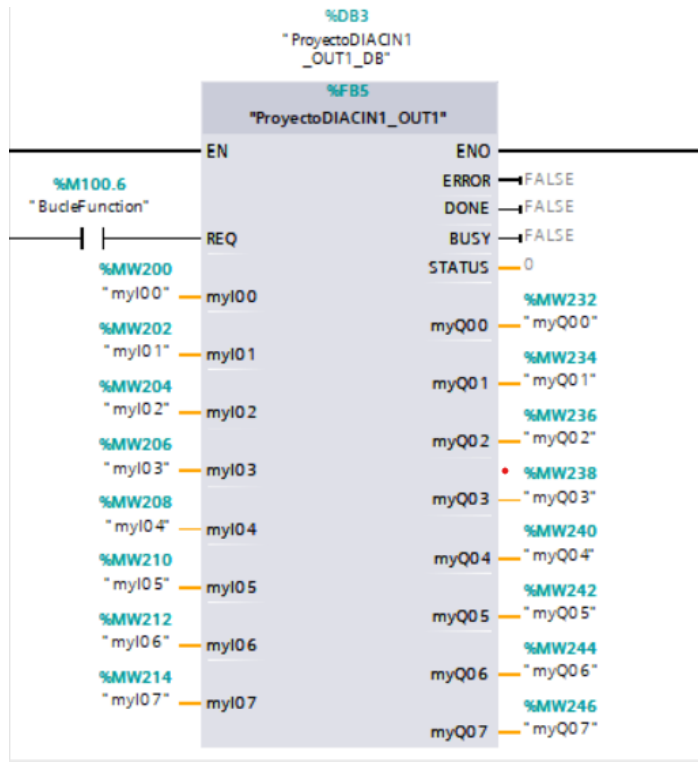


**Figura 29.** Funciones del ODK del módulo de salida digital y analógica

**Elaborado por:** Investigadores

Una vez compilado el programa se genera un archivo DLL que será cargado en un directorio específico del IPC, no reemplazable, ubicado en el disco local C (C:\Program Files\Siemens\Automation\ODK1500S).

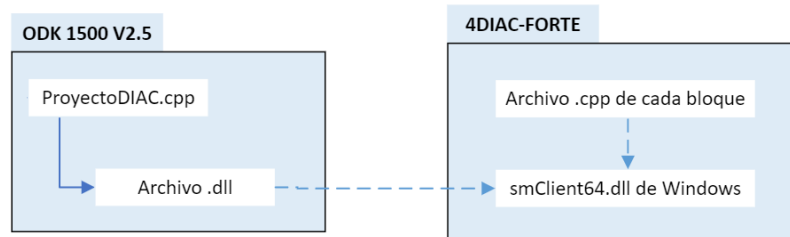
Se genera el archivo SCL el cual contiene bloques estructurados permitiendo el flujo de los datos, previo a esto se agrega el dispositivo (SIMATIC S7 Open Controller, versión de CPU 1515SP) con su respectiva dirección IP y se procede a importar el SCL en TÍA PORTAL. Una vez agregado el archivo se genera los bloques a partir de fuente como se muestra en la figura 30.



**Figura 30.** Bloques funcionales generados por ODK a través del archivo SCL

**Elaborado por:** Investigadores

Al tener configurado la lógica programable de 4DIAC-FORTE y ODK 1500S se establece la conexión entre estos softwares a través de los archivos DLL, como se observa en la figura 31 el paquete de desarrollo ODK 1500S V2.5 genera un archivo SimpleSort\_CPP.dll que apunta al archivo propio de Windows “smClient64.dll”, a su vez los programas .cpp de los bloques funcionales ejecutados en 4DIAC-FORTE también son direccionados al archivo mencionado mediante memorias compartidas. Cabe recalcar que los archivos de Windows se ejecutan dentro del FORTE permitiendo una transmisión bidireccional entre todos los datos (las entradas y salidas del sistema).



**Figura 31.** Conexión entre 4DIAC-FORTE a través de los archivos DLLs

**Elaborado por:** Investigadores

Esta conexión entre el 4DIAC y ODK 1500S permite la creación de sub-aplicaciones para cada ET 200SP. El IPC principal y de respaldo del Módulo Físico siendo estructuradas por medio de bloques de función tal como se aprecia en la figura 32, posee bloques DI\_IPC (1) que lee 4 entradas del controlador S7-1500 del ET 200SP correspondientes a los sensores de presencia, inductivo, óptico y reflectivo, también posee una salida STATE (2) que se activa en caso de existir un fallo en la comunicación con el IPC, son empaquetados en SUBAPP (3) luego transmiten al MD mediante un bloque PUBLISH (4). Para controlar las acciones que debe realizar el proceso de clasificación el MD transmite al bloque SUBSCRIBE (5) las ordenes que son ejecutadas por el bloque DO\_IPC (6).

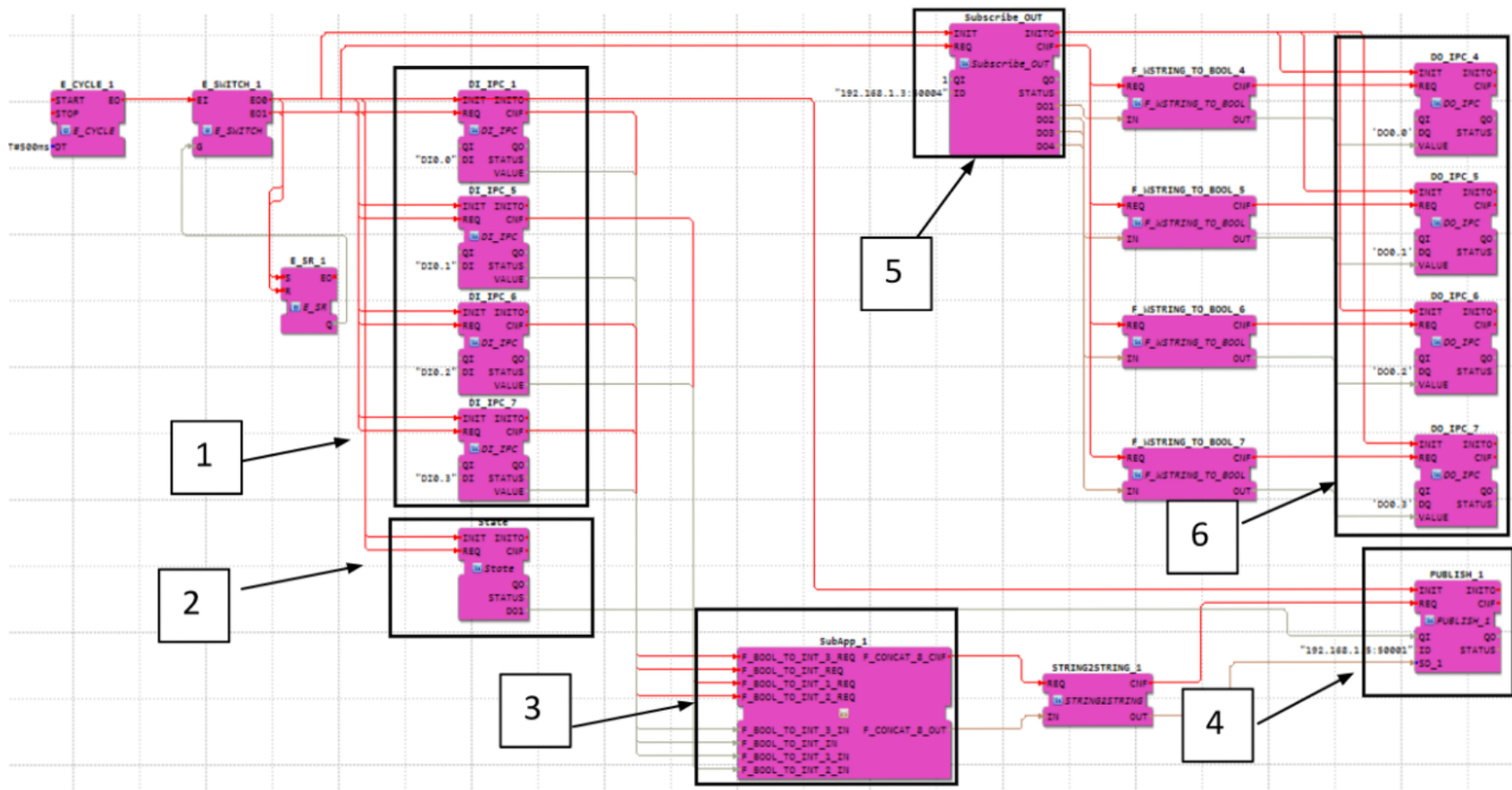


Figura 32. MF Aplicación desarrollada en 4Diac para el control de salidas y entradas de los IPCs

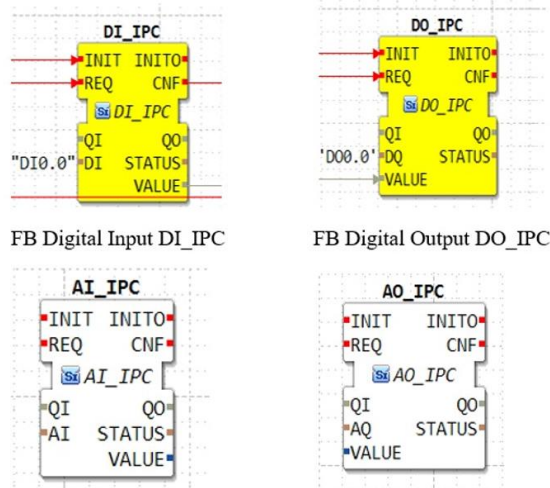
Elaborado por: Investigadores

Los FB's están diseñados para la lectura de valores tanto digitales como analógicos así mismo para la escritura de valores de salida digitales y analógicos desde el IPC. En la figura 36 se representa cada FB, los cuales constan de dos eventos en entrada: el evento INIT se ejecuta al iniciar el FB y el evento REQ que se ejecuta cada 500 milisegundos para leer el valor de la entrada, del mismo modo tiene dos eventos de salida: el evento INITO indica que se completó la inicialización del FB y el evento CNF que indica que se ejecutó correctamente el evento de entrada REQ.

El FB Digital Input y el FB Analogic Input constan de una entrada de datos tipo WSTRING llamada DI y AI respectivamente para el control de entradas digitales y analógicas, en donde se encuentra el nombre de la entrada y una salida de datos tipo BOOL para la entrada digital e INT para la entrada analógica nombrada VALUE la cual indica su valor.

Los FB' Digital Output y Analog Output están diseñados para la escritura de valores a las salidas digitales y analógicas del IPC. Constan de dos eventos en la entrada: el evento INIT se ejecuta al iniciar el FB y el evento REQ que se ejecuta periódicamente para escribir el valor de salida, del mismo modo tiene dos eventos de salida: el evento INITO indica que se completó la inicialización del FB y el evento CNF que indica que se ejecutó correctamente el evento de entrada REQ.

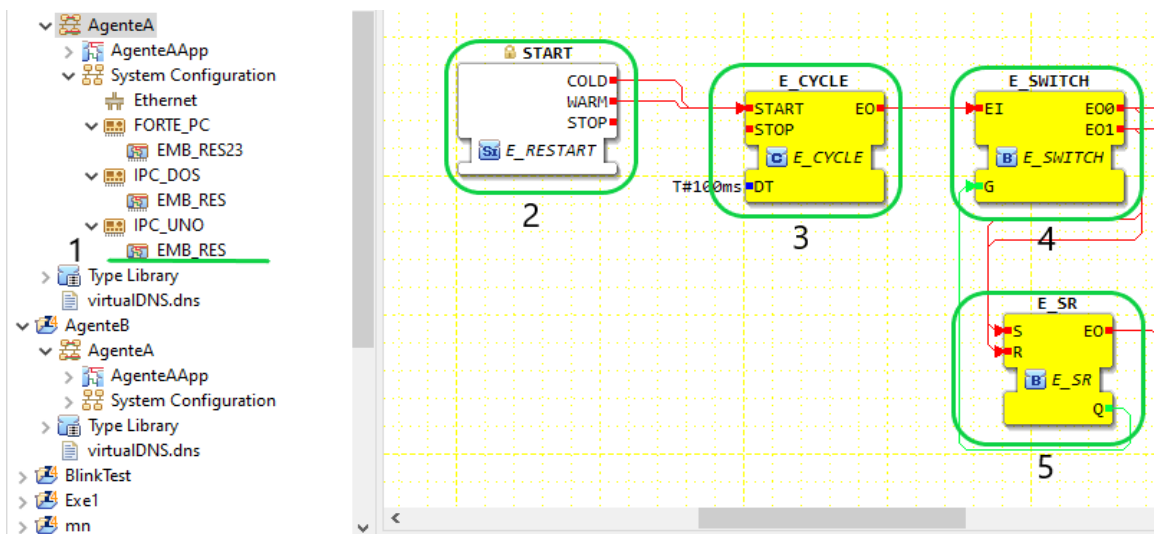
Además, el FB Digital Output consta de dos entradas de datos tipo STRING llamada DQ y el Analog Output consta de dos entradas de datos tipo WSTRING llamada AQ, en donde se encuentran los nombres de las salidas una de tipo BOOL nombrada VALUE la cual indica su valor. En la figura 33 se representa cada FB.



**Figura 33.** Bloques Funcionales entradas y salidas del IPC

**Elaborado por:** Investigadores

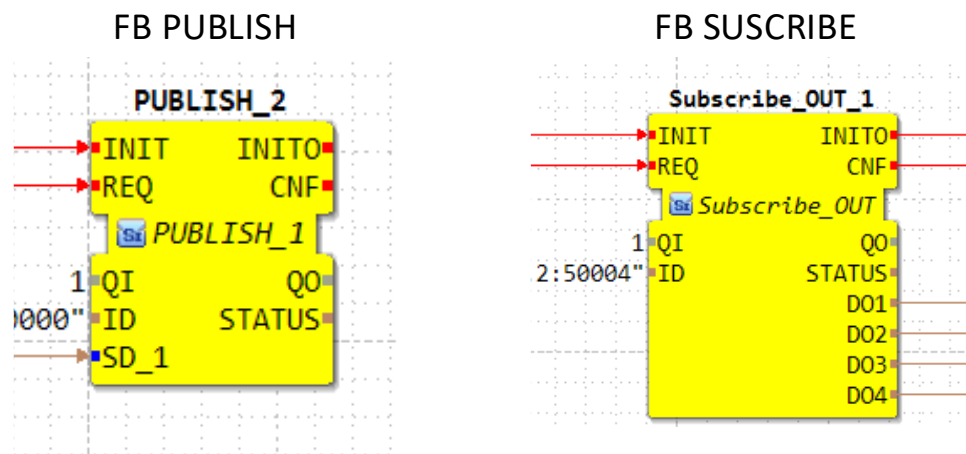
La configuración se inicia creando dentro de cada dispositivo un evento de arranque de un ciclo, llegando un evento a (3) E\_CYCLE.START, en donde se asigna la frecuencia que tendrá la ejecución del ciclo y disparará una salida de evento en E\_CYCLE.EO. (4) E\_SWITCH disparará una salida de evento EO0 cuando G sea cero, y EO1 cuando G sea uno. El valor Q de (5) E\_SR será 1 cuando llegue un evento a S y 0 cuando llegue un evento a R, esto a su vez son conectados con los bloques funcionales de entradas y salidas del IPC. Ver figura 34.



**Figura 34.** Evento de arranque

**Elaborado por:** Investigadores

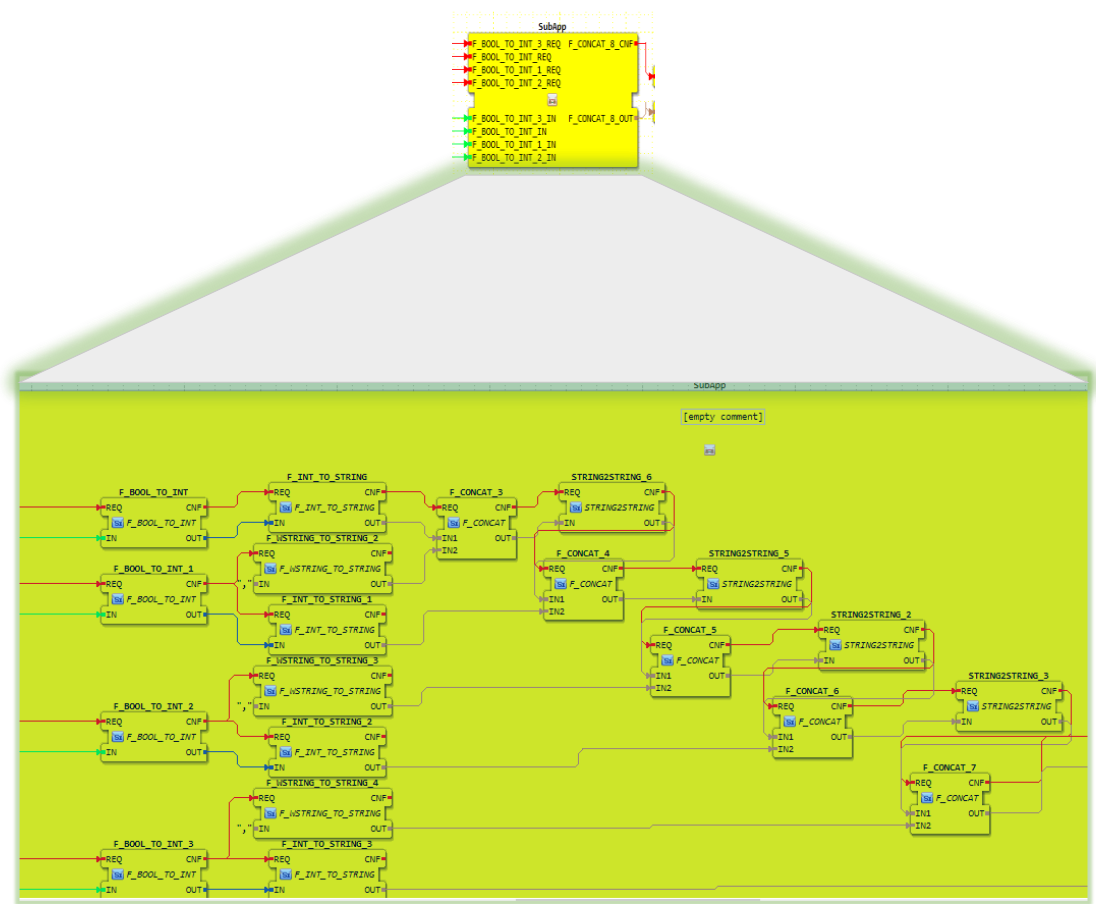
Para comunicarse dentro de la aplicación de 4Diac con distintos dispositivos Forte establecidos en IPC S7-1500 se utiliza el método de publicación y suscripción. El FB PUBLISH\_X, representado en la figura 35, se utiliza para enviar mensajes a través de la red que son recibidos por un FB SUBSCRIBE\_X. Cada vez que se activa un REQ, se envía un mensaje de acuerdo con el ID introducido. Sino se especifica un protocolo, se utilizará el protocolo predeterminado en el "IEC 61499 Compliance Profile for Feasibility Demonstrations" El número X de PUBLISH\_X es el número de elementos de datos que desea enviar en el mensaje.



**Figura 35.** BF publish y subscribe

**Elaborado por:** Investigadores

Debido a que el FB publish solo admite un mensaje en cadena por cada evento, se crea una Sub-aplicación (unión de FBs dentro de un solo bloque) para enviar los datos de entrada obtenidas del IPC mediante el FB Publish, concatenando todas las entradas en una única salida. Ver figura 36.



**Figura 36.** Sub-aplicación para envío de datos al publish

Elaborado por: Investigadores

### 3.1.2. Módulo Digital

#### Multiagentes

El módulo digital, dimensionado mediante bloques de función de interfaz de servicio, contiene la obtención de los estados de los sensores y la lógica de funcionamiento de los actuadores de la estación de clasificación MPS 500 simulada en Factory.IO, además contiene la lógica programable de monitoreo, toma de decisiones y ejecución en caso de existir algún fallo por parte del equipo principal.

Está compuesto por 4 bloques Funcionales que representan a cada agente, dos agentes de Monitorización AM, uno de análisis AA, y un agente de ejecución AE; los mismos están desarrollados en Python bajo la librería SPADE que permite la comunicación de los agentes a través del protocolo XMPP. Cada agente está programado para realizar un proceso diferente con un objetivo compartido, el comportamiento de cada agente se describe en la figura 37.

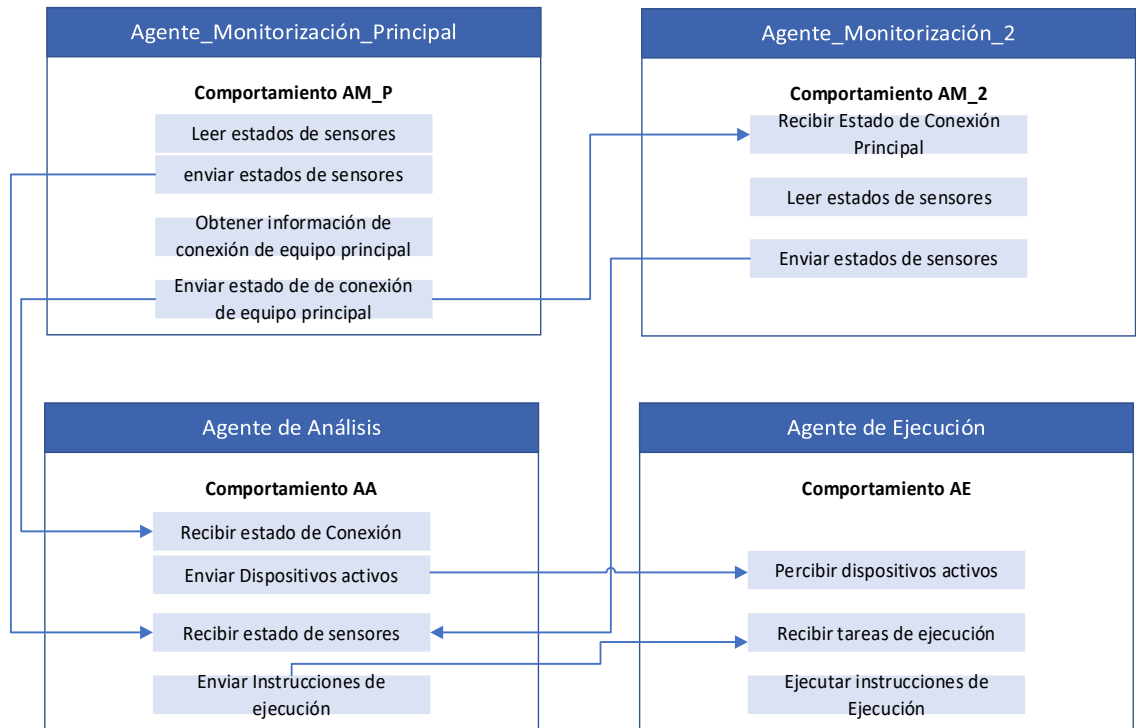


El agente de Monitorización principal AM\_P, es el encargado de recibir cada estado de los sensores simulados, obtenidas mediante la DAC (ATMEGA328 Shield Ethernet), conectados al IPC principal, dichos estados son enviados al sistema operativo del IPC, posteriormente las lecturas de los sensores son enviadas al Agente de Análisis AA. En el caso de existir algún fallo por desconexión del IPC principal, el Agente de Monitorización enviará dicha información al Agente de Análisis AA y al Agente de Monitorización de respaldo.

El Agente de Monitorización de respaldo AM\_2 solamente funcionará cuando perciba que el Agente de Monitorización principal AM haya dejado de trabajar, realizando las tareas de lectura y transmisión de estado de los sensores simulados, que se encuentran conectados de manera paralela al IPC de respaldo.

El Agente de Análisis captura los estados de los sensores enviados desde el Agente de Monitoreo AM y mediante programación decide las acciones a ejecutarse mediante los actuadores. Si una pieza es detectada mediante el sensor de presencia activa la banda transportadora y un pistón de retención para detener su movimiento durante 4 segundos en el sensor inductivo, luego el motor arrancará y el pistón retorna a su lugar, de acuerdo a la señal recibida por el sensor inductivo se activará el primer, segundo o ningún pistón de clasificación, al final cuando se detecta la señal del sensor reflectivo se apagará el motor y retorna cada pistón. Cuando el Agente de Análisis AA recibe una notificación de fallo por parte del Agente de Monitoreo principal le ordena al Agente de Ejecución que el dispositivo activo será el secundario.

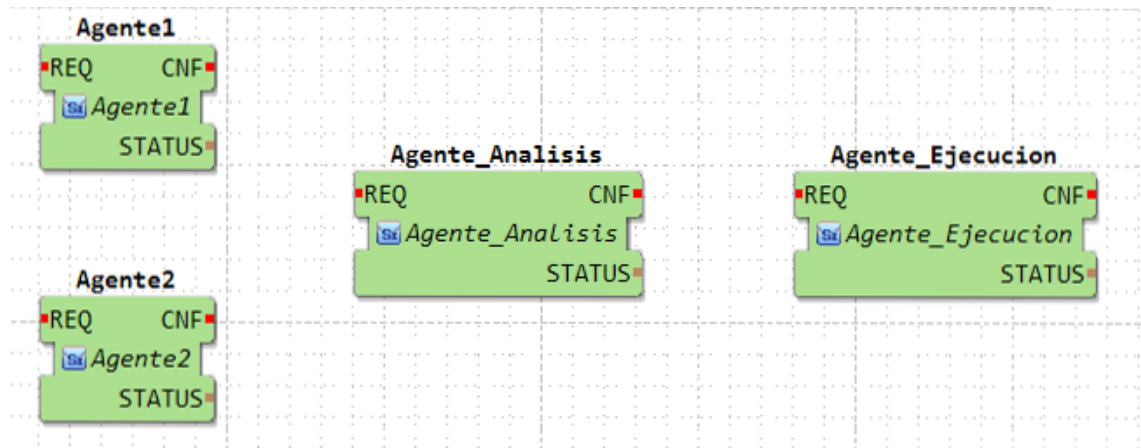
El agente de Ejecución AE recibe todas las tareas enviadas por el Agente de Análisis AA, posteriormente condiciona la orden para transmitir al IPC principal para su ejecución mediante los actuadores. En caso de percibir la señal de fallo del IPC principal, el Agente de Ejecución AE redireccionará las ordenes hacia el IPC de respaldo.



**Figura 37.** Lógica de funcionamiento de los agentes

**Elaborado por:** Investigadores

Como se mencionó anteriormente el módulo Digital está compuesto por cuatro bloques funcionales los cuales pertenecen a cada agente. Al igual que los Bloques funcionales anteriormente mencionados los agentes son creados utilizando los bloques de interfaz de servicio programados en Python embebidos en C++, gracias a la librería de programación de agentes SPADE los Bloques Funcionales se ejecutan cíclicamente al poner en marcha todo el proceso iniciando así a cada agente y manteniéndolo activado en espera de información por parte de sensores para que mediante su programación propia tome decisiones y se comuniquen con el siguiente agente. Figura 38.



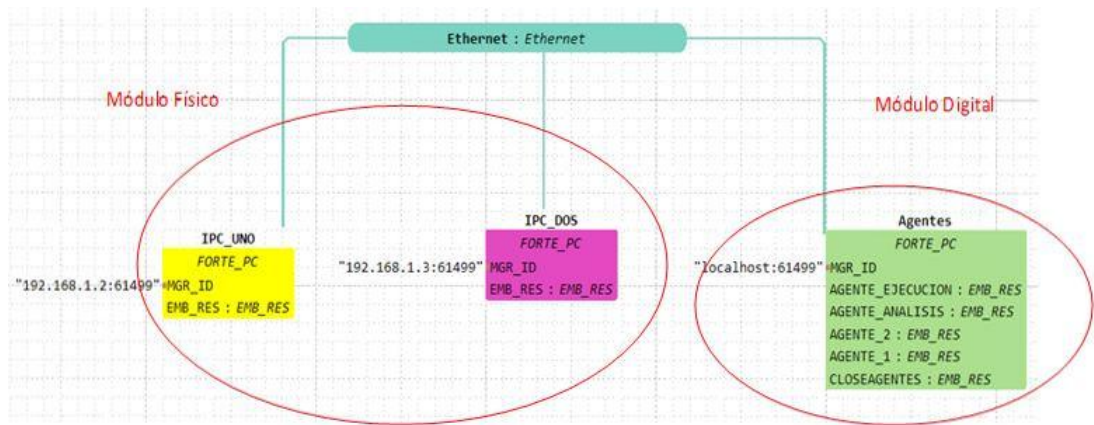
**Figura 38.** Módulo Digital

**Elaborado por:** Investigadores

### 3.1.3. Comunicación entre el Módulo Físico y el Módulo Digital

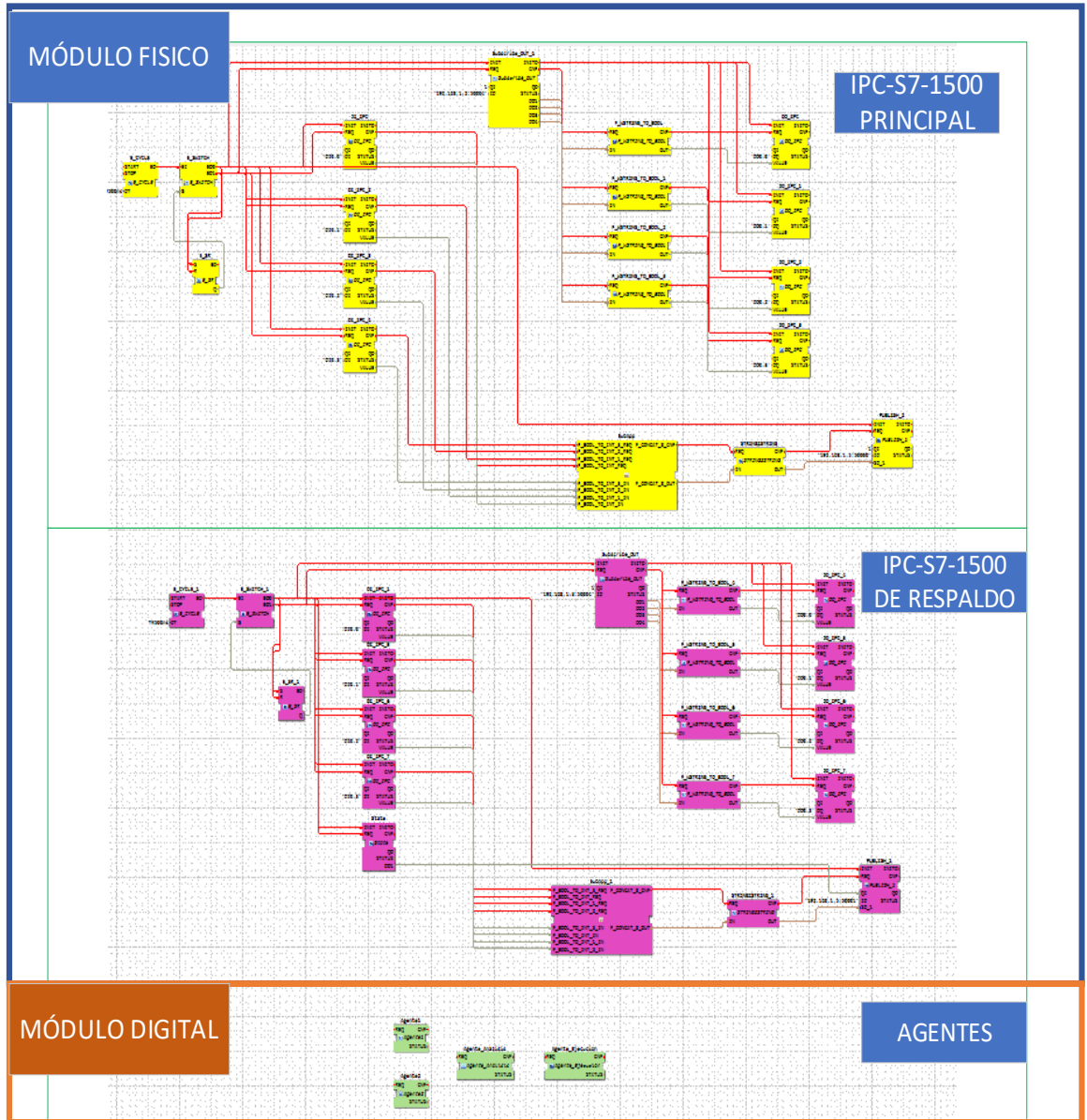
La comunicación de datos entre el módulo digital MD y el módulo físico MF se realiza mediante bloques funcionales FB bajo la norma IEC 61499, mientras que los bloques funcionales FB usan el protocolo de transporte UDP (User Datagram Protocol), que comparado al protocolo TCP este no requiere establecer una conexión antes de enviar los datos, es decir no solicita una comunicación emisión - recepción con mensaje de confirmación. El protocolo UDP es normalmente requerido en aplicaciones en tiempo real, como en señales en vivo de audio y video, o donde en la comunicación la prioridad del emisor sea publicar o enviar los datos a su disposición, como es el caso de la monitorización de sensores y actuadores en las que a pesar de no existir un receptor en condiciones de trabajo la información se sigue transmitiendo.

En la figura 39 se representa la interconexión del Módulo Físico y el Módulo Digital la cual se realiza por medio de un sistema de configuración de red mediante Ethernet el cual genera un evento de arranque para cada grupo de Bloques Funcionales por el recurso asignado Emb-Res al ejecutarse el Runtime Forte en conjunto con la dirección IP y el puerto del dispositivo, conectado el Módulo Físico MF con el Módulo Digital MD. Dando como resultado la comunicación entre el Módulo Físico y el Módulo Digital permitiendo el control de la estación de clasificación simulada mediante los Bloques Funcionales y un dispositivo de respaldo en caso de fallos por desconexión. Un esquema final de los bloques funcionales desarrollados se presenta en la figura 40.



**Figura 39** Conexión de los módulos

**Elaborado por:** Investigadores

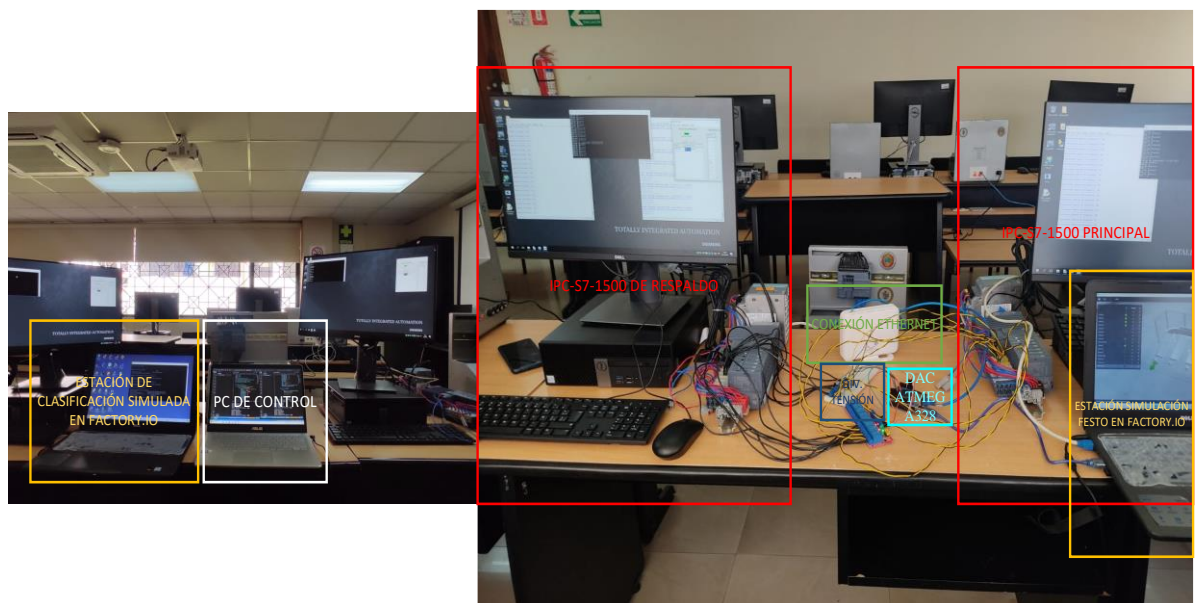


**Figura 40.** Esquema Final de control de proceso

**Elaborado por:** Investigadores

### 3.1.4. Pruebas del sistema implementado

El proyecto se implementó mediante la conexión del software Factory IO hacia un módulo DAC ATMEGA328 Shield Ethernet (trabaja de intermediario para decodificar las señales virtuales en señales físicas) permitiendo la conexión de las señales de manera paralela hacia cada ET 200SP Open Controller CPU 1515S PC. Las pruebas se desarrollaron conectando los equipos como se describe en el ítem desarrollo, la conexión del hardware se observa en la figura 41. Además, el direccionamiento de las entradas y salidas tanto físicas como digitales de los módulos de los ET 200SP Open Controller CPU 1515S PCy 4DIAC se encuentran registrados en la tabla número 1 anexos.



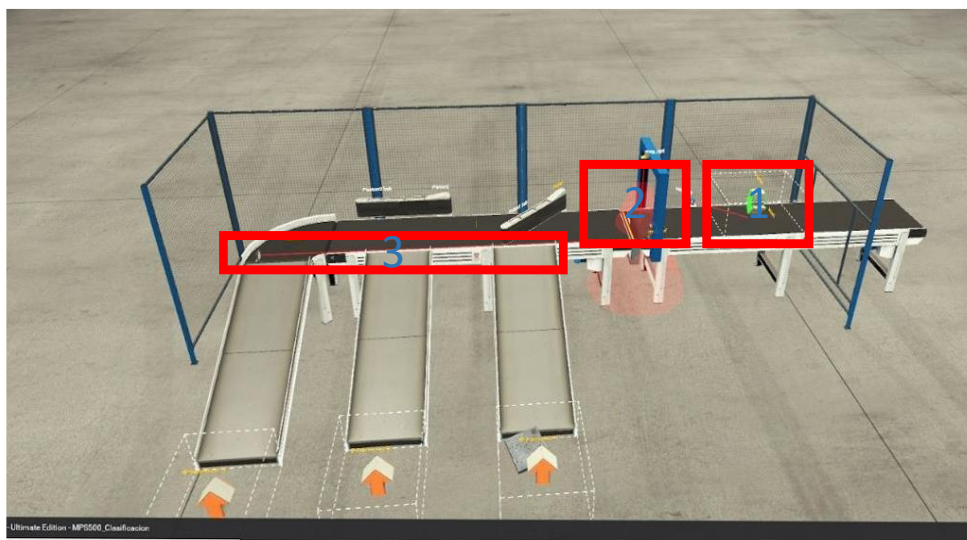
**Figura 41.** Conexión hardware

**Elaborado por:** Investigadores

Las pruebas se realizaron cuando el equipo PLC- ET 200SP Open Controller CPU 1515S PC principal presente alguna falla por desconexión entonces el segundo equipo ET 200SP Open Controller CPU 1515S PC de respaldo entra en funcionamiento, teniendo en cuenta los siguientes dos criterios principales: primero, el equipo de respaldo debe continuar desde el punto en que el equipo principal dejó de funcionar, y segundo, el equipo de respaldo debe tomar el control en un tiempo relativamente corto,

así asegurando que no existan pérdidas considerables por la paralización de proceso de clasificación.

Para comprobar que el sistema implementado podía continuar con su funcionamiento al presentar algún fallo desde el punto en que se encontraba el proceso, se procedió a realizar 20 pruebas de desconexión en diferentes puntos durante su transcurso para así comprobar el tiempo de respuesta del equipo de respaldo verificando su funcionalidad y que tan rápido actúa, en este caso se toma en cuenta tres puntos indicados en la figura 42.



**Figura 42.** Puntos de prueba

**Elaborado por:** Investigadores

El tiempo que tarda el equipo de respaldo en tomar el control del proceso se toman con la ayuda del software Whire shark el cual es un analizador de red. Los resultados obtenidos mediante este software en las pruebas de desconexión se presentan en la Figura 43 donde se capturan la información de los datos que salen desde la PC de control hacia los dos ET 200SP CPU 1515SP que tienen una comunicación por medio del protocolo UDP. El agente de Ejecución envía las ordenes hacia la IP 192.168.1.2 con puerto 50004 hacia el equipo principal y al ocurrir una falla por desconexión el Agente de Análisis redirecciona el flujo de datos hacia la IP 192.168.1.3 con puerto 50004 donde se encuentra el equipo de respaldo. El tiempo que tarde en realizar el cambio de IP, es decir, el tiempo en que el equipo de respaldo empieza a funcionar se refleja en la columna “Time” de aproximadamente 2.5 segundos, debido a que la

estación de clasificación FESTO es simulada mediante el software FACTORY IO y sus señales son captadas mediante la DAC existe un retardo en la comunicación y en el análisis de tráfico de red captadas por el software Wireshark, este tiempo es similar en cada uno de los tres puntos que se realizan las pruebas de funcionamiento.

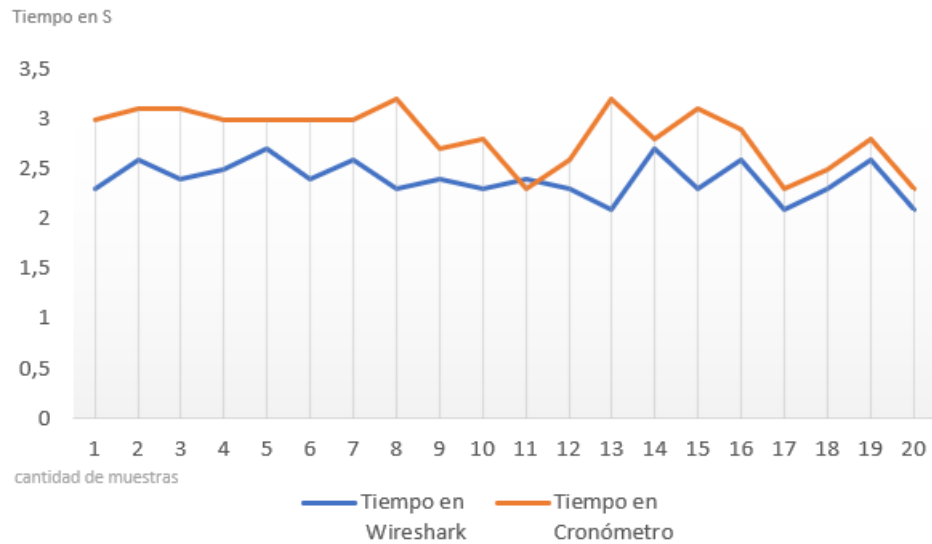
Como otra fuente de medida se procesa a tomar el tiempo de reconexión mediante un cronómetro manual siendo esta una manera didáctica de comprobar el tiempo de funcionamiento del equipo luego de su desconexión. En la Figura 44 se observa una gráfica comparativa de los datos recopilados por el cronómetro y los tiempos medidos mediante el software Whireshark en la estación de clasificación, la comparación da como resultado una diferencia de aproximadamente 1 segundo. Esta diferencia de tiempos se atribuye a que el software Wireshark realiza la medición cuando el Agente de Ejecución envía la información al equipo de respaldo mientras que el cronómetro manipulado manualmente mide el tiempo desde la desconexión hasta la puesta en marcha por el equipo de respaldo, es decir que el cronómetro mide el tiempo de análisis del Agente de Análisis, Agente de ejecución y tiempo de respuesta del equipo de respaldo.

No.	Time	Source	Destination	Protocol	Length	Info
609	40.063559	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
646	42.845966	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
739	49.824888	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
750	50.823160	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
763	51.931636	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
765	52.040591	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
800	54.826707	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
801	54.929806	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
819	55.825100	192.168.1.5	192.168.1.2	UDP	53	58061 → 50004 Len=11
869	59.039373	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
873	59.152757	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
886	60.037946	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
888	60.147426	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
912	62.038290	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
972	67.040204	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
994	69.147126	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
997	69.252956	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11
999	69.363022	192.168.1.5	192.168.1.3	UDP	53	58143 → 50004 Len=11

**Figura 43.** Tiempo registrado por Wireshark

**Elaborado por:** Investigadores

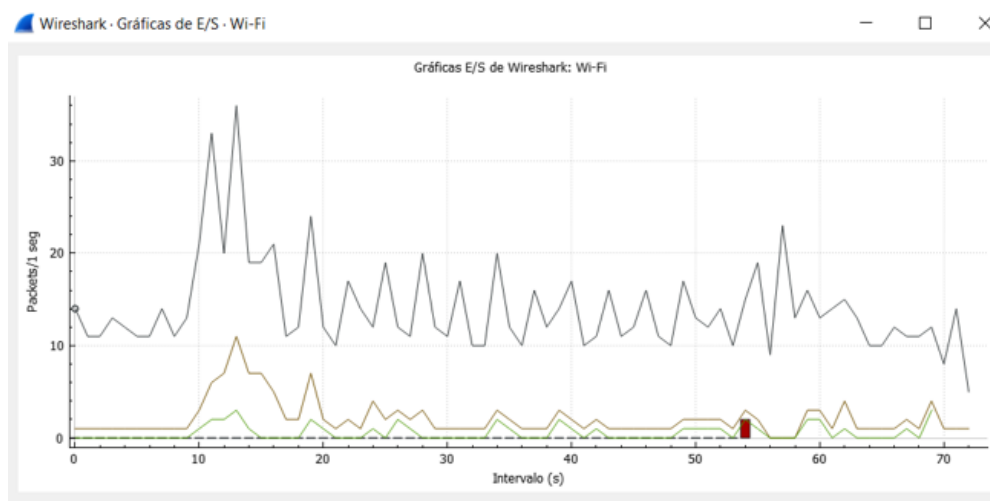




**Figura 44.** Gráfica comparativa de tiempos

**Elaborado por:** Investigadores

Como respaldo de los datos tomados mediante el software Whireshark se presenta una gráfica, figura 45, en la cual se observa los valores de retardo existentes en la planta de clasificación, donde al provocar una desconexión en el equipo principal durante el segundo 10 se presenta una latencia de aproximadamente 2,3 segundos, hasta retomar el control normal de la planta por el segundo IPC de respaldo. Lo cual denota que al tener un fallo por desconexión en la planta de clasificación existe un retardo de los datos, hasta que se establezca el sistema por medio del equipo de respaldo.



**Figura 45.** Gráfica de retardo Whireshark

**Elaborado por:** Investigadores

Con todos los datos recabados mediante las pruebas realizadas en la planta de clasificación se comprueba que el funcionamiento del sistema multiagente puede recuperar el control del proceso, desde cualquier punto en el que se produzca una falla, permitiendo continuar su operación luego de un tiempo estimado de 3 segundos desde la perspectiva humana. Aun cuando parezca un tiempo muy alto, en un proceso de producción masiva este sistema es muy rápido y eficaz lo cual evita pérdidas económicas y de tiempo, además, se tiene en cuenta que la comunicación que utilizan los agentes mediante el protocolo XMPP no es inmediata ya que depende de factores como la velocidad de red y tiempos de retraso entre paquetes de datos de los agentes de monitoreo.

## **CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES**

### **4.1 Conclusiones**

Con los resultados expuestos en la sección anterior se ratifica la factibilidad de implementar una plataforma Multiagente basado en la norma IEC-61499 que realiza una reconfiguración dinámica en la estación de clasificación FESTO MPS 500 (sea esta simulada o física) para responder ante cualquier falla por desconexión y retomar el proceso en el punto en el que este haya quedado. Las pruebas que simularon fallas por desconexión realizadas en diferentes puntos del proyecto son evidencias de su funcionalidad, así se comprobó que el sistema es capaz de retomar la normalidad del proceso en un tiempo no mayor a 3 segundos, este tiempo es considerado relativamente corto si se compara con los procesos que normalmente no poseen este sistema, ya que el tiempo de paro, reconfiguración manual y conexión de los mismos son ejecutados en lapsos de horas.

El proyecto realizado tiene la capacidad de adaptarse a los procesos de escala industrial en los cuales se necesite asegurar el funcionamiento continuo sin afectar el rendimiento, la calidad del producto o la intervención directa en una nueva etapa del proceso. Los Bloques Funcionales de entrada y salida creados bajo la norma IEC-61499 extiende un mundo de aplicaciones al proyecto, debido a su capacidad modular posibilita una mejor distribución de la arquitectura facilitando su uso e implementación, por tal motivo siempre que la compatibilidad de software este garantizada se puede implementar más bloques funcionales que permitan la escalabilidad del proyecto de acuerdo a su aplicación; incluso se puede añadir nuevas funciones que permitan un mejor rendimiento del sistema.

El comportamiento cíclico de los agentes se dio gracias al uso de la librería SPADE, permitiendo que los agentes se encuentren siempre a la espera de información para analizar, procesar y tomar decisiones; mediante el protocolo de comunicación XMPP se envían la información por medio de un servidor posibilitando la comunicación entre agentes.

## 4.2 Recomendaciones

- Revisar la documentación de los dispositivos y herramientas a utilizar para comprobar su compatibilidad debido a que para este caso de estudio el ODK utilizado solo trabaja con el CPU 1505SP.
- Utilizar Microsoft Visual Studio hasta la versión 17 según los manuales establecidos.
- Diseñar bloques funcionales individuales para cada tipo de entrada y salida debido a que no es práctico crear un solo bloque funcional ya que en la parte práctica se necesita más recursos.
- Según los manuales recomiendan colocar el dll generado en el ODK en un directorio específico del disco local C para que pueda comunicarse el CPU y el Windows siendo la dirección: C:\Program Files\Siemens\Automation\ODK1500S,
- Si desea utilizar el entorno de desarrollo Microsoft Visual Studio, es recomendable instalarlo antes de instalar ODK.

## C. MATERIALES DE REFERENCIA

### REFERENCIAS BIBLIOGRÁFICAS

- [1] D. Drozdov, V. Dubinin, y V. Vyatkin, «Speculative computation in IEC 61499 function blocks execution — Modeling and simulation», en *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, Poitiers, France, jul. 2016, pp. 748-755. doi: 10.1109/INDIN.2016.7819259.
- [2] G. Caiza, M. Saeteros, W. Oñate, y M. V. Garcia, «Fog computing at industrial level, architecture, latency, energy, and security: A review», *Heliyon*, vol. 6, n.º 4, p. e03706, abr. 2020, doi: 10.1016/j.heliyon.2020.e03706.
- [3] M. V. Garcia, E. Irisarri, F. Perez, M. Marcos, y E. Estevez, «From ISA 88/95 meta-models to an OPC UA-based development tool for CPPS under IEC 61499», en *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, Imperia, jun. 2018, pp. 1-9. doi: 10.1109/WFCS.2018.8402362.
- [4] C. Pang, V. Vyatkin, y W. Dai, «IEC 61499 based model-driven process control engineering», en *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Barcelona, Spain, sep. 2014, pp. 1-8. doi: 10.1109/ETFA.2014.7005088.
- [5] W. Lepuschitz y A. Zoitl, «An engineering method for batch process automation using a component oriented design based on IEC 61499», en *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, Hamburg, Germany, sep. 2008, pp. 207-214. doi: 10.1109/ETFA.2008.4638394.
- [6] P. Gsellmann, M. Melik-Merkumians, y G. Schitter, «Comparison of Code Measures of IEC 61131-3 and 61499 Standards for Typical Automation Applications», en *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Turin, sep. 2018, pp. 1047-1050. doi: 10.1109/ETFA.2018.8502464.

[7] Condoy Carrión, Alex Rubén, «Agentes inteligentes para la gestión de recursos digitales de aprendizaje». 13 de septiembre de 2019. [En línea]. Disponible en:

<https://dspace.unl.edu.ec/jspui/bitstream/123456789/22803/1/Condoy%20Carri%C3%B3n%20Alex%20Rub%C3%A9n.pdf>

[8] M. Sahnoun, Y. Xu, B. Belgacem, B. Imen, B. David, y A. Louis, «Fractal modeling of Cyber physical production system using multi-agent systems», en *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*, Elazig, Turkey, sep. 2019, pp. 1-6. doi: 10.1109/ICAAID.2019.8934976.

[9] M. Ya. Afanasev, Y. V. Fedosov, A. A. Krylova, y S. A. Shorokhov, «Modular industrial equipment in cyber-physical production system: Architecture and integration», en *2017 21st Conference of Open Innovations Association (FRUCT)*, Helsinki, nov. 2017, pp. 1-9. doi: 10.23919/FRUCT.2017.8250158.

[10] M. J. Ávila-Gutiérrez, F. Aguayo-González, M. Marcos-Bárcena, J. R. Lama-Ruiz, y M. E. Peralta-Álvarez, «Arquitectura holónica de referencia para empresas de fabricación sostenibles distribuidas», *DYNA*, vol. 84, n.º 200, pp. 160-168, ene. 2017, doi: 10.15446/dyna.v84n200.53095.

[11] S. Braun, C.-T. Cheng, S. Dowey, y J. Wollert, «Survey on Security Concepts to Adapt Flexible Manufacturing and Operations Management based upon Multi-Agent Systems», en *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, Delft, Netherlands, jun. 2020, pp. 480-484. doi: 10.1109/ISIE45063.2020.9152210.

[12] V. Marik y D. McFarlane, «Industrial Adoption of Agent-Based Technologies», *IEEE Intell. Syst.*, vol. 20, n.º 1, pp. 27-35, ene. 2005, doi: 10.1109/MIS.2005.11.

[13] H. Lee, E. J. Rouse, y H. I. Krebs, «Summary of Human Ankle Mechanical Impedance During Walking», *IEEE J. Transl. Eng. Health Med.*, vol. 4, pp. 1-7, 2016, doi: 10.1109/JTEHM.2016.2601613.

- [14] M. V. Garcia, E. Irisarri, F. Perez, E. Estevez, D. Orive, y M. Marcos, «Plant floor communications integration using a low cost CPPS architecture», en *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, sep. 2016, pp. 1-4. doi: 10.1109/ETFA.2016.7733631.
- [15] Z. Jiang, Y. Jin, M. E, y Q. Li, «Distributed Dynamic Scheduling for Cyber-Physical Production Systems Based on a Multi-Agent System», *IEEE Access*, vol. 6, pp. 1855-1869, 2018, doi: 10.1109/ACCESS.2017.2780321.
- [16] R. W. Brennan, X. Zhang, Y. Xu, y D. H. Norrie, «A reconfigurable concurrent function block model and its implementation in real-time Java», *ICA*, vol. 9, n.º 3, pp. 263-279, jul. 2002, doi: 10.3233/ICA-2002-9306.
- [17] J. Wan *et al.*, «Toward Dynamic Resources Management for IoT-Based Manufacturing», *IEEE Commun. Mag.*, vol. 56, n.º 2, pp. 52-59, feb. 2018, doi: 10.1109/MCOM.2018.1700629.
- [18] V. Hipola, Pedro Quesada, «Agentes inteligentes: definición y tipología. Los agentes de información», *IWE*, vol. 8, n.º 4, p. 9, abril de 1999.
- [19] A. S. M. Mantilla Miranda, «Diseño de reconfiguración dinámica basada en agentes inteligentes para sistemas distribuidos de bajo costo.», Institucional, Escuela Superior Politécnica de Chimborazo, Riobamba, 2019. [En línea]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/10170>
- [20] AENOR, «Uso y referencia a normas ISO e IEC en la reglamentación técnica». AENOR, septiembre de 2017. [En línea]. Disponible en: [https://www.en.une.org/normalizacion\\_documentos/referencia\\_normas\\_iso\\_iec\\_reg\\_tecnica.pdf](https://www.en.une.org/normalizacion_documentos/referencia_normas_iso_iec_reg_tecnica.pdf)
- [21] E. Querol, J. A. Romero, A. M. Estruch, y F. Romero, «Norma IEC-61499 para el control distribuido. Aplicación al CNC.», *CEA-IFAC*, p. 8, sep. 2014.
- [22] Catalán Cantero, Carlos, «Modelos y plataforma IEC 61499 adaptados al control distribuido de máquinas herramienta en sistemas de fabricación ágil». Zaragoza, septiembre de 2015. [En línea]. Disponible en: <http://zaguan.unizar.es/collection/Tesis>

- [23] F. Eclipse, «4DIAC IDE», *Documentación Eclipse 4DIAC*, septiembre de 2017. [https://www.eclipse.org/4diac/en\\_help.php?helppage=html/installation/install.html#4DIAC-IDE](https://www.eclipse.org/4diac/en_help.php?helppage=html/installation/install.html#4DIAC-IDE)
- [24] SIEMENS, «SIMATIC ET 200SP Open Controller CPU1515 PC2». SIEMENS, diciembre de 2016. [Windows 10 IoT Interprise]. Disponible en: [support.industry.siemens.com](http://support.industry.siemens.com)
- [25] SIEMENS, «Módulo de salidas analógicas AQ. 2xU/I HS (6ES7135-6HB00-0DA1)». SIEMENS, septiembre de 2016. [Windows 10 IoT Interprise]. Disponible en: [support.industry.siemens.com](http://support.industry.siemens.com)
- [26] «Módulo de entradas analógicas AI 2xU/I 2-/4-wire HS (6ES7134-6HB00-0DA1)», [En línea]. Disponible en: [support.industry.siemens.com](http://support.industry.siemens.com)
- [27] SIEMENS, «Módulo de entradas digitales DI. 8x24VDC HF (6ES7131-6BF00-0CA0)». SIEMENS, diciembre de 2017. [Windows 10 IoT Interprise]. Disponible en: [support.industry.siemens.com](http://support.industry.siemens.com)
- [28] SIEMENS, «Módulo de salidas digitales DQ . 8x24VDC/0.5A HF (6ES7132-6BF00-0CA0)». SIEMENS, diciembre de 2017. [Windows 10 IoT Interprise]. Disponible en: [support.industry.siemens.com](http://support.industry.siemens.com)
- [29] B. Koray Tuncalp y S. Taskin, «Real Time and Remote Control by LabVIEW of MPS Modular Production System and Technical Education Application», 2007, doi: 10.13140/2.1.3919.4245.
- [30] C. Espinoza y M. Alfredo, «Automatización y monitoreo para planta festo mps-500 del laboratorio de fabricación flexible de UPS-G.», Institucional, Universidad Politécnica Salesiana Sede Guayaquil, Ecuador, 2019. [En línea]. Disponible en: <https://dspace.ups.edu.ec/handle/123456789/17254>
- [31] SIEMENS, «ODK Open Development Kit 1500S V2.5». diciembre de 2017. [En línea]. Disponible en: [https://cache.industry.siemens.com/dl/files/683/109752683/att\\_937390/v1/s71500\\_odk1500s\\_manual\\_en-US\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/683/109752683/att_937390/v1/s71500_odk1500s_manual_en-US_en-US.pdf)



- [32] S. Pérez Frayle, «Integración de agentes deliberativos en la plataforma SPADE: Desarrollo de pGomas», Institucional, Universidad Politécnica de Valencia, Valencia, 2014. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <http://polipapers.upv.es/index.php/IA/article/view/3293>
- [33] J.-L. Granda, L. Chamba-Eras, M. Labanda-Jaramillo, E. Coronel, R. Guaman-Quinche, y C. Maldonado-Ortega, «OpenChatBotUNL: Propuesta de plataforma de ejecución de Agentes Conversacionales».
- [34] SIEMENS, «Módulo de entradas analógicas. AI 2xU/I 2-/4-wire HS (6ES7134-6HB00-0DA1)». SIEMENS, septiembre de 2018. [Windows 10 IoT Enterprise]. Disponible en: [support.industry.siemens.com](http://support.industry.siemens.com)
- [35] M. J. Ayora Madero, «Diseño de un sistema de control distribuido usando Factory IO y Codesys V3 (communication through OPC)», Institucional, Escuela Técnica Superior de Ingeniería Universidad de Sevilla, Sevilla, 2018. [En línea]. Disponible en: <https://idus.us.es/handle/11441/83982>

## ANEXOS

### ANEXO 1. Manual ET 200SP Open Controller CPU 1515S PC



**Figura 1** ET 200SP Open Controller CPU 1515S PC

**Tabla 1** Características técnicas del CPU 1515S PC

General information	
Product type designation	CPU 1515SP PC2 F
HW functional status	from FS04
Firmware version	V21.9
Engineering with	
<ul style="list-style-type: none"> <li>STEP 7 TIA Portal configurable/integrated from version</li> </ul>	V17
Installed software	
<ul style="list-style-type: none"> <li>Visualization</li> <li>Control</li> </ul>	WinCC Runtime Advanced V17 S7-1500 Software Controller CPU 1505SP F
Configuration control	
via dataset	Yes
Control elements	
Mode selector switch	1
Supply voltage	
Rated value (DC)	24 V
permissible range, lower limit (DC)	19.2 V
permissible range, upper limit (DC)	28.8 V
Reverse polarity protection	Yes
Mains buffering	
<ul style="list-style-type: none"> <li>Mains/voltage failure stored energy time</li> </ul>	5 ms
Input current	
Current consumption (rated value)	1.8 A; Full processor load, incl. ET 200SP modules and using USB
Current consumption (in no-load operation), typ.	0.5 A
Current consumption, max.	2.9 A
$I^2t$	0.426 A <sup>2</sup> ·s; with starting current inrush

**ANEXO 2. Manual Módulo de entradas digitales DI 8x24VDC HF (6ES7131-6BF00-0CA0)**



**Figura 2** Módulo de entradas digitales DI

**Tabla 2** Datos técnicos del DI 8x24VDC HF

Referencia	6ES7131-6BF00-0CA0
<b>Información general</b>	
Designación del tipo de producto	DI 8x24 VDC HF
Versión de firmware	V2.0
<ul style="list-style-type: none"> <li>Es posible actualizar el FW.</li> </ul>	Sí
BaseUnits utilizables	BU tipo A0
Código de color para etiqueta de identificación por color de módulo	CC01
<b>Función del producto</b>	
<ul style="list-style-type: none"> <li>Datos de I&amp;M</li> </ul>	Sí; I&M0 a I&M3
<b>Ingeniería con</b>	
<ul style="list-style-type: none"> <li>STEP 7 TIA Portal configurable/integrado desde versión</li> </ul>	V13 SP1/-
<ul style="list-style-type: none"> <li>STEP 7 configurable/integrado desde versión</li> </ul>	V5.5 / -
<ul style="list-style-type: none"> <li>PCS 7 configurable/integrada desde versión</li> </ul>	V8.1 SP1
<ul style="list-style-type: none"> <li>PROFIBUS, versión GSD/revisión GSD o sup.</li> </ul>	un archivo GSD respectivamente con revisión 3 y 5 o sup.
<ul style="list-style-type: none"> <li>PROFINET, versión GSD/revisión GSD o sup.</li> </ul>	GSDML V2.3
<b>Modo de operación</b>	
<ul style="list-style-type: none"> <li>DI</li> </ul>	Sí
<ul style="list-style-type: none"> <li>Contadores</li> </ul>	No
<ul style="list-style-type: none"> <li>Sobremuestreo</li> </ul>	No
<ul style="list-style-type: none"> <li>MSI</li> </ul>	Sí
<b>Tensión de alimentación</b>	
Valor nominal (DC)	24 V
Rango admisible, límite inferior (DC)	19,2 V
Rango admisible, límite superior (DC)	28,8 V
Protección contra inversión de polaridad	Sí

**ANEXO 3.** Manual del módulo de entradas analógicas AI 2xU/I 2-/4-wire HS (6ES7134-6HB00-0DA1)



**Figura 3.** Módulo de entradas analógicas AI

**Figura 4.** Datos técnicos del AI 2xU/I 2-/4-wire HS

Referencia	6ES7134-6HB00-0DA1
<b>Información general</b>	
Designación del tipo de producto	AI 2xU/I 2-/4-wire HS
Versión de firmware	V2.0.1
<ul style="list-style-type: none"> <li>Es posible actualizar el FW.</li> </ul>	Sí
BaseUnits utilizables	BU tipo A0, A1
Código de color para etiqueta de identificación por color de módulo	CC00
<b>Función del producto</b>	
<ul style="list-style-type: none"> <li>Datos de I&amp;M</li> </ul>	Sí; I&M0 a I&M3
<ul style="list-style-type: none"> <li>Rango de medida escalable</li> </ul>	No
<ul style="list-style-type: none"> <li>Valores medidos escalables</li> </ul>	No
<ul style="list-style-type: none"> <li>Adaptación del rango de medida</li> </ul>	No
<b>Ingeniería con</b>	
<ul style="list-style-type: none"> <li>STEP 7 TIA Portal configurable/integrado desde versión</li> </ul>	V13 SP1
<ul style="list-style-type: none"> <li>STEP 7 configurable/integrado desde versión</li> </ul>	V5.5 SP3/-
<ul style="list-style-type: none"> <li>PROFIBUS, versión GSD/revisión GSD o sup.</li> </ul>	GSD revisión 5
<ul style="list-style-type: none"> <li>PROFINET, versión GSD/revisión GSD o sup.</li> </ul>	GSDML V2.3
<b>Modo de operación</b>	
<ul style="list-style-type: none"> <li>Sobremuestreo</li> </ul>	Sí; 2 canales por módulo
<ul style="list-style-type: none"> <li>MSI</li> </ul>	No

**ANEXO 4. Manual del módulo de salidas digitales DQ 8x24VDC/0.5A HF (6ES7132-6BF00-0CA0)**



**Figura 5** Módulo de salidas digitales DQ

**Tabla 3.** Datos técnicos del DQ 8x24VDC/0.5A HF

Referencia	6ES7132-6BF00-0CA0
<b>Información general</b>	
Designación del tipo de producto	DQ 8x24 VDC/0,5 A HF
Versión de firmware	V2.0
<ul style="list-style-type: none"> <li>Es posible actualizar el FW.</li> </ul>	Sí
BaseUnits utilizables	BU tipo A0
Código de color para etiqueta de identificación por color de módulo	CC02
<b>Función del producto</b>	
<ul style="list-style-type: none"> <li>Datos de I&amp;M</li> </ul>	Sí; I&M0 a I&M3
<b>Ingeniería con</b>	
<ul style="list-style-type: none"> <li>STEP 7 TIA Portal configurable/integrado desde versión</li> </ul>	V13 SP1/-
<ul style="list-style-type: none"> <li>STEP 7 configurable/integrado desde versión</li> </ul>	V5.5 / -
<ul style="list-style-type: none"> <li>PCS 7 configurable/integrada desde versión</li> </ul>	V8.1 SP1
<ul style="list-style-type: none"> <li>PROFIBUS, versión GSD/revisión GSD o sup.</li> </ul>	un archivo GSD respectivamente con revisión 3 y 5 o sup.
<ul style="list-style-type: none"> <li>PROFINET, versión GSD/revisión GSD o sup.</li> </ul>	GSDML V2.3
<b>Modo de operación</b>	
<ul style="list-style-type: none"> <li>DQ</li> </ul>	Sí
<ul style="list-style-type: none"> <li>DQ con función de ahorro energético</li> </ul>	No
<ul style="list-style-type: none"> <li>PWM</li> </ul>	No
<ul style="list-style-type: none"> <li>Sobremuestreo</li> </ul>	No
<ul style="list-style-type: none"> <li>MSO</li> </ul>	Sí

**ANEXO 5.** Manual del módulo de salidas analógicas AQ 2xU/I HS  
(6ES7135-6HB00-0DA1)

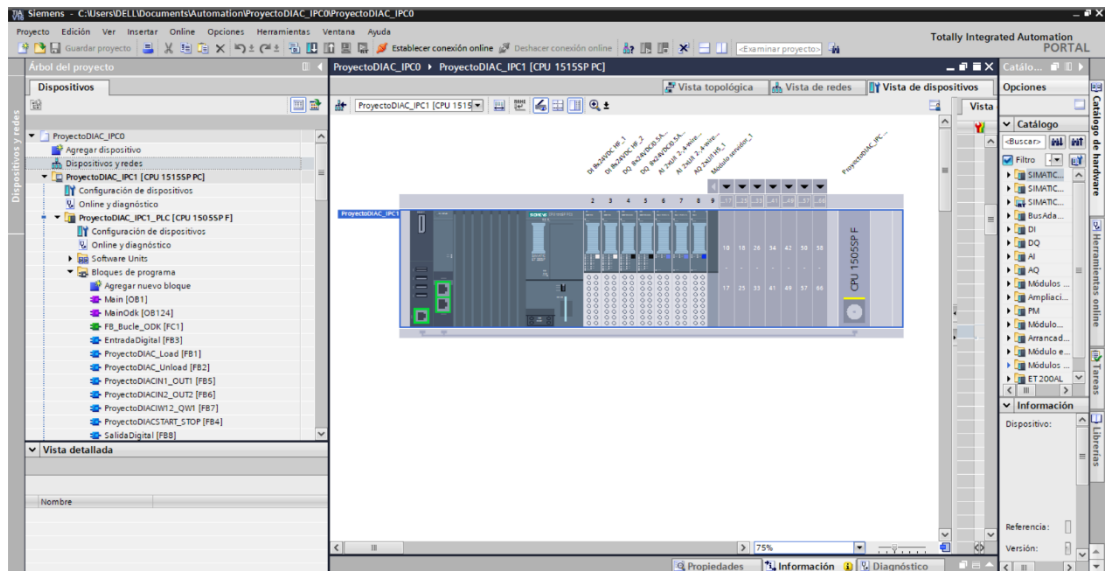


**Figura 6.** Módulo de salidas analógicas AQ

**Tabla 4.** Datos técnicos del AQ 2xU/I HS

	6ES7135-6HB00-0DA1
Designación de tipo del producto	ET 200SP, AQ 2xU/I High Speed
<b>Información general</b>	
Versión de firmware	V2.0.1
BaseUnits utilizables	Tipo de BU A0, A1
Código de color para etiqueta de identificación por color del módulo	CC00
<b>Función del producto</b>	
Datos I&M	Sí; I&M0 a I&M3
<b>Ingeniería</b>	
Configurable/integrado con STEP 7 TIA Portal desde versión	V13 SP1
Configurable/integrado con STEP 7 desde versión PROFIBUS, desde versión GSD/revisión GSD	V5.5 SP3/- GSD revisión 5
PROFINET, desde versión GSD/revisión GSD	GSDML V2.3
<b>Modo de operación</b>	
Oversampling	Sí; 2 canales por módulo
<b>Configuración CiR en RUN</b>	
Reparametrización en RUN	Sí
Posibilidad de calibración en RUN	Sí
<b>Tensión de alimentación</b>	
Valor nominal (DC)	24 V
Rango admisible, límite inferior (DC)	19,2 V
Rango admisible, límite superior (DC)	28,8 V

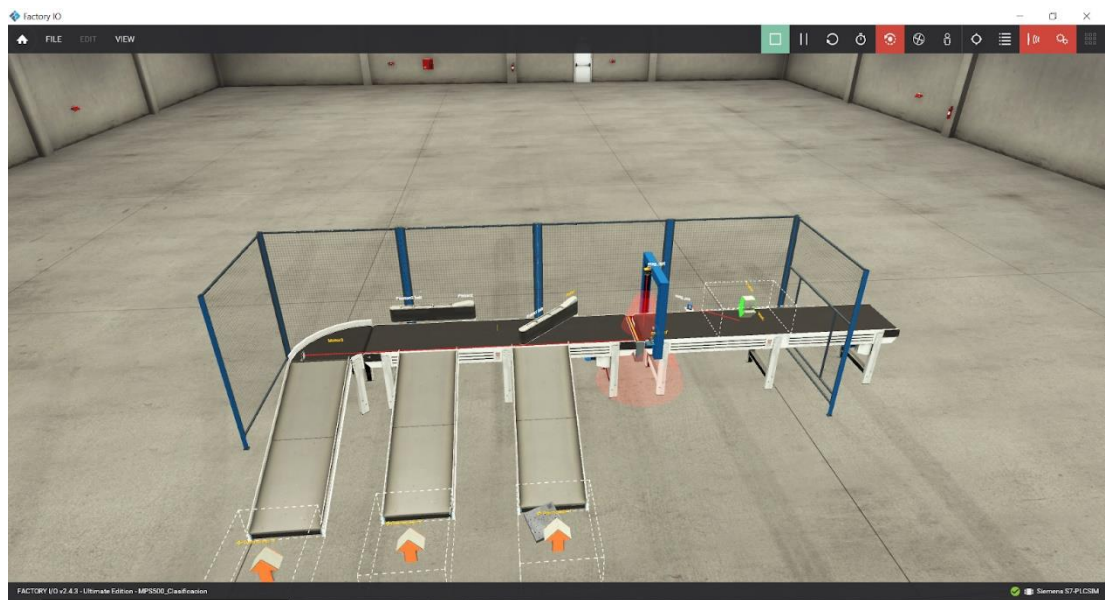
## ANEXO 6.



**Figura 46.** ET 200SP Open Controller CPU 1515S PC en TÍA PORTAL

**Elaborado por:** Investigadores

## ANEXO 7.



**Figura 47** Proceso de clasificación en Factory IO

**Elaborado por:** Investigadores

## ANEXO 8.

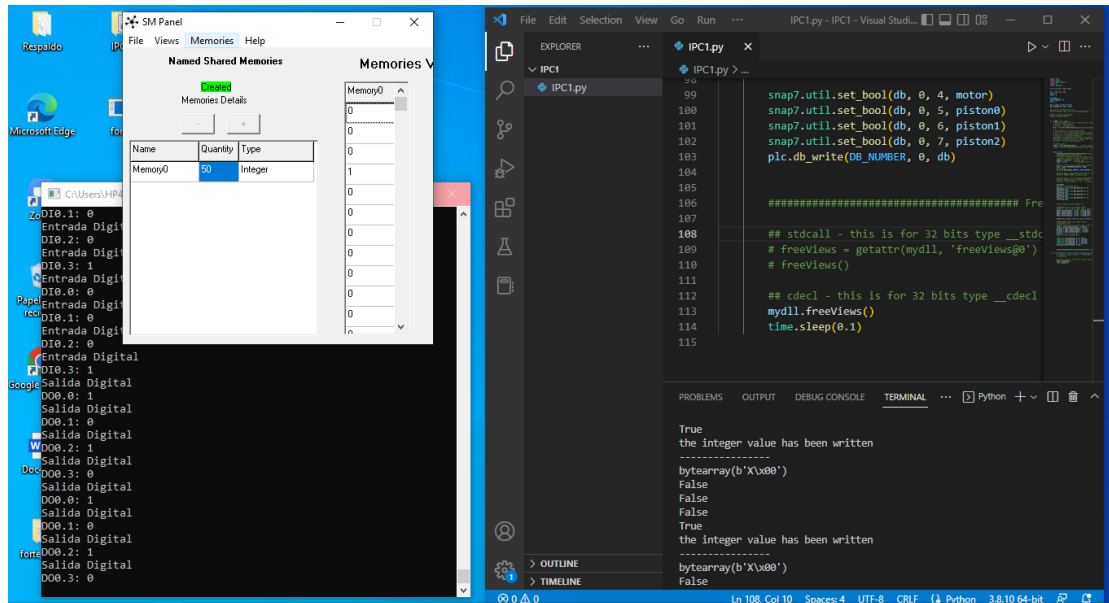


Figura 48 Comunicación con el IPC 1

Elaborado por: Investigadores

## ANEXO 9.

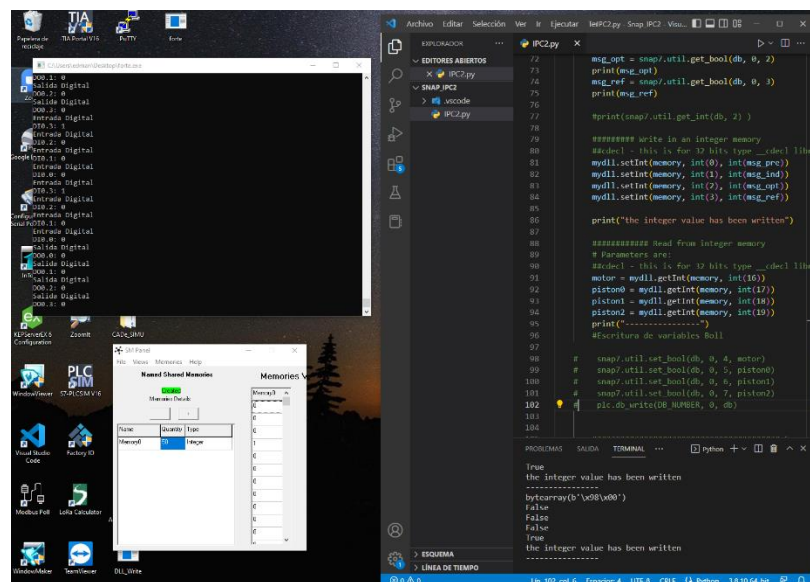


Figura 49. Comunicación con el IPC 2

Elaborado por: Investigadores



## ANEXO 10.

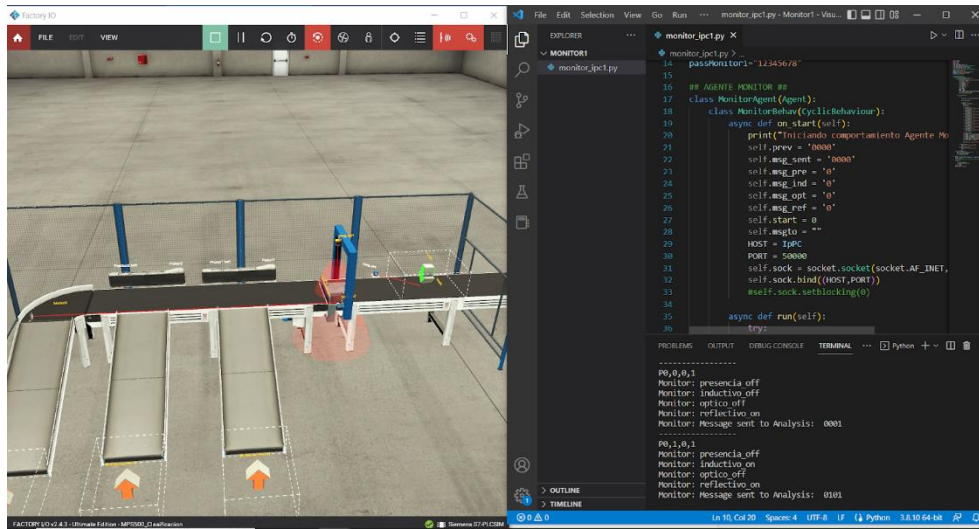


Figura 50. Agente de Monitoreo 1 (AM1)

Elaborado por: Investigadores

## ANEXO 11.

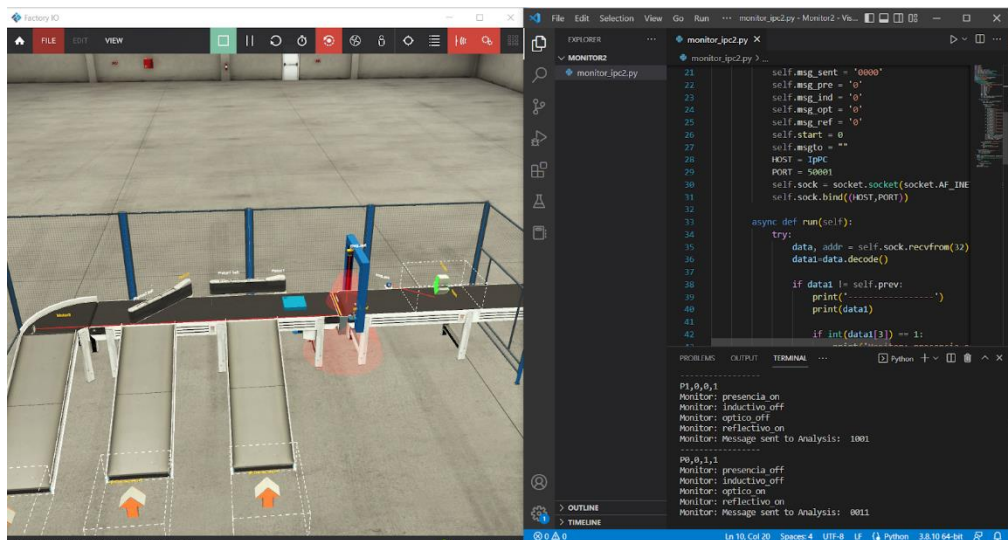


Figura 51. Agente de Monitoreo 2 (AM2)

Elaborado por: Investigadores

## ANEXO 12.

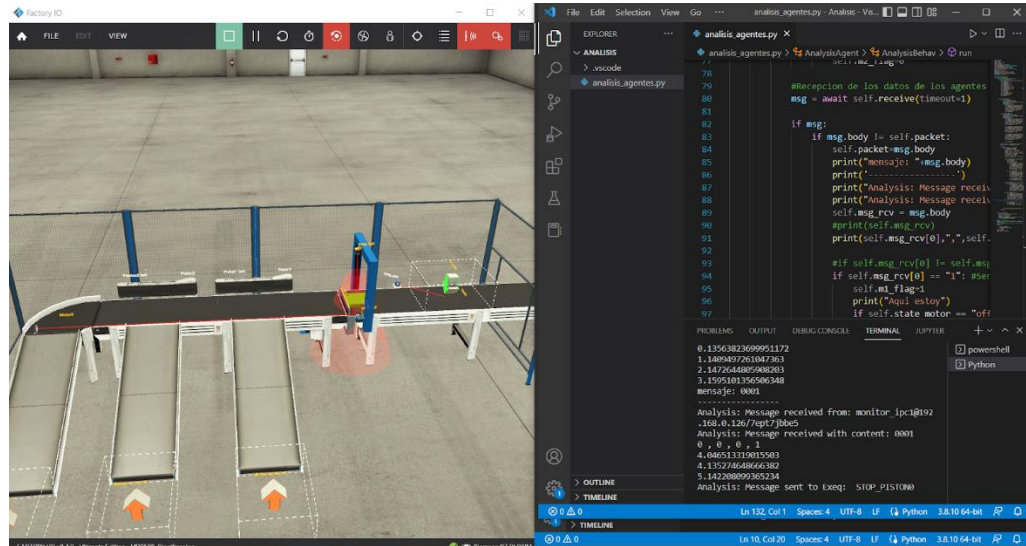


Figura 52 Agente de Análisis (AA)

Elaborado por: Investigadores

## ANEXO 13.

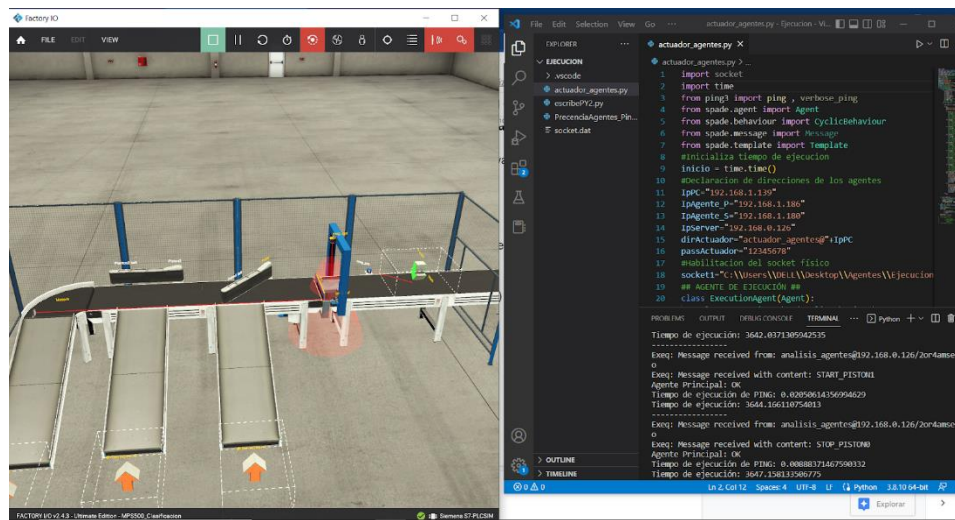
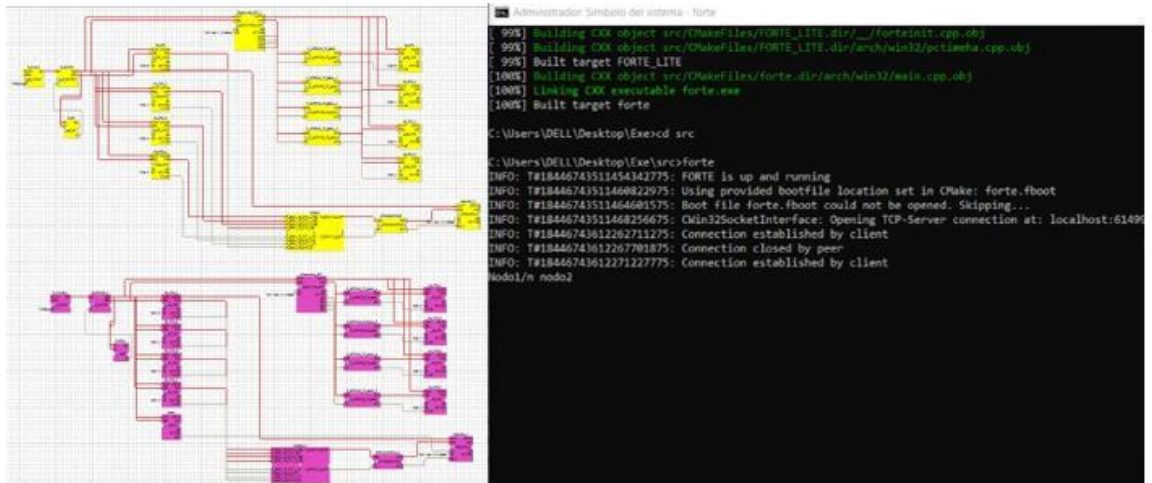


Figura 53. Agente de ejecución (AE)

Elaborado por: Investigadores

## ANEXO 14.

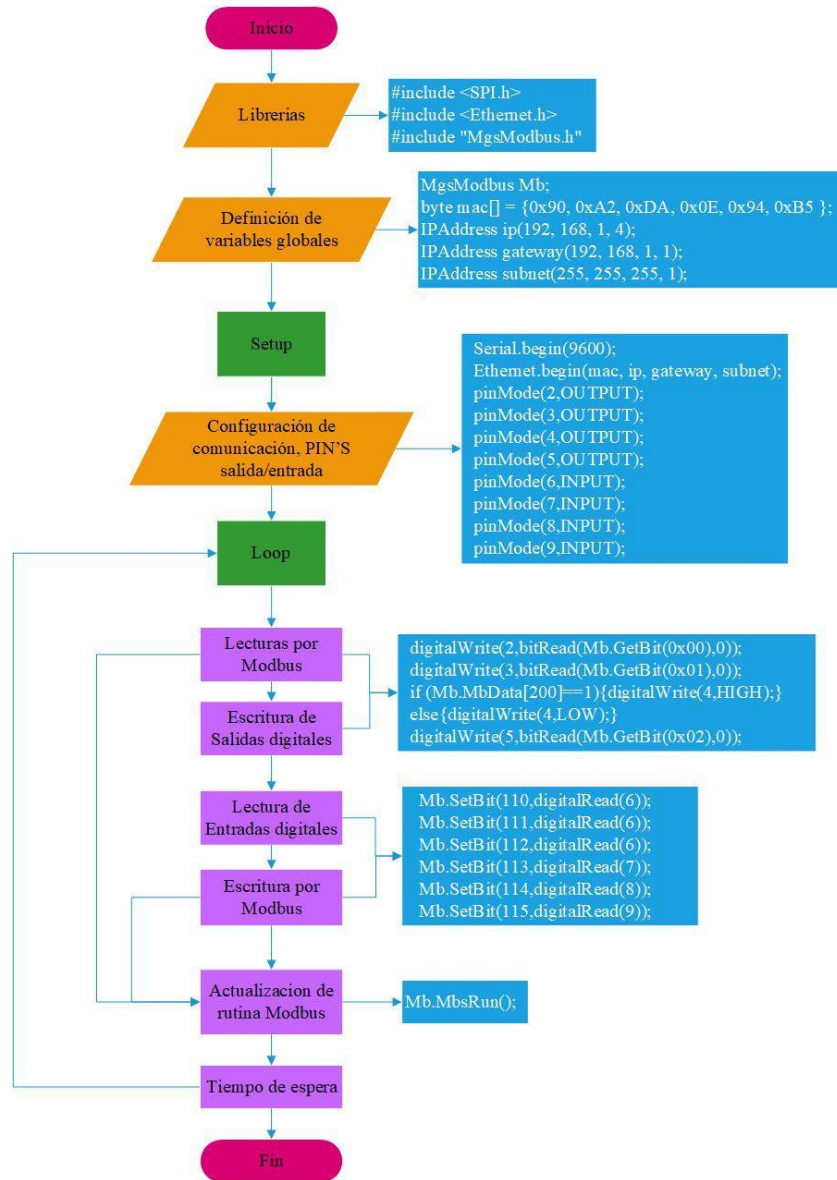


**Figura 54.** Ejecución del 4DIAC FORTE

**Elaborado por:** Investigadores

**ANEXO 15.**

Link: Programación ATMEGA328 Shield Ethernet (TESIS)



**Figura 55** Diagrama de flujo para la decodificación de las señales virtuales en señales físicas en ATMEGA328 Shield Ethernet.

**Elaborado por:** Investigadores

ANEXO 16.

Tabla 5. Registro del entradas y salidas en el IPC y 4DIAC

	Módulos	Tag	Variable	Dirección	4DIAC
Dirección de entradas físicas de los módulos digitales	Módulo 1	I00	Bool	%I0.0	"DI0.0"
		I01	Bool	%I0.1	"DI0.1"
		I02	Bool	%I0.2	"DI0.2"
		I03	Bool	%I0.3	"DI0.3"
		I04	Bool	%I0.4	"DI0.4"
		I05	Bool	%I0.5	"DI0.5"
		I06	Bool	%I0.6	"DI0.6"
	I07	Bool	%I0.7	"DI0.7"	
	Módulo 2	I10	Bool	%I1.0	"DI1.0"
		I11	Bool	%I1.1	"DI1.1"
		I12	Bool	%I1.2	"DI1.2"
		I13	Bool	%I1.3	"DI1.3"
		I14	Bool	%I1.4	"DI1.4"
		I15	Bool	%I1.5	"DI1.5"
I16		Bool	%I1.6	"DI1.6"	
I17	Bool	%I1.7	"DI1.7"		
Dirección de salidas físicas de los módulos digitales	Módulo 1	Q00	Bool	%Q0.0	'DO0.0'
		Q01	Bool	%Q0.1	'DO0.1'
		Q02	Bool	%Q0.2	'DO0.2'
		Q03	Bool	%Q0.3	'DO0.3'
		Q04	Bool	%Q0.4	'DO0.4'
		Q05	Bool	%Q0.5	'DO0.5'
		Q06	Bool	%Q0.6	'DO0.6'
	Q07	Bool	%Q0.7	'DO0.7'	
	Módulo 2	Q10	Bool	%Q1.0	'DO1.0'
		Q11	Bool	%Q1.1	'DO1.1'
		Q12	Bool	%Q1.2	'DO1.2'
		Q13	Bool	%Q1.3	'DO1.3'
		Q14	Bool	%Q1.4	'DO1.4'
		Q15	Bool	%Q1.5	'DO1.5'
Q16		Bool	%Q1.6	'DO1.6'	
Q17	Bool	%Q1.7	'DO1.7'		
Dirección de entradas físicas de los módulos Analógicos	Módulo 1	IW2	Int	%IW2	"AI0.0"
		IW4	Int	%IW4	"AI0.1"
	Módulo 2	IW6	Int	%IW6	"AI1.0"
		IW8	Int	%IW8	"AI1.1"
Dirección de salidas físicas de los módulos Analógicos	Módulo 1	QW2	Int	%QW2	"AO0.0"
		QW4	Int	%QW4	"AO0.1"