



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL

CARRERA DE INGENIERÍA INDUSTRIAL EN PROCESOS DE
AUTOMATIZACIÓN

Tema:

DESARROLLO DE UNA ARQUITECTURA DE COMUNICACIÓN DE
PROCESOS INDUSTRIALES BASADOS EN LA INDUSTRIA 4.0
MEDIANTE LA INCURSIÓN DE LOS CONTENEDORES Y EL
ESTÁNDAR IEC 61499

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado para la obtención del título de Ingeniero Industrial en Procesos de Automatización.

ÁREA: Industrial y manufactura

LÍNEAS DE INVESTIGACIÓN: Tecnología de la Información y Sistemas de Control

AUTOR: Walter Lizardo Chimborazo Chugchilan

TUTOR: Ing. Marcelo García, PHD.

Ambato – Ecuador

marzo - 2023

APROBACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Titulación con el tema: DESARROLLO DE UNA ARQUITECTURA DE COMUNICACIÓN DE PROCESOS INDUSTRIALES BASADOS EN LA INDUSTRIA 4.0 MEDIANTE LA INCURSIÓN DE LOS CONTENEDORES Y EL ESTÁNDAR IEC 61499, desarrollado bajo la modalidad Proyecto de Investigación por el señor Walter Lizardo Chimborazo Chugchilan, estudiante de la Carrera de Ingeniería Industrial en Procesos de Automatización, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado 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

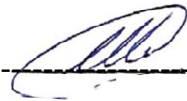
Ing. Marcelo García, PhD

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: DESARROLLO DE UNA ARQUITECTURA DE COMUNICACIÓN DE PROCESOS INDUSTRIALES BASADOS EN LA INDUSTRIA 4.0 MEDIANTE LA INCURSIÓN DE LOS CONTENEDORES Y EL ESTÁNDAR IEC 61499, es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2023



Walter Lizardo Chimborazo Chugchilan

CC: 1804953972

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y 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



Walter Lizardo Chimborazo Chugchilan

CC: 1804953972

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Walter Lizardo Chimborazo Chugchilan, estudiante de la Carrera de Ingeniería Industrial en Procesos de Automatización, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **DESARROLLO DE UNA ARQUITECTURA DE COMUNICACIÓN DE PROCESOS INDUSTRIALES BASADOS EN LA INDUSTRIA 4.0 MEDIANTE LA INCURSIÓN DE LOS CONTENEDORES Y EL ESTÁNDAR IEC 61499**, 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. Mg. Elsa Pilar Urrutia Urrutia
PRESIDENTE DEL TRIBUNAL

Ing. Elizabeth Paulina Ayala
PROFESOR CALIFICADOR

Ing. Jaime Bolívar Ruiz Banda
PROFESOR CALIFICADOR

DEDICATORIA

A mi madre, quien con su amor y apoyo constante ha fomentado en sus hijos el espíritu de superación y el valor de la humildad, el respeto, el compañerismo y la responsabilidad. Ella ha sido un ejemplo a seguir a lo largo de su vida, alcanzando pequeños y grandes logros que me han motivado a seguir adelante y a ser un ser humano mejor para mi familia y la sociedad.

A mi familia, que ha sido una constante en mi vida y que ha estado presente en cada momento de dificultad, demostrando siempre su cariño y amor sincero. Gracias por ser una fuente de apoyo inquebrantable en todo momento.

Walter Lizardo Chimborazo Chugchilan

AGRADECIMIENTO

Agradezco a Dios por haberme bendecido con el conocimiento y la sabiduría necesarios para llegar hasta este punto de mi vida y por haberme brindado la oportunidad de vivir experiencias que me han ayudado a formarme como ser humano.

A mi madre, quiero expresar mi sincera gratitud por su apoyo incondicional durante toda mi trayectoria y por tener siempre palabras de motivación para superar cualquier obstáculo que se presentara. Gracias por enseñarme valores y por darme consejos y guía que me han ayudado a alcanzar varios de mis objetivos de vida. A mi familia, amigos y docentes, agradezco sus palabras de apoyo y su presencia a lo largo de este trayecto.

También quiero agradecer al Dr. Marcelo García, mi tutor, por su valiosa ayuda con su conocimiento y experiencia, y por motivarme a desarrollar la investigación e innovación para mejorar mi formación como futuro profesional.

Walter Lizardo Chimborazo Chugchilan

ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR	II
AUTORÍA	III
APROBACIÓN DEL TRIBUNAL DE GRADO.....	V
DERECHOS DE AUTOR.....	IV
DEDICATORIA.....	VI
AGRADECIMIENTO	VII
ÍNDICE GENERAL DE CONTENIDOS.....	VIII
RESUMEN	XIII
ABSTRACT	XIV
INTRODUCCIÓN	XV
CAPITULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.2 Antecedentes Investigativos	1
1.2.1 Contextualización del problema.....	4
1.2.2 Fundamentación teórica	5
1.3 Objetivos.....	17
1.3.1 Objetivo General.....	17
1.3.2 Objetivos Específicos.....	17
CAPÍTULO II.....	18
METODOLOGÍA.....	18
2.1. Materiales	18
2.2. Métodos	18

2.2.1. Modalidad de la Investigación	18
2.2.2 Recolección de Información	19
2.2.3 Procesamiento y Análisis de Datos	20
2.2.4 Desarrollo del Proyecto	21
CAPÍTULO III	22
RESULTADOS Y DISCUSIÓN	22
3.1 Desarrollo de la propuesta	22
3.1.1 Selección del Software.....	23
3.1.1.1 Selección de la plataforma de contenedores de software	23
3.1.1.2 Selección de los protocolos de comunicación.....	25
3.1.1.3 Selección del software de control.....	27
3.1.1.4 Selección del software de ejecución.....	27
3.1.1.5 Selección del software para el desarrollo de la interfaz gráfica.....	28
3.1.2 Selección del Hardware.....	28
3.1.2.1 Selección del hardware para la adquisición/envío de datos	28
3.1.2.2 Selección del hardware para el control de los procesos.....	28
3.1.3 Desarrollo de elementos del sistema de control	29
3.1.3.1 Bloques Funcionales para la adquisición/envío de datos.....	29
3.1.3.2 Bloques Funcionales para el control de los procesos	31
3.1.3.3 Bloques Funcionales para el control de los contenedores.....	35
3.1.3.4 Creación de contenedores de los procesos	36
3.1.3.5 Diseño de la interfaz gráfica.....	37
3.1.3.6 Protocolo de comunicación UDP – Bloques Funcionales	38
3.1.3.7 Protocolo de comunicación OPC UA – Bloques Funcionales	40
3.1.4 Desarrollo del sistema de control	42

3.1.4.1	Red de dispositivos en 4Diac IDE	42
3.1.4.2	Diseño del sistema de control.....	43
3.1.5	Resultado final.....	44
3.1.5.1	Sistema de control basado en el Estándar IEC 61499 y el uso de contenedores...	44
3.2	Análisis de resultados	47
3.2.1	Análisis del tiempo de ejecución y paro de los contenedores	47
3.2.2	Análisis de los protocolos de comunicación	49
3.2.2.1	Análisis del protocolo de comunicación UDP	49
3.2.2.2	Análisis del protocolo de comunicación OPC UA	52
CAPITULO IV		54
CONCLUSIONES Y RECOMENDACIONES		54
4.1	Conclusiones.....	54
4.2	Recomendaciones.....	55
Referencias Bibliográficas		57
Anexos.....		62
Anexo 1		62
Anexo 2		64
Anexo 3		65

ÍNDICE DE TABLAS

Tabla 1.	Directrices del estándar IEC 61499.....	11
Tabla 2.	Estructura del estándar IEC 61499.....	12
Tabla 4.	Tiempos promedio de ejecución y paro de cada uno de los contenedores	49

ÍNDICE DE FIGURAS

Figura 1. Estructura del sistema de control.	23
Figura 2. Esquema del FB (Bloque Funcional) para las entradas y salidas de la IPC.	29
Figura 3. Esquema del FB compuesto para las entradas digitales de la IPC.	30
Figura 4. Diagrama de clase para la adquisición/envío de datos mediante memoria compartida.	31
Figura 5. Esquema del FB del proceso 1.	32
Figura 6. Gráfico de control de ejecución del proceso 1.	33
Figura 7. Esquema del FB del proceso 2.	34
Figura 8. Gráfico de control de ejecución del proceso 2.	35
Figura 9. Esquema del FB para el control de los contenedores.	36
Figura 10. Estructura de los contenedores en los dispositivos de control del sistema.	37
Figura 11. Interfaz Gráfica para el control y supervisión de los procesos.	38
Figura 12. Configuración del protocolo de comunicación UDP entre IPC y Raspberry Pi.	39
Figura 13. Estructura del proceso de comunicación entre las tarjetas raspberry pi y el IPC mediante protocolo UDP.	39
Figura 14. Configuración del protocolo de comunicación OPC UA entre Raspberry Pi #1 - Proceso 1 e Interfaz Gráfica.	41
Figura 15. Estructura del proceso de comunicación entre las tarjetas raspberry pi y la Interfaz Gráfica mediante protocolo OPC UA.	41
Figura 16. Red de dispositivos en el entorno de 4Diac IDE.	42
Figura 17. Diagrama de clases del sistema de control.	44
Figura 18. Estructura del sistema de control de los procesos y la adquisición/ envío de datos.	45
Figura 19. Bloques Funcionales para el control de ejecución y paro de los contenedores.	46
Figura 20. Control de los procesos.	46
Figura 21. Gráfica de los tiempos de ejecución y paro del contenedor 1.	48
Figura 22. Gráfica de los tiempos de ejecución y paro del contenedor 2.	48

Figura 23. Tráfico de datos mediante protocolo UDP entre el Computador Industrial y la Raspberry Pi B+.....	49
Figura 24. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #1.	50
Figura 25. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2.	50
Figura 26. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #1 – contenedor 1.	51
Figura 27. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2.	51
Figura 28. Tráfico de datos entre el Computador Industrial y la Raspberry Pi B+ #1 – contenedor 1 mediante el protocolo OPC UA.....	52
Figura 29. Estadísticas de jerarquía entre el Computador Industrial y la Raspberry Pi B+ #1 – contenedor 1 mediante el protocolo OPC UA.....	52
Figura 30. Tráfico de datos entre el Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2 mediante el protocolo OPC UA.....	53
Figura 31. Estadísticas de jerarquía entre el Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2 mediante el protocolo OPC UA.....	53

RESUMEN EJECUTIVO

El proyecto de investigación que se desarrolla pretende dar solución al problema de la automatización y el control de procesos en un mercado cambiante y exigente como el actual. Para ello, se está desarrollando sistemas de control basados en el estándar IEC 61499 y en la tecnología de contenedores. Estos sistemas buscan garantizar una producción eficiente y de alta calidad, reducir costos y mejorar la seguridad en el trabajo, adaptándose a las necesidades específicas de cada industria y proceso. Con el uso del estándar IEC 61499 y los contenedores se busca brindar una solución escalable y flexible para este problema.

El sistema de control desarrollado está compuesto por tres dispositivos físicos de control: el computador industrial ET 200SP, que es el encargado de la adquisición y envío de datos a través de su CPU 1515SP mediante memoria compartida, y dos tarjetas Raspberry Pi B+ que se encargan del control de los procesos mediante el uso de contenedores que poseen las aplicaciones de control correspondientes. Estos contenedores reciben y envían datos al computador industrial a través del protocolo de comunicación UDP (User Datagram Protocol). Para facilitar el control y supervisión de los procesos, se ha desarrollado una interfaz gráfica que se comunica con los contenedores que se encuentran ejecutándose en las tarjetas Raspberry Pi a través del protocolo OPC UA (Unified Architecture).

El sistema de control, combina el uso del estándar IEC 61499 y contenedores, logrando una estabilidad en el tráfico de datos entre los dispositivos, con una reducción significativa de los picos de tráfico de datos y una distribución equitativa del mismo. Esto se debe a que los dispositivos y aplicaciones están utilizando los recursos de manera eficiente y equitativa, mejorando el rendimiento de la red y reduciendo la posibilidad de cuellos de botella durante los procesos de envío, recepción y procesamiento de datos.

Palabras clave: IEC 61499, bloques funcionales, control distribuido de parámetros, memoria compartida, contenedores, computador industrial.

ABSTRACT

The research project being developed aims to provide a solution to the problem of automation and process control in today's changing and demanding market. To this end, control systems based on the IEC 61499 standard and container technology are being developed. These systems seek to guarantee efficient, high-quality production, reduce costs and improve work safety, adapting to the specific needs of each industry and process. The use of the IEC 61499 standard and containers seeks to provide a scalable and flexible solution to this problem.

The control system developed is composed of three physical control devices: the ET 200SP industrial computer, which is in charge of data acquisition and sending through its 1515SP CPU by means of shared memory, and two Raspberry Pi B+ cards that are in charge of process control through the use of containers that have the corresponding control applications. These containers receive and send data to the industrial computer through the UDP (User Datagram Protocol) communication protocol. To facilitate the control and supervision of the processes, a graphical interface has been developed that communicates with the containers running on the Raspberry Pi cards through the OPC UA (Unified Architecture) protocol.

The control system combines the use of the IEC 61499 standard and containers, achieving stability in data traffic between devices, with a significant reduction in data traffic peaks and an equitable distribution of data traffic. This is because devices and applications are using resources efficiently and equitably, improving network performance and reducing the possibility of bottlenecks during the processes of sending, receiving and processing data.

Keywords: IEC 61499, functional blocks, distributed parameter control, shared memory, containers, industrial computer.

INTRODUCCIÓN

En la actualidad, la implementación de la industria 4.0 implica la digitalización de los procesos de producción, el uso de sensores, la automatización y la integración de los sistemas de información, entre otros aspectos. Sin embargo, la integración de estos procesos y sistemas puede presentar desafíos, como la falta de compatibilidad entre diferentes tecnologías y la complejidad en la gestión de los procesos y sistemas.

La integración de contenedores y el estándar IEC 61499 pueden abordar estos desafíos, al permitir la creación de entornos de ejecución independientes y la programación modular y reutilizable de sistemas de control, respectivamente. Además, la utilización de dispositivos como la Raspberry Pi y el IPC Siemens ET200SP CPU1500S permite una mayor flexibilidad y escalabilidad en la implementación de la arquitectura de comunicación.

La arquitectura de comunicación desarrollada en esta investigación puede aplicarse en diferentes ámbitos de la producción industrial, como el control de procesos de fabricación, la supervisión y gestión de la cadena de suministro y el monitoreo de los recursos energéticos y ambientales.

En este sentido, los resultados de esta investigación pueden tener un impacto significativo en la mejora de la eficiencia y productividad de la producción industrial, a través de una arquitectura de comunicación flexible, modular y escalable, que permita la integración de diferentes procesos y sistemas en la producción industrial.

CAPITULO I

MARCO TEÓRICO

1.1 Tema de investigación

DESARROLLO DE UNA ARQUITECTURA DE COMUNICACIÓN DE PROCESOS INDUSTRIALES BASADOS EN LA INDUSTRIA 4.0 MEDIANTE LA INCURSIÓN DE LOS CONTENEDORES Y EL ESTÁNDAR IEC 61499.

1.2 Antecedentes Investigativos

El desarrollo de la parte bibliográfica de la investigación, se realizó a través del uso de diferentes bases de datos de nivel nacional como internacional, recopilando así información necesaria para el desarrollo y sustentación del tema de investigación propuesto.

En el campo de la virtualización en sistemas industriales, el uso de contenedores ha sido propuesto como una oportunidad única para desvincular el control de los diferentes procesos, lo que permite un mejor aprovechamiento del potencial de los procesos en cuanto a la distribución y transferibilidad. Según el estudio de Hofer et al. los contenedores proporcionan una mejor flexibilidad al desvincular el control de los procesos industriales y permiten una mejor ejecución dentro de los límites de tiempo críticos. Además, se establece que una arquitectura de comunicación basada en contenedores permite que las aplicaciones en contenedores puedan ejecutarse en recursos compartidos sin comprometer la ejecución dentro de los límites de tiempo determinados. Esto demuestra que el uso de contenedores en sistemas industriales es una opción prometedora y está siendo objeto de investigación en el campo de la virtualización [1].

En el campo de la automatización, la investigación realizada por Michael Sollfrank, Frieder Loch y Birgit Vogel-Heuser ha demostrado que las arquitecturas basadas en la virtualización de las aplicaciones permiten un desarrollo independiente de la plataforma y el despliegue de las aplicaciones de manera segura y aislada. Sin embargo, esta virtualización conlleva el uso de componentes de software adicionales y por lo tanto representa un esfuerzo de

procesamiento adicional. A pesar de esto, el uso de aplicaciones virtualizadas en sistemas embebidos presenta beneficios significativos con un menor esfuerzo en el desarrollo, ya que permite el uso de una multiplataforma y la compilación cruzada [2].

En el estudio realizado por Cushpa P. ha determinado que la implementación de contenedores facilita la implementación y ejecución de los sistemas de control, ofreciendo ventajas significativas en comparación con otras tecnologías de virtualización como las máquinas virtuales. El uso de contenedores permite una mayor eficiencia en el uso de la memoria de almacenamiento y procesamiento, lo que se traduce en una mayor eficiencia en la implementación y ejecución de los sistemas de control. Esto sugiere que el uso de contenedores es una opción prometedora en el campo de la automatización y es un tema de investigación relevante [3].

Un estudio realizado por Llamuca J. analizó la integración de los estándares ISA-95 e IEC-61499 para el monitoreo de sistemas de control distribuidos. La investigación concluyó que las aplicaciones desarrolladas bajo el estándar IEC-61499 ofrecen una mayor flexibilidad y eficiencia en la comunicación, lo cual reduce la carga de datos y mejora el rendimiento en entornos IoT. Además, la integración de los estándares ISA-95 e IEC-61499 mejora la interoperabilidad entre los componentes del sistema de monitoreo, lo que reduce los esfuerzos al momento de intercambiar información [4].

En una investigación realizada por García M., Irisarri y Pérez F., enfocada a la arquitectura CPPS de bajo coste mediante OPC UA y estándar IEC 61499, se estableció que dicha estructura de comunicación permite comunicaciones M2M a nivel de la planta de tal manera que se obtiene la integración en el proceso productivo de dispositivos de niveles superiores, lo cual facilita la construcción rápida y sencilla de nuevas aplicaciones distribuidas basadas en componentes [5].

Según Chicaiza F., García C., Castellanos E., Sánchez C., Rosero C. y García M., establecen que una de las metodologías para generar sistemas flexibles de automatización de fabricación se basa en la implementación de los modelos introducidos por la norma ISA-88 para generar

FBs compatibles con IEC-61499, lo cual en otras palabras permite el control de procesos discretos a través de componentes de software configurables, haciendo posible de esta manera programar dispositivos de hardware de marcas comerciales no relacionadas con la misma herramienta de software [6].

La investigación realizada por Cabral Jose, Wenger Mónica y Zoitl Alois ha demostrado que varios factores han contribuido al cambio en el enfoque de la automatización, de centralizado a modular y distribuido. Los controladores lógicos programables (PLC) han evolucionado para satisfacer las necesidades actuales y los nuevos estándares están tratando de seguir esta tendencia. El estándar IEC 61499 está diseñado para modelar soluciones industriales distribuidas con un formato independiente del proveedor. Sin embargo, este cambio también presenta nuevos desafíos, especialmente en cuanto a la implementación virtual de plantas, que puede ser más compleja cuando el sistema es distribuido e involucra diferentes módulos interconectados. Un problema importante en la implementación virtual es la interconexión entre los diferentes sistemas físicos y las herramientas de control, ya que muchos proveedores todavía ofrecen soluciones integradas cerradas. El estándar Functional Mockup Interface (FMI) busca resolver este problema proporcionando una interfaz de cosimulación que permite empaquetar modelos en una Unidad de Mockup Funcional (FMU) y permite que un algoritmo maestro de cosimulación acceda al modelo interno. Se propone una herramienta que puede exportar modelos IEC 61499 en FMUs, lo que permitiría la cosimulación de plantas físicas y el software de los PLCs que las controlan. Se presenta un experimento en el que se exporta un controlador proporcional-integral-derivativo (PID) simple de un sistema de tanques como una FMU y se cosimula junto con un sistema de tanques modelado con OpenModelica [7].

De acuerdo a Stanciu Alejandro en el campo de la computación en la nube, el Edge Computing se ha convertido en un modelo emergente para proporcionar recursos computacionales cerca de los dispositivos finales conectados a la red. Este enfoque tiene una amplia variedad de aplicaciones en áreas como Internet de las cosas, redes inteligentes, atención médica, hogar inteligente, entre otras. El estándar CEI 61499 y la plataforma

Hyperledger Fabric son soluciones relevantes en este ámbito ya que permiten la implementación de contratos inteligentes en el nivel de supervisión [8].

En cuanto a la integración con los nodos perimetrales, responsables del control del proceso, se ha propuesto una arquitectura basada en microservicios, en donde se utilizan contenedores Docker para implementar bloques de funciones y la plataforma Kubernetes para orquestar la ejecución de los mismos a través de los recursos de borde. Es importante destacar que, a pesar de los avances en el área del Edge Computing, todavía existen desafíos relevantes a ser abordados, como la escalabilidad, seguridad y la integración con otras tecnologías. Sin embargo, el potencial del Edge Computing en la mejora de la eficiencia y la eficacia en diferentes industrias es indudable, lo que lo convierte en un tema de gran interés en la investigación actual.

1.2.1 Contextualización del problema

La industria 4.0 representa una nueva era en la producción industrial que implica la digitalización de los procesos de producción y la integración de sistemas y tecnologías. La integración de diferentes tecnologías y sistemas en la producción industrial puede presentar desafíos, como la falta de compatibilidad entre diferentes tecnologías y la complejidad en la gestión de los procesos y sistemas. Por esta razón, es importante desarrollar soluciones que permitan la integración y gestión de los procesos y sistemas de la producción industrial.

El desarrollo de una arquitectura de comunicación entre tarjetas Raspberry Pi y un IPC Siemens ET 200SP CPU 1500 mediante el uso de contenedores y el estándar IEC 61499 puede abordar estos desafíos, al permitir la creación de entornos de ejecución independientes y la programación modular y reutilizable de sistemas de control. La Raspberry Pi es una tarjeta de computadora de bajo costo y alto rendimiento que puede utilizarse para controlar y monitorear sistemas en tiempo real. Por otro lado, el IPC Siemens ET 200SP CPU 1500 es un controlador programable de automatización de fábricas con capacidad para monitorear y controlar procesos de producción [9].

La utilización de contenedores y el estándar IEC 61499 permiten la programación modular y reutilizable de sistemas de control, lo que facilita la integración y gestión de diferentes procesos y sistemas en la producción industrial. Los contenedores permiten la creación de entornos de ejecución independientes que pueden ejecutar aplicaciones de manera aislada, lo que mejora la seguridad y el aislamiento de los procesos. Por su parte, el estándar IEC 61499 define un modelo de programación para la automatización de fábricas basado en la modularidad y la reutilización de componentes [10].

La arquitectura de comunicación desarrollada en esta investigación tiene como objetivo permitir la integración de diferentes procesos y sistemas en la producción industrial, mejorando la eficiencia y productividad de la misma. La arquitectura de comunicación se basa en la utilización de contenedores y el estándar IEC 61499, lo que permite la creación de entornos de ejecución independientes y la programación modular y reutilizable de sistemas de control, lo que facilita la integración y gestión de diferentes procesos y sistemas en la producción industrial [11].

La arquitectura de comunicación desarrollada en esta investigación puede aplicarse en diferentes ámbitos de la producción industrial, como el control de procesos de fabricación, la supervisión y gestión de la cadena de suministro y el monitoreo de los recursos energéticos y ambientales. La implementación de esta arquitectura de comunicación en la producción industrial puede mejorar la eficiencia y productividad de los procesos, reducir los costos y mejorar la calidad de los productos finales. Por lo tanto, la investigación sobre la arquitectura de comunicación entre tarjetas Raspberry Pi y un IPC Siemens ET 200SP CPU 1500 mediante el uso de contenedores y el estándar IEC 61499 es de gran relevancia en el contexto actual de la producción industrial [12].

1.2.2 Fundamentación teórica

Industria 4.0

La Industria 4.0 es un concepto que nace en Alemania a inicios del año 2010, de un grupo de investigadores cuyo propósito se basaba en el diseño de un programa que permitiese mejorar la productividad de la industria manufacturera. En el año 2013 el grupo de investigadores el

termino de Industria 4.0 se convirtió en el eje central del Plan estratégico de innovación de Alemania [13].

La industria 4.0 se basa en el desarrollo de la manufactura computarizada debido a la rápida evolución a nivel computacional y de conectividad, lo cual permite un control en tiempo real de los diferentes procesos [14].

Control de procesos

La automatización industrial tiene como objetivo facilitar al operador el desarrollo de actividades repetitivas y de considerable exigencia, de tal manera que se obtenga una mayor producción a menor inversión, de esta manera el proceso de fabricación se vuelve evolutivo y flexible a gracias a los elementos tecnológicos e informáticos. De esta manera la automatización permite optimizar los procesos productivos de una manera eficiente y eficaz.

Alrededor del mundo los procesos de manufactura se llevaban a cabo de forma manual por un operador. El operador controlaba el proceso y sus ajustes en base a las instrucciones de operación y su conocimiento sobre el proceso [15].

Controladores Locales

Los controladores locales surgen de la ineficacia a la hora de actuar frente a los problemas que se presentaban en los procesos, con la finalidad de obtener productos de mejor calidad. Este tipo de controladores facilitaban al operador el control de dos o más Lazos del proceso, no obstante, los controladores debían ser distribuidos alrededor de toda la planta de producción, lo cual generaba pérdidas de tiempo al momento de establecer los parámetros de ajustes [16].

IPC (Computador Industrial) ET 200SP Siemens

El IPC (Computador Industrial) es la evaluación de los computadores de oficina ya que este tiene la posibilidad de trabajar en ambientes industriales de una manera más eficaz y segura, por lo general este tipo de computadores trabajan con conectividad de tipo Ethernet

permitiendo la comunicación con módulos para el intercambio de datos con los diferentes dispositivos de campo [17] [18].

La CPU ET 200SP de Siemens es una unidad central de procesamiento (CPU) para sistemas de automatización SIMATIC S7-1200 y S7-1500. Es una CPU compacta y de alta disponibilidad, con un alto rendimiento y una gran capacidad de procesamiento. Es compatible con las plataformas SIMATIC S7-1200 y S7-1500, lo que permite una fácil integración en un sistema de automatización existente. Está equipada con un procesador Intel de 32 bits que proporciona un rendimiento rápido y fiable con una velocidad de hasta 1.2 GHz [19].

Además, cuenta con una gran capacidad de memoria RAM de hasta 4 GB y una memoria de datos integrada de hasta 2 GB, lo que permite almacenar grandes cantidades de datos y programas. Cuenta con entradas y salidas digitales y analógicas, lo que permite la conexión de sensores y actuadores para la automatización de procesos. También tiene una interfaz Ethernet integrada para conectividad de red y soporta protocolos de comunicación estándar como PROFINET, PROFIBUS, y Modbus TCP/IP [20].

Raspberry Pi B+

La Raspberry Pi B+ tiene un procesador de cuatro núcleos a 1 GHz, 1 GB de RAM y una ranura para tarjetas microSD. Tiene un conector HDMI para conectarse a un monitor o televisor, así como un conector para auriculares o altavoces. También cuenta con un puerto Ethernet de 10/100, cuatro puertos USB 2.0, un puerto de video compuesto, un conector de alimentación micro USB y una conexión GPIO de 40 pines. La tarjeta también tiene una ranura para tarjetas microSD, donde se almacena el sistema operativo y los datos.

Esta tarjeta es compatible con una amplia variedad de sistemas operativos, incluyendo Raspbian, Ubuntu, Windows 10 IoT Core, entre otros. Esto permite a los usuarios personalizar y optimizar su configuración según sus necesidades específicas. Además, existen una gran cantidad de proyectos y tutoriales disponibles en línea para ayudar a los usuarios a sacar el máximo provecho de su Raspberry Pi B+ [21].

Sistemas de supervisión centralizados

Los sistemas de supervisión centralizados permitían obtener un nivel de seguridad alto de tal manera que evitaban que una falla paralice todo el sistema. De esta manera se implementaron los Controladores Analógicos de forma directa con el proceso, optimizando el control de la variable del proceso. Por lo cual estos controladores son los elementos que controlan el proceso, permitiendo al ordenador central cambiar los puntos de consigna, en otras palabras, el valor de referencia con el que se compara la variable controlada de tal manera que se optimiza a la misma [22].

Sistemas de control industrial

Los sistemas de control hacen referencia a una variedad de sistemas compuestos por computadores, elementos eléctricos, hidráulicos y mecánicos, que se encuentran dentro de un proceso, que en gran parte del mismo es supervisado por un operador. Este tipo de sistemas permiten tener procesos automatizados o semiautomatizados en las diferentes industrias tales como la farmacéutica, química, alimenticia, automotriz y muchas otras.

Los sistemas de control industrial comprenden desde la implementación de sistemas de supervisión y adquisición de datos SCADA hasta la implementación de controladores lógicos programables PLC (Controlador Lógico Programable), los mismos que se encuentran presentes en gran parte de las industrias y procesos críticos de manufactura [23], [24].

Protocolo de comunicación de redes industriales

Existen tendencias tecnológicas que van desde el internet de las cosas inteligencia artificial, robótica que influyen en la transformación de la industria dando paso a la denominada 4ta revolución industrial o industria 4.0. Donde existe la comunicación M2M que permite el intercambio de información entre las máquinas con sistemas de control distribuido o bien el internet de las cosas que ya no solo se basa en la comunicación de máquinas sino también

con clientes o proveedores, dicha comunicación se ha logrado gracias a los protocolos que pueden ser privados o abiertos como por ejemplo MQTT, OPC UA, aquellos basados en Ethernet IP, entre otros [25], [26].

Hay dos tipos de protocolos de comunicación: cliente/servidor y publicación/suscripción. El protocolo cliente/servidor requiere que el cliente se conecte al servidor y realice una solicitud. Los protocolos de publicación/suscripción, por otro lado, requieren elementos informáticos, eléctricos, hidráulicos y mecánicos dentro de un proceso de dispositivo para conectarse a un "tema" de un administrador intermedio y publicar información [14].

Existen varios protocolos disponibles en el mercado, algunos de ellos son propietarios mientras que otros son estándares abiertos. Estos protocolos tienen la capacidad de conectar equipos industriales con diferentes plataformas, permitiendo la interoperabilidad y la comunicación entre dispositivos y sistemas. Algunos de los protocolos más conocidos y utilizados en la industria incluyen:

UDP (User Datagram Protocol)

Es un protocolo de red de la capa de transporte del modelo OSI que permite el envío de mensajes denominados datagramas sin garantía de entrega ni de ordenación. Esto significa que el protocolo UDP no realiza ningún tipo de control sobre la entrega de los datagramas ni sobre el orden en que llegan a su destino.

UDP es utilizado en aplicaciones que no requieren una entrega garantizada de los mensajes, sino que se basan en la rapidez de transmisión de estos. Al no realizar ningún tipo de control sobre la entrega de los mensajes, UDP es más rápido que otros protocolos de la capa de transporte como TCP (Transmission Control Protocol).

UDP es utilizado en aplicaciones que necesitan enviar grandes cantidades de datos a altas velocidades, como en el caso de transmisiones de audio y vídeo en tiempo real. También es utilizado en aplicaciones de red de tiempo real, como el protocolo de juegos en red [27].

OPC UA (Unified Architecture)

Es una plataforma abierta y gratuita diseñada para facilitar la interoperabilidad y el intercambio de datos entre dispositivos conectados a una misma red, especialmente en entornos de automatización industrial. Esta plataforma ha sido desarrollada para contribuir a los avances tecnológicos que han surgido con la Revolución Industrial y ofrecer una forma óptima de integrar sistemas automatizados.

OPC UA es una opción ideal para instituciones interesadas en implementar un sistema automatizado moderno, ya que permite una fácil integración entre sistemas tradicionales y sistemas modernos. Además, al ser una plataforma abierta y gratuita, está disponible para cualquier institución interesada en implementarla [28].

Estándar IEC 61499

El estándar IEC 61499 se establece como una arquitectura de referencia para el diseño de sistemas de control ya que facilita el desarrollo de los mismos a través una lógica descentralizada, basada en una arquitectura basada en bloques funcionales (FB) que se establecen como el pilar fundamental para la estructura básica de los modelos de control. De esta manera el modelo de control se caracteriza por sus entradas, salidas y funciones internas, de la misma manera en la que trabajan los sistemas de control clásicos. Por consiguiente, por medio de la interconexión de bloques de funciones simples se puede conseguir modelar aplicaciones y modelos de control complejos. El estándar aprovecha el conocimiento que poseen los profesionales a través de esta estructura de control [29] [17].

El estándar ejecuta los FB en el momento que los mismos reciben una señal de entrada, permaneciendo así en un estado de reposo hasta que sea necesaria su ejecución, a diferencia de los componentes de la IEC 61131, en donde las subrutinas se verifican de forma periódica. Esta metodología facilita la portabilidad y ejecución de los FB, ya que cada FB posee su propio contexto y es ejecutado ante una señal de entrada, comportándose como una unidad independiente ante las demás.

De esta manera el estándar pretende conseguir un nivel alto de control distribuido, ya que puede ser utilizada para descentralizar la lógica de control, de tal manera que se implantan controles holónicos o multiagente, incrementando la flexibilidad de los sistemas. Además, debido a la creciente importancia del software en los sistemas de control hace que la reutilización del código se establezca como una actividad fundamental, en otras palabras, se podría diseñar un mismo sistema de control para dos máquinas de diferentes fabricantes [30].

Por lo cual el estándar IEC 61499 tiene como objetivo establecer una guía para el diseño de sistemas de control industriales distribuidos, modulares y flexibles. Basándose en las siguientes directrices:

Tabla 1. Directrices del estándar IEC 61499.

Directrices del estándar IEC 61499	
Directrices	Descripción
Portabilidad	Es la facilidad para intercambiar las aplicaciones de control y elementos de biblioteca entre diferentes softwares
Configurabilidad	Es la capacidad de configurar dispositivos de control por diversos softwares.
Interoperabilidad	Permite estandarizar e intercambiar datos entre diferentes dispositivos de control.

Fuente: [3] [4].

El cumplimiento de estas directrices se logra mediante la provisión de diversos modelos de referencia especificados en el estándar IEC 61499, publicados en su primera edición en 2005 y en su segunda edición en 2012.

Estas especificaciones dotan al antiguo estándar IEC 61131 de soporte directo para la distribución de información y la flexibilidad de los sistemas de control. En los sistemas de control basados en PLC, la ejecución de los bloques de función (FB) es cíclica. En el estándar IEC 61499, la mayor parte de la ejecución de los sistemas de control se basa en eventos,

permitiendo la ejecución asíncrona de bloques de funciones y redes de bloques de funciones mediante el uso de un control adecuado de eventos especiales. No obstante, el concepto sincrónico aún se admite [30].

Estructura de los sistemas de control según IEC 61499

La estructura de aplicación se basa en un modelo basado en bloques de funciones que realizan funciones de control por separado de cualquier problema de implementación específico. De esta manera la estructura se distribuye en cuatro partes, cada una independiente de la otra y con objetivos diferentes, como se puede observar a continuación:

Tabla 2. Estructura del estándar IEC 61499.

Estructura del estándar	
Partes	Descripción
Parte 1	Establece el diseño de referencia para el desarrollo, reutilización e implementación en los bloques de funciones. Además, se definen los conceptos de bloque funcional básico, bloque funcional compuesto, aplicación, interfaz, dato, señal, etc.
Parte 2	Establece los requerimientos a cumplir por las herramientas de desarrollo para poder establecer el diseño propuesto en la Parte 1. No obstante, los requerimientos no garantizan la interoperabilidad, portabilidad y flexibilidad del sistema de control. Además, se definen los esquemas XML para el intercambio de fichas de información entre las distintas de herramientas.
Parte 3	Muestra las funcionalidades y ejemplos de aplicación del estándar en un amplio conjunto de dominios mediante contenido didáctico.
Parte 4	Establece los perfiles de complicación de tal manera que facilite el cumplimiento de los objetivos del estándar IEC 61499 en los dispositivos compatibles y las herramientas de desarrollo.

Fuente: [31].

Modelo – Implementación del estándar IEC 61499

El estándar IEC 61499 se puede considerar como una manera para diseñar sistemas de control por medio de los bloques de funciones, ya que permite el encapsulamiento de otros FB y la visualización de las interconexiones entre componentes. Lo cual permite observar el sistema de control desde diferentes puntos de vista según la necesidad. Además, la arquitectura se establece como la implementación del mismo sistema de control, siendo implementado y ejecutado de forma directa, lo cual resulta confuso al momento de implementar el estándar [32].

Estándar IEC 61499 – Modelos

El estándar IEC 61499 es un estándar internacional para la programación de sistemas distribuidos y sistemas de control basados en eventos. Este estándar define un marco para la representación de los componentes y funciones de un sistema de control.

El modelo de sistema es un modelo de alto nivel que describe la estructura y el comportamiento de un sistema de control en su conjunto. Este modelo incluye información sobre los componentes del sistema, incluidos los dispositivos, recursos y aplicaciones.

El modelo de dispositivo describe un dispositivo individual en un sistema de control, como un controlador programable (PLC) o un dispositivo de entrada/salida (I/O). Este modelo incluye información sobre las características físicas y técnicas del dispositivo, así como sobre cómo se conecta y comunica con otros componentes del sistema.

El modelo de recurso describe los recursos disponibles en un sistema de control, como las variables y las estructuras de datos. Este modelo se utiliza para representar los datos que se utilizan en el sistema y para describir cómo se accede y se utiliza esa información.

El modelo de aplicación describe las funciones y tareas que se realizan en un sistema de control. Este modelo se utiliza para describir cómo se organizan y se ejecutan las funciones y tareas en el sistema, y cómo se utilizan los recursos del sistema para realizar esas funciones.

El modelo de bloque funcional es un modelo de bajo nivel que describe las funciones individuales que realiza un sistema de control. Este modelo incluye información sobre las entradas, las salidas y el procesamiento de cada función. Cada bloque funcional se puede representar como una estructura de datos que contiene información sobre sus entradas, salidas y procesamiento [33].

4DIAC IDE

Este software es una herramienta compuesta por un conjunto de elementos desarrollados por el entorno integrado IDE de la compañía Eclipse. 4DIAC es uno de los softwares de programación industrial basados en el estándar IEC 61499 con el objetivo de proporcionar un estándar abierto y gratuito para la automatización, con la finalidad de promover la investigación. Presenta una interfaz de ingeniería extensible en base al estándar IEC 61499 para sistemas de control distribuido, los mismos pueden ser descargados en dispositivos de campo capaces de soportar dicho estándar y además facilita la integración de diferentes plugins [32].

FORTE

Es una aplicación portátil con entorno de ejecución para sistemas desarrollado por 4DIAC basado en el estándar IEC 61499, está enfocada principalmente a dispositivos pequeños de 16 o 32 bits, está basada en C++ y tiene capacidades de comunicación flexibles. Puede ser ejecutada en sistemas reducidos embebidos y de tiempo real porque consta con los componentes adecuados para que pueda adaptarse a ellos. Además, es independiente a la plataforma donde se esté ejecutando por ello puede ser ejecutada en distintos sistemas operativos como POSIX, Windows 32, eCos, entre otros. Adicionalmente, permite el desarrollo de una arquitectura extensible a la medida del diseño de la aplicación [32].

CMake

Permite el uso de herramientas de plataformas de código abierto como ITK y VTK ofrece una gran cantidad de ventajas a los desarrolladores de software. Estas herramientas permiten

la creación, experimentación y empaquetamiento de software de manera fácil y eficiente. Además, el proceso de compilación se controla mediante una plataforma sencilla y configuraciones autónomas al compilador, lo que permite un mayor grado de flexibilidad.

Una de las principales ventajas de estas herramientas es que generan archivos de trabajo que pueden ser utilizados en cualquier entorno del compilador que el desarrollador elija. Esto significa que el software puede ser compilado y ejecutado en una amplia variedad de plataformas, lo que aumenta su alcance y accesibilidad [32].

Contenedores de software

Es una tecnología de virtualización, el entorno virtual que se ejecuta sobre el kernel del sistema operativo (anfitrión o base), donde utiliza recursos básicos que sean comunes para los diferentes sistemas operativos soportados, por ello se puede ejecutar en diferentes SO, en otras palabras, no se despliega un SO completo, solo aquellos recursos que son comunes entre los SO heterogéneos [34].

Docker

Es una arquitectura cliente-servidor. El cliente de Docker se comunica con el dominio de Docker, que es responsable de crear, ejecutar y distribuir contenedores, y todos pueden ejecutarse en el mismo sistema, o puede conectar el cliente a un dominio de Docker remoto, y el cliente de Docker y el dominio hablan entre sí. Se comunican a través de sockets o RESTfull API. El dominio de Docker crea contenedores a partir de imágenes, que pueden ser locales o estar en el Registro de Docker [32].

Docker utiliza funciones clave del kernel de Linux para aislar procesos entre sí y, gracias a su propio entorno de ejecución runC, permite que múltiples aplicaciones se ejecuten en paralelo en contenedores aislados sin desperdiciar recursos de la máquina. Con esto, esta plataforma liviana se consolida como una alternativa a la virtualización basada en hipervisor. Docker es una plataforma de contenedores que permite a los usuarios empaquetar, desplegar y ejecutar aplicaciones de manera rápida y fácil en cualquier entorno. Una de las principales

ventajas de Docker es su compatibilidad con varios sistemas operativos y plataformas en la nube. Además, con Swarm y Compose, la plataforma Docker ya proporciona herramientas de administración de clústeres para ayudar a los usuarios a escalar y administrar sus aplicaciones.

Además, con Docker Hub, los usuarios pueden tener un registro central de recursos para almacenar y compartir sus contenedores con otros. El creciente ecosistema de Docker proporciona a sus usuarios una variedad de herramientas, complementos y componentes de infraestructura para ayudarles a sacar el máximo provecho de la plataforma. Sin embargo, una desventaja de Docker es que solo admite su propio formato de contenedor, lo que puede limitar la capacidad de los usuarios para utilizar contenedores de otros proveedores. Además, el software se proporciona como un archivo monolítico que contiene todas sus funciones, lo que puede ser un problema para los usuarios que deseen personalizar o modificar su configuración [18] [32].

1.3 Objetivos

1.3.1 Objetivo General

- Desarrollar una arquitectura de comunicación de procesos industriales basados en la Industria 4.0 mediante la incursión de los contenedores y el estándar IEC 61499.

1.3.2 Objetivos Específicos

- Analizar los diferentes contenedores de software y protocolos de comunicación utilizados en la Industria 4.0.
- Diseño de la arquitectura de comunicación de procesos industriales en base a contenedores para el envío y recepción de datos.
- Implementar el módulo de comunicación de control industrial para la comprobación del funcionamiento de la arquitectura propuesta.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

La metodología utilizada en el presente proyecto requirió la utilización de diferentes materiales, como artículos publicados en revistas científicas, proyectos de investigación relacionados con el tema principal, libros y otra información sobre sistemas que utilizan tecnología similar al presente proyecto. Para el hardware del sistema, se utilizaron dispositivos y equipos específicos como el IPC SIEMENS ET200SP que cuenta con un controlador de la serie 1500 y tarjetas Raspberry B+. En cuanto al software, se utilizaron aplicaciones como 4Diac IDE, Forte, Wireshark, Factory I/O y TIA PORTAL para la configuración de los equipos y para la programación del desarrollo del sistema. Todos estos materiales y fuentes de información fueron esenciales para el desarrollo de la metodología y el éxito del proyecto.

2.2. Métodos

2.2.1. Modalidad de la Investigación

El objetivo del proyecto fue investigar y desarrollar una solución innovadora para mejorar la comunicación y la virtualización de procesos industriales en la Industria 4.0. El proyecto se realizó en base al uso de diferentes técnicas de investigación y desarrollo, como la revisión de la literatura, el análisis de datos y la experimentación. Estas técnicas permitieron identificar las necesidades y desafíos del mercado y desarrollar un prototipo eficiente. Además, el proyecto también incluyó pruebas y validaciones para garantizar la calidad y confiabilidad del prototipo final. Se aseguró de que el prototipo cumpla con las necesidades del mercado y sea fiable antes de su implementación.

Investigación de campo

Se investigó el comportamiento de la máquina industrial IPC mediante pruebas de funcionamiento en el campo, con el objetivo de establecer la relación entre el estándar IEC 61499 y los sistemas de control distribuidos utilizando contenedores. Determinando la

metodología para manejar los datos en tiempo real y evaluando el comportamiento y las complicaciones potenciales durante el desarrollo del proyecto.

El estudio de esta relación permitió identificar las ventajas y desventajas de utilizar estos elementos de automatización en la Industria 4.0, estableciendo si la propuesta es eficiente y cómo puede ser implementada en la industria.

Investigación bibliográfica

Se realizó una investigación bibliográfica para la recolección de información relevante para el proyecto propuesto, buscando en diferentes bases de datos académicas nacionales e internacionales. Una vez recolectada la información, se evaluó la factibilidad de utilizar contenedores en los sistemas de control distribuidos basado en el estándar IEC 61499. Este estudio permitió conocer el estado actual y futuro de las industrias en el país, y cómo la implementación de contenedores en los sistemas de control distribuidos basado en el estándar IEC 61499 puede impactar en la industria. Estableciendo la viabilidad y cómo se podría implementar el proyecto desarrollado en la industria.

Investigación cuantitativa

Se realizó una investigación mediante el desarrollo de una gran variedad de pruebas, con el objetivo de crear un sistema de control industrial basado en el estándar IEC 61499 que pueda ser utilizado para monitorear y supervisar una gran variedad de procesos. El objetivo de estas pruebas fue validar las ventajas del sistema desarrollado en relación a las características necesarias para alcanzar la Industria 4.0. Estableciendo de esta manera si el sistema cumple con las necesidades y si es viable para su implementación en la industria.

2.2.2 Recolección de Información

Para recolectar información se utilizaron diversas fuentes como documentos, revistas, libros y proyectos relacionados con la virtualización de procesos industriales y los protocolos de comunicación. Se buscó utilizar bases de datos confiables y actualizadas para obtener información relevante sobre estos temas y se tuvo en cuenta normas y estándares basados en

la Industria 4.0 de los últimos años. Se llevó a cabo una investigación bibliográfica cuidadosa y se seleccionaron solo fuentes de confianza para garantizar la calidad y precisión de los datos recopilados.

Además, se estableció que los datos del funcionamiento del prototipo desarrollado serán recolectados durante una jornada de 4 horas con todos los dispositivos y procesos funcionando de forma automática mediante el software Wireshark, de tal manera que cualquier problema o falla sea registrado. La información recopilada se utilizó para desarrollar el prototipo y para validar su efectividad y confiabilidad. Se aseguró que el prototipo cumpla con las necesidades y sea fiable antes de su implementación.

2.2.3 Procesamiento y Análisis de Datos

Una vez recopilada y seleccionada la información apropiada para este proyecto a través de la investigación planteada, se llevó a cabo un proceso analítico para evaluar y procesar dicha información. Este proceso analítico incluyó diversas actividades, como:

- Revisión y organización de la información recopilada: para facilitar su análisis y comprensión.
- Síntesis de la información: a partir de la revisión y organización de la información.
- Análisis de la información: una vez sintetizada la información generada por Wireshark, se realizarán diferentes tipos de análisis para determinar si los protocolos de comunicación funcionaron correctamente, prestando especial atención a cualquier problema o falla documentada durante la prueba.
- Presentación de resultados: finalmente, se interpretarán los resultados de la prueba para determinar si los protocolos de comunicación son adecuados para el sistema de control y si se necesitan ajustes o mejoras. Se identificarán los problemas y se propondrán soluciones para mejorar el rendimiento de los protocolos.

2.2.4 Desarrollo del Proyecto

A continuación, se describen a detalle las actividades realizadas para el cumplimiento del proyecto de investigación:

1. Recolección de información relacionada con la virtualización de los sistemas y los diferentes de protocolos de comunicación utilizados dentro de la Industria 4.0.
2. Análisis de los contenedores dentro de la virtualización de procesos industriales.
3. Descripción y selección del contenedor adecuado para la virtualización de procesos industriales.
4. Análisis de protocolos de comunicación implementados dentro de la Industria 4.0.
5. Establecer el protocolo de comunicación adecuado para la virtualización de procesos industriales en base al estándar IEC 61499.
6. Desarrollar la arquitectura de comunicación y virtualización mediante el protocolo de comunicación seleccionado y el estándar IEC 61499 de acuerdo a las especificaciones de la Industria 4.0.
7. Realización de pruebas de funcionamiento de la arquitectura de comunicación y virtualización.
8. Virtualización del proceso seleccionado.
9. Validación de la arquitectura planteado de acuerdo a los datos obtenidos a través de las pruebas de funcionamiento.
10. Elaboración del Informe Final.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1 Desarrollo de la propuesta

La investigación desarrolla una arquitectura de comunicación y virtualización de procesos industriales, mediante la creación de aplicaciones utilizadas como runtimes llamadas “FORTE”, en las cuales se integran la configuración, comunicación y archivos necesarios para relacionar los diferentes dispositivos utilizados, todos estos aspectos se diseñan en el software industrial 4DIAC IDE, el cual se basa en el estándar IEC 61499, a través del lenguaje de programación por FBs (Bloques de Funciones), adicionalmente se crean contenedores en los cuales se les agrega un directorio con el runtime FORTE del proceso virtualizado.

El sistema se desarrolla mediante la adquisición/envío de datos de los procesos mediante el IPC (Computador Industrial), para lo cual se utilizan librerías y programación en base a C++, para que a través del uso de un archivo DLL del software SIMATIC ODK 1500s adquirir los datos mediante memoria compartida. Los datos adquiridos por el runtime Forte encargado de la adquisición/ envío son enviados a través del protocolo de comunicación UDP. El protocolo de comunicación UDP es utilizado para la comunicación entre los dispositivos de control que son el IPC (Computador Industrial) y las Raspberry Pi, los cuales son utilizados para la adquisición/envío de datos y el control de los procesos. Por otro lado, se utiliza el protocolo de comunicación OPC UA para el envío de datos desde las Raspberry Pi a la Interfaz Gráfica y viceversa.

Los procesos se virtualizan mediante el uso de contenedores Docker y son almacenados en un repositorio local en dos tarjetas Raspberry Pi de tal manera que, si un dispositivo deja de funcionar, el otro dispositivo se encarga del control del proceso, obteniendo de esta manera procesos productivos muy flexibles, eficaces y con una respuesta rápida ante fallas o problemas del dispositivo de control.

La Interfaz Gráfica se desarrolla mediante el software Labview, dicha interfaz permite el control y supervisión de los procesos, presenta la visualización del estado del proceso, el

estado de los sensores y con contadores de la cantidad de ítems producidos por cada proceso. Además, permite el inicio y paro de cada uno de los procesos ya que consta de controles de Start y Stop, de esta manera el sistema que se desarrolla presenta la estructura mostrada en la Figura 1.

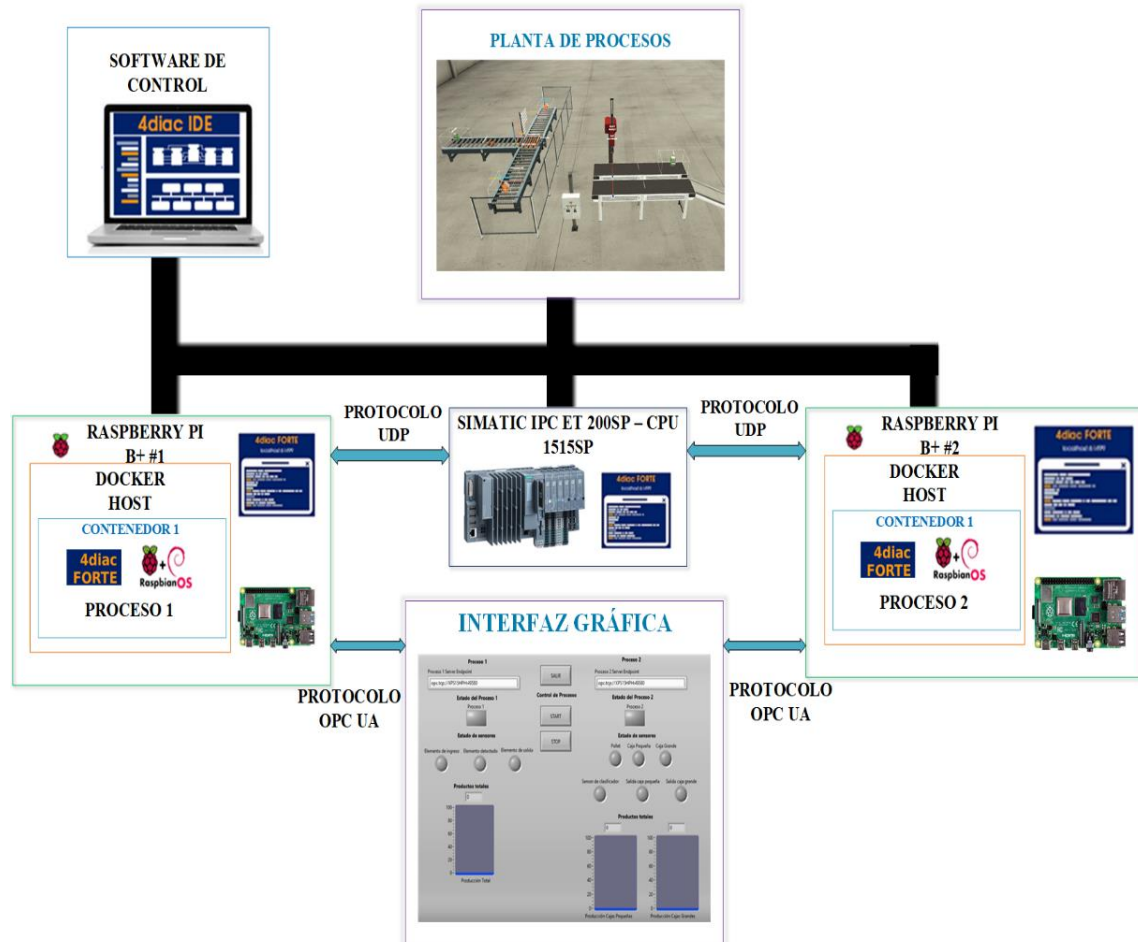


Figura 1. Estructura del sistema de control.

3.1.1 Selección del Software

3.1.1.1 Selección de la plataforma de contenedores de software

Los contenedores de software son una forma eficiente y portátil de empaquetar y distribuir aplicaciones. Existen varios tipos de software de contenedores disponibles en el mercado, cada uno con sus propias fortalezas y debilidades. Algunos de los más populares son Docker, LXC, rkt, OpenVZ y Windows Containers.

Docker es uno de los contenedores más populares y ampliamente utilizados en la industria. Permite empaquetar y distribuir aplicaciones en contenedores aislados, lo que permite una ejecución consistente y reproducible en diferentes entornos. Además, Docker es de código abierto y cuenta con una gran comunidad de desarrolladores y usuarios que contribuyen y comparten soluciones y mejoras, lo que brinda una amplia variedad de recursos y herramientas disponibles para los usuarios de Docker.

LXC (Linux Containers) es un proyecto de contenedores de código abierto que proporciona un sistema de contenedores para el kernel de Linux. Es una opción popular para aquellos que buscan una solución de contenedores ligera y eficiente en términos de recursos. Aunque no cuenta con la misma cantidad de recursos y herramientas que Docker, LXC es una excelente opción para aquellos que buscan una solución de contenedores más simple y eficiente en términos de recursos.

rkt (Rocket) es un contenedor de código abierto desarrollado por CoreOS. Se destaca por su enfoque en la seguridad y la privacidad de los datos de las aplicaciones empaquetadas. Aunque no es tan popular como Docker, rkt es una excelente opción para aquellos que buscan una solución de contenedores segura y privada.

OpenVZ es una plataforma de virtualización basada en contenedores que proporciona un sistema de contenedores completo para servidores. Aunque no es tan popular como Docker, OpenVZ es una excelente opción para aquellos que buscan una solución de contenedores más avanzada y completa para servidores.

Windows Containers es una tecnología de contenedores de Microsoft que permite ejecutar aplicaciones en contenedores aislados en sistemas Windows. Aunque no es tan popular como Docker, Windows Containers es una excelente opción para aquellos que buscan una solución de contenedores en sistemas Windows.

De esta manera la plataforma de contenedores de software que se utiliza es Docker el cual es un software que se utiliza a menudo en el desarrollo de aplicaciones y sistemas de control. La característica principal por la que se utiliza Docker es porque permite la automatización y creación de nuevos contenedores en tiempos cortos, lo que permite que las instancias se inicien rápidamente en cuestión de segundos. Los contenedores Docker se caracterizan por ocupar espacios reducidos en comparación con otros tipos de software de contenedores, esto permite un mejor rendimiento y un menor consumo de recursos en comparación con la virtualización tradicional.

Además, los contenedores creados en Docker permiten la fácil portabilidad de las aplicaciones y sistemas de control, ya que los contenedores pueden ser fácilmente transportados de un dispositivo a otro sin tener que preocuparse por la configuración o dependencias del sistema operativo. Esto hace que sea muy útil en entornos donde se necesita implementar sistemas de control en diferentes equipos y plataformas.

3.1.1.2 Selección de los protocolos de comunicación

La industria 4.0 es un término que se refiere a la cuarta revolución industrial, caracterizada por la automatización, la interconexión y la inteligencia artificial en la fabricación y la producción. Una de las claves para lograr esta revolución es la capacidad de los dispositivos y sistemas para comunicarse eficientemente entre sí. Por lo tanto, los protocolos de comunicación son un aspecto crítico en la industria 4.0.

Uno de los protocolos de comunicación más utilizados en la industria 4.0 es el protocolo Ethernet. Ethernet es un estándar de red de área local (LAN) que permite la transmisión de datos a altas velocidades y proporciona una conectividad fiable y segura. Ethernet se utiliza en una amplia variedad de aplicaciones industriales, desde la automatización de procesos hasta la gestión de la ciberseguridad.

Otro protocolo de comunicación importante en la industria 4.0 es el protocolo de User Datagram Protocol (UDP) es un protocolo de comunicación en la capa de transporte que se utiliza para la transmisión de datos en la industria 4.0. UDP es un protocolo de transmisión

de datos no confiable que no realiza una confirmación de recepción de los paquetes enviados. Sin embargo, UDP es una opción popular en la industria 4.0 debido a su rapidez y eficiencia en la transmisión de datos en tiempo real. UDP es adecuado para aplicaciones que requieren una gran cantidad de datos en tiempo real, como la transmisión de video y audio en tiempo real, ya que no requiere una confirmación de recepción de los paquetes enviados y, por lo tanto, es más rápido que otros protocolos.

Además, el protocolo MQTT (Message Queue Telemetry Transport) es un protocolo de comunicación utilizado en la industria 4.0 para la transmisión de datos de Internet de las cosas (IoT). MQTT es un protocolo de comunicación eficiente en términos de recursos y permite la transmisión de datos en tiempo real en entornos con baja capacidad de ancho de banda y alta latencia.

Otro protocolo importante en la industria 4.0 es el protocolo OPC UA (OPC Unified Architecture), que es un estándar abierto de comunicación utilizado en la industria para la transmisión de datos de manera segura y confiable. OPC UA permite la integración de diferentes sistemas y tecnologías, lo que brinda una visibilidad completa y una gestión eficiente de la información en la industria 4.0.

Por lo cual, se optó por utilizar el protocolo UDP como el protocolo OPC UA, debido a sus características únicas y complementarias. UDP es adecuado para aplicaciones que requieren una gran cantidad de datos en tiempo real, ya que no requiere una confirmación de recepción de los paquetes enviados y, por lo tanto, es más rápido que otros protocolos. Por otro lado, OPC UA es un protocolo de comunicación seguro y escalable que permite la integración de diferentes sistemas y la transmisión de datos críticos con un alto nivel de fiabilidad y seguridad.

Al utilizar ambos protocolos juntos, se puede obtener una solución de comunicación completa que combina la velocidad y eficiencia de UDP con la seguridad y escalabilidad de OPC UA. Esto permite aprovechar al máximo las fortalezas de cada protocolo para lograr

una mayor eficiencia y productividad en la transmisión de datos y la interconexión de sistemas.

3.1.1.3 Selección del software de control

El software de control que se utiliza es 4DIAC IDE ya que permite la supervisión y control de procesos a nivel industrial. Esta herramienta facilita la configuración de los dispositivos involucrados en el sistema, generando aplicaciones para cada uno de ellos, lo que facilita la interoperabilidad, cumpliendo de esta manera con una de las características de la industria 4.0.

En el entorno de 4DIAC IDE se permite el desarrollo de FBs (Bloques Funcionales) según los requerimientos de los procesos y simulación del funcionamiento de cada uno de ellos de forma individual, lo cual es muy útil ya que depura los errores que pueden afectar al sistema de control final. Además, este software permite la integración de distintos dispositivos y protocolos de comunicación, lo cual facilita la conexión entre diferentes componentes del sistema de control.

3.1.1.4 Selección del software de ejecución

El software de ejecución que se utiliza es FORTE de 4Diac IDE, ya que ofrece soporte para estándares industriales, como AUTOSAR y IEC 61499, lo que permite el desarrollo de aplicaciones de control de sistemas de alta calidad y confiabilidad.

Además de ser independiente de la plataforma en la que se ejecuta, pudiendo así ser ejecutado en diferentes dispositivos, proporcionando de esta manera una estructura de comunicación flexible y una arquitectura con la capacidad de adaptarse al cualquier diseño de las aplicaciones. En resumen, el uso de FORTE en Eclipse puede ayudar a mejorar la eficiencia y productividad del desarrollo de software, así como aumentar la calidad del código resultante.

3.1.1.5 Selección del software para el desarrollo de la interfaz gráfica

El software de para el desarrollo de la interfaz gráfica que se utiliza es LabVIEW (Laboratory Virtual Instrument Engineering Workbench), el cual es un entorno de desarrollo de sistemas de adquisición de datos y control que se utiliza ampliamente en el campo de la ingeniería ya que es un entorno visual que permite a los usuarios crear interfaz gráfica de usuario mediante una amplia gama de herramientas y componentes gráficos que pueden utilizarse para crear una interfaz gráfica compleja y atractiva.

Labview, incluye una biblioteca de funciones integradas que permiten a los usuarios realizar tareas comunes, como adquirir datos de sensores y dispositivos, procesar y analizar datos, y controlar sistemas. Lo cual facilita la integración con una amplia variedad de hardware y software, siendo muy útil en entornos donde se necesita conectar y controlar distintos dispositivos y sistemas.

3.1.2 Selección del Hardware

3.1.2.1 Selección del hardware para la adquisición/envío de datos

El hardware para la adquisición/envío de datos que se utiliza es un PLC SIMATIC ET 200SP ya que es un controlador programable lógico que se utiliza en una amplia variedad de aplicaciones industriales debido a su fiabilidad, facilidad de programación y mantenimiento, y alta eficiencia y flexibilidad. Además, el PLC SIMATIC ET 200SP está diseñado para trabajar en entornos industriales exigentes y puede integrarse fácilmente en sistemas de automatización más grandes. Además, permite la automatización de procesos industriales de forma rápida y eficiente, lo que reduce el tiempo y costo de producción, ya que cuenta con una amplia variedad de módulos y dispositivos de entrada y salida que permiten la conexión y control de distintos equipos y dispositivos.

3.1.2.2 Selección del hardware para el control de los procesos

El hardware para el control de los procesos que se utiliza es una Raspberry Pi B+, la cual se le reconoce como un controlador lógico programable de bajo costo, siendo fácil de usar, tiene un rendimiento sólido y está diseñada para ser utilizada en proyectos de desarrollo como es

en este caso. La tarjeta Raspberry Pi B+, presenta un precio muy asequible, lo que la hace ideal para proyectos de investigación ya que presenta versatilidad en cuanto al desarrollo de prototipos semi industriales. Además, la Raspberry Pi B+ cuenta con una amplia variedad de módulos y dispositivos de entrada y salida que permiten la conexión y control de distintos dispositivos, ya que cuenta con un rendimiento sólido.

3.1.3 Desarrollo de elementos del sistema de control

3.1.3.1 Bloques Funcionales para la adquisición/envío de datos

Los FBs (Bloques Funcionales) se crean en el software 4Diac IDE, en base a la estructura establecida por el estándar IEC 61499, partiendo de un FB de servicio que el mismo software proporciona, con la diferencia que en el ingreso “PARAMS” se designa la entrada o salida con la cual se va a trabajar, en la salida del FB se tendrá un dato de entrada que será de tipo Bool para señales digitales y de tipo int para señales analógicas.

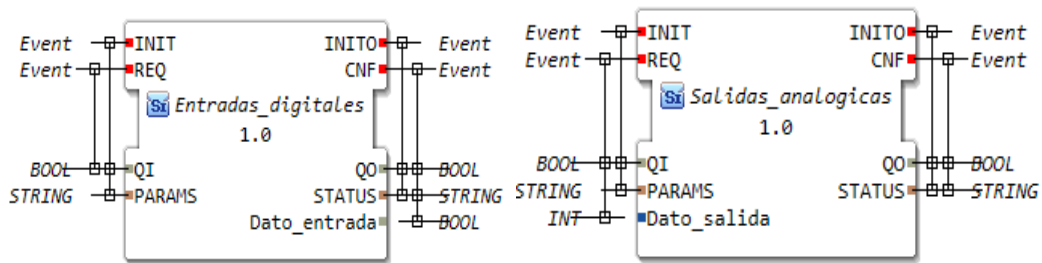


Figura 2. Esquema del FB (Bloque Funcional) para las entradas y salidas de la IPC.

Para un mejor manejo de las entradas digitales de la IPC se realiza un FB compuesto en base a la estructura del FB individual, como se observa en la Figura 3.

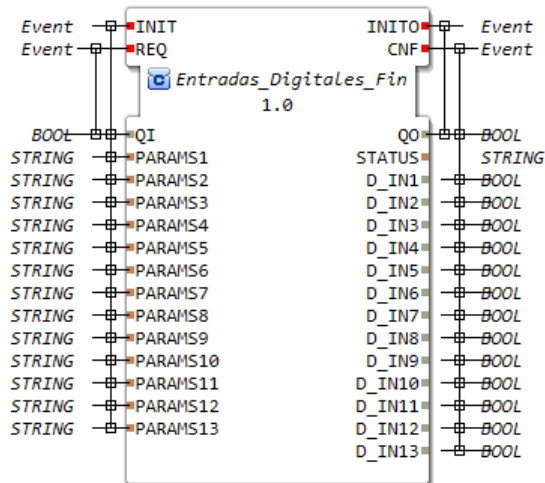


Figura 3. Esquema del FB compuesto para las entradas digitales de la IPC.

Al crear los FBs (Bloques Funcionales) se procede a exportarlos desde el software 4Diac, obteniendo dos archivos en C++ con el nombre de los FBs creados, en este caso se obtiene Entradas_Digitales_Fin.cpp, Entradas_Digitales_Fin.h, Salidas_analogicas.cpp, Salidas_analogicas.h, Salidas_digitales.cpp y Salidas_digitales.h. En los cuales se les agregan configuraciones para la adquisición/envío de datos por medio de memoria compartida.

La adquisición/envío de datos se desarrolla en base a programación en C++ y el uso de un archivo .dll de Simatic ODK 1500S el cual permite trabajar con programación C++ al IPC (Computador Industrial). De esta manera se utilizan archivos .cpp y .h en donde se programa mediante C++ la adquisición/envío de datos de las entradas y salidas del IPC. Dichos archivos se utilizan en conjunto a las demás librerías requeridas para la creación de la aplicación FORTE que usa en el IPC.

En otras palabras, la memoria compartida en este caso permite a acceder y compartir datos en una ubicación común en la memoria del sistema del IPC (Computador Industrial). Cuando un proceso quiere enviar datos a otro proceso a través de la memoria compartida, primero escribe los datos en la ubicación de memoria compartida. Luego, notifica al otro proceso de que hay datos disponibles en la memoria compartida para leer. Cuando el otro proceso recibe la notificación, lee los datos de la memoria compartida y los procesos según sea necesario.

En resumen, la memoria compartida permite a los procesos enviar y recibir datos de forma rápida y eficiente mediante el uso de una ubicación común en la memoria del sistema del IPC, como se observa en el diagrama de clases de la Figura 4.

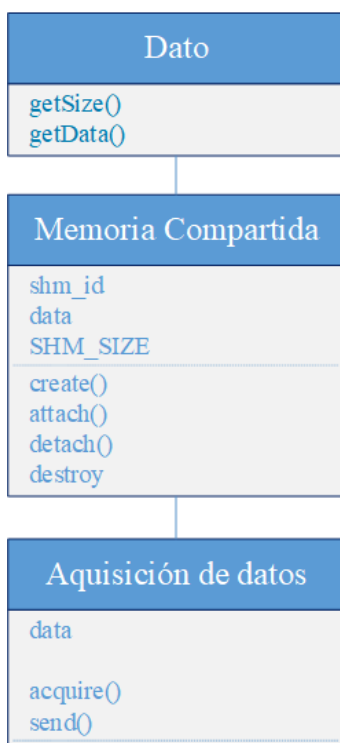


Figura 4. Diagrama de clase para la adquisición/envío de datos mediante memoria compartida.

Después de realizar las configuraciones necesarias se procede a ubicar los archivos generados en un directorio, para de esta manera compilarlos con las demás librerías requeridas para generar la aplicación FORTE que se ejecutara en el IPC (Computador Industrial).

3.1.3.2 Bloques Funcionales para el control de los procesos

Los FBs (Bloques Funcionales) para el control de los procesos se crearon a partir de la estructura de un FB básico al cual se le agrego un cierto número de entradas y salidas dependiendo de la cantidad de sensores, controles, contadores y actuadores a utilizar en cada proceso.

El proceso1 corresponde a una estación de clasificación que consta de bandas transportadoras, un brazo robótico y sensores para la detección del producto. Dicho proceso se encarga de mover los ítems de la banda de ingreso cada vez que es detectado por el sensor detección y moverlos hacia la banda de salida, de esta manera el FB (Bloque funcional) de dicho proceso presenta la estructura que se muestra en la Figura 5.

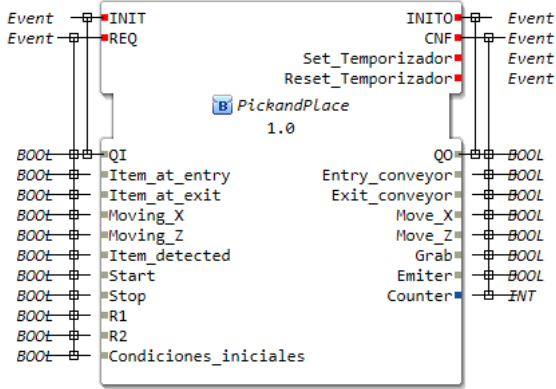


Figura 5. Esquema del FB del proceso 1.

La configuración del ECC (Gráfico de Control de Ejecución) del bloque funcional es esencial para garantizar la eficiencia del proceso. En este gráfico, se unen de forma consecutiva los eventos con los datos necesarios para la activación del requerimiento, como se muestra en la Figura 6. Además, se pueden establecer condiciones y reglas para que el proceso se lleve a cabo de manera automatizada, lo que reduce la posibilidad de errores humanos. También se pueden establecer alertas y notificaciones en caso de que algo salga fuera de lo esperado, permitiendo tomar medidas rápida y eficazmente. Además, se pueden establecer indicadores de desempeño para medir y mejorar la eficiencia del proceso.

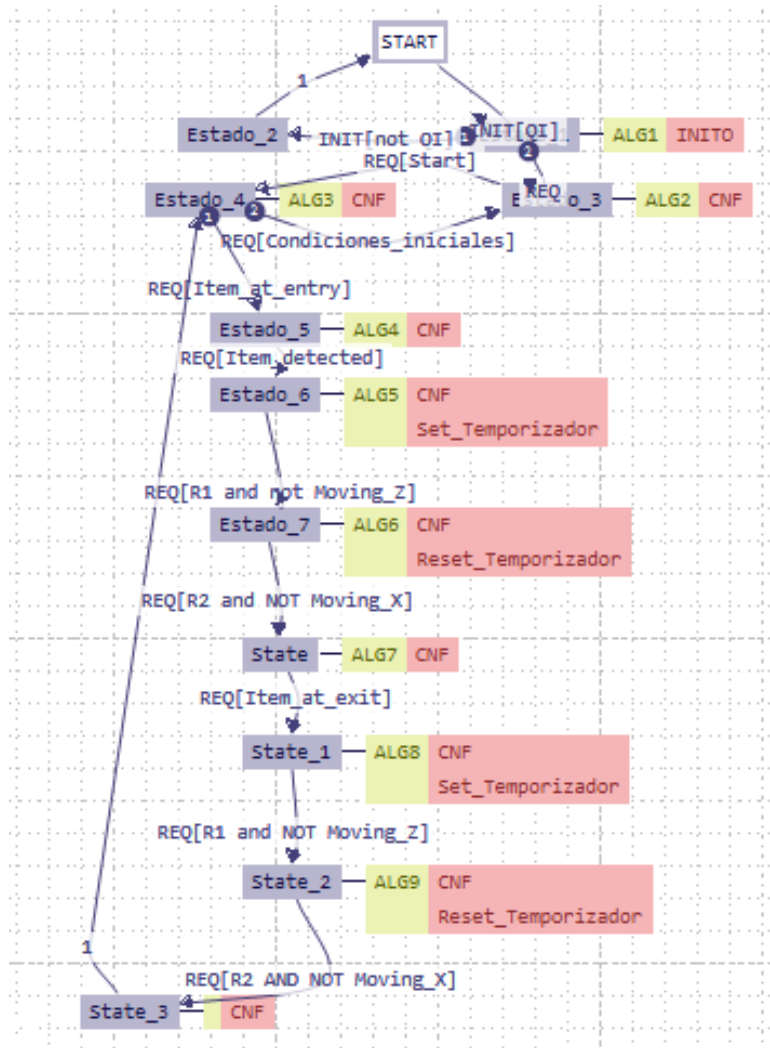


Figura 6. Gráfico de control de ejecución del proceso 1.

A partir de la configuración del ECC (Gráfico de control de ejecución) se realiza la exportación del FB (Bloque Funcional) para posteriormente con los archivos .cpp y .h compilar en conjunto las demás librerías necesarias para generar la aplicación FORTE que se encargara de la ejecución del proceso 1 en el dispositivo de control.

De esta manera al igual que con el proceso 1, el FB (Bloque Funcional) del proceso 2 se diseña en base a la cantidad de entradas y salidas necesarias, de acuerdo a la cantidad de sensores, controles y actuadores involucrados en el proceso.

El proceso 2 corresponde a un proceso de clasificación por tamaños de productos, en el cual se clasifican cajas grandes y pequeñas, constando de esta manera con una banda de ingreso,

una banda multidireccional que moverá las cajas hacia la izquierda o hacia la derecha dependiendo de su tamaño, de dos bandas de salidas para cada tipo de caja y sensores (reflectivos, inductivos, capacitivos, etc.). Además, el proceso posee contadores de productos ingresados y un contador por cada tipo de caja, para de esta manera supervisar el proceso, como se observa en la Figura 7.

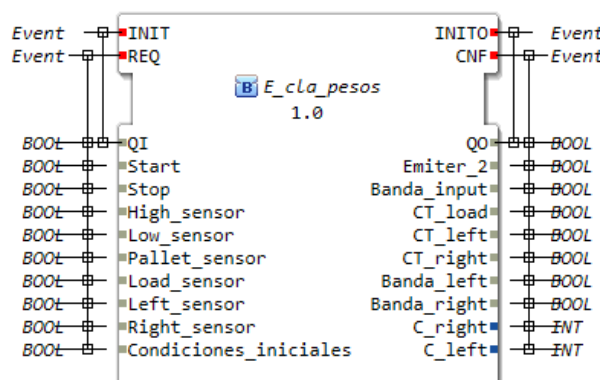


Figura 7. Esquema del FB del proceso 2.

Al igual que con el proceso 1, para el FB (Bloque Funcional) del proceso 2 se realizó la configuración del ECC (Gráfico de control de ejecución) con el objetivo de garantizar la eficiencia del proceso. En este gráfico, se unen de forma consecutiva los eventos con los datos necesarios para la activación del requerimiento. Además, se establecen las interacciones entre los diferentes bloques funcionales involucrados, permitiendo una mayor coordinación y colaboración entre los diferentes departamentos, como se observa en la Figura 8. También se establecen los puntos de control clave para garantizar que el proceso se lleve a cabo de manera eficiente y se cumplan los objetivos establecidos. Es importante mencionar que en este proceso se aprovechan las ventajas de la automatización y la monitorización continua para mejorar el rendimiento del proceso.

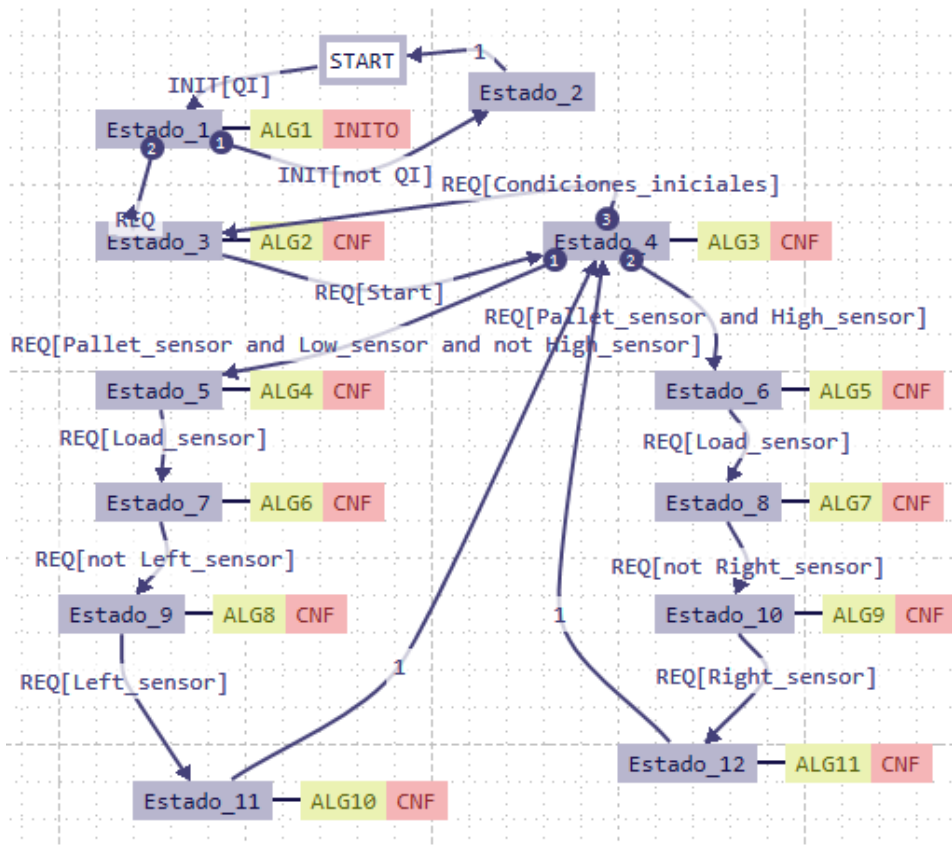


Figura 8. Gráfico de control de ejecución del proceso 2.

De esta manera a partir de la configuración del ECC (Gráfico de control de ejecución) se realiza la exportación del FB (Bloque Funcional) para posteriormente con los archivos .cpp y .h compilar en conjunto las demás librerías necesarias para generar la aplicación FORTE que se encargara de la ejecución del proceso 2 en el dispositivo de control.

3.1.3.3 Bloques Funcionales para el control de los contenedores

Los FBs (Bloques Funcionales) para el control de los contenedores se crearon a partir de la estructura de un FB de servicio, con la diferencia que en “PARAMS” se ingresaran los comandos para correr o parar los contenedores. Los comandos que se ingresaron son los siguientes:

- 'run; p# (p1 o p2 dependiendo del proceso a ejecutar); # (1 o 2 dependiendo del proceso a ejecutar)'

- 'kill; p# (p1 o p2 dependiendo del proceso a ejecutar); # (1 o 2 dependiendo del proceso a ejecutar)

De esta manera el FB para el control de los contenedores presenta la estructura mostrada en la Figura 9.

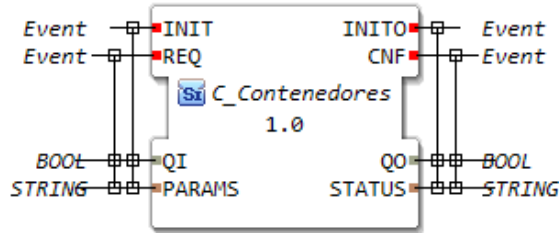


Figura 9. Esquema del FB para el control de los contenedores.

Se exporta el FB (Bloque Funcional) para posteriormente configurar los archivos .cpp y .h generados. Con los archivos configurados se procede a compilar la aplicación FORTE que se encarga de la ejecución del control de los contenedores desde los dispositivos de control que en este caso son las Raspberry Pi.

3.1.3.4 Creación de contenedores de los procesos

Los procesos a controlar se virtualizan en este caso mediante la creación de los contenedores de cada uno de los mismos, de esta manera se procede a crear los contenedores mediante las aplicaciones FORTE que controlan el proceso 1 y el proceso 2. De esta manera en las tarjetas Raspberry Pi se establece un directorio para la aplicación forte y para el archivo Dockerfile. En el archivo Dockerfile se realizarán las configuraciones necesarias para la creación del contenedor. Posteriormente en mediante el comando “docker build” se procede a construir el contenedor del proceso en cada una de las Raspberry Pi, obteniendo la siguiente estructura de los contenedores en relación a los dispositivos como se muestra en la Figura 10.

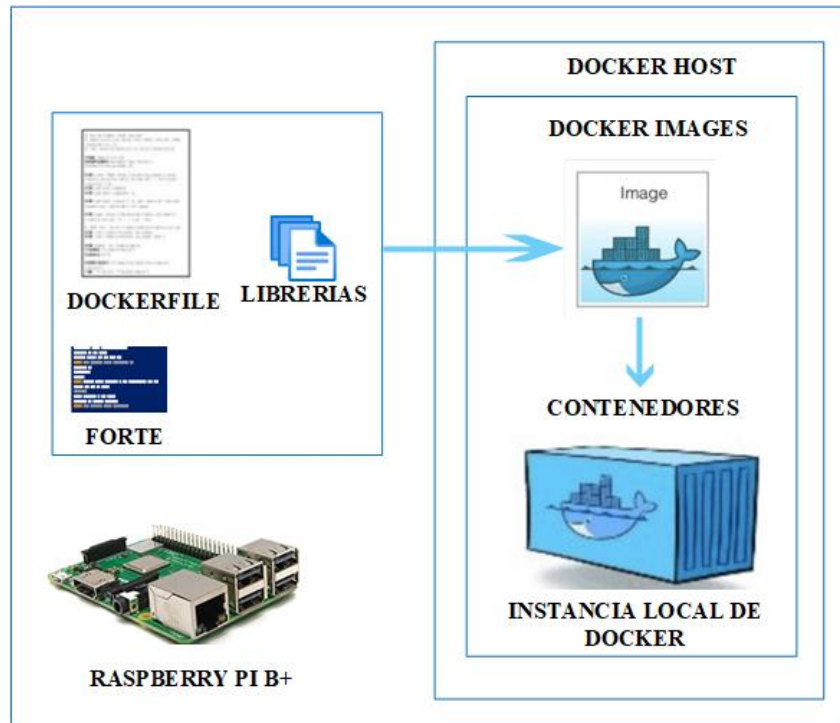


Figura 10. Estructura de los contenedores en los dispositivos de control del sistema.

3.1.3.5 Diseño de la interfaz gráfica

La interfaz gráfica es una aplicación que se crea en base al software Labview en base a ciertos lineamientos de la normativa ISA101 con la finalidad de mejorar la eficiencia y la seguridad de control de los procesos, de tal manera que la misma sea fácil de usar y entender por el operador del proceso. Por ende, la interfaz gráfica se compone de indicadores de estado de los procesos y de los sensores, contadores de ítems producidos y un control de paro e inicio de los procesos.

Los datos son enviados y recibidos desde la interfaz mediante el protocolo de comunicación OPC UA, de tal manera que para ello se utilizan las funciones de Labview relacionadas a la publicación y suscripción de datos, por lo cual a la interfaz se le configuro la dirección y el puerto de enlace para la comunicación con cada uno de los procesos y los NODE ID de cada una de las variables que se van a supervisar y controlar de los procesos, obteniendo de esta manera la estructura mostrada en la Figura 11.

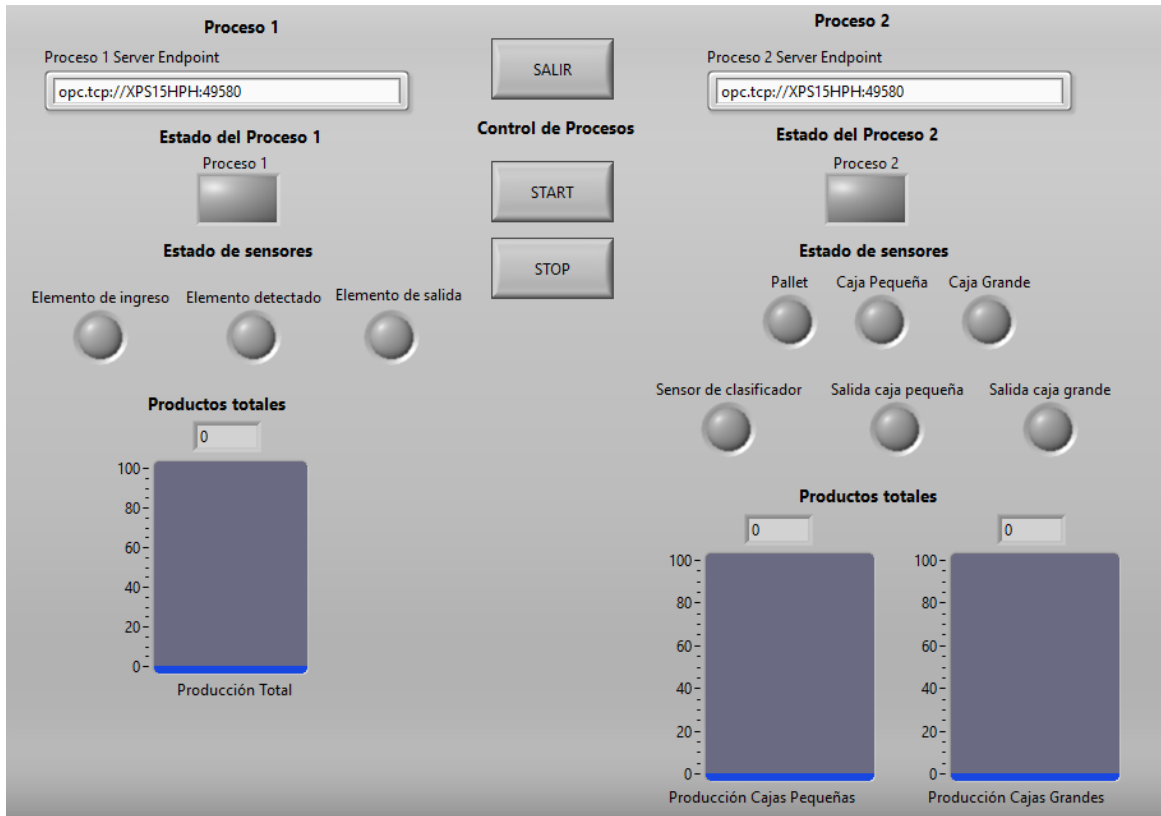


Figura 11. Interfaz Gráfica para el control y supervisión de los procesos.

3.1.3.6 Protocolo de comunicación UDP – Bloques Funcionales

El protocolo de comunicación UDP se utiliza para la comunicación entre los dispositivos de adquisición/ envío de datos y de control que son el IPC (Computador Industrial) y las Raspberry Pi respectivamente. Debido a que dicho protocolo de comunicación presenta una alta velocidad de transferencia de datos, flexibilidad y adaptabilidad para los dispositivos con el entorno, siendo ideal para aplicaciones que requieren una transferencia de datos rápida, como la transmisión de datos de los procesos en tiempo real, permitiendo que el sistema de control detecte en tiempo real cualquier acontecimiento que se presente en los procesos ejecutados.

De esta manera se utilizan dos FBs (Bloques Funcionales) que el PUBLIS y SUBSCRIBE, en los cuales se establece en el “ID” del FB la dirección y el puerto de los dispositivos, como se muestra en la Figura 12.

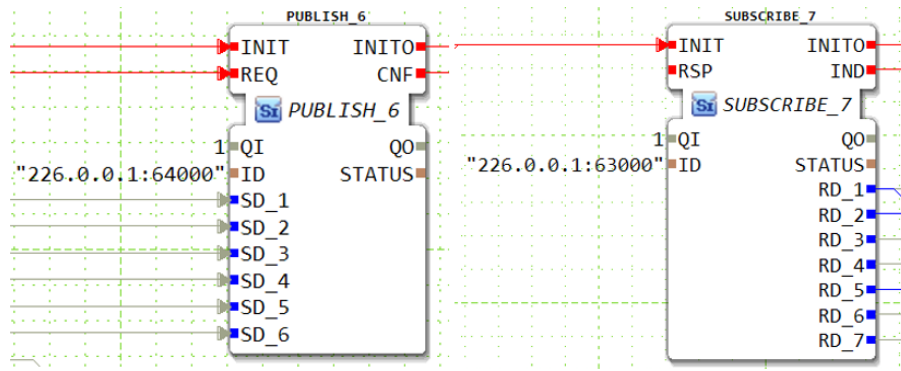


Figura 12. Configuración del protocolo de comunicación UDP entre IPC y Raspberry Pi.

En el sistema propuesto la estructura del proceso de comunicación UDP entre los dispositivos IPC y las Raspberry Pi corresponde a la mostrada en la Figura 13.

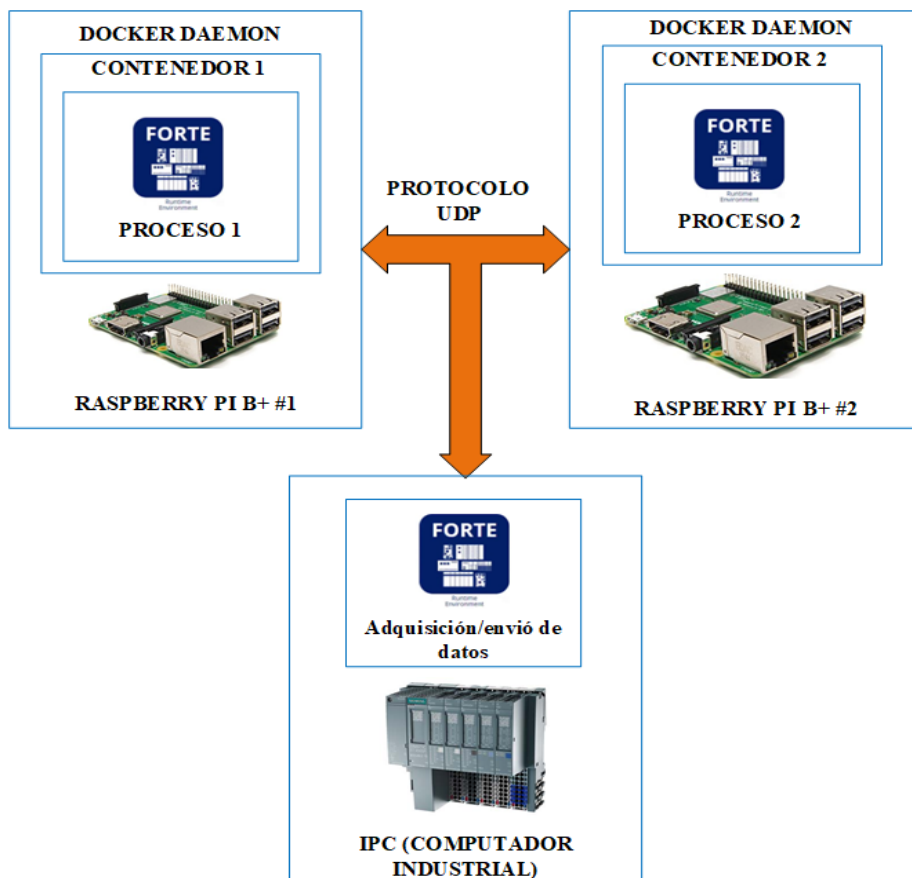


Figura 13. Estructura del proceso de comunicación entre las tarjetas raspberry pi y el IPC mediante protocolo UDP.

3.1.3.7 Protocolo de comunicación OPC UA – Bloques Funcionales

El protocolo de comunicación OPC UA se utiliza para el desarrollo de la comunicación entre los dispositivos de control que son las Raspberry Pi en las cuales se va a ejecutar el control de los procesos con la Interfaz Gráfica con la cual se realizará la supervisión y control de los procesos. Dicho protocolo se implementa en el sistema actual debido a que es un estándar de comunicación industrial ampliamente utilizado en diferentes campos, como la automatización industrial, el transporte y la energía.

Debido a que garantiza la interoperabilidad entre diferentes dispositivos del sistema, dado que el protocolo OPC UA utiliza un lenguaje común para la comunicación de datos, permitiendo que los diferentes dispositivos se comuniquen entre sí de manera eficiente y fiable. Además, ofrece un excelente nivel de seguridad durante el envío/recepción de datos. El protocolo incluye diversas medidas de seguridad, como autenticación de usuarios, cifrado de datos y autorización de acceso, garantizando de esta manera que los datos se transfieren de manera segura entre los dispositivos.

En el desarrollo de la comunicación entre los dispositivos de control y la interfaz gráfica se utilizan los FBs (Bloques Funcionales) PUBLISH que en este caso se utiliza para publicar los datos del proceso en la interfaz gráfica y SUBSCRIBE que se utiliza para suscribirse a los datos que se envían de la interfaz gráfica hacia el proceso. En los FBs en “ID” se ingresa los siguientes comandos que corresponden a lectura y escritura respectivamente:

- "opc_ua[Función(READ_o_WRITE);/Objects/NodeID(1:data/1:Cl,1:s=init);/Objects/Node ID (1:data/1:Cl,1:s=stop)]"

El ingreso de los comandos de acuerdo a lo descrito anteriormente se realiza de la manera representada en la Figura 14.

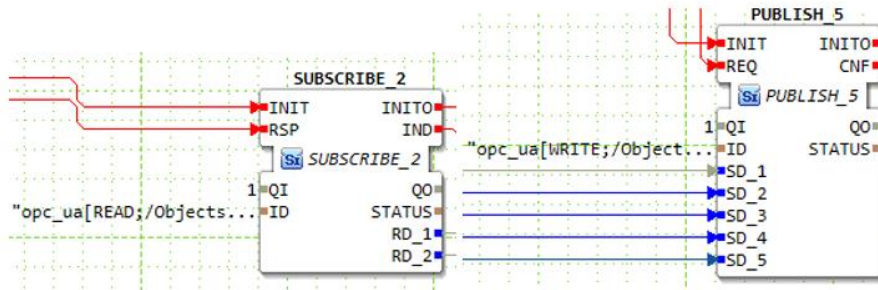


Figura 14. Configuración del protocolo de comunicación OPC UA entre Raspberry Pi #1 - Proceso 1 e Interfaz Gráfica.

La estructura de la comunicación entre las tarjetas Raspberry Pi y la interfaz gráfica mediante el protocolo de comunicación OPC UA corresponde a la estructura de la Figura 15.

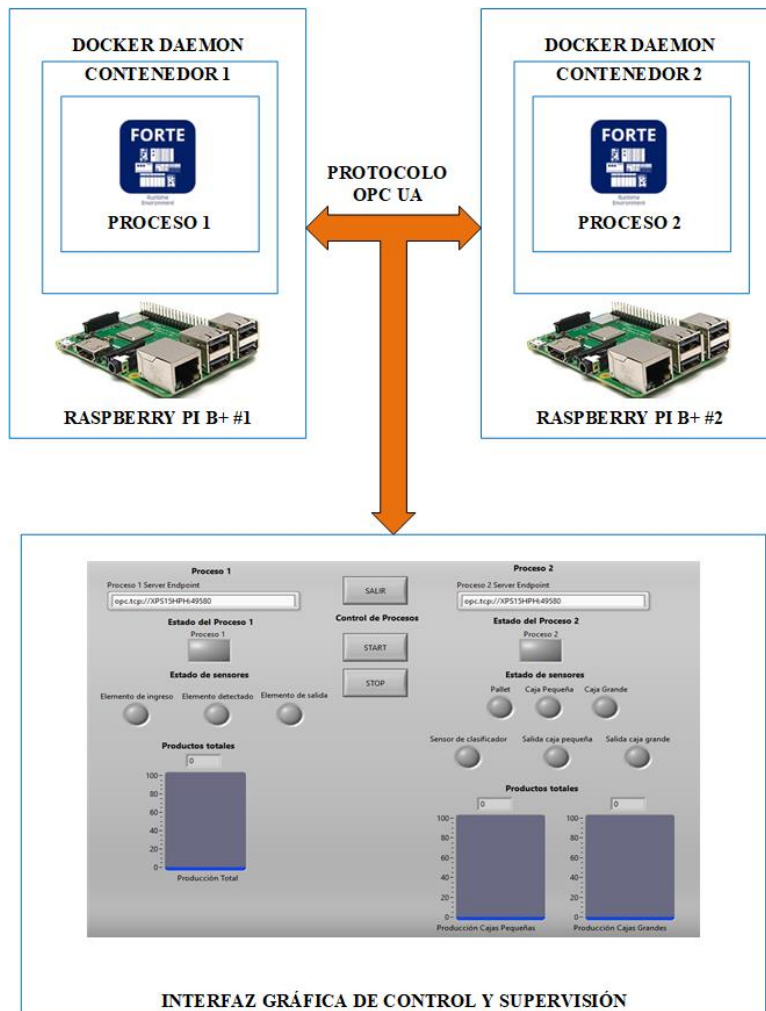


Figura 15. Estructura del proceso de comunicación entre las tarjetas raspberry pi y la Interfaz Gráfica mediante protocolo OPC UA.

3.1.4 Desarrollo del sistema de control

3.1.4.1 Red de dispositivos en 4Diac IDE

La red de dispositivos que se crea en el entorno de 4Diac IDE está compuesta por los controladores que son las Raspberry Pi, el IPC (Computado Industrial), los contenedores con los que se virtualizaron los procesos, tal y como se muestra en la Figura 16.

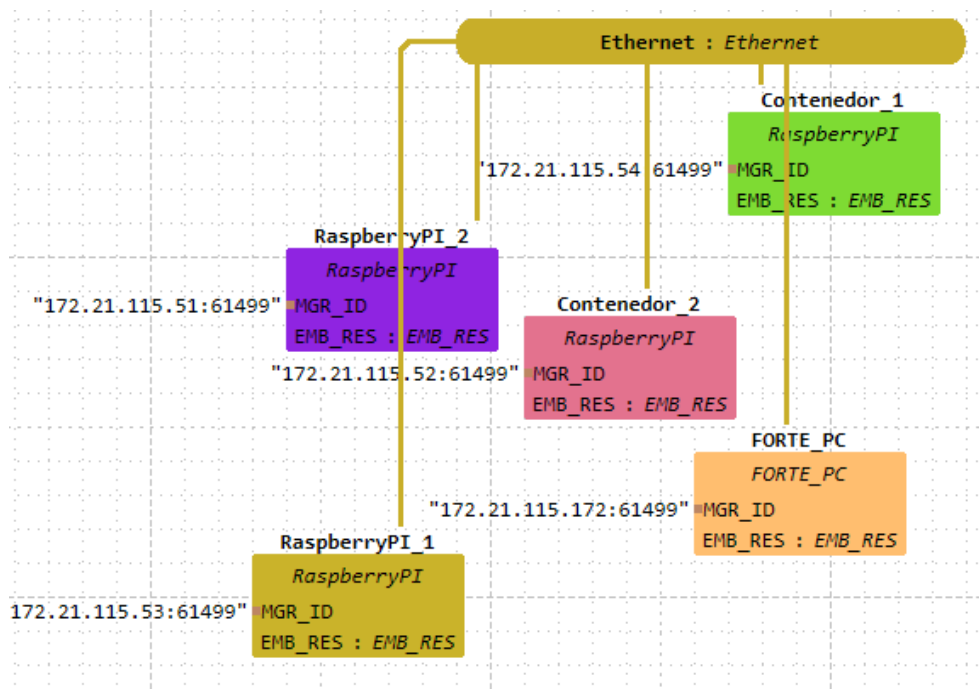


Figura 16. Red de dispositivos en el entorno de 4Diac IDE.

Los dispositivos que se encuentran conectados a la red representan cada una de las aplicaciones FORTE generadas tanto para los dispositivos físicos como para los contenedores. De esta manera se establece lo siguiente:

- FORTE_PC=Es la aplicación FORTE que se encarga de la adquisición/envío de datos de los procesos a través del IPC (Computador Industrial), y a su establece la comunicación entre el IPC y las Raspberry Pi mediante el protocolo de comunicación UDP.
- RaspberryPI_1 y RaspberryPI_2= Corresponden a las aplicaciones FORTE que se encargaran del control de los contenedores de los procesos virtualizados.

- Contenedor_1 y Contenedor_2=Corresponden a las aplicaciones FORTE que se encargaran de ejecutar el control de los procesos y a su vez de la comunicación de los mismos con la interfaz gráfica mediante el protocolo de comunicación OPC UA, siempre y cuando los contenedores se encuentren ejecutados en las tarjetas Raspberry Pi.

3.1.4.2 Diseño del sistema de control

El sistema de control se crea con el objetivo de que se va utilizar en diversas industrias para monitorear y controlar los procesos de producción y otros procesos de gran relevancia para la industria. El IPC es un dispositivo de control específicamente diseñado para el desarrollo de aplicaciones industriales como en este caso, este dispositivo va a recopilar y enviar datos de los procesos al resto del sistema de control. Para la comunicación con los dispositivos de control, que en este caso son las tarjetas Raspberry Pi, el IPC utiliza el protocolo de comunicación UDP, ya que es un protocolo de comunicación que envía y recibe los datos en tiempo real, reduciendo de esta manera los tiempos de envío y recepción entre los dispositivos.

Las tarjetas Raspberry Pi se han vuelto populares debido a su capacidad de desarrollar proyectos de automatización y control de manera económica. En este sistema, las tarjetas Raspberry Pi se encargan de ejecutar los contenedores, que son aplicaciones que funcionan en un entorno aislado y pueden ser fácilmente transportadas de un dispositivo a otro, proporcionando una solución flexible e interoperable.

Cuando los contenedores están en funcionamiento en las tarjetas Raspberry Pi, controlan los procesos virtualizados y comunican y reciben datos a través de la interfaz gráfica. El sistema de control que se crea en base a los dispositivos mencionados corresponde al diagrama de clase que se observa en el diagrama de clase de la Figura 17.

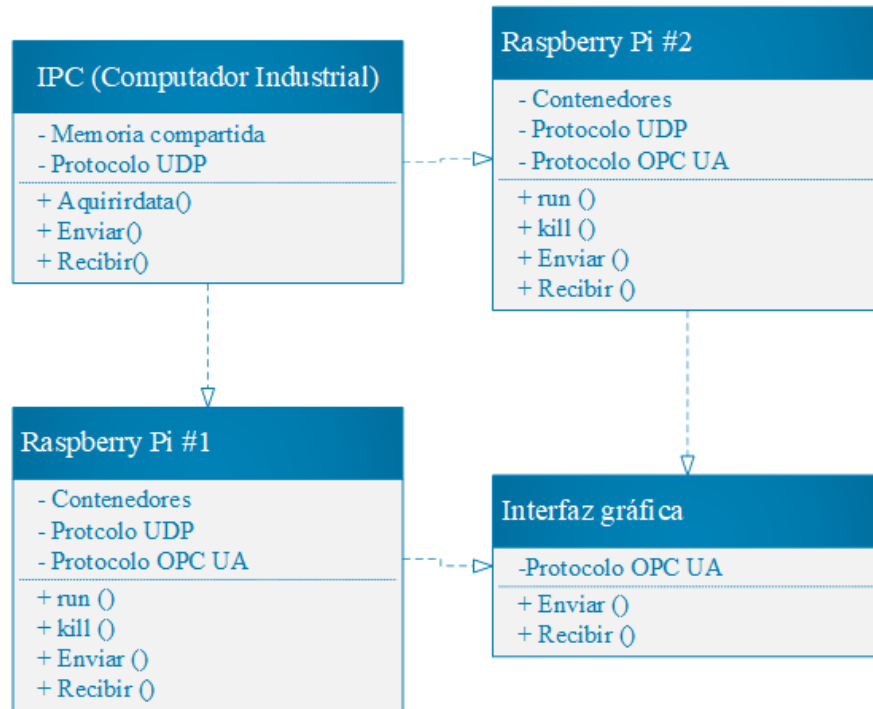


Figura 17. Diagrama de clases del sistema de control.

3.1.5 Resultado final

3.1.5.1 Sistema de control basado en el Estándar IEC 61499 y el uso de contenedores.

El proyecto investigativo se desarrolla como un sistema de control distribuido que es altamente modulizable y escalable, debido a que no se enfoca en el control de un solo proceso, ya que permite el control y supervisión de un gran número de procesos sin la necesidad de alterar los demás procesos que se encuentren relacionados al sistema.

El sistema desarrollado se basa en el concepto de FBs (Bloques Funcionales), los cuales son unidades modulares de control que son reutilizadas y combinadas para crear los sistemas de control de los procesos como se observa en la Figura 18, en donde los controles de los procesos reutilizan los FBs de la adquisición/envío de datos para cada uno de los procesos sin la necesidad de modificar el FB de control de ningún proceso. Esto permite una gran flexibilidad, lo cual facilita la modificación y actualización del control de cada uno de los procesos sin la necesidad de volver a diseñar todo el sistema de control desde cero.



Figura 18. Estructura del sistema de control de los procesos y la adquisición/envío de datos.

El uso de contenedores Docker permite aislar cada bloque funcional en un contenedor separado y ejecutarlo en un entorno aislado. Esto permite que el sistema de control sea más fácil de desplegar y mantener, ya que cada Bloque Funcional se puede actualizar o modificar de forma independiente sin afectar el funcionamiento del resto del sistema. Además, el uso de contenedores Docker permite que el sistema de control se ejecute en diferentes plataformas y entornos sin tener que realizar cambios en el código del sistema, lo cual se observa en la Figura 19, en donde el contenedor del proceso 1 funciona con normalidad mientras el contenedor del proceso 2 se encuentra parado, observando de esta manera el funcionamiento aislado de los procesos.

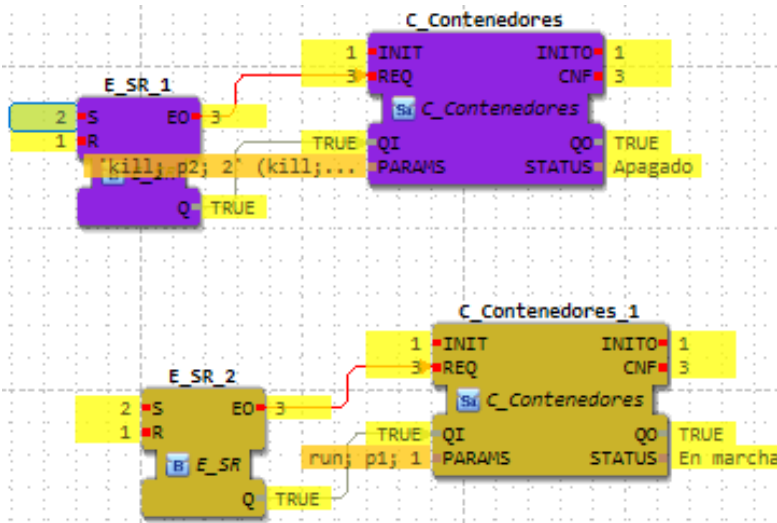


Figura 19. Bloques Funcionales para el control de ejecución y paro de los contenedores.

Los contenedores de los procesos también llegan a ejecutarse al mismo tiempo, sin la necesidad de modificar ningún Bloque Funcional, debido a que se ejecutan en entornos aislados de un proceso a otro, lo cual permite un fácil despliegue sin la necesidad de que los procesos lleguen a entrar en conflicto entre los mismo, como se observa en la Figura 20.

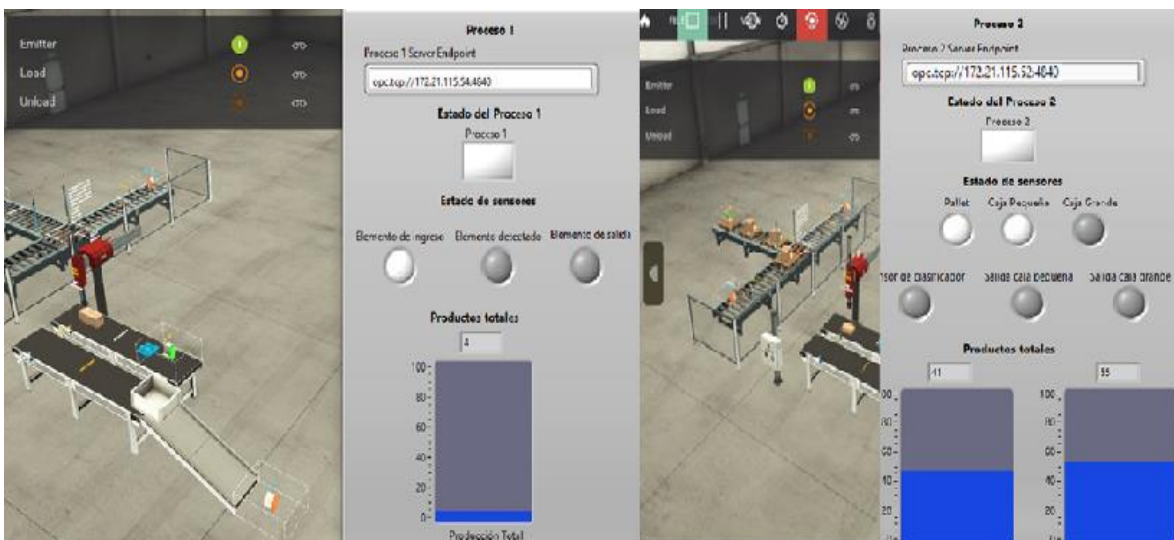


Figura 20. Control de los procesos.

3.2 Análisis de resultados

El sistema de control desarrollado proporcionó resultados para evaluar los tiempos de inicio y paro de los contenedores en los dispositivos de control. Esto permitió comparar y evaluar el rendimiento de los contenedores ejecutados en los dispositivos y determinar cómo se están utilizando.

Además, se muestra el rendimiento de los protocolos de comunicación utilizados para la comunicación entre los dispositivos de control y la interfaz gráfica, lo que permitió determinar qué protocolos son más eficientes y confiables para esta aplicación. También se demostró la ventaja de utilizar el computador industrial ET 200SP con el controlador 1500s, ya que proporciona una solución de control robusta y fiable para entornos industriales.

3.2.1 Análisis del tiempo de ejecución y paro de los contenedores

Para realizar el análisis de los tiempos de ejecución y paro de cada uno de los contenedores, se llevaron a cabo 30 mediciones. Para evaluar el tiempo de ejecución, se midió el tiempo que transcurrió desde que se ingresó el comando "run" al bloque de función (FB) de control de los contenedores hasta que éstos comenzaron a trabajar. En el caso del tiempo de paro, se midió el tiempo desde que se ingresó el comando "kill" al FB de control hasta que el contenedor se detuvo.

Los resultados de los tiempos de ejecución y paro del contenedor 1 se muestran en la Figura 21, mientras que los tiempos de ejecución y paro del contenedor 2 se muestran en la Figura 22.

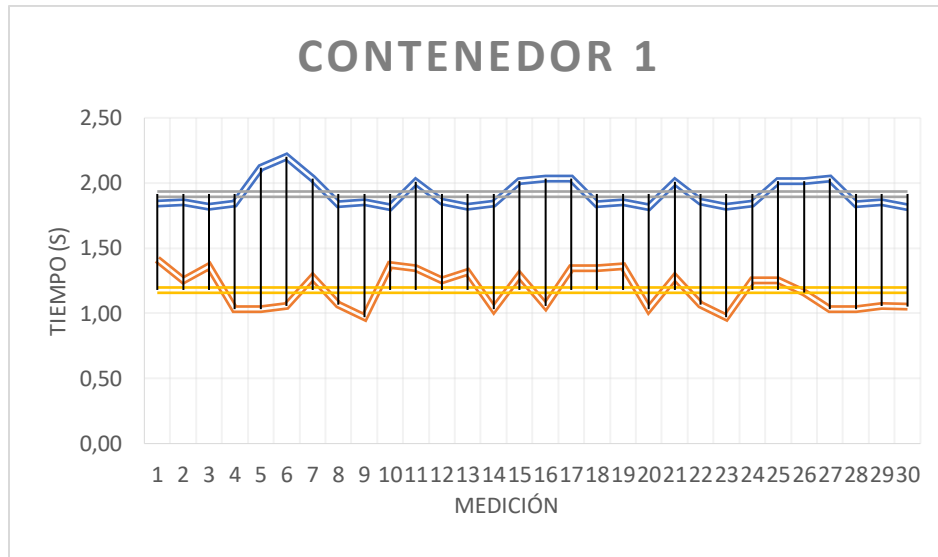


Figura 21. Gráfica de los tiempos de ejecución y paro del contenedor 1.

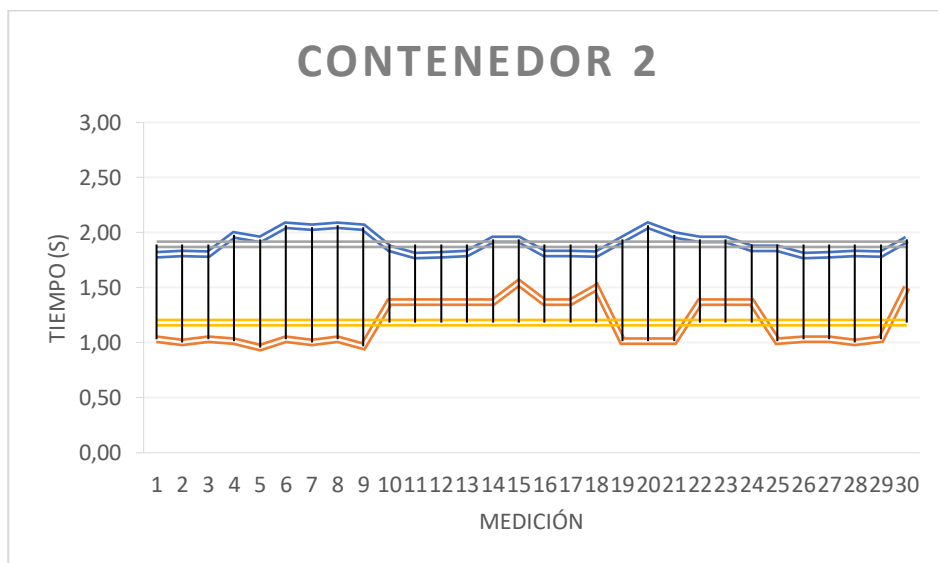


Figura 22. Gráfica de los tiempos de ejecución y paro del contenedor 2.

De las mediciones realizadas se determinó el tiempo promedio de ejecución y el tiempo promedio de paro de cada uno de los contenedores, obteniendo los datos mostrados en la Tabla 5.

Tabla 3. Tiempos promedio de ejecución y paro de cada uno de los contenedores

	Tiempo de Ejecución (s)	Tiempo de Paro (s)
Contenedor 1	1,91	1,18
Contenedor 2	1,89	1,18

De esta manera los tiempos de inicio y parada de los contenedores son similares en las dos tarjetas Raspberry Pi B+. Los tiempos de inicio son de 1.91 segundos y 1.89 segundos para el contenedor 1 y el contenedor 2 respectivamente, y el tiempo de parada es de 1.18 segundos para ambos contenedores. Esto sugiere que ambos contenedores están funcionando en el mismo entorno y no hay desigualdades en el rendimiento.

3.2.2 Análisis de los protocolos de comunicación

3.2.2.1 Análisis del protocolo de comunicación UDP

En el desarrollo de las pruebas de funcionamiento, se realizó en primera instancia el análisis del protocolo de comunicación UDP, el cual sirvió para realizar la comunicación entre el Computador Industrial y las tarjetas de Raspberry Pi B+, de esta manera mediante el software Wireshark se verificó el tráfico de datos, para verificar la integridad de los mismos, obteniendo de esta manera los siguientes datos mostrados en la Figura 23.

38	0.000008	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
65	0.000008	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
74	0.000007	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
110	0.000010	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
111	0.000799	172.21.115.52	225.0.0.1	UDP	60	37867 → 62000	Len=7
112	0.000158	172.21.115.52	225.0.0.1	UDP	60	37867 → 62000	Len=7
138	0.000186	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
150	0.000006	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
174	0.000271	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
204	0.000007	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
222	0.000006	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
251	0.000194	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8
266	0.000175	172.21.115.172	225.0.0.1	UDP	60	58528 → 61000	Len=8

Figura 23. Tráfico de datos mediante protocolo UDP entre el Computador Industrial y la Raspberry Pi B+.

Además, se realizó una comprobación estadística de los datos obtenidos en relación al proceso de comunicación entre el IPC y la tarjeta Raspberry Pi B+ #1 – contenedor 1, obteniendo la gráfica mostrada en la Figura 24.

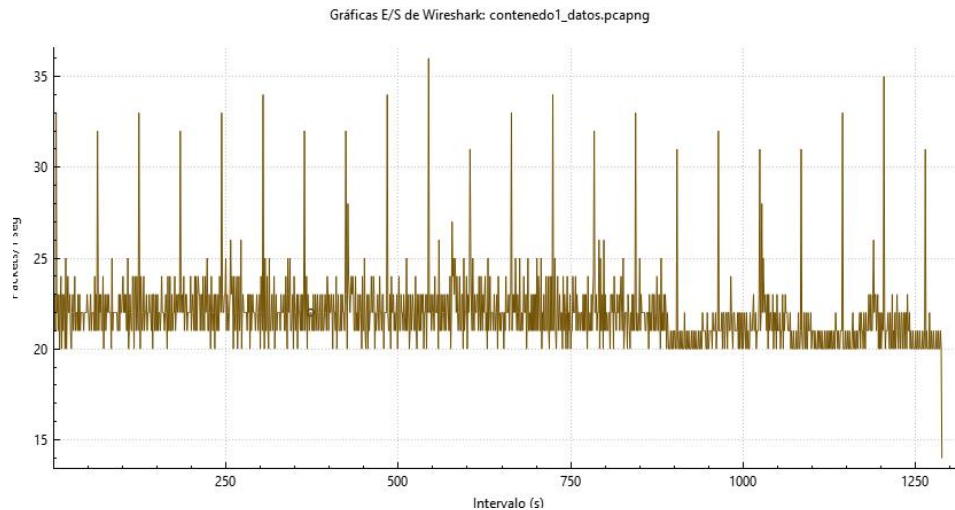


Figura 24. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #1.

De la misma manera se obtuvo la Gráfica de Entradas/Salidas entre el IPC y la Raspberry Pi B+ #2, como se muestra en la Figura 25, obteniendo un resultado similar al que se consiguió con la otra Raspberry Pi, evidenciando que tanto en las dos tarjetas el envío de paquetes de datos se realizó de forma rápida, sin excepción alguna.

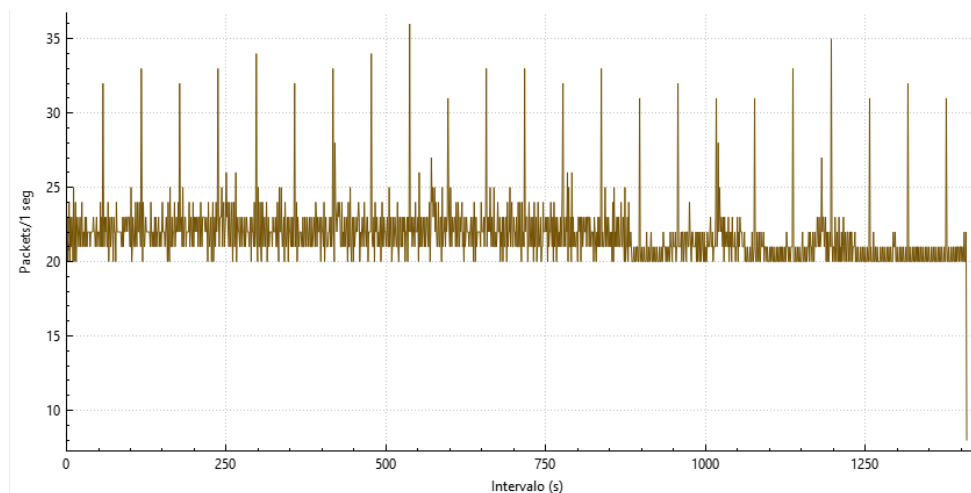


Figura 25. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2.

De esta manera se evidencia que entre los dispositivos existe una estabilidad en el tráfico de datos ya que las gráficas de entradas y salidas obtenidas muestran picos reducidos en el tráfico de datos y que el tráfico en su mayoría se encuentra dentro de un mismo rango, indicando una buena gestión de los recursos en la red, ya que los dispositivos y aplicaciones están utilizando los recursos de manera eficiente y equitativa, lo cual mejora el rendimiento de la red y reduce la posibilidad de que se generen cuellos de botella.

Además, mediante el análisis estadístico entre los puntos finales, se determinó que el protocolo de comunicación UDP funcionó al 100% en ambos dispositivos, el IPC y la Raspberry Pi B+ #1, lo que resultó en un envío y recepción de 12890 paquetes como se muestra en la Figura 26. Esto demuestra que el protocolo de comunicación UDP es eficiente y confiable para esta aplicación.

Dirección	Port	Paquetes	Bytes	Total Packets	Percent Filtered
225.0.0.1	61000	12,588 KiB	755,273 KiB	12.890	100.00%
172.21.115.172	58528	12,588 KiB	755,273 KiB	12.890	100.00%

Figura 26. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #1 – contenedor 1.

Durante el proceso de comunicación entre el Computador Industrial y la tarjeta Raspberry Pi B+ #2, se comprobó que el protocolo de comunicación UDP funcionó de manera óptima en ambos dispositivos. Se logró enviar y recibir un total de 14096 paquetes entre ambos dispositivos, como se muestra en la Figura 27. Esto demuestra la efectividad del protocolo de comunicación UDP en el proceso de comunicación entre estos dispositivos.

Dirección	Port	Paquetes	Bytes	Total Packets	Percent Filtered
226.0.0.1	63000	13,766 KiB	825,938 KiB	14.096	100.00%
172.21.115.172	58528	13,766 KiB	825,938 KiB	14.096	100.00%

Figura 27. Gráfica Entradas/Salidas entre Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2.

3.2.2.2 Análisis del protocolo de comunicación OPC UA

Para analizar el proceso de comunicación entre la Interfaz Gráfica y las tarjetas Raspberry Pi donde se ejecutaron los contenedores, se llevó a cabo un estudio del protocolo de comunicación OPC UA. Para ello, se utilizó WireShark para analizar el tráfico de datos entre la interfaz gráfica y la Raspberry #1 del contenedor 1, como se muestra en la Figura 28. De manera similar, se obtuvo el tráfico de datos entre la interfaz gráfica y la Raspberry #2 del contenedor 2.

29	0.005301	172.21.115.54	172.21.115.181	OpcUa
34	0.008232	172.21.115.54	172.21.115.181	OpcUa
39	0.000304	172.21.115.54	172.21.115.181	OpcUa
54	0.000121	172.21.115.54	172.21.115.181	OpcUa
66	0.006722	172.21.115.54	172.21.115.181	OpcUa
69	0.018430	172.21.115.54	172.21.115.181	OpcUa
78	0.005560	172.21.115.54	172.21.115.181	OpcUa
82	0.000362	172.21.115.54	172.21.115.181	OpcUa
98	0.000363	172.21.115.54	172.21.115.181	OpcUa
256	0.045014	172.21.115.54	172.21.115.181	OpcUa
261	0.040798	172.21.115.54	172.21.115.181	OpcUa
269	0.027895	172.21.115.54	172.21.115.181	OpcUa
274	0.018177	172.21.115.54	172.21.115.181	OpcUa

Figura 28. Tráfico de datos entre el Computador Industrial y la Raspberry Pi B+ #1 – contenedor 1 mediante el protocolo OPC UA.

De acuerdo a los datos obtenidos, se procedió a determinar las estadísticas de jerarquía del protocolo OPC UA entre la Interfaz Gráfica y la tarjeta Raspberry Pi #1. Los resultados mostraron que se enviaron y recibieron el 100% de los paquetes entre ambos dispositivos, con un total de 20370 paquetes generados durante el proceso de comunicación. Esto se muestra en la Figura 29.

	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets
et	100.0	20370	100.0	3120979	22 k	0
Ethernet Protocol Version 4	100.0	20370	9.1	285180	2058	0
Transmission Control Protocol	100.0	20370	13.1	407400	2940	0
OpcUa Binary Protocol	100.0	20370	77.8	2428399	17 k	0
OpcUa Binary Protocol	100.0	20370	64.8	2020999	14 k	20370

Figura 29. Estadísticas de jerarquía entre el Computador Industrial y la Raspberry Pi B+ #1 – contenedor 1 mediante el protocolo OPC UA.

Seguindo la metodología utilizada para analizar la comunicación entre la interfaz gráfica y la Raspberry Pi #1 del proceso 1, se llevó a cabo un análisis del tráfico de datos entre la interfaz gráfica y la Raspberry Pi #2 del proceso 2. Los resultados se presentan en la Figura 30.

1180...	0.009045	172.21.115.52	172.21.115.181	OpcUa
1179...	0.000111	172.21.115.52	172.21.115.181	OpcUa
1179...	0.000191	172.21.115.52	172.21.115.181	OpcUa
1179...	0.000216	172.21.115.52	172.21.115.181	OpcUa
1179...	0.018162	172.21.115.52	172.21.115.181	OpcUa
1179...	0.041609	172.21.115.52	172.21.115.181	OpcUa
1179...	0.031577	172.21.115.52	172.21.115.181	OpcUa
1179...	0.042635	172.21.115.52	172.21.115.181	OpcUa
1179...	0.006037	172.21.115.52	172.21.115.181	OpcUa
1179...	0.043628	172.21.115.52	172.21.115.181	OpcUa
1179...	0.030520	172.21.115.52	172.21.115.181	OpcUa

Figura 30. Tráfico de datos entre el Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2 mediante el protocolo OPC UA.

En base a los datos obtenidos, se procedió a determinar las estadísticas de jerarquía del protocolo OPC UA entre la Interfaz Gráfica y la tarjeta Raspberry Pi #2. Los resultados indican que se enviaron y recibieron el 100% de los paquetes entre ambos dispositivos, con un total de 27843 paquetes generados durante el proceso de comunicación. Este resultado demuestra la eficiencia y seguridad del protocolo de comunicación OPC UA, ya que no se produjo pérdida de información. Esto se muestra en la Figura 31.

	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets
net	100.0	27843	100.0	4198913	26 k	0
Internet Protocol Version 4	100.0	27843	9.3	389802	2460	0
Transmission Control Protocol	100.0	27843	13.3	556860	3515	0
OpcUa Binary Protocol	100.0	27843	77.5	3252251	20 k	0
			64.2	2695391	17 k	27843

Figura 31. Estadísticas de jerarquía entre el Computador Industrial y la Raspberry Pi B+ #2 – contenedor 2 mediante el protocolo OPC UA.

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El uso del estándar IEC 61499 y los contenedores ha permitido el desarrollo de sistemas de control flexibles e interoperables, capaces de integrar y gestionar de manera eficiente diferentes dispositivos y sistemas de control. En este sentido, se ha demostrado que es posible automatizar de manera rápida y sencilla procesos productivos mediante la implementación de un computador industrial Siemens de la serie ET 200SP y tarjetas Raspberry Pi B+ utilizando un único lenguaje de control. Esto ha permitido unificar controladores lógicos programables de alta calidad, como el computador industrial, con controladores de bajo costo como las tarjetas Raspberry Pi, ofreciendo una solución estable, segura y eficiente para la automatización de procesos industriales.
- Los resultados obtenidos en el análisis de los tiempos de ejecución y paro de los contenedores Docker que contienen aplicaciones desarrolladas bajo el estándar IEC 61499 sugieren que la virtualización de los procesos a través de contenedores puede proporcionar un ahorro significativo de tiempo y recursos en la ejecución de sistemas de control. Debido a que la virtualización de procesos permite que las aplicaciones de control con sus dependencias se detengan y se ejecuten en cuestión de 1 a 2 segundos. Esto se debe a que la virtualización de procesos permite que cada aplicación de control tenga su propio entorno de ejecución y recursos aislados, lo que evita conflictos y mejora la eficiencia del sistema.
- La implementación de la arquitectura de comunicación en el sistema de control demuestra ser eficaz y cumplir con los estándares necesarios para la Industria 4.0. La eficiencia alcanzada en las pruebas de funcionamiento con el protocolo de comunicación UDP es del 100%, lo que indica que los requisitos de seguridad, calidad y eficiencia se satisfacen de manera óptima. Esto se constató por la ausencia de pérdidas de información en los intercambios de datos entre el Computador Industrial y las tarjetas Raspberry Pi #1 y Raspberry Pi #2, que generaron un total de 12890 y 14096 paquetes, respectivamente.

- El protocolo de comunicación OPC UA tiene una alta eficiencia en la transmisión de información entre los dispositivos. Los resultados obtenidos en las pruebas de funcionamiento muestran que no se produjeron pérdidas de información y que funciona de manera óptima en la comunicación entre las tarjetas Raspberry Pi y la interfaz gráfica teniendo una eficiencia del 100% en la entrega de los paquetes, ya que de los 20370 y 27843 datos generados entre la interfaz y cada una de las tarjetas Raspberry Pi respectivamente no existió errores durante el proceso de comunicación. Esto indica que la elección de los protocolos de comunicación fue acertada y que contribuye a la eficiencia del sistema de control en su totalidad.

4.2 Recomendaciones

- Se recomienda continuar realizando sistemas de control basados en el estándar IEC 61499 y utilizando los controladores más recientes disponibles en el mercado. Esto permite establecer una base sólida para el desarrollo de sistemas de control flexibles e interoperables, lo que facilita la creación de aplicaciones de control para diferentes plataformas. Además, utilizar controladores actualizados y estándares de comunicación aprobados en la industria garantiza una mayor seguridad y calidad en los sistemas de control.
- Es recomendable que, al desarrollar los contenedores, la estructura de los bloques funcionales dentro del software de control se realice teniendo en cuenta todos los aspectos del proceso, ya que una vez que se genere el archivo. Fboot, no se podrá modificar la estructura. Esto garantizará que el contenedor esté optimizado para el proceso y tenga un rendimiento óptimo. Es importante tener en cuenta que la estructura de los bloques funcionales puede afectar el rendimiento y la eficiencia del contenedor, por lo que es esencial realizar un diseño cuidadoso y bien pensado.

- Para garantizar una mejor comprensión y manejo de la interfaz gráfica, se recomienda seguir una normativa en el diseño de la misma. Esto permitirá que la interfaz sea fácil de entender y utilizar para el operador, reduciendo la probabilidad de problemas de visualización y control. Además, el seguimiento de una normativa en el diseño de la interfaz gráfica puede mejorar la eficiencia y productividad del operador al utilizar el sistema. Algunas consideraciones que pueden ser útiles al diseñar la interfaz gráfica incluyen la claridad y legibilidad del texto, la organización y disposición de los elementos de la interfaz y la utilización de iconos y otros elementos visuales para ayudar a la comprensión.

Referencias Bibliográficas

- [1] F. Hofer, M. Sehr, A. Sangiovanni-Vincentelli, y B. Russo, «Industrial control via application containers: Maintaining determinism in IAAS», *Syst. Eng.*, vol. 24, n.º 5, pp. 352-368, 2021, doi: 10.1002/sys.21590.
- [2] M. Sollfrank, F. Loch, y B. Vogel-Heuser, «Exploring Docker Containers for Time-sensitive Applications in Networked Control Systems», en *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, jul. 2019, vol. 1, pp. 1760-1765. doi: 10.1109/INDIN41052.2019.8972165.
- [3] P. D. Cushpa Telenchana, «Comunicación y virtualización de procesos industriales basados en Industria 4.0», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera Ingeniería Electrónica y Comunicaciones, 2020. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/30704>
- [4] J. D. Llamuca, C. A. Garcia, J. E. Naranjo, C. Rosero, E. Alvarez-M, y M. V. Garcia, «Integrating ISA-95 and IEC-61499 for Distributed Control System Monitoring», en *Information and Communication Technologies of Ecuador (TIC.EC)*, Cham, 2020, pp. 66-80. doi: 10.1007/978-3-030-35740-5_5.
- [5] M. V. García, E. Irisarri, y F. Pérez, «Integración Vertical en plantas industriales utilizando OPC UA e IEC-61499», *Enfoque UTE*, vol. 8, n.º 1, pp. 287-299, feb. 2017, doi: 10.29019/enfoqueute.v8n1.132.
- [6] F. Chicaiza, C. A. García, E. X. Castellanos, C. Sánchez, C. Rosero, y M. García, «Arquitectura Flexible Basada en ISA-88 para el Diseño del Diagrama de Control de Ejecución en Aplicaciones Distribuidas mediante IEC-61499», *Enfoque UTE*, vol. 9, n.º 1, pp. 149-165, mar. 2018, doi: 10.29019/enfoqueute.v9n1.186.
- [7] J. Cabral, M. Wenger, y A. Zoitl, «Enable Co-Simulation for Industrial Automation by an FMU Exporter for IEC 61499 Models», presentado en *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2018, vol. 2018-September, pp. 449-455. doi: 10.1109/ETFA.2018.8502654.

- [8] A. Stanciu, «Blockchain Based Distributed Control System for Edge Computing», presentado en Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017, 2017, pp. 667-671. doi: 10.1109/CSCS.2017.102.
- [9] S. P. Bustos Pulluquitin., «Desarrollo de un sistema de control industrial basado en el estándar IEC-61499», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería Industrial en Procesos de Automatización, 2017. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/26656>
- [10] M. V. García Sánchez, «Metodologías para el diseño de sistemas de control distribuido bajo el estándar IEC 61499 aplicados al control de procesos», 2018. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <http://addi.ehu.es/handle/10810/26649>
- [11] D. R. Coello Plaza, «Estudio comparativo de las tecnologías para el desarrollo de software en máquinas virtuales y contenedores en la Empresa Electconstru S.A.», bachelorThesis, Babahoyo: UTB-FAFI. 2022, 2022. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <http://dspace.utb.edu.ec/handle/49000/12549>
- [12] A. I. Basco, G. Beliz, D. Coatz, y P. Garnero, *Industria 4.0: Fabricando el Futuro*. Inter-American Development Bank, 2018.
- [13] J. E. Mellado Aceitón, «El IOT-PLC : una nueva generación de controladores lógicos programables para la industria 4.0», 2020, Accedido: 8 de febrero de 2022. [En línea]. Disponible en: <https://repositorio.uc.cl/handle/11534/50019>
- [14] L. Cruz Salazar, «Industria 4.0 y Sistemas de Producción Ciber-Físicos: Casos de referencia para Latinoamérica», 27 de abril de 2020. doi: 10.13140/RG.2.2.28173.08160.
- [15] R. W. P. Rivas, C. A. F. Diaz, y M. P. B. Acevedo, «PROPUESTAS DE ESTRATEGIAS PARA EL MEJORAMIENTO DEL PROCESO LOGISTICO DEL GRUPO MIZU S.A.S», p. 53.
- [16] C. Burga y J. Sergio, «Propuesta de diseño de una línea de producción para la fabricación de cajas de buzón en la ciudad de Piura para una empresa de prefabricados de

concreto», *Univ. Peru. Cienc. Apl. UPC*, mar. 2013, Accedido: 29 de junio de 2021. [En línea]. Disponible en: <https://repositorioacademico.upc.edu.pe/handle/10757/273504>

[17] W. Dai, V. N. Dubinin, y V. Vyatkin, «Migration From PLC to IEC 61499 Using Semantic Web Technologies», *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 44, n.º 3, pp. 277-291, mar. 2014, doi: 10.1109/TSMCC.2013.2264671.

[18] M. X. López Flores, «Industria 4.0 para la monitorización de un proceso industrial», masterThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Maestría en Automatización y Sistemas de Control, 2019. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/29852>

[19] F. A. Moya Paredes y D. F. Bolaños Paredes, «Automatización del proceso de extracción de hojuelas vegetales para la producción de parafina.», masterThesis, Universidad de las Fuerzas Armadas ESPE. Maestría en Electrónica y Automatización Mención Redes Industriales., 2022. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/34585>

[20] J. L. Campozano Pin y I. T. Velecela Arias, «Diseño e implementación de un sistema de control de temperatura, para prácticas en el laboratorio de automatización industrial, basado en la red PROFINET con un sistema de periferia remota», bachelorThesis, 2021. Accedido: 13 de enero de 2023. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/20833>

[21] Y. A. Areiza, Y. Garzón, y O. Charry, «RASPBerry PI B+. Revisión técnica: guía de uso y programación», mar. 2016. doi: 10.13140/RG.2.1.1063.5921.

[22] R. Segarra y X. Alberto, «Diseño e implementación de un sistema de control y supervisión HMI para máquina barnizadora de la empresa “El Telégrafo”.», p. 113.

[23] J. R. López Yule, «Sistema de control de nivel con supervisión mediante HMI y SCADA para la planta de bombas centrífugas de la Corporación Universitaria Autónoma del Cauca», Thesis, Uniautónoma del Cauca. Facultad de Ingeniería. Programa de Ingeniería Electrónica, 2019. Accedido: 5 de junio de 2022. [En línea]. Disponible en: <http://repositorio.uniautonomo.edu.co:8080/xmlui/handle/123456789/553>

- [24] C. L. Z. Molina, M. E. Arburola, y A. C. Palacios, «DISEÑO DE UN MÓDULO DE ENTRENAMIENTO EN SISTEMAS DE AUTOMATIZACIÓN INDUSTRIAL, INTERFAZ HOMBRE- MÁQUINA Y SISTEMA SCADA CON SIMATIC S7-1200 APLICADO AL CONTROL Y MONITOREO DE UN ASCENSOR DE TRES NIVELES A ESCALA.», p. 128.
- [25] R. W. Emerson, «Automatización de una línea de montaje con cinta transportadora», p. 90.
- [26] Y. Ocaña-Fernández, L. A. Valenzuela-Fernández, y L. L. Garro-Aburto, «Inteligencia artificial y sus implicaciones en la educación superior», *Propósitos Represent.*, vol. 7, n.º 2, pp. 536-568, jul. 2019, doi: 10.20511/pyr2019.v7n2.274.
- [27] S. A. Still Puebla, «Estudio y diseño de una infraestructura del protocolo UDP/IP para comunicaciones ópticas submarinas», *Study and design of a UDP/IP protocol infrastructure for optical underwater communications*, sep. 2018, Accedido: 13 de enero de 2023. [En línea]. Disponible en: <https://repositorio.unican.es/xmlui/handle/10902/14509>
- [28] C. A. García, E. X. Castellanos, M. V. García, C. A. García, E. X. Castellanos, y M. V. García, «Desarrollo de sistemas ciber-físicos de producción para Procesamiento por lotes usando normas IEC-61499 e ISA-88», *Ingeniare Rev. Chil. Ing.*, vol. 27, n.º 3, pp. 443-453, sep. 2019, doi: 10.4067/S0718-33052019000300443.
- [29] I. D. E. Campos, «Optimización de productividad en líneas de producción para sensores automotrices aplicando el Internet de las Cosas en sistemas de visión», p. 113.
- [30] E. Querol, J. A. Romero, A. M. Estruch, y F. Romero, «NORMA IEC-61499 PARA EL CONTROL DISTRIBUIDO. APLICACIÓN AL CNC.», p. 8, 2014.
- [31] L. Ferrarini, C. Veber, y G. Fogliazza, «IEC 61499 implementation of a Modular Control Model for Manufacturing Systems», en *2005 IEEE Conference on Emerging Technologies and Factory Automation*, Catania, Italy, 2005, vol. 1, pp. 315-321. doi: 10.1109/ETFA.2005.1612540.

- [32] G. M. Mafla Medina, «Desarrollo de un sistema distribuido bajo la norma IEC-61499 para control de robot Kuka Modelo Youbot.», mar. 2019, Accedido: 5 de junio de 2022. [En línea]. Disponible en: <http://dspace.esoch.edu.ec/handle/123456789/10286>
- [33] E. Querol, J. A. Romero, A. M. Estruch, y F. Romero, «NORMA IEC-61499 PARA EL CONTROL DISTRIBUIDO. APLICACIÓN AL CNC.», p. 8, 2014.
- [34] C. Rodríguez Hernández, «Personalización de entorno de desarrollo y despliegue sobre contenedores», 2019, Accedido: 5 de junio de 2022. [En línea]. Disponible en: <https://idus.us.es/handle/11441/91381>

Anexos

Anexo 1

Creación de contenedores Docker en las tarjetas Raspberry Pi B+

1. Asegúrate de tener Docker instalado en tu Raspberry Pi. Si no lo tienes instalado, puedes seguir estas instrucciones: <https://docs.docker.com/engine/install/debian/>
2. Descarga o clona el repositorio de la aplicación FORTE que deseas ejecutar en un contenedor. Por ejemplo, si deseas ejecutar la aplicación "Hello World" de FORTE, puedes clonar el repositorio siguiendo estas instrucciones: https://github.com/4SICS/HelloWorld_4DIAC.git
3. Accede al directorio del repositorio descargado y crea un archivo "Dockerfile" con el siguiente contenido:

```
FROM arm32v7/debian:buster-slim

# Install FORTE dependencies
RUN apt-get update && apt-get install -y \
    autoconf \
    automake \
    build-essential \
    cmake \
    libtool \
    pkg-config \
    uuid-dev \
    libboost-all-dev \
    libpthread-stubs0-dev \
    libpthread-stubs0 \
```

```
libpthread-stubs0-dev \  
libpthread-stubs0 \  
libpthread-stubs0-dev \  
libpthread-workqueue0 \  
libpthread-workqueue-dev \  
libpq-dev \  
libxml2-dev \  
libxslt-dev \  
libssl-dev  
  
# Compile and install FORTE  
COPY . /usr/src/forte  
WORKDIR /usr/src/forte  
RUN ./build.sh  
  
# Run FORTE  
CMD ["/usr/local/bin/4diacRTE", "-c", "HelloWorld_4DIAC.fbc"]
```

4. Construye el contenedor usando el comando "docker build":

```
docker build -t forte-hello-world .
```

5. Ejecuta el contenedor con el comando "docker run":

```
docker run -it forte-hello-world
```

Anexo 2

Creación de la aplicación FORTE para la comunicación OPC UA

1. Descargue FORTE 4Diac en su sistema. FORTE 4Diac está disponible de forma gratuita en la página de descargas de la plataforma 4Diac.
2. Una vez instalado, inicie FORTE 4Diac y cree un nuevo proyecto. Puede hacerlo haciendo clic en "File -> New -> FORTE Project" en la barra de menús.
3. A continuación, agregue un elemento OPC UA al proyecto. Puede hacerlo haciendo clic derecho en el árbol del proyecto y seleccionando "New -> OPC UA Communication Layer".
4. Ahora puede configurar la comunicación OPC UA. En el panel de configuración de la capa de comunicación OPC UA, puede establecer la dirección del servidor OPC UA al que desea conectarse, así como los ajustes de seguridad y autenticación.
5. Una vez configurada la comunicación OPC UA, puede comenzar a desarrollar la lógica de su aplicación. FORTE 4Diac utiliza un lenguaje de programación de bloques visuales para crear la lógica de la aplicación. Puede utilizar los bloques OPC UA proporcionados por FORTE 4Diac para enviar y recibir datos a través de la comunicación OPC UA.
6. Finalmente, compile y ejecute su aplicación. Puede hacerlo haciendo clic en "Build -> Build All" y luego en "Execute -> Start FORTE" en la barra de menús.

Anexo 3

Creación de FBs (Bloques Funcionales) en 4DIAC IDE

1. Inicie FORTE 4Diac y cree un nuevo proyecto. Puede hacerlo haciendo clic en "File -> New -> FORTE Project" en la barra de menús.
2. A continuación, agregue un nuevo bloque funcional al proyecto. Puede hacerlo haciendo clic derecho en el árbol del proyecto y seleccionando "New -> FB Type".
3. Una vez creado el bloque funcional, puede comenzar a diseñar la lógica del bloque. FORTE 4Diac utiliza un lenguaje de programación de bloques visuales para crear la lógica de la aplicación. Puede arrastrar y soltar bloques de programación desde el panel de bloques disponibles y conectarlos para crear la lógica del bloque.
4. Una vez diseñada la lógica del bloque, puede configurar los parámetros de entrada y salida del bloque. Los parámetros de entrada y salida son variables que el bloque utiliza para comunicarse con el resto de la aplicación. Puede configurar los parámetros de entrada y salida haciendo clic derecho en el bloque y seleccionando "Edit -> Parameters".
5. Finalmente, puede probar el bloque funcional para asegurarse de que funciona correctamente. Para hacerlo, puede crear una aplicación de prueba que utilice el bloque funcional y ejecutarla en FORTE 4Diac.