



**UNIVERSIDAD TÉCNICA DE AMBATO**

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

**CARRERA DE INGENIERÍA EN ELECTRONICA Y COMUNICACIONES**

**Tema:**

---

**SERVIDOR DE AGREGACIÓN OPC UA BASADO EN LA NUBE PARA  
CONTROL DE PROCESOS INDUSTRIALES USANDO LA NORMA IEC-  
61499**

---

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

**ÁREA:** Electrónica

**LÍNEA DE INVESTIGACIÓN:** Tecnología de la información y sistemas de  
control

**AUTOR:** Henry José Pastuña Pullutasig

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

**Ambato - Ecuador**

**marzo - 2023**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: **SERVIDOR DE AGREGACIÓN OPC UA BASADO EN LA NUBE PARA CONTROL DE PROCESOS INDUSTRIALES USANDO LA NORMA IEC-61499**, desarrollado bajo la modalidad Proyecto de Investigación por el señor Henry José Pastuña Pullutasig, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que 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. Ayala Baño Elizabeth Paulina

**TUTORA**

## AUTORÍA

El presente Proyecto de Investigación titulado: SERVIDOR DE AGREGACIÓN OPC UA BASADO EN LA NUBE PARA CONTROL DE PROCESOS INDUSTRIALES USANDO LA NORMA 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.



---

Henry José Pastuña Pullutasig

C.C 180479420-2

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.



---

Henry José Pastuña Pullutasig

C.C 180479420-2

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Henry José Pastuña Pullutasig, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **SERVIDOR DE AGREGACIÓN OPC UA BASADO EN LA NUBE PARA CONTROL DE PROCESOS INDUSTRIALES USANDO LA NORMA 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 Presidenta del Tribunal.

Ambato, marzo 2023.

-----

Ing. Pilar Urrutia, Mg.

**PRESIDENTA DEL TRIBUNAL**

-----

Ing. Pamela Castro

**PROFESOR CALIFICADOR**

-----

Ing. Franklin Salazar

**PROFESOR CALIFICADOR**

## **DEDICATORIA**

El presente trabajo dedico a mis queridos padres Segundo José Pastuña y Martha Patricia Pullutasig, quienes siempre han sido mi fuente de inspiración y motivación en los momentos difíciles. Su amor y apoyo incondicional me ha permitido llegar hasta aquí. También agradezco a mis humanos quienes me dieron su apoyo para seguir adelante

A mis amigos, por su apoyo incondicional y por hacerme reír en los momentos más difíciles. A todas las personas que de alguna manera han contribuido a la realización de este trabajo, les agradezco sinceramente.

Henry José Pastuña Pullutasig

## **AGRADECIMIENTO**

Quisiera expresar mi más sincero agradecimiento a todas las personas que han contribuido a la realización de este trabajo.

En primer lugar, a mis padres por su amor incondicional, por su confianza en mí y por su apoyo incansable a lo largo del proceso.

A mi tutora, por su valiosa orientación, por su paciencia y dedicación, y por guiarme en el camino correcto para la elaboración de este trabajo.

A mis amigos, por su apoyo, amistad y compañía en los momentos difíciles. A todas las personas que han colaborado conmigo directa o indirectamente, por su tiempo y esfuerzo, agradezco sinceramente su valiosa contribución. Sin su ayuda y apoyo, este trabajo no habría sido posible.

Henry José Pastuña Pullutasig

## ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR .....	iv
APROBACIÓN TRIBUNAL DE GRADO .....	v
DEDICATORIA .....	vi
ÍNDICE DE FIGURAS.....	x
INDICE DE TABLAS .....	xi
RESUMEN EJECUTIVO .....	xii
ABSTRACT.....	xiii
CAPITULO I.....	1
MARCO TEORICO.....	1
1.1    Tema de Investigación.....	1
1.2    Antecedentes Investigativos .....	1
1.2.1    Contextualización del Problema .....	4
1.2.2    Fundamentación Teórica.....	6
1.2.2.1    SIMATIC ET 200SP Open Controller PC2 .....	6
1.2.2.2    Descripción general de OPC UA.....	8
1.2.2.3    Servidor de agregación OPC UA .....	9
1.2.2.4    Modelo de información .....	10
1.2.2.5    Modelo de espacio de direcciones.....	11
1.2.2.6    Stack de comunicaciones OPC UA .....	12
1.2.2.7    Open62541 .....	13
1.2.2.8    Norma IEC-61499.....	14
1.2.2.8.1    Estructura .....	14
1.2.2.8.2    Modelo de bloque de función .....	15
1.3    Objetivos .....	16
1.3.1    Objetivo general .....	16
1.3.2    Objetivos específicos .....	16
CAPITULO II .....	17
METODOLOGIA .....	17
2.1    Materiales .....	17
2.2    Métodos .....	17
2.2.1    Modalidad de la investigación .....	17
2.2.2    Investigación Bibliográfica.....	17
2.2.3    Investigación Experimental .....	18



2.2.4	Investigación de campo .....	18
2.2.5	Recolección de Información .....	18
2.2.6	Procesamiento y Análisis de Datos .....	18
2.2.7	Desarrollo del Proyecto.....	19
CAPITULO III.....		20
RESULTADOS Y DISCUSIÓN .....		20
3.1	Análisis y discusión de los resultados .....	20
3.2	Desarrollo de la Propuesta.....	21
3.2.1	Etapas del Sistema.....	21
3.2.2	Selección de Software .....	24
3.2.2.1	Selección de las librerías para la comunicación OPC UA .....	24
3.2.2.2	Selección del Entorno de desarrollo .....	25
3.2.3	Selección de Hardware.....	27
3.2.4	Diseño del servidor de agregación OPC UA con la norma IEC 6149929	
3.2.5	Análisis de Bloques Funcionales .....	30
3.2.5.1	Bloque Funcional CLIENT .....	30
3.2.5.2	Bloque Funcional FB E_CYCLE .....	32
3.2.6	Configuración de los FB para la conexión de los clientes y servidores OPC UA.....	33
3.2.7	Implementación del Servidor de agregación OPC UA.....	35
3.2.7.1	Clientes y servidores utilizados en el Sistema .....	36
3.2.7.2	Programación del servidor de agregación OPC UA.....	38
3.3	Validación del sistema.....	42
3.3.1	Tiempo de respuesta del sistema .....	42
3.3.2	Cantidad de datos enviados y recibidos .....	44
3.3.3	Prueba de estabilidad del servidor de Agregación.....	46
3.3.4	Fiabilidad del sistema .....	47
CAPITULO IV.....		50
CONCLUSIONES Y RECOMENDACIONES.....		50
4.1	Conclusiones .....	50
4.2	Recomendaciones.....	51
BIBLIOGRAFÍA .....		53
Anexo 1 .....		56

## ÍNDICE DE FIGURAS

Figura 1 SIMATIC ET 200SP Open Controller PC2 [11].	7
Figura 2 Patrón de servidor de agregación [13].	10
Figura 3 Modelo de Información [13].	11
Figura 4 Estructura de un Objeto en OPC UA [14].	12
Figura 5 OPC UA Communication Stack [4].	13
Figura 6 Modelo de bloque de función del estándar IEC-61499 [7].	15
Figura 7 Esquema del servidor de agregación OPC UA.	20
Figura 8 Esquema general de los clientes y servidores agregados.	22
Figura 9 Diagrama de Clases del Servidor de agregación	23
Figura 10 Bloque funcional CLIENT [18].	31
Figura 11 Bloque Funcional E_CYCLE [19].	33
Figura 12 Esquema Servidor de Agregación OPC.	35
Figura 13 Dirección IP de los servidores Agregados.	37
Figura 14 Modelo de información de los clientes Agregados.	37
Figura 15 Modelo de información de los Servidores Agregados.	38
Figura 16 Bloques Funcionales con identificador de variable cliente.	39
Figura 17 Bloques Funcionales con identificador de variable servidor.	39
Figura 18 Programación del servidor de agregación con FB.	40
Figura 19 Funcionamiento de Servidor de agregación.	40
Figura 20 Clientes Agregados en Forte.	41
Figura 21 Servidores Agregados en Forte.	41
Figura 22 Datos obtenidos con UAexpert.	42
Figura 23 Tiempo de respuesta hacia el cliente 1	43
Figura 24 Tiempo de respuesta hacia el cliente 2	44
Figura 25 Numero de datos Enviados y recibidos	45
Figura 26 Porcentaje de datos perdidos	45
Figura 27 Datos enviado en un tiempo prolongado	47
Figura 28 Integración de la librería Open62541 en Forte	56
Figura 29 Ejecución de Forte con la librería Open62541	57
Figura 30 Configuración del equipo con las librerías de Open62541	57

## INDICE DE TABLAS

**Tabla 1.-** Tabla comparativa de programas que utilizan la Norma IEC 61499..... 25

## RESUMEN EJECUTIVO

El objetivo principal de este proyecto es mejorar la automatización de los procesos industriales mediante la optimización y simplificación de la producción. Esto se logra mediante una eficiente comunicación entre los diferentes procesos, lo cual es esencial para el control y el monitoreo de las operaciones industriales. Para alcanzar esta meta, se utilizó la Norma IEC-61499 junto con el protocolo OPC UA para diseñar una arquitectura de comunicación basada en modelos OPC UA cliente/servidor. Esto permitirá una comunicación bidireccional entre los diferentes dispositivos y sistemas que conforman la infraestructura industrial, permitiendo un control en tiempo real y una monitorización continua de los procesos.

Se implementó un servidor de agregación OPC UA utilizando un controlador lógico programable como soporte. Esta implementación permitirá acceder a los datos de varios servidores y clientes OPC UA agregados de manera centralizada, reduciendo costos y disminuyendo la complejidad de implementación. El servidor de agregación será una gran ayuda para el control de procesos industriales, permitiendo que aplicaciones TI y OT actúen sobre la misma fuente de datos directamente a través de la red troncal OPC UA. Esto significa que se podrá tomar decisiones y realizar ajustes en tiempo real, de manera ágil y precisa, logrando una mayor eficiencia y eficacia en la producción.

Además, con la utilización de un servidor de agregación OPC UA, se podrá tener una visibilidad global de todos los procesos industriales, lo cual permitirá identificar cualquier problema o ineficiencia en tiempo real y tomar medidas rápida y eficazmente.

**Palabras clave:** Norma IEC 61499, OPC UA, Servidor de agregación, centralización, procesos industriales

## ABSTRACT

The main objective of this project is to improve the automation of industrial processes by optimizing and simplifying production. This is achieved through efficient communication between the different processes, which is essential for the control and monitoring of industrial operations. To achieve this goal, the IEC-61499 Standard was used together with the OPC UA protocol to design a communication architecture based on OPC UA client/server models. This will allow bidirectional communication between the different devices and systems that make up the industrial infrastructure, allowing real-time control and continuous monitoring of processes.

An OPC UA aggregation server was implemented using a programmable logic controller as support. This implementation will allow data from multiple aggregated OPC UA servers and clients to be accessed in a centralized manner, reducing costs and decreasing implementation complexity. The aggregation server will be a great help for industrial process control, allowing IT and OT applications to act on the same data source directly through the OPC UA backbone. This means that decisions and adjustments can be made in real time, in an agile and precise manner, achieving greater efficiency and effectiveness in production.

In addition, with the use of an OPC UA aggregation server, it will be possible to have global visibility of all industrial processes, which will make it possible to identify any problem or inefficiency in real time and take action quickly and efficiently.

**Keywords:** IEC 61499 Standard, OPC UA, Aggregation server, centralization, industrial processes

# CAPITULO I

## MARCO TEORICO

### **1.1 Tema de Investigación**

Servidor de Agregación OPC UA basado en la nube para control de procesos industriales usando la norma IEC-61499

### **1.2 Antecedentes Investigativos**

En el año 2021 los autores Jácome Sagñay y Fernando Javier en el artículo “Estudio comparativo mediante simulación de las variantes de la tecnología OPC-UA y su utilización en controladores industriales”. Se lleva a cabo un análisis en el que se compararán diferentes opciones del protocolo de comunicación OPC UA a través de una simulación entre un cliente y un servidor OPC UA. También se llevó a cabo una observación de las comunicaciones entre un sistema SCADA y un controlador industrial basado en PC. Para el estudio, se probaron algunos escenarios basados en la realidad utilizando máquinas virtuales y la aplicación OPC UA. Los resultados obtenidos de esta investigación permitirán determinar el ancho de banda utilizado por las aplicaciones OPC UA para un tipo de dato determinado y evaluar la posibilidad de reemplazar la tecnología OPC anterior utilizada en las comunicaciones SCADA, que no es independiente del sistema operativo y carece de las características de seguridad actuales [1].

En el año 2021 los autores García Sánchez, Marcelo Vladimir, Bustos Pulluquitin y Sergio Patricio en el artículo “Desarrollo de un sistema de control industrial basado en la norma IEC-61499”, En este proyecto se determina el diseño de una red de control con algunos nodos que permiten incorporar algoritmos distribuidos de manera sencilla y centralizada. El uso de la norma IEC-61499 ayudo a obtener todas estas características de diseño. La programación distribuida es ligeramente más fácil gracias a los modelos del sistema. Como resultado de este proyecto, se demostraron las ventajas de este tipo de programación y las posibilidades que ofrecen las herramientas de software actualmente disponibles para el modelado e implementación de algoritmos según la norma IEC-61499 [2].

En el año 2020 los autores Chacón Ramírez, Cardillo Albarrán, y Uribe Hernández, por medio de la publicación “Industria 4.0 en América Latina: Una ruta para su implantación”. Se llevó a cabo una investigación sobre las industrias 4.0, ya que es un término relativamente nuevo que promete grandes beneficios. En países que están a la vanguardia de la tecnología, como Europa, Estados Unidos y Asia, las grandes industrias lideran la implementación de las industrias 4.0 siguiendo estándares de automatización. Sin embargo, las pequeñas y medianas empresas (PyMEs) tienen una gran desventaja, ya que a menudo carecen de conocimientos sobre las nuevas tecnologías y corren el riesgo de quedarse rezagadas en la revolución industrial. En América Latina, la situación es aún más grave, ya que la mayoría de las industrias no están automatizadas y muchas PyMEs se enfrentan a un gran riesgo debido a la revolución industrial. Como resultado, este trabajo identificó los requisitos que deben cumplir las empresas para ser consideradas parte de las industrias 4.0 en América Latina [3].

En el año 2020 los autores Santosh Panda, Mainak Majumder, Lukasz Wisniewski y Jurgen Jasperneitey mediante el artículo “Real-time Industrial Communication by using OPC UA Field Level Communication”. Esta investigación fue llevada a cabo con el fin de examinar la importancia de las comunicaciones en tiempo real en el campo de las industrias. Esto es muy importante ya que a medida que las industrias se actualizan con las nuevas tecnologías de Ethernet en tiempo real, se ha encontrado que los controladores de distintos proveedores son incompatibles entre sí. En la era de la Industria 4.0, se necesita un cambio para solucionar este problema de interoperabilidad. Como solución a esta falta de compatibilidad, se propone utilizar OPC UA como un protocolo de comunicación en el campo junto con el estándar de redes TSN [4].

En el año 2020 los autores Selvine Mathias, Sebastian Schmied y Daniel Grossmann en el artículo “An Investigation on Database Connections in OPC UA Applications”. Se investigó el uso de la arquitectura unificada (OPC UA) como un protocolo de comunicación que ha ayudado a impulsar el IoT en los últimos años. Muchos casos de uso en el ámbito de la fabricación demuestran la gran interoperabilidad y fortalezas de la conectividad multiplataforma de OPC UA. Sin embargo, para consolidar su lugar

en el mundo del IoT, OPC UA necesita ser capaz de cerrar ciertas brechas, como la interconexión entre múltiples motores de bases de datos en diferentes redes. Este documento investigó la dinámica de agregación de OPC UA que involucra a múltiples servidores de bases de datos en un entorno específico. Los clientes externos al sistema no pueden acceder a las bases de datos principales en redes restringidas, como plantas de producción. En estos casos, los servicios OPC UA pueden proporcionar a los clientes una vista selectiva pero imperativa de la información de estas bases de datos [5].

En el año 2018 los autores Gerardo González y Francisco Rodríguez Permuy por medio del artículo “Automatización de una planta industrial de alimentación mediante control distribuido”. Se indica que para lograr la automatización en las industrias 4.0, es importante tener en cuenta la logística, ya que le permite controlar procesos como la llenado, velocidad, clasificación, transporte y empaquetado. Estos procesos son cruciales para garantizar la sostenibilidad de las industrias. Para mejorar la revolución industrial 4.0, se debe considerar la implementación de automatizaciones, y para ello, el uso de controladores distribuidos es la opción más adecuada para las plantas industriales del sector de servicios de alimentación [6].

En el año 2017 los autores Marcelo Garcia, Edurne Irisarri y Federico Perez por medio de artículo “Integración Vertical en plantas industriales utilizando OPC UA e IEC-61499”. Se presenta una forma de utilizar OPC UA y la norma IEC-61499, ya que en la actualidad los sistemas de automatización industrial deben enfrentar desafíos nuevos al tratar de satisfacer las necesidades de los mercados altamente competitivos. En la automatización industrial, estos desafíos ayudan a crear sistemas de automatización basados en sistemas ciber-físicos de producción (CPPS). Estos permiten a los sistemas tradicionales de adquisición de datos y a los sistemas inteligentes de datos extraer información para mejorar el rendimiento de los sistemas productivos. Como resultado, esta investigación llevará a cabo aplicaciones bajo la norma IEC-61499, lo que permitirá un intercambio de datos entre las capas más altas utilizadas en el estándar industrial OPC UA y el nivel de plata, ofreciendo mecanismos que permiten la integración eficiente de recursos de diferentes dispositivos [7].



En el año 2017 los autores García Sánchez, Marcelo Vladimir, Bustos Pulluquitin y Sergio Patricio por medio del artículo “Desarrollo de un sistema de control industrial basado en el estándar IEC-61499”. En este estudio se llevó a cabo la implementación de un sistema de control basado en la norma IEC-61499, utilizando el entorno de programación 4DIAC (4DIACIDE) y el run-time FORTE (4DIAC-RTE) en dispositivos embebidos de bajo costo. El objetivo era evaluar los beneficios de utilizar esta norma en sistemas de control y verificar su funcionamiento en este tipo de equipos [8].

En el año 2014 los autores Daniel Großmann, Markus Bregulla, Suprateek Banerjee y Roland Braun en el artículo “OPC UA server aggregation — The foundation for an internet of portals”. Los autores presentan la implementación de un sistema de control basado en la norma IEC-61499, utilizando herramientas de programación de 4DIAC y un ambiente de ejecución de FORTE en dispositivos de bajo costo. Esto se hizo para evaluar los beneficios de la norma IEC-61499 en los sistemas de control. Además, discuten cómo OPC UA mejora el OPC tradicional al ser más fácil de simular en sistemas operativos no basados en Microsoft y al tener un modelo de seguridad mejorado. Para abordar el problema de la conectividad en la malla resultante, se propone el concepto de agregación de servidores conectados a dispositivos en un entorno de automatización industrial. Finalmente, se presenta un prototipo basado en esta arquitectura como prueba de concepto [9].

### **1.2.1 Contextualización del Problema**

Los avances tecnológicos han llevado a las industrias a implementar cambios en sus cadenas de producción para mejorar su competitividad en el mercado. Esto incluye la integración de comunicaciones fluidas tanto verticales como horizontales, y un control descentralizado de procesos [1]. Estos sistemas se han propagado en todas las industrias ya que utilizan servidores OPC distribuidos en distintos puntos [2]. Todo esto conlleva a que los sistemas de automatización industrial estén evolucionando y necesiten implementar nuevas cadenas de producción para mantener su competitividad en el mercado. Esto incluye la integración de modelos de comunicación vertical y horizontal fluidos, así como el control descentralizado de procesos. Los sistemas de

control distribuidos se utilizan para vincular procesos industriales complejos, que constan de flujos de datos divididos en conjuntos de procesos gestionados por algoritmos de control. Sin embargo, estos sistemas pueden ser muy complejos debido a la necesidad de utilizar varios servidores OPC distribuidos en diferentes puntos del proceso industrial, lo que puede dificultar que la información esté disponible para los clientes de nivel superior en el momento necesario [3].

En Ecuador, los sistemas de control industriales suelen estar basados en tecnologías de transmisión de datos que no son compatibles entre sí, lo que dificulta la interoperabilidad entre controladores de diferentes fabricantes [4]. En el ámbito industrial, es fundamental que se integren adecuadamente las diferentes capas de la jerarquía de automatización, lo que implica la necesidad de intercambiar información entre ellas [5]. Por ende, OPC UA es una tecnología que puede ser usada en diferentes plataformas y es escalable, lo que la hace fundamental para el Internet industrial [4].

Para solucionar el problema de la conectividad entre diferentes sistemas de automatización, se propone el uso de un servidor de agregación que permita la integración de datos de distintos servidores en un único lugar. De esta manera, se puede facilitar el acceso de los clientes a la información relevante y centralizar, normalizar y enriquecer la información proveniente de los servidores subyacentes. Además, el uso de este tipo de servidor ayuda a minimizar el número de conexiones necesarias para los dispositivos con recursos limitados [6]. Por consiguiente, Se señala que OPC UA es una opción ideal como estándar de comunicación para usarse en servidores de agregación debido a su independencia de plataforma y habilidad para conectarse entre distintos productos de diferentes proveedores. Esto permite utilizar servicios en la nube como Amazon Web Services, Google Cloud Platform o cualquier nube privada [4].

Se ha demostrado que la integración de los sistemas de control es crucial para la eficiencia en las industrias. Por lo tanto, se ha implementado OPC UA como un estándar de comunicación para garantizar la interconexión de distintos productos de diversos fabricantes. Además, se ha utilizado un servidor de agregación para centralizar, normalizar y enriquecer la información de los servidores subyacentes y hacerla disponible para clientes de nivel superior, como aplicaciones de TI y en la

nube. Esto permite que los sistemas de control proporcionen su propia información y que esta sea accesible para diferentes aplicaciones de OT (Operation Technologies) como HMI (Human-Machine Interface), SCADA (Supervisory Control and Data Acquisition) y sistemas MES (Manufacturing Execution System) [6].

En la Universidad Técnica de Ambato en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial en la carrera de Ingeniería en Electrónica y Comunicaciones como parte del proyecto de investigación de Plataforma de reconfiguración dinámica de control automático de un proceso industrial basado en IEC-61499 y multiagente, Se busca investigar la posibilidad de utilizar nuevas aplicaciones en procesos industriales y evaluar la factibilidad de desarrollar aplicaciones de automatización colaborativas mediante la reconfiguración dinámica de sistemas conforme a la norma IEC-61499, tomando en cuenta las tendencias de la Industria 4.0 en Ecuador.

Con todo ello se pretende realizar el siguiente trabajo de investigación en el cual se creará un servidor de agregación OPC UA, con la ayuda de la norma IEC-61499 y usando protocolos propios de la Industria 4.0. Ya que, en la actualidad, la automatización industrial se está viendo afectada por una serie de desafíos que deben ser abordados para asegurar su éxito en un mercado altamente competitivo. Una de las soluciones más prometedoras es la implementación de sistemas de automatización basados en Sistemas Ciber-Físicos de Producción (CPPS), los cuales permiten integrar sistemas de adquisición de datos tradicionales y sistemas inteligentes de datos para mejorar el rendimiento de los sistemas productivos. Una forma de lograrlo es mediante la utilización de OPC UA y la norma IEC-61499, que ofrecen mecanismos de integración eficiente de recursos de diferentes dispositivos y permiten la reconfiguración dinámica de los sistemas de automatización. Además, el servidor de agregación es esencial para la centralización, normalización y enriquecimiento de la información de los servidores subyacentes, haciéndola disponible para aplicaciones de TI, OT y en la nube a través de la red troncal OPC UA.

## **1.2.2 Fundamentación Teórica**

### **1.2.2.1 SIMATIC ET 200SP Open Controller PC2**

El SIMATIC ET 200SP Open Controller PC2 está diseñado para ser utilizado en

aplicaciones de seguridad y cumple con todos los requisitos de seguridad relevantes. Además, es compatible con el controlador abierto ET 200SP (CPU 1515SP PC2 F), lo que le permite ser integrado de forma sencilla en sistemas existentes.

Cuenta con una memoria de trabajo de 1,5 MB para código y 5 MB para datos, lo que le permite almacenar y procesar grandes cantidades de información. Está diseñado para ser utilizado en aplicaciones de seguridad y es compatible con PROFIsafe V2, un estándar de seguridad en la industria [10].

Además, el SIMATIC ET 200SP Open Controller PC2 posee una serie de funciones tecnológicas avanzadas, como Motion Control, regulación, contaje y medición, y tracing. Estas funciones le permiten realizar una amplia variedad de tareas de control y automatización, y le brindan la flexibilidad necesaria para adaptarse a diferentes aplicaciones industriales.



**Figura 1** SIMATIC ET 200SP Open Controller PC2 [11].

Es compatible con aplicaciones ODK para Windows y tiempo real. Esto le permite ser utilizado en entornos informáticos y le brinda la capacidad de procesar y analizar datos en tiempo real. Además, es compatible con OPC UA Server Data Access, Client Data Access y Companion Specifications, lo que le permite integrarse con otros sistemas y dispositivos de forma sencilla.

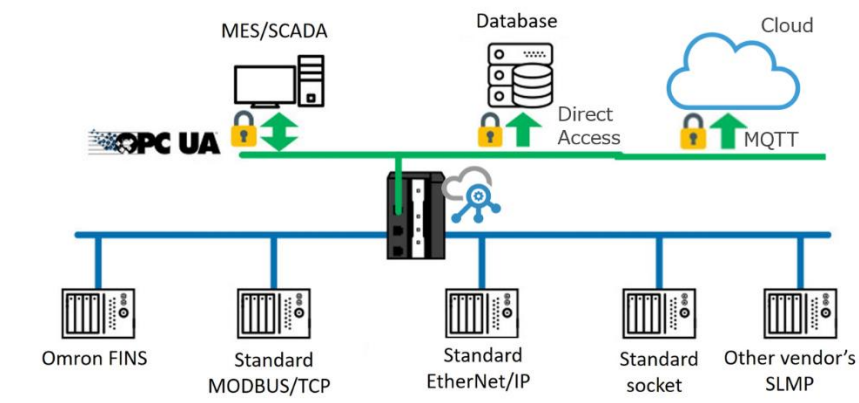
El SIMATIC ET 200SP Open Controller PC2 es compatible con PROFINET IO y es opcionalmente compatible con PROFIBUS DP en modo maestro/esclavo. Esto le permite conectarse a diferentes dispositivos y sistemas de forma rápida y sencilla, y le

brinda la capacidad de transmitir datos de forma segura y confiable como se muestra en la figura 1 [12].

El controlador de software S7-1500 para seguridad es una solución avanzada y versátil para la automatización y el control de procesos industriales. Cuenta con una memoria de trabajo de 1,5 MB para código y 5 MB para datos, y es compatible con PROFI-safe V2. Además, tiene una serie de funciones tecnológicas avanzadas y es compatible con aplicaciones ODK para Windows y tiempo real. Es compatible con OPC UA Server Data Access, Client Data Access y Companion Specifications, y es compatible con PROFINET IO y opcionalmente con PROFIBUS DP en modo maestro/esclavo [11].

### 1.2.2.2 Descripción general de OPC UA

OPC UA es una tecnología de comunicación diseñada para permitir la transferencia de datos en tiempo real de manera sencilla y eficiente. Utiliza protocolos comunes en el ámbito industrial y de sistemas de comunicación para la interconexión de dispositivos y máquinas que aumentan la eficiencia en la producción. Se encuentra en la capa de comunicación y permite obtener información de diversas fuentes en distintos formatos y protocolos, interpretándola para una configuración específica que pueda ser utilizada por el destinatario. OPC UA es autosuficiente, compuesto por diversos estándares que proporcionan un solo espacio de direcciones. Se utiliza principalmente para el modelado y transporte de datos, mediante métodos enfocados en objetos y modelado de información [12].



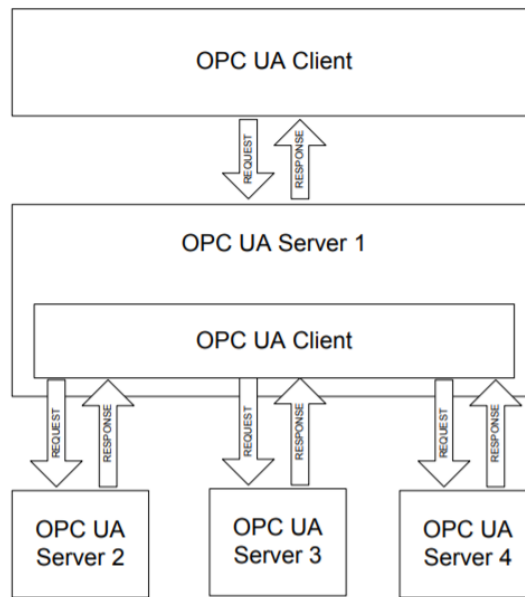
**Figura 2** Conexión de varios softwares con comunicación OPC-UA [12].

OPC UA es una evolución significativa de OPC Classic como se ve en la figura 2 y presenta varios puntos fuertes en comparación con su predecesor. Una de las

principales ventajas de OPC UA es su arquitectura independiente de plataforma. Esto significa que puede ser implementado en diferentes sistemas operativos, arquitecturas de hardware y lenguajes de programación, lo que facilita su adopción en una amplia gama de entornos de automatización industrial. Además, OPC UA tiene una seguridad mejorada, con encriptación de extremo a extremo, autenticación y autorización basadas en certificados digitales, y funciones de auditoría incorporadas. También cuenta con funciones de diagnóstico y mantenimiento que facilitan la administración y actualización de dispositivos de forma remota, y es interoperable con otros sistemas, incluidos los sistemas basados en OPC Classic. En general, OPC UA es una tecnología avanzada y segura que ofrece una amplia gama de características y funcionalidades para una conectividad más sólida y confiable en entornos de automatización industrial [12].

### **1.2.2.3 Servidor de agregación OPC UA**

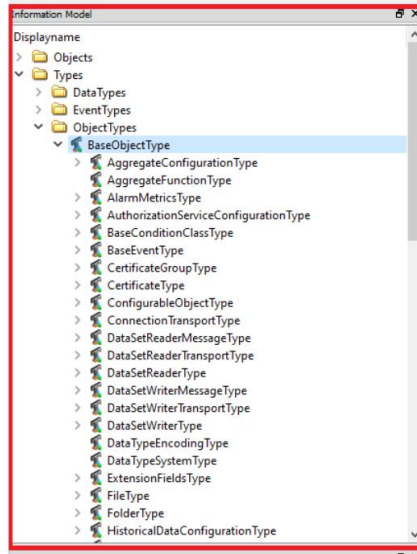
Un servidor de agregación OPC UA es un componente de software que se utiliza para conectar y agregar la información de varios servidores OPC UA en un solo lugar como se muestra en la figura 2. Esto permite que los clientes de nivel superior, como aplicaciones de TI, OT y en la nube, tengan acceso a una fuente centralizada y normalizada de datos provenientes de diferentes dispositivos y sistemas de control de distintos fabricantes. Un servidor de agregación también permite la interoperabilidad entre los servidores subyacentes y puede realizar tareas de enriquecimiento de datos, como la conversión de unidades o la adición de información contextual. Además, un servidor de agregación puede desempeñar un papel importante en la minimización de la cantidad de conexiones que los dispositivos con recursos limitados necesitan establecer. Al utilizar OPC UA como su protocolo de comunicación, un servidor de agregación es independiente de la plataforma y puede ser utilizado en entornos de nube y en la industria [13].



**Figura 3** Patrón de servidor de agregación [13].

#### 1.2.2.4 Modelo de información

La información se adquiere con datos simples que han sido transformados por metadatos, y el modelado de información establece reglas, conceptos y restricciones para verificar la funcionalidad de la información. Los modelos de información se relacionan con un área específica y pueden tener diferentes modelos para diferentes dominios. Habitualmente, estos modelos representan tipos de objetos generales y sus propiedades y operaciones, pero no tratan con instancias individuales. Al modelar objetos del mundo real, se deben establecer restricciones y propiedades del objeto. A través del modelado de información, se establece una relación entre entidades y se describe cómo se encuentran conectadas y relacionadas. Existen tres relaciones genéricas ampliamente utilizadas, pero hay muchas más disponibles si son compatibles con el lenguaje o requerimientos del modelado. Para crear estructuras complejas, es importante tener en cuenta la relación entre objetos y cómo estas estructuras tienen diferentes semánticas y relaciones con entidades físicas.



**Figura 4** Modelo de Información [13].

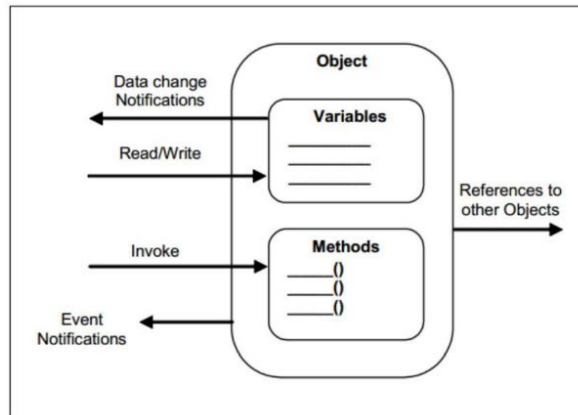
El principal objetivo del modelado de información es crear una estructura de datos organizada y accesible como se muestra en la figura 3. Al tener una organización más estructurada, es más fácil para los usuarios acceder a la información de manera rápida y eficiente [13].

#### **1.2.2.5 Modelo de espacio de direcciones**

El sistema de direcciones OPC UA es el que controla su modelo de información. OPC UA utiliza un modelo de datos basado en objetos con metadatos detallados. Este tipo de sistema de direcciones permite modelar procesos complejos como objetos con variables y métodos. Las variables almacenan información que puede ser utilizada por el cliente y los métodos proporcionan funcionalidad. La figura 3 muestra la estructura



de un objeto OPC UA.



**Figura 5** Estructura de un Objeto en OPC UA [14].

La Figura 4 muestra la estructura de un Objeto en OPC UA. Para un servidor OPC UA, uno de los componentes principales en las especificaciones del espacio de direcciones es el Nodo, que es un conjunto de Atributos y Referencias asociadas a uno de los ocho NodeClasses (Objeto, Tipo de Objeto, Variable, Tipo de Variable, Método, Tipo de referencia, Vista y tipo de datos) establecidos en las especificaciones de OPC UA.

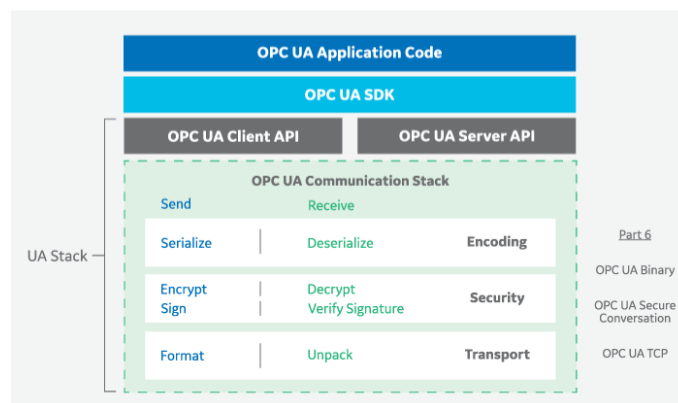
En un servidor OPC UA, todos los datos del espacio de direcciones se construyen a partir de un Nodo base. Al organizar la información, se deben hacer referencia a otros nodos en el espacio de direcciones de un servidor OPC UA. La jerarquía entre las referencias no es absoluta, pero en algunas aplicaciones específicas estas referencias pueden variar.

La referencia de diferentes tipos puede realizar cambios en los nodos básicos. En el lenguaje de programación, los tipos tienen una relación equivalente al concepto de clases. La composición de los tipos incluye clases de nodos y, cuando todo se combina, se presenta como una estructura común en todas las instancias de ese tipo. El modelo de direcciones establece la definición de un conjunto de tipos de Objetos, tipos de Datos y tipos de referencia estándar [14].

#### **1.2.2.6 Stack de comunicaciones OPC UA**

El Stack de comunicaciones OPC UA es un componente fundamental para el correcto funcionamiento de aplicaciones de servidor y cliente OPC UA. Como se puede ver en la Figura 5, proporciona una variedad de funcionalidades de bajo nivel que son esenciales para el envío y recepción de mensajes entre estos sistemas. Para enviar un

mensaje, el Stack se encarga de codificar, proteger y formatear el mensaje de acuerdo con los estándares establecidos por OPC UA. Asimismo, cuando se recibe un mensaje, el Stack se encarga de decodificar, descifrar y descomprimir el mensaje para poder ser procesado adecuadamente por el servidor o el cliente. Es importante destacar que la implementación del Stack de comunicaciones puede variar dependiendo de la aplicación y el entorno en el que se utilice. En la Figura 5 se muestra un ejemplo de cómo las funciones de bajo nivel del Stack pueden asignarse a un protocolo binario basado en TCP. En cualquier caso, lo que siempre permanece constante es la necesidad de contar con un Stack de comunicaciones de alto rendimiento y fiable para garantizar la correcta transmisión de datos en el sistema OPC UA [4].



**Figura 6** OPC UA Communication Stack [4].

### 1.2.2.7 Open62541

OPC UA es un protocolo utilizado en la industria, por lo que se beneficia de la utilización de código abierto, que es más económico y cuya implementación es totalmente gratuita. Esto lleva a la utilización de open62541, una herramienta de código abierto que se puede utilizar con lenguajes de programación como C y C++. Al utilizar open62541, es posible generar las herramientas necesarias para crear un servidor OPC UA dedicado o un cliente OPC UA con los compiladores principales. Además, las librerías generadas por open62541 permiten establecer comunicaciones basadas en OPC UA para sistemas existentes. Una de las principales ventajas de open62541 es su autonomía, ya que no depende de la plataforma OPC UA.

Las librerías generadas por open62541 realizan una configuración directa dentro del stack de protocolos binarios de OPC UA para la comunicación entre servidor y cliente. Actualmente, el perfil de servidores de punto de conexión Micro Embedded es muy

compatible con open62541 y sus diferentes características que se pueden añadir al generar las librerías. Los archivos binarios que contienen el servidor tienen un tamaño muy pequeño, inferior a 100 KB, dependiendo del tipo de modelo de información que contengan [15].

#### **1.2.2.8 Norma IEC-61499**

Una de las principales características de la norma IEC-61499 es proporcionar un alto grado de control distribuido, que permite utilizar sistemas internos existentes para descentralizar los métodos de control y optimizar los sistemas. Además, esta norma establece métodos para sistematizar el progreso del código de manera que sea autónomo y no requiera de un equipo para su ejecución. Los temas principales abordados por esta norma son:

- La portabilidad del programa de computadora debe permitir que se adapte y funcione correctamente en diferentes entornos y sistemas.
- La Configurabilidad es la capacidad de los diferentes instrumentos de procesamiento de ser adaptados y configurados para trabajar en cualquier equipo y con cualquier software.
- La interoperabilidad es la capacidad de los equipos para trabajar de manera integrada y colaborativa para realizar las funciones necesarias de las aplicaciones distribuidas.

Es importante tener en cuenta que estas características son esenciales para garantizar el correcto funcionamiento y la eficiencia del programa de computadora [7].

##### **1.2.2.8.1 Estructura**

El estándar IEC61499 se estructura por medio de cuatro apartados:

- La arquitectura se refiere a las pautas y normas necesarias para la definición y utilización de los diferentes tipos de funcionalidades básicas en el programa de computadora.
- Los requisitos de herramientas de software establecen las exigencias que deben cumplir las herramientas de software utilizadas en la arquitectura definida en

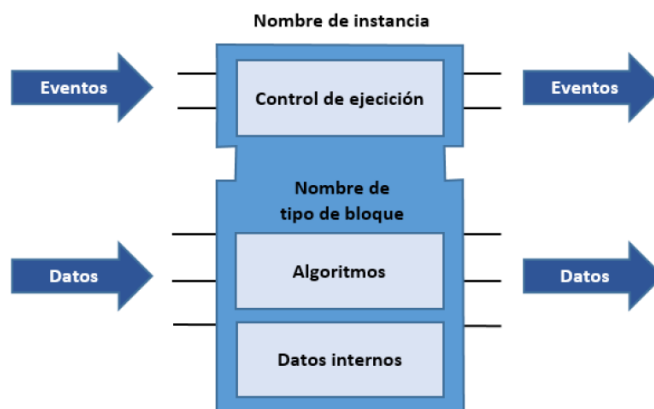
el primer punto.

- El manual informativo proporciona una descripción detallada, a través de ejemplos, de cómo se aplica la arquitectura y otras herramientas de software para cumplir con los estándares.
- Las reglas y perfiles de conformidad especifican los diferentes tipos de dispositivos y herramientas de desarrollo que deben cumplir con ciertos requisitos de compatibilidad para ser utilizados en el programa de computadora.

En resumen, estos aspectos son cruciales para garantizar la coherencia y eficiencia del programa de computadora [7].

#### 1.2.2.8.2 Modelo de bloque de función

Un bloque de función es un componente clave en el modelo de programación de la norma IEC-61499. Se trata de un módulo de código independiente que puede ser utilizado para realizar una tarea específica dentro de un sistema de control. Cada bloque de función tiene sus propias constantes y algoritmo, lo que le permite ser reutilizado en diferentes aplicaciones sin necesidad de modificaciones. Estos módulos son específicos de la norma IEC-61499 y no pueden distribuir otros recursos. En la figura 6 se muestra la estructura básica de un bloque de función, que se compone de una serie de entradas y salidas y una sección de código que define su comportamiento. La utilización de bloques de función es una de las principales ventajas de la norma IEC-61499, ya que permite una mayor modularidad y reutilización del código, lo que facilita la implementación y mantenimiento de sistemas de control [7].



**Figura 7** Modelo de bloque de función del norma IEC-61499 [7].

Los bloques de función se dividen en dos áreas.

La parte superior de los bloques de función se encarga de gestionar y ejecutar el código del bloque a través de las entradas y salidas de eventos. Las entradas de eventos se encuentran en el lado izquierdo y las salidas en el lado derecho. El control de ejecución se encarga de ejecutar el código del bloque. El funcionamiento de los bloques de función es similar al de una máquina de estados, en el sentido de que se encargan de realizar diferentes tareas en función de los eventos de entrada y los estados en los que se encuentren [16].

La parte inferior de los bloques de función se encarga de gestionar las entradas y salidas de datos, que pueden ser cualquier número de entradas y salidas. Cada bloque debe tener al menos una salida para poder difundir los datos. Es fundamental que cada dato esté bien definido para evitar problemas. La utilización de bloques de función es una de las principales ventajas de la norma IEC-61499, ya que permite una mayor modularidad y reutilización del código, lo que hace más fácil la implementación y mantenimiento de sistemas de control. Esto se debe a que cada bloque de función es un módulo de código independiente que puede ser utilizado de manera repetida en diferentes aplicaciones sin necesidad de modificaciones [7].

### **1.3 Objetivos**

#### **1.3.1 Objetivo general**

- Implementar un servidor de Agregación mediante la programación de librerías OPC UA para control de procesos industriales usando la norma IEC-61499.

#### **1.3.2 Objetivos específicos**

- Efectuar el Servidor de Agregación dentro del estándar OPC UA para control de procesos industriales mediante la modificación de librerías del Stack de comunicaciones.
- Integrar el servidor de Agregación para control de procesos industriales dentro del Run-time por medio de la norma IEC-61499.
- Validar el servidor de Agregación para control de procesos industriales que determine su correcto funcionamiento.

## **CAPITULO II**

### **METODOLOGIA**

#### **2.1 Materiales**

Para llevar a cabo este proyecto de investigación, se han utilizado diferentes recursos para recopilar información y apoyar su desarrollo. Estos recursos incluyen libros, artículos científicos, revistas, tesis y fuentes encontradas en Internet. Además, se han utilizado dispositivos electrónicos y software específico para realizar la programación necesaria para el proyecto. Todo esto se han elegido cuidadosamente para asegurar que proporcionan información precisa y relevante para el proyecto. La utilización de esta información recopilada nos ha permitido obtener una visión completa y detallada del tema en cuestión y ha sido clave para el progreso y el éxito del proyecto de investigación.

#### **2.2 Métodos**

##### **2.2.1 Modalidad de la investigación**

Para el desarrollo del presente proyecto, se utilizó la investigación aplicada a las Industrias, debido a que la mayoría de las industrias poseen esquemas descentralizados de comunicaciones para lo cual se pondrá en práctica los conocimientos adquiridos para la implementación del Servidor de Agregación OPC UA se utilizo la norma IEC-61499, para ello se tomaron en cuentas las diferentes modalidades de investigación.

##### **2.2.2 Investigación Bibliográfica**

La investigación bibliográfica ha sido fundamental en este proyecto de investigación, ya que nos ha permitido tener acceso a información importante a través de libros, artículos científicos, tesis, revistas y otras fuentes. Estos recursos nos han ayudado a tener un mayor conocimiento sobre los temas relacionados con OPC UA, el Servidor de Agregación y la norma IEC-61499 y sus aplicaciones. La investigación bibliográfica es crucial para el desarrollo de este proyecto, ya que nos permite comprender mejor estos temas y utilizar la información de manera efectiva en nuestro trabajo.

### **2.2.3 Investigación Experimental**

El enfoque experimental fue elegido para este trabajo debido a que nos permitió evaluar los beneficios de utilizar Servidores de Agregación OPC UA que implementan la norma IEC-61499 en el control de procesos que benefician a las industrias. A través de pruebas y mediciones, pudimos analizar cómo estos servidores influyen en la eficiencia y precisión del control en las industrias. El enfoque experimental nos brindó la oportunidad de obtener datos y resultados concretos que nos permitieron comprender mejor cómo estos servidores pueden mejorar el funcionamiento de los procesos industriales.

### **2.2.4 Investigación de campo**

Para evaluar el rendimiento del Servidor de Agregación OPC UA en diferentes entornos industriales, se llevo a cabo una investigación de campo utilizando herramientas y técnicas específicas. La norma IEC-61499 se utilizo para determinar los beneficios del control de procesos industriales y medir cómo el servidor puede mejorar la eficiencia y la productividad en distintas aplicaciones. La investigación de campo también nos permitio comprender cómo se puede utilizar el Servidor de Agregación OPC UA para optimizar el control y la supervisión de procesos industriales y cómo puede contribuir a una mayor seguridad y fiabilidad en estos entornos.

### **2.2.5 Recolección de Información**

Para obtener datos para nuestro proyecto de investigación, examinamos diversas fuentes de información, incluyendo libros, tesis y artículos científicos publicados en los últimos años y que abordan el uso de servidores de agregación OPC UA y sistemas de control basados en la norma IEC\_61499. Estos serán utilizados como fundamento para la implementación del proyecto que estamos llevando a cabo.

### **2.2.6 Procesamiento y Análisis de Datos**

Para el procesamiento y análisis de datos se realizó las siguientes actividades:

- Organización de la investigación compilada de los servidores subyacentes.

- Reconocimiento de la información escogida.
- Análisis de los datos recopilados del servidor de agregación OPC UA.
- Interpretación mediante la norma IEC\_61499
- Determinación de la idea de solución con la utilización de la norma IEC\_61499.
- Los resultados logrados se añadieron al sistema para observar si el servidor de agregación se encuentra un óptimo funcionamiento.

### **2.2.7 Desarrollo del Proyecto**

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

- Análisis, descripción y evaluación de los diferentes métodos para efectuar el servidor de agregación OPC UA e implementación de la norma IEC\_61499.
- Determinación las diferentes técnicas de programación del servidor de agregación OPC UA y la norma IEC\_61499.
- Comparación los tipos de programación que se puede utilizar para el desarrollo del servidor de agregación OPC UA y la norma IEC\_61499, que se va a utilizar en el proyecto de investigación.
- Efectuar el tipo de programación adecuada para el desarrollo del servidor de agregación OPC UA y la norma IEC\_61499.
- Realización la preparación del hardware y software para el sistema.
- Instalación de los softwares necesarios que se van a utilizar para el funcionamiento del servidor de agregación OPC UA y la norma IEC\_61499.
- Programación del prototipo.
- Elaboración de una interfaz de monitoreo.
- Ejecución de pruebas de funcionamiento del servidor de agregación OPC UA y la norma IEC\_61499, por medio de pruebas adecuadas que comprueben el correcto funcionamiento de cada una de las partes del sistema.
- Análisis de posibles fallas del sistema.
- Elaboración del informe final.



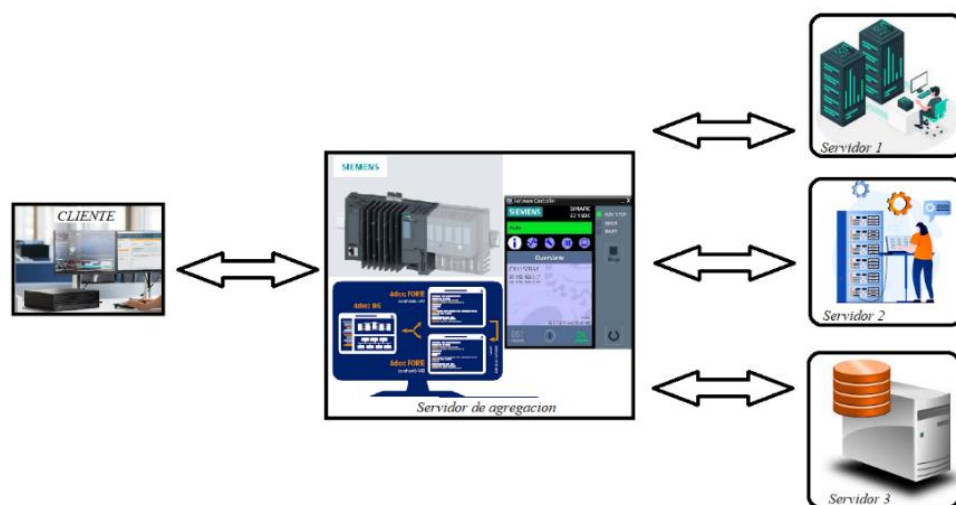
## CAPITULO III

### RESULTADOS Y DISCUSIÓN

#### 3.1 Análisis y discusión de los resultados

En la actualidad, es fundamental que las industrias avancen a la par con las nuevas tecnologías y se adapten a la automatización de sus procesos con el objetivo de optimizar y simplificar la producción. Es esencial que la comunicación entre los distintos procesos sea eficiente para garantizar que todo funcione de manera coherente. Una de las principales razones por las que se llevó a cabo este proyecto fue la reformulación de códigos existentes, lo que permitirá diseñar nuevos códigos para mejorar la comunicación entre los procesos industriales.

Para diseñar la arquitectura de comunicación de este proyecto, se utilizó la Norma IEC-61499 junto con el protocolo OPC UA, muy popular en las industrias 4.0. Esto permitirá crear nuevos modelos en los sistemas distribuidos de conexión. El proyecto se basó en los modelos de comunicación básicos OPC UA cliente/servidor para realizar acciones de control para ello se utilizó softwares para configuración de librerías necesarias para garantizar el correcto funcionamiento del proyecto para el cual se utilizó un controlador lógico programable como se muestra en la figura 7. Luego de la configuración, se llevó a cabo la aplicación con la comunicación de datos en tiempo real y las correspondientes librerías para proporcionar un enlace seguro y fiable.



**Figura 8** Esquema del servidor de agregación OPC UA

**Elaborado por:** Investigador

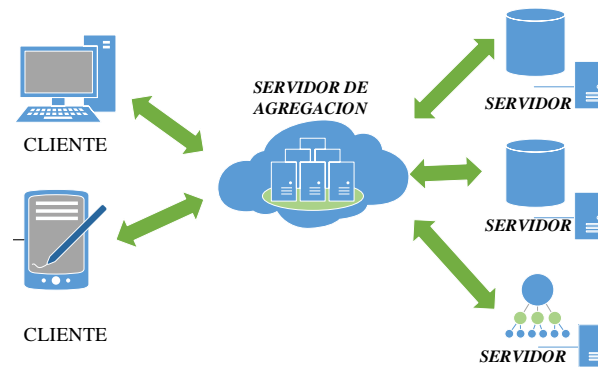
Para implementar el servidor de agregación OPC UA basado en la Norma IEC\_61499, se utilizó un equipo especial como soporte, ya que debe permitir que varios activos proporcionen su propia información, de manera centralizada y se normalice para que esté disponible para clientes agregados. Al realizar la comunicación entre un cliente y el servidor de agregación, por medio de este equipo, se puede acceder a los datos proporcionados por los distintos servidores OPC UA agregados. Estas solicitudes enviadas emitirán una respuesta al cliente. La Norma IEC\_61499 permite programar algoritmos distribuidos de manera sencilla y centralizada a través de su sistema integrado, que puede agregar a los subservidores. Esta estructura es útil para los clientes de nivel superior, que pueden ser servidores OPC UA que realizan diversas tareas en un proceso industrial.

Esto hace que el servidor de agregación sea verdaderamente una ayuda para el control de procesos industriales, ya que todo esto reduciría costos, al disminuir la complejidad de implementación ya que otros sistemas requieren de equipos y softwares especializados para este tipo de trabajos, todo ello será de gran ayuda ya que ahora las aplicaciones de TI, OT pueden actuar sobre la misma fuente de datos que provienen de las máquinas directamente a través de la red troncal OPC UA.

## **3.2 Desarrollo de la Propuesta**

### **3.2.1 Etapas del Sistema**

El servidor de Agregación UA incluye tanto una implementación de servidor como de cliente. El cliente integrado se utiliza para acceder a otros servidores, conocidos como servidores de origen o servidores agregados. Una estructura típica de un servidor agregado se muestra en la figura 8. La función de un servidor agregado es reunir información de uno o más servidores fuente en su propio espacio de direcciones.



**Figura 9** Esquema general de los clientes y servidores agregados.

**Elaborado por:** Investigador

Uno de los principales beneficios de un servidor de agregación es que proporciona un acceso centralizado a varias fuentes de información. Un cliente que necesite obtener información solo debe conectarse a un servidor central en lugar de realizar conexiones y solicitudes de servicio individualmente a cada uno de los servidores de origen. Por esta razón, los servidores de agregación son adecuados para tareas de supervisión y monitoreo, como en los sistemas SCADA, especialmente si el servidor de agregación también implementa los servicios de Alarmas y Condiciones de OPC UA. Además, un servidor de agregación puede actuar como un administrador de seguridad centralizado para el control de conexión y facilita enormemente los procesos de administración y supervisión.

### **Diagrama de clases de un servidor de agregación**

Un servidor de agregación basado en la norma IEC 61499 tendrá un diagrama de clases como se muestra en la figura 9 de la cual se describe cada una de sus partes a continuación:

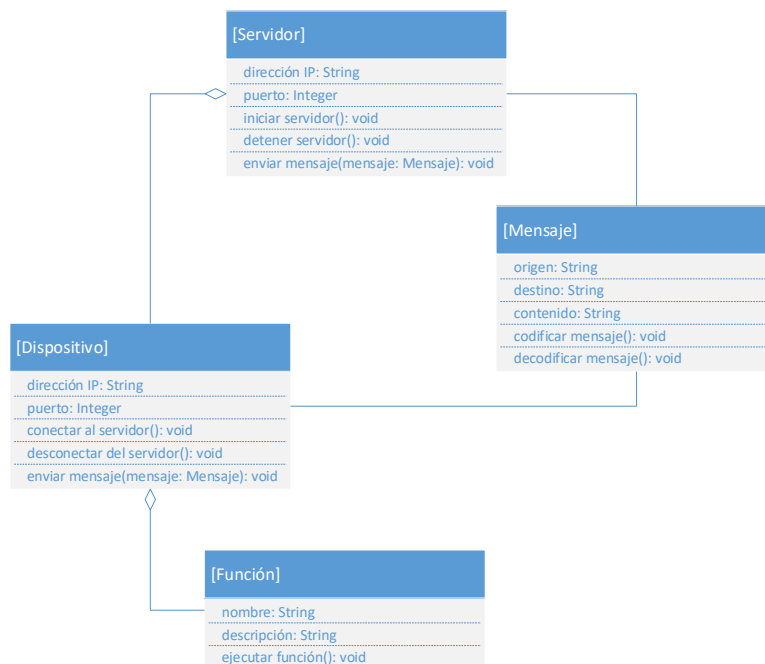
- **Servidor:** Esta clase representaría al servidor de agregación en sí, y podría tener atributos como "dirección IP" y "puerto" y métodos como "iniciar servidor", "detener servidor" y "enviar mensaje".
- **Dispositivo:** Esta clase representaría a los dispositivos que se conectan al servidor, y podría tener atributos como "dirección IP" y "puerto" y métodos como "conectar al servidor", "desconectar del servidor" y "enviar mensaje".
- **Mensaje:** Esta clase representaría a los mensajes que se envían y reciben a

través del servidor, y podría tener atributos como "origen", "destino" y "contenido" y métodos como "codificar mensaje" y "decodificar mensaje".

- Función: Esta clase representaría a las funciones que se ejecutan en los dispositivos conectados al servidor, y podría tener atributos como "nombre" y "descripción" y métodos como "ejecutar función".

### El diagrama tendrá las siguientes relaciones:

- La clase Servidor tendrá una relación de agregación con la clase Dispositivo, ya que un servidor puede contener varios dispositivos.
- La clase Dispositivo tendrá una relación de agregación con la clase Función, ya que un dispositivo puede contener varias funciones.
- La clase Servidor tendrá una relación de asociación con la clase Mensaje, ya que el servidor puede enviar y recibir mensajes.
- La clase Dispositivo tendrá una relación de asociación con la clase Mensaje, ya que los dispositivos pueden enviar y recibir mensajes a través del servidor. Esto significa que el servidor contiene varios dispositivos, que a su vez pueden contener varias funciones, y que tanto el servidor como los dispositivos pueden enviar y recibir mensajes a través de una asociación.



**Figura 10** Diagrama de Clases del Servidor de agregación

**Elaborado por:** Investigador

### **3.2.2 Selección de Software**

Se consideró cada uno de los elementos, sus características y la disponibilidad en el país mediante un análisis técnico. Los detalles de la selección de los elementos para el servidor de agregación OPC UA se describen a continuación.

#### **3.2.2.1 Selección de las librerías para la comunicación OPC UA**

Open62541 es una herramienta útil para implementar de manera gratuita código abierto de OPC UA (arquitectura unificada). Está escrito en lenguajes C99 y C++98 y es muy utilizada debido a que puede ser desarrollada con cualquiera de los compiladores principales. Además, proporciona todos los materiales necesarios para ejecutar servidores y clientes OPC UA dedicados o para establecer comunicaciones basadas en OPC UA de manera efectiva. Open62541 es una librería muy útil para el desarrollo de sistemas que utilizan OPC UA y puede ser muy útil en el ámbito industrial y en la automatización de procesos [17].

Es muy popular en la industria debido a sus siguientes características:

- Es de código abierto: Open62541 es un proyecto de código abierto, lo que significa que es gratuito y puede ser utilizado y modificado por cualquier persona. Esto lo convierte en una opción atractiva para las empresas que buscan una solución de bajo costo para implementar la norma IEC 61499.
- Es de alta calidad: Open62541 es una implementación de alta calidad de la norma IEC 61499 y ha sido probada en diferentes entornos industriales. Esto le da confianza a los usuarios de que es una solución confiable y sólida.
- Es altamente configurable: Open62541 es altamente configurable y puede ser adaptado a diferentes necesidades y requisitos. Esto lo convierte en una solución muy versátil y adaptable.
- Es ampliamente utilizado: Open62541 en las industrias cuenta con una gran

comunidad de usuarios y desarrolladores. Esto hace que sea fácil obtener ayuda en soporte y con ello, haya una gran cantidad de recursos disponibles.

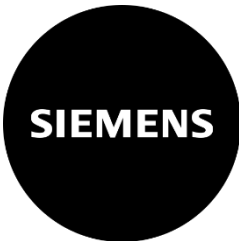
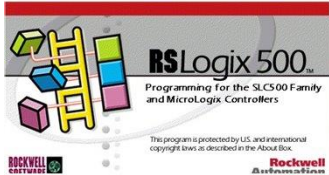
### 3.2.2.2 Selección del Entorno de desarrollo

La norma IEC 61499 se utiliza para definir la arquitectura y el funcionamiento de sistemas de control distribuidos. Una de las formas más comunes de implementar estos sistemas es mediante el uso de bloques funcionales (FB, por sus siglas en inglés). Los FB son elementos de software que implementan una función específica dentro de un sistema de control distribuido y se pueden combinar para crear una aplicación completa.

Algunos programas que utilizan FB para la implementación de la norma IEC 61499 incluyen:

**Tabla 1.-** Tabla comparativa de programas que utilizan la Norma IEC 61499

Programa	Icono	Descripción
4DIAC		<p>4diac es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) de código abierto basado en la norma IEC 61499. 4diac proporciona una amplia gama de FBs predefinidos que se pueden utilizar para crear aplicaciones de control distribuido.</p>
FESTO		<p>FESTO ofrece una amplia gama de FBs predefinidos que se pueden utilizar para crear aplicaciones de control distribuido utilizando la norma IEC 61499. Estos FBs están disponibles para su uso con el software de programación de FESTO, como el entorno de desarrollo integrado (IDE) FESTO Motion Terminal. Algunos de estos productos son de pago y otros están disponibles de forma</p>

		gratuita.
SIMATIC		SIMATIC ofrece una amplia gama de FBs predefinidos que se pueden utilizar para crear aplicaciones de control distribuido utilizando la norma IEC 61499. Estos FBs están disponibles para su uso con el software de programación de SIMATIC, como el entorno de desarrollo integrado (IDE) SIMATIC STEP 7. Estos productos y soluciones son de pago.
RSLogix		RSLogix es un software de programación de PLCs que utiliza FBs para la implementación de la norma IEC 61499. RSLogix proporciona una amplia gama de FBs predefinidos que se pueden utilizar para crear aplicaciones de control distribuido. Es un producto comercial que se vende por un precio en el mercado industrial.

Para el presente proyecto se decidió utilizar 4DIAC-IDE, ya que, en él, se puede desarrollar el código abierto, el cual se destina a sistemas distribuidos en las industrias, también se puede decir que es una aplicación robusta e intuitiva. En este entorno de desarrollo se puede programar mediante la utilización de Bloques de Funciones, y dichos Bloques Funcionales se pueden modificar según las necesidades del proyecto. Todos los programas que se desarrollen en este software se pueden utilizar en los distintos dispositivos de campo los cuales se definen por la norma IEC61499, ya que es compatible con los equipos existentes en este momento.

Para la implementación del sistema de control requería la selección de una herramienta adecuada y, después de considerar varias opciones, se decidió utilizar 4DIA-FORTE. Esta plataforma es independiente de la plataforma en la que se esté trabajando y puede

ser ejecutada en diferentes hardware y sistemas operativos, lo que es muy beneficioso debido a su compatibilidad con muchas tarjetas embebidas y controladores lógicos programables (PLC). Además, 4DIA-FORTE ofrece una estructura de comunicación flexible en la capa de comunicación y una arquitectura escalable, lo que es muy importante para sistemas que pueden requerir más capacidad en el futuro.

Cuando se trata de implementar un sistema de control, es fundamental tener en cuenta la flexibilidad y adaptabilidad de la herramienta utilizada. Al elegir una plataforma que puede ser ejecutada en diferentes entornos hardware y sistemas operativos, se pueden evitar muchos problemas y complicaciones a largo plazo. Esto es especialmente importante en el ámbito industrial, donde se utilizan diferentes tipos de equipos y dispositivos que pueden tener requisitos específicos en términos de hardware y software.

Además, la capacidad de comunicación flexible es esencial para cualquier sistema de control, ya que permite la integración de diferentes dispositivos y sistemas en un entorno de trabajo coherente. Con una estructura de comunicación flexible, es más fácil conectar y controlar diferentes dispositivos y sistemas, lo que puede mejorar la eficiencia y reducir los tiempos de inactividad.

### **3.2.3 Selección de Hardware**

Para seleccionar el hardware adecuado para un servidor de agregación OPC UA, es necesario considerar los siguientes factores:

- Es necesario considerar los requisitos del sistema a la hora de seleccionar el hardware adecuado para un servidor de agregación OPC UA. Esto incluye el tamaño y la complejidad del proceso que se va a controlar, el número de dispositivos de entrada y salida y la cantidad de datos que deben ser procesados.
- El rendimiento del hardware es crucial para asegurar que el servidor de agregación OPC UA pueda procesar y transmitir datos de forma eficiente. Por lo tanto, es importante tener en cuenta el rendimiento al seleccionar el hardware adecuado para el sistema.



- Escalabilidad: Es importante elegir hardware que pueda escalarse fácilmente en caso de que se necesite más capacidad de procesamiento o memoria en el futuro.
- Compatibilidad: Es importante asegurarse de que el hardware elegido sea compatible con el software y los dispositivos que se utilizarán en el sistema.
- Costo: Por último, es importante tener en cuenta el presupuesto disponible y elegir hardware que se ajuste a él sin sacrificar el rendimiento y la calidad.

Es importante tener en cuenta que la selección del hardware adecuado para un servidor de agregación OPC UA puede ser un proceso complejo y requerir conocimientos especializados en tecnología de la información y control de procesos industriales.

Para el presente proyecto se tomó en cuenta el SIMATIC ET 200SP Open PC2 un controlador programable versátil y potente que puede ser utilizado como servidor de agregación OPC UA en diferentes aplicaciones industriales. Una de las principales ventajas del controlador SIMATIC ET 200SP Open PC2 es su capacidad de procesamiento. Cuenta con un procesador de alta velocidad y una memoria de trabajo de gran tamaño, lo que le permite procesar y almacenar grandes cantidades de datos. Esto lo hace ideal para aplicaciones que requieren un rendimiento y una capacidad de procesamiento elevados.

Otra de las ventajas del controlador SIMATIC ET 200SP Open PC2 es su conectividad. Cuenta con una gran cantidad de conectores de entrada y salida y es compatible con diferentes protocolos de red, como PROFINET IO y PROFIBUS DP. Esto le permite conectarse a diferentes dispositivos y sistemas y transmitir datos de forma rápida y segura. Esta conectividad lo hace ideal para aplicaciones que requieren una gran cantidad de interconexión y comunicación.

Además, el controlador SIMATIC ET 200SP Open PC2 cuenta con medidas de seguridad integradas, como la encriptación de datos y la autenticación de usuarios. Esto lo hace ideal para aplicaciones que requieren un alto nivel de seguridad y confidencialidad de los datos.

Es fácil de programar, configurar y cuenta con una interfaz de usuario intuitiva. Esto

lo hace ideal para aplicaciones que requieren una configuración y un uso sencillo. Además, cuenta con un manual de usuario detallado y hay una gran cantidad de recursos y soporte disponibles, lo que facilita su uso y mantenimiento.

### **3.2.4 Diseño del servidor de agregación OPC UA con la norma IEC 61499**

Definir el objetivo del sistema de control: Es importante tener una comprensión clara de lo que se quiere lograr con el sistema de control, ya sea mejorar la eficiencia del proceso, reducir costos o aumentar la calidad del producto final.

- Identificar los componentes del sistema: Se deben identificar y describir todos los componentes del sistema, incluyendo los dispositivos de entrada y salida, los sensores y actuadores, el hardware y el software.
- Diseñar el algoritmo de control: Se debe desarrollar un algoritmo que determine cómo se deben controlar los componentes del sistema para lograr el objetivo deseado.
- Probar y validar el sistema: Es importante realizar pruebas exhaustivas del sistema para asegurarse de que funciona de manera efectiva y estable.
- Implementar y mantener el sistema: Una vez que el sistema de control ha sido diseñado y probado, se debe implementar en el servidor de agregación OPC UA y llevar a cabo el mantenimiento adecuado para garantizar su correcto funcionamiento a largo plazo.

Tomado en consideración estos aspectos para el diseño del servidor de agregación, se utilizó el controlador SIMATIC ET 200SP Open PC2 como base para la implementación. La norma IEC 61499, utilizada en 4DIAC, se utilizó para programar los bloques funcionales y agregarlos al servidor. Estos bloques funcionales utilizaron la librería Open62541, con la ayuda de 4Diac forte, para incluirse en el stack de comunicación de OPC UA y con todo ello lograr la integración el servidor de Agregación para control de procesos industriales dentro del Run-time con la ayuda de la norma IEC-61499.

El modelo de información utilizado en OPC UA es esencial para identificar las

direcciones de los datos y garantizar la fiabilidad del servidor de agregación. Este modelo establece una estructura jerárquica de nodos y atributos, que permite la identificación unívoca de cada dato dentro del sistema. Además, OPC UA utiliza un sistema de nombres que permite la referenciación de los datos de manera clara y sencilla.

La librería Open62541 es una implementación de código abierto del estándar OPC UA, que se utiliza para la comunicación entre dispositivos en sistemas industriales. Esta librería es esencial para garantizar el correcto funcionamiento del servidor de agregación, ya que proporciona las herramientas necesarias para implementar el protocolo OPC UA y establecer la comunicación entre los distintos elementos del sistema.

### **3.2.5 Análisis de Bloques Funcionales**

Los bloques funcionales son una de las herramientas de programación utilizadas en el entorno de desarrollo de aplicaciones industriales 4diac (FORTE). Los bloques funcionales son componentes reutilizables que se pueden utilizar para crear aplicaciones y programas en 4diac. Cada bloque funcional representa una función o un proceso particular y puede ser conectado a otros bloques funcionales para crear un flujo de datos y un programa completo.

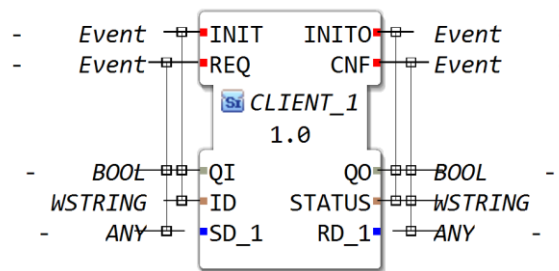
Los bloques funcionales en 4diac se dividen en diferentes categorías, como entrada/salida, procesamiento, lógica y comunicación, entre otras. Cada categoría incluye diferentes bloques funcionales específicos que se pueden utilizar para realizar diferentes tareas y procesos. Los bloques funcionales en 4diac son muy versátiles y pueden ser utilizados para crear aplicaciones para una gran variedad de sistemas y dispositivos industriales [18].

#### **3.2.5.1 Bloque Funcional CLIENT**

El bloque funcional CLIENT en 4diac es un componente que se utiliza para establecer conexiones y comunicaciones con servidores OPC UA. Este bloque funcional permite a una aplicación cliente en 4diac conectarse a un servidor OPC UA y enviar y recibir

datos a través de la conexión.

El bloque funcional CLIENT en 4diac cuenta con diferentes entradas y salidas como se ve en la figura 11 la cuales se pueden utilizar para configurar y controlar la conexión con el servidor OPC UA. Por ejemplo, se pueden utilizar las entradas para especificar la dirección del servidor y las credenciales de autenticación, y las salidas para recibir los datos enviados por el servidor y para obtener el estado de la conexión [18].



**Figura 11** Bloque funcional CLIENT [18].

El bloque funcional CLIENT en 4diac está compuesto por las siguientes partes:

#### Eventos de Entrada

- **INIT** se emplea para poner en marcha un servicio específico que se ofrece a través de un bloque, como la transmisión de datos mediante un enlace serial.
- **REQ** se utiliza para solicitar información a un agente externo, por ejemplo, cuando se necesita acceder a los datos de un dispositivo externo.

#### Eventos de Salida

- **INITO** indica que el SIFB ha terminado su proceso de inicio, aunque es importante tener en cuenta que este evento no garantiza que el servicio se haya iniciado correctamente. Por tanto, se utiliza una salida de "STATUS" para comprobar el estado del servicio.
- **CNF** se produce cuando el bloque ha completado la transmisión de una solicitud a un agente externo, confirmando que se ha procesado una solicitud de lectura de un punto de entrada/salida específica por el subsistema del controlador de entrada/salida.

#### Datos de entrada

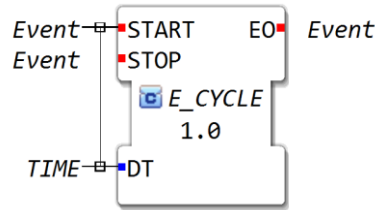
- **QI** es una entrada booleana que se utiliza junto con el evento INIT para indicar si un servicio debe iniciarse (cuando QI es "true") o finalizarse (cuando QI es "false").
- Las entradas de datos **SD\_1** a **SD\_N** se utilizan para enviar y recibir información específica del tipo de servicio proporcionado por el bloque. Por ejemplo, en la lectura de datos de diferentes salidas de un dispositivo, estos datos pueden ser de diferentes tipos.
- **ID** es la parte del bloque funcional que se encarga de establecer y mantener la conexión con el servidor OPC UA también se encarga de enviar y recibir los datos a través de la conexión en la cual se incluye la dirección del servidor, las credenciales de autenticación para la respuesta del servidor.

#### Datos de salida

- **QO (BOOL)** se utiliza para indicar la finalización del servicio para cualquier evento de entrada. Por ejemplo, después de la inicialización de un evento INIT, un valor de "true" indica que la inicialización se ha completado correctamente, mientras que un valor de "false" indica que la inicialización ha fallado.
- La salida **STATUS** se puede establecer con cualquier evento de entrada y se utiliza para proporcionar el estado de procesamiento del evento seleccionado. Cuando se inicializa un servicio con el evento INIT y se ejecuta con éxito, se establece la salida STATUS.
- **RD\_1, ... RD\_N (ANY)** se utilizan para transmitir los datos recibidos de confirmaciones e indicaciones. El número de salidas y sus tipos de datos son específicos para el servicio proporcionado por el bloque, al igual que las entradas **SD\_1 ... SD\_N** [19].

#### 3.2.5.2 Bloque Funcional *FB E\_CYCLE*

En la figura 12 el FB *E\_CYCLE* es una función bloque (FB) en el lenguaje de programación IEC 61499 utilizado en sistemas de control de procesos y automatización de fábricas. La función bloque *E\_CYCLE* se utiliza para realizar una acción específica cada vez que se completa un ciclo de ejecución en el programa [20].



**Figura 12** Bloque Funcional E\_CYCLE [19].

La función bloque E\_CYCLE consta de las siguientes partes:

- El evento **START** se utiliza una entrada de enable para habilitar o deshabilitar la función bloque E\_CYCLE.
- El evento **STOP**, al recibir una señal en true, realizará la parada del bloque funcional, mientras que si recibe una señal en false, el bloque continuará cumpliendo su función.
- **EO** es el evento de salida el cual se desencadenará periódicamente con ello enviará una señal de salida a otras partes del programa o a dispositivos externos.
- El parámetro **DT** (tiempo de ciclo) se utiliza para configurar la frecuencia con la que se ejecuta la función del FB. Esta función se encarga de establecer un tiempo determinado para la realización de una acción específica.

### 3.2.6 Configuración de los FB para la conexión de los clientes y servidores OPC UA

OPC UA (Unified Architecture) es un estándar de comunicación industrial que se utiliza para interconectar diferentes sistemas y dispositivos en una red de automatización. 4diac es un entorno de desarrollo de aplicaciones para sistemas de automatización que incluye soporte para la comunicación OPC UA [21].

Los bloques funcionales de comunicación OPC UA en 4diac son componentes lógicos que se pueden utilizar para establecer y mantener la comunicación con un servidor OPC UA y para publicar o suscribirse a variables o estados en el servidor. Los bloques funcionales de comunicación OPC UA se pueden utilizar en aplicaciones 4diac para implementar diferentes funcionalidades y servicios OPC UA, como la publicación de datos de proceso o la suscripción a eventos o notificaciones.

Para configurar y utilizar los bloques funcionales de comunicación OPC UA en 4diac, es necesario especificar la dirección del servidor OPC UA y el identificador de la variable o el estado en el servidor al que se desea publicar o suscribir. Estos parámetros se especifican en la entrada de datos del bloque funcional y se utilizan para establecer la conexión con el servidor y para determinar qué variable o estado se desea publicar o suscribir.

Además, es posible configurar opciones adicionales en los bloques funcionales de comunicación OPC UA para personalizar el comportamiento de los mismos.

En el contexto de la norma IEC 61499, un identificador de variable se utiliza para identificar y acceder a una variable específica dentro de un FB. En el caso de la comunicación OPC UA, estos identificadores de variables se utilizan para permitir que los FB se comuniquen entre sí y con otros sistemas a través de una red de comunicación basada en OPC UA.[22]

Los identificadores de variables se pueden asignar a las variables dentro de un FB de diferentes maneras, dependiendo del entorno de desarrollo y del programa utilizado. Algunas opciones comunes incluyen:

- Asignar manualmente un identificador de variable: En algunos entornos de desarrollo, es posible asignar manualmente un identificador de variable a una variable específica dentro de un FB. Esto se hace a través de una propiedad o configuración del FB.
- Utilizar un estándar de nomenclatura: En algunos casos, es posible utilizar un estándar de nomenclatura para asignar identificadores de variables a las variables dentro de un FB. Por ejemplo, se podría utilizar una nomenclatura basada en el prefijo "ns=1;s=FB\_Name.Variable\_Name" para identificar una variable dentro de un FB específico.

Para el presente proyecto se utilizó el parámetro con un bloque funcional de comunicación en la norma IEC 61499 con asignación manual de un identificador de

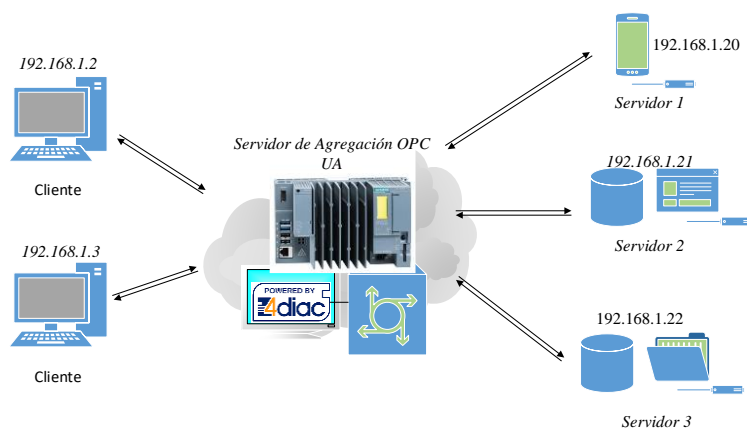
variable.

**"opc\_ua[ACCIÓN;PUNTO FINAL;PAR1;PAR2;...]"**

- **ACCIÓN:** esta es la acción que se realizará con el bloque funcional de comunicación. Puede ser una de las siguientes opciones: LEER, ESCRIBIR, CREAR\_METHOD, CALL\_METHOD, SUSCRIBIR, CREAR\_OBJETO, ELIMINAR\_OBJETO, CREAR\_VARIABLE o ELIMINAR\_VARIABLE.
- **PUNTO FINAL:** es la dirección del servidor OPC UA al que se accederá para realizar la acción. Debe incluir el protocolo de comunicación, la dirección IP y el puerto. Debe terminar con un carácter '#'.  
Ejemplo: opc\_ua[LEER;opc.tcp://192.168.1.20:4840/;#]
- **PAR1;PAR2;...:** son parámetros adicionales que se utilizarán para la acción especificada. Pueden incluir información de navegación y node ID para acceder a nodos específicos en el servidor OPC UA, y otros parámetros específicos para cada acción. [23]

### 3.2.7 Implementación del Servidor de agregación OPC UA

Para la implementación del sistema de tomo en consideración el siguiente esquema de la figura 13.



**Figura 13** Esquema Servidor de Agregación OPC

**Elaborado por:** Investigador

En el presente proyecto se realizó la simulación de dos clientes y tres servidores para lo cual el servidor de agregación OPC UA se utilizó para recopilar datos de los tres



servidores y proporcionar acceso a estos datos a los clientes de manera segura y fiable. Esto es útil ya que, por ejemplo, si los tres servidores están encargados de realizar diferentes tareas y los clientes necesitan acceder a los datos generados por ellos para realizar un proceso de toma de decisiones.

Para garantizar un rendimiento y fiabilidad adecuados, el servidor de agregación OPC UA se utilizó el SIMATIC ET 200SP Open Controller PC2 como hardware principal. Este controlador cuenta con un procesador potente y una amplia memoria para permitir el procesamiento de grandes cantidades de datos y la ejecución de tareas de procesamiento de datos y control. Además, cuenta con una amplia variedad de entradas y salidas digitales y analógicas para permitir la conectividad con una amplia variedad de dispositivos de campo.

Además, es importante considerar la escalabilidad del servidor para asegurar que pueda manejar una carga de trabajo creciente sin perder rendimiento o estabilidad. Utilizando el SIMATIC ET 200SP Open Controller PC2 como hardware principal, se puede garantizar un rendimiento y fiabilidad adecuados para el servidor de agregación OPC UA.

### **3.2.7.1 Clientes y servidores utilizados en el Sistema**

#### **Clientes**

El cliente puede ser cualquier tipo de dispositivo o aplicación que utilice el protocolo OPC UA para acceder a los datos del servidor de agregación. Por ejemplo, el cliente podría ser un PC con una aplicación de monitoreo o un PLC con un programa de control. El cliente utiliza los datos recibidos del servidor para realizar tareas específicas, como la toma de decisiones o el control de dispositivos de campo.

En este caso se utilizó una dos PC como se muestra en la figura 14, para simular los clientes los cuales se utilizan para acceder a los datos recopilados por el servidor y utilizarlos para realizar tareas de toma de decisiones.

```

Adaptador de Ethernet Ethernet:
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::ccd6:7080:cf0d:e69b%2
Dirección IPv4. . . . . : 192.168.1.2
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.1.1

Adaptador de túnel isatap.{2885F1E4-64F0-4F34-BF6E-110A97299F28}:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . :

C:\Users>

Adaptador de Ethernet Ethernet:
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::c4c:c6f8:8e27:b059%3
Dirección IPv4. . . . . : 192.168.1.3
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.1.1

Adaptador de túnel isatap.{2885F1E4-64F0-4F34-BF6E-110A97299F28}:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . :

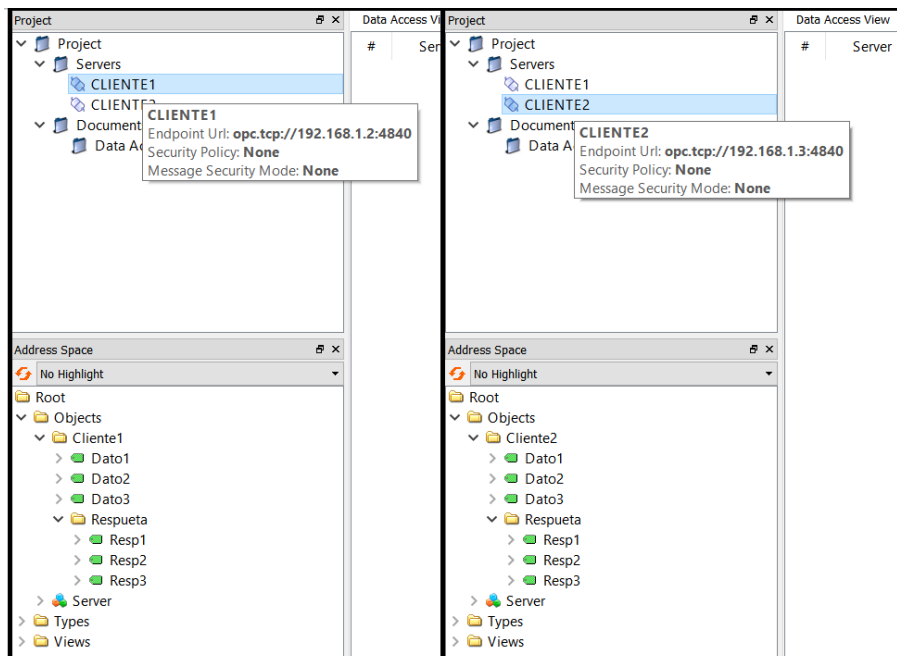
C:\Users>

```

**Figura 14** Dirección IP de los servidores Agregados.

**Elaborado por:** Investigador

Para establecer la conexión entre los clientes y los servidores OPC UA, es necesario especificar la dirección en la que se encuentran los datos tanto para el envío como para la recepción de la respuesta como se ve en la figura 15. Esto se puede hacer utilizando UaExpert ya que es una herramienta de programación especializada en el protocolo OPC UA. Con la ayuda de UaExpert, se pueden fijar las direcciones de los datos y utilizar los bloques funcionales y la configuración de 4DIAC ID para realizar la programación y establecer la conexión entre los clientes y los servidores.



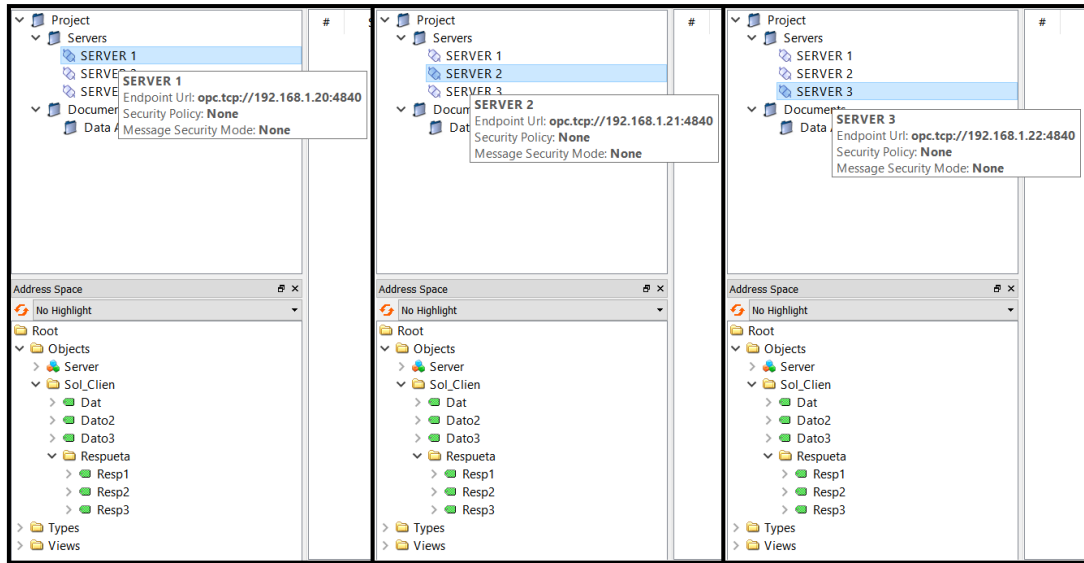
**Figura 15** Modelo de información de los clientes Agregados.

**Elaborado por:** Investigador

## Servidores

Al igual que en el caso de los clientes, para los servidores se utilizó un PC para

representar la forma en la que los datos llegarían y se enviarían a través de la red.



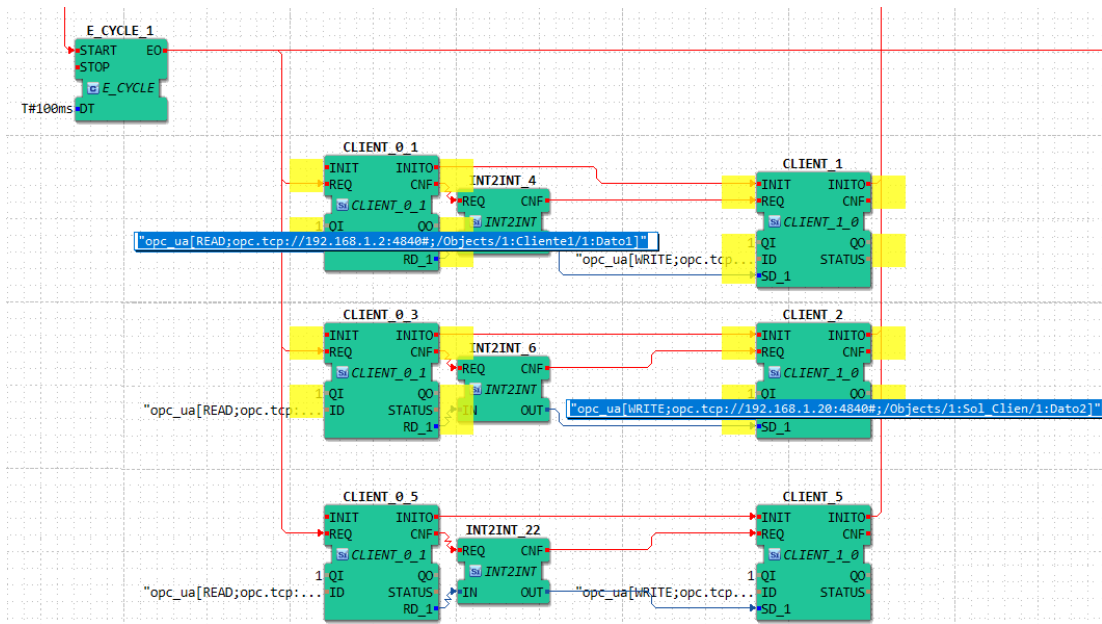
**Figura 16** Modelo de información de los Servidores Agregados.

**Elaborado por:** Investigador

En este caso, el servidor se encarga de recibir los datos en el modelo de información que se visualiza en la figura 16 del servidor de agregación y procesarlos de acuerdo a las tareas asignadas. Una vez que se han procesado los datos, el servidor envía la respuesta al servidor de agregación, que a su vez la distribuye al cliente que ha realizado la solicitud original.

### **3.2.7.2 Programación del servidor de agregación OPC UA**

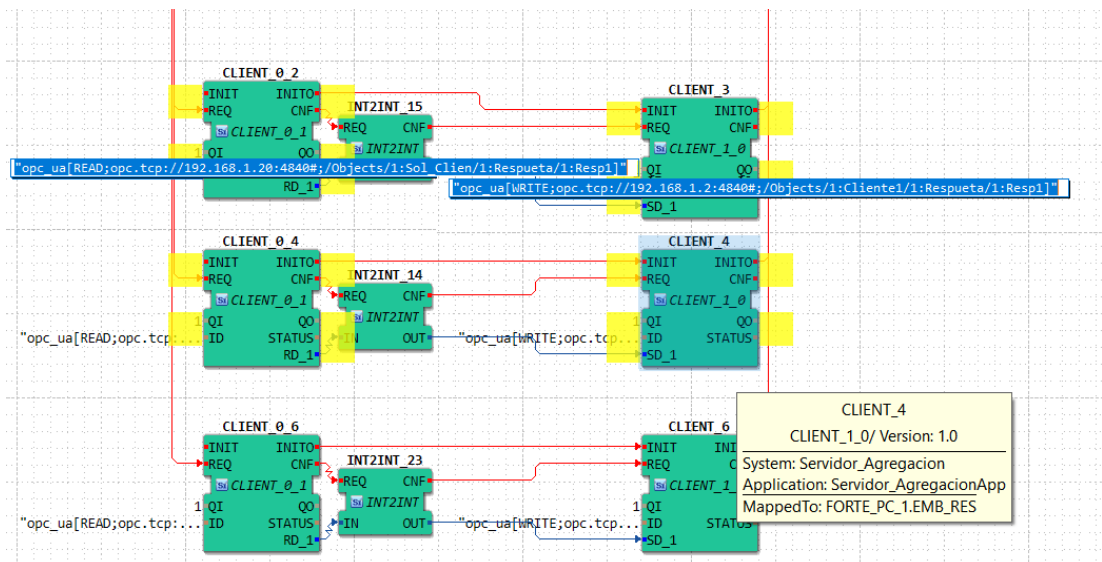
Para la agregación de clientes y solicitudes hacia los servidores se utilizó el parámetro con un bloque funcional de comunicación en la norma IEC 61499 con asignación manual de un identificador de variable como se presenta en la figura 17 y figura 18. Con esto, se puede utilizar servidores de agregación para recopilar y distribuir las solicitudes de los clientes a los servidores correspondientes.



**Figura 17** Bloques Funcionales con identificador de variable cliente.

**Elaborado por:** Investigador

La agregación de servidores y respuestas hacia el cliente es un proceso en el que se recopilan las respuestas de los servidores a las solicitudes de datos y cambios de estado de los clientes y se envían al cliente que realizó la solicitud original.



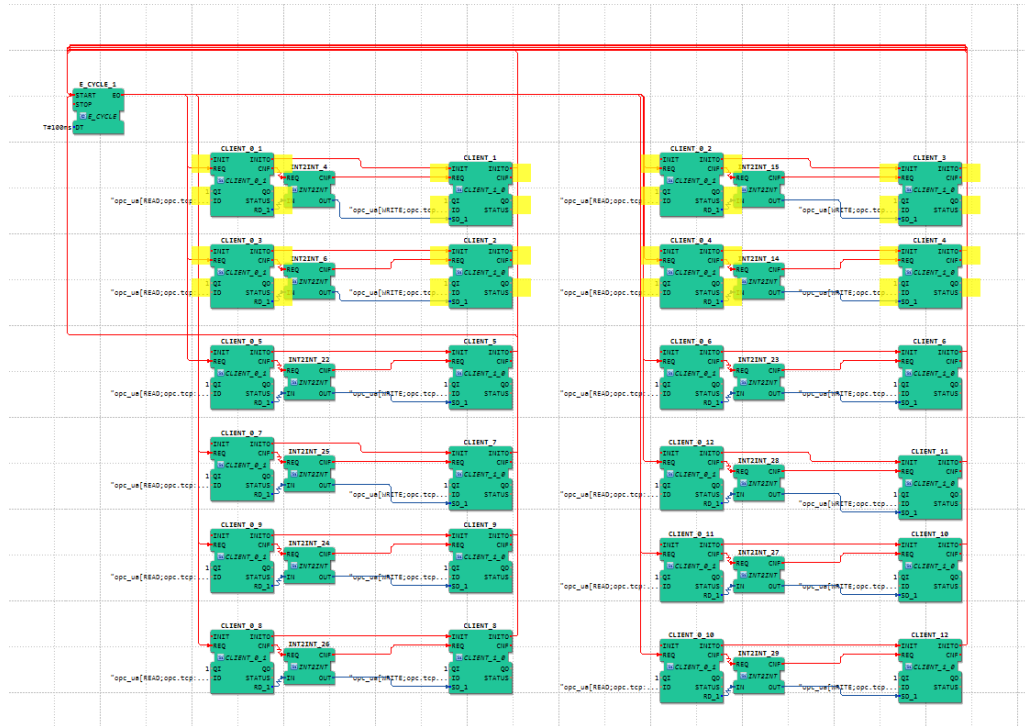
**Figura 18** Bloques Funcionales con identificador de variable servidor.

**Elaborado por:** Investigador

### Servidor de agregación OPC UA

Para el presente sistema, la interfaz principal del servidor de agregación es la parte del sistema que se encarga de gestionar las solicitudes de datos y cambios de estado de los

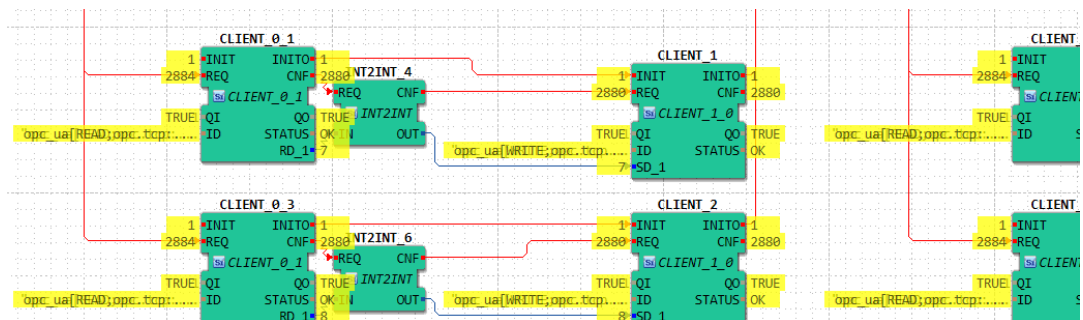
clientes y enviarlos a los servidores correspondientes, lo cual se presenta en la figura 19. También se encarga de recopilar y distribuir las respuestas de los servidores a los clientes que realizaron la solicitud original, para ello de adecuo de la mejor manera los FB en 4DIAC ID con lo cual se puede facilitar la agregación de más clientes y servidores, así como el sistema lo requiera.



**Figura 19** Programación del servidor de agregación con FB.

**Elaborado por:** Investigador

Es importante asegurarse de que el sistema esté funcionando correctamente antes de comenzar a utilizarlo en el control de procesos industriales. Esto implica poner en funcionamiento tanto el 4Diac ID como el 4Diac FORTE al mismo tiempo, con lo cual todo el sistema se establece una conexión estable como se indica en la figura 20.



**Figura 20** Funcionamiento de Servidor de agregación.

## Elaborado por: Investigador

Al tener ambas herramientas de programación en funcionamiento, se puede ver cómo los servidores y los clientes se conectan al sistema sin ninguna dificultad como se ve en la figura 21 figura 22. Esto es esencial para garantizar la estabilidad del sistema y evitar problemas en el futuro. Además, al ver cómo los servidores y los clientes se conectan al sistema, se puede hacer un seguimiento de su funcionamiento y detectar cualquier problema potencial de manera temprana.

```
Connecting to endpoint opc.tcp://192.168.1.2:4840
SecurityPolicy not specified -> use default #None
Security policy None is used to create SecureChannel. Accepting all certificates
TCP connection established
to the server is open
Opened SecureChannel with SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
to the server is open
Endpoint and UserTokenPolicy unconfigured, perform GetEndpoints
Found 1 endpoints
Endpoint 0 has 2 user token policies
Selected Endpoint opc.tcp://192.168.1.2:4840 with SecurityMode None and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
Selected UserTokenPolicy open62541-anonymous-policy with UserTokenType Anonymous and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None

Connecting to endpoint opc.tcp://192.168.1.3:4840
SecurityPolicy not specified -> use default #None
Security policy None is used to create SecureChannel. Accepting all certificates

TCP connection established
to the server is open
Opened SecureChannel with SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
to the server is open
Endpoint and UserTokenPolicy unconfigured, perform GetEndpoints
Found 1 endpoints
Endpoint 0 has 2 user token policies
Selected Endpoint opc.tcp://192.168.1.3:4840 with SecurityMode None and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
Selected UserTokenPolicy open62541-anonymous-policy with UserTokenType Anonymous and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
```

Figura 21 Clientes Agregados en Forte

## Elaborado por: Investigador

```
Connecting to endpoint opc.tcp://192.168.1.20:4840
SecurityPolicy not specified -> use default #None
Security policy None is used to create SecureChannel. Accepting all certificates

TCP connection established
to the server is open
Opened SecureChannel with SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
to the server is open
Endpoint and UserTokenPolicy unconfigured, perform GetEndpoints
Found 1 endpoints
Endpoint 0 has 2 user token policies
Selected Endpoint opc.tcp://192.168.1.20:4840 with SecurityMode None and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
Selected UserTokenPolicy open62541-anonymous-policy with UserTokenType Anonymous and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None

Connecting to endpoint opc.tcp://192.168.1.21:4840
SecurityPolicy not specified -> use default #None
Security policy None is used to create SecureChannel. Accepting all certificates
TCP connection established
to the server is open
Opened SecureChannel with SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
to the server is open
Endpoint and UserTokenPolicy unconfigured, perform GetEndpoints
Found 1 endpoints
Endpoint 0 has 2 user token policies
Selected Endpoint opc.tcp://192.168.1.21:4840 with SecurityMode None and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
Selected UserTokenPolicy open62541-anonymous-policy with UserTokenType Anonymous and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None

Connecting to endpoint opc.tcp://192.168.1.22:4840
SecurityPolicy not specified -> use default #None
Security policy None is used to create SecureChannel. Accepting all certificates
TCP connection established
to the server is open
Opened SecureChannel with SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
to the server is open
Endpoint and UserTokenPolicy unconfigured, perform GetEndpoints
Found 1 endpoints
Endpoint 0 has 2 user token policies
Selected Endpoint opc.tcp://192.168.1.22:4840 with SecurityMode None and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
Selected UserTokenPolicy open62541-anonymous-policy with UserTokenType Anonymous and SecurityPolicy http://opcfoundation.org/UA/SecurityPolicy#None
```

Figura 22 Servidores Agregados en Forte

## Elaborado por: Investigador

Una vez que se puso en funcionamiento el sistema, se utilizó UAexpert para verificar el envío de solicitudes y la recepción de respuestas en la figura 23 se muestra los datos obtenidos. Con esta herramienta, se pudo verificar que el servidor funcionaba de manera confiable y que los clientes y servidores especificados lograron ser agregados de manera exitosa al sistema.

Data Access View								
#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	CLIENT 1	NS1 Numer...	Dato1	7	Int16	11:40:32.904	11:40:32.904	Good
2	CLIENT 2	NS1 Numer...	Dato2	8	Int16	11:40:54.138	11:40:54.138	Good
3	CLIENT 1	NS1 Numer...	Resp1	10	Int16	11:40:33.107	11:40:33.107	Good
4	CLIENT 2	NS1 Numer...	Resp2	11	Int16	11:40:54.404	11:40:54.404	Good

**Figura 23** Datos obtenidos con UAexpert

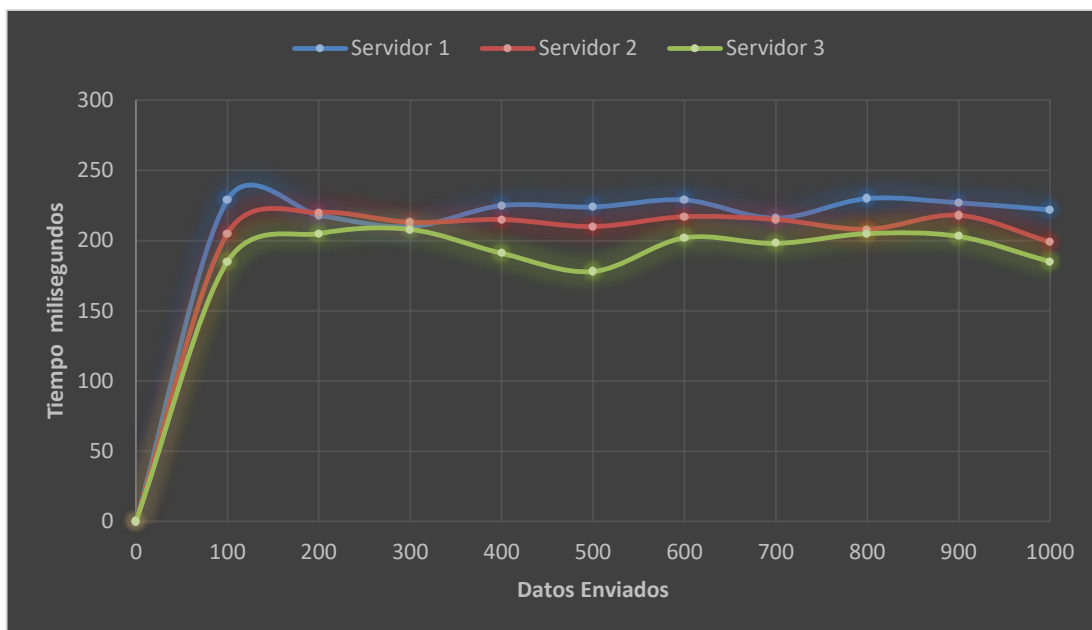
**Elaborado por:** Investigador

Este resultado fue muy importante ya que indica que el servidor de agregación está funcionando de manera eficiente y puede ser utilizado de manera confiable en el control de procesos industriales. Además, el hecho de que los clientes y servidores especificados hayan sido agregados de manera exitosa al sistema indica que el sistema está listo para ser utilizado y es capaz de manejar las solicitudes y respuestas de manera eficiente.

### 3.3 Validación del sistema

#### 3.3.1 Tiempo de respuesta del sistema

En esta prueba de rendimiento, se llevó a cabo el envío de 1000 datos, los cuales son de tipo int16 ya que utilizan 2 bytes de memoria y son comúnmente utilizados para el control de procesos industriales, todo esto con el objetivo de medir el tiempo de respuesta del sistema, desde que se envía la solicitud hasta el momento en el que se recibe la respuesta en el cliente. Es importante tener en cuenta que los datos deben atravesar el servidor de agregación tanto de ida como de regreso, y esto puede afectar el tiempo de respuesta del sistema.

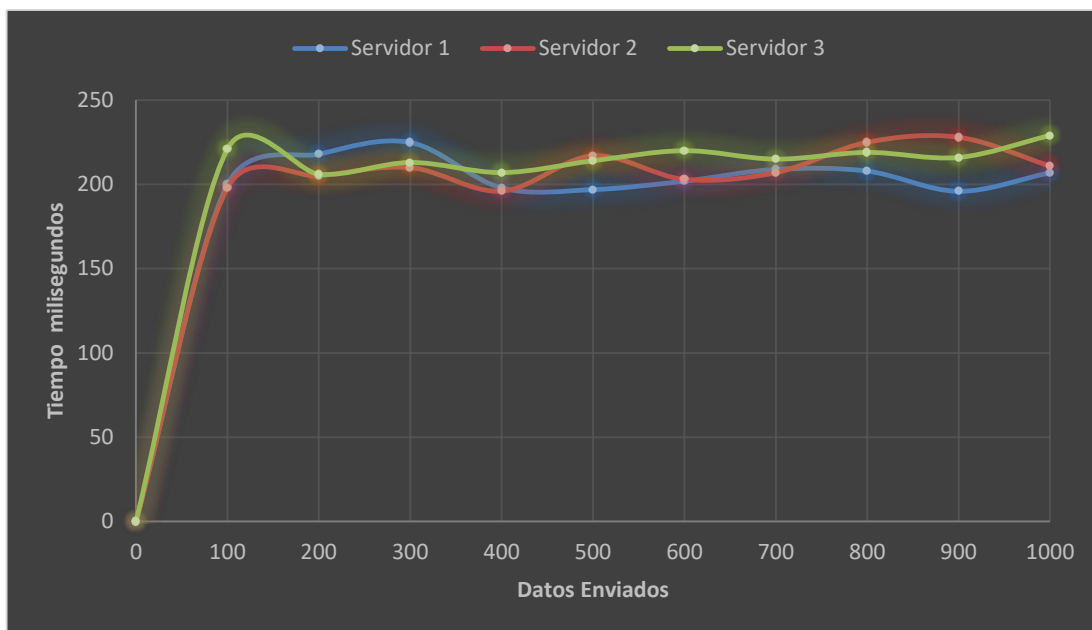


**Figura 24** Tiempo de respuesta hacia el cliente 1

**Elaborado por:** Investigador

En el proceso de realizar las pruebas en dos clientes distintos, se determinó que el tiempo de respuesta promedio en ambos clientes se encontraba entre 185 ms y 233 ms. A partir de estos resultados, se generaron gráficos que muestran la tendencia en cuanto al tiempo promedio de respuesta. Estos gráficos se observan en las figuras 24 y 25, en los cuales se puede apreciar un registro por cada cien datos y se sacó un promedio de tiempo con lo cual se mantuvo una tendencia con el tiempo promedio de 210ms. Es importante mencionar que estos resultados se obtuvieron tanto en el primer cliente como en el segundo, lo cual indica una consistencia en los tiempos de respuesta.





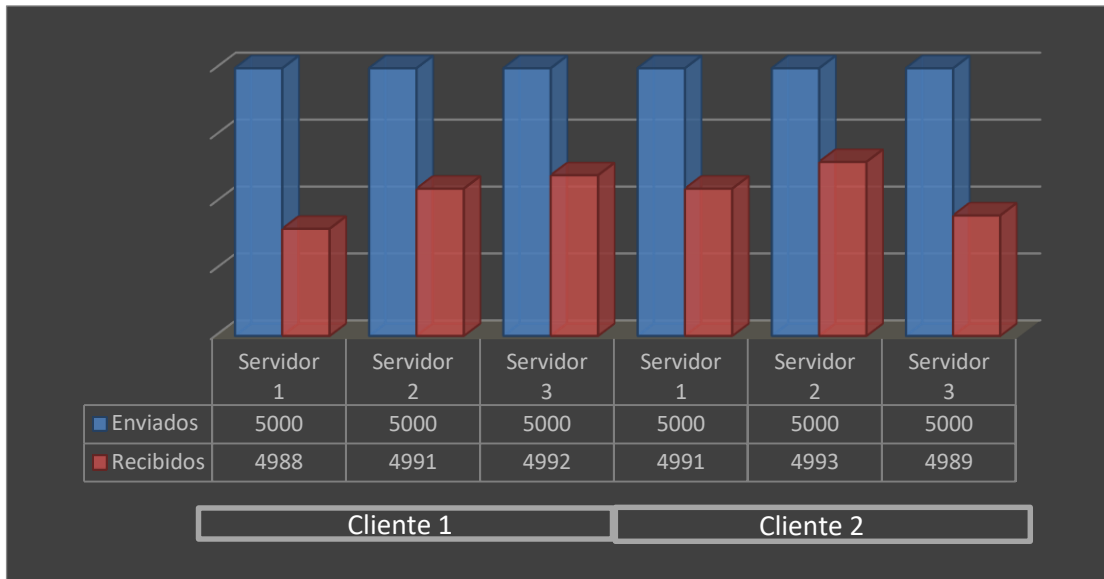
**Figura 25** Tiempo de respuesta hacia el cliente 2

**Elaborado por:** Investigador

Además, es importante considerar que el resultado obtenido en esta prueba, solo es un indicador del rendimiento del sistema bajo una carga específica. Por lo tanto, es importante realizar múltiples pruebas bajo diferentes condiciones, para obtener una visión completa del rendimiento del sistema.

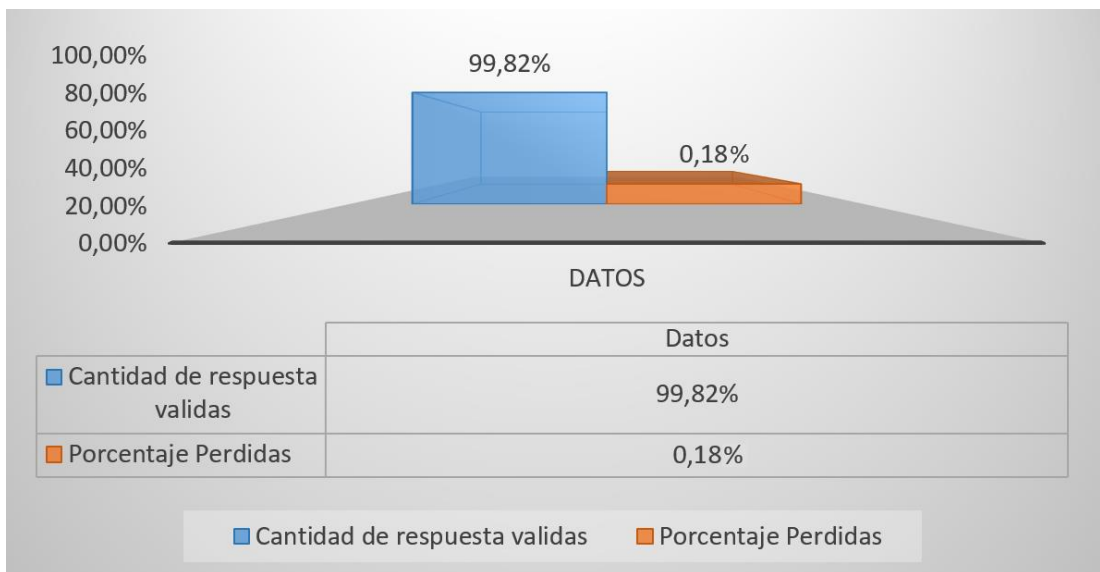
### 3.3.2 Cantidad de datos enviados y recibidos

La prueba de sobrecarga realizada tiene como objetivo evaluar la capacidad del sistema para manejar una gran cantidad de datos de manera eficiente. Es importante determinar esto ya que, en algunas aplicaciones, el sistema puede recibir una gran cantidad de datos simultáneamente, y es necesario que el sistema pueda procesar y gestionar estos datos sin sufrir interrupciones o caídas.



**Figura 26** Numero de datos Enviados y recibidos

**Elaborado por:** Investigador



**Figura 27** Porcentaje de datos perdidos

**Elaborado por:** Investigador

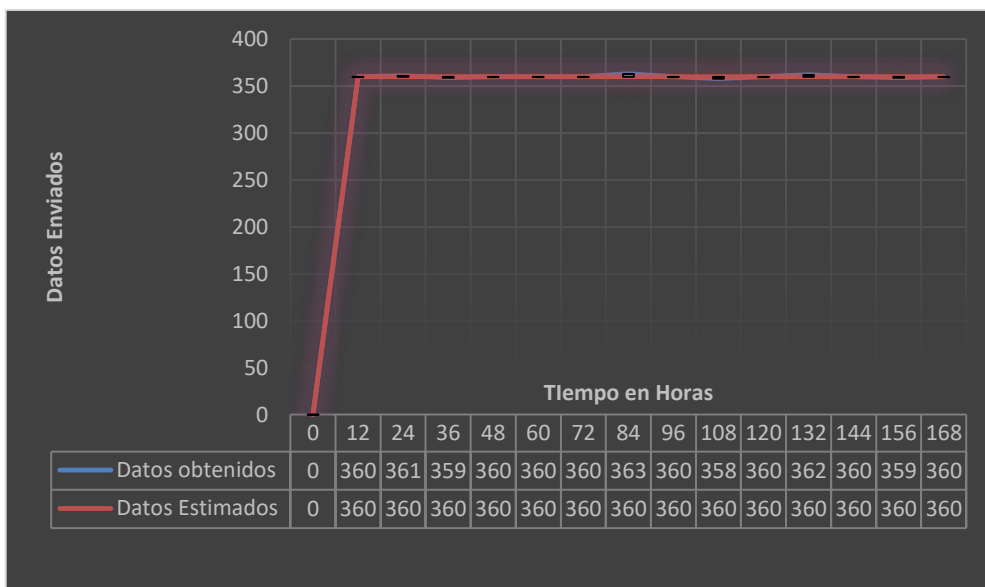
En esta prueba, se enviaron 5000 datos de manera simultánea al sistema para comprobar su capacidad de respuesta esto tomo un tiempo de 168 horas. Al observar los resultados obtenidos se muestran en la figura 26 y 27 en las cuales se puede constatar que el porcentaje de datos perdidos fue de 0.18%. Este resultado es de manera global, pero para tener un detalle más preciso se puede ver en la tabla, la respuesta del servidor 1 hacia el cliente 1 es la mayor en la cual se pierde un total de 12 datos lo cual supera al valor promedio. Por otra parte, el cliente 2 con el servidor 2 tiene una pérdida

mínima de datos la cual es de 7 datos perdidos. Con ello podemos ver tanto el valor máximo como el valor mínimo de los datos perdidos. Esto nos permite tener una visión más precisa del rendimiento del sistema y hacer las mejoras necesarias para optimizar el rendimiento del mismo. Además, al tener una visión detallada de los datos perdidos en cada transacción entre cliente y servidor, se pueden tomar medidas para minimizar esta pérdida en el futuro.

### **3.3.3 Prueba de estabilidad del servidor de Agregación**

La prueba del servidor de agregación OPC UA fue realizada con el objetivo de evaluar su estabilidad y fiabilidad a lo largo de un período prolongado de tiempo. La prueba se llevó a cabo durante una semana completa, con él envío de datos int16. Con la utilización de estos datos se comprobó la estabilidad del sistema dentro del tiempo especificado, al realizar el reporte de datos enviados y la obtención de las respuestas en un lapso de tiempo de 12 horas, al realizar un estimado de los datos que deben fluir por el sistema es igual a 360 datos por cada 12 horas con forme a ello los resultados que se obtuvieron en la prueba son similares a lo que se estimó para esta prueba, esta igualdad nos permite determinar que el sistema no se ha detenido ni ha sufrido algún contratiempo. Lo que permitió observar en la figura 28 es el comportamiento del sistema en condiciones normales de operación.

En los resultados de la prueba se evidenció que los clientes y los servidores agregados no se desconectaron y mantuvieron comunicación de manera fluida sin caídas del sistema ya que en el reporte de las solicitudes enviadas y respuestas recibidas se encuentran dentro de lo calculado.



**Figura 28** Datos enviado en un tiempo prolongado

**Elaborado por:** Investigador

Para la prueba del servidor de agregación OPC UA se evidenció que el sistema es estable, y se logró mantener una comunicación fluida entre clientes y servidores agregados durante una semana completa. La realización de la prueba fue en un ambiente controlado lo cual permitió monitorear el sistema de manera detallada y actuar de manera inmediata en caso de problemas.

### 3.3.4 Fiabilidad del sistema

Por medio de la prueba antes realizada se puede medir la fiabilidad del sistema mediante la fórmula de confiabilidad acumulada.

La fórmula de confiabilidad acumulada, también conocida como fórmula de confiabilidad para un tiempo dado, se utiliza para calcular la probabilidad de que un sistema o componente continúe funcionando correctamente después de un período de tiempo específico. La fórmula se expresa como:

$$R(t) = 1 - (1 - e^{-\lambda t})^n \quad (1)$$

Donde

$R(t)$  es la confiabilidad acumulada en el tiempo  $t$ ,  $\lambda$  es la tasa de fallos del sistema, y  $n$  es el número de componentes independientes en el sistema.

La tasa de fallo ( $\lambda$ ) se utiliza en las fórmulas de confiabilidad para medir la probabilidad de que un sistema o componente falle en un período de tiempo específico. Puede ser calculada utilizando la siguiente fórmula:

$$\lambda = \frac{\text{número de fallos}}{\text{número de intentos}} \quad (2)$$

La fórmula se basa en la idea de que la confiabilidad se acumula a medida que el tiempo transcurre. La fórmula también se puede utilizar para calcular la confiabilidad acumulada de un sistema que tiene varios componentes independientes. En este caso, se asume que cada componente tiene una tasa de fallos independiente y se multiplica la confiabilidad de cada componente para obtener la confiabilidad acumulada del sistema [24].

Para calcular la confiabilidad acumulada de un sistema que operó durante 168 horas, tuvo un promedio de 10 fallos de 5000 datos enviados y tiene 5 componentes independientes, los cuales se podría decir que los clientes y servidores son componentes independientes en un sistema de servidor de agregación OPC UA, se puede utilizar la fórmula de confiabilidad acumulada:

$$R(t) = 1 - (1 - e^{-\lambda t})^n \quad (3)$$

Para calcular  $\lambda$ , necesitamos el número de fallos y el número de intentos:

$$\lambda = \frac{10}{5000} = 0.002 \text{ fallos por intento}$$

Entonces, podemos calcular la confiabilidad acumulada:

$$R(t) = 1 - (1 - e^{-0.002 \cdot 168})^5$$

$$R(t) = 99.81\%$$

La confiabilidad acumulada sería del 99.81% lo cual significa que el sistema tiene un 99.81% de posibilidades de estar operando después de haber operado durante 168 horas y haber tenido un promedio de 10 fallos entre 5000 intentos y contar con 5 componentes independientes.

## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- La integración del servidor de agregación en el control de procesos industriales es crucial para garantizar la eficiencia y estabilidad del sistema. Para lograr esto, se utilizó Open62541, un Stack de comunicaciones OPC UA de código abierto, para garantizar la compatibilidad del servidor con el estándar OPC UA. Además, al modificar esta librería se pudo realizar la programación en 4Diac, ya que esta herramienta proporciona un entorno de desarrollo integrado (IDE) para la programación de aplicaciones de control de procesos basadas en la norma IEC-61499. Esta norma es esencial para la creación de proyectos de control de procesos industriales, ya que proporciona un marco estandarizado para la integración de los sistemas, lo que ayuda a garantizar la compatibilidad y la interoperabilidad entre diferentes sistemas y componentes.
- La integración del servidor de agregación para el control de procesos industriales dentro del entorno de ejecución (run-time) mediante la norma IEC-61499 es un paso importante, ya que en primer lugar ayuda a garantizar la eficiencia del servidor de agregación al permitir una comunicación fluida entre los diferentes clientes y servidores que se encuentren agregados al sistema. En segundo lugar, ayuda a garantizar la estabilidad del sistema al permitir una mejor supervisión y control de los procesos que se encuentren descentralizados. Además, la norma IEC-61499 proporciona un marco estandarizado para la integración de estos sistemas, lo que ayuda a garantizar la compatibilidad y la interoperabilidad entre diferentes sistemas y componentes.
- La validación del servidor de agregación para el control de procesos industriales, se llevaron a cabo varias pruebas para determinar su correcto funcionamiento. Una de las pruebas consistió en medir el tiempo que tardaba el servidor en recibir, procesar las solicitudes y enviar las respuestas correspondientes. El resultado de esta prueba dio un promedio de 210(ms) al envía 1000 datos int16, con ello los resultados de

esta prueba mostraron que el servidor era capaz de procesar y enviar respuestas hacia los clientes agregados.

Otra prueba consistió en poner en funcionamiento el servidor durante una semana sin interrupciones al enviar datos int16. Esto se hizo para evaluar la estabilidad del servidor y su capacidad para mantener su rendimiento en condiciones de funcionamiento continuo. El servidor se mantuvo estable durante todo el tiempo ya que ni los clientes ni los servidores se desconectaron del sistema y no presentaron problemas de rendimiento.

Por último, se llevó a cabo una prueba para evaluar la capacidad del servidor para manejar grandes cantidades de datos. Se trató de sobrecargar el sistema con él envió de 5000 datos int16 de cada cliente a cada servidor para ver cómo respondía. El servidor fue capaz de procesar y enviar respuestas a pesar de la carga adicional de datos, por medio de esta prueba dio una confiabilidad acumulada de 99.81%.

## **4.2 Recomendaciones**

- Considerar el uso de la librería Open62541 como una herramienta esencial en la creación de un servidor de agregación OPC UA. Al ser de código abierto es muy útil para la implementación de este sistema, esta librería puede ser utilizada para modificar y mejorar el stack de comunicaciones OPC UA, con lo cual nos permite cumplir las especificaciones necesarias para utilizarla en conjunto con la norma IEC 61499.

Es importante mencionar que antes de utilizar esta librería, es necesario un análisis y evaluación detallado del sistema y su cumplimiento con la norma IEC 61499, ya que Open62541 es solo una herramienta para mejorar el sistema. Una vez que se tiene un conocimiento profundo sobre cómo utilizarla, Open62541 ofrece una interfaz de programación de aplicaciones clara y documentada lo que facilita el desarrollo del software. Además, incluye características de seguridad como autenticación y encriptación lo que ayuda a proteger el sistema contra posibles ataques.



- Realizar un análisis detallado sobre el funcionamiento de los bloques funcionales de 4diac con el protocolo OPC UA antes de comenzar con la programación del sistema de agregación. Esto implica comprender cómo se relacionan los bloques funcionales con el protocolo y cómo afecta esta relación al funcionamiento del sistema.

Es importante tener en cuenta que 4Diac es un entorno de desarrollo para sistemas distribuidos basado en la norma IEC 61499, por lo que es esencial entender los aspectos técnicos de esta norma y cómo se integra con OPC UA para garantizar un funcionamiento adecuado. Al conocer esta relación se puede programar el sistema de manera eficiente y funcional.

- Tener en cuenta tanto el modelo de información como el espacio de dirección al momento de programar los bloques funcionales en 4Diac para una comunicación eficiente con los clientes y servidores que utilizan el protocolo OPC UA.

El modelo de información es esencial para comprender como los datos son representados y como se relacionan entre sí, mientras que el espacio de dirección es necesario para identificar de manera precisa dónde se encuentran los datos dentro de los clientes y servidores. Conociendo estos aspectos se podrá programar los bloques funcionales de manera efectiva y garantizar una comunicación adecuada entre los dispositivos que utilizan el protocolo OPC UA.

## BIBLIOGRAFÍA

- [1] F. J. Jácome Sagñay, «“Estudio comparativo mediante simulación de las variantes de la tecnología OPC-UA y su utilización en controladores industriales”», 2016, Accedido: 20 de septiembre de 2022. [En línea]. Disponible en: <http://repositorio.puce.edu.ec:80/handle/22000/13154>
- [2] J. C. Mercé Godoy, «Desarrollo de un sistema de control en red mediante el estándar IEC-61499», 2015, Accedido: 20 de septiembre de 2022. [En línea]. Disponible en: <http://repositori.uji.es/xmlui/handle/10234/148085>
- [3] E. A. Chacón-Ramírez, J. J. Cardillo-Albarrán, y J. Uribe-Hernández, «Industria 4.0 en América Latina: Una ruta para su implantación», *Rev. Ingenio*, vol. 17, n.º 1, Art. n.º 1, ene. 2020, doi: 10.22463/2011642X.2386.
- [4] S. K. Panda, M. Majumder, L. Wisniewski, y J. Jasperneite, «Real-time Industrial Communication by using OPC UA Field Level Communication», en *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, sep. 2020, vol. 1, pp. 1143-1146. doi: 10.1109/ETFA46521.2020.9211998.
- [5] S. G. Mathias, S. Schmied, y D. Grossmann, «An Investigation on Database Connections in OPC UA Applications», *Procedia Comput. Sci.*, vol. 170, pp. 602-609, ene. 2020, doi: 10.1016/j.procs.2020.03.132.
- [6] Universidade da Coruña, Escuela Politécnica Superior,15403, Ferrol, España, G. González-Filgueira, F. J. R. Permy, y Universidade da Coruña, Escuela Politécnica Superior,15403, Ferrol, España, «Automatización de una planta industrial de alimentación mediante control distribuido», *RISTI - Rev. Ibérica Sist. E Tecnol. Informação*, vol. 27, pp. 1-17, jun. 2018, doi: 10.17013/risti.27.1-17.
- [7] M. V. García, E. Irisarri, y F. Pérez, «Integración Vertical en plantas industriales utilizando OPC UA e IEC-61499», *Enfoque UTE*, vol. 8, pp. 287-299, feb. 2017, doi: 10.29019/enfoqueute.v8n1.132.
- [8] S. P. Bustos Pulluquitin., «Desarrollo de un sistema de control industrial basado en el estándar IEC-61499», 2017, Accedido: 20 de septiembre de 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/26656>
- [9] D. Großmann, M. Bregulla, S. Banerjee, D. Schulz, y R. Braun, «OPC UA server aggregation — The foundation for an internet of portals», en *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, sep. 2014, pp.

- 1-6. doi: 10.1109/ETFA.2014.7005354.
- [10] V. C. Almache Barahona, «Diseño de un HMI en Web Servers del PLC S7-1200 / 1500 para el control de un proceso multivariable de un módulo didáctico para el Laboratorio de Hidrónica y Neutrónica de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga.», bachelorThesis, Universidad de las Fuerzas Armadas ESPE Extensión Latacunga. Carrera de Ingeniería Electromecánica., 2017. Accedido: 8 de enero de 2023. [En línea]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/13718>
- [11] «CPU 1507S». <https://mall.industry.siemens.com/mall/es/es/Catalog/Products/10268313?activeTab=ProductInformation> (accedido 2 de enero de 2023).
- [12] C. P. Calán Juna, «Diseño e implementación de un sistema de control supervisorio en plataformas web de controladores S7-1500 Simatic de Siemens, para la empresa InGel-Pro», bachelorThesis, Quito, 2019., 2019. Accedido: 8 de enero de 2023. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/20049>
- [13] S. P. Simlinger Benedict, «From modelling to execution - OPC UA Information Model Tutorial • OPC UA rocks», *OPC UA rocks*, 29 de agosto de 2020. <https://opcua.rocks/from-modelling-to-execution-opc-ua-information-model-tutorial/> (accedido 7 de enero de 2023).
- [14] M. Postol, *OPC UA Address Space Interchange XML*. 2016. doi: 10.13140/RG.2.2.12228.37768.
- [15] «Introducción — documentación de open62541 1.2.0-5-gcc3a3d396-dirty». <http://www.open62541.org/doc/1.2/index.html> (accedido 15 de octubre de 2022).
- [16] C. A. García, E. X. Castellanos, y M. V. García, «Desarrollo de sistemas ciberfísicos 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.
- [17] «Introducción — documentación de open62541 1.3.3-undefined». <https://www.open62541.org/doc/master/> (accedido 13 de enero de 2023).
- [18] X. A. Cárdenas Medina, «Integración de la capa de comunicación de la norma IEC-61499 basada en los protocolos de la Industria 4.0», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica

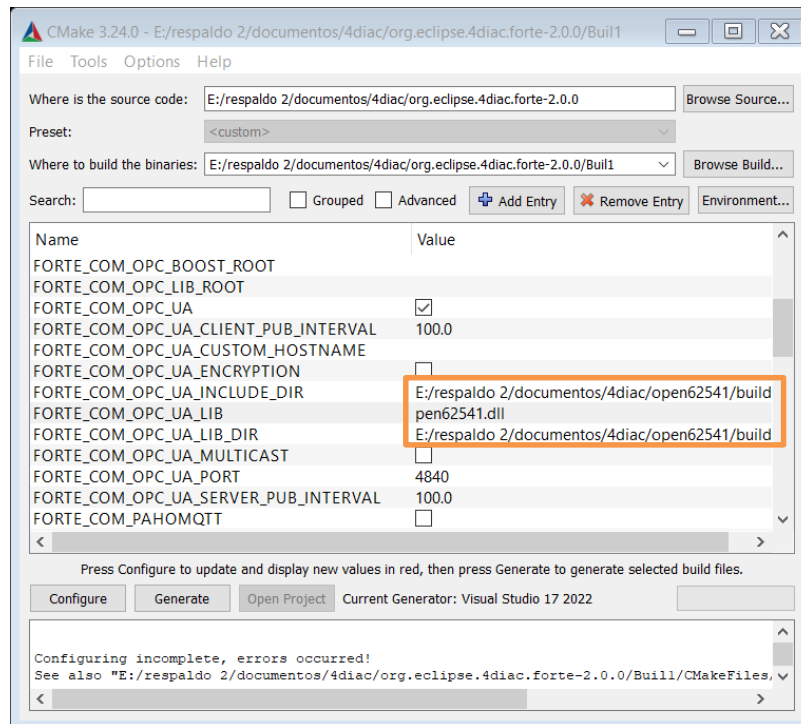
- e Industrial. Carrera de Ingeniería en Electrónica y Comunicaciones, 2020. Accedido: 8 de enero de 2023. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/31640>
- [19] E. X. Castellanos Narvárez y C. A. García Sánchez, «Control distribuido en procesos industriales utilizando sistemas empotrados de bajo costo, para verificar la aplicabilidad de la norma IEC-61499.», bachelorThesis, Universidad de las Fuerzas Armadas ESPE. Extensión Latacunga. Carrera de Ingeniería en Electrónica e Instrumentación., 2017. Accedido: 8 de enero de 2023. [En línea]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/13944>
- [20] «Eclipse 4diac: el entorno de código abierto para sistemas de control y automatización industrial distribuidos». <https://www.eclipse.org/4diac/index.php> (accedido 13 de enero de 2023).
- [21] W. Mahnke y S.-H. Leitner, «System Architecture», en *OPC Unified Architecture*, W. Mahnke, S.-H. Leitner, y M. Damm, Eds. Berlin, Heidelberg: Springer, 2009, pp. 265-282. doi: 10.1007/978-3-540-68899-0\_9.
- [22] S. Kozar y P. Kadera, «Integration of IEC 61499 with OPC UA», en *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, sep. 2016, pp. 1-7. doi: 10.1109/ETFA.2016.7733538.
- [23] A. Tendero-Vegas, O. Miguel-Escrig, J.-A. Romero-Pérez, B. Wiesmayr, V. Ashiwal, y A. Zoitl, «Integración de FACTORY IO y 4DIAC-FORTE para la validación de software de control en la norma IEC 61499», en *XLIII Jornadas de Automática: libro de actas: 7, 8 y 9 de septiembre de 2022, Logroño (La Rioja), 2022.*<sup>a</sup> ed., Servizo de Publicacións da UDC, 2022, pp. 949-955. doi: 10.17979/spudc.9788497498418.0949.
- [24] A. Birolini, «Reliability Analysis During the Design Phase», en *Reliability Engineering: Theory and Practice*, A. Birolini, Ed. Berlin, Heidelberg: Springer, 2017, pp. 43-44. doi: 10.1007/978-3-662-54209-5\_2.

## Anexo 1

### Compilación de 4Diac FORTE con la Librería Open62541

La compilación de FORTE con la librería Open62541 es un proceso que involucra la integración de la librería Open62541 en el entorno de desarrollo de FORTE y la compilación del código fuente de FORTE utilizando esta librería.

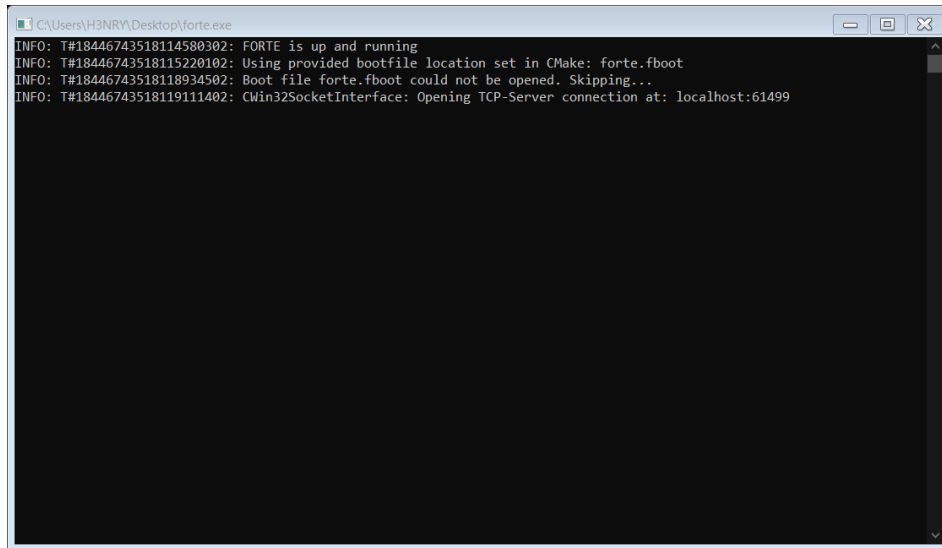
Para llevar a cabo esta tarea, primero es necesario descargar e instalar la librería Open62541 en tu sistema. A continuación, debes añadir la librería Open62541 al entorno de desarrollo de FORTE como se muestra en la figura 29. Esto se puede hacer mediante la configuración de las opciones de enlace y la configuración de los directorios de incluir en FORTE.



**Figura 29** Integración de la librería Open62541 en Forte

**Elaborado por:** Investigador

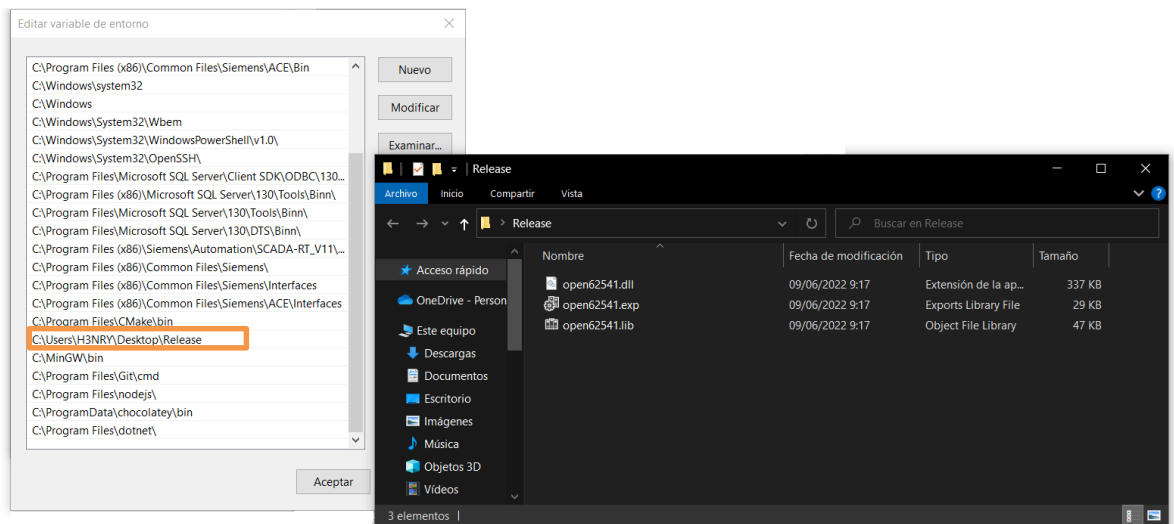
Una vez que la librería Open62541 está integrada en el entorno de desarrollo de FORTE, puedes utilizarla para compilar el código fuente de FORTE. Para ello, debes abrir el proyecto de FORTE en tu entorno de desarrollo y seleccionar la opción de compilación. Esto generará un archivo ejecutable que puedes utilizar para ejecutar tu aplicación FORTE utilizando la librería Open62541 como se muestra en la figura 30.



**Figura 30** Ejecución de Forte con la librería Open62541

**Elaborado por:** Investigador

Es importante tener en cuenta que pueden ser necesarias algunas configuraciones adicionales para que la compilación de FORTE con la librería Open62541 sea exitosa como se indica en la figura 31, como la configuración de las opciones de optimización y la configuración de los directorios de librerías.



**Figura 31** Configuración del equipo con las librerías de Open62541

**Elaborado por:** Investigador

La compilación de FORTE con la librería Open62541 implica la integración de la librería Open62541 en el entorno de desarrollo de FORTE, la configuración de las opciones de enlace y los directorios de incluir, y la compilación del código fuente de FORTE utilizando esta librería.