



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica-Eléctrica

**ANÁLISIS DEL MÉTODO SOFTWARE IN THE LOOP, COMO ALTERNATIVA PARA
SIMULACIÓN EN EL LABORATORIO DE AUTOMATIZACIÓN INDUSTRIAL DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Pedro Josué Rubén Fuentes Alonzo

Asesorado por el Ing. Gustavo Benigno Orozco Godínez

Guatemala, septiembre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Llorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. José Guillermo Bedoya Barrios
EXAMINADOR	Ing. Julio Rolando Barrios Archila
EXAMINADOR	Ing. Gustavo Benigno Orozco Godínez
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**ANÁLISIS DEL MÉTODO SOFTWARE IN THE LOOP, COMO ALTERNATIVA PARA
SIMULACIÓN EN EL LABORATORIO DE AUTOMATIZACIÓN INDUSTRIAL DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 30 de octubre de 2019.

Pedro Josué Rubén Fuentes Alonzo

Guatemala, 12 de junio de 2020.

Ingeniero Fernando Moscoso
Coordinador Área de Potencia
Escuela Mecánica Eléctrica
Facultad de Ingeniería
USAC.

Estimado Ingeniero:

De acuerdo con la designación efectuada por la Dirección de Escuela, me permito informarle que he procedido a asesorar el Trabajo de Graduación titulado: **ANALISIS DEL METODO SOFTWARE-IN-THE-LOOP, COMO ALTERNATIVA PARA SIMULACION EN EL LABORATORIO DE AUTOMATIZACION INDUSTRIAL, DE LA UNIVESIDAD DE SAN CARLOS DE GUATEMALA**, desarrollado por el estudiante PEDRO JOSUE RUBEN FUENTES ALONZO, carne 2015-03781 y, encontrándolo satisfactorio en su contenido y resultados, me permito dar aprobación al mismo, remitiéndolo a esa Coordinación para el tramite pertinente, en el entendido que el Autor y este Asesor somos responsables del contenido y conclusiones del Trabajo.

Agradeciendo la atención a la presente, me es grato suscribirme, deseándole éxitos en sus labores cotidianas.

Atentamente,



Ing. Gustavo Benigno Orozco Godínez.
Colegiado 1,879
ASESOR

ING. GUSTAVO B. OROZCO
COLEGIADO 1879



Guatemala, 25 de marzo de 2021

Ingeniero
Armando Alonso Rivera Carrillo
Director
Escuela de Ingeniero Mecánica Eléctrica
Facultad de Ingeniería USAC

Ingeniero Rivera:

Por este medio, con base a lo indicado en el REGLAMENTO DE TRABAJOS DE GRADUACION vigente, tengo a bien proponer la aprobación del trabajo de graduación titulado:

**ANÁLISIS DEL MÉTODO SOFTWARE IN THE LOOP, COMO ALTERNATIVA PARA
SIMULACIÓN EN EL LABORATORIO DE AUTOMATIZACION INDUSTRIAL DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

del estudiante Pedro Josué Rubén Fuentes Alonzo, habiendo cumplido con los requisitos establecido en el referido reglamento.

Sin otro particular

Atentamente,
ID Y ENSEÑAD A TODOS

A handwritten signature in black ink, appearing to read 'Fernando Moscoso Lira', written over a diagonal line.

Ingeniero Fernando Alfredo Moscoso Lira
Coordinador Área de Potencia
Escuela de Ingeniería Mecánica Eléctrica.

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA



REF. EIME 81. 2021.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; PEDRO JOSUÉ RUBÉN FUENTES ALONZO titulado; ANÁLISIS DEL MÉTODO SOFTWARE IN THE LOOP, COMO ALTERNATIVA PARA SIMULACIÓN EN EL LABORATORIO DE AUTOMATIZACIÓN INDUSTRIAL DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, procede a la autorización del mismo.


Ing. Armando Alonso Rivera Carrillo


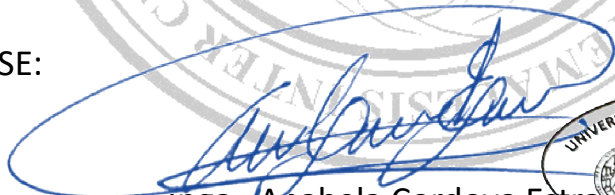


GUATEMALA, 27 DE ABRIL 2021.

DTG. 405-2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **ANÁLISIS DEL MÉTODO SOFTWARE IN THE LOOP, COMO ALTERNATIVA PARA SIMULACIÓN EN EL LABORATORIO DE AUTOMATIZACIÓN INDUSTRIAL DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario: **Pedro Josué Rubén Fuentes Alonzo**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana

Guatemala, septiembre de 2021

AACE/asga

ACTO QUE DEDICO A:

- Dios** Por haberme brindado sabiduría, salud y vida durante todos estos años. Por brindarme bendiciones y compañía.
- Mis padres** Mariela Alonzo y Fredy Fuentes, por apoyarme y brindarme su amor incondicional, por ser un claro ejemplo de valores y superación.
- Mi abuelo** Rubén Fuentes Julián, por ser una parte importante de mi vida y brindarme su apoyo y amor desde el cielo.
- Mi abuela** Carlota Gómez Alonzo, por su apoyo y ejemplo de superación y trabajo.
- Mis hermanos** José, Josué Alonzo y Diego Fuentes por su apoyo y motivación a seguir adelante.
- Mis amigos** Brandon Atz, Pacífico Tumax, Dennis López, Luis Monzón, José Gálvez, Brandon Mayorga, Diego Coronado, Alejandro Lopez, Roberto Marticorena, Marcos Guoz, Alex Dávila, Kevin Duarte, Diego Serech, Renato Cifuentes, Ángel Alvizures, Erwin Lemus.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala Por brindarme la oportunidad de ser un profesional.

Gustavo Orozco Por su apoyo y por haber aceptado este reto conmigo.

Simumatik Por haberme ayudado ante cualquier duda que tenía sobre su software.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. CONCEPTOS GENERALES SOBRE AUTOMATIZACIÓN INDUSTRIAL	1
1.1. Definición.....	2
1.2. Sensores	3
1.2.1. Definición	3
1.3. Actuadores	3
1.3.1. Definición	4
1.4. Controlador Lógico Programable (PLC).....	4
1.4.1. Historia	5
1.4.2. Funciones básicas de un PLC	6
1.4.3. Campos de aplicación	8
1.4.4. Programación básica	10
1.4.4.1. LD (Load) y LD NOT (Load Not)	10
1.4.4.2. OUT Y OUT NOT.....	11
1.4.4.3. AND	12
1.4.4.4. OR	12
1.4.5. Temporizadores y contadores	13
1.4.5.1. Temporizadores TIM.....	14

1.4.5.2.	Contadores CNT	14
2.	CONCEPTOS DE COMUNICACIÓN INDUSTRIAL PLC	15
2.1.	Redes de comunicación industrial.....	16
2.1.1.	Comunicaciones industriales.....	16
2.1.2.	Topologías de redes.....	19
2.1.2.1.	Topología punto a punto.....	19
2.1.2.2.	Topología de bus.....	20
2.1.2.3.	Topología anillo	21
2.1.3.	Técnicas de control de errores	22
2.2.	Redes de comunicación industrial Ethernet	23
2.2.1.	Características de Ethernet.....	24
2.2.2.	Comunicación entre varios PLC mediante protocolo TCP/IP	26
2.3.	Servidores OLE for Process Control (OPC)	28
2.3.1.	Conceptos generales OPC.....	29
2.3.2.	Comunicación OPC-SOFTWARE	32
2.3.3.	Comunicación OPC-PLC.....	34
3.	ALTERNATIVAS DE MÉTODOS DE SIMULACIÓN PARA AUTOMATIZACION INDUSTRIAL.....	37
3.1.	Model-in-the-loop	38
3.1.1.	Definición.....	38
3.1.2.	Funciones de la metodología MIL	40
3.1.3.	Ventajas y desventajas del método de simulación MIL	43
3.2.	Software-In-The-Loop	45
3.2.1.	Definición.....	46
3.2.2.	Funciones de la metodología SIL.....	48

3.2.3.	Ventajas y desventajas del método de simulación SIL	51
3.3.	Hardware-In-The-Loop	53
3.3.1.	Definición	54
3.3.2.	Funciones de la metodología HIL	57
3.3.3.	Ventajas y desventajas del método de simulación HIL	59
4.	ALTERNATIVAS DE SOFTWARE Y MODELAMIENTO PARA EL MÉTODO SIL	61
4.1.	Factory I/O.....	63
4.1.1.	Conceptos básicos	64
4.1.1.1.	Navegación.....	64
4.1.1.1.1.	<i>Orbit camera</i>	66
4.1.1.1.2.	<i>Fly camera</i>	67
4.1.1.1.3.	<i>First person camera</i>	68
4.1.1.2.	Escenas.....	69
4.1.2.	Análisis de plataforma Factory I/O.....	71
4.1.2.1.	Elementos.....	71
4.2.	Codesys.....	72
4.2.1.	Conceptos básicos	73
4.2.2.	Análisis de plataforma Codesys.....	75
4.2.2.1.	Programación	75
4.2.2.2.	Visualización HMI	77
4.3.	Simumatik3D	78
4.3.1.	Conceptos básicos	78
4.3.1.1.	Navegación 2D	79
4.3.1.2.	Navegación 3D	80
4.3.1.3.	Funcionabilidad.....	82

4.3.2.	Análisis de plataforma	85
4.3.2.1.	Escenarios.....	86
4.3.2.2.	Componentes	88
4.3.2.3.	Descripción de componentes	90
4.3.3.	Vinculación Codesys-Simumatik	92
5.	APLICACIÓN DEL MÉTODO DE SIMULACIÓN SIL	93
5.1.	Simumatik APP	94
5.2.	Visualización del método SIL en Simumatik APP	95
5.3.	Conexión OPC Codesys-Simumatik APP	99
5.3.1.	Proceso de conexión OPC	99
5.4.	Descripción de procesos	103
5.4.1.	Estructuración de componentes	104
5.4.2.	Conexión y cableado	106
5.5.	Ejemplificación	107
5.5.1.	Ejemplo básico de conexión y cableado	107
5.5.2.	Ejemplo de modelamiento mediante PLC	112
5.5.3.	Visualización del método de simulación SIL en Factory I/O.....	115
	CONCLUSIONES.....	119
	RECOMENDACIONES	121
	BIBLIOGRAFÍA.....	123
	APÉNDICE	125

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Modelo estructural de un sistema automatizado	2
2.	Controlador lógico programable	6
3.	Contacto normalmente abierto	11
4.	Contacto normalmente cerrado	11
5.	Salida normalmente abierta	11
6.	Salida normalmente cerrada	12
7.	Contactos en serie	12
8.	Contactos en paralelo	13
9.	Temporizador TON.....	13
10.	Temporizador TOF.....	14
11.	Pirámide de comunicaciones	18
12.	Topología punto a punto	20
13.	Topología de bus	21
14.	Topología anillo.....	21
15.	Relación de comunicación entre 3 PLC	26
16.	Ejemplo de Diagrama de Bloques HIL utilizando Labview	29
17.	Arquitectura típica OPC.....	33
18.	OPC en varios servidores	34
19.	Ejemplo de sincronización de motor por órdenes OPC.....	35
20.	Proceso PLC-OPC	36
21.	Estructura MIL.....	39
22.	Torque con carga aplicado al motor de inducción simulado.....	40
23.	Ejemplo de diseño de célula de producción PSL en SketchUp.....	42

24.	Estructura SIL	47
25.	Ejemplo de utilización del método SIL en Flexsim	51
26.	Estructura HIL	55
27.	Comparación de pruebas de simulación efectivas	56
28.	Ejemplo de simulación HIL en aeronáutica	58
29.	Enlace Factory I/O y Codesys	62
30.	Logo del software Factory I/O	64
31.	Ambiente estándar de navegación.....	65
32.	Íconos de cámara en software	65
33.	Ejemplos de escenas en el software Factory I/O	70
34.	Ejemplos de elementos en Factory I/O	72
35.	Logo del software Codesys.....	75
36.	Ejemplo de definición de variables en Codesys.....	76
37.	Ejemplo de visualización HMI en Codesys	78
38.	Interfaz 2D en Simumatik3D	79
39.	Visualización de componente en 2D.....	80
40.	Visualización 3D en Simumatik3D	81
41.	Ejemplo de proceso 3D.....	82
42.	Barra de herramientas Simumatik3D	84
43.	Menú de objetos utilizados y su conexión.....	85
44.	Ejemplo de escenario	87
45.	Ejemplo de descripción de la composición del proceso	88
46.	Ejemplo de objetos disponibles Simumatik3D	89
47.	Parámetros de un motor AC en Simumatik3D	90
48.	Parámetros de un sensor de proximidad en Simumatik3D	91
49.	Visualización de servidor en la nube.....	96
50.	Escenario general en Simumatik APP	97
51.	Ejemplo de descripción en servidor	98
52.	Creación de variables globales.....	100

53.	Ejemplo declaración de variables globales	101
54.	Opción de configuración de símbolos	101
55.	Selección de variables a comunicar	102
56.	Activación del Gateway	102
57.	PLC virtual de Codesys.....	103
58.	Estructuración de un motor	105
59.	Estructuración de un botón	106
60.	Modelo de pulsadores y luces.....	108
61.	Estructuración	109
62.	Conexión H01	110
63.	Conexión H02	110
64.	Conexión H03	110
65.	Conexión S01.....	111
66.	Conexión S02.....	111
67.	Conexión S03.....	111
68.	Estructura de modelo y PLC	113
69.	Conexiones establecidas al PLC como entradas	114
70.	Conexiones establecidas al PLC como salidas.....	115
71.	Ejemplo de modelo SIL en Factory I/O.....	116
72.	Ejemplo de establecimiento de variables en un PLC (HIL O SIL)	117

TABLAS

I.	Componentes activos y pasivos en red de comunicación.....	23
II.	Ejemplo de enlace de protocolo.....	27
III.	Cuadro comparativo de las funciones MIL.....	43
IV.	Cuadro comparativo de las funciones del método SIL.....	52
V.	Cuadro comparativo de las funciones del método HIL.....	60
VI.	Control de <i>Orbit camera</i>	67
VII.	Control de <i>Fly camera</i>	68
VIII.	Control de <i>First person camera</i>	69

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperios
I	Corriente
I/O	Entrada y salida
L1	Línea 1
L2	Línea 2
L3	Línea 3
Nm	Newton - metro
P	Potencia
X1	Terminal 1
X2	Terminal 2
Ts	Tiempo de paso
Tv	Tiempo de voltaje
V	Voltaje

GLOSARIO

API	Autómata programable industrial, nombre anterior al PLC.
DDE	Dynamic Data Exchange, se basa en referencias de datos y sus importaciones entre softwares y programas.
ETHERNET	Es una tecnología que conecta redes de área local cableadas y permite que el dispositivo se comuniquen entre sí a través de un protocolo que es el lenguaje de red común.
GRAFSET	Diagrama funcional que describe los procesos a automatizar, teniendo en cuenta las acciones a realizar y los procesos intermedios que provocan estas acciones.
HIL	Hardware-in-the-loop, es un tipo de simulación industrial o didáctica que hace uso de un controlador físico y el proceso virtual.
HMI	Interfaz humano-máquina, utilizada por operarios y supervisores de línea para coordinar y controlar procesos industriales y de fabricación.

LADDER	Lenguaje escalera, tipo de programación para PLC que se basa en una forma escalonada.
LAN	Red de área local, es una red de computadoras que abarca un área reducida a una casa, un departamento o un edificio.
MIL	Model-In-the-loop, consiste en simulación exclusivamente de la planta o proceso.
OPC	Es un estándar de comunicación en el campo del control y supervisión de procesos industriales, permite que componentes de software individuales interactúen y compartan datos.
OPCUA	Es un protocolo de comunicación máquina a máquina para automatización industrial.
PID	Controlador proporcional, integral y derivativo. Es un mecanismo de control simultáneo por realimentación ampliamente usado en sistemas de control industrial, este calcula la desviación o error entre un valor medido y un valor deseado.
PLC	Controlador lógico programable, se refiere básicamente a una computadora industrial que procesa todos los datos entrantes controla los datos salientes.

RIL	Real-In-the-Loop, simulación que conlleva controlador y proceso real.
SCADA	Control supervisado y adquisición de datos, que se refiere a la capacidad de supervisar datos en tiempo real e historizarse.
SIL	Software-in-the-loop, tipo de simulación que conlleva el controlador y proceso simulado virtualmente.

RESUMEN

En el presente trabajo de graduación se presenta una alternativa de simulación aplicado al curso de Automatización Industrial utilizando el software Simumatik para poder implementarse, pudiéndose aplicar al laboratorio de dicho curso en el ámbito de diseño y simulación de ambientes automatizados, así también como la modelación de un sistema mediante alternativas de software.

En el primer capítulo se describen los conceptos básicos de PLC, así como su historia, influencia y formas de uso.

En el capítulo 2, se detallan conceptos de comunicación en el ámbito de la automatización industrial, referido a varias alternativas y técnicas de control, introduciendo así el concepto de servidor OPC.

En el capítulo tercero se describen las alternativas de métodos de simulación más actualizadas, así como su conceptualización y aplicación.

En el capítulo cuarto se detallan alternativas de software para el diseño y modelamiento lógico-virtual en el enfoque de simulación en sistemas automatizados.

En el capítulo quinto se ejemplifican diseños básicos para la estructuración y funcionamiento general del software Simumatik.

OBJETIVOS

General

Analizar o realizar una alternativa correspondiente a la simulación de procesos en automatización flexible para el laboratorio de Automatización Industrial de la Universidad de San Carlos de Guatemala, con la ayuda del software Simumatik y Codesys.

Específicos

1. Evaluar la importancia de mejorar la eficiencia de diseño y programación aplicando una simulación para procesos de automatización flexible.
2. Evaluar los procedimientos de la utilización del método software-in-the-loop.
3. Evaluar las ventajas y diseños eficientes para el uso del método software-in-the-loop.
4. Identificar alternativas de uso del método software-in-the-loop.
5. Determinar los parámetros adecuados, para un mejor rendimiento en la simulación de procesos automatizados y su posible implementación en el laboratorio de Automatización Industrial de la Universidad de San Carlos de Guatemala.

INTRODUCCIÓN

En la elaboración de un proceso automatizado, en base a las necesidades o mejora continua en relación a algún proceso, tanto secuencial como continuo, se debe realizar una planeación efectiva conllevando así un diseño, modelado y simulación antes de realizar una implementación. Esto se refiere a mejorar la eficiencia del proceso a automatizar, obteniendo alternativas y analizando el comportamiento de la lógica establecida.

Una de las partes importantes antes de implementar un proceso automatizado, es la simulación de dicho proceso, tanto el modelo como la lógica que se utilizará para controlarlo, de forma que se establezca un error mínimo y que la eficiencia del diseño sea máxima, al reducir la relación costo - beneficio.

Tomando en cuenta las condiciones establecidas, es importante recalcar este estilo de simulación de forma didáctica, cuyo caso será importante en un futuro dado a la novedad del método de simulación, donde reducir la complejidad del análisis de sistemas debe ser prevista, y es necesario en la industria porque con la utilización de una simulación previa a la implementación, disminuye la cantidad de errores y verifica el diseño previsto.

La utilización del software y la virtualización de los componentes, se ha estado utilizando de forma global. El análisis de datos, diseño de sistemas, transferencia de información y comunicación se ha ido estableciendo de forma que existan menos componentes físicos posibles, y en el caso de la comunicación entre softwares enfocados a la automatización industrial, se debe establecer

mediante servidores y modelamientos virtuales de manera que se reduzca el uso de memoria y componentes físicos.

El establecimiento de la lógica de programación de cualquier controlador, concatenado a un software de diseño, es una forma reducida de transferencia de datos, realizándolo mediante servidores, en este caso un protocolo OPC, que habilita la etiqueta y compartición de datos establecidos en cualquier software de diseño, por tanto es importante obtener alternativas de emulación y simulación SIL para reducir una simulación industrial o didáctica.

1. CONCEPTOS GENERALES SOBRE AUTOMATIZACIÓN INDUSTRIAL

La automatización de procesos ha sido muy importante para la industria a lo largo de los años, donde tiene como objetivo aumentar la eficiencia de procesos industriales, siendo más efectivos y rápidos. De esta manera, las industrias pueden proveer mayores masas o aumento de velocidad en el proceso.

La automatización industrial ha ido evolucionando, donde primeramente se tenían solamente contactores, hoy en día hay microcontroladores y simuladores para evaluar diferentes tareas y aumentar la productividad.

Actualmente, hay una variedad de métodos de diseño y simulación enfocado a la automatización industrial, dependiendo del objetivo del proceso, por lo cual, es importante recalcar que cualquier método de diseño debe ser comprobado mediante pruebas previas, para determinar su funcionamiento y modelación del proceso diseñado previo a su implementación.

Para estos procesos hay una variedad de softwares tanto simples como complejos operando una tarea específica, siendo como programación, diseño o simulación. Métodos actuales incluyen ciertas etapas o completamente todas, así que es destacado el uso de estos softwares en industrias avanzadas donde se toman en cuenta el escenario, máquinas de uso, diseño y funcionamiento.

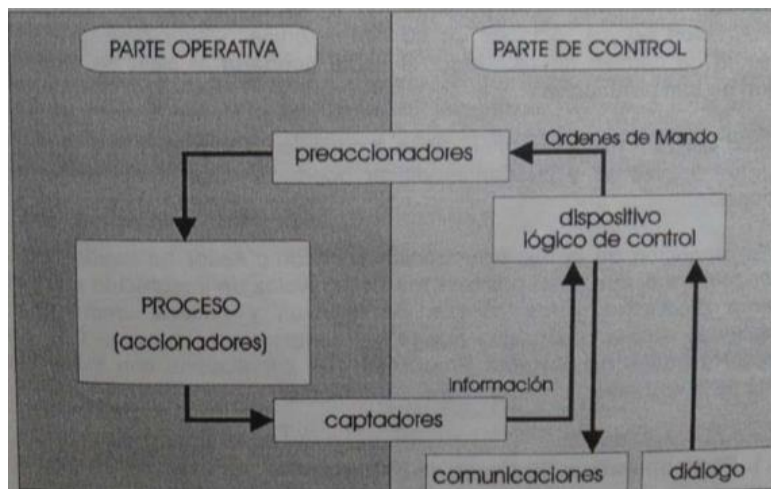
1.1. Definición

El concepto de automatización se refiere a realizar un proceso de manera automática o la intervención de factores que hagan un proceso automático, es decir, convertir ciertos movimientos en movimientos automáticos o indeliberados.

En todo proceso industrial automatizado, debe haber contactores, sensores, actuadores o controladores que evalúen el proceso y determinen la tarea de manera automática, en donde presenta estas características:

- Optimiza recursos
- Reduce tiempos de producción y esfuerzo
- Mayor eficiencia en los procesos

Figura 1. **Modelo estructural de un sistema automatizado**



Fuente: MORENO, Emilio. *Automatización de procesos industriales*. p. 11.

1.2. Sensores

Los sensores tienen como principal tarea medir parámetros predeterminados por el proceso. Mediante estas mediciones, el proceso automatizado actúa y realiza las distintas tareas según sea el ambiente y parámetros.

1.2.1. Definición

De acuerdo a la normativa IEEE 1451.0, un sensor mide parámetros físicos y devuelve una representación digital que representa dicho parámetro. Al recibir un *trigger* (disparo), si el *triggering* está activado, el sensor debe empezar a recibir y almacenar un conjunto de datos en el TIM. El tiempo de cada muestra en el conjunto de datos será controlado por el TIM y es función de los modos de operación del sensor. Un sensor, cuando el transductor está en estado operativo, responderá al comando Read Transducer Channel devolviendo el conjunto de datos apropiado. Entre los parámetros que puede medir un sensor están: Temperatura, Desplazamiento, Aceleración, Presión, Caudal, Velocidad lineal, Velocidad angular, Color, Densidad, Fuerza, entre otros

1.3. Actuadores

De acuerdo a la normativa IEEE 1451.0, un actuador realiza un determinado cambio físico o provoca una cierta acción. La salida de un actuador cambia a fin de ajustarse al conjunto de datos cuando ocurre un evento *triggering*. Si hay más de un punto de información en un conjunto de datos, el intervalo entre la emisión del conjunto de puntos de información debe estar bajo el control del TIM.

1.3.1. Definición

Los actuadores se definen como los elementos finales que permiten modificar las variables a controlar en un proceso determinado.

En el caso de la automatización, es capaz de transformar la energía eléctrica entregada del PLC hacia un contacto. Este pulso de electricidad permite el movimiento de las distintas variables a controlar.

1.4. Controlador Lógico Programable (PLC)

El controlador lógico programable (PLC) es una parte fundamental para poder automatizar un proceso tanto secuencial como continuo. A lo largo de la historia, los avances respecto a los primeros controladores, donde se necesita menor cableado y las funciones actualmente permiten más tareas de trabajo, así como también la capacidad de simularlos virtualmente.

Estos controladores son capaces de enviar pulsos eléctricos a sus contactos, de tal manera que los procesos se lleven a cabo mediante la división de tareas por contacto. Estos son capaces de trabajar con sensores, actuadores, transductores, motores y demás elementos que forman parte de un proceso de automatización.

Estos han sido de gran ayuda debido a que los procesos actualmente sustituyen fácilmente el trabajo humano. Como referencia, se tienen las industrias embotelladoras que, mediante la automatización y el uso de controladores lógicos programables, logran mayores masas de productos en los que superan considerablemente a las masas de productos trabajados de manera manual, y se toma en cuenta el ahorro de recursos y espacio de trabajo.

Actualmente es muy común el uso de controladores lógicos programables en la industria, cuya tecnología no solo se aplica en automatizar procesos, sino que también en la simple metodología de contacto a rangos más pequeños, como objetos de luminosidad que funcionan mediante esta tecnología tanto en domótica como a nivel singular.

1.4.1. Historia

El concepto de autómeta programable industrial (API) surge por los años 60, por parte de la empresa General Motors a fin de optimizar sus automóviles, y además fue necesario debido a los altos costos que tenían la reparación de los antiguos armarios de relés. De igual manera, la empresa R.E. Moreley desarrollaba prototipos similares a los de General Motors, cuyas especificaciones de diseño eran:

- Reutilizables
- Adaptación a los distintos ambientes
- Lenguaje de programación comprensible
- Fácil mantenimiento
- Implementación mediante electrónica estática

Los primeros autómetas programables disponían de memoria, basada en núcleos de ferrita de tipo cableado, y su diseño se establecía mediante un computador.

En 1968, los autómetas de equipamiento poseían memoria cableada y una unidad de central constituida por circuitos integrados, enfocados ya de manera secuencial, eran mucho más simples de espacio que los antiguos armarios de

relés, su principal competencia, pero estos últimos carecían de optimización de espacio y de optimización de mantenimiento.

Con los años, fue avanzando la utilización del PLC de tal forma que en la década de los 80, hubo nueva velocidad de respuesta, reducción de las dimensiones, mayor concentración de entradas y salidas I/O, desarrollo de módulos de control continuo, PID, servo controladores e incremento en los tipos de lenguaje de programación.

Figura 2. **Controlador lógico programable**



Fuente: Omron. *Autómata programable CP1H*. <https://industrial.omron.es/es/products/cp1h>.

Consulta: diciembre 2019.

1.4.2. Funciones básicas de un PLC

Las funciones básicas de un PLC comprenden a la base de un proceso a automatizar, y realizándolo de manera sistemática, de tal forma que se exponen de la siguiente forma:

- Puesta en alimentación
 - Comprobación de memoria y componentes
 - Comprobación de programa usuario

- Comprobación de módulos de entrada y salida I/O
- Comprobación de comunicación
- Desactivación de las salidas

- Fase de ejecución de programa
 - Adquisición de entradas
 - Procesamiento de ecuaciones lógicas
 - Actualización de las salidas
 - Diagnóstico interno
 - Funciones de comunicación

- Parada del sistema
 - Cese de ejecución de programa
 - Desactivación de las salidas

El PLC comprende programación, detección, mando y diálogo hombre-máquina (HMI) para los procesos automatizados.

La programación se refiere al tipo de lenguaje que se va a utilizar, siendo el más común el lenguaje tipo LADDER, que además es el más sencillo. Otros tipos de lenguaje son tanto GRAFCET como el tipo secuencias, y de tipo de compuertas.

El mando se refiere a enviar las instrucciones respectivas, para que el proceso se pueda llevar a cabo, que comprende a las medidas que proporcionan los sensores, y otros factores externos.

El diálogo hombre-máquina se refiere a la comunicación de PLC y sus instrucciones a un entorno físico-virtual controlable por el hombre. De manera

que, por ejemplo, el uso de botones o actualmente el monitoreo SCADA y activación de puntos de manera virtual le comprenden.

Las funciones básicas del PLC establecen igualmente los parámetros y diseño del proceso, cuyo caso suele ser la esencia del controlador lógico programable, debido a que el controlador puede ir concatenado a todos los procesos y sub-procesos que se puedan llevar a cabo, entonces funciona como el núcleo de operación en una industria automatizada.

1.4.3. Campos de aplicación

La utilización del controlador lógico programable (PLC) es muy extensa alrededor de los campos de la ingeniería, y se utiliza primordialmente en control de procesos, procesos de maniobra, señalización, entre otros. Cabe destacar que el uso del PLC suele ser una mejora en la eficiencia del proceso, ya que se obtiene mayores masas en menos tiempo, a comparación de un proceso manual secuenciado por humanos. De esta manera, es muy importante en el campo de la automatización de procesos, así como su debido control.

Entre sus utilidades en el campo industrial están:

- Señalización de procesos
- Maniobra de instalaciones
- Maniobra de máquinas
- Control de procesos
- Evaluación de programación en un proceso especificado
- Industria del automóvil
- Industria de maquilas
- Ensamblaje
- Dosificación

- Tratamiento de aguas residuales
- Embotellado
- Empaquetado
- Turbinas
- Control de motores
- Centrales eléctricas
- Instalaciones eléctricas

Los procesos previamente descritos comprenden a los distintos campos industriales como:

- Maquinaria
- Instalaciones
- Industria automotriz
- Industria química y petroquímica
- Metalúrgica
- Industria alimenticia
- Maderas y papeleras
- Producción de energía

En el caso de procesos continuos, también se puede aplicar en las señales de sensores en base a un valor determinado, este es el caso de un control PID.

Estos se dividen en tipo lazo abierto y lazo cerrado. El control de tipo lazo abierto se basa en la no realimentación del proceso al controlador, y el control de tipo lazo cerrado sí realimenta al controlador, por lo tanto se tiene un valor de referencia, y en base a la medición, el controlador tomará una decisión. En el caso del software Simumatik y Codesys, igualmente se pueden aplicar este tipo de procesos, sin embargo, se priorizarán los procesos secuenciales y discretos

de forma que se entienda la aplicación didáctica al laboratorio del curso de Automatización Industrial

1.4.4. Programación básica

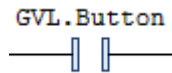
La programación de un PLC se enfoca en determinar y dirigir el procedimiento que se requiere en un proceso a automatizar. Ante varios tipos de programación, se enfocará en el lenguaje tipo LADDER, incluyendo los símbolos más utilizados y su significado en un proceso en la programación del programa.

Cabe destacar que la programación es una parte fundamental en la automatización de un proceso, debido a que es la descripción del proceso de interés. Así que, para evaluar la eficiencia de la programación, es importante simular el proceso mediante la programación para evitar errores futuros o evaluar mejoras en el proceso, porque la simulación debe evaluar el diseño y la descripción del proceso, descrito por la programación y por el diseño gráfico si se tuviese.

1.4.4.1. LD (Load) y LD NOT (Load Not)

(LD) Leer bit: se refiere al estado del bit, normalmente conocido como contacto normalmente abierto. Si el bit se encuentra activo, entonces el contacto se cierra y hay continuidad en el programa, de lo contrario los procesos involucrados por el contacto en ese punto se detienen.

Figura 3. **Contacto normalmente abierto**



Fuente: elaboración propia, empleando toma de pantalla de Codesys.

(LD NOT) Leer bit negado: se refiere al estado del bit, normalmente conocido como contacto normalmente cerrado. Si el bit se encuentra activo, no dará continuidad de ejecución en ese punto del programa, de lo contrario, simplemente dará continuidad a la ejecución del programa en ese punto y a los procesos involucrados.

Figura 4. **Contacto normalmente Cerrado**

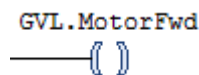


Fuente: elaboración propia, empleando toma de pantalla de Codesys.

1.4.4.2. **OUT Y OUT NOT**

(OUT) Salida: Para la ejecución de la salida, será del valor ON u OFF, dependiendo del rango respectivo. También es conocido como bobina, y depende de la continuidad del programa.

Figura 5. **Salida normalmente abierta**



Fuente: elaboración propia, empleando toma de pantalla de Codesys.

(OUT NOT) Salida Negada: para la ejecución de la salida, a un valor ON le corresponderá OFF, y a un valor OFF corresponderá un valor ON. Esto se conoce también como salida de bobina negada y depende de la no continuidad de los contactos y de la ejecución del programa.

Figura 6. **Salida normalmente cerrada**

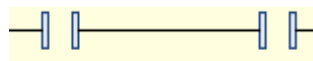


Fuente: elaboración propia, empleando toma de pantalla de Codesys.

1.4.4.3. **AND**

Se conoce también como función Y. Se refiere a contactos en serie, ya que se deben de cumplir ejecuciones para que un proceso obtenga una salida respectiva. De esta manera, se necesitan requisitos para una salida estricta. Los contactos en serie pueden ser negados o no negados, depende de la lógica programada.

Figura 7. **Contactos en serie**



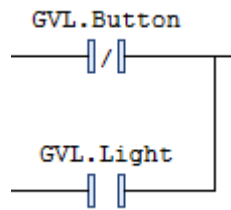
Fuente: elaboración propia, empleando toma de pantalla de Codesys.

1.4.4.4. **OR**

También conocida como función O, cuya función es establecer que, si alguna de las condiciones se cumple en la lógica de programación, hay continuidad en la ejecución. Estos son contactos en paralelo, e igualmente

pueden utilizarse indistintamente de forma negada o no negada, dependiendo de la lógica del proceso.

Figura 8. **Contactos en paralelo**

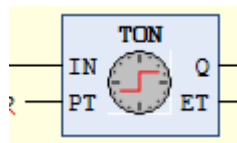


Fuente: elaboración propia, empleando toma de pantalla de Codesys.

1.4.5. **Temporizadores y contadores**

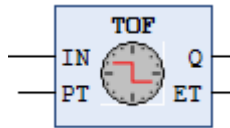
Estos tipos de controladores lógicos programables son muy utilizados en la industria, porque permiten programar cierto tiempo de trabajo en un proceso, conexión o desconexión de una tarea en específico también. Los más utilizados son los TON y los TOF, que son retardo a la conexión y retardo a la desconexión respectivamente.

Figura 9. **Temporizador TON**



Fuente: elaboración propia, empleando toma de pantalla de Codesys.

Figura 10. **Temporizador TOF**



Fuente: elaboración propia, empleando toma de pantalla de Codesys.

1.4.5.1. Temporizadores TIM

Cuando su condición de ejecución es ON, el temporizador se activa, y se resetea cuando su condición de ejecución es OFF.

De esta manera, se le reconoce como un temporizador con retardo a la conexión.

1.4.5.2. Contadores CNT

Los contadores son muy utilizados para descontar a partir del valor establecido (SV) cuando la condición de ejecución en el impulso de entrada pase de OFF a ON, el valor será reducido en uno de manera periódica, cada vez que CNT se ejecute con una condición de ejecución ON para la tarea en específico.

Si la condición de ejecución sigue de igual forma, o lo ha hecho de ON a OFF, el valor del CNT no cambiará. El indicador de finalización para un contador se pone en ON cuando alcanza cero y permanecerá en ON hasta que el contador se resetee.

2. CONCEPTOS DE COMUNICACIÓN INDUSTRIAL PLC

Es necesario conocer los objetivos de una comunicación industrial de PLC efectiva, debido a que para poder obtener una mejora de flujo de información y así estar estableciendo un control sobre los componentes de un proceso o sistema, los datos provenientes de los componentes de dicho sistema permiten verificar la eficiencia que se está obteniendo, así también como medidas de seguridad.

El control de procesos industriales de manera automática ha tomado una gran relevancia, y así mismo ha obtenido una mayor demanda en las industrias, debido a todas las ventajas que se pueden ofrecer, tal como seguridad, precisión, exactitud, aumento en la producción, aumento en la calidad, disminución de desechos, y así se efectúa el aumento de la eficiencia en un proceso, que es lo que toda industria y empresa busca, ser lo más eficiente posible.

Cabe destacar que, ante mayor demanda y producción de cierto producto, aumenta el uso de máquinas y sensores y actuadores, por consiguiente la información resulta ser mayor y de esta manera se necesita ampliar el proceso de comunicación, de tal manera que resulta necesario. Por consiguiente, es muy importante el uso de las comunicaciones industriales, debido a que un pequeño fallo en la maquinaria o en algún sensor y un proceso resulte afectado, disminuye de manera considerable la eficiencia y así el factor económico de la empresa.

2.1. Redes de comunicación industrial

Una empresa o industria se encuentra en la necesidad de incorporar y aumentar la eficiencia de sus procesos de manera directa y confiable mediante un sistema de comunicaciones integrado para poder obtener un beneficio, como pueden ser:

- La reducción de costes de producción
- La mejora de la calidad
- La mejora de la productividad
- La reducción del almacenaje
- La mejora de la efectividad de sus sistemas
- La reducción de los costes de mantenimiento

Y así también, para conseguir estos objetivos el sistema de comunicaciones debe permitir:

- Sistemas de comunicaciones que enlacen la planta de producción con la de gestión e ingeniería de la empresa.
- La integración de las bases de datos de la empresa.
- Compartir aplicaciones tanto a nivel de software como a nivel de hardware.

2.1.1. Comunicaciones industriales

A lo largo de la historia ha habido diferentes formas de comunicación entre las personas y entre dispositivos. La forma de comunicación por medio del teléfono fomentó las bases de la comunicación a como se conoce hoy, tanto a nivel social como a nivel de industria y tecnología.

Se podría realizar una comparación totalmente paralela a la expuesta anteriormente cuando se habla de la evolución experimentada en los dispositivos industriales y así en la forma de comunicación actual de los sistemas de información compartida, de manera que se expone de forma clara los parámetros de comunicación de un controlador lógico programable en este caso, y de esa manera se puede observar cómo esas mismas tecnologías que poseen los dispositivos utilizados por las personas aparecen implementadas en las comunicaciones entre diferentes dispositivos industriales y efectuar una transmisión de datos clara y concisa.

Un sistema de comunicación de datos industrial es tanto más eficiente cuanto más cerca del proceso y la comunicación se encuentran en la simulación del PLC, debido a que tanto el método que se decida elegir para simular el proceso debe ser de manera especificada a un PLC, y así poder implementar en algún caso de manera física, uno de los tipos de simulación que se expondrán posteriormente, o de manera virtual como es el caso de la comunicación por medio de OPC, y ciertas características enfocan su aplicación en distintas redes de comunicación de tal forma que es importante verificar los datos.

Son tres las principales características que determinan la aplicación de las diferentes redes de comunicación, como son:

- Volumen de datos
- Velocidad de transmisión
- Velocidad de respuesta

Ante los términos expuestos anteriormente, tomando en cuenta la relación de evaluación, simulación y transmisión, deben existir diferentes niveles de redes de comunicación de datos que cumplan en cada caso con las exigencias

funcionales especificadas. De ahí nace lo que se conoce como pirámide de las comunicaciones, que es reconocida por todos los fabricantes de dispositivos para las redes de datos, formada por cuatro niveles, exponen cada una de las tareas y objetivos a tomar en cuenta para una comunicación respectiva.

Figura 11. **Pirámide de comunicaciones**



Fuente: GUERRERO, Vicente. *Comunicaciones industriales*. p. 8.

Ante lo antes expuesto, es necesario conocer las ventajas de una comunicación eficiente en una empresa o industria, de manera de efectuar una comparativa especificada, como por ejemplo:

- Tras una solicitud de cierto efecto, todos los elementos de un sistema o proceso reciben de manera simultánea la información.
- Permitir el control de la producción, ya que todos los elementos deben almacenar y comparar la información respectiva.
- Centralizar señales de alarmas en distintos procesos, así se obtiene mayor seguridad de comunicación y efectividad en el proceso.

De esta manera, al efectuar una simulación, se podrían obtener los datos de manera real utilizando el modelo Hardware-in-the-loop que se expondrá posteriormente, que es una escala mayor al método Software-in-the-loop, siendo este último de manera interna, por lo tanto, es importante conocer la conexión entre las comunicaciones industriales y métodos de simulación.

Efectuando las ventajas de las comunicaciones industriales y la importancia de la simulación tomando en cuenta el proceso y diseño del mismo, se aumenta la eficiencia del proceso, obteniendo una mayor proximidad respecto a las distintas tareas del sistema, y dependiendo de las capacidades de la industria o empresa, se efectúa el método de simulación más conveniente a los recursos económicos y procesos a tomar en cuenta.

2.1.2. Topologías de redes

La topología de las redes suele ser un aspecto muy importante, se refiere al aspecto físico del cableado y de los equipos en un sistema. Algunos sistemas industriales cuentan con las siguientes topologías:

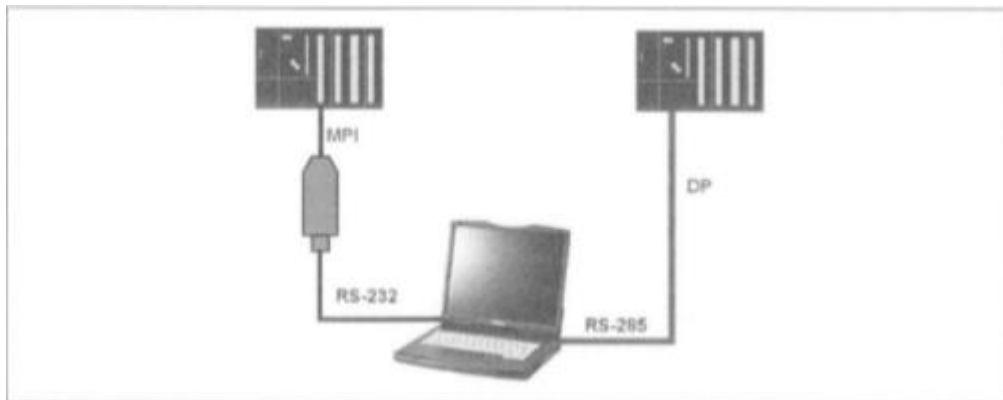
- Punto a punto
- Bus
- Árbol
- Anillo
- Estrella

2.1.2.1. Topología punto a punto

Es una de las topologías más simples entre los equipos, debido a su conexión directa entre dos ordenadores. Para esta topología no es necesario que

dentro de la trama del mensaje se incluyan las direcciones tanto de origen como de destino. El sistema de cableado es tan sencillo que la mayoría de las veces no necesita de un adaptador.

Figura 12. **Topología punto a punto**



Fuente: GUERRERO, Vicente. *Comunicaciones industriales*. p. 42.

2.1.2.2. **Topología de bus**

Esta topología se basa en una línea compartida por todos los nodos de la red de interés. Al ser un bus dividido por nodos, al enviar un mensaje cada nodo debe verificar si el bus está libre para que la información proceda, así que significa que solamente un mensaje puede circular por cada momento.

Aunque el coste de la instalación es bajo y el control del flujo es sencillo, es limitado por la distancia de atenuación, así que para distancias mayores se suele necesitar de repetidores para mejorar la atenuación en cada instalación de ordenadores.

Figura 13. **Topología de bus**



Fuente: Google. *Topología bus*.

<https://sites.google.com/site/topologiasdered708/home/topologia-bus>. Consulta: diciembre de 2019.

2.1.2.3. **Topología anillo**

Tiene cierto parecido a la conexión en bus, con la diferencia de que los 2 extremos se unen para formar un bus cerrado, por lo tanto la información fluye únicamente en un solo sentido y puede haber varios mensajes en el mismo momento, y el control es bastante simple desde el punto de vista de hardware y software.

Figura 14. **Topología anillo**



Fuente: Informática y Telemática. *Topología anillo* . <https://informaticaytelematica>. Consulta: diciembre de 2019.

Dependiendo de la topología utilizada, mediante la partición y envío de datos hacia los distintos ordenadores o softwares, se obtendría un control a tiempo real mediante la simulación y verificación de los mismos. Tomando en cuenta varias de las ventajas y desventajas de cada una de las topologías, brevemente se hace notar la importancia del envío de datos y mediante qué topología utilizar.

2.1.3. Técnicas de control de errores

Si mediante el envío de la información, suele ocurrir un error de mensaje, se impone tratar de corregirlo. Así que, existen 2 tipos de corrección a estos errores:

- Corrección hacia adelante
- Corrección hacia atrás

La corrección hacia adelante utiliza la información de relevancia que el emisor introdujo a la información enviada o mensaje, el receptor localiza los posibles errores y trata de reconstruirlos, debido a la gran pérdida de eficiencia en el sistema y el envío de datos en tiempo real, hay métodos de corrección que son los siguientes:

- Método Reed-Muller
- Método de Orchard
- Método de Hamming

La corrección hacia atrás sucede cuando el receptor no dispone de los medios para corregir el error, porque solicita el reenvío de los datos o mensaje, y

el coste de la retransmisión aumenta. Éste es el sistema utilizado en las comunicaciones industriales por ser un volumen de datos relativamente pequeño.

2.2. Redes de comunicación industrial Ethernet

Actualmente, Ethernet es con una proporción aproximada de un 80 %, el más importante en redes LAN. Las múltiples posibilidades del uso de internet, junto a extranet e intranet son muy aprovechables en el ámbito de automatización manufacturera y de procesos.

La tecnología de Ethernet permite al usuario adaptar de manera eficiente las necesidades en una red precisa de exigencias. La velocidad de transmisión de datos se puede elegir según las necesidades a tomar en cuenta y posee ventajas esenciales respecto a la comunicación y recepción de información.

Hay tanto componentes pasivos como componentes activos en el uso industrial de Ethernet, que se componen de la siguiente manera:

Tabla I. **Componentes activos y pasivos en red de comunicación**

COMPONENTES ACTIVOS	COMPONENTES PASIVOS
Cables	Switchs
Herramientas pelacables	Tarjetas de comunicación
Conectores	Módulos

Fuente: elaboración propia.

De manera que las tarjetas de comunicación, tales como CP 243-1 para el PLC S7-200 y CP 443-1 para el PLC S7-400, siendo muy utilizadas industrialmente, tienen las siguientes características:

- Están elaboradas para funcionar en ambientes industriales duros
- Funcionalidad en TI
- Eficiencia en la comunicación

Por lo cual, se ofrecen distintos tipos de módulos según el PLC obtenido, de manera que, según la actualización y software empleado, la eficiencia del PLC se asegura en la comunicación de estos módulos, obteniendo una mejora en el proceso de la lectura y recepción de datos empleados.

2.2.1. Características de Ethernet

La tecnología Ethernet es muy utilizada actualmente, donde industrialmente es muy ventajoso debido a sus características efectivas y mejora de los procesos, donde las características más esenciales son:

- Puesta en marcha de manera rápida, debido a un sistema interconectado de manera simple, por lo tanto es muy eficiente.
- Alta disponibilidad, muy útil en ampliaciones de ordenadores o del sistema en general.
- Rendimiento de comunicación prácticamente ilimitado, donde es posible variar la velocidad de transmisión de datos y amplificación del mismo.
- Posibilidad de acoplamiento entre diferentes sistemas, dando lugar a un sistema WAN o internet.
- Reserva de ancho de banda en LAN.

En el caso del monitoreo de datos mediante la tecnología Ethernet, se puede monitorizar el comportamiento del sistema en tiempo real, tanto de control como de lectura. Esto sirve para observar los tipos y cantidades de tráfico dentro de cada operación activa en un proceso determinado, de tal forma que la comunicación sea efectiva. Del mismo modo, se identifica el uso inesperado de la red y se detectan fugas de los sistemas y equipos primarios a los secundarios, donde la corrección de errores en el mensaje debe de ser empleado.

Tanto los puertos espejo en *switch* con Ethernet industrial como el historial y las estadísticas pueden usarse de manera que sirvan para identificar el comportamiento de los diferentes ordenadores o PLC's para la medición respectiva en sensores. Los usuarios pueden identificar fácilmente los inconvenientes y conocer la manera en que se está utilizando un ancho de banda mayor al estimado en la transmisión de datos o en el envío de información a otros sistemas en una misma red.

Desde el punto de vista de la seguridad, la integración creciente de tecnologías de la comunicación y Ethernet Industrial y su continua mejora para ofrecer nuevos niveles de beneficio y eficiencia en los sistemas industriales y manufactureros, también plantea posibles vulnerabilidades a un sistema dado.

Al monitorear información procedente de sistemas de control en niveles de planta, significa que el equipo también se extiende en la otra dirección de la transmisión de datos. Esto aumenta enormemente la vulnerabilidad de la ampliación de la red de intrusiones y amenazas.

Estas medidas de seguridad de red se pueden agrupar en varias categorías, incluidas las de control de acceso y autenticación, seguridad en la

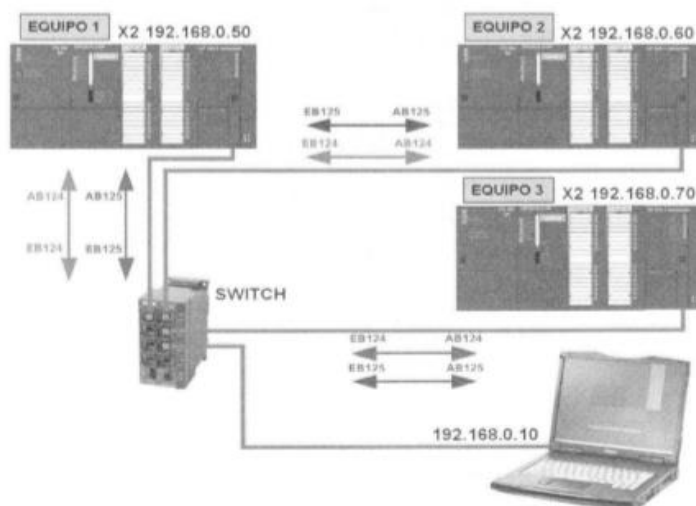
conectividad y administración, de manera que la confidencialidad de la información debe ser importante mediante sistemas de medición continuos.

2.2.2. Comunicación entre varios PLC mediante protocolo TCP/IP

Como la comunicación entre varios PLC en un ambiente industrial debe ser por medio de algún protocolo basado en la tecnología de Ethernet, de esta manera se asegura una comunicación efectiva de datos y de lectura.

Para poder realizar la respectiva comunicación, es necesario utilizar una tarjeta de comunicación compatible con el tipo de PLC a utilizar, y por consiguiente de esta forma es evidente la realización de sistemas de datos compartidos de sensores a un PLC, obteniendo así una eficiencia en comunicación y en control de los procesos respectivos.

Figura 15. Relación de comunicación entre 3 PLC



Fuente: GUERRERO, Vicente. *Comunicaciones industriales*. p. 278.

La configuración de cada PLC debe ser respecto a cada CPU disponible en el sistema, dando así una dirección IP por ordenador. De esta manera, el modelo se podría reforzar mediante más PLC siempre con este ordenamiento, todo conectado a una red Ethernet.

Una de las opciones de protocolos a utilizar es mediante el protocolo TCP/IP, referente a una dirección IP por PLC respectivo, así que la forma de creación de enlaces es de la siguiente manera:

Tabla II. **Ejemplo de enlace de protocolo**

NÚMERO DE ENLACE	NOMBRE	PUERTO	EST. ACTIVO COMO ENLACE
1	X	2001	Sí

Fuente: elaboración propia.

Cabe destacar que el uso de estos protocolos y configuración de varios PLC, es debido a control de macro sistemas, así que es importante mencionar debido a la continua actualización de formas de comunicación entre distintos componentes.

Una industria que posea diferentes tipos de procesos unificados a un sistema podría hacer uso de estos protocolos para poder simular y verificar dichos procesos, mediante el modelamiento y control de motores y generadores a nivel hardware, o a sistemas más sencillos a nivel software.

2.3. Servidores OLE for Process Control (OPC)

Un servidor OPC permite un lenguaje común para el intercambio de información entre diferentes tipos de dispositivos y fabricantes en el área de la automatización y el control industrial, obteniendo así distintos fabricantes y gamas de productos que puedan compartir información mediante un lenguaje común que todos procesen y tenga entendimiento entre ellos. Para eso, es necesario protocolos en común que se relacionan en el proceso a automatizar, así pudiendo simular de manera efectiva dicho proceso.

Este tipo de servidores surgieron como solución de la gran diversidad y diferencias de protocolos propietarios que existían en los inicios de la automatización con equipos electrónicos. Como respuesta a estos inconvenientes, nació la “OPC Foundation”, la cual surgió como un ente regulador de los estándares que permiten la comunicación y procesamiento de datos entre dispositivos de automatización de diferente tipo.

Cada estándar nace del consenso de la industria y el sector académico, y que para tal fin se generan comités creadores multidisciplinarios sobre OPC en específico, en donde participan las principales empresas del mundo en el área de la automatización y universidades con investigaciones en el área, tales como Ford o la universidad de Estocolmo, donde surgen artículos sobre avances en este sistema.

A partir de los estándares OPC mencionados, nacen aplicaciones especializadas a este método, donde las actividades comerciales son el mayor ejemplo como soluciones en la industria. Pero para poder aplicar este tipo de método de comunicación y dar solución a problemas específicos, es

indispensable conocerla, y obtener información sobre qué alternativas y formas de implementación se cuenta.

2.3.1. Conceptos generales OPC

El servidor OPC, es el software encargado de acceder directamente a los datos comunicándose con la fuente primaria de datos en distintas fuentes de comunicación; este intercambio de información con la fuente (comúnmente un PLC), se realiza usualmente mediante un bus de campo según el protocolo de comunicación utilizado, de esta manera se puede definir al cliente OPC, como el software que sirve de mecanismo genérico de acceso a datos de campo, poniéndolos a disposición de las aplicaciones de usuario. Por ejemplo, en una utilización de simulación por el método HIL o SIL, mediante la herramienta Labview, que se basa en una supervisión de información, serviría como ejemplo base de esta herramienta, siendo de la siguiente forma:

Figura 16. **Ejemplo de Diagrama de Bloques HIL utilizando Labview**



Fuente: GODOY, Antonio. *Plataforma hardware-in-the-loop basada en labview, plc y opc. Experiencia de uso docente en automatización y supervisión.* p. 5.

Donde se obtiene un lenguaje en común tanto para un PLC y para Labview, obteniendo así una comunicación efectiva y simulación de los procesos que ejerce un PLC.

La ventaja de separar el acceso a los datos con un servidor y un cliente, es que de esta forma y un servidor OPC, los clientes pueden acceder de forma general a cualquier servidor de datos OPC de comunicación efectiva, evitando que cada aplicación de cliente necesite conocer o implementar la comunicación con todos los posibles hardware, los cuales son principalmente autómatas programables (PLC), de los diferentes fabricantes.

Se le delega a cada fabricante el acceso a los datos de sus controladores programables, exigiéndole, no obstante, que los publique de forma estándar bajo algún modelo de la OPC Foundation mediante un servidor OPC.

Para la propuesta de uso de Simumatik, este software posee entre sus descargas un uso de OPC, que tiene previamente ya implementado dentro del software. De esta manera no se realiza una comunicación externa con un OPC, sino que se realiza de forma local mediante un procedimiento general de OPC-PLC, en este caso, el uso de Codesys.

En cuanto al acceso a datos de campo, la solución más sencilla es la utilización de un producto ya construido, en este caso el uso de Simumatik y Codesys, que permita el acceso a datos del proceso.

Los sistemas de supervisión son el software que comúnmente se usa para el acceso, manejo y visualización de los datos de proceso, como Labview o Mathlab, aunque dicha comunicación suele ser un poco más compleja, no suelen tener sus propios clientes y servidores OPC implementados, con los que el usuario en tiempo de desarrollo interactúa para crear la aplicación.

En el caso del autómata S7-1200 de Siemens existe incluso un simulador proporcionado por el propio fabricante. Sin embargo, desde un punto de vista

didáctico, el empleo del HIL o SIL favorece una percepción más realista del proceso de diseño y validación de controladores, debido a prueba de funcionamiento de dicho PLC. Se tratan cuestiones relativas a las comunicaciones industriales (protocolo OPC) y la monitorización y supervisión (LabVIEW).

Cabe remarcar que tanto el control como la comunicación PLC-PC a través de OPC son realizadas de forma real, no simulada, así que de esta manera se utiliza Simumatik y Codesys, ya que la comunicación es realizada de manera de simulación.

En un inicio las primeras tecnologías representadas por las siguientes siglas OLE, DDE, COM y DCOM OLE permitieron la comunicación entre aplicaciones y dispositivos. DDE2 permitía compartir sesiones para el intercambio de datos entre aplicaciones y llegó a ser utilizado como interfaz entre los autómatas programables de los procesos, y los computadores.

Debido a la no globalización de la utilización de controladores y de autómatas programables de manera conjunta, el desarrollo de posteriores tecnologías para optimizar el rendimiento de DDE trajo consigo que los fabricantes diseñaran soluciones de tipo propietario con características diferentes e incompatibles entre ellas, de esta manera solamente unos controladores y softwares de lectura funcionaban con cierta marca o tipo de PLC.

OPCUA es una manera y método de OPC para obtener y convertir datos en tiempo real de manera local y rápida, de esta manera igualmente, Simumatik obtiene implementado este tipo de OPC.

2.3.2. Comunicación OPC-Software

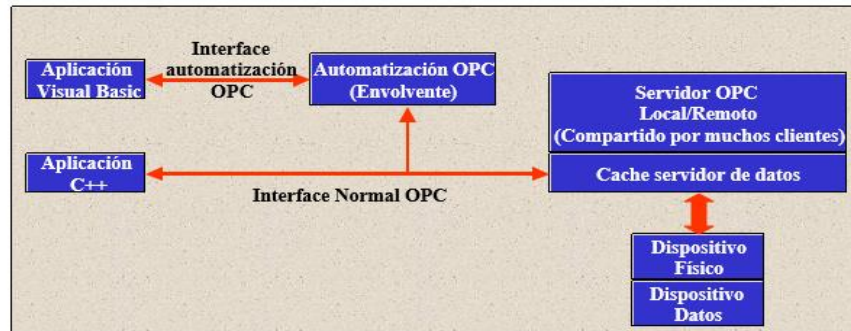
OPC se diseñó para permitir aplicaciones donde el cliente tenga acceso a datos Simulación-Planta de una manera consistente y a tiempo real. Con aceptación por parte de los principales fabricantes, OPC proporcionará beneficios tales como:

- Los fabricantes de Hardware sólo tienen que desarrollar composiciones al software para que los clientes los puedan utilizar en sus aplicaciones.
- Los diseñadores de Software no tendrán que volver a reescribir sus Drivers (Upgrade), debido a cambios de las características de su Software, para el reconvertimiento del lenguaje.
- Los Clientes tendrán más opciones de desarrollar sus sistemas de Simulación- Planta, haciendo uso de la integración de una gama más amplia de sistemas de Hardware de diversos fabricantes en el caso de una simulación HIL.

Debido a dichas ventajas, es de manera generalizada una muy buena opción para proveer bases de fundamento para comunicaciones de OPC más establecidas en el mercado, según el software establecido, así será necesario su crecimiento de comunicación de OPC.

Debido a la arquitectura que se debe de obtener mediante este tipo de conexiones, una forma muy explícita de la explicación de esto es mediante la siguiente figura:

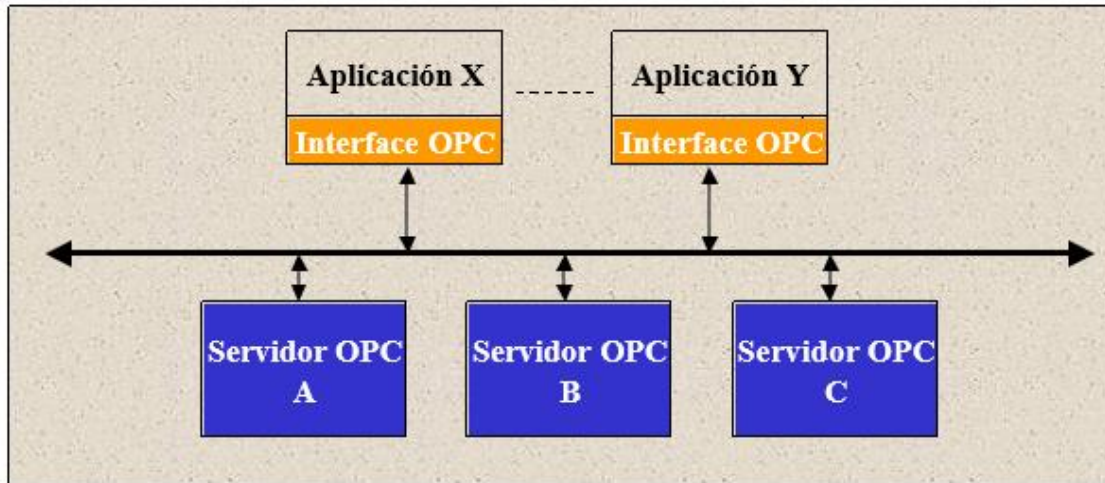
Figura 17. **Arquitectura típica OPC**



Fuente: ETITUDELA. *Opc: un estándar en las redes industriales y buses de campo*. p. 18.

Y por consiguiente, el uso del OPC puede ser utilizado mediante varios tipos de servidores, por tanto el uso de la comunicación entre Codesys y Simumatik puede funcionar entre varios proyectos, debido a la virtualización y partición del servidor, así que es posible establecer diferentes conversiones de código mediante OPC a tiempo real, tanto desde un PLC hacia un controlador (FPGA, entre otros.) de manera que son varios funcionamientos en un mismo tiempo del software de la siguiente manera:

Figura 18. **OPC en varios servidores**



Fuente: ETITUDELA. *Opc: un estándar en las redes industriales y buses de campo.* p. 8.

El servidor OPC genera de cierta forma una línea de comunicación entre Hardware y Software, donde recorre la información de manera que un servidor OPC provee un mecanismo para recolectar los datos de un emisor de información y comunicarla a cualquier usuario o cliente, en forma general y directa, así también como una línea de información.

OPC proporciona una interface entre el servidor y cualquier cliente, para así poder acceder a los dispositivos y obtener la información, comunicándola ya sea a un controlador físico, o a una Planta simulada mediante Codesys y Simumatik.

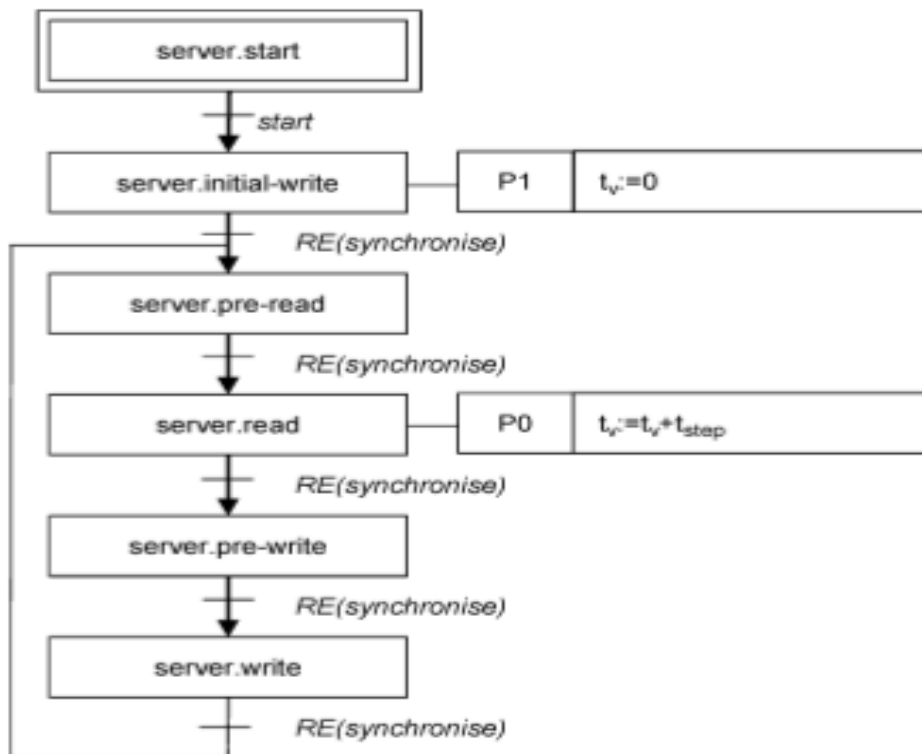
2.3.3. Comunicación OPC-PLC

Definidamente, la comunicación OPC-PLC ya suele estar establecida previamente por parte del servidor OPC o propiamente del software del PLC en

donde se establecen todos los parámetros consecuentes al modelo y tipo de programación establecidos.

Danielsson explica en su artículo *Methods for Reliable Simulation-Based PLC Code Verification* que es de suma importancia un estado de ambiente virtualizado donde un PLC pueda ser establecido, ya sea teniéndolo físicamente o simuladamente, en donde se puedan verificar las mediciones de sensores también, donde utiliza una serie de órdenes al servidor OPC para la simulación de una sincronización de un motor simulado, que es lo siguiente:

Figura 19. **Ejemplo de sincronización de motor por órdenes OPC**

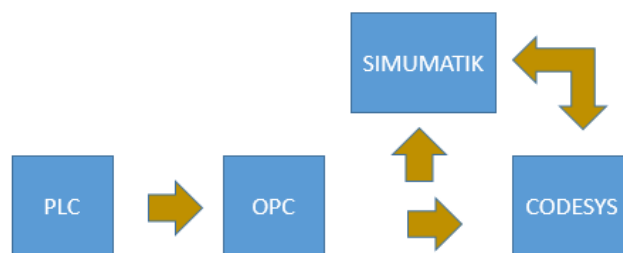


Fuente: SVENSSON, Bo. *Methods for reliable simulation based plc code verification, IEEE.*
p. 272.

De esta manera, Danielsson usa un OPC que su conexión es determinada mediante series de órdenes manuales, previamente descritas en su proyecto. Para fines de sincronización, las diferentes órdenes de sincronización pertenecen a ambientes diferentes de dicha sincronización, de manera que el servidor avise y provea la información a la simulación.

En este caso, la comunicación OPC-PLC es directa, siendo el intermediario OPC o el software a usar del PLC, de esta manera no es necesario conexiones externas o uso de servidores locales externos al software. Una de las ventajas del software Simumatik es que su servidor OPC es compatible con el software Codesys, por lo tanto toda la comunicación se realiza mediante el software y de manera local, ahorrándose comunicaciones agregadas al procedimiento, siendo de la siguiente manera:

Figura 20. **Proceso PLC-OPC**



Fuente: elaboración propia.

El proceso será simplificado con este software de manera que no serán necesarias las órdenes manuales, debido a que la programación del PLC en Codesys será convertido a un lenguaje común desde el OPC para que el software Simumatik pueda reconocer las órdenes y los comandos.

3. ALTERNATIVAS DE MÉTODOS DE SIMULACIÓN PARA AUTOMATIZACION INDUSTRIAL

Actualmente, el realizar una simulación parecida al modelo de planta que se tiene, dependiendo del tipo de simulación que se requiera, es una ventaja en el aspecto económico y metodológico.

Ante este avance, es posible ahorrar recursos en pruebas no eficientes o a daños de equipo al tener un error de órdenes hacia el controlador lógico programable, y se establece de manera ordenada la secuencia del proceso, estableciéndola antes de la implementación en una planta o modelo de máquina.

Simular consiste en usar un modelo establecido para verificar cierta variable de salida, con el objetivo primordial de producir efectos similares ante idénticos estímulos, de manera que no sea necesaria la intervención física de los componentes.

En una simulación es necesario conocer el valor de las entradas (editables o no), porque, lo que se pretende calcular es el valor de las salidas que dependen de ellas, y ver el efecto de la simulación con la intervención de estas.

Para realizar la simulación se necesita establecer las variables ante el modelo verificado, de manera que las siguientes variables dependientes sean comprendidas de los efectos, estableciendo así un comportamiento verificable, debido a que las industrias se ven forzadas a construir costosas plantas piloto para simular el comportamiento de nuevos procesos o efectos para aplicarse después en la planta real.

D'Arthenay establece que la simulación en una segunda planta suele ser complicada, costosa y a veces imprecisa dado los cambios de escala entre la planta piloto y la planta real, y que de allí nace la necesidad de desarrollar en computadores simulaciones de plantas industriales.

La simulación es una herramienta que se desarrolló en paralelo con la aparición del computador y que la idea de complementar al desarrollo industrial se ha establecido debido a la mayor velocidad y capacidad de gráficos y memoria que las computadoras han estado teniendo.

3.1. Model-in-the-loop

El método de simulación Model-In-The-Loop (MIL), parte como la base de simulación de los 2 métodos posteriores a explicar. En este método de simulación, se fomentan las necesidades de la planta, tipo y manera del proceso a simular, de manera que es meramente el diseño en sí de la variable a simular.

3.1.1. Definición

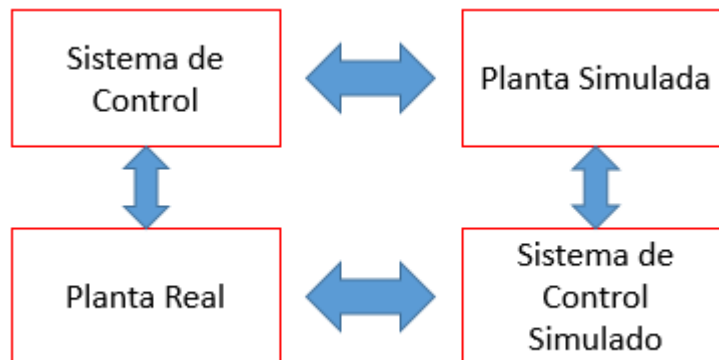
El método de simulación Model-In-The-Loop se define como el diseño preliminar a simular, de manera que debe ser la base de simulación de todo el proceso. En esta etapa, se determina el modelo de la planta o máquina a simular, tanto a nivel físico de simulación, como a programación de los componentes que le conforman. Entonces, se define el modelo a simular, sin haberlo simulado a nivel software o a nivel hardware.

Cabe destacar que se enfoca meramente en el funcionamiento del diseño del modelo a ser simulado posteriormente. Bergstrom en su tesis en la

universidad de Lund, da a entender que es la base de la simulación virtualizada sin haber sido simulado de ninguna forma, y se entiende solamente como diseño.

El método MIL no solamente comprende a nivel de construcción de componentes físicos en una simulación, o la programación de estos, sino que se refiere también al diseño por medio de diagrama de bloques, o de diseño por medio de instrucciones en el caso de Simulink o Matlab. La estructura de cómo va conformado este método es de la siguiente manera:

Figura 21. **Estructura MIL**



Fuente: elaboración propia.

Donde se refiere puramente a diseño, sin simulación depurada, debido a que ese objetivo se subdivide en los métodos SIL y HIL, por consiguiente, la estructuración previa corresponde a la base de la simulación automatizada a profundidad, concatenando el lenguaje en común entre software y hardware, a un diseño simulado.

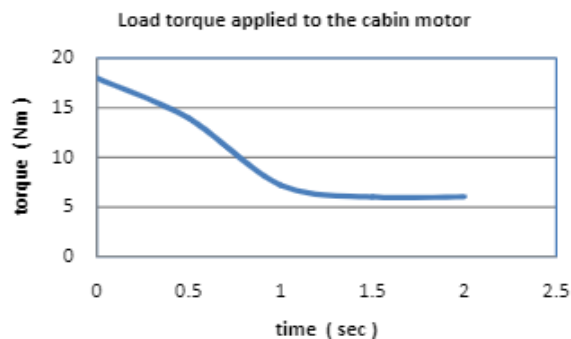
3.1.2. Funciones de la metodología MIL

Dentro de las funciones más elementales está el diseño por medio de simulación de parámetros, utilizados comúnmente en softwares como Matlab o Simulink, donde el diseño de la simulación tiende a reflejar posibles efectos de un diseño preliminar.

Como explica Dewasurendra las demandas de una simulación MIL sostienen una mejora continua en la calidad de los diseños a verificar, obteniendo así los resultados operantes del sistema mediante un previo diseño e ir mejorándolo a tal punto de que una simulación SIL y HIL pueda ser posible.

Para ejemplificar, en Matlab es posible simular comportamientos dados, igualmente en Simulink entre otros softwares. Por consiguiente, al emular un modelo de cierta máquina y realizar el modelo, se procede a la simulación y obtención de resultados según el modelo establecido mediante las debidas ecuaciones de comportamiento y de ejecución de ciertos ambientes de trabajo, por ejemplo:

Figura 22. **Torque con carga aplicado al motor de inducción simulado**



Fuente: DEWASURENDRA. *Model in the loop testing of Complex Reactive Systems, IEEE.*

p. 5.

En este ejemplo, previamente se había establecido el modelo referente al motor de inducción simulado, de esta manera, se puede simular el comportamiento ante ciertos ambientes de manera paramétrica, en estos tipos de software, de manera que el diseño es mediante el modelo exacto de una máquina, y cierto modelo se puede complementar con un controlador para dar origen a una simulación SIL y HIL.

Otra de las funciones, que es la que se tomará en cuenta, se basa en el diseño del modelo de una planta mediante la estructuración de componentes a coordenadas, tomando lugar en ambiente 3D modificable, de manera que los comandos ya están establecidos dentro de las variables a tomar en cuenta y mediante un diseño de programación para la ejecución de órdenes, se establece el modelo MIL, previo a ser simulado para lograr una simulación SIL.

El software a utilizar para el diseño de la planta mediante 3D será Simumatik, y el diseño de la programación del PLC para la ejecución de órdenes será el software Codesys, esto debido a su simplicidad y a su bajo costo, por lo que funcionará como base para software más avanzado de mayor costo.

Hay variedad de software válidos para realizar este tipo de simulaciones, y para profundizar en las técnicas previamente, se ejemplificará mediante otro tipo de software la referencia a esas técnicas, como, por ejemplo:

Figura 23. **Ejemplo de diseño de célula de producción PSL en SketchUp**



Fuente: YAO, Charlie. *Simulation-based verification of plc programs*. p. 20.

Esta célula de producción se basa en el diseño preliminar mostrado, de la misma manera funciona el software Simumatik, donde este es un claro ejemplo de la estructuración por coordenada en un modelo basado en 3D, donde se le debe vincular las distintas variables asociadas a una instrucción para el PLC.

Luego se establece el modelo de órdenes para el control del diseño 3D, y esto sucede simplemente mediante la programación del PLC asociando las variables y órdenes hacia el diseño ejecutado por coordenadas.

3.1.3. Ventajas y desventajas del método de simulación MIL

El método MIL comprende la base fundamental de las simulaciones posteriores más avanzadas, pues conlleva la parte más importante que es mediante el diseño en todo sentido de la planta, máquina o proceso a simular.

Esto también comprende los conceptos y variables de manera singular hacia el conjunto de programación a tener en cuenta, siendo así primordial la simulación por etapas para verificar la eficiencia y el desarrollo de los procesos, así que en el siguiente cuadro comparativo se relacionan las ventajas y desventajas de la utilización del método de simulación MIL:

Tabla III. Cuadro comparativo de las funciones MIL

	Realizado por MIL
Diseño preliminar del proceso	✓
Diseño preliminar de la máquina	✓
Diseño preliminar de la planta	✓
Diseño preliminar de la ejecución de órdenes	✓
Verificación de datos	
Comunicación OPC	
Verificación del diseño	
Simulación efectiva del diseño correspondiente	
Simulación efectiva de la ejecución de órdenes	
Edición de diseño	✓
Verificación de errores	
Comprobación del diseño según controlador real-simulado	

Fuente: elaboración propia.

Generalizando el concepto del método MIL, es que se basa en una simulación meramente del diseño base para sus posteriores simulaciones de comportamiento. Esto comprende:

- Diseño de la programación del PLC
- Diseño de la programación de cualquier otro controlador
- Diseño físico simulado de una planta, máquina o proceso
- Diseño del diagrama de bloques de un proceso o comportamiento
- Diseño de la formulación de cierto fenómeno o comportamiento de cierto evento basado en un proceso, máquina o planta.

Por consiguiente, basado en la definición del método MIL, será complementario a los métodos de simulación SIL y HIL, debido a que estos últimos se enfocan meramente en simulación virtual y física correspondientemente, basado solamente en diseño, se efectúa la arquitectura para dichas simulaciones.

En el uso del método MIL, no comprende variables más allá de puramente diseño, así como la comunicación por medio de OPC, verificación de datos, verificación de errores, sino meramente en el diseño y edición de este mismo.

3.2. Software-In-The-Loop

El método Software-In-The-Loop (SIL), es la parte encargada de concatenar la comunicación por parte del diseño en un software respectivo con el diseño de la programación, obteniendo así una simulación de comportamiento efectiva y obteniendo de manera virtual la forma de trabajo de los entornos diseñados.

Normalmente, esto se puede notar de manera didáctica en cualquier software de programación de PLC ante ciertos ambientes predeterminados diseñados para practicar la lógica, especialmente ambientado en un ambiente 2D, sin embargo, el objetivo de este trabajo es de fomentar este método de simulación mediante un software de construcción y diseño en 3D, y comunicar ese software para su debida programación, convirtiendo de esta manera en un lenguaje en común entre ambos tipos de software, para la debida simulación de estos escenarios realizados por el usuario.

En la realidad, el PLC está conectado de forma física con el hardware obteniendo así mayores riesgos para el hardware, de manera que puede afectar una línea de producción para las pruebas del sistema. En orden para no permitir esto, una de las alternativas más sólidas es obtener un modelo de simulación aproximado al modelo físico real.

Simulado el controlador y la planta o máquina, se obtiene la simulación efectiva llamada SIL, donde previamente se establece el diseño mediante el método MIL, para posteriormente obtener los resultados y comportamientos mediante una simulación de datos, llamada también método de simulación SIL.

3.2.1. Definición

La simulación SIL se define como la simulación virtual de controladores, máquinas o plantas sin la intervención de hardware o elementos físicos, siendo así una manera muy ventajosa de pruebas a equipos estableciendo el modelo anteriormente indicado, no afectando el equipo físico que se tiene.

La formalización del diseño consiste en la creación de modelos que contienen no solo la estructura del sistema sino también la definición de la funcionalidad. En el caso de que los modelos sean ejecutables en un software de simulación, la prueba basada en simulación de componentes individuales, así como el sistema general, es factible. Los resultados de la simulación se compararán con la definición de requisitos del proyecto.

Tan pronto como ocurran las diferencias, deben resolverse mediante un ajuste iterativo de los modelos o una definición de proceso más precisa. Cuando la funcionalidad general de los modelos cumple con todas las solicitudes de la definición, se ejecuta una generación automática de código durante la cual se genera el código para el controlador de destino común entre los softwares de diseño y de programación.

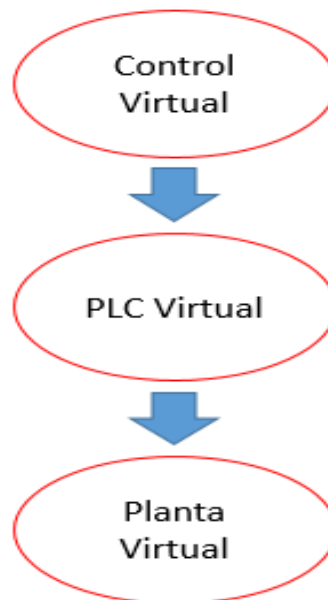
Para la verificación final del código generado se llevan a cabo pruebas de software en el bucle. En estas pruebas basadas en simulación, el programa de control real se combina con componentes de modelo de máquina y operador que proporcionan señales de solicitud y respuesta.

Es importante recordar que no solamente se puede lograr mediante 2 o más software, porque hay softwares muy avanzados actualmente que contienen esta comunicación ya establecida, o establecida a ambientes predeterminados. Se

debe resaltar que el concepto de simulación SIL es válida para esos programas de simulación 2D con ambientes preestablecidos, sin embargo, no se tiene profundidad en el diseño actual de la planta.

La forma establecida del trabajo utilizando el método de simulación SIL se establece meramente de manera virtual, sin la intervención de hardware, sensores o máquinas físicas, por lo que para la simulación SIL de cualquier planta mediante un PLC o cualquier otro controlador se puede determinar de la siguiente manera:

Figura 24. **Estructura SIL**



Fuente: elaboración propia.

Basado en un funcionamiento virtual, cada una de las etapas y pasos previamente descritos, es necesario el enlace de estos elementos obteniendo así

una simulación efectiva. El control virtual se refiere a la programación del PLC que irá concatenado a un PLC virtual, de manera que se exprese el lenguaje utilizado y se convierta a un lenguaje en común con el software de diseño mediante el servidor OPC, donde el PLC virtual es representado por el software donde se desarrolla el control, en este caso es Codesys.

Lo referente a la planta virtual, es el diseño de la planta elaborada mediante el software de diseño, en este caso Simumatik, donde se tendrá la elaboración de las variables a manipular igualmente, consiguiendo así el establecimiento total para su simulación tipo SIL, haciendo referencia a que siempre es necesario de un diseño MIL, porque el método SIL se enfoca meramente en la comunicación y simulación de los procesos, tomando en cuenta el resultado final y mediciones adecuadas.

3.2.2. Funciones de la metodología SIL

La metodología SIL se utiliza mayormente como simulación de procesos industriales a manera de comprobar el funcionamiento de la lógica de la ejecución de órdenes de cierto controlador.

Tanto un controlador PLC como FPGA por ejemplo, se pueden utilizar de forma que no sea necesario un controlador físico, de esta forma supone un ahorro de costo mayor, debido a que solamente se necesita el software de programación compatible con un servidor OPC y un software de diseño para poder lograr una simulación del proceso para posteriormente emularla en real.

En general, la simulación se utiliza como una herramienta de experimentación para encontrar soluciones óptimas para el sistema. Lo hace con procesos que no son necesariamente similares a los del sistema. Es la salida que

es similar a la salida de los sistemas reales. La simulación se puede utilizar para calcular problemas de logística en la producción y distribución de productos para, por ejemplo, encontrar cuellos de botella.

Es posible obtener la optimización de la producción más rápido que las formas tradicionales al realizar múltiples ejecuciones con diferentes entradas en una fracción del tiempo en una computadora. De hecho, la simulación también se puede usar cuando el sistema ni siquiera existe, para probar cómo debería funcionar el sistema y poder encontrar los cambios necesarios en el diseño del sistema.

Al sacar conclusiones sobre lo que muestra la simulación, siempre se deben tener en cuenta las incertidumbres. Los errores pueden acumularse y, al final, crear una desviación mucho mayor que las entradas pueden sugerir. Por lo tanto, el modelo de simulación no es necesariamente similar al sistema real, el resultado puede desviarse y las conclusiones deben tener esto en cuenta.

La emulación intenta hacer coincidir exactamente la salida del sistema y producir el proceso lo más similar posible a los del sistema. El propósito de emular es sustituir el sistema real, el utilizar un hardware con software haciendo lo mismo. Los beneficios de hacer esto son principalmente que la experimentación se puede hacer sin dañar el equipo costoso y aun así obtener una retroalimentación que sea confiable.

Sin embargo, el beneficio de un menor consumo de tiempo no es tan grande como en la simulación, debido a que el sistema se ejecuta en tiempo real y el equipo circundante aún podría ser parte del sistema real y, por lo tanto, funciona a la misma velocidad que lo haría en el sistema real. Sin embargo, los cambios en el modelo requieren mucho menos tiempo y, por supuesto, también cuestan

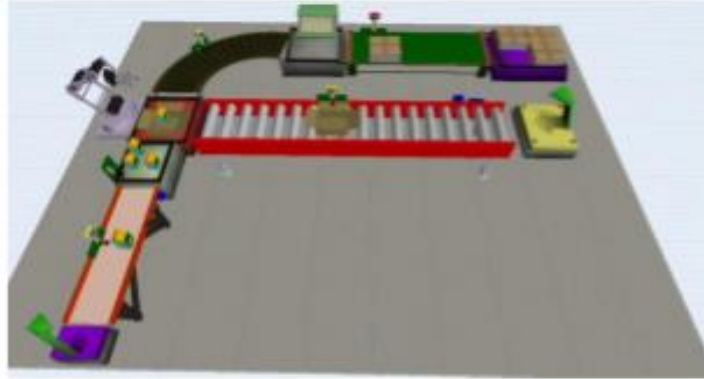
menos. Los modelos de emulación a menudo se usan como una herramienta para verificar el rendimiento de partes del sistema real sin ningún riesgo considerable, debido a que, si el programa es consecuente a la lógica de programación al utilizar el método SIL, se puede proceder a un hardware real usando el método HIL.

Taipalus determina que el objetivo principal de este tipo de soluciones a mejorar la eficiencia de un proceso, se debe a la razón de hacer un modelo virtual de cierto elemento en específico, verificar cómo funciona y minimizar el riesgo de problemas antes de hacer la máquina o proceso real.

Esto reduce los costos de producción porque si puede encontrar los problemas en esta etapa, la reparación no es de mucho valor, por lo que también facilita la enseñanza porque no necesariamente necesita la máquina real cerca, porque puede usar el modelo 3D virtual y verificar su comportamiento.

Un claro ejemplo de la funcionalidad de este tipo de simulación es el proyecto elaborado por Tebani, K., donde simula una parte de planta por medio de una banda transportadora, utilizando el software de diseño Flexsim, un servidor OPC externo y programación de PLC tipo *Ladder*, donde el diseño de la planta es el siguiente:

Figura 25. **Ejemplo de utilización del método SIL en Flexsim**



Fuente: TEBANI, K. *Hardware-in-the-loop simulation for validating plc programs*. p. 3.

La conexión del modelo previo es mediante un PLC físico, dando lugar al método de simulación HIL, sin embargo, es posible haber simulado la programación del PLC en un software, y realizar todo el entorno de manera virtual, dando lugar al método de simulación SIL.

3.2.3. Ventajas y desventajas del método de simulación SIL

Entre la función principal de este método y su utilización conlleva a la simulación efectiva del proceso, donde se verifica el comportamiento de la lógica de ejecución de órdenes, así como la verificación de soluciones y errores al diseño de la planta, máquina o proceso. Esto conlleva a centrarse al comportamiento en específico de las variables manipuladas por el controlador de manera virtual, entonces no se enfoca en el diseño en sí, sino en el funcionamiento de este.

Aun así, no verifica el funcionamiento en un controlador real en una planta simulada dado que es un controlador virtual, pero sí conlleva a la validez del

programa ejecutado en la simulación, y en la validez del diseño físico virtual del elemento a simular, dado que es un gran avance para la posible implementación en cierta empresa o industria, así como la verificación de mejora continua y eficiencia en los procesos automatizados, dando lugar también como método de enseñanza hacia los estudiantes, tratando de verificar el mejor diseño de planta y la programación de la ejecución de esta.

Tabla IV. **Cuadro comparativo de las funciones del método SIL**

	Realizado por SIL
Diseño preliminar del proceso	
Diseño preliminar de la máquina	
Diseño preliminar de la planta	
Diseño preliminar de la ejecución de órdenes	
Verificación de datos	✓
Comunicación OPC	✓
Verificación del diseño	✓
Simulación efectiva del diseño correspondiente	✓
Simulación efectiva de la ejecución de órdenes	✓
Edición de diseño	
Verificación de errores	✓
Comprobación del diseño según controlador real-simulado	

Fuente: elaboración propia.

Mediante el cuadro comparativo anterior, se suele reconocer que el método SIL se centra en el funcionamiento de los diseños estipulados anteriormente, y a una edición posterior de este. Cabe destacar que es muy importante la relación MIL-SIL, dado que un diseño preliminar debe ser verificado mediante una simulación de funcionamiento para establecer la eficiencia y determinar si es factible y eficaz.

Es importante saber que la simulación aumenta considerablemente la eficiencia y confiabilidad de ciertos procesos, porque son pruebas previas a una implementación de algún proceso o planta. Cabe destacar que hay tanto simulaciones virtuales y físicas, pero el tener una alternativa de realizarlo de manera virtual de forma específica, es recomendable para el ahorro de recursos y optimización de elementos debido a su virtualización.

3.3. Hardware-In-The-Loop

El método de simulación Hardware-In-The-Loop es la forma más efectiva de comprobar el funcionamiento total de una planta simulada mediante el controlador. Esto permite utilizar un controlador físico, con su debida lógica de ejecuciones de órdenes para ser simulado virtualmente.

La simulación de modelos físicos reales, utilizados para la prevención y control del comportamiento del sistema, es una práctica habitual en algunas industrias como por ejemplo en la industria automotriz o en la aeroespacial.

Sin embargo, las plataformas y controladores utilizados normalmente para esta finalidad tienen un costo elevado y suelen tener una portabilidad muy limitada, debido a la complejidad de la adaptación del controlador físico y un modelo simulado. No siempre será rentable realizar un sistema Hardware-in-the-loop, y será necesario un estudio de factibilidad previo.

Mediante este tipo de simulación se demuestra la eficacia del programa previo a la implementación, obteniendo así una relación directa controlador-planta simulada, debido a que el controlador necesita de una comunicación OPC externa especial para poder ser vinculado a una simulación en un software en particular, que aparte de su alta complejidad, la cantidad de software y OPC para

este funcionamiento se encuentran de manera limitada, por lo tanto solamente mediante una gran inversión es posible este tipo de prueba, aunque se puede establecer de manera muy básica en el campo de la didáctica.

3.3.1. Definición

La simulación Hardware in the Loop (HIL), consiste en un método utilizado en el desarrollo en tiempo real de sistemas, modelos y procesos complejos, de manera que se pueda comprobar la eficiencia del mismo. Se trata de una plataforma efectiva y funcional, pues incluye toda la complejidad del sistema físico controlado por el sistema de control, donde solamente la planta será virtual.

Navarro define el proceso de este método de simulación mediante un modelo matemático que recorre todos los aspectos y características que afectan a la dinámica y comportamiento del elemento de estudio con el que el sistema interactúa, donde la simulación HIL debe tener en cuenta aparte del modelo físico, la simulación de las variables, sensores y actuadores que intervienen en este, porque estos sirven como la unión entre el modelo y el sistema. Los valores de los sensores se ven afectados por el modelo de estudio, y son leídos por el sistema modelado.

Una vez simulada la respuesta del sistema controlado, un lenguaje en común de control manda las señales necesarias a los actuadores para el respectivo procedimiento, mediante una comunicación OPC externa.

La simulación HIL en tiempo real tiene una amplia aplicación, como la investigación de perturbaciones en la calidad de la energía y el desarrollo moderno de unidades de control electrónico automotriz como la principal aplicación.

La aplicación de RTS (*Real-Time-Simulator*), al sistema electrónico de potencia es un campo de investigación muy activo complementario al método de simulación HIL. En efecto, la electrónica de potencia puede considerarse un desafío significativo para RTS y HIL, considerando el requisito en términos de precisión de tiempo.

El principal inconveniente viene dado por el costo de la plataforma adoptada. Por otro lado, en la dirección de limitar el costo de la plataforma, es importante contar con un procedimiento capaz de superar las restricciones dadas por el requisito de precisión de tiempo, dado que es necesario mediante la comunicación del controlador real y la planta simulada.

La estructura de funcionamiento del método de simulación HIL es muy diferenciada respecto a los métodos de simulación MIL y SIL, que se determina de la siguiente manera:

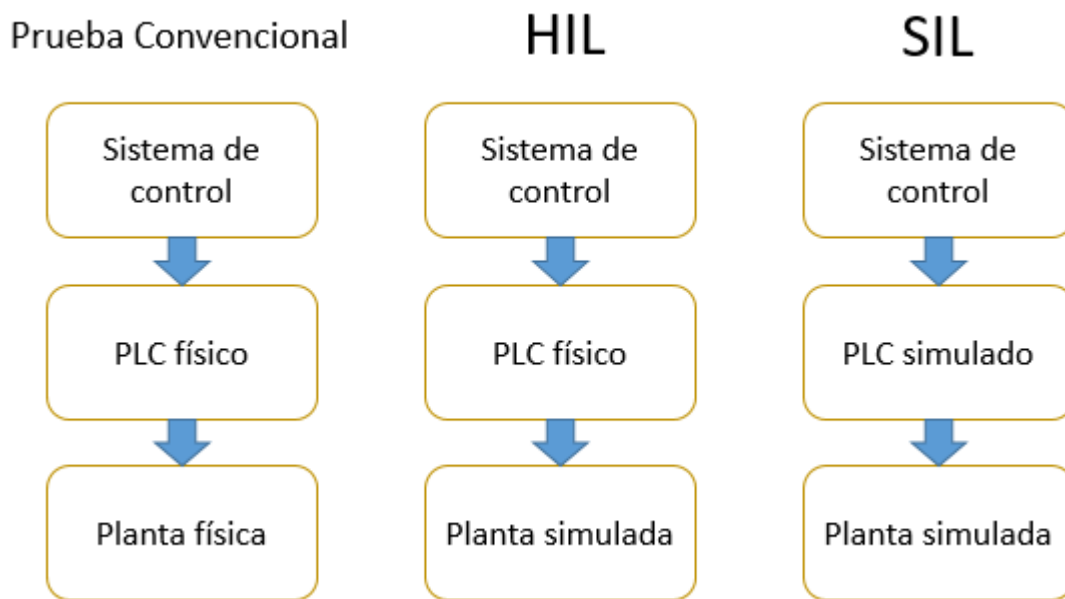
Figura 26. **Estructura HIL**



Fuente: elaboración propia.

Dada la estructura de funcionamiento del método de simulación HIL, ya es posible comparar los tipos de simulación más importantes respecto a una prueba convencional, de manera que se establezca la eficiencia y confiabilidad:

Figura 27. **Comparación de pruebas de simulación efectivas**



Fuente: elaboración propia.

De esta manera, se clarifica la diferencia de pruebas entre sí, siendo así el método de simulación SIL como la menos riesgosa costosa, y el método de simulación HIL como la opción más efectiva respecto a la proximidad del proceso.

Es importante señalar que en el caso de que sea un PLC simulado y la planta sea física, inversamente al método de simulación HIL, corresponde al método de simulación RIL (*Real-In-The-Loop*), sin embargo, no se profundizará en el tema debido a que no es muy utilizado en las industrias actualmente.

3.3.2. Funciones de la metodología HIL

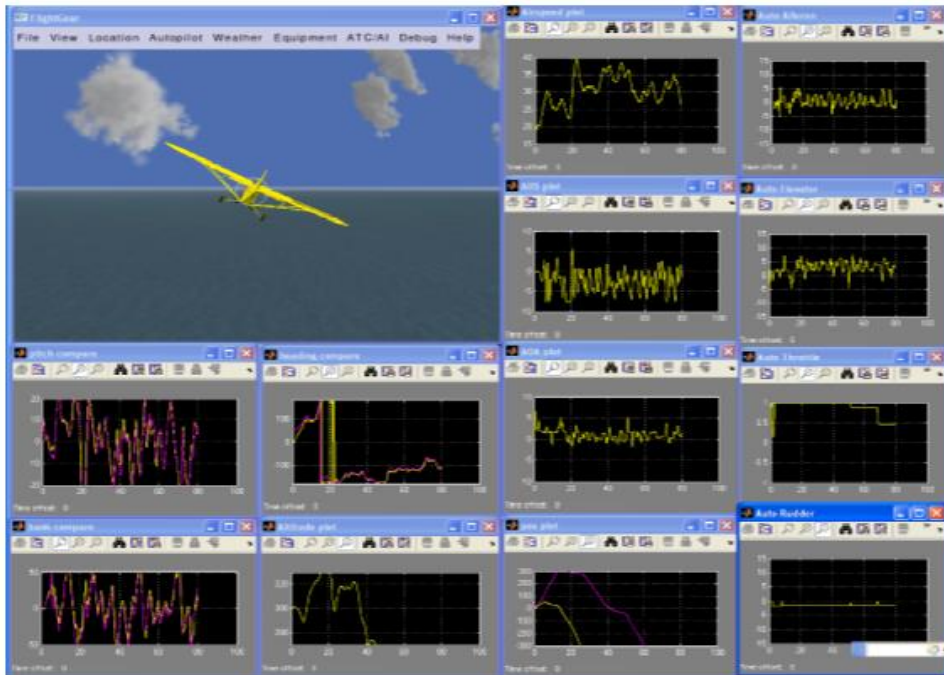
La metodología de simulación tipo HIL se refiere a la utilización de un controlador físico, donde se evaluará el diseño de la planta simulada y la lógica que conlleva el controlador, mayormente en los casos previstos es necesario de un modelo específico y de comportamiento al tomar en cuenta una simulación HIL, debido a que al usar un PLC físico, este debe tener conocimiento de que se está ejecutando en un proceso real, así que el diseño de los modelos son muy complejos.

El método HIL es mayormente utilizado en la industria automotriz, donde grandes empresas como Ford fomentan el uso de este tipo de simulaciones, y se encargan de modelar el comportamiento del vehículo o de ciertos procesos simulándolos en una tarjeta de programación avanzada.

Uno de los ejemplos más claros es el elaborado por Deepa Ramaswamy en su tesis *A Case Study in Hardware-In-the-Loop Testing: Development of an ECU for a Hybrid Electric Vehicle*, donde simula ciertos comportamientos de un vehículo híbrido, y se enfoca en la potencia del vehículo en ciertos tramos, tomando en cuenta el comportamiento y la obtención de estas medidas.

Una de las industrias que más utilizan este método es la industria de la aeronáutica, debido a la simulación de posibles clases de vuelos virtualizados, donde previamente se establece el diseño del comportamiento siendo así controlado por algún controlador físico en específico, siendo de esta manera un claro ejemplo el trabajo elaborado por Tsiotras Panagiotis en su artículo *Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle*, donde simula de manera efectiva una avioneta y su vuelo respectivo, siendo de la siguiente manera:

Figura 28. Ejemplo de simulación HIL en aeronáutica



Fuente: PANAGIOTIS, T. *Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle*. p. 11.

Mediante este ejemplo, se puede observar el diseño de la avioneta bajo un ambiente técnico, así como también al ser simulado por un controlador físicamente, se logra obtener los comportamientos de momentos, velocidad y aceleración respectiva a la simulación.

Claro está que es un diseño muy complejo para la obtención de este modelo, debido a que es necesario establecer las ecuaciones de momento y fuerza, así como de velocidad, aceleración y despegue. Todas estas ecuaciones las desarrolla Panagiotis, sin embargo, es un proyecto de demasiada inversión y muy sofisticado, por ese motivo su uso está limitado.

3.3.3. Ventajas y desventajas del método de simulación HIL

El método HIL tiene fundamento similar al método de simulación SIL, y se enfoca en la simulación efectiva de los diseños previamente propuestos, sin embargo, este tipo de simulación puede verificar modelos más complejos y completos, de manera que realmente toma en cuenta todas las características y variables disponibles en el sistema.

Esto conlleva a los comportamientos físicos y matemáticos que dicho modelo pueda elaborar, esto es necesario debido a que el PLC físico necesita reconocer el modelo de la planta como si fuera real, de lo contrario sería solamente una prueba de entradas/salidas y la verificación de señales.

El método de simulación HIL es el método más completo a comparación de los métodos SIL y MIL, esto debido a que los diseños se acercan más a la realidad y toman en cuenta el controlador real, por lo que es el paso previo a la implementación real de dicho proceso.

En relación a lo mencionado, es importante recalcar que la factibilidad y costo de una simulación HIL son de un costo notoriamente elevado respecto a los métodos SIL y MIL, debido a que los softwares necesarios para una comunicación de este tipo, un servidor OPC externo especial si es necesario y el controlador real son de mayor valor que una simulación totalmente virtualizada, porque en una simulación de tipo virtual en todo aspecto no conlleva el involucramiento de equipos físicos.

A continuación, se reconocerán las actividades realizadas por este método:

Tabla V. **Cuadro comparativo de las funciones del método HIL**

	Realizado por HIL
Diseño preliminar del proceso	
Diseño preliminar de la máquina	
Diseño preliminar de la planta	
Diseño preliminar de la ejecución de órdenes	
Verificación de datos	✓
Comunicación OPC	✓
Verificación del diseño	✓
Simulación efectiva del diseño correspondiente	✓
Simulación efectiva de la ejecución de órdenes	✓
Edición de diseño	
Verificación de errores	✓
Comprobación del diseño según controlador real-simulado	✓
Modelo matemático-físico del proceso o planta a diseñar	✓

Fuente: elaboración propia.

Respecto al anterior cuadro comparativo, es notorio que el método HIL es muy similar respecto al método SIL, a diferencia de la profundidad del modelo y a la necesidad de un controlador real.

4. ALTERNATIVAS DE SOFTWARE Y MODELAMIENTO PARA EL MÉTODO SIL

Actualmente, hay varias alternativas disponibles para la evaluación de los distintos métodos de simulación mencionados anteriormente, tanto de manera compleja como de manera introductoria, de tal forma que es posible obtener una introducción para posteriormente utilizar sistemas más complejos.

Los softwares de simulación son muy utilizados de manera didáctica e industrial, donde es posible obtener simulaciones aproximadas a la realidad, obteniendo la ventaja de ver los procesos según la lógica programada.

El software de modelación debe ser de forma explícita en la información que proveen y leen en relación a la programación. Tanto para controladores PLC, como para controladores electrónicos como FPGA especialmente, son de manera más relacionada a la industria hoy en día, donde es posible transformar dicha programación a un lenguaje en común para los simuladores, concatenando de esta manera la simulación industrial o didáctica.

El software Factory I/O es un software basado en diseño de eventos o ambientes según lo desea el usuario, teniendo la ventaja de ser 3D y de manera más conceptual, siendo más industrializado. Mediante este software, es posible realizar simulaciones de prácticas, simular fallas o adjuntar un sistema de control por medio de PID.

Es posible realizar una simulación SIL y HIL, debido a que cuenta con una conexión OPC UA y OPC DUA, siendo ambos compatibles con la programación

elaborada en Codesys, siendo programación Ladder en este caso, así como también es posible programar de manera local mediante lógica booleana o compuertas.

Figura 29. **Enlace Factory I/O y Codesys**



Fuente: elaboración propia, toma de pantalla de Factory I/O.

El software Codesys es el software más recomendado para estos casos, porque es un software capaz de enlazar conexiones OPC, para posteriormente ser concatenado a un software de diseño gráfico, donde es posible elaborar y simular en un PLC u otro controlador que el usuario disponga.

El software Codesys es capaz de transmitir los datos hacia el software sin retardo de información, y es posible cargar programas al controlador físico, de esta manera, elaborar simulaciones más avanzadas a la industria hoy en día.

El software Simumatik dispone de una manera de simulación más compleja a comparación de Factory I/O, porque se tiene en cuenta más variables al momento de simular alguna máquina en específico. Este software es capaz de emular y simular diseños de máquinas que se hayan elaborado en otro software,

exportándolos y simulándolos en dicho software. Dado que actualmente no es el software más avanzado para una simulación SIL o HIL, una ventaja es que es gratuito y posee una lógica muy parecida al software de este tema más avanzados, posee una conexión OPC compatible con Codesys, donde es posible enlazar de manera directa dicha comunicación, sin necesidad de tercerizar el proceso, siendo en este sentido parecido al software Factory I/O, siendo la mayor diferencia la complejidad de elaboración y simulación de las máquinas.

Hoy en día, muchos softwares enfocados a la educación y simulación de procesos enfocados a la automatización, son elaborados en 2D en ambientes ya preestablecidos, tomando en cuenta más que todo la lógica de la programación. Es importante recalcar el diseño del mismo, que sea eficiente para la automatización del proceso, y enlazar más variables a los procesos, siendo más parecido a la realidad.

4.1. Factory I/O

Factory I/O es elaborado por *Real Games*, enfocado a la educación en colegios, universidades y otras organizaciones en específico. *Real Games* se especializa en simulaciones de fábrica en 3D, teniendo la ventaja de elaboración de escenarios y programación.

Factory I/O está diseñado para ser simple y permitir construir una fábrica virtual, donde es posible simularla utilizando una selección de piezas industriales comunes. También incluye muchos escenarios inspirados en ambientes industriales comunes, contando con una amplia gama en cuanto a niveles de dificultad.

Figura 30. **Logo del software Factory I/O**



Fuente: elaboración propia, toma de pantalla de Factory I/O.

4.1.1. Conceptos básicos

Es necesario establecer los conceptos básicos respecto a los controles y formas de interacción respecto a este software.

4.1.1.1. Navegación

Para conseguir los objetivos planteados, es necesario conocer el funcionamiento del modelamiento mediante el software Factory I/O, así como la creación de ambientes virtuales, programación, edición de escenas, entre otros aspectos. Igualmente, cada una de las explicaciones estará apoyada por una imagen de ejemplo del software.

Tener habilidad a la hora de utilizar las cámaras en FACTORY I/O es fundamental para el uso de este software. El software empleado genera una escena en tres dimensiones, y es importante ser capaz de navegar en dicho espacio tridimensional sin problema alguno. Las cámaras son pieza fundamental para interactuar con las distintas partes presentadas en el software y serán totalmente necesarias a la hora de crear o editar cualquier escena o ambiente empleado.

Figura 31. **Ambiente estándar de navegación**

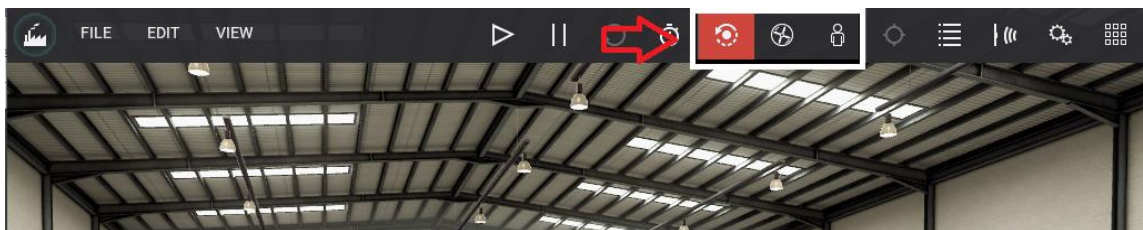


Fuente: elaboración propia, toma de pantalla de Factory I/O.

Respecto a la navegación existen tres tipos de navegación de cámara, que son las siguientes:

- *Orbit camera*
- *Fly camera*
- *First person camera*

Figura 32. **Íconos de cámara en software**



Fuente: elaboración propia, toma de pantalla de Factory I/O.

4.1.1.1.1. *Orbit camera*

Consiste en la cámara por defecto y es fundamental en el diseño y construcción de escenas y ambientes ya que es la única que permite al usuario el movimiento a través de los componentes presentes en la escena sin producir colisión o interferencia alguna con los mismos.

Su funcionamiento es simple, girando alrededor de un punto de interés que es indicado por un punto blanco. El punto de interés se establece haciendo doble clic izquierdo en cualquiera de los componentes empleados en la escena o ambiente.

Cada vez que se mueve una parte, el punto de interés se establece automáticamente en el centro de la pieza. Una vez definido el punto de interés, se podrá girar la cámara alrededor del mismo manteniendo presionado el botón derecho y arrastrando el ratón.

Es importante recalcar que los componentes que se arrastren desde la paleta se crearán a la altura definida por este punto, a excepción de las partes que normalmente se colocan en el suelo, como cintas transportadoras, pero estas pueden ser editadas en cualquier momento según conveniencia.

Cabe destacar que cada una de las cámaras tiene su control según teclas o el ratón, en el caso de *Orbit camera*, su control es el siguiente:

Tabla VI. **Control de *orbit camera***

CONTROL	ACCIÓN
<i>Backspace</i>	Reestablece la cámara a por defecto
Doble <i>click</i> izquierdo	Establece punto de interés y la cámara girará en función de este.
Rueda del ratón	Acerca o aleja la cámara
<i>Click</i> derecho + arrastre	Gira la cámara en relación del punto de interés
<i>Click</i> rueda + arrastre	La cámara se mueve de forma horizontal

Fuente: elaboración propia.

4.1.1.1.2. *Fly camera*

Mayormente esta cámara es utilizada para una movilidad libre en el espacio tridimensional.

A diferencia de la anterior, esta cámara colisiona con los distintos componentes presentes en la escena, aunque no es reconocida por los sensores empleados en la escena.

Con esta cámara se podrá inspeccionar el espacio 3D de forma segura y rápida, pudiendo alcanzar perspectivas imposibles por parte de sus alternativas.

Sus controles son de la siguiente manera:

Tabla VII. **Control de *Fly camera***

CONTROL	ACCIÓN
Tecla W	Se mueve la cámara hacia adelante
Tecla S	Se mueve la cámara hacia atrás
Tecla A	Se mueve la cámara hacia la izquierda
Tecla D	Se mueve la cámara hacia la derecha
Doble click izquierdo	La cámara gira hacia el cursor del ratón
Click derecho + arrastre	Gira la cámara
Rueda del ratón	Desplaza la cámara de forma vertical
Click izquierdo + derecho	Se mueve la cámara hacia adelante

Fuente: elaboración propia.

4.1.1.1.3. *First person camera*

Esta cámara muestra la visión de una persona de 1,80 metros de altura y suele ser utilizada al simular a una persona o trabajador en la escena o ambiente virtual, donde no puede variar de esa altura y es predeterminado.

Es la cámara utilizada para comprobar el manejo de los procesos y visualización de los mismos, porque se expresa como un empleado y su visión del mismo.

Esta cámara choca con las partes de la escena y, de forma predeterminada, no es detectada por los sensores, así que es de movimiento sin interferencia. Los controles de esta cámara son muy similares a los anteriores, que se expresan de la siguiente manera:

Tabla VIII. **Control de *First person camera***

CONTROL	ACCIÓN
Space	Salta
Tecla W	Se mueve la cámara hacia adelante
Tecla S	Se mueve la cámara hacia atrás
Tecla A	Se mueve la cámara hacia la izquierda
Tecla D	Se mueve la cámara hacia la derecha
Click izquierdo + derecho	Se mueve la cámara hacia adelante
Doble click izquierdo	La cámara gira hacia el cursor del ratón
Click derecho + arrastre	Gira la cámara

Fuente: elaboración propia.

4.1.1.2. Escenas

El software Factory I/O posee escenas predeterminadas con la opción de ser editables o escenas totalmente personalizadas. En el punto de vista didáctico, las escenas predeterminadas están distribuidas de manera que la dificultad de la lógica de los eventos va en aumento.

La escena depende primordialmente de los elementos y proceso a elaborar, por lo que es básico la estructuración de componentes de manera que sea eficiente y lógico según lo determinado.

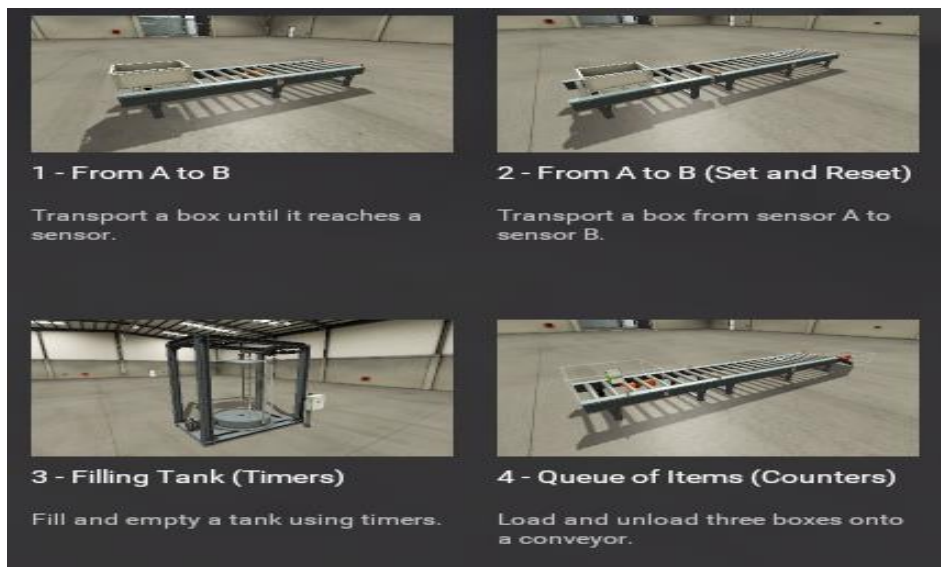
Si se desea utilizar una escena predeterminada, es sencillo seguir los pasos a llevar a cabo para la utilización de estos:

- 1. Abrir un nuevo proyecto
- 2. Clic en la opción *Welcome Window*

- 3. Clic en la opción *Scenes*
- 4. En la opción *Scenes* se muestran los diferentes tipos de escenarios a practicar o a editar según sea el caso.

Cada escena tiene su descripción del funcionamiento antes de ser seleccionada, de esta forma puede funcionar como base de una simulación personalizada más compleja al estar disponible la opción de ser editables.

Figura 33. **Ejemplos de escenas en el software Factory I/O**



Fuente: elaboración propia, toma de pantalla de Factory I/O.

Las opciones van determinadas desde un simple proceso de banda transportadora, hasta un proceso de llenado de tanque por medio de control PID o reconocimiento según tamaño de caja.

4.1.2. Análisis de plataforma Factory I/O

El análisis de la plataforma I/O consiste en evaluar sus funcionalidades generales como alternativa válida para el método de simulación prevista.

4.1.2.1. Elementos

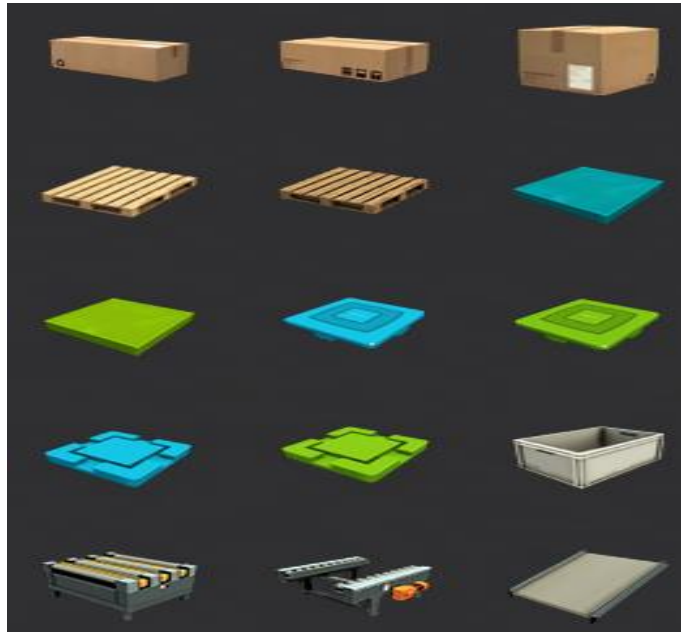
Los elementos es la parte fundamental de las escenas y del software Factory I/O, porque es la composición de la simulación del proceso a elaborar y es la demostración de la eficiencia del proceso en sí.

Al iniciar en el software Factory I/O, es posible visualizar la cantidad de componentes disponibles para poder simular, estando divididas en las siguientes categorías:

- Ítems
- Elementos de carga pesada
- Elementos de carga ligera
- Sensores
- Operadores
- Estaciones
- Dispositivos de emergencia
- Vías peatonales

Igualmente, estas categorías se muestran en la paleta de elementos, donde se podrá seleccionar cualquier elemento disponible en cualquier momento.

Figura 34. **Ejemplos de elementos en Factory I/O**



Fuente: elaboración propia, toma de pantalla de Factory I/O.

Cabe destacar que igualmente hay disponibles sensores y actuadores, por lo tanto, la simulación se fortalecerá en el ámbito de eficiencia del proceso, y las variables de dichos sensores y actuadores, denominadas Tags, también se pueden visualizar.

4.2. Codesys

Es necesario establecer un lenguaje de programación a las simulaciones, por lo que Codesys es el software que se recomienda y el que se utilizó.

4.2.1. Conceptos básicos

Codesys es un software enfocado a la programación de controladores mediante diferentes tipos de lenguajes, como por ejemplo *Ladder* o *Grafcet*, así que Codesys tiene varias aplicaciones a nivel industrial, dado que posee múltiples funciones en la elaboración de procedimientos de lógico, y una plataforma para elaborar ejemplos HMI.

Los usuarios suelen utilizar Codesys para la elaboración eficiente de aplicaciones de automatización, que conllevan a varios campos de utilización como por ejemplo controladores industriales presentes en fábricas, máquinas móviles, sistemas de automatización de edificios y de energía, así como en muchas otras áreas.

Las características que han conseguido que Codesys adquiriera la importancia que actualmente tiene son las siguientes:

- Todo está integrado en una sola interfaz de usuario, así que es posible elaborar diferentes procedimientos a la vez en su misma plataforma.
- Modelización efectiva para una implementación efectiva de proyectos complejos de automatización.
- Sistema confiable y ampliamente utilizado para aplicaciones industriales establecidas.

El software Codesys provee certeza en el tipo de lógica que conlleva, obteniendo de esta manera una facilidad viable en una comunicación efectiva de

servidores de distinto tipo, siendo compatible con distintos modelos de PLC a comunicar.

Los fabricantes de dispositivos tienen como objetivo principal la realización de un dispositivo de automatización eficiente e inteligente con un nivel de exposición de gastos limitados con una alta aceptación en el mercado, por lo tanto además de las ventajas descritas anteriormente, destaca la posibilidad de multifunción en distintos tipos y marcas, así como globalización de lenguajes.

En lo que conlleva a la comunicación OPC, siendo otra de sus ventajas, destaca la posibilidad de una comunicación interna evaluada por el mismo software Codesys, compartiendo de la misma manera la opción de compartir dicha información en una comunicación de red industrial. Es adaptable a una comunicación OPC externa con cualquier software.

Se describe que la utilización de Factory I/O y Simumatik con una comunicación OPC utilizando Codesys es una simplificación de comunicación externa, pudiendo ser evaluado con software de diseño más avanzados, como por ejemplo Xcelgo Xperior, en donde la comunicación según la red industrial puede aumentar en eficiencia de información, así que se enfoca meramente en una base de funcionamiento más complejo.

Del punto de vista didáctico, siendo el enfoque prioritario, es suficiente fomentar como base la utilización de Codesys con otros tipos de software y evaluar posibilidades de utilización, siendo muy avanzadas y pocas utilizadas a nivel escolar o universitario.

Figura 35. **Logo del software Codesys**



Fuente: elaboración propia, toma de pantalla de Codesys.

4.2.2. Análisis de plataforma Codesys

La plataforma de Codeys posee varias ventajas, entre las que se encuentra la concatenación directa de datos y servidor OPC, que simplifica bastante el proceso de comunicación.

4.2.2.1. Programación

La versión del software a utilizar es denominada 3,5, siendo de las más actuales versiones. La programación en Codesys se puede evaluar de diferentes maneras, siendo la de tipo Ladder la propuesta de interés. La importancia del software en la utilización de la programación es debido a su alto rendimiento de procesamiento y facilidad de comunicación OPC con otro software, concatenándolo a un lenguaje en común.

La programación en el software Codesys se define en distintas categorías siendo las siguientes:

- General
- Operadores Lógicos
- Operadores matemáticos
- Otros operadores
- Módulos de función
- Elementos de diagramas de contadores

El software se puede visualizar en diferentes tipos de utilización según las descritas anteriormente, siendo la categoría de elementos de diagramas de contactores y módulos de función de interés, debido a que la investigación se enfoca en la simulación SIL en el laboratorio de automatización industrial, enfocado solamente a PLC.

En referente a las variables, se van definiendo de manera global dentro del programa a medida de uso, de forma que se puedan compartir mediante la conexión OPC, ejemplificando de la siguiente manera:

Figura 36. **Ejemplo de definición de variables en Codesys**

```
VAR_GLOBAL  
  Button: BOOL;  
  SensorClose: BOOL;  
  SensorOpen: BOOL;  
  Light: BOOL;  
  MotorFwd: BOOL;  
  MotorRev: BOOL;  
END_VAR
```

Fuente: elaboración propia, empleando Codesys 2020.

Las variables igualmente deben de tener referencia en el lenguaje Ladder a programar, de tal forma que las variables sean referentes.

4.2.2.2. Visualización HMI

Otra de las características importantes del software Codesys, es que desde la perspectiva 2D, comúnmente en visualización de procesos, es posible la elaboración de simulación HMI, involucrando la programación con acciones representadas por objetos 2D en el mismo software.

Esto es muy útil para la comprobación de procesos simples, o modelamiento de estructura 2D de intervención máquina-hombre, así que, puede ser útil de comprobación de procesos antes de elaborarlo en 3D, siendo este último más complejo.

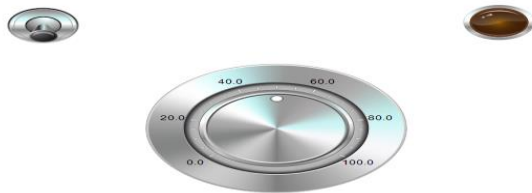
Dada la estructura de un programa en PLC, en el ícono de aplicación, se debe de agregar un objeto e ir a la opción de agregar visualización, de esta manera se crea una nueva ventana que se complementa de una barra de herramientas acoplada, obteniendo las siguientes herramientas:

- Base
- Elementos de control generales
- Gestor de alarma
- Instrumentación de medición
- Lámparas o Switch
- Elementos especiales de control
- Elementos de control fecha y hora

Al elaborar la programación en *Ladder* y definidas las variables, es posible vincularlas a objetos 2D disponibles en Codesys, donde en cuya descripción se modifican los parámetros a interés dependiendo del elemento seleccionado.

Al poder seleccionar varios componentes 2D de la barra de herramientas, se pueden crear escenarios 2D de medición, visualización o de funcionamiento según la necesidad requerida por el usuario, donde la visualización de los componentes 2D es de la siguiente manera:

Figura 37. **Ejemplo de visualización HMI en Codesys**



Fuente: elaboración propia, toma de pantalla de Codesys.

La imagen anterior está compuesta por un interruptor, una lámpara y un potenciómetro, obteniendo así un ejemplo de medición, ejecución y visualización básica de componentes.

4.3. Simumatik3D

Simumatik3D es una versión anterior a Simumatik App, que se enfoca en la base de estructura de modelamientos y conexiones directas al servidor OPC.

4.3.1. Conceptos básicos

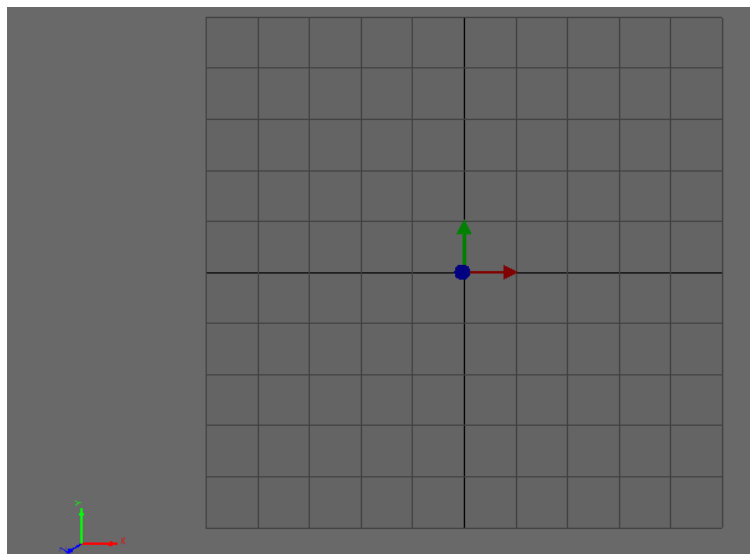
En referencia a los conceptos necesarios para establecer una interactividad de diseño y programación, se necesita obtener un conocimiento básico del diseño y estructura de los componentes y ambientes.

4.3.1.1. Navegación 2D

Simumatik tiene la opción de trabajar la simulación en dimensión 2D, de manera que sea útil de forma HMI, o comprobación de procesos simples según la manipulación de acciones.

Este tipo de navegación es comúnmente utilizada para la simulación de funcionamiento de sensores o actuadores, de manera que se efectúe el funcionamiento y se compruebe la validez del mismo.

Figura 38. Interfaz 2D en Simumatik3D

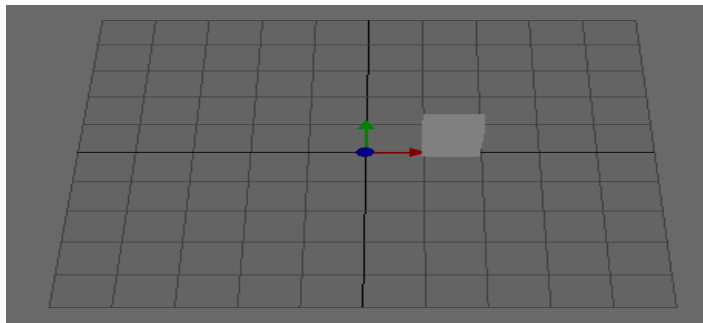


Fuente: elaboración propia, toma de pantalla de Simumatik3D.

El plano se puede mover a conveniencia y efectuar cambios, de manera que los componentes son libres de posicionamiento dentro de la interfaz gráfica, siendo una interfaz plana sin la opción de giro.

En el momento de seleccionar un tipo de motor o máquina elaborada en 3D, y la visualización mostrarla en 2D, se visualizará dicho componente como plano, por lo que un error común es no establecer una interfaz gráfica 3D.

Figura 39. **Visualización de componente en 2D**



Fuente: elaboración propia, toma de pantalla de Simumatik3D.

En la visualización anterior, se observa que hay flechas de movimiento y sus posibles sentidos, por lo tanto, en 2D los sentidos son como movimiento plano. El objeto es un gabinete de control y se observa en 2D como un rectángulo sin orientación, así que es recomendable utilizar la interfaz 2D como simulación de procesos simples de funcionamiento y visualización de funciones de sensores o HMI.

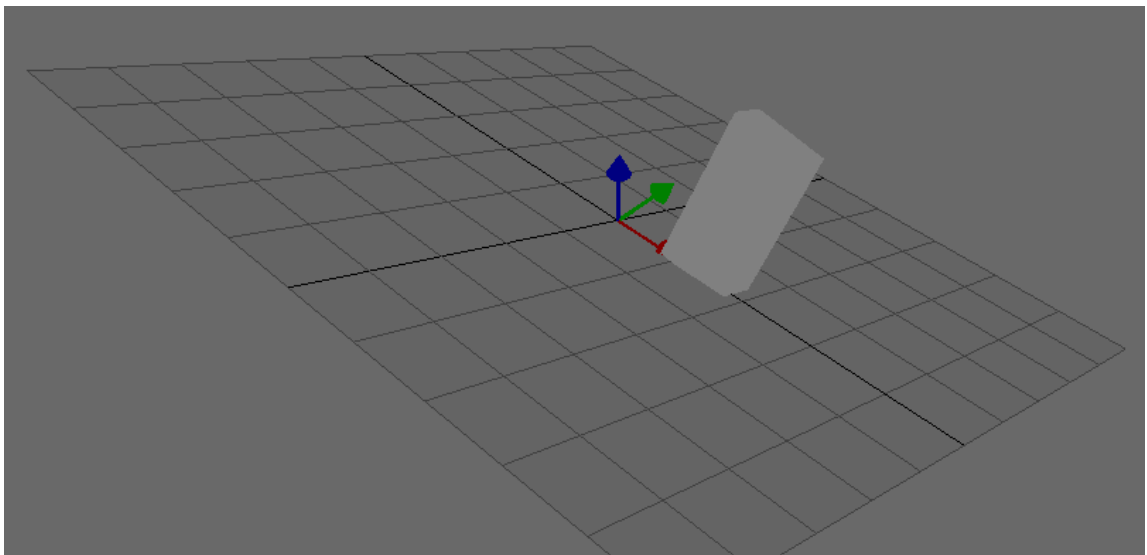
4.3.1.2. Navegación 3D

La función principal del software Simumatik es la simulación, diseño y comprobación de procesos automatizados en 3D. Su principal ventaja en una simulación 3D es la composición de funcionamiento de las distintas máquinas disponibles y de la posibilidad de importar modelos creados por software de diseño hacia su plataforma. Actualmente, la navegación 3D de Simumatik tiene

una cerca relación con el software Robotstudio, donde es posible importar el modelo, procesarlo y simularlo dentro de la interfaz gráfica.

Dado que la navegación no se toma en referencia a un plano, lo que conlleva a ejes X, Y y Z. Los distintos objetos disponibles se pueden movilizar a lo largo de la interfaz gráfica mediante estos ejes, siendo de esta manera el diseño de orientación y tamaño de objetos una modificación importante donde los objetos no son predeterminados y se tiene profundidad en su diseño.

Figura 40. **Visualización 3D en Simumatik3D**



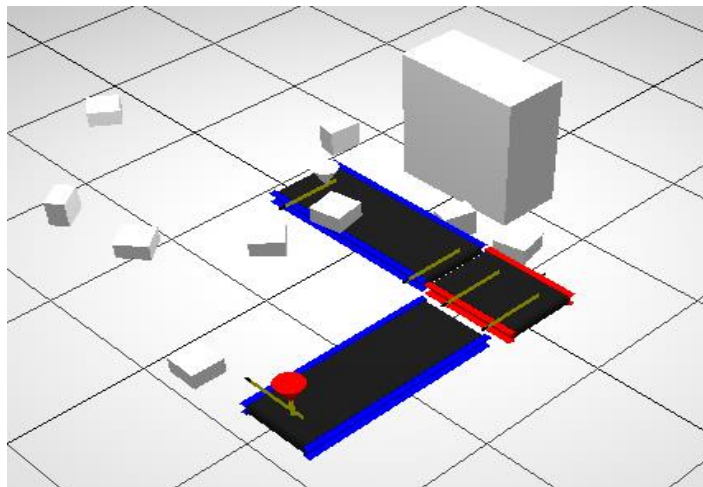
Fuente: elaboración propia, toma de pantalla de Simumatik3D.

En la figura anterior se observa que las flechas de dirección de movimiento se basan en los ejes de movimiento disponibles, así que los objetos son libres de modificación según a conveniencia y se pueden obtener tamaños a base de cierto modelo.

Las diferentes visualizaciones en 3D conllevan igualmente a la rotación de los objetos, donde la creación de los escenarios es de forma libre teniendo la opción habilitada de la importación de modelos en 3D compatibles con el software Simumatik.

En los diferentes tipos de creación de modelos, siendo RobotStudio la más utilizada, se conlleva a una simulación de mayor complejidad, donde se deben de establecer los modelos físicos y matemáticos de los mismos, así como las variables a manipular para dicho diseño y establecerlas, para que se pueda obtener un control de movimiento.

Figura 41. **Ejemplo de proceso 3D**



Fuente: elaboración propia, toma de pantalla de Simumatik3D.

4.3.1.3. Funcionabilidad

La función principal del software Simumatik es de simular procesos industriales automatizados, siendo su función tanto industrial como didáctico, por

lo tanto, da lugar a poder aprender su funcionamiento como base para software más complejos en el uso de la simulación SIL.

La función de Simumatik también se puede emplear como HIL, pero la conexión OPC es más compleja, además de la declaración de componentes globales que debe de ser más descriptivo y específico, por lo que se enfoca solamente en el método de simulación SIL.

La barra de herramientas principal del software cuenta con las siguientes opciones de visualización:

- *Window always on top*
- *View or Hide logging Panel*
- *Debug*
- *Object Library*
- *Select object*
- *Move object*
- *Rotate Object*
- *Move to Origin*
- *2D visualization*
- *3D visualization*
- Opciones de guardado
- Opciones de simulación

Figura 42. **Barra de herramientas Simumatik3D**



Fuente: elaboración propia, toma de pantalla de Simumatik3D.

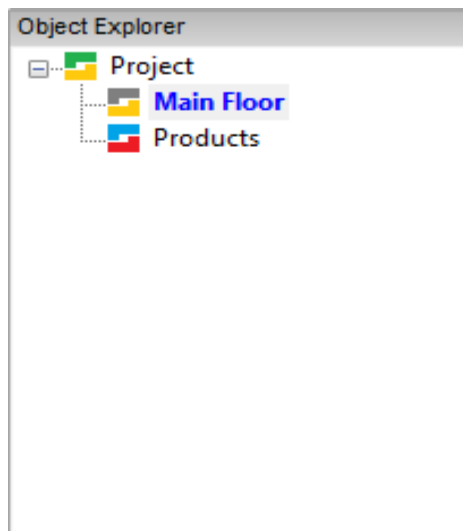
Funciones de la barra de herramientas:

- *Window always on top* se refiere al punto de referencia del objeto a enfocarse, siendo una opción de visualización.
- *View or Hide Logging Panel* se refiere a mostrar o esconder el panel de errores o de muestra de simulación correcta.
- La opción *Debug* es la descomposición de todos los componentes que conforman un proceso.
- *Object Library* es la opción para agregar componentes según la opción que se esté utilizando, entonces los objetos (componentes utilizados) es en función del control propuesto.
- *Select, Move y Rotate* se refiere a la manipulación de los objetos en relación al tamaño, posición, entre otros.
- *Move to origin* es la opción que se utiliza para ir al centro del proceso general en la interfaz.

Aparte de la barra de herramientas de Simumatik, se observa la composición de escenarios, donde se tendrá el listado de señales, componentes

y controles expuestos en el proceso automatizado, así como su descripción y su función a realizar.

Figura 43. **Menú de objetos utilizados y su conexión**



Fuente: elaboración propia, toma de pantalla de Simumatik3D.

Simumatik necesita la composición de la máquina para poder controlar, dado que se debe de enfocar en las variables a manipular de manera específica, así como también necesita de la descripción del proceso de manera escalonada como se verá más adelante, donde el menú de objetos expuestos señala la sistematización del proceso automatizado.

4.3.2. Análisis de plataforma

Simumatik3D es una plataforma interactiva que refleja el modelo, diseño y práctica de programación PLC, siendo diferente a otros softwares hoy en día.

4.3.2.1. Escenarios

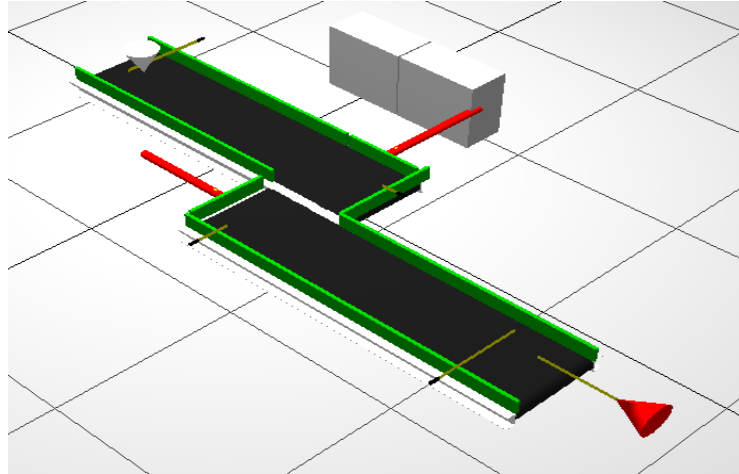
Como se expuso anteriormente, los escenarios no serán solamente físicos y controlables, sino que además serán descritos mediante la composición de objetos, por lo que se verá el funcionamiento y relación de las máquinas y botones hacia el PLC, su control y sus variables. Igualmente, la composición se enfoca meramente como descripción, donde mostrará en su menú si algún sensor o máquina no está en funcionamiento.

Los escenarios se componen del suelo, objetos y diseños importados de otro software. Los escenarios suelen tener el mismo fondo de trabajo, y solamente será diferenciado por los objetos o componentes a utilizar.

Es posible el modelamiento de una planta similar a la realidad, o el enfoque a un proceso en específico, así como también la posibilidad de la utilización de HMI'S y de escenarios 2D enfocados meramente en comprobación de botones de emergencia, luces, entre otros.

Los escenarios serán compuestos por distintas acciones disponibles en Simumatik, como dispensadora de cajas por ejemplo, en donde se simboliza mediante objeto flotante que indica la posición de la acción a realizar, esto enfocado meramente a procesos industriales y no a un objeto en específico:

Figura 44. **Ejemplo de escenario**



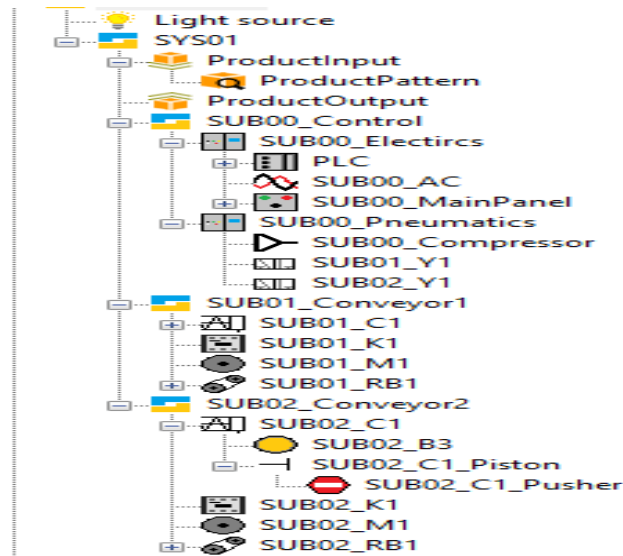
Fuente: elaboración propia, toma de pantalla de Simumatik3D.

En la imagen anterior se puede observar un objeto blanco flotante, que es la acción de dispensadora de cajas. En resumen, dicho escenario está compuesto por una acción de dispensadora de cajas, gabinete de control, 2 bandas transportadoras y 6 sensores que describen el movimiento de la caja.

Los sensores detectan la posición de la caja, y al terminar el curso de la primera banda transportadora, se gira y se activa la banda transportadora número 2 para continuar con el curso de transporte.

El menú de la composición y descripción del proceso de funciones muestra cada uno de los objetos, su conexión y la sistematización de los componentes dentro del proceso, describiendo de manera específica el proceso automatizado, que se mostraría de la siguiente manera utilizando el mismo proceso mostrado anteriormente:

Figura 45. **Ejemplo de descripción de la composición del proceso**



Fuente: elaboración propia, toma de pantalla de Simumatik3D.

La descripción del proceso es de manera sistematizada como lo mostrado anteriormente, de esta forma la simulación SIL es meramente descriptiva en relación al proceso automatizado, y así su importancia como base de funcionamiento para software más avanzados respecto a gráficos y descripción de procesos.

4.3.2.2. Componentes

Los componentes es la estructuración física de los procesos y escenarios a establecer, cuya misión será ejercer la lógica de programación y su debido comportamiento respecto al diseño respectivo evaluado.

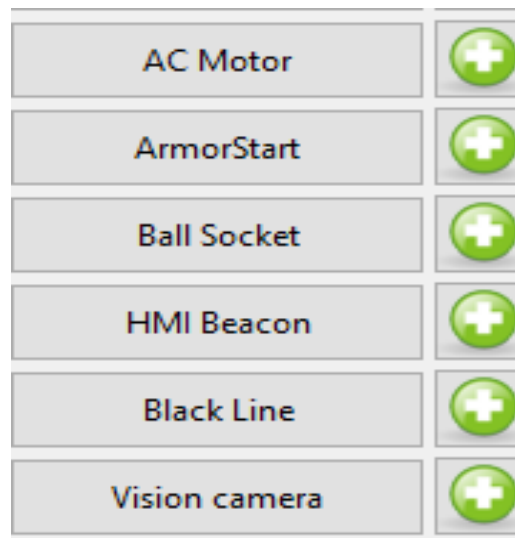
Siendo una parte fundamental de una simulación, estableciendo la conexión física con la comunicación del PLC, se demuestra el proceso automatizado de

manera que, mediante el diseño establecido, se pueda determinar la eficiencia tanto de la programación como el diseño físico de la simulación y del proceso.

Los componentes disponibles para utilizar serán según el objeto a afectar, dado un grupo creado en la interfaz gráfica, hay componentes de control y de sistema. Las opciones de motores, bandas transportadoras, sensores, actuadores, así como señalizaciones, uniones y acciones. Esto es respecto a lo utilizado, y serán habilitadas las acciones disponibles según se vaya estableciendo el sistema de automatización.

Por ejemplo, en el momento de crear un grupo, se establecerán los objetos disponibles a la categoría establecida. Dado un motor, se establecerán los objetos disponibles para dicho motor, así como uniones y control de este.

Figura 46. **Ejemplo de objetos disponibles Simumatik3D**



Fuente: elaboración propia, toma de pantalla de Simumatik3D.

4.3.2.3. Descripción de componentes

La descripción de los componentes es una parte del objeto en sí, que conlleva a las características del objeto en específico, así como su orientación, rotación, color y variable a manipular o controlar. Al ser creado un objeto, necesita de una caracterización que pueda ser manipulada y descrita, por lo tanto, en esta parte es posible la referencia de variables en comunicación de Codesys y el software Simumatik.

La vista de esta caracterización se muestra en el panel de parámetros, y dependiendo del objeto en cuestión, se mostrarán los parámetros descriptivos y físicos a manipular, por lo que es de vital importancia dado que se debe de concatenar las variables descritas en el proceso, y vincularlas a su respectiva descripción para poder obtener un comportamiento o manipulación.

Por ejemplo, en la elaboración de una banda transportadora, se necesita un motor AC para su debido proceso, y al insertar el objeto de motor AC, su panel de parámetros se describe de la siguiente manera:

Figura 47. **Parámetros de un motor AC en Simumatik3D**

Parameter	Value
Class	AC Motor
Name	acmotor
State	Stop
Nominal Voltage	380.0
Nominal Frequency	50.0
Poles	4
Input	None
Speed	0.0

Fuente: elaboración propia, toma de pantalla de Simumatik3D.

De esta manera, la caracterización del motor AC se lleva a cabo mediante el número de polos, clase de motor, estado inicial, voltaje nominal, frecuencia nominal, señal de entrada y su velocidad. En la elaboración de algún componente, en la opción *SPEED* (velocidad) se debe de tener como referencia a qué componente estará vinculado, como una puerta o banda transportadora, por ejemplo.

Por lo tanto, dicha descripción está disponible para todos los objetos disponibles dentro del software, y de esta manera es posible la vinculación de señales y contactos dentro del software, porque es necesario una estructura de vinculación específica para poder obtener mejor eficiencia en la elaboración de procesos industriales automatizados.

Para una interfaz 2D, igualmente está disponible esta descripción, así que se vinculan los botones respecto a señales generadas u obtenidas, teniendo de esta manera una sistematización menos compleja de la forma de trabajo del software y su caracterización.

Figura 48. **Parámetros de un sensor de proximidad en Simumatik3D**

Parameter Panel	
Parameter	Value
Class	Sensor
Name	sensor
State	Not detecting
Type	Proximity
Logic	Normally Off
Length	0.5
Filter time	0.0
Hide range	False
Signal	0

Fuente: elaboración propia, toma de pantalla de Simumatik3D.

4.3.3. Vinculación Codesys-Simumatik

En general, las versiones de Simumatik, tanto la versión 3D como la versión APP, permiten la conexión mediante Codesys, donde se deben de establecer ciertos protocolos de comunicación que permitan la partición y vinculación de los datos y variables a establecer. Esto se puede realizar ante cualquier lenguaje de programación, como por ejemplo el lenguaje *Ladder* o Secuencial.

La vinculación mediante el servidor OPC, permite que los datos y variables globales establecidas en Codesys, se detecten dentro del mismo ordenador, siendo llamadas y catalogadas por el servidor OPC y ordenándolas según el software que lo esté solicitando.

La realización de la vinculación OPC se puede elaborar de distintas maneras entre distintos softwares, y lo importante sería simplificar el proceso y cantidad de conexiones internas en el ordenador. Tanto los softwares Codesys, Simumatik en sus 2 versiones y Factory I/O poseen la opción de establecer una comunicación OPC mediante un PLC virtual vinculado al ordenador.

Lo importante del proceso es obtener un programa que registre los datos y los convierta en un lenguaje común para la partición y etiqueta de datos, por lo tanto el software Codesys es primordial debido a que tiene esa capacidad de vinculación. Codesys establece un PLC virtual dentro del ordenador, donde ante cualquier software con estas opciones pueda detectar las variables compartidas y vincularlas a sí mismas, estableciendo así en Codesys las variables globales y tipo de etiqueta.

5. APLICACIÓN DEL MÉTODO DE SIMULACIÓN SIL

Como se ha definido previamente, el método SIL se establece antes de poder realizar una simulación HIL. Como concepto, una simulación SIL consiste en la simulación totalmente virtual, tanto el modelo físico como el funcionamiento del PLC y su lógica programada. De la misma manera, es importante la forma de conexión de los componentes y la estructuración de los objetos para poder realizar el proceso de la forma más eficiente posible.

Ya obtenida la simulación totalmente funcional en todos los aspectos, se puede establecer una simulación HIL que consiste en utilizar el modelo físico virtual, pero se establece la programación a un PLC físico, donde se comunica al software para establecer el funcionamiento, para ya posteriormente establecer ambos modelos de manera física, lo que permite disminuir el error y aumentar la eficiencia del proceso automatizado.

El software a utilizar será Simumatik APP, que consiste de una versión web que no necesita de instalación, así que el almacenamiento de datos se establece en la nube del servidor que se utiliza en los proyectos a realizar. Esta versión alternativa de Simumatik tiene una versión demo que está limitada, o de una suscripción dependiendo del tipo de uso a realizar, tanto de manera didáctica como profesional.

Simumatik APP es la alternativa más reciente para elaborar proyectos, y su forma de trabajo consiste de la misma manera que Simumatik3D. La diferencia más significativa es que Simumatik APP tiene una forma más familiar en el enfoque didáctico, y se pueden establecer ejemplos que no necesiten de un PLC,

que sean meramente ejercicios de creación de escenarios y cableado de los componentes.

5.1. Simumatik APP

Simumatik APP es un software que permite modelar y simular distintos tipos de escenas, así como la creación libre de estos y permite la simulación del proceso mediante un PLC virtual o solamente mediante cableado de componentes. Simumatik APP es una alternativa a Simumatik3D, con la ventaja de ser más actualizado y sin necesidad de su instalación.

Simumatik APP tiene la capacidad de trabajar bajo servidores establecidos en nubes de internet, donde se almacenan los datos y se trabajan los proyectos, y mediante protocolos es posible la vinculación de un servidor OPC al servidor del proyecto donde se desea aplicar.

La forma de trabajo de Simumatik APP es similar a Simumatik3D, siendo superior en la capacidad de aprendizaje y metodología de conexión y aplicación. Simumatik APP es vinculable a otros softwares con la capacidad de exportar modelos más específicos, como por ejemplo el software RobotStudio, donde se tiene aplicaciones enfocadas a la automatización mediante robots.

La modulación y aplicación en este software es muy simplificada de manera que resulte didáctico, por lo tanto su entorno es eficiente respecto a la forma de conexión y evaluación de los proyectos a realizar, dado que se enfoca además en una forma de automatización tanto por cableado como por programación de PLC.

5.2. Visualización del método SIL en Simumatik APP

El método de simulación SIL en Simumatik en general, provee las posibilidades de una simulación eficiente de una idea o proyecto a automatizar mediante los distintos componentes que provee, como sensores u objetos físicos, que se pueden adecuar a conveniencia y diseño estipulado.

La comunicación para la evaluación de este método incluye un protocolo de comunicación entre Codesys y Simumatik APP por medio de un servidor OPC, en el que permite establecer ejercicios o escenarios que necesiten de un PLC virtual y las tareas como tal.

Mediante el modelo planteado o ejemplificado, la simulación se puede comprobar mediante una característica secuencial o Ladder, según sea el lenguaje del PLC, para que, de esta manera, Codesys pueda establecer y convertir dicho lenguaje a un lenguaje común, mediante el servidor OPC establecido en el ordenador.

El servidor OPC entonces identifica al software Simumatik como cliente DA (Data Access), por el que se declaran y comparten las variables y programación establecidas en Codesys. Es importante recalcar que esto será necesario si se necesita del uso de un PLC dentro del escenario virtual, porque recientemente se han implementado alternativas de sistemas donde los procesos que no necesiten de un PLC, se pueden simular simplemente mediante cableado.

Figura 49. **Visualización de servidor en la nube**



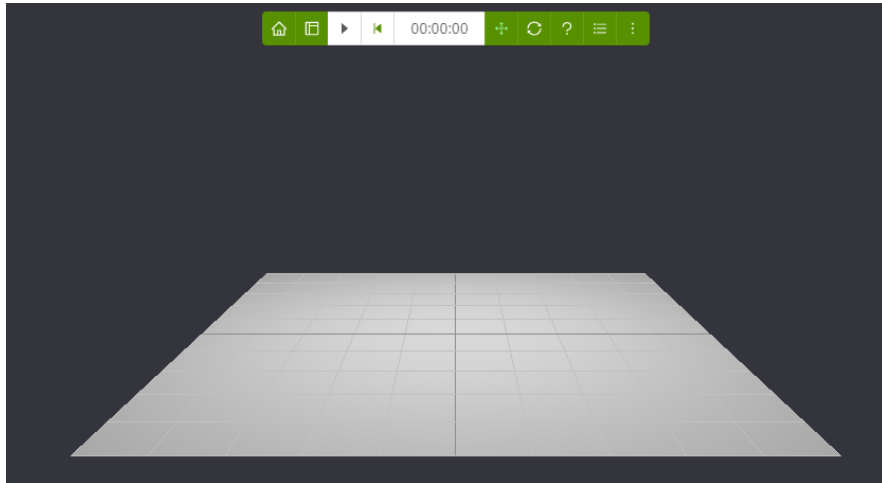
Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

El servidor se inicia de forma manual, en donde se establece un lugar dentro de los servidores de Simumatik APP, obteniendo así mayor espacio de almacenamiento y procesamiento de memoria y datos en general.

Así mismo, se pueden crear varios servidores mediante una dirección IP estipulada por usuario, mejorando la capacidad de comunicación entre usuarios y el compartimiento de datos entre ellos, pudiendo así unirse a un escenario proyectado por usuario, trabajando de manera simultánea sin problemas de procesamiento de servidor o interferencia de señales OPC.

De la misma manera, se requiere del deshabilitamiento de cookies de terceras partes dentro de la página web, esto debido a interferencia de servidores externos que conllevan una comunicación entre Simumatik APP y el ordenador, no pudiendo validar la información.

Figura 50. Escenario general en Simumatik APP



Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

A los íconos que se observan en la barra de menú, corresponden las siguientes opciones:

- Home
- Show Panels
- Play
- Reset
- Move
- Rotate
- Learn
- Logs
- More

La opción *Home* se refiere a regresar al menú donde se encuentran los servidores, y poder establecer nuevas comunicaciones. La opción *Show Panels*

establece los escenarios de prueba que se tienen, así como también establecer distintos componentes, declarar variables y construir escenarios. La opción *Play* simula el escenario. La opción *Reset* vuelve a establecer a un inicio la simulación correspondiente.

Las opciones *Move* y *Rotate* se refieren a la manipulación en cuestiones de movimiento de los sensores, motores o cualquier objeto elaborado. La opción *Learn* se refiere a consultar el manual de manejo. *Logs* es la opción que establece la comunicación externa o compartición de escenarios entre usuarios, la opción *More* describe la información del servidor y el establecimiento de la comunicación OPC.

Figura 51. **Ejemplo de descripción en servidor**

Server	Scene	Gateway	Logs
Server Version			0.13.2
Domain Name	11a92441-9bb5-11ea-be6b-06fe7d6a1d26.oep.simumatik.com		
Remaining Time			01:40:43
Connection Status		● Connected	<input checked="" type="checkbox"/>
Participants		1	
DebugDrawPhysics		OFF	<input type="checkbox"/>
DebugDrawPhysics Interval	milliseconds	<input type="text" value="1000"/>	
Monitoring			

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

5.3. Conexión OPC Codesys-Simumatik APP

La conexión OPC es necesaria en los softwares que no incluyen una plataforma de programación y comunicación adjunta, esta alternativa es muy efectiva debido a la conceptualización de un PLC a un software de diseño y emulación de ciertos escenarios simulados.

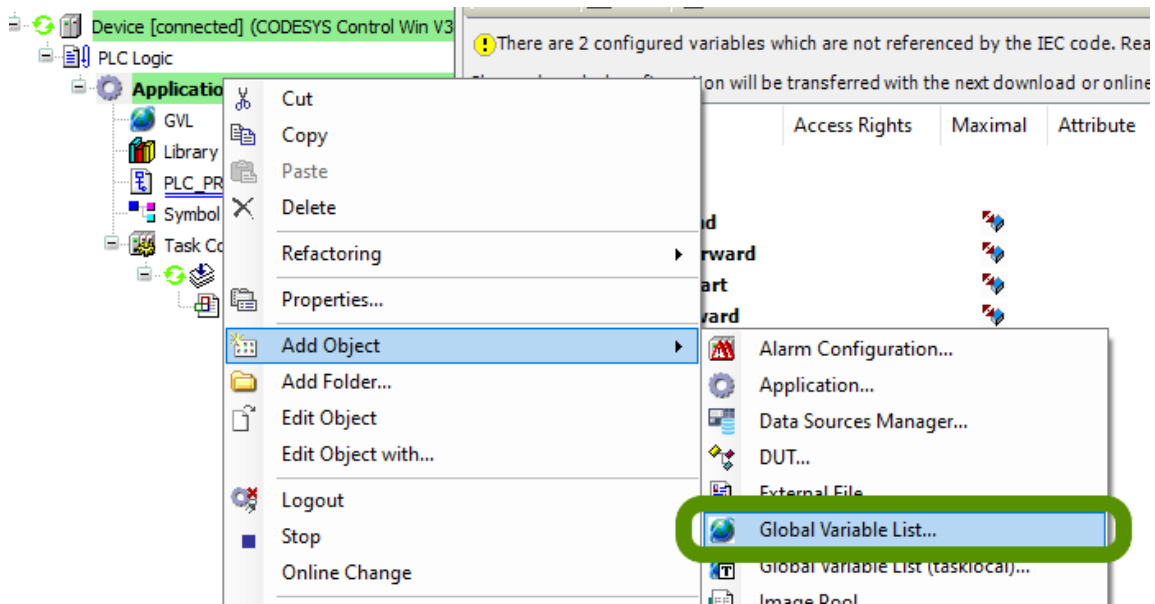
El protocolo de comunicación OPC se encarga de que la comunicación entre un software de programación y un software de diseño tengan un lenguaje en común, para que de esta forma se puedan transferir los datos y procesos estipulados sin necesidad de una conversión de datos manual.

Para una simulación SIL, el concepto de simulación totalmente virtual es primordial debido al ahorro de recursos y efectividad de la conexión y modelo de la lógica a diseñar y programar cierto escenario. Una conexión OPC, simulando un PLC virtual, enfoca todos los objetos y variables de manera que su transferencia de función sea elegible.

5.3.1. Proceso de conexión OPC

Para establecer la comunicación entre Codesys y Simumatik APP se requieren de la declaración de variables y establecer el protocolo OPC para el establecimiento de la relación entre el ordenador y el PLC virtual, por lo que las variables deben de declararse globales.

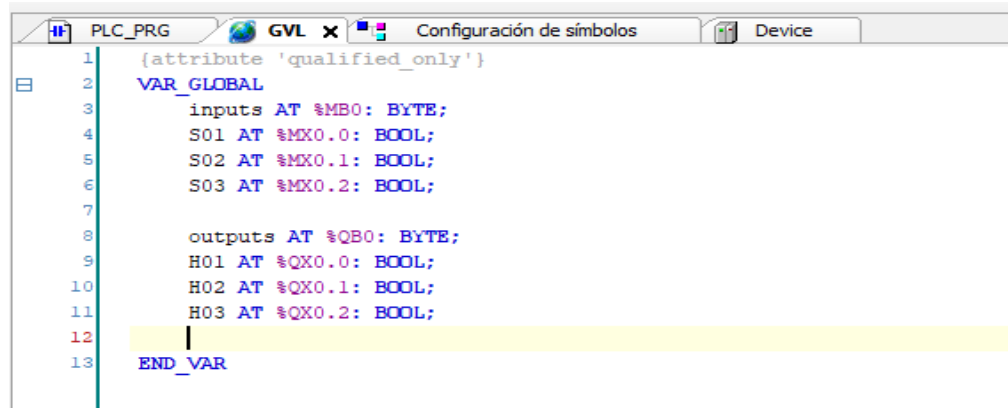
Figura 52. Creación de variables globales



Fuente: elaboración propia, empleando toma de pantalla en Codesys.

En Codesys, se debe especificar en la opción *Application*, que consiste en el establecimiento de toda la lógica correspondiente al programa a elaborar. Dentro de dichas opciones se debe de agregar un objeto, que en este caso es una creación de variables globales.

Figura 53. Ejemplo declaración de variables globales



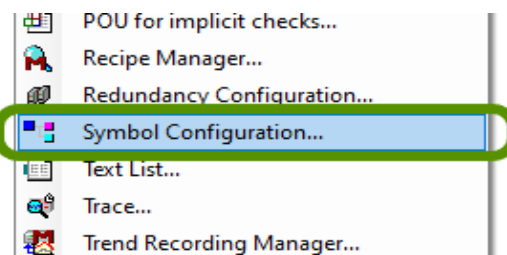
```
1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3   inputs AT %MB0: BYTE;
4   S01 AT %MX0.0: BOOL;
5   S02 AT %MX0.1: BOOL;
6   S03 AT %MX0.2: BOOL;
7
8   outputs AT %QB0: BYTE;
9   H01 AT %QX0.0: BOOL;
10  H02 AT %QX0.1: BOOL;
11  H03 AT %QX0.2: BOOL;
12
13 END_VAR
```

Fuente: elaboración propia, empleando Codesys 2020.

Cabe destacar que la forma de la declaración de las variables globales debe de ser de tal forma que las variables output deben inicializarse con la memoria Q y las variables globales inputs con la memoria M. Un dato importante es que, en las propiedades de dichas variables globales, dentro de la opción *Build*, se debe habilitar la opción de “vincular siempre”.

Luego de comprobar dicha característica, se debe crear una configuración de símbolos en *Application*, agregando así otro objeto. De la misma manera, se debe habilitar la opción *Support OPC UA Features*.

Figura 54. Opción de configuración de símbolos



Fuente: elaboración propia, empleando toma de pantalla en Codesys.

En la configuración de símbolos, basta con que se seleccione el grupo de variables globales tanto output como input generales.

Figura 55. **Selección de variables a comunicar**

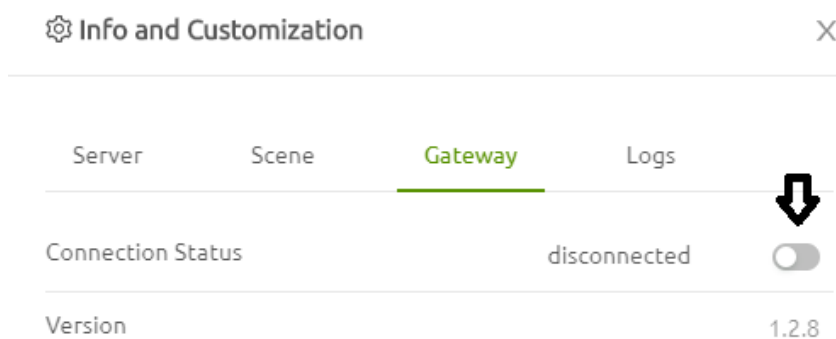


Fuente: elaboración propia, empleando toma de pantalla en Codesys.

En lo que corresponde a Simumatik APP, para permitir la comunicación OPC, se necesita el uso de un *Gateway*, que está ya implementado en el software. Se debe descargar e instalar al ordenador, para que de esta manera reconozca las variables establecidas.

Obtenido el *Gateway*, se debe de activar en el escenario de la siguiente manera:

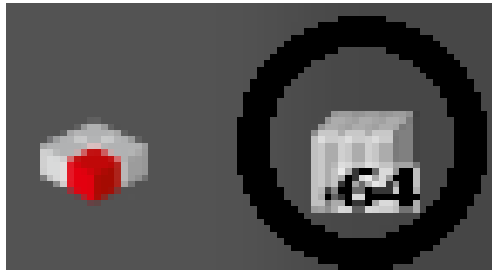
Figura 56. **Activación del Gateway**



Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Establecido el *Gateway*, se procede a comunicar el software Codesys y Simumatik APP, habilitando el PLC virtual de Codesys a modo *RUN*, para posteriormente ir en línea y correr el programa de PLC:

Figura 57. **PLC virtual de Codesys**



Fuente: elaboración propia, empleando toma de pantalla en Desktop personal.

Estando el PLC virtual en modo *RUN*, se debe de establecer el programa en línea y correr el programa, todo esto se elabora en Codesys.

Cada componente está establecido y preparado para la comunicación, además estando conectado el *Gateway* de Simumatik APP, se procede a correr la simulación y a comprobar el funcionamiento del programa y de la lógica programada en Codesys.

5.4. Descripción de procesos

El procesamiento y la automatización de los procesos debe ser lo más eficiente posible, de manera que se aprovechen todos los recursos y que el proceso cumpla con lo estipulado. Siguiendo ese modelo, el proceso dentro de Simumatik APP debe de establecer que el reconocimiento y comportamiento

físico de los objetos o complementos a utilizar, deben de cumplir con lo estipulado en la programación.

De forma que, ante procesos que requieran del uso del PLC, debe de establecerse la conexión OPC y compartir los datos, de forma que el modelo a simular pueda reproducirse sin problemas. Ante esto, modelos que no necesiten de un controlador PLC, pueden establecerse plenamente por conexión mediante cableado de componentes, a lo que solamente puedan estar vinculados a contactores sin control alguno.

Es importante recalcar que los procesos de conexión mediante cableado solamente, no necesitan de un servidor OPC para simularse, dado que no habría un PLC que lo controle, por lo tanto la simulación de procesos solamente mediante diagramas de fuerza, se pueden elaborar.

5.4.1. Estructuración de componentes

Para el modelamiento efectivo, es necesario saber establecer el modelo tal como si fuera la vida real, esto se refiere a que los componentes deben de estar interconectados, así como una debida alimentación y estructuración de objetos.

Para la determinación de terminales, se suele nombrar como x1 y x2 los de dicho componente, así como L1, L2 y L3 para los componentes que sean trifásicos.

Por ejemplo, para la alimentación y estructuración de un motor, se debe de establecer el contactor que le controlará y la alimentación trifásica en dicho objeto.

Figura 58. Estructuración de un motor

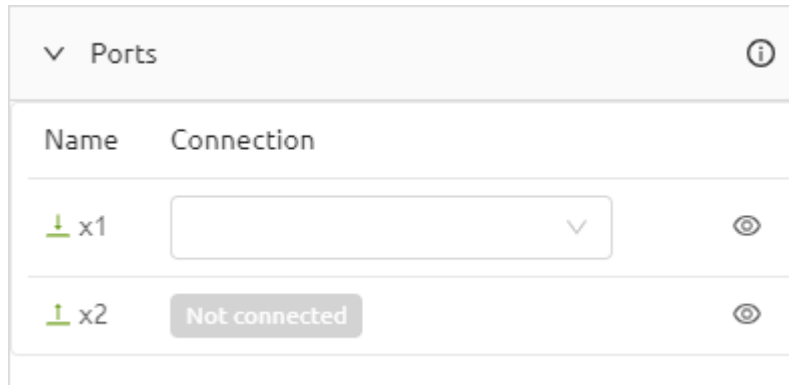
Ports		
Name	Connection	
l1	<input type="text"/>	
l2	<input type="text"/>	
l3	<input type="text"/>	
axis	Not connected	

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

La conexión entre los componentes se va denominando ante las otras conexiones disponibles para ser establecidas, de forma que la relación entre las variables sea simple y eficiente.

Ante botones que solamente necesiten de 2 terminales, se debe de relacionar con la fuente y cada terminal a donde sea referido, por ejemplo:

Figura 59. **Estructuración de un botón**



Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

La relación será establecida ante la fuente que le alimenta y el otro objeto que desea manipular, de forma que solo es establecido mediante 2 terminales, x1 y x2.

Todos los componentes disponibles tienen una estructuración similar, de forma que se relacionan mediante *Tags* entre objetos, donde así también se van relacionando las variables ante lo que se estipula manipular.

5.4.2. Conexión y cableado

La conexión y cableado se refiere a la relación entre los componentes, la composición del modelo y el diagrama de conexión. En Simumatik APP, las relaciones se establecen enlazando las terminales, ejes o variables de cierto objeto, respecto a otro componente que se haya elaborado dentro del escenario a trabajar.

Por consecuente, las relaciones de terminales se pueden establecer mediante un diagrama de conexión de fuerza, donde solo se refiere a la conexión eléctrica de los objetos y componentes simulados dentro del proyecto a elaborar.

En este aspecto, la relación de Codesys y su vinculación a dichos proyectos solo es respecto a variables de PLC, por lo tanto dichas variables se deben de relacionar a lo que manipulará, respecto a las conexiones de variables y tareas a ejercer.

Todo este aspecto de conexión se observa en la opción *Ports*, donde se deben de relacionar todas las conexiones establecidas. Sin dicha conexión, el modelo no se podrá simular y será considerado una simulación Model-In-The-Loop, donde solo se establece el modelo físico virtual.

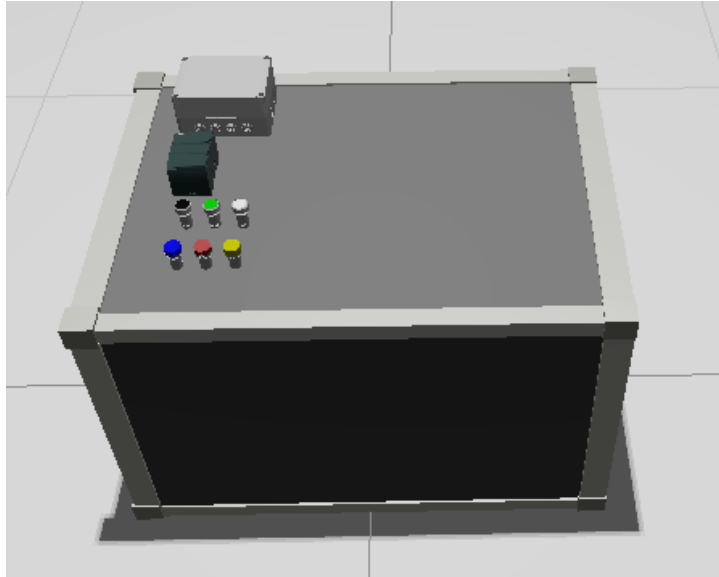
5.5. Ejemplificación

Como guía básica de conexión y relación de una comunicación OPC, se establecerá un mismo modelo donde se puede ejercer mediante solo cableado, y así también mediante un control de PLC.

5.5.1. Ejemplo básico de conexión y cableado

Para este ejemplo se utilizarán 3 botones o pulsadores que, al ser activadas, cada uno encenderá una luz correspondiente, y el modelo físico es de la siguiente manera:

Figura 60. **Modelo de pulsadores y luces**



Fuente: elaboración propia, empleando Simumatik APP 2020.

Consiste en:

- 3 botones pulsadores DC (S01, S02, S03)
- 3 Luces DC (HO1, H02, H03)
- Fuente de voltaje DC (PS01)
- Caja de almacenamiento (*BOX*)
- Mesa (*TABLE*)

Figura 61. **Estructuración**



Fuente: elaboración propia, empleando Simumatik APP 2020.

Cada luz y pulsador son de corriente continua, por consiguiente tienen 2 terminales de conexión cada uno. Por lo tanto, una terminal de cada luz (x1) debe ir a una salida de un pulsador, y la otra terminal (x2) al negativo de la fuente. Para la conexión de un pulsador, la terminal restante (x1) debe ir al positivo de la fuente.

De forma que, las conexiones se elaboren de la siguiente manera:

Figura 62. **Conexión H01**

Ports	
Name	Connection
x1	station.S01.x2
x2	station.PS01.dc_n

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Figura 63. **Conexión H02**

Ports	
Name	Connection
x1	station.S02.x2
x2	station.PS01.dc_n

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Figura 64. **Conexión H03**

Ports	
Name	Connection
x1	station.S03.x2
x2	station.PS01.dc_n

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Figura 65. **Conexión S01**

Ports	
Name	Connection
↓ x1	station.PS01.dc_p
↓ x2	1 Connection

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Figura 66. **Conexión S02**

Ports	
Name	Connection
↓ x1	station.PS01.dc_p
↓ x2	1 Connection

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Figura 67. **Conexión S03**

Ports	
Name	Connection
↓ x1	station.PS01.dc_p
↓ x2	1 Connection

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

5.5.2. Ejemplo de modelamiento mediante PLC

El siguiente modelo es similar al anteriormente planteado, con la diferencia que será controlado mediante un PLC, esto para demostrar el funcionamiento y la concatenación entre el control y el PLC. De la misma manera, se elaboró mediante lógica booleana, simplificando el ejercicio y la forma de construcción correspondiente.

El modelo comprende los siguientes componentes:

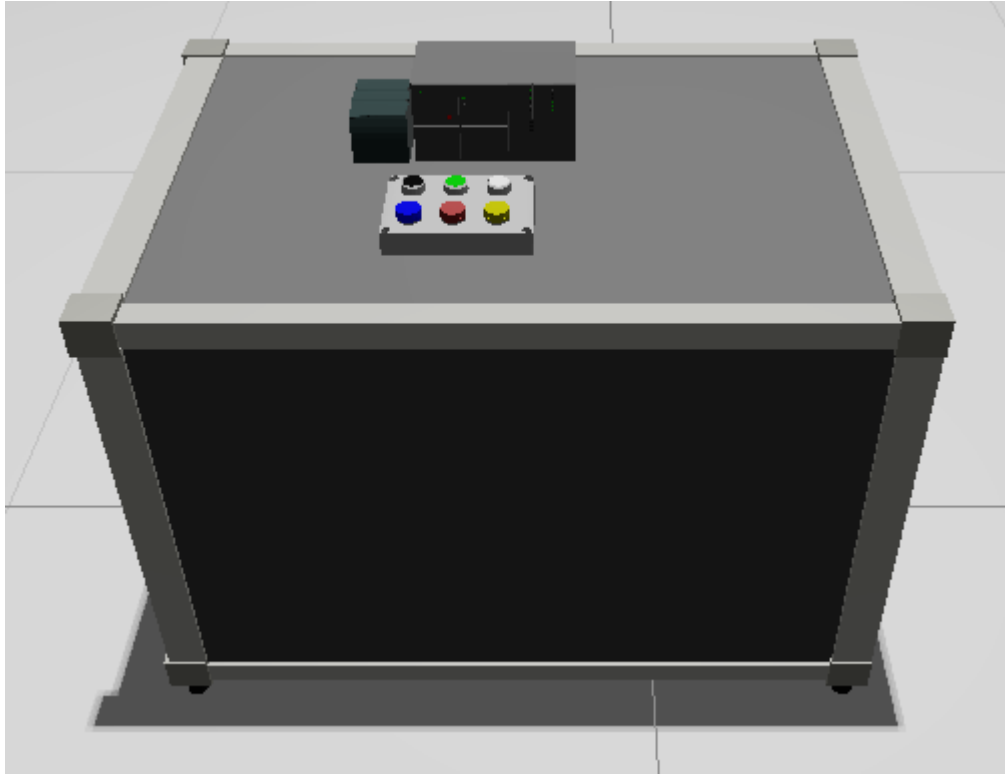
- 1 Controlador lógico programable (PLC)
- 3 botones pulsadores DC (S01, S02, S03)
- 3 Luces DC (HO1, H02, H03)
- Fuente de voltaje DC (PS01)
- Caja de almacenamiento (*BOX*)
- Mesa (*TABLE*)

Mediante una fuente DC, se proporcionará la alimentación a los pulsadores, PLC y cerrará el circuito mediante las luces.

Al pulsar un botón respectivo, el PLC concatenará el comportamiento y habilitará la luz que le corresponda a la tarea.

El modelo virtual de la estructura se elaboró de la siguiente forma:

Figura 68. **Estructura de modelo y PLC**



Fuente: elaboración propia, empleando Simumatik APP 2020.

Para el funcionamiento y conexiones, se estableció de forma que cada una de las luces piloto, una terminal fuera hacia el negativo de la fuente y la otra terminal a las salidas del PLC, de manera que la estructura sea ordenada y secuencial.

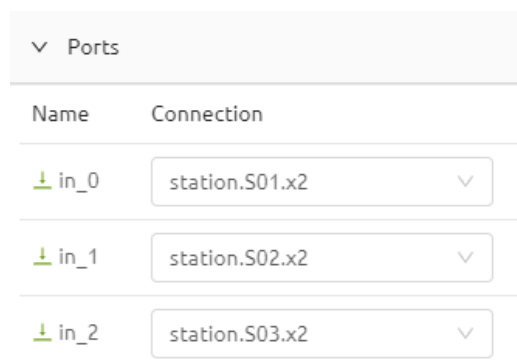
A lo que corresponde a los pulsadores, de igual manera, una de las terminales es conectada al positivo de la fuente, y la otra terminal a la entrada del PLC, de forma que sea escalonada y de manera respectiva a la luz a controlar.

Para la conexión en Codesys, el establecimiento de las variables debe de tener el mismo nombre que en Simumatik APP, de manera que corresponde a:

- H01 es la luz piloto 1
- H02 es la luz piloto 2
- H03 es la luz piloto 3
- S01 es el pulsador 1
- S02 es el pulsador 2
- S03 es el pulsador 3

Para el servidor OPC, se estableció mediante Codesys y el *Gateway* proporcionado por Simumatik APP, de forma que la programación fue booleana y la etiqueta de las variables elaborada correspondiente al objeto relacionado.

Figura 69. **Conexiones establecidas al PLC como entradas**



Name	Connection
in_0	station.S01.x2
in_1	station.S02.x2
in_2	station.S03.x2

Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Figura 70. **Conexiones establecidas al PLC como salidas**



Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Referente a las salidas, se establecieron proporcionalmente a las entradas, por lo tanto out_0 corresponde a H01, out_1 corresponde a H02 y la salida out_2 corresponde a H03.

5.5.3. Visualización del método de simulación SIL en Factory I/O

El software Factory I/O es una alternativa interesante para elaborar modelos y estructuras de automatización, así como el establecimiento de un servidor OPC mediante Codesys y establecer la lógica de eventos. A diferencia de Simumatik, Factory I/O posee varias alternativas de lenguajes de programación no enfocadas a un PLC, sino al comportamiento en general, de manera que por ejemplo se puede programar mediante compuertas e instrucciones de efecto.

Figura 71. **Ejemplo de modelo SIL en Factory I/O**



Fuente: elaboración propia, empleando toma de pantalla en Factory I/O.

El modelo se establece de forma que cada componente se estructure en base al diseño de automatización establecido, el comportamiento dependerá de la posición y etiqueta correspondiente.

Al diseñar la lógica del funcionamiento, debe ser claro que tanto una comunicación OPC o que elabora de forma local, la etiqueta de los objetos que componen la estructura deben de ser enlistados con el mismo nombre, de forma que se eviten conflictos en la falla de comunicación o la simulación errónea del escenario diseñado.

El proceso de comunicación OPC con Codesys es de forma más sencilla e igualmente no se requiere de un agente externo para dicha comunicación, sin embargo, la profundidad de los modelos y de la estructuración de los componentes no es tan específica como Simumatik.

Para una simulación HIL, Factory I/O posee un establecimiento de variables y datos para distintos tipos de PLC, donde se genera la conexión entre la variable a controlar y la variable que emula, de la siguiente forma:

Figura 72. **Ejemplo de establecimiento de variables en un PLC (HIL O SIL)**

Moving X	I0.0	Q0.0	Move X
Moving Z	I0.1	Q0.1	Move Z
Item detected	I0.2	Q0.2	Grab
Lid at place	I0.3	Q0.3	Lids conveyor
Lid clamped	I0.4	Q0.4	Clamp lid
Pos. at limit (lids)	I0.5	Q0.5	Pos. raise (lids)
Base at place	I0.6	Q0.6	Bases conveyor
Base clamped	I0.7	Q0.7	Clamp base
Pos. at limit (bases)	I1.0	Q1.0	Pos. raise (bases)
Part leaving	I1.1	Q1.1	Start light
Start	I1.2	Q1.2	Reset light
Reset	I1.3	Q1.3	Stop light
Stop	I1.4	(DINT) QD30	Counter
Emergency stop	I1.5		
Auto	I1.6		
FACTORY I/O (Running)	I1.7		

Fuente: elaboración propia, empleando toma de pantalla en Factory I/O.

CONCLUSIONES

1. El diseño del modelo y de la lógica del controlador determinan la eficiencia del proceso a automatizar, donde mediante la simulación SIL por medio del software Simumatik APP, se estipulan futuros errores o mejoras para su implementación futura.
2. La determinación del software de diseño y un software de lógica de programación para la transmisión y etiqueta de datos, se relaciona mediante un servidor OPC que convierte a un lenguaje en común entre los simuladores.
3. El método SIL consiste en el ahorro de recursos físicos para una debida simulación, de forma que se puede establecer de forma virtual sin necesidad de una intervención externa, por lo tanto el ordenador establece los parámetros y la posibilidad de compartir dichos datos.
4. El software Simumatik en general y Factory I/O son alternativas de modelación y diseño para una comunicación OPC. El software Codesys es una alternativa interesante para una programación lógica y comunicación, dado que posee un PLC virtual que simula y ejerce como un PLC real.
5. La descripción del funcionamiento y estructuración de los componentes del software utilizado, así como recomendaciones de uso y diseño de planta, se establecen mediante procedimientos directos y no externos, por lo que el ordenador obtiene la información, transferencia de etiquetas y

datos. Para su implementación, se requiere de una velocidad de internet amplia, de tiempos de subida / bajada considerablemente rápida, y ordenadores con sistemas operativos actualizados.

RECOMENDACIONES

1. Para el uso eficiente de una simulación SIL, se necesita información de los parámetros efectuados por cada software, así también como el procedimiento de una comunicación estable y sin conflicto con el servidor.
2. Mediante la estructura de cada software para una comunicación OPC, y el enlace de transferencia entre estos, requiere de ordenadores actualizados y con capacidad suficiente de memoria y simulación de gráficos.
3. Los servidores de Simumatik APP y el enlace entre el *Gateway* y una comunicación OPC, requieren de una conexión de internet estable, dado que todo proyecto se establece en la nube de información, y la vinculación de las variables en un estado dependiente.
4. Utilizar el modelo y diseño establecido, para una simulación HIL ya dependiente de un PLC físico, donde se tendrá mayor exactitud de la vinculación y actuación de las variables. Además considerar un estudio más profundo referente a Codesys y Simumatik APP para la simulación de plantas automatizadas, así también como el establecimiento de acciones físicas y *Virtual Comissioning*
5. Para su aplicación, tener en cuenta distintas alternativas para una simulación SIL, donde se pueden establecer simulaciones e interacciones virtuales, dado que el funcionamiento es en base a cómo

funciona Simumatik y Codesys. Asociado a la implementación, se debe contratar 1 suscripción mensual por cada 4 alumnos en laboratorio de valor de \$30, de forma que los alumnos se empiecen a familiarizar con este nuevo método de simulación. Para su docencia, tener en cuenta un mínimo de 4 horas semanales para el uso del software, distribuyendo 2 horas para prácticas de programación y conexiones virtuales, y 2 horas para diseño de ambientes.

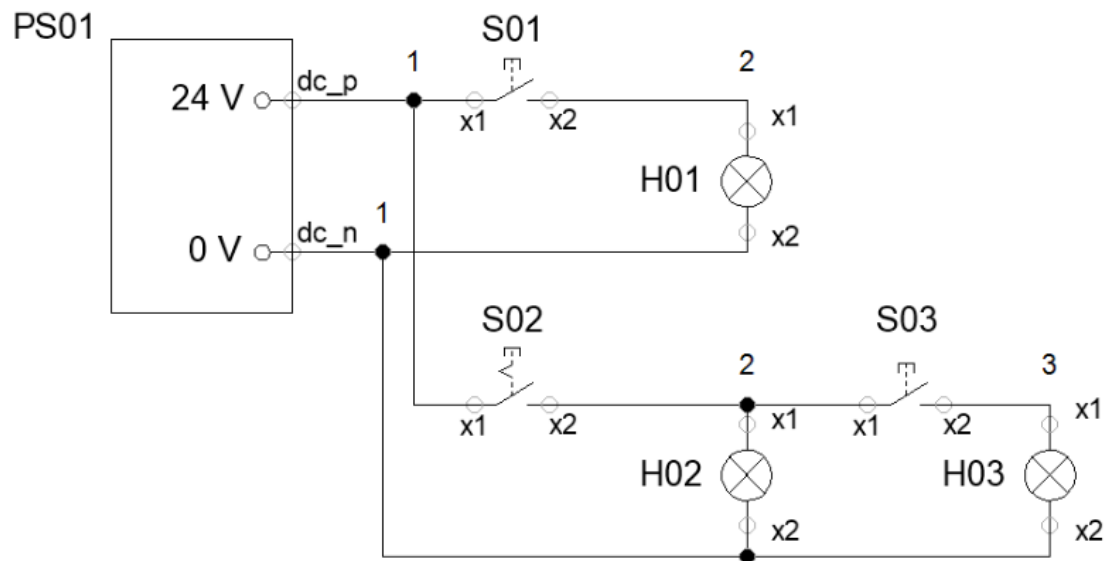
BIBLIOGRAFÍA

1. BERGSTROM, David; GORANSSON, Robert. *Model- and hardware-in-the-loop testing in a model-based design workflow*. Tesis de Master en Control Automático en el Departamento de Automatización y Control, LUND University, 2016. 57 p.
2. CALDERÓN GODOY, Antonio José. *Plataforma hardware-in-the-loop basada en labview, plc y opc. experiencia de uso docente en automatización y supervisión*. Cádiz, España: Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas, Universidad de Extremadura, 2016. 10 p.
3. DZINIC, Jasmin; YAO, Charlie. *Simulation-based verification of PLC programs*. Suecia: Tesis de Master en Ciencias en Ingeniería de Producción, 2013. 33 p.
4. GAEDKE, Martin; EIBL, Mximilian. *Model-in-the-loop and software-in-the-loop testing of closed-loop automotive software with arttest. Informatik*. Alemania: 2007. 1512 p.
5. GARCÍA MORENO, Emilio. *Automatización de procesos industriales*. 1a ed. Valencia, España: Alfaomega, 2001. 380 p.
6. GUERRERO, Vicente; YUSTE, Ramón; MARTÍNEZ, Luis. *Comunicaciones industriales*. 1a ed. México: Marcombo, S.A. 2010. 410 p

7. Laboratoire de Conception et Conduite des Systemes de Production L2CSP. *Hardware-in-the-loop simulation for validating plc programs*. Algeria: University of Mouloud Mammeri de Tizi Ouzou, 2015. 5 p.
8. LU, Bin., WU, Xin; FIGUEROA, Hernan. *A low-cost real-time hardware-in-the-loop testing approach of power electronics controls*. IEEE transactions on industrial electronics, 2007. 931 p.
9. RAMASWAMY, Deepa, y MCGEE, Ryan. *A Case Study in hardware-in-the-loop testing: development of an ecu for a hybrid electric vehicle*. SAE World Congress, 2004. 13 p.
10. ULRICH DONATH, Stephan Seidel. *Error-free control programs by means of graphical program design, simulation-based verification and automatic code generation*. Design Automation Division, 2011. 10 p.

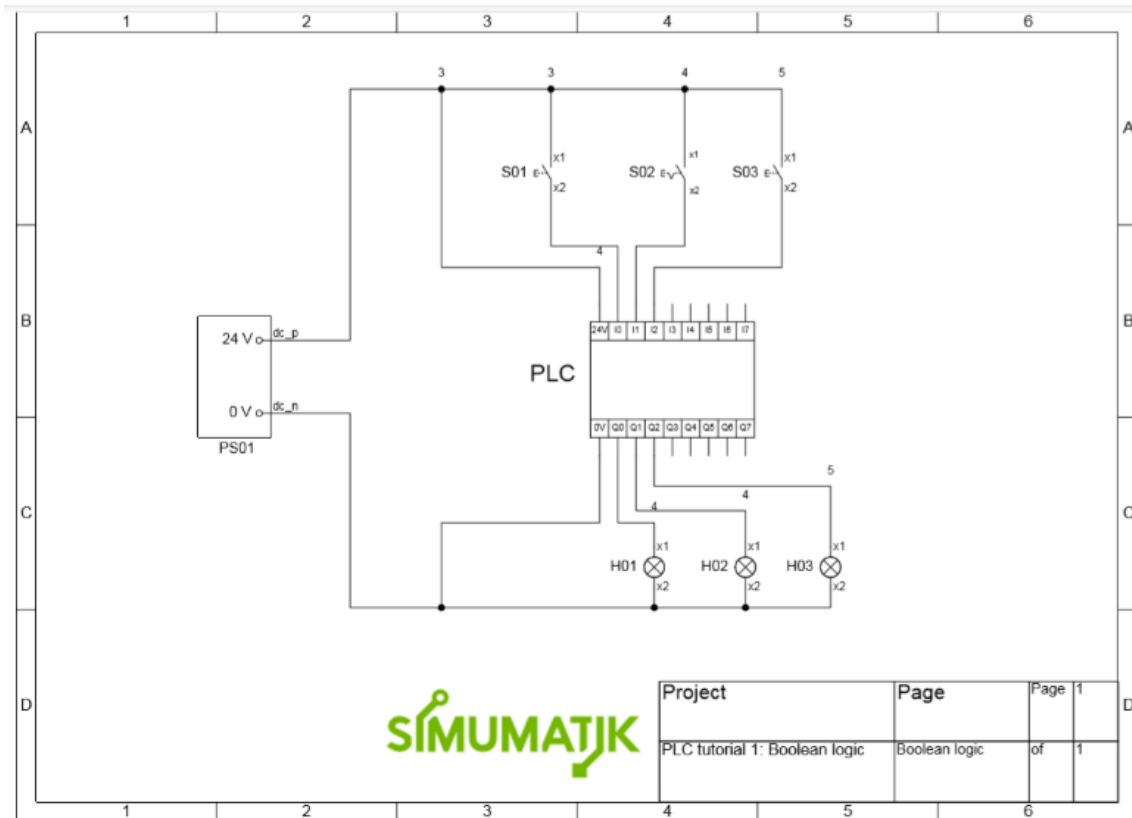
APÉNDICES

Apéndice 1. **Diagrama de conexión para ejemplo de cableado**



Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Apéndice 2. Diagrama de fuerza de ejemplo PLC



Fuente: elaboración propia, empleando toma de pantalla en Simumatik APP.

Apéndice 3. Programación de ejemplo booleano PLC



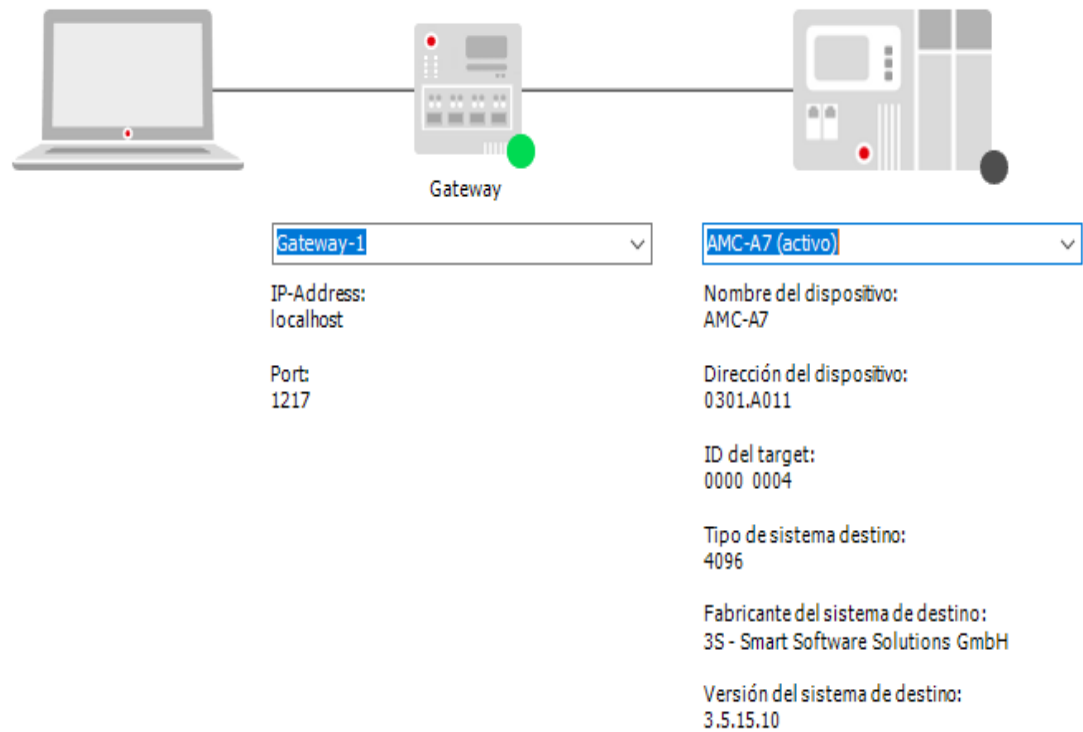
Fuente: elaboración propia, empleando Codesys 2020.

Apéndice 4. Declaración de variables de ejemplo booleano PLC

```
{attribute 'qualified_only'}  
VAR_GLOBAL  
  inputs AT %MB0: BYTE;  
  S01 AT %MX0.0: BOOL;  
  S02 AT %MX0.1: BOOL;  
  S03 AT %MX0.2: BOOL;  
  
  outputs AT %QB0: BYTE;  
  H01 AT %QX0.0: BOOL;  
  H02 AT %QX0.1: BOOL;  
  H03 AT %QX0.2: BOOL;  
  
END_VAR
```

Fuente: elaboración propia, empleando Codesys 2020.

Apéndice 5. Vinculación OPC en Codesys



Fuente: elaboración propia, empleando toma de pantalla en Codesys.

Apéndice 6. **Links de tutoriales de ejercicios para Simumatik**

1. https://youtu.be/mH3IFD_a9rw
2. <https://youtu.be/kIVdGgV0hzk>
3. <https://youtu.be/D5eqPuobBZA>

Fuente: elaboración propia.