

**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

TEMA:

**Prototipos de sensores inteligentes de caudal y nivel para la operación
de un subsistema de filtrado en una planta potabilizadora de agua**

AUTOR:

Andramuño Cando, Joselyne Paulette

Trabajo de Titulación previo a la obtención del título de
INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO

TUTOR:

M. Sc. Mendoza Merchán, Eduardo Vicente

Guayaquil, Ecuador

27 de Febrero del 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Srta.
Andramuño Cando, Joselyne Paulette como requerimiento para la
obtención del título de **INGENIERO ELECTRÓNICO EN CONTROL Y
AUTOMATISMO.**

TUTOR

M. Sc. Mendoza Merchán, Eduardo Vicente

DIRECTOR DE CARRERA

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, a los 27 días del mes de Febrero del año 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Andramuño Cando, Joselyne Paulette**

DECLARO QUE:

El trabajo de titulación **“Prototipos de sensores inteligentes de caudal y nivel para la operación de un subsistema de filtrado en una planta potabilizadora de agua”** previo a la obtención del título de **Ingeniero Electrónico en Control y Automatismo**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 27 días del mes de febrero del año 2020

EL AUTOR

ANDRAMUÑO CANDO, JOSELYNE PAULETTE



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

AUTORIZACIÓN

Yo, **Andramuño Cando, Joselyne Paulette**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Prototipos de sensores inteligentes de caudal y nivel para la operación de un subsistema de filtrado en una planta potabilizadora de agua”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 27 días del mes de febrero del año 2020

EL AUTOR

ANDRAMUÑO CANDO, JOSELYNE PAULETTE

REPORTE DE URKUND

REVISIÓN DE TRABAJO DE TITULACIÓN CON PROGRAMA URKUND CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y AUTOMATISMO

TÍTULO: “Prototipos de sensores inteligentes de caudal y nivel para la operación de un subsistema de filtrado en una planta potabilizadora de agua”.
AUTOR: ANDRAMUÑO CANDO JOSELYNE PAULETTE

The screenshot displays the URKUND software interface. On the left, a sidebar shows document details: 'Documento' (Tesis-ANDRAMUÑO CANDO JOSELYNE PAULETTE.docx), 'Presentado' (2020-02-13 01:11 (-05:00)), 'Presentado por' (joselynepau067@gmail.com), 'Recibido' (eduardo.mendoza01.ucsg@analysis.orkund.com), and 'Mensaje' (Tesis- Andramuño Cando Joselyne Paulette). The main area shows a 'Lista de fuentes' (List of sources) table with columns for 'Categoría' and 'Enlace/nombre de archivo'. The table lists various sources with their respective similarity percentages. At the bottom, a toolbar includes icons for 'Reiniciar', 'Exportar', and 'Compartir'.

Categoría	Enlace/nombre de archivo	Porcentaje
	TESIS.docx	
	UNIVERSIDAD CATÓLICA DE SANTIA...	73%
	UNIVERSIDAD CATÓLICA DE SANTIA...	60%
	Conclusiones. 59 Recomendaciones...	100%
	para la operación a escala de un sub...	83%
	Caracterizar desde el punto de vista ...	100%
	para el caso de una planta que tiene ...	100%
	Si el número máximo de marcas que...	100%
	un subsistema de filtración en una p...	93%
	un subsistema de filtración en una p...	73%
	ADJUNTO PDF:	97%

Después de analizar el resultado enviado por el programa Urkund, se determinó que el trabajo de titulación del estudiante: **ANDRAMUÑO CANDO JOSELYNE PAULETTE**, observa un porcentaje inferior al 4% de coincidencias con otros documentos encontrados en el internet.

.....
Ing. Eduardo Mendoza Merchán, Mgs
DOCENTE TUTOR

DEDICATORIA

Este trabajo de graduación está dedicado a Dios, por su amor infinito, por ser mi amigo fiel e incondicional, por guiar mi vida, por su misericordia, por ser pilar fundamental en mi vida y porque me ha dado fe, fortaleza, inteligencia, sabiduría y sapiencia para culminar uno de mis tan preciados sueños en mi vida profesional.

A mis padres, que a lo largo de toda mi vida me apoyaron en todas las metas que me propuse, que me dieron mucho amor, que me enseñaron a discernir lo bueno y lo malo, que me dieron sabios consejos cuando lo necesité, que me educaron y me formaron con preceptos éticos y religiosos, que me guiaron por el camino de la verdad, de la honradez y de la honestidad, y que día a día me ayudaron a forjar el camino de mi carrera profesional.

A la memoria de mi mamá Lucrecia y de mi papá Vicente, por su gran amor a mí, por la gran calidad humana que demostraron tener, por la ternura y dulzura con que me trataron, por lo consejos llenos de sabiduría que me dieron, y porque han sido los mentores de mi vida.

Joselyne Paulette

AGRADECIMIENTO

Agradezco de corazón a Dios por tener en mi vida su guía espiritual, la que hizo posible que no me rinda y me ayude a vencer los obstáculos que se me presentaron en mi vida estudiantil. También le agradezco a mi Dios, por darme sabiduría e inteligencia para entender y comprender las enseñanzas que me impartieron en mi carrera estudiantil.

Agradezco a mis padres, por ser ejemplo de vida, porque siempre me motivaron a que siga estudiando y no me rinda jamás, porque con su gran amor han forjado en mí la fortaleza espiritual que ha hecho posible enfrentar con decisión y optimismo los retos en mi vida.

Agradezco al Ing. Mendoza, mi tutor, por ser mi guía en mi tesis, por dar su valioso tiempo y aportar con su criterio acertado a este trabajo de graduación. Le agradezco por sus invalorable consejos que me ha brindado, los que han hecho posible elevar el nivel de mis conocimientos. También le agradezco por ser un gran profesional, a quien admiro mucho y por brindarme su apoyo y su amistad.

Joselyne Paulette



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

TRIBUNAL DE SUSTENTACIÓN

f. _____

M. Sc. ROMERO PAZ, MANUEL DE JESUS

DECANO

f. _____

M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO

COORDINADOR DEL ÁREA

f. _____

M. Sc CÓRDOVA RIVADENEIRA, LUIS SILVIO

OPONENTE

Índice General

Índice de Figuras	XIII
Índice de Tablas	XV
RESUMEN	XVI
ABSTRACT	XVII
INTRODUCCIÓN	2
CAPÍTULO I: ASPECTOS GENERALES	4
1.1. Planteamiento del Problema.....	4
1.2. Justificación.....	5
1.3. Delimitación	5
1.4. Objetivos	5
1.4.1. Objetivo General	5
1.4.2. Objetivos Específicos.....	6
1.5. Metodología de Investigación.....	6
CAPÍTULO II: MARCO TEÓRICO.....	8
2.1. Estado de Arte	8
2.2. Lenguaje Unificado de Modelado (UML)	11
2.2.1. Comprensión de Modelado y UML.....	12
2.2.2. Diagramas de UML.....	12
2.2.2.1 Diagramas de Casos de Uso.....	12
2.2.2.2 Diagramas de Estados	13
2.2.2.3 Diagramas de Secuencia.....	14
2.2.2.4 Diagramas de Actividades	14
2.3. Sensores Inteligentes.....	15
2.3.1. Definición de Sensor	15
2.3.2. Definición de Sensor Inteligente	16
2.3.3. Funciones del Sensor Inteligente	17
2.4. Arquitecturas para sensores inteligentes.....	17
2.4.1. Arquitectura de un sistema de sensado inteligente “Micrófono Inteligente”	19

2.4.2.	Arquitectura de un sistema de sensado inteligente y semi - inteligente	20
2.4.3.	Arquitectura de sensor inteligente para dispositivos móviles	22
2.4.4.	Arquitecturas deliberativas.....	22
2.4.4.1	Agente Deliberativo BDI.....	23
2.5.	Sensor de Caudal	24
2.5.1.	Pasos para elegir un sensor de Caudal adecuado	24
2.5.2.	Tipos de Sensor de Caudal.....	24
2.6.	Sensor de Nivel.....	25
2.6.1.	Clasificación de los sensores de Nivel	26
2.6.2.	Funcionamiento de los Sensores de Nivel	26
2.6.3.	Unidades Utilizadas en Nivel de Líquidos.....	28
2.7.	Protocolo MODBUS.....	28
2.7.1.	Tipos de datos que permite MODBUS.....	28
2.7.2.	Estados del Maestro MODBUS.....	29
2.7.3.	Estados del Esclavo MODBUS	30
2.7.4.	Modo de Transmisión RTU	31
2.7.5.	Modo de Transmisión ACII.....	32
2.8.	Microcontroladores.....	32
2.8.1.	Componentes de un microcontrolador	33
2.8.2.	Arquitectura von Neumann y Harvard.....	34
2.8.2.1	Arquitectura von Neumann.....	34
2.8.2.2	Arquitectura Harvard	35
2.8.3	Arquitecturas RISC y CISC	35
2.8.3.1	Arquitectura CISC	35
2.8.3.2	Arquitectura RISC	36
2.9	Redes de Petri.....	37
2.9.2	Elementos de las redes de Petri.....	37
2.9.3	Interpretación de los grafos en RdP	38
2.9.4	Propiedades de las redes de Petri.....	39

2.9.5	Condiciones de diseño para las redes de Petri.....	39
CAPÍTULO III: ANÁLISIS DEL PROTOTIPO DE SENSORES INTELIGENTES DE CAUDAL Y NIVEL.....40		
3.1.	Consideraciones Iniciales del Sistema.....	40
3.1.1	Descripción del Sistema	40
3.1.2	Caracterización del Sistema.....	42
3.1.3	Funciones del Sistema.....	43
3.2.	Diseño de la Arquitectura Nodo Sensor/ Actuador.....	44
3.2.1.	Análisis de la primera etapa de la Arquitectura del Sistema....	45
3.2.2.	Análisis de la segunda etapa de la Arquitectura del Sistema ..	47
3.2.3.	Análisis de la tercera etapa de la Arquitectura del Sistema.....	48
3.2.4.	Arquitectura Final del Sistema.....	49
3.3.	Modelado UML del Nodo Sensor/Actuador.....	49
3.3.1	Diagrama de Casos de Usos.....	50
3.3.1.1	Casos de Uso del Agente.....	50
3.3.1.2	Casos de Uso del Sensor.....	51
3.3.1.3	Casos de Uso del Técnico	51
3.3.1.4	Casos de Uso del Autómata	52
3.3.2	Diagrama de Secuencia.....	52
3.3.2.1	Diagrama de Secuencia Autómata – Agente	54
3.3.2.2	Diagrama de Secuencia Técnico – Agente	55
3.3.3	Diagrama de Estados.....	56
3.3.3.1	Diagrama de Estados: Configuración del Sensor.....	57
3.3.3.2	Diagrama de Estados: Configuración Agente/ PLC.....	58
3.3.3.3	Diagrama de Estados: Detección de Errores	59
3.3.3.4	Diagrama de Estados: Selección de Comportamiento.....	59
3.4.	Diseño de la Red de Petri Nodo Sensor/ Actuador	61
3.4.1	Red de Petri del Sensor Inteligente	63
3.4.2	Redes de Petri de los Sub-estados del Flujo General.....	65
3.4.2.1	Red de Petri del Sub-estado: Configuración del Sensor	65

3.4.2.2	Red de Petri del Sub-estado: Configuración Agente/PLC	66
3.4.2.3	Red de Petri del Sub-estado: Detección de Errores	67
3.4.2.4	Red de Petri del Sub-estado: Selección de Comportamiento ..	68
3.4.3	Análisis de las Invariantes de la Red de Petri	69
3.4.3.1	Análisis de las Invariantes de la Red de Petri General	69
3.4.3.2	Análisis de las Invariantes del Sub-estado 1	70
3.4.3.3	Análisis de las Invariantes del Sub-estado 2	71
3.4.3.4	Análisis de las Invariantes del Sub-estado 3	72
3.4.3.5	Análisis de las Invariantes del Sub-estado 4	72
3.5.	Trama del Protocolo MODBUS	73
3.6.	Pruebas Experimentales del Sensor Inteligente	74
3.5.1	Prueba Experimental de Nivel	75
3.5.2	Prueba Experimental de Caudal	78
CAPÍTULO IV: PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS		81
4.1	Presentación de Resultados	81
4.1.1	Presentación de Resultados relacionados al Estado de Arte	81
4.1.2	Resultados obtenidos en Hardware	82
4.1.3	Resultados obtenidos en Software	82
4.1.4	Resultados obtenidos en las Pruebas Experimentales	83
4.2	Discusión de Resultados	84
4.2.1	Análisis de las características y funciones del sistema	84
4.2.2	Cumplimiento de los Objetivos	85
4.2.2.1	Objetivo General	85
4.2.2.2	Objetivos Específicos	85
4.2.3	Análisis de la Metodología	86
4.2.4	Análisis de la Solución del Problema de Investigación	86
4.2.5	Discusión Final	87
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES		88
5.1	CONCLUSIONES	88
5.2	RECOMENDACIONES	89
Bibliografía		90

Índice de Figuras

Capítulo 1:

No se encuentran elementos de tabla de ilustraciones.

Capítulo 2

Figura 2. 1: Creadores de UML(Lenguaje Unificado de Modelado).....	11
Figura 2. 2: Ejemplo del diagrama de casos de uso: sistema bancario	13
Figura 2. 3: Ejemplo de diagrama de estados, de un individuo busca su alimento para comerlo.	13
Figura 2. 4: Ejemplo de diagrama de secuencia básico.	14
Figura 2. 5: Ejemplo de diagrama de Actividades de un sistema.....	15
Figura 2. 6: Sensor de presión Industrial.....	16
Figura 2. 7: Estructura de un sensor inteligente.....	16
Figura 2. 8: Estructura de una arquitectura de sensor inteligente.....	18
Figura 2. 9: Arquitectura de un micrófono inteligente	19
Figura 2. 10: Arquitectura de un sensor inteligente	20
Figura 2. 11: Arquitectura de un sensor semi- inteligente	21
Figura 2. 12: Arquitectura de sensor inteligente para dispositivos móviles ...	22
Figura 2. 13: Arquitectura de agente deliberativo BDI	23
Figura 2. 14: Diagrama que representa el volumen de un fluido en un recipiente	25
Figura 2. 15: Sensores de Medida Directa.....	27
Figura 2. 16: Sensores de Flotador	27
Figura 2. 17: Diagrama de estados del maestro MODBUS.....	30
Figura 2. 18: Diagrama de estados del maestro MODBUS.....	31
Figura 2. 19: Secuencia del chequeo de paridad en RTU.....	32
Figura 2. 20: Secuencia del chequeo de paridad en ASCII.....	32
Figura 2. 21: Diagrama de bloques de un microcontrolador	33
Figura 2. 22: Componentes de un microcontrolador	34
Figura 2. 23: Diagrama de la arquitectura Von Neumann	34
Figura 2. 24: Diagrama de la arquitectura Harvard.....	35
Figura 2. 25: Diagrama de la arquitectura CISC	36
Figura 2. 26: Diagrama de la arquitectura RISC	36
Figura 2. 27: Carl Adam Petri creador de las redes de Petri.....	37
Figura 2. 28: Elementos de la Red de Petri	38

Capítulo 3

Figura 3. 1: Diagrama de Flujo general	42
Figura 3. 2: Elementos del modelo BDI	45
Figura 3. 3: Relación de los elementos del modelo BDI	45
Figura 3. 4: Primera etapa de la arquitectura del sistema	46
Figura 3. 5: Segunda etapa de la arquitectura del sistema	47
Figura 3. 6: Tercera etapa de la arquitectura del sistema	48
Figura 3. 7: Arquitectura final del sistema	49

Figura 3. 8: Casos de Uso del Sensor Inteligente.....	50
Figura 3. 9: Diagrama de Secuencia General del Sensor Inteligente.....	53
Figura 3. 10: Diagrama de Secuencia Autómata - Agente.....	54
Figura 3. 11: Diagrama de Secuencia Técnico- Agente.....	55
Figura 3. 12: Diagrama de Estados general del sistema.....	57
Figura 3. 13: Diagrama de Estados del Sub- proceso Configuración del Sensor.....	58
Figura 3. 14: Diagrama de Estados del Sub- proceso Configuración Agente/PLC.....	59
Figura 3. 15: Diagrama de Estados del Sub- proceso Detección de Errores.....	60
Figura 3. 16: Diagrama de Estados del Sub- proceso Selección de Comportamiento.....	61
Figura 3. 17: Diagrama de Flujo del Sensor Inteligente.....	62
Figura 3. 18: Red de Petri del Sensor Inteligente.....	64
Figura 3. 19: Red de Petri del Sub-estado: Configuración del Sensor.....	66
Figura 3. 20: Red de Petri del Sub-estado: Configuración Agente/PLC.....	67
Figura 3. 21: Red de Petri del Sub-estado: Detección de errores.....	68
Figura 3. 22: Red de Petri del Sub-estado: Selección de Comportamiento ..	69
Figura 3. 23: Pruebas con Nivel de Agua.....	74
Figura 3. 24: Pruebas con Caudal de Agua.....	75
Figura 3. 25: Gráfica Nivel vs Tiempo.....	76
Figura 3. 26: Gráfica Nivel vs Tiempo.....	77
Figura 3. 27: Gráfica Caudal vs Frecuencia.....	79
Figura 3. 28: Gráfica Caudal vs Frecuencia.....	79
Figura 3. 29: Sensor No inteligente vs Sensor Inteligente.....	80

Índice de Tablas

Capítulo 2

Tabla 2. 1: Unidades utilizadas en la Medición de Nivel (Cárdenas, 2009) ..	28
Tabla 2. 2: Datos que se utilizan en MODBUS (Barastegui & Castaño, 2017)	
.....	28

Capítulo 3

Tabla 3. 1 Invariantes de las Transiciones: RdP General (Autor)	70
Tabla 3. 2 Invariantes de los Lugares: RdP General (Autor)	70
Tabla 3. 3 Invariantes de las Transiciones: Sub-estado1 (Autor).....	70
Tabla 3. 4 Invariantes de los Lugares: Sub-estado 2 (Autor).....	71
Tabla 3. 5 Invariantes de las Transiciones: Sub- estado 2 (Autor)	71
Tabla 3. 6 Invariantes de los Lugares: Sub-estado 2 (Autor).....	71
Tabla 3. 7 Invariantes de los Lugares: Sub-estado 3 (Autor).....	72
Tabla 3. 8 Invariantes de los Lugares: Sub-estado 4 (Autor).....	72
Tabla 3. 9 Trama MODBUS RTU (Autor)	73
Tabla 3. 10 Trama MODBUS RTU (Autor).....	73
Tabla 3. 11 Tiempo de Estabilización del Nivel (Autor).....	76
Tabla 3. 12 Porcentaje de Error Nivel (Autor).....	77
Tabla 3. 13 Tiempo de Estabilización del Nivel (Autor).....	78
Tabla 3. 14 Porcentaje de Error Caudal (Autor)	80

RESUMEN

El uso de técnicas de inteligencia artificial en arquitecturas para procesos industriales, son de diferentes tipos, como es el uso de agentes inteligentes para el control de procesos, pero no se ha encontrado mucha base científica en el uso de agentes de tipo deliberativos para el diseño de una arquitectura inteligente que sea adaptable a cualquier variable que el usuario desee controlar. El presente trabajo de investigación propone una arquitectura para el control de un sub-sistema de filtrado para una planta potabilizadora de agua utilizando agentes inteligentes BDI para que sea apto a cualquier variable de control, para modelar este sistema se utiliza UML una herramienta muy utilizada por los informáticos y muy poco utilizado para la industria. Para establecer una comparación entre un estado ideal y la realidad se tiene las Redes de Petri, con las que se puede simular la arquitectura y el modelado planteado previamente. La investigación es de tipo correlacional con un enfoque cuantitativo, los métodos que se utilizan son: análisis y síntesis para valorar todas las fuentes de información que se obtengan, simulación para evaluar el desempeño del sistema y experimental para comprobar si funciona en el campo real.

Este trabajo de investigación tiene como resultados la arquitectura adaptable para cualquier variable, el modelado de UML y la RdP correctamente planteados, que se comprueba en la valoración teórico práctica con las pruebas experimentales que arrojan un tiempo de respuesta menor a los 100ms, y se mantiene estable en cuanto cambia de una medida a otra.

Palabras Claves: SENSOR INTELIGENTE, ARQUITECTURA INTELIGENTE, UML, REDES DE PETRI, BDI.

ABSTRACT

The use of artificial intelligence techniques in architectures for industrial processes, are of different types, such as the use of intelligent agents for process control, but not much scientific basis has been found in the use of deliberative type agents for design of an intelligent architecture that is adaptable to any variable that the user wishes to control. The present research paper proposes an architecture for the control of a filtering sub-system for a water purification plant using intelligent BDI agents to be suitable for any control variable, to model this system a tool used by UML is used by the computer and very little used for the industry. To establish a comparison between an ideal state and reality, you have the Petri Nets, with which you can simulate the architecture and modeling outlined previously. The research is correlational with a quantitative approach, the methods used are: analysis and synthesis to assess all sources of information obtained, simulation to evaluate the performance of the system and experimental to check if it works in the real field.

This research work has the results of the adaptable architecture for any variable, the modeling of the UML and the correctly proposed RdP, which is verified in the practical theoretical assessment with the experimental tests that yield a response time of less than 100ms, and is maintained stable as soon as it changes from one measure to another.

Keywords: SMART SENSOR, ARCHITECTURE, MODELING LANGUAGE, PETRI NET, BDI.

INTRODUCCIÓN

La evolución tecnológica a través de los años, han dado paso a la Industria 4.0 o la cuarta revolución industrial, la que implica el uso de métodos nuevos para el control de procesos industriales a través de la inteligencia artificial. Pero no es el único método que se puede emplear para el control industrial.

Los agentes inteligentes utilizados para elaborar software aportan con autonomía, sociabilidad, capacidad de reacción e iniciativa. Estas ventajas se pueden utilizar para el control de procesos industriales de forma eficiente y fiable. Es decir que los agentes inteligentes también se pueden utilizar para el control y la automatización industrial.

La arquitectura que se propone en este trabajo de investigación está basada en agentes inteligentes del tipo deliberativos BDI, se compone de creencias que es todo lo que el agente sabe del entorno en este caso sabrá en que momento está midiendo caudal y en que momento el nivel, debido a una pequeña "base de datos" donde se almacenan rangos de operación, frecuencias de trabajo, entre otros valores que son necesarios para que el agente aplique las funciones necesarias para estas variables.

Las funciones de este sensor para que se convierta en inteligente son muchas pero la que resalta más es la capacidad de auto calibrarse cada vez que lo necesite, lo que lleva a que la arquitectura diseñada sea para una variable a la vez, si se quiere agregar una nueva variable el sistema se interrumpe para darle paso a esta nueva variable y continuar cumpliendo sus funciones para llegar a comunicarse con el exterior, que es el autómatas del sistema.

La comunicación con el autómatas debe ser fiable y segura, por lo que se implementa el protocolo industrial MODBUS, para que la comunicación sea bidireccional Agente/PLC y PLC/Agente.

Una vez establecida el protocolo de comunicación, se procede al modelado del sistema general, dando paso al uso de un método nuevo para sistemas de control industrial, pero comúnmente utilizado en el modelado de software por la rama de Ingeniería en Sistema, este tipo de modelado se lo conoce como UML.

El uso de UML para modelar este sistema, permite comprender su funcionamiento con mayor detalle, en esta sección es donde se define cuantos actores o dispositivos van a intervenir en el sistema, las actividades que cada debe realizar, como las va a realizar y en que tiempo las harán. Pero para comprobar que todo esto

es correcto se utilizan las Redes de Petri para simular y ver si funciona de forma adecuada la arquitectura en el sensor inteligente.

El modelado de la Red de Petri permite ver cuantos sub-estados se generan a partir del sistema general, y en caso de que exista alguno se realiza cada sub-red por separado, para observar mediante simulación la respuesta del sistema general con estos sub-estados y como se comportan independientemente.

Los resultados de simulación se los utiliza para modelar la programación en Arduino IDE, ya que este sistema cuenta con una placa Arduino Mega 2560 de 8 bits lo necesario para enviar o recibir dato del PLC. Al probar el funcionamiento de este programa se comprueba si la arquitectura es la correcta, si el modelado estuvo bien realizado y con la simulación de las Redes de Petri, si la red estuvo bien construida y no existe algún error que en la práctica se sale del estado ideal, dando paso al estado real donde el sensor tiene que responder de manera óptima para la variable que se esté midiendo en ese momento.

Al desarrollar este trabajo de investigación con cada uno de los componentes de estudio, se cumple con los objetivos planteados, lo que comprueba que esta nueva arquitectura basada en agentes inteligentes es totalmente viable para ser implementada en el control de procesos industriales, convirtiendo así al sensor en inteligente debido al agente o microcontrolador que se va a dedicar a cumplir funciones del sensor, no de otras partes del sistema en general.

Se logra proponer un método novedoso para una arquitectura inteligente utilizando agentes de tipo deliberativos BDI, para que sea adaptable a cualquier variable, microcontrolador y rangos de operación que el usuario desee controlar. Esto se logra a través del modelado en UML donde se establecen las funciones, los dispositivos que intervienen, en que momento van a intervenir y de que forma. Lo mismo que fue comprobado mediante la simulación en Redes de Petri para la red general y cada sub-estado que se genera a partir de las interrupciones que el sistema contiene. A través de la valoración teórico- práctica se comprobó que la arquitectura planteada funciona de forma eficiente en los sensores que se utilizan para nivel y caudal. Los tiempos de estabilización están por debajo de los 100ms y la precisión del sensor es muy buena, con respecto a la variable que controla en el tiempo.

CAPÍTULO I: ASPECTOS GENERALES

1.1. Planteamiento del Problema

Actualmente existen diferentes tipos de arquitecturas de control para procesos industriales, esto ha permitido que los sistemas utilicen nuevos controladores como la lógica difusa o control borroso integrando así la inteligencia artificial a estos sistemas. Estas arquitecturas tienen la ventaja de que permiten nuevas formas de control o uno más avanzado ya que muestran la información del proceso a todo nivel, es decir un control distribuido que hace al sistema más robusto y eficiente. Existen muy pocos estudios que ofrecen los prototipos de arquitecturas de control utilizando un lenguaje de modelado unificado que luego puede ser transcrito al lenguaje de programación que se desee en cualquier sistema embebido que se vaya a utilizar (Selvakumar et al., 2019). El lenguaje de modelado unificado o UML son una alternativa que utilizan los desarrolladores de software que les brindan el análisis y diseño de sistemas computacionales. En el campo del control de procesos industriales muy pocos autores han utilizado UML para diseñar el sistema haciéndolo inteligente; sin embargo, el uso de estos diagramas aporta con muchas ventajas siendo una de ellas es que presenta la forma de operación del sistema de una forma organizada y entendible. El uso de UML para sensores inteligentes permite que la forma de funcionamiento o diseño de la arquitectura de control, sea permitida para cualquier plataforma en la que se quiera programar. El problema se encuentra en la poca información científica que se tiene sobre este tema, por lo cual no se puede establecer mucha comparativa con los futuros resultados que se obtengan.

En base a lo anterior presentado se ha planteado la siguiente interrogante de investigación:

“¿Cómo incide el uso de técnicas de inteligencia artificial en la capacidad de procesamiento y desempeño de un sensor inteligente basado en microcontroladores?”.

1.2. Justificación

Las arquitecturas de control diseñadas con UML (Unified Modeling Language) aportan con gran valor científico ya que no solo puede ser usado por profesionales en sistemas computacionales, sino que también pueden ser implementados en la industria de procesos. Es de gran relevancia tecnológica ya que se utilizan sensores inteligentes para el control de un proceso industrial, es decir se utiliza la inteligencia artificial lo que hace al sistema mejor distribuido y autónomo. El diseño de prototipos de sensores inteligentes para caudal y nivel que se estudiará es de gran importancia en las industrias ya que una de sus principales funciones es quitarle carga a los PLC, ya que estos sensores realizan más tareas de forma autónoma. Los resultados que se obtengan una vez diseñada la arquitectura van a ser demostrados mediante simulaciones para observar el correcto funcionamiento de la misma dentro de las especificaciones que se establezcan del proceso general. Estos sensores se van a basar en el control de un proceso de filtrado de agua con el fin de que se brinde un mejor servicio para el usuario final.

1.3. Delimitación

Los resultados del estudio se orientan a obtener prototipos de sensores inteligentes de caudal y nivel, los cuales serán modelados utilizando herramientas tales como UML y Redes de Petri. Estos prototipos serán simulados y probados experimentalmente para determinar su comportamiento y el rendimiento en términos de métricas, que luego permitirán realizar comparativas con trabajos de otros investigadores.

1.4. Objetivos

1.4.1. Objetivo General

Proponer una arquitectura para prototipos de sensores inteligentes de caudal y nivel basados en microcontroladores y comunicación con autómatas programables, para la operación a escala de un sistema de filtración de una planta potabilizadora de agua.

1.4.2. Objetivos Específicos

- Proponer una arquitectura de sensores inteligente para la medición de nivel y caudal considerando los requerimientos en el control del subsistema de filtración en una planta potabilizadora de agua.
- Realizar el modelado en UML y la programación del microcontrolador para el control inteligente de los sensores según la arquitectura propuesta.
- Realizar una valoración teórico-práctica de la operación de los sensores de nivel y caudal propuestos.

1.5. Metodología de Investigación.

El estudio de los prototipos de sensores inteligentes de caudal y nivel contiene diversos temas de estudio como es la lógica difusa para la programación de las tareas que van a realizar estos sensores, el modelado en UML del algoritmo que se va a utilizar para estos dispositivos. Todo esto lleva a que el tipo de investigación que se va a realizar en este trabajo de investigación es del tipo correlacional con un enfoque cuantitativo, porque al obtener los resultados serán de forma numérica experimental con el fin de realizar comparativas entre lo ideal y lo real del sistema. Por otro lado, para tener una base científica de lo que se va a realizar se va a utilizar referencias bibliográficas y datos experimentales. La primera parte se basa en todo lo recopilado de artículos científicos de diferentes repositorios, libros, etc. La segunda parte se la va a realizar una vez obtenido el modelado en UML del sistema y realizado el algoritmo de control para los sensores inteligentes.

En base a la metodología propuesta, se muestran las tareas de investigación para cumplir con los objetivos planteados:

- 1- Caracterizar desde el punto de vista teórico, histórico y contextual los sensores inteligentes para subsistemas de filtración de plantas potabilizadora de agua.
- 2- Describir y caracterizar los requerimientos a nivel de sensores y actuadores en el control del subsistema de filtración en una planta potabilizadora de agua.

- 3- Definir las operaciones y variables para los sensores de nivel y caudal para el caso de una planta que tiene seis unidades de filtración.
- 4- Proponer una arquitectura de sensores inteligente para sensores de nivel y caudal con un enfoque multi-agente.
- 5- Proponer un protocolo de comunicación sensor inteligente – PLC con enfoque de inteligencia distribuida.
- 6- Realizar el diseño de los sensores para control inteligente distribuido utilizando modelado en UML.
- 7- Realizar la programación para el control inteligente de los sensores según la arquitectura propuesta.
- 8- Realizar una valoración teórico-práctica de la operación de los sensores de nivel y caudal propuestos.

CAPÍTULO II: MARCO TEÓRICO

2.1. Estado de Arte

En esta sección se va a revisar los últimos avances científicos desarrollados por diferentes investigadores en los últimos años, los cuales sirven de soporte científico para el trabajo a desarrollar.

Los sensores inteligentes son dispositivos que integran la detección y el procesamiento de datos en una sola unidad física, o también se lo puede definir como la mezcla entre un sensor analógico/digital más un procesador, una memoria y un controlador en un mismo espacio físico. Según(Ando, 1997) los objetivos de los sensores inteligentes es unificar la automatización y la optimización de recursos, lo cual quedó demostrado en su artículo que describe el uso de los mismos para el aprendizaje de mecanismos sensoriales humanos, en el que usó una arquitectura de espina de pescado para organizar las funciones y un sensor táctil estructurado. Este estudio lo llevó a la conclusión de que este tipo de sensores contribuye con información en tiempo real lo que ayuda a la toma de decisiones y a procesar los datos de manera distribuida(Premkumardeepak & Krishnan, 2017).

En los siguiente artículos de (Shen et al., 2009) y (Schonberg et al., 2009) utilizan los sensores inteligentes para diferentes aplicaciones como medir la velocidad y dirección del viento, con el fin de estabilizar la temperatura en un sistema de control. Por otro lado, también presentan el uso de un sensor inteligente para medir temperatura de un perno industrial, lo que les permitió reducir el uso de empaques térmicos tradicionales para los componentes de los sensores comunes. Con esto demostraron que el desarrollo de dispositivos industriales inteligentes que sean capaces de enviar información sobre el estado de la máquina en tiempo real permite que se elabore un mantenimiento eficaz en la planta de trabajo, o también permite un mejor control de un sistema que su perturbación dependa de agentes externos naturales como el viento.

Según (Rivera et al., 2011) existe un problema de escasez de agua potable que aumenta progresivamente con el paso de los años, lo que genera la creación de sistemas inteligentes para el monitoreo y gestión del agua. Para esto son buenos los sensores inteligentes con entrega de datos en tiempo real ya que permite un monitoreo

real y eficaz. En este artículo presentan el método de fusión de datos que se basa en la técnica de cuadrados mínimos para el diseño de un sensor inteligente para lo cual utilizaron un microcontrolador. Para esto emplearon tres variables diferentes para la comparación de resultados entre un consumo de agua real y el consumo que genera con un monitoreo inteligente. Este sensor tiene la tarea de reconocer si el consumo de agua es normal o presenta fugas, este realiza la comparación con valores reales de un área del Tecnológico de Chihuahua. En caso de que el sistema presente fugas o no se esté haciendo un buen consumo del agua toma acciones y emite alarmas. Este estudio fue realizado con el fin de advertir sobre el mal uso del agua y realizar una gestión eficaz de este recurso natural.

Otra aplicación que se le puede dar a los sensores inteligentes es el control de la turbidez, y de cloro en el agua. Este control es importante ya que las redes de distribución de agua suelen estar contaminadas por diversos factores que a larga perjudican la salud del individuo. Por este motivo las empresas que se dedican a la distribución de agua han optado por la implementación de dispositivos inteligentes que hagan un control completo de este recurso, lo que garantiza una detección inmediata de anomalías. En (Saab et al., 2017) presentan un estudio en el campus de la Universidad de Lille en el que utilizan sensores inteligentes para medir tres factores como turbidez, que el agua esté libre de cloro y el índice de refracción, también presentan el análisis de las señales que obtuvieron de la calidad de agua y la comparación de los valores reales con los simulados en el laboratorio.

Según (Wang, 2007) y (Lambrou et al., 2014) presentan un enfoque de control inteligente para el control de agua potable y bebible. Una de las propuestas se basa en nodos de sensores de bajo costo que benefician en el monitoreo y evaluación en tiempo real del sistema. El sistema de control se compone de sensores electroquímicos, ópticos dentro de una tubería, las ventajas que tiene implementar este sistema es el bajo costo, es liviana y a largo plazo genera una operación confiable con muy poca tendencia a fallos. Aparte realizan un estudio de investigación de mercado para saber que tan viable es la implementación de este sistema en industrias o en zonas donde no hay mucho control de la contaminación del agua. Después de las pruebas que realizaron demostraron el funcionamiento eficaz del sistema y comprobaron las ventajas que este ofrece.

Los sensores inteligentes también pueden ser usados para elaborar una red en la que cada nodo utiliza estos dispositivos para una acción específica, como por ejemplo una de las funciones que se le da en (Casans et al., 2006) es medir el pH de una solución, estos sensores también pueden trabajar en conjunto con un instrumento independiente como lo es un control remoto. Por otro lado, también se pueden utilizar estos sensores para medir el pH en el agua ya que es un parámetro que determina la calidad del agua, también permite que los animales acuáticos se desarrollen con normalidad en su ambiente, siempre que posea un excelente pH en el agua, para países como China en donde el monitoreo de la calidad del agua en tiempo real se vuelve más importante con el paso del tiempo por la incidencia de la acuicultura en este país. Los métodos que existen para medir el pH en el agua tienen ciertas desventajas por lo que determinaron que es viable el uso de un sensor inteligente que mida el pH en este medio, ya que al realizar estos dispositivos aporta al sistema con eficiencia, estabilidad y tiempo de respuesta más rápido(Wei et al., 2018).

2.2. Lenguaje Unificado de Modelado (UML)

En esta sección se describe acerca de UML, su historia, la concepción entre modelo - modelado UML, los diagramas que utiliza y las herramientas que se necesita para lenguaje unificado de modelado.

UML (Lenguaje Unificado de Modelado) fue creado por los denominados “tres amigos” es decir Grady Booch, Ivar Jacobson y James Rumbaugh, cada uno trabajaba en compañías distintas en los años 80 y a principios de los 90. Cada quien por su lado diseñó una metodología diferente para el análisis y diseño en base a figuras, estas propuestas prevalecieron sobre otras que presentaron demás personas. A mediados de los años 90 comenzaron a construir su metodología en conjunto, ya que comenzaron a intercambiar ideas para llegar a la conclusión de que podían realizar un solo lenguaje de modelado. Esto sucede en Rational Software Corporation, empresa en la cual trabajaba Booch, después en 1994 ingresó Rumbaugh y por último ingresó Jacobson en 1995. Al juntarse los tres amigos dieron origen a los anteproyectos de UML, que lo pusieron a prueba en la compañía, pero tuvieron que realizar algunos ajustes, para esto en 1997 produjeron la versión 1.0 de UML, después de eso quedó como un lenguaje de modelado estándar. Años más tarde los tres amigos crearon la versión 1.1 y en 1998 tuvieron dos versiones más. UML es la herramienta más utilizada por los desarrolladores de software a nivel mundial y se ha convertido en lo más importante por lo que su evolución continua de forma permanente, en la figura 2.1 se muestra los creadores de UML.(Schmuller, 2000)



Figura 2. 1: Creadores de UML(Lenguaje Unificado de Modelado)

Fuente: (Schmuller, 2000)

2.2.1. Comprensión de Modelado y UML

Un modelado es un conjunto de imágenes y texto que simbolizan algo, es decir un algoritmo de software. Cabe recalcar que no todo modelo representa un software, pero en este caso solo se va a estudiar los que representan un programa. Los modelos tienen diferentes ventajas una de ellas es la facilidad de dibujar formas o figuras con respecto a escribir un código de programación que explique lo mismo, otra ventaja es la flexibilidad que presentan al momento de hacer cambios o resolver problemas. En base a esto aparece UML que es un Lenguaje Unificado de Modelado que es un lenguaje de símbolos, que tiene todas las ventajas de los modelos, pero en caso de que se utilice UML para un software todo lo que tiene que ver con respecto al mismo debe estar realizados en base a este lenguaje de modelos. Por lo que es importante que se aprenda este lenguaje con todos los significados de los símbolos o imágenes que se planteen. Es importante saber de UML y utilizarlo ya que es un lenguaje de modelado estándar eso quiere decir que la mayoría del software que existen tienen como base estos gráficos y la gramática, que al momento de estudiarla ya es más comprensible el código que contiene el mismo y se realiza alguna modificación, también es más fácil y flexible a aceptar mejoras continuas en el código.

2.2.2. Diagramas de UML

Para crear modelos hay que saber los diferentes tipos que existen para utilizar el que mejor se adapte a lo que se vaya a realizar, a continuación, se describe cada uno de los tipos de diagramas que tiene UML.

2.2.2.1 Diagramas de Casos de Uso

Los diagramas de casos de uso son la descripción del sistema o de sus acciones desde el punto de vista del usuario, por tal motivo los símbolos primordiales son el actor o usuario y el óvalo donde se expone el caso de uso que realiza. La ventaja de este tipo de diagrama es que tiene la técnica de éxito y error lo que permite obtener los requerimientos o especificaciones del sistema por parte del usuario. Por tanto, los diagramas de casos de uso son una especie de lista que contiene las especificaciones del sistema y lo que debe hacer. En la figura 2.2 se muestra como se representa en un diagrama de casos de uso de un sistema bancario. (Schmuller, 2000) (Kimmel, 2008)

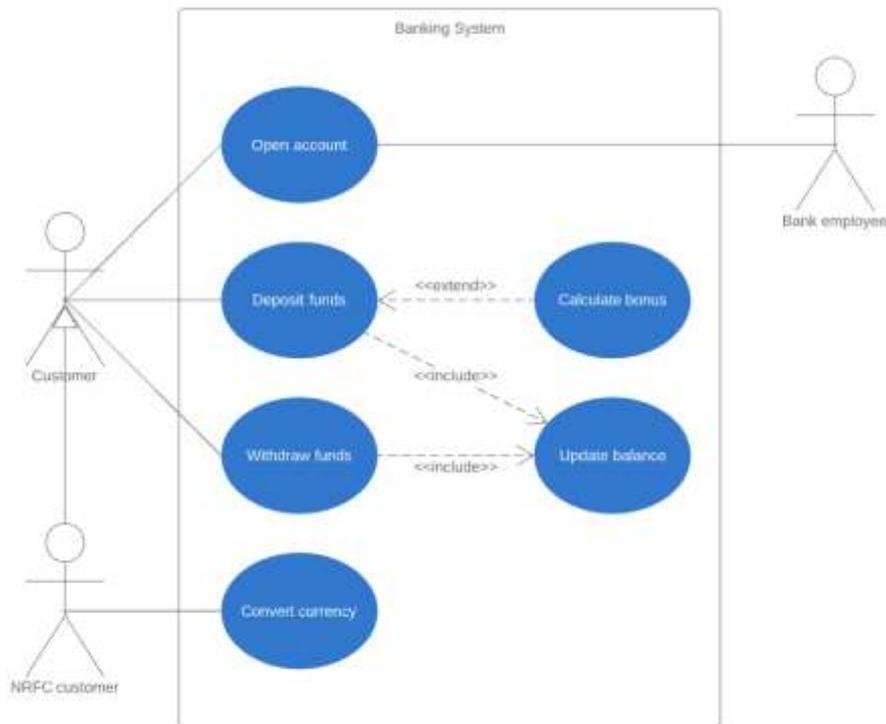


Figura 2. 2: Ejemplo del diagrama de casos de uso: sistema bancario
 Fuente: (Lucidchart, 2019)(Schmuller, 2000)

2.2.2.2 Diagramas de Estados

Estos diagramas muestran los cambios de estados de un componente, puede ser pequeño o de dimensiones grandes representando todo el sistema. Son más utilizados para modelado de sistemas embebidos y para protocolos que se vaya a implementar de un sistema. En la figura 2.3 se muestra un diagrama de estados de un individuo que busca su alimento para comérselo posteriormente. (Joyanes & Zahonero, 2014)

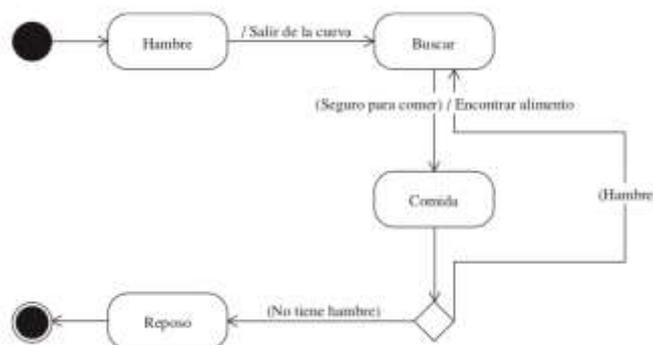


Figura 2. 3: Ejemplo de diagrama de estados, de un individuo busca su alimento para comerlo.

Fuente: (Kimmel, 2008)

2.2.2.3 Diagramas de Secuencia

Los diagramas de secuencia presentan de forma secuencial los elementos y como actúan; es importante el tipo y el orden del envío de los mensajes de los componentes mientras se ejecutan. Este tipo de diagramas muestra la interacción del sistema conforme al tiempo. La ventaja que poseen es que se puede modelar y analizar el sistema si funciona bajo las especificaciones dadas conforme pasa el tiempo o hay que corregir errores. En la figura 2.4 se muestra un ejemplo de diagramas de secuencia básico.(Joyanes & Zahonero, 2014)

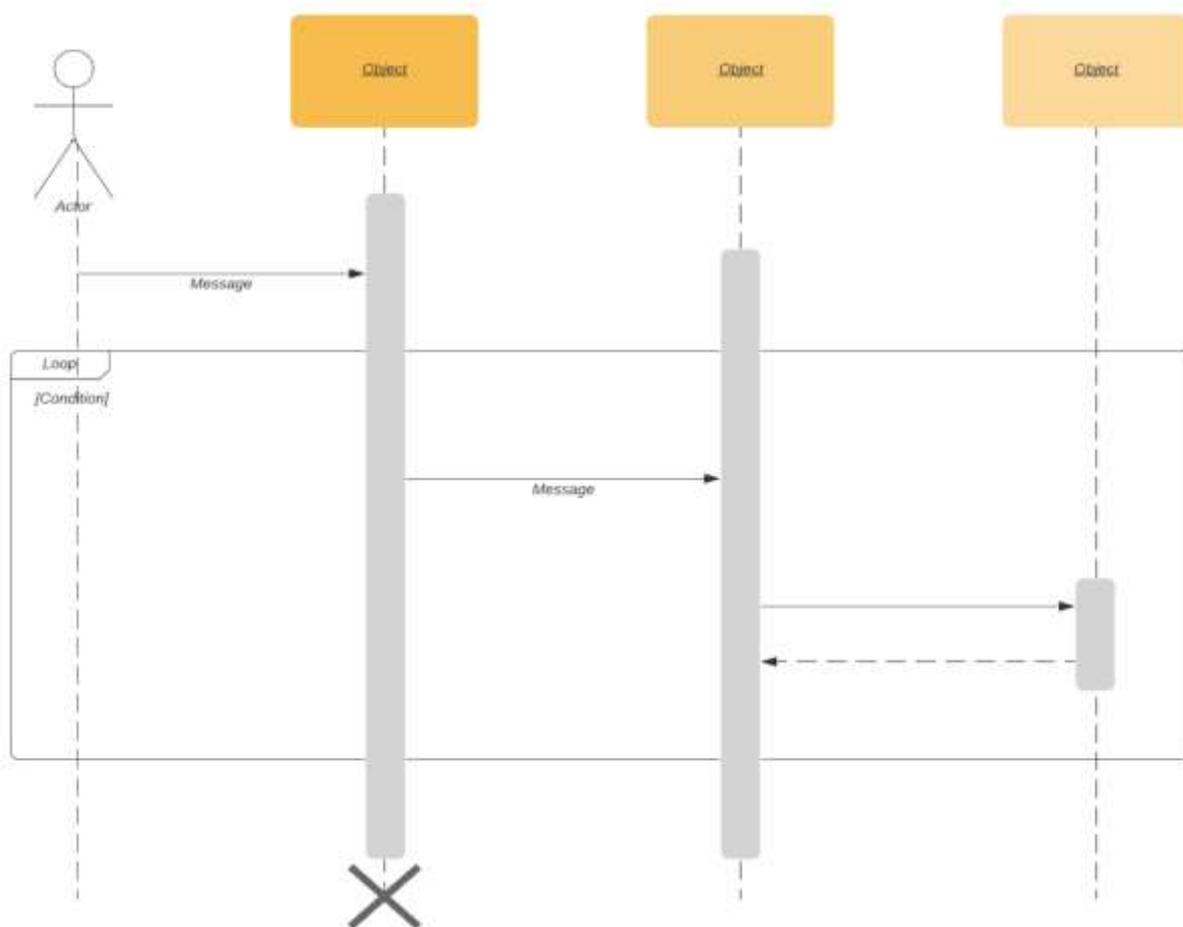


Figura 2. 4: Ejemplo de diagrama de secuencia básico.
Fuente: (Schmuller, 2000)(Lucidchart, 2019)

2.2.2.4 Diagramas de Actividades

Los diagramas de actividades representan la parte dinámica del sistema, son utilizados para modelar la función de un sistema en base al flujo del control de los objetos. En la figura 2.5 se muestra un ejemplo de diagrama de actividades. (Alarcón, 2000)

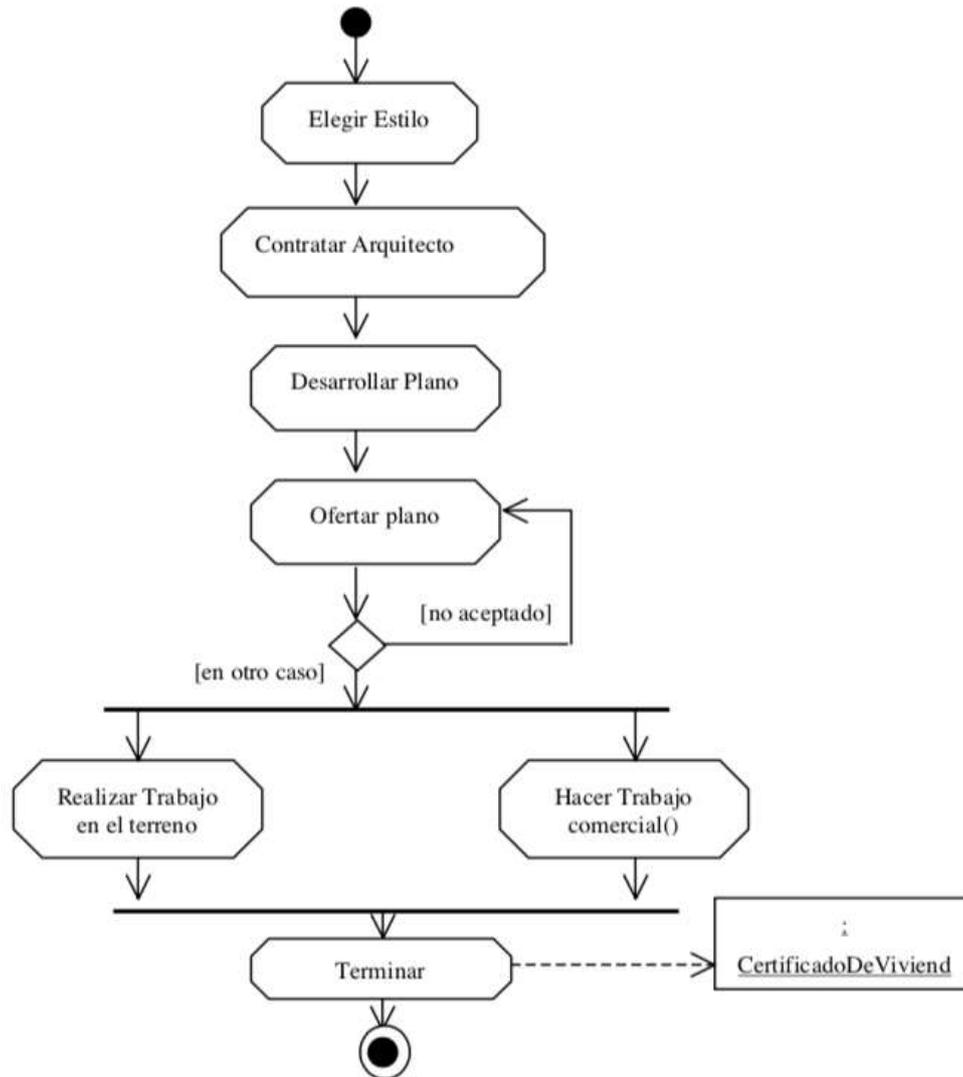


Figura 2. 5: Ejemplo de diagrama de Actividades de un sistema
Fuente: (Alarcón, 2000)

2.3. Sensores Inteligentes

En esta sección se describe la definición de sensor, sensor inteligente y por último sensores inteligentes de caudal y nivel, lo que serán utilizados para la elaboración de este proyecto de investigación.

2.3.1. Definición de Sensor

Sensor es el dispositivo que convierte las magnitudes físicas en valores que son medibles para la magnitud que representen como pueden ser: luz, caudal, caudal, nivel, posición, entre otros. Los sensores pueden ser eléctricos o mecánicos. En la figura 2.6 se muestra un sensor de caudal industrial (Rodríguez, 2016)



Figura 2. 6: Sensor de presión Industrial
Fuente: (Naylamp, 2019)

2.3.2. Definición de Sensor Inteligente

Los sensores inteligentes son un sistema en el que se combina el sensor ya sea analógico o digital con un microprocesador en una misma placa. Existen sensores que consisten en un solo chip como los sensores de temperatura, sensores de imagen, entre otros. Por otro lado, hay algunos sensores que necesitan más de un chip como los acelerómetros que utilizan dos chips en su sistema. (Gerard Meijer et al., 2014)

Según la definición dada por la IEEE 1451.2 los sensores inteligentes son los que brindan más funciones de las normales para proporcionar una precisa representación de la variable que monitorice. Estos sensores aportan con un nuevo paso en la evolución de la industria automatizada, en los ambientes domésticos, en los transportes, entre otros. Es importante tener en claro que los sensores siempre necesitarán la información del entorno que lo rodea y al ser inteligentes necesitan también de la información del propio funcionamiento. En la figura 2.7 se presenta la estructura de un sensor inteligente. (Morillo et al., 2013)

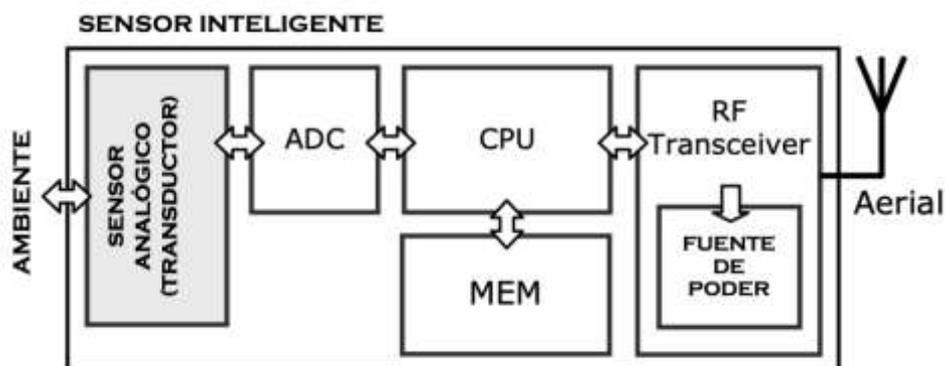


Figura 2. 7: Estructura de un sensor inteligente
Fuente: Morillo et al., (2013)

2.3.3. Funciones del Sensor Inteligente

Las funciones principales de los sensores inteligentes que permiten a sus herramientas de medición trabajar de forma óptima, son las que se detallan a continuación: (Pérez, 2017)

- Auto calibración, esto permite que el sensor no tenga que recurrir a la calibración que está contenida en la computadora central o general, de modo que el recuerde los valores iniciales siempre que lo necesite.
- Señal de salida, las señales de salida que envía este tipo de sensor serán digitales a diferencia de los sensores normales que emiten señales análogas.
- Reconocimiento del sensor, esto quiere decir que el sistema reconoce el sensor de inmediato.
- Comunicación, el sensor inteligente se puede comunicar con la computadora central, no solo enviarle información también puede recibirla y sucesivamente.
- Mejoras en el sistema, al ser inteligentes se pueden incluir cualquier tipo de mejoras que contenga el sistema, así se libera la computadora general de tareas secundarias.
- Autodiagnóstico, los sensores inteligentes tienen la capacidad de auto-diagnosticarse y emitir alarmas en caso de ser necesario.

2.4. Arquitecturas para sensores inteligentes

Las arquitecturas de los sistemas de sensado inteligente tienen como elemento principal los sensores, es por esto que requieren de flexibilidad y adaptabilidad en la misma. Un gran ejemplo para sistemas de sensado inteligente es el sistema humano sensorial ya que es muy avanzado y autónomo. La sensibilidad y selectividad que posee el humano hace que se adapten a cualquier entorno que lo rodee. En la figura 2.8 se muestra la estructura jerárquica que se asemeja a la del ser humano y es utilizada en sistemas complejos que ejecuten funciones avanzadas. Por tal motivo es utilizada para sistemas que requieran de arquitecturas distribuidas, que les ayude a tomar decisiones en momentos de fallo, que les recuerde sus rangos de operación establecidos previamente en las configuraciones iniciales del sistema. (Yamasaki, 1996)

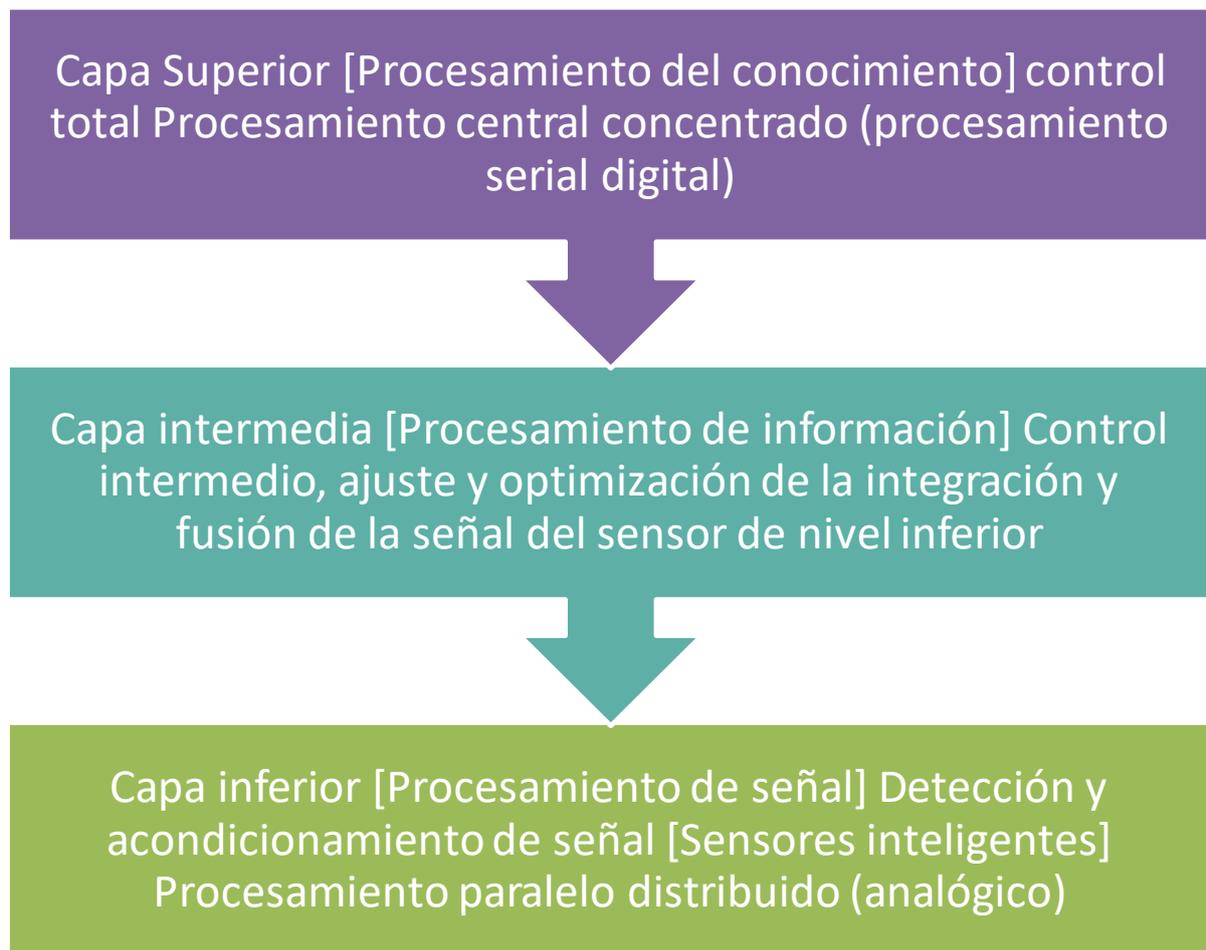


Figura 2. 8: Estructura de una arquitectura de sensor inteligente
Fuente: (Yamasaki, 1996)

Según (Pérez, 2017) la arquitectura de un sensor inteligente tiene tres partes fundamentales, las cuales se analizan a continuación:

1. **Procesamiento de señales:** esta etapa es la más importante ya que aumenta la señal y disminuye el efecto del ruido exterior, lo que permite aprovechar el rango del convertidor analógico – digital, antes de que envíe la señal al exterior.
2. **Control digital:** estos sensores deben ser compatibles con sistemas basados en microcontroladores, en el cual se van a ejecutar las funciones elementales de un sensor inteligente.
3. **Comunicación:** en esta parte es importante implementar un bus que permita interactuar con el controlador que maneja el sistema central, esta comunicación debe ser en tiempo real, con funciones que permitan recibir y enviar órdenes del sensor al controlador y viceversa.

2.4.1. Arquitectura de un sistema de sensado inteligente “Micrófono Inteligente”

Este sistema es una mezcla entre control y visualización de señales. Recibe los sonidos necesarios de una señal en diferentes ambientes con un mejorado S/N. La ubicación del sonido no la proporciona, pero la señal de sonido se basa en el objeto que la genere. El sistema está conformado por dos subsistemas uno de audio y el otro visual. El audio consta de múltiples micrófonos, un filtro de entrada múltiple y el sistema de autoaprendizaje. El sistema adaptativo tiene tres capas, la inferior que incluye los micrófonos y los convertidores A/D, la capa intermedia que consta de un filtro de entrada múltiple lineal, que combina las señales del sensor que generan una salida integrada, y en la capa superior cuenta con un ordenador que controla el rendimiento del filtro. En la figura 2.9 se muestra la arquitectura de este micrófono inteligente. (Yamasaki, 1996)

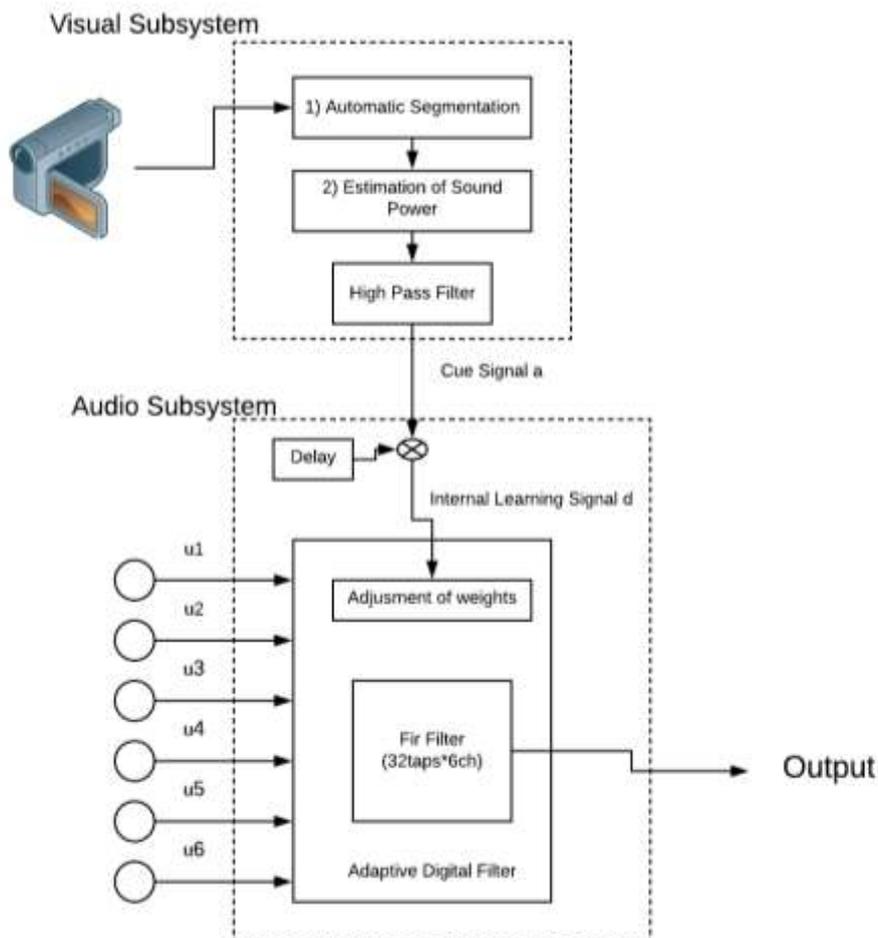


Figura 2. 9: Arquitectura de un micrófono inteligente
Fuente: (Yamasaki, 1996)

2.4.2. Arquitectura de un sistema de sensado inteligente y semi - inteligente

Por lo general la arquitectura de un sistema de un sensor inteligente tiene los siguientes aspectos(Mekid, 2006):

- Un elemento que comunique el mundo externo al sistema mediante una señal eléctrica, como por ejemplo voltaje o corriente, en base al entorno que rodea al sistema.
- Una interfaz para la adquisición y conversión de datos y señales en tiempo real. Estas señales que emite el sensor pasan por el procesador donde se transforman en datos digitales de tiempo discreto.
- La unidad de procesamiento que incluye el microcontrolador con memoria y el software que maneje el mismo. Está sección es la parte medular de la arquitectura ya que aquí se procesan las señales que recibe del sensor en tiempo real.
- El protocolo de comunicación que proporcione una comunicación bidireccional entre la unidad de procesamiento y el usuario. Se utilizan fibras ópticas, buses seriales, interface o pueden ser inalámbricas para comunicarse con el mundo exterior.
- Una fuente de energía para alimentar el sistema de sensado inteligente.

En la figura 2.10 se muestra una arquitectura de un sensor inteligente, es decir que integra todos los elementos en una única estructura.

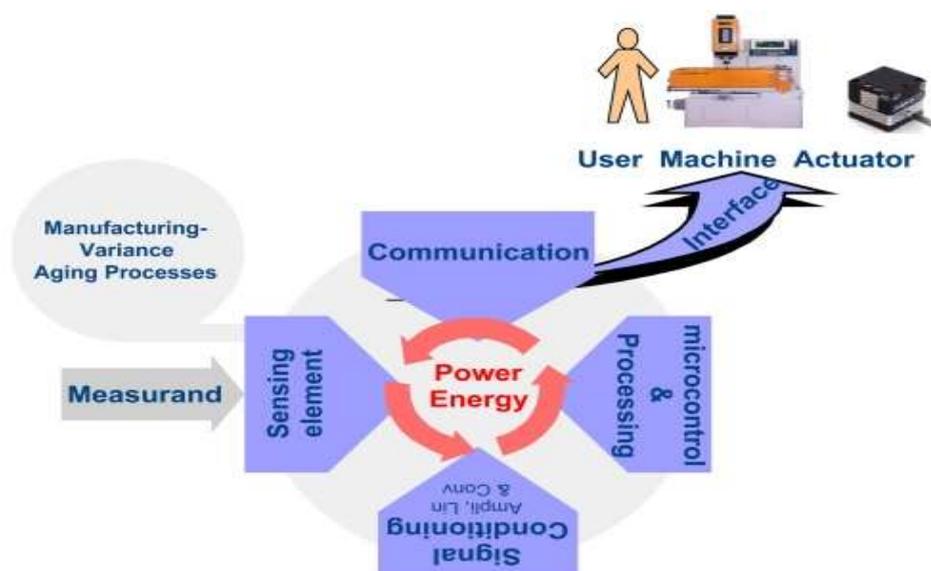


Figura 2. 10: Arquitectura de un sensor inteligente
Fuente: (Mekid, 2006)

En la figura 2.11 se muestra la arquitectura de un sensor semi- inteligente, es decir que separa la parte de la detección de señales y la interfaz en dos estructuras diferentes. (Mekid, 2006)

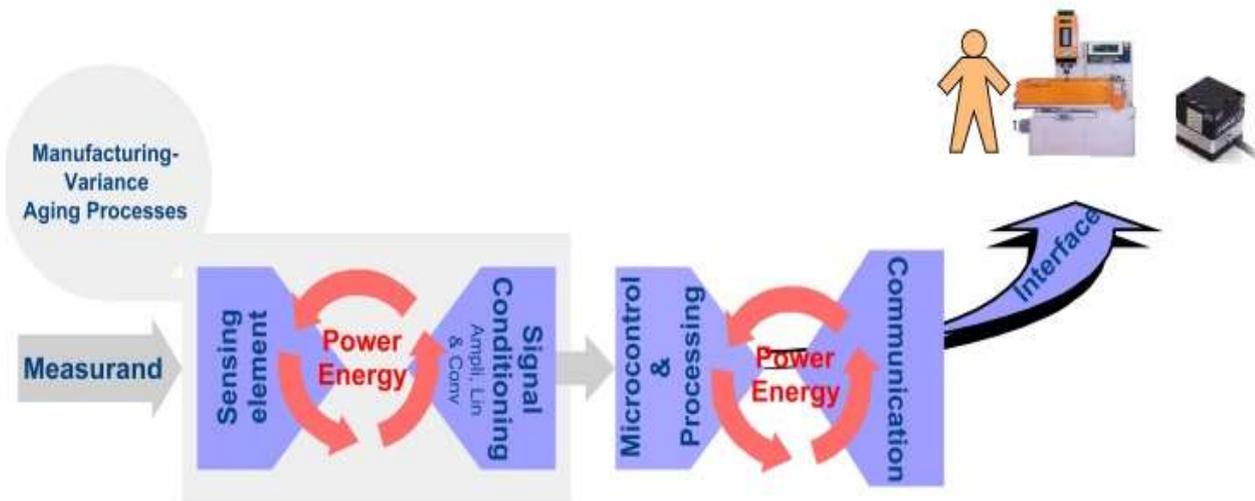


Figura 2. 11: Arquitectura de un sensor semi- inteligente
Fuente: (Mekid, 2006)

Un sistema de sensor inteligente está limitado en cuanto a la duración de la batería, la capacidad de la memoria, la velocidad de procesamiento de señales, para esto existen los sensores que contienen todo integrado que aportan con mayor precisión en las medidas que toma, son de menor tamaño, mejora el consumo de energía o también existen los sensores autoalimentados que su energía proviene de la vibración mecánica de la fuente. (Mekid, 2006)

Al fusionar el soporte bluetooth, WIFI, la fibra óptica o fieldbus con los sensores inteligentes hacen que la comunicación inalámbrica de estos sea más eficiente y óptima en la industria de procesos.

Las ventajas de los sensores inteligentes son:

- Reducción en el tiempo de inactividad
- Sistema robusto, tolerante a fallas
- Adaptabilidad para la auto calibración y compensación del sistema
- Confiabilidad
- Mapeo de sensores maestro/ esclavo
- Menor costo
- Menor mantenimiento

2.4.3. Arquitectura de sensor inteligente para dispositivos móviles

Esta arquitectura propone cuatro entidades de hardware: dispositivo terminal, nodo del sensor local, sensor RFID y el servidor. El dispositivo terminal es un celular con adecuaciones adicionales para proporcionar una funcionalidad del tipo MIMOSA. El celular actúa como un enlace entre el entorno de los sensores y el internet. MIMOSA proporciona Bluetooth, Wibree y RFID para la lectura de los sensores, estas tecnologías utilizan módulo RF analógico.

El protocolo es implementado en un FPGA con el terminal MIMOSA, es decir el SPI que es un bus de comunicación con líneas de interrupción adicionales para el sensor local. (Jantunen & Huuskonen, 2008). En la figura 2.12 se muestra la arquitectura para dispositivos móviles.

Al integrar los dispositivos móviles, los protocolos de comunicación con internet o inalámbricos con sistemas que manejen sensores inteligentes aporta con muchas ventajas, ya que proporciona un control eficiente, en tiempo real y más cerca del usuario. (Jantunen & Huuskonen, 2008)

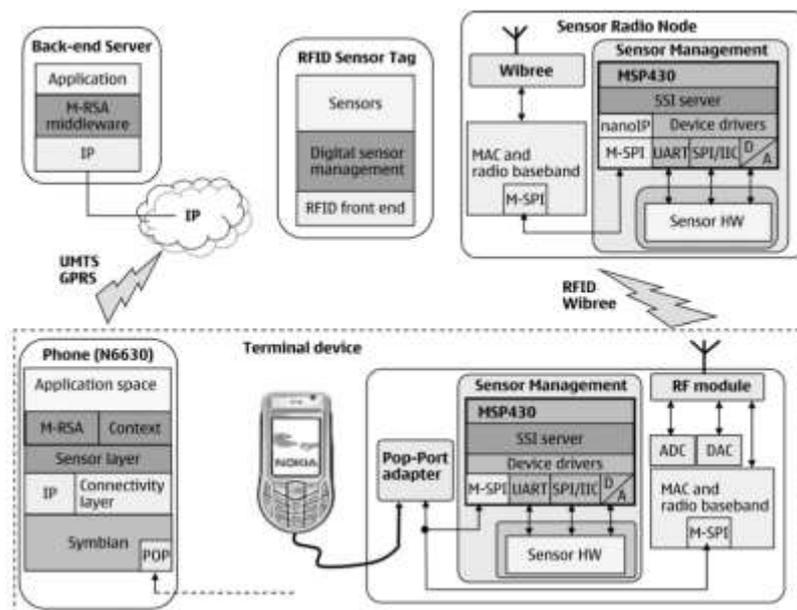


Figura 2. 12: Arquitectura de sensor inteligente para dispositivos móviles
Fuente: (Jantunen & Huuskonen, 2008)

2.4.4. Arquitecturas deliberativas

En la arquitectura deliberativa se representa de forma muy explícita el comportamiento de los agentes inteligentes, mediante algún modelo que ya este establecido previamente. Los agentes toman las decisiones a través del razonamiento

lógico, es decir que los procesos que cumplen estos sensores son procesos lógicos. Un ejemplo de agente deliberativo es BDI denominado agente racional ya que presenta actitudes mentales en su desarrollo. A continuación, se detalla que es lo que contiene BDI (Beliefs, Desires, Intentions).

2.4.4.1 Agente Deliberativo BDI

Según (Aguilar, 2013) el agente deliberativo BDI (Beliefs, Desires, Intentions) posee Creencias, Deseos e Intenciones, las cuales se describen a continuación:

- **Creencias**, es todo lo que conoce el agente sobre el entorno, puede ser sobre sí mismo o de otro agente. Esto quiere decir toda la información que se tenga del exterior, o lo que el agente cree que es cierto o verdadero del exterior. Esta información se almacena en una base de datos o de conocimientos, que con el tiempo puede variar, en algunos casos lo verdadero se convierte en falso.
- **Deseos**, representan los objetivos que debe cumplir el agente, como pueden ser estados que quiera alcanzar el agente o eventos que este quiera lograr.
- **Intenciones**, aquí es donde se encuentra el estado deliberativo del agente, ya que las intenciones son lo que el agente decide realizar o hacer. Otra forma de expresarlo es que son metas que el agente se traza para cumplir los objetivos propuestos en los deseos.

Este tipo de agente deliberativo es más utilizado en el desarrollo de software que se basen en el uso de nuevos procedimientos o tipos de datos que desee captar. Convirtiendo el concepto de intención en una forma más agradable de describir, explicar y especificar como se comporta un sistema complejo. (Aguilar, 2013)



Figura 2. 13: Arquitectura de agente deliberativo BDI
Fuente: (Luis et al., 2016)

2.5. Sensor de Caudal

Caudal es una variable que relaciona el volumen de un líquido o gas con relación en el tiempo, La ecuación 1 muestra la fórmula base del caudal: (Cosco, 2013)

$$Caudal = \frac{Volumen}{Tiempo} \quad (1)$$

Las medidas en las que se expresa o representa el caudal son las siguientes:

- Litros/minuto
- Litros/segundo
- Galones/minuto
- Barriles/hora

El caudal indica que rápido se mueve un fluido en un sistema de control industrial. Se expresa de tres formas diferentes: (Cosco, 2013)

- Q= Caudal volumétrico: muestra el volumen de un fluido que pasa por un punto por una unidad de tiempo [m³/sg].
- Q_m= Caudal másico: se expresa en unidades de masa [Kg/s].
- Q_v=Velocidad del Caudal: está dada en [m/s].

2.5.1. Pasos para elegir un sensor de Caudal adecuado

Los sensores de caudal, son varios, pero al momento de escoger el sensor ideal que se ajuste al sistema, se debe seguir los siguientes pasos, para no cometer errores en el momento de la compra del sensor. (Cosco, 2013)

1. Se debe saber si la información que debe arrojar el sensor debe ser continua o totalizada y si es necesaria de forma remota o local.
2. Es importante saber las características del fluido a controlar y las características que va a tener el caudal. Es decir, los rangos de operación y la frecuencia con la que va a trabajar.
3. Los rangos de temperatura y presión son importantes que se sepan al momento de elegir un caudalímetro.
4. La tubería y el lugar de instalación, ya que es importante saber si va a existir vibraciones o campos magnéticos que interfieran en la operación del sensor.

2.5.2. Tipos de Sensor de Caudal

- Caudalímetros de área variable de gases y líquidos.

- Caudalímetros con pistón y muelles de gases y líquidos
- Caudalímetro utilizado para gas másico
- Caudalímetro ultrasónicos (no intrusivo o Doppler) solo para líquidos
- Caudalímetro de turbina
- Caudalímetro de paletas
- Caudalímetro de desplazamiento positivo
- Caudalímetros Vortex
- Tubos pitot o sensores de presión diferencial para líquidos y gases
- Caudalímetro magnético para líquidos conductivos
- Anemómetros para la medición del caudal de aire

2.6. Sensor de Nivel

Los sensores de nivel de fluidos trabajan con instrumentos como flotadores que proporcionan medidas directas, o utilizan instrumentos que generan medidas indirectas como son los sensores de membrana. También utilizan las características de los fluidos sobre los cuales van a trabajar para realizar las mediciones de forma eficiente, por eso existen los capacitivos, conductivos, resistivos, de radiación, entre otros. El nivel (h) es usado para producir volumen (V), a través de un área específica (A) del recipiente en el que se encuentre como se muestra en la figura 2.14. (Cosco, 2013)

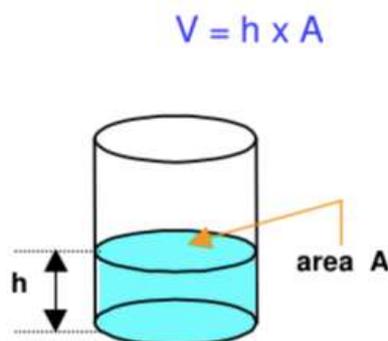


Figura 2. 14: Diagrama que representa el volumen de un fluido en un recipiente
Fuente: (Cosco, 2013)

El nivel también se lo utiliza para determinar la masa (m), con base en la fórmula de la densidad.

Véase en la ecuación 2 se muestra la fórmula de la densidad (ρ):

$$\rho = \frac{m}{V} \quad (2)$$

Al despejar la fórmula anterior, se tiene la ecuación 3 para obtener la masa en base al nivel:

$$m = V * \rho \quad (3)$$

El volumen de la ecuación se lo halla en la ecuación 4:

$$V = h * A \quad (4)$$

Finalmente, en la Ecuación 2 se puede hallar el valor de la masa con base al nivel.

2.6.1. Clasificación de los sensores de Nivel

Los sensores de nivel varían según (Cosco, 2013) :

- **La medición a realizar:**
 - Medida directa
 - Basados en la presión hidrostática
 - Basados en el desplazamiento
 - Basados en la emisión de rayos Gamma
 - Ionización Química
 - Transductores: Conductor y Potencial Electrolítico
- **El tipo de material a sensar:**
 - Para líquidos
 - Para sólidos
- **La medida a realizar:**
 - De Punto Fijo
 - De Nivel Continuo

2.6.2. Funcionamiento de los Sensores de Nivel

El funcionamiento de los sensores se divide en los siguientes grupos:

- Sensores de Medida Directa, se divide en dos diferentes sensores según el medidor ya sea de sonda o de tubo de vidrio. La diferencia consiste en que el

medidor de sonda es una varilla con una longitud adecuada para introducirla en el reservorio que va a ser controlado, por otro lado, está el de tubo de vidrio está conectado en sus extremos a bloques metálicos que sirven para impedir la fuga del líquido en caso de rotura del cristal. En la figura 2.15 se presenta los sensores de Medida Directa. (Cosco, 2013)

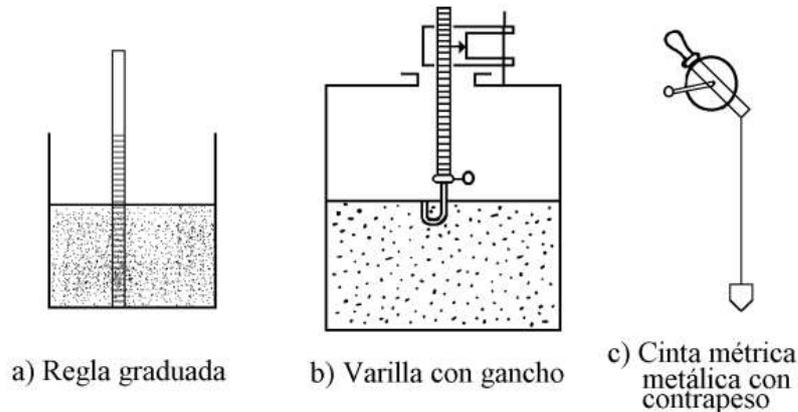


Figura 2. 15: Sensores de Medida Directa
Fuente: (Cárdenas, 2009)

- Sensores de Flotador, estos equipos poseen un flotador que se lo utiliza para medir la fuerza de gravedad versus la fuerza que se opone al líquido. Está compuesto por un brazo rígido que indica el nivel por el ángulo que se forma, para interpretar la posición se utiliza un potenciómetro. En la figura 2.16 se muestra los sensores de Flotador (Cosco, 2013).

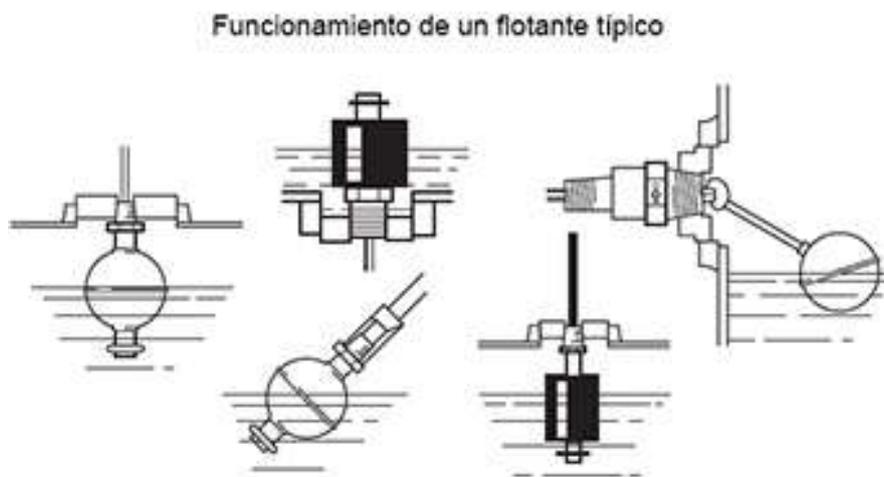


Figura 2. 16: Sensores de Flotador
Fuente: (Cárdenas, 2009)

2.6.3. Unidades Utilizadas en Nivel de Líquidos

La unidad más utilizada para la medición de Nivel según el Sistema Internacional de Unidades es el metro (m), pero en la industria se utilizan diferentes unidades como son: pie (ft), pulgada (in), centímetro (cm). En la tabla 2.1 se muestran estas unidades con sus factores de conversión (Cárdenas, 2009).

Tabla 2. 1: Unidades utilizadas en la Medición de Nivel (Cárdenas, 2009)

Unidades	Pies	Pulgadas	Metro	Centímetro
ft	1	12	0.3048	30.48
in	0.0833	1	0.0254	2.54
m	3.2808	39.37	1	100
cm	0.0328	0.3937	0.01	1

2.7. Protocolo MODBUS

MODBUS es un protocolo que define a un nivel de aplicación, sin embargo, no cumple con todas las capas de un modelo OSI, solo contiene los niveles 2 y 7 que son usados para MODBUS en serie y MODBUS TCP/IP. En este protocolo no se especifica cual es la capa física, sin embargo, lo que más se utiliza es un soporte metálico que tiene una velocidad de transmisión de 75 a 19200 baudios. La topología de MODBUS es maestro – esclavo, donde pueden existir uno o más esclavos que van a ejecutar las órdenes del maestro. El método MODBUS serie tiene dos modos de interpretación sea ASCII o RTU que depende del formato en como este codificada la trama ya sea binario o en código ASCII (Barastegui & Castaño, 2017).

2.7.1. Tipos de datos que permite MODBUS

En base al tamaño de la información que se envía se clasifican los datos en dos tipos: bits individuales y registro de 2 bytes. Los bits que individuales generalmente son para entradas o salidas digitales y los que llevan registro son para variables de gran tamaño. En la tabla 2.2 se encuentran los tipos de datos que se tiene a disponibilidad del usuario:

Tabla 2. 2: Datos que se utilizan en MODBUS (Barastegui & Castaño, 2017)

Tipo de Objeto	Acceso	Tamaño
Discrete input	Only read	1bit
Coil	Read and write	1bit
Input register	Only read	16 bits
Holding register	Read and write	16 bits

- Discrete input, se genera por una entrada o salida (Barastegui & Castaño, 2017).
- Coil, se genera por un programa de aplicación (Barastegui & Castaño, 2017).
- Input register, se genera por una entrada o salida (Barastegui & Castaño, 2017).
- Holding register, se genera por un programa de aplicación (Barastegui & Castaño, 2017).

2.7.2. Estados del Maestro MODBUS

Para el maestro MODBUS se tiene 5 estados, que son los siguientes:

- IDLE, el estado inicial, donde solo se envían solicitudes desde este estado y no saldrá al menos que envíe una solicitud de modo broadcast que vaya dirigida a todos los esclavos, o puede ser unicast en caso que solo vaya dirigida a un esclavo (Barastegui & Castaño, 2017).
- TURNAROUND, en caso de que se envíe una solicitud de modo broadcast se activa este estado y en caso de permanecer se activa “Wait turnaround” esto es para los esclavos a que tengan un tiempo prudente de atender la solicitud antes que regresen a sus estado inicial en el cual es la oportunidad de lanzar una nueva orden (Barastegui & Castaño, 2017).
- PROCESA ERROR/RESPUESTA, esto ocurre en caso que se envíe una orden unicast y se activa un temporizador que pasa al estado “Wait response”, donde se sale si y solo si el temporizador expira, posterior pasa al estado “Processing error”, si se recibe una respuesta que detenga el temporizador pasa al estado “Processing response” (Barastegui & Castaño, 2017).
- Finalmente, una vez que procesa la respuesta regresa a su estado inicial, si y solo si la respuesta fue correcta, caso contrario pasa a un estado “Processing error”, con el que regresa al estado inicial de inmediato para realizar un nuevo intento. El número de reintentos válidos depende en la configuración que el maestro posea (Barastegui & Castaño, 2017).

Según (Barastegui & Castaño, 2017) existen errores en la trama, los cuales consisten en:

- Revisión de paridad de cada uno de los caracteres
- Revisión de redundancia en la trama completa

A continuación, en la figura 2.17 se muestra el diagrama con los estados que contiene el maestro MODBUS, el cual se convierte en el principal que gobierna en todo el sistema. El maestro siempre va a emitir las órdenes que sean necesarias a los esclavos del protocolo para que se ejecuten las acciones que ya estén preestablecidas del sistema general.

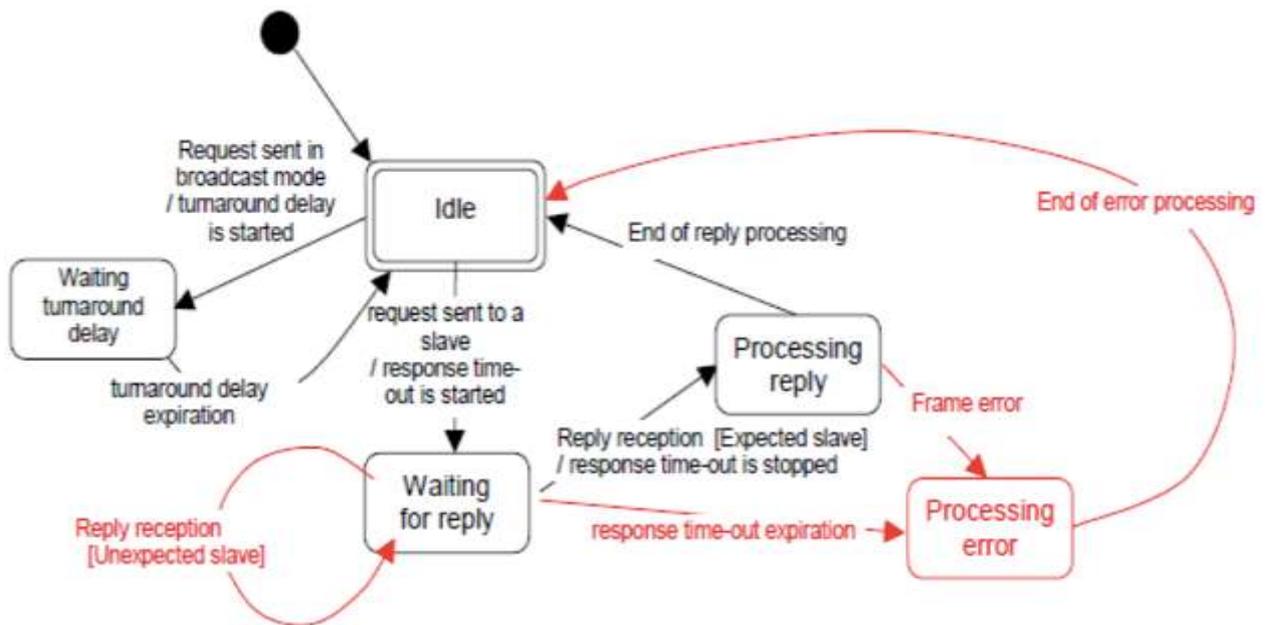


Figura 2. 17: Diagrama de estados del maestro MODBUS
Fuente: (Barastegui & Castaño, 2017)

2.7.3. Estados del Esclavo MODBUS

Para el esclavo MODBUS se tiene 4 estados, que son los siguientes:

- IDLE, el estado inicial, si se recibe solicitudes pasa a un estado de “Verificando orden”, lo que se comprueba si la solicitud no tiene errores. Si la solicitud está en orden se pasa al estado “Procesando la acción requerida”. En caso de no estar correcta la solicitud depende de dos factores, que el error este alojado en los datos de la solicitud o que la trama sea la incorrecta.
- FORMATTING ERROR REPLAY, si el error se encuentra en los datos de la solicitud, pasa a este estado, lo que hace es mandar una respuesta donde

notifica el error y este se envía al maestro, después pasa por el estado inicial o de reposo.

- Si la trama que recibe es totalmente errónea o no va dirigida a ese esclavo, la ignora y regresa a su estado inicial.
- Para salir del estado “Processing required action” existen tres formas:
 - Si pasa al estado “Formatting normal reply” en caso de que se haya completado la acción de la solicitud sin error alguno, siempre que haya sido recibida en modo unicast.
 - Para comprobar si hubo algún error al momento de realizar la acción que solicite, pasa al estado “Formatting error reply”.
 - En caso de que haya recibido la orden completamente correcta pasa al estado de reposo, en espera de otra acción vía broadcast.

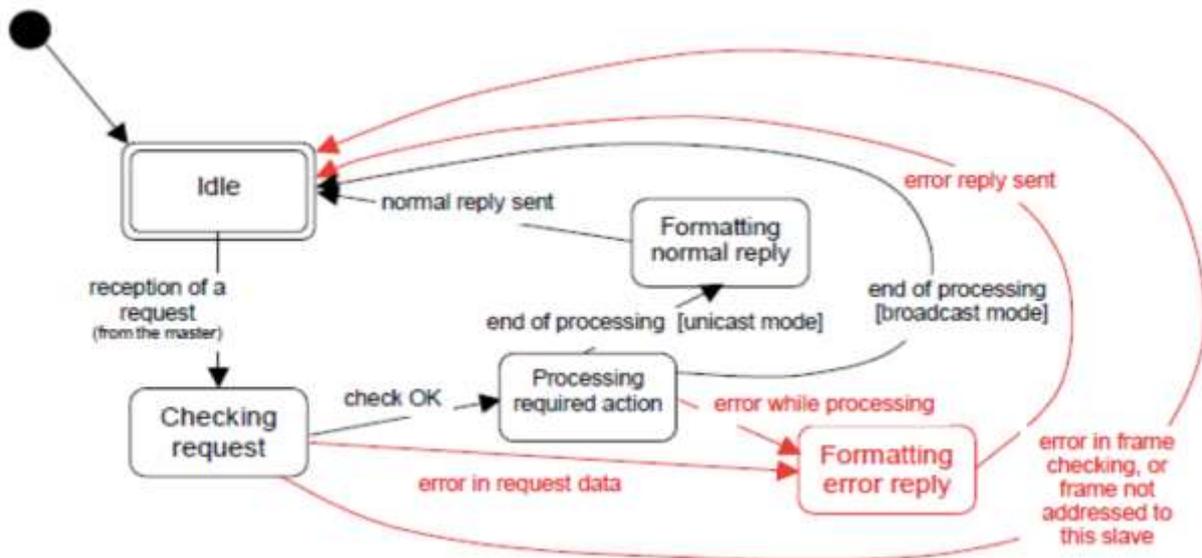


Figura 2. 18: Diagrama de estados del maestro MODBUS
Fuente: (Barastegui & Castaño, 2017)

2.7.4. Modo de Transmisión RTU

El modo de transmisión RTU utiliza bits de inicio y stop para transmitir un byte para que el receptor tenga conocimiento de donde comienza y termina la información. En la figura 2.19 se aprecia que existen dos modos que depende de si está comprobada o no la paridad. La paridad es un bit que es “1” de forma que el número total de bits es “1” en el byte para que coincida con el modo de paridad anteriormente electo. Los bits se emiten de izquierda a derecha, desde el menos significativo hasta el más significativo. En caso de existir no paridad se sustituye el bit por otro de stop (Barastegui & Castaño, 2017).

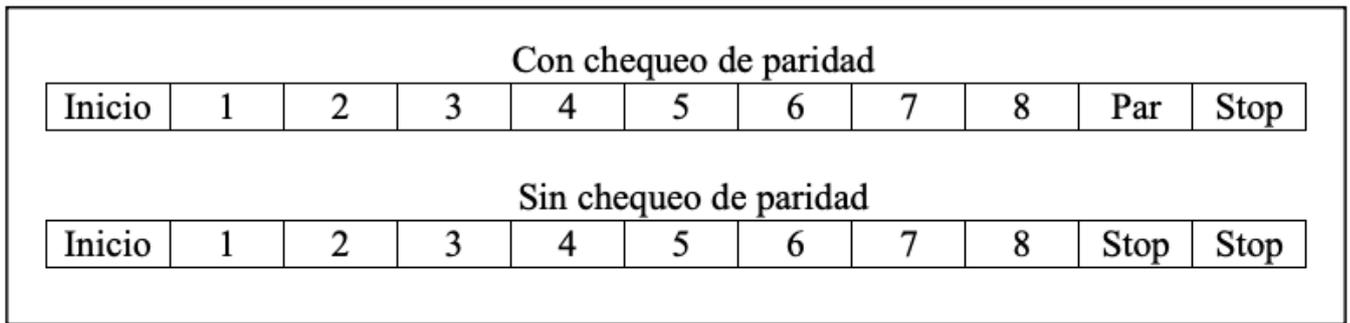


Figura 2. 19: Secuencia del chequeo de paridad en RTU
Fuente: (Barastegui & Castaño, 2017)

2.7.5. Modo de Transmisión ASCII

En ASCII también existen dos modos que dependen si se ha comprobado la paridad o no. El modo que viene por defecto de MODBUS es con paridad par, donde los modos de paridad impar son opcionales. Los bits son transmitidos en este orden: de izquierda a derecha, desde el bit menos significativo llegando al más significativo. Así mismo en el modo de no paridad se sustituye por otro bit de stop. En la imagen 2.20 se observa esta secuencia de paridad y no paridad en ASCII (Barastegui & Castaño, 2017).

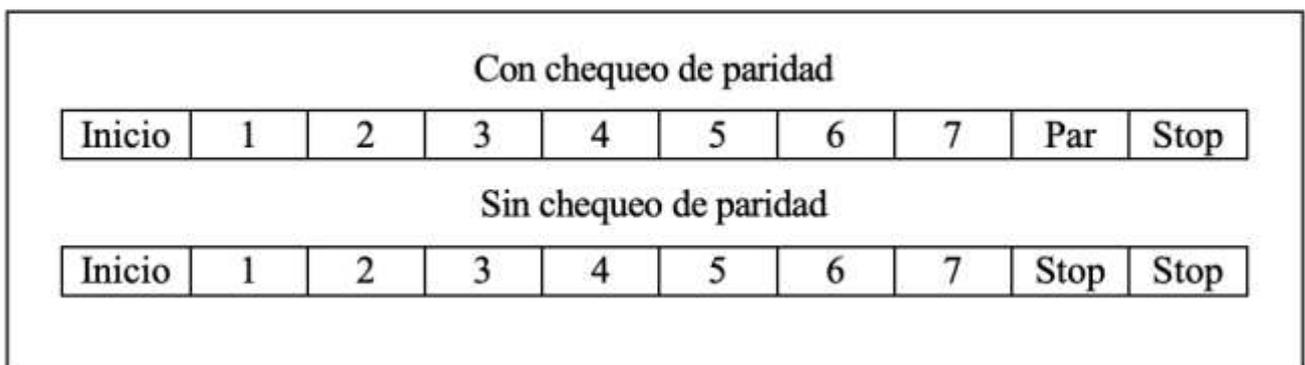


Figura 2. 20: Secuencia del chequeo de paridad en ASCII
Fuente: (Barastegui & Castaño, 2017)

2.8. Microcontroladores

Los microcontroladores tienen diversas aplicaciones, ya sea para la automatización, en equipos de comunicación, en telefonía, en la electrónica, en medicina, en diversos juguetes, electrodomésticos y en la industria de control de procesos. Estos dispositivos se crearon para ser utilizados en aplicaciones específicas o en donde realizan el menor número de tareas, a un menor costo de implementación.

Para iniciar la ejecución de las tareas que va a realizar se debe cargar un programa previamente codificado en el lenguaje que soporte el mismo, que se almacenara en la memoria para interactuar con el mundo exterior a través de las entradas y salidas que posee. En caso que sean muchas tareas las que deba realizar el microcontrolador se debe usar más para repartirlas en forma equitativa y evitar la sobrecarga de trabajo en el mismo. (Valdés & Pallás, 2007)

En la figura 2.21 se muestra el diagrama de bloques general para un microcontrolador.

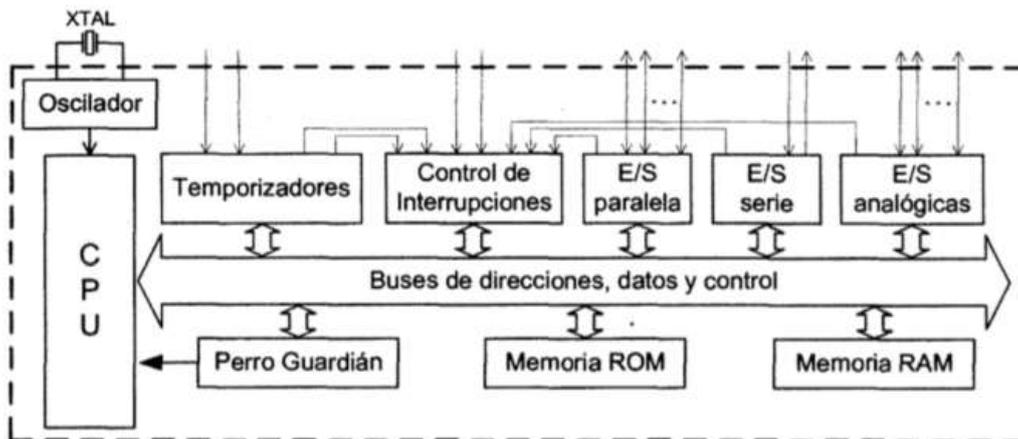


Figura 2. 21: Diagrama de bloques de un microcontrolador
Fuente: (Valdés & Pallás, 2007)

2.8.1. Componentes de un microcontrolador

Un microcontrolador se compone de los siguientes elementos:

- Un CPU o microprocesador.
- Una memoria RAM que almacena los datos de forma temporal.
- Memorias de lectura como: EPROM, ROM, FLASH o EEPROM para el almacenamiento del programa que contiene las tareas a ejecutar o datos que se desee almacenar.
- Puertos de entrada y salida para establecer las comunicaciones con el mundo exterior.
- Módulos para periféricos como: Temporizadores, convertidores de señales analógicas a digitales y viceversa.
- Generador de pulsos de reloj para sincronizar el funcionamiento del sistema. (Inzunza, 2016)

En la figura 2.22 se muestran los componentes que contiene un microcontrolador.

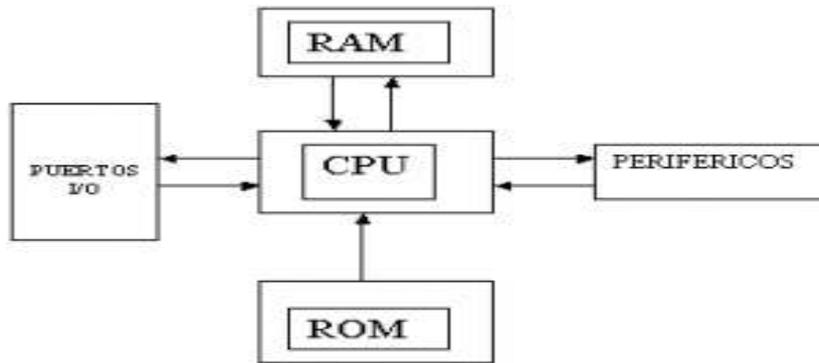


Figura 2. 22: Componentes de un microcontrolador
Fuente: (Inzunza, 2016)

2.8.2. Arquitectura von Neumann y Harvard

Las arquitecturas que se presentan en esta sección son modelos generalizados del hardware de los ordenadores, pero presentan arreglos diferentes con el problema de conexión del microprocesador con la memoria, también ofrece soluciones a la falta de organización entre la memoria y el espacio donde se almacenan las instrucciones.

2.8.2.1 Arquitectura von Neumann

Esta arquitectura propone el diseño de un computador digital conformado por la unidad de procesamiento que contiene la unidad aritmético lógica, los registros que genera el procesador, la unidad de control que contiene las instrucciones del programa, la memoria para almacenar los datos e instrucciones, almacenamiento externo y los puertos de entrada/salida. Comparten un bus común de comunicación, que es conocido como el cuello de botella de Von Neumann, lo que limita el rendimiento del sistema. Esta arquitectura es mucha más fácil que la Harvard. (Valdés & Pallás, 2007)

En la figura 2.23 se muestra el diagrama general de la arquitectura Von Neumann.

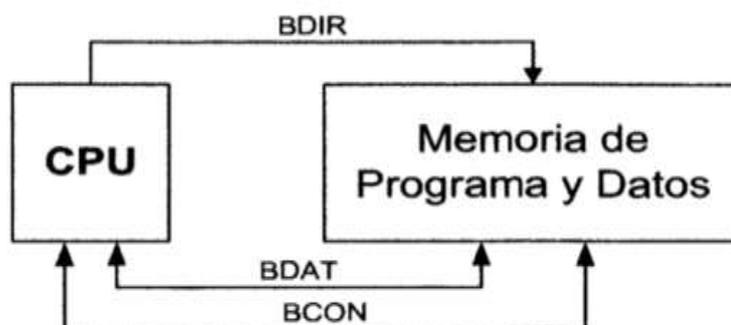


Figura 2. 23: Diagrama de la arquitectura Von Neumann
Fuente: (Valdés & Pallás, 2007)

2.8.2.2 Arquitectura Harvard

A diferencia de la arquitectura Von Neumann esta tiene memorias separadas para las instrucciones. La memoria de programa tiene su propio bus de direcciones, propio bus de datos y el propio bus de control. Así como la memoria de datos también posee sus propios buses de datos, direcciones y control, que son independientes de los buses de la memoria del programa. La memoria de datos permite leer y escribir a diferencia de la otra. (Valdés & Pallás, 2007)

En la figura 2.24 se muestra el diagrama general de la arquitectura Harvard.

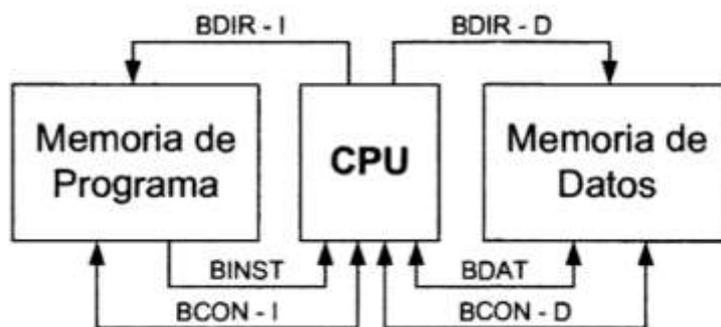


Figura 2. 24: Diagrama de la arquitectura Harvard
Fuente: (Valdés & Pallás, 2007)

2.8.3 Arquitecturas RISC y CISC

Las arquitecturas CISC y RISC son modelos de ordenadores que recaen en la CPU, un ordenador CISC tiene instrucciones complejas a diferencia de RISC que contiene instrucciones reducidas.

2.8.3.1 Arquitectura CISC

La arquitectura CISC es muy compleja, posee repertorios de instrucciones que realicen acciones muy simples como mover un dato a un lado específico junto a otras que realizan actividades más complejas como mover una cadena de datos a la memoria. Al momento en que las instrucciones tenían diferentes longitudes, provocó que los direccionamientos sean más elaborados. Este aumento en la complejidad de la arquitectura se mostró en un hardware muy complejo, por lo que era necesario designar un espacio grande al circuito integrado para la decodificación y posterior ejecución de las instrucciones. (Valdés & Pallás, 2007) En la figura 2.25 se muestra el diagrama general de la arquitectura CISC.

CISC

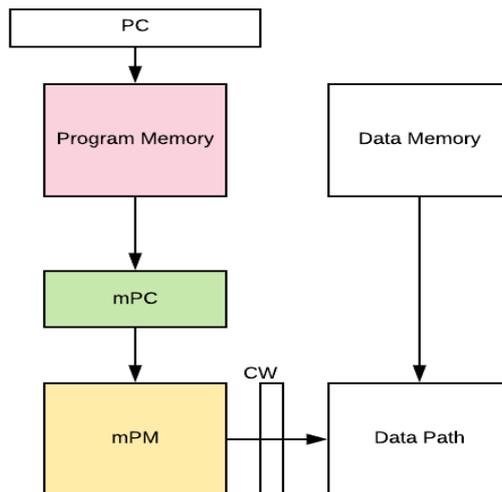


Figura 2. 25: Diagrama de la arquitectura CISC
Fuente: (Microcontrollers, 2019)

2.8.3.2 Arquitectura RISC

La arquitectura RISC tiene un repertorio de instrucciones corto y sencillas. Cada una de las instrucciones pueden realizar operaciones simples, como mover un dato a la memoria, pero a una alta velocidad con la misma longitud. Los direccionamientos de datos son muy pocos y se aplican a todas las celdas de la memoria de datos. No posee una CPU compleja lo que permite aumentar la frecuencia del oscilador del CPU, por lo tanto, se incrementa la velocidad de las instrucciones. Al tener menos transistores es una arquitectura más económica al momento de diseñarla o producirla. Los PIC están formados por arquitecturas RISC. (Valdés & Pallás, 2007)

En la figura 2.26 se presenta el diagrama general de la arquitectura RISC.

RISC

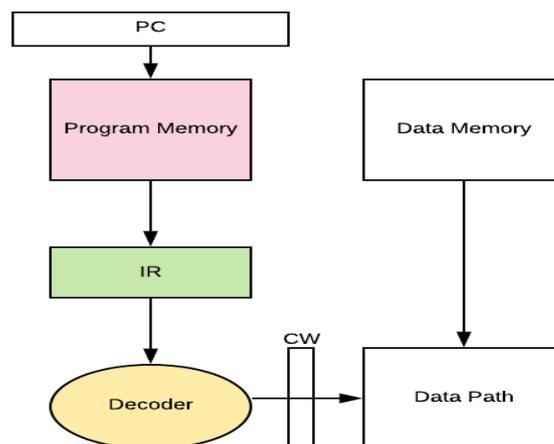


Figura 2. 26: Diagrama de la arquitectura RISC
Fuente: (Microcontrollers, 2019)

2.9 Redes de Petri

Carl Adam Petri fue el creador de las redes que llevan su nombre que son una herramienta gráfica, por los que se utilizan para representar la dinámica de eventos, la evolución en paralelo, la sincronización, entre otros fenómenos. Aparecen con frecuencia en sistemas de producción, en diversos protocolos de comunicación, en computadoras y redes compuestas por las mismas, en software que sean de tiempo real, entre otros. Las redes de Petri son utilizadas para representar de forma gráfica a los sistemas de eventos discretos, con diferentes estados y transiciones. De las cuales se obtienen un resultado de sus invariantes, el gráfico de alcance y cobertura de la red, los resultados de simulación y el tiempo de respuesta que posee el sistema.

En la figura 2.27 se muestra el creador de las redes de Petri.(Cohen, 2001)



Figura 2. 27: Carl Adam Petri creador de las redes de Petri
Fuente: (Cohen, 2001)

2.9.2 Elementos de las redes de Petri

Las redes de Petri (RdP) generalmente están compuestas por dos elementos fundamentales:

- Elementos estáticos: son los lugares o estaciones que se representan con una circunferencia, y las transiciones del sistema representado por una línea recta, que al unirse forman la red. Los arcos no pueden unir dos elementos que sean del mismo tipo, estos siempre tienen un número asociado que equivale al peso, en caso de no tenerlo su peso es la unidad.
- Elementos dinámicos: son las marcas representadas por un punto, la RdP puede contener un número indeterminado de estas.(Torres, 2007)

Los elementos de la red de Petri son:

- Círculos de Tamaño Normal: representan los lugares o estados.
- Rectángulo de color Negro: representan las transiciones discretas.

- Círculos de Tamaño Pequeño: representan a las decisiones del diagrama de flujo.
- Círculos de Tamaño Grande: representan a los sub- estados que se generan a partir de las decisiones que toma el sistema.
- Las flechas o arcos son de tres formas diferentes:
 - \longrightarrow Arcos Negros: Se utilizan para unir las transiciones con los lugares.
 - \dashrightarrow Arcos de Prueba: Se utilizan para las decisiones que llevan a los sub- estados del sistema.
 - \longrightarrow ● Arcos Inhibidores: Se utilizan para aislar el sub- proceso, solo se activa cuando se coloque la marca o token dentro del círculo pequeño que lleve a cada uno.
- ● Token o Marca: se utilizan para activar los diferentes estados y sub- estados que tiene el sistema.

En la figura 2.28 se muestra un diagrama con los símbolos o elementos que componen las redes de Petri.

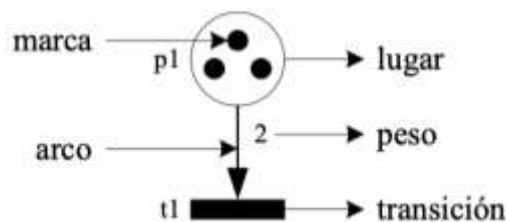


Figura 2. 28: Elementos de la Red de Petri
Fuente: (Torres, 2007)

2.9.3 Interpretación de los grafos en RdP

Para interpretar los elementos de la red de Petri, primero se debe asociar a los lugares una lista de acciones o salidas que van a estar separadas por comas, estas salidas van a estar activadas cuando el lugar o estación esté marcado. Para las transiciones se las asocia con una condición, la función lógica de las entradas, una lista de salidas, que se activarán cuando la transición se active. Las salidas están enlazadas a las entradas lógicas, por esto la activación depende que las entradas lo estén primero. El conjunto de marcas en un lugar o estación determinado determina un estado en la red de Petri.(Torres, 2007)

2.9.4 Propiedades de las redes de Petri

Para realizar las redes de Petri se deben tener en cuentas las propiedades que estas poseen, a continuación, se las enuncian. (Torres, 2007)

- La red de Petri es útil siempre y cuando el sector inicial en que se la vaya a utilizar sea compatible con todas sus transiciones, en caso de serlo se inicia su marcado inicial. Una transición es útil o viva siempre que el marcado inicial sea posible alcanzarlo al momento de darle inicio. Que la red de Petri este viva indica que no contiene bloqueos.
- Una red de Petri general contiene una serie de propiedades que no aseguran totalmente que la red este correcta, pero sí que no tenga errores potenciales.
- Una red de Petri debe ser limitada, esto es una condición necesaria para realizarla de forma física.
- En caso de que la red Petri solo pueda contener marcas de 1 en cualquier lugar de la misma, se dice que la red es 1-limitada o binaria.
- Si el número máximo de marcas que contiene la red es k , se dice que es k -limitada para el marcado inicial.
- Si un conjunto de transiciones de un mismo lugar tiene la intersección lógica de las condiciones asociadas nula, se dice que la red está potencialmente en conflicto
- Un conflicto se genera si se sensibilizan varias de las transiciones y satisface las condiciones asociadas a el lugar de entrada, pero no dispone de un número de marcas suficientes para dispararlas de forma simultánea.
- Para evitar un conflicto se da prioridad a unas transiciones con respecto a otras modificando la interpretación de las mismas.

2.9.5 Condiciones de diseño para las redes de Petri

Para realizar las redes de Petri de un proceso automatizado secuencial se necesita seguir los siguientes pasos:

1. Reconocer y denominar las entradas y salidas del sistema.
2. Detallar los símbolos es decir los lugares, transiciones, arcos, condiciones, entradas y salidas a utilizar que estén asociadas a la red de Petri que se vaya a realizar.
3. Fijar el marcado de inicio.

CAPÍTULO III: ANÁLISIS DEL PROTOTIPO DE SENSORES INTELIGENTES DE CAUDAL Y NIVEL

Este capítulo inicia con el planteamiento de las condiciones iniciales que son requeridas para proyectar y diseñar los sensores inteligentes industriales. En base a estas consideraciones se propone una arquitectura multiagente basada en el modelo BDI, para un sensor inteligente cuyo diseño, modelado, simulación y validación se desarrollará utilizando una metodología basada en UML – PN (Lenguaje de modelado unificado y redes de Petri). Se realiza la implementación utilizando microcontroladores de 8 bits.

3.1. Consideraciones Iniciales del Sistema

Las consideraciones iniciales del sistema, se ha considerado la descripción, las características y funciones que se asignarán a los sensores inteligentes. Todo esto es previo al análisis de la arquitectura, el modelado en UML y las redes de Petri.

3.1.1 Descripción del Sistema

Los prototipos de sensores inteligentes industriales de caudal y nivel, son analizados considerando los requerimientos de un subsistema de filtrado en una planta potabilizadora de agua. Para la implementación de estos prototipos se utiliza dispositivos de bajo costo como el microcontrolador, y demás componentes que forman parte del sensor. Por lo consiguiente, en los siguientes párrafos se describe el funcionamiento general de los sensores para comprender su propósito y las tareas que va a tener que ejecutar.

Los sensores inteligentes de caudal y nivel tienen funciones específicas previamente programadas por lo que es ideal proponer una nueva arquitectura que se adapte a las especificaciones que requieren estos dispositivos. Después de diseñar la arquitectura, es necesario evaluar para determinar su comportamiento y si es necesario cambios. El sistema será modelado en UML en sus diferentes diagramas como son: casos de uso, de secuencia y el de estados. Una vez que se ha elaborado estos diagramas, se realiza el modelado en redes de Petri para este sistema, a través de la red de Petri se podrán obtener las ecuaciones necesarias para el modelamiento matemático del controlador que llevarán estos sensores. El lenguaje de programación, es lenguaje C.

La ventaja que posee el realizar el modelado en UML y Redes de Petri antes de realizar la programación que represente el sistema, es que permite verificar y validar el comportamiento del sistema y facilita considerablemente la escritura del código para cualquier tipo de lenguaje, ya que solo se necesita interpretar la arquitectura, los diferentes modelados de UML y la red de Petri para comprender como van a funcionar los sensores inteligentes. La arquitectura propuesta en este capítulo será apta tanto para la variable de caudal como para la de nivel, es decir que se podrá utilizar la misma estructura para cualquiera de las dos variables.

La arquitectura propuesta está basada en el modelo BDI, que es un agente deliberativo, en el cual se tiene las creencias que es todo lo que se sabe del entorno, que lleva a plantear unos objetivos que son los deseos, y que a su vez tiene sub - objetivos que son las intenciones, posterior a eso tiene las metas que producen los planes en donde se comprueba si lo anterior planteado esta correcto o necesita cambios.

El sistema de sensado inteligente, permite realizar funciones predeterminadas en el algoritmo de programación, el cual genera un trabajo eficiente sobre la variable que controla. Una de estas variables, es la auto calibración del sensor, que reconfigura los parámetros del sensor, conforme a los valores que se encuentre registrados en una base de datos del microcontrolador, que incluye los rangos a los que debe llegar, y determinar los valores de operación, entre otras tareas que realiza.

En la imagen 3.1 se muestra de forma general como opera el sistema a través de un diagrama de flujo. Al inicio carga la base de datos de las creencias, donde se encuentran los rangos de operación, señal de voltajes, frecuencia de trabajo, entre otros valores necesarios para la operación del sistema. Luego continúa su flujo verificando si existe un usuario si es el caso pasa a la configuración de la variable solo si es una nueva variable que no se encuentre registrada en la base de datos de creencias, si es el caso contrario pasa a verificar la señal con el sensor, si no responde se emite una alarma de error, pero si existe un sensor, el sistema pasa a verificar la comunicación con PLC, si existe dicha comunicación recibe la señal, procesa la señal y la envía al PLC, si el PLC desea emitir una orden la recibe a través del intérprete de comandos donde se ejecutan las funciones solicitadas y repite el bucle anterior. Si es el caso contrario de que no existe comunicación alguna emite una alarma generando un error en el sistema.

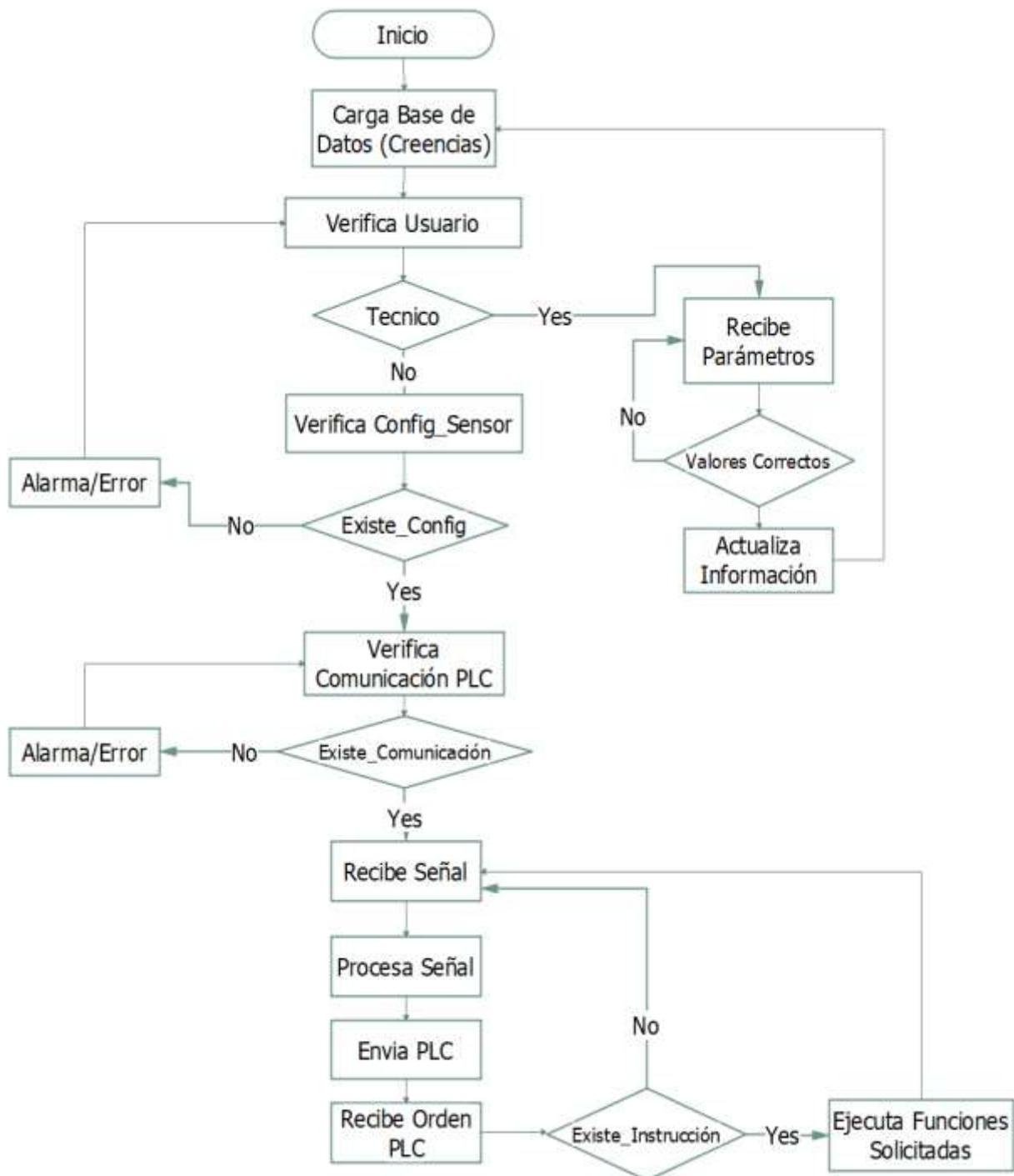


Figura 3. 1: Diagrama de Flujo general
Elaborado por: Autor

3.1.2 Caracterización del Sistema

En esta sección se describirá las características técnicas que contiene el sistema de sensado inteligente para las variables de caudal y nivel. Los sensores que se utilizan, los rangos de operación y el voltaje de funcionamiento del sistema general.

- El sistema se compone por un microcontrolador ATmega 2560, que tiene un voltaje operativo de 5V, tensiones de entrada en un rango de 7 a 12 voltios, 54 pines digitales de comunicación sea de entrada o salida, de los cuales 14 cuentan con salida PWM, 16 pines análogos, la memoria flash es de 256KB de los cuales 8KB son usado por el bootloader, la memoria SRAM es de 8KB, la EEPROM de 4KB y la velocidad del reloj es de 16MHz.
- El sistema cuenta con comunicación industrial tipo Modbus para comunicar el sistema de sensado inteligente con el controlador o PLC que procederán a comunicarse con el actuador.
- Uso de Redes de Petri para comprobar el funcionamiento de la arquitectura planteada para el sensor inteligente.
- El sensor para medir nivel es el ultrasónico HC – SR04, la tensión de alimentación es de 5Vcc, la frecuencia a la que trabaja es de 40 KHz con un rango máximo 4.5m y mínimo de 1.7cm. La duración mínima del pulso de disparo es de 10 μ S y la duración del pulso de salida 100-25000 μ S. El tiempo que demora entre una medida y otra es de 20 mS.
- Para caudal se utiliza el sensor YF-S201, es compatible con microcontroladores como Arduino, Raspberry, Pic, etc. Su voltaje de operación es de 5V-18VDC, consume 15 mA de corriente y la capacidad de carga es de 10mA, la señal de salida es una onda cuadrada, el rango del flujo es de 1-30L/min, emite 4s0 pulsos por litro y el volumen promedio por cada pulso es de 2.25mL.
- Los rangos de valores que va a manejar el sensor son diferentes para Caudal: 0-10 lt/seg y para Nivel: 0-100cm.
- El sistema general utiliza una fuente de tensión de 12 voltios DC.

3.1.3 Funciones del Sistema

El sistema de sensado inteligente cuenta con las siguientes funciones:

- Digitalización de la señal, en esta sección lo que hace el microcontrolador es receptor los valores analógicos que le emite el sensor y los convierte a digital para la mejor interpretación de la información en el programa.
- Auto calibración, esta función es una de las más importantes por lo que el sensor siempre debe mantenerse dentro de los rangos preestablecidos, lo que

hace esta función es que el recuerde de forma permanente estos valores, reduciendo los fallos en el sistema.

- Análisis y Promedio de valores, cada cierto tiempo el sistema debe promediar los valores que obtiene del sensor y verificar si está todo en el orden correcto o algo está fallando, en caso de fallar se genera una interrupción dentro del código de programación, que se la considera una función adicional.
- Generar Alarmas, es la función del sistema que trabaja en base a un promedio de valores, en caso que algo está fallando dentro de la operación del sistema, este va generar una alarma que avise al usuario de que algo no está corriendo bien dentro del programa.
- Extrapolación de Datos, esta función se utiliza cuando un sensor deja de funcionar, y almacena el último valor que este emite para continuar con su funcionamiento de forma natural. Esta función va a ser útil solo si existen fallos en el sensor.
- Verificación y Validación, en esta parte es donde se alojan todas las tareas que deben ejecutarse, aquí verifica - valida que todos los datos estén correctos para ser ejecutados por el autómata, en caso de ser datos basura los desecha, y no pasa la información a la siguiente función.
- Adaptación de Comunicación, esta sección sirve para conectar el autómata con el microcontrolador y viceversa, de forma segura y eficiente. Por esto se lo realiza a través de un protocolo de comunicación industrial fiable y seguro.
- Interprete de comandos, es una función que permite la recepción de órdenes que emita el autómata al microcontrolador. Verifica que los datos no sean basura, y procede a ejecutar todas las tareas preestablecidas del sistema.

3.2. Diseño de la Arquitectura Nodo Sensor/ Actuador

Para el diseño de la arquitectura del nodo sensor/Actuador (SEUDO AGENTE INTELIGENTE) basada en el modelo BDI. Estos elementos se describen a continuación, que servirá como base para proyectar y proponer una arquitectura de nodo sensor-actuador orientada a entornos industriales. En la figura 3.2 se muestra el orden del modelo de arquitectura BDI, a continuación de la imagen se describe cada uno de los elementos de este modelo.

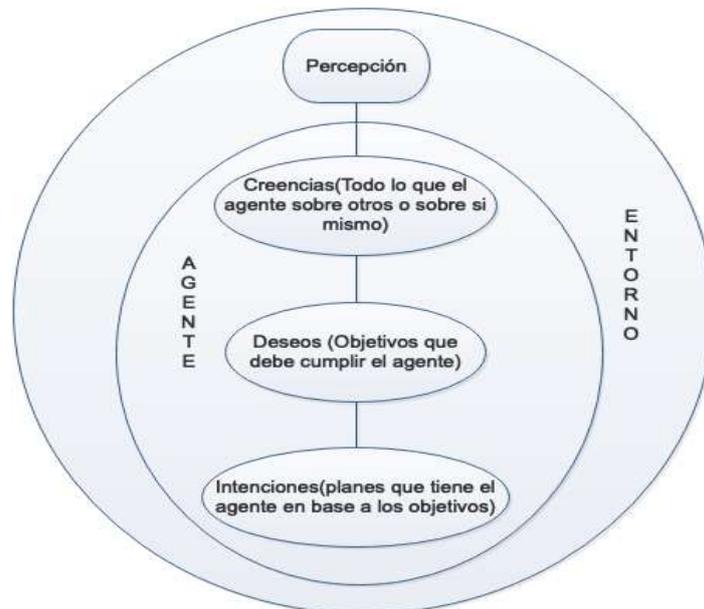


Figura 3. 2: Elementos del modelo BDI
Adaptado de: (Loarte et al., 2015)

En la figura 3.3 se muestra un gráfico donde se muestra la relación de los tres elementos del modelo BDI, considerando las entradas y salidas del sistema. A partir de los elementos analizados del modelo DBI en el ítem anterior y que han sido revisados con más detalle en el marco teórico, se analiza por etapas la siguiente arquitectura, hasta llegar a la parte final.

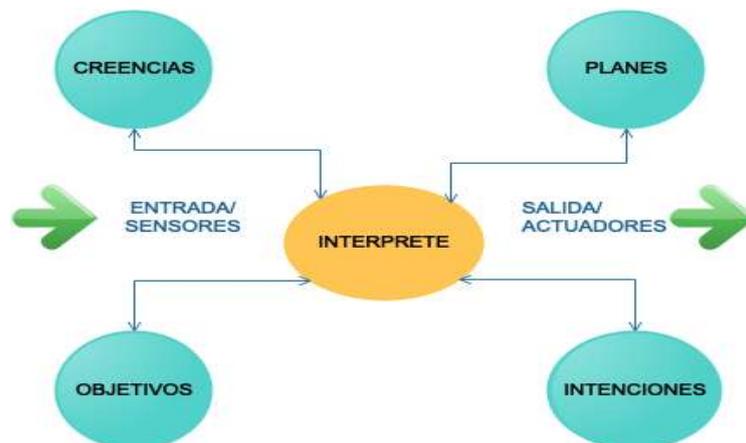


Figura 3. 3: Relación de los elementos del modelo BDI
Adaptado de: (Loarte et al., 2015)

3.2.1. Análisis de la primera etapa de la Arquitectura del Sistema

La primera parte de la arquitectura del sistema es un clásico inicio de una BDI, pero es necesario porque así el sistema determina que variable es la que va a sensor en ese momento, es decir para el caso de este trabajo solo será nivel o caudal, pero

en caso de que sea otra variable lo que vaya a controlar, está arquitectura es totalmente apta y flexible para la función que realice.

En la figura 3.4 se muestra la primera etapa de la Arquitectura del sistema.

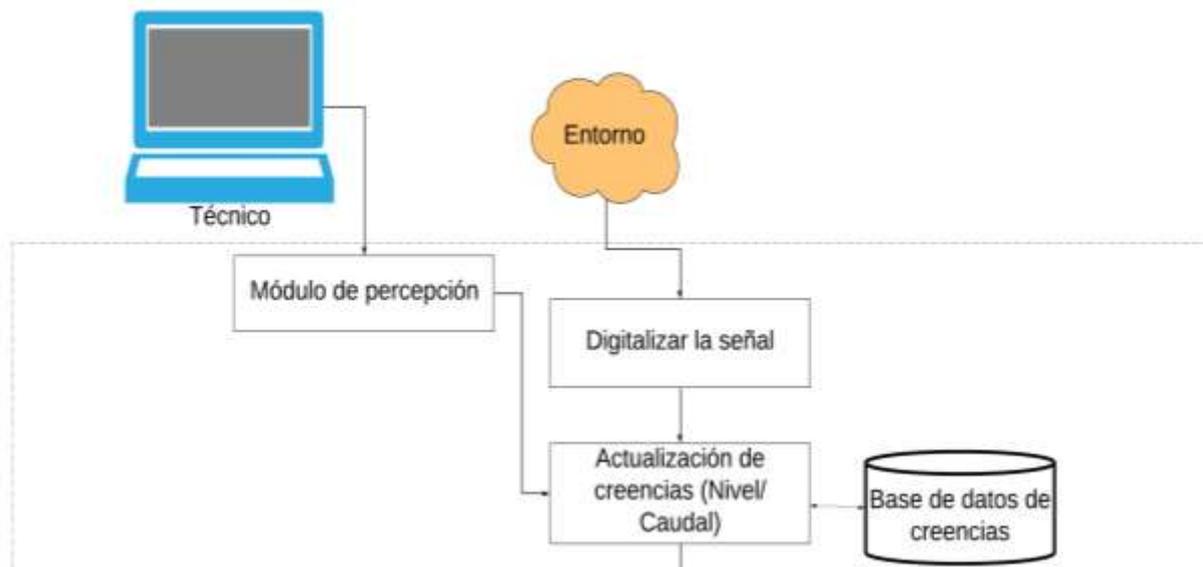


Figura 3. 4: Primera etapa de la arquitectura del sistema
Elaborado por: Autor

La primera etapa está compuesta por los siguientes elementos:

- Entorno, es lo que se sabe del sistema, en este caso la variable que va a controlar, esto es el inicio de toda arquitectura BDI.
- Módulo de percepción, se la utiliza para establecer la variable que va a controlar el sistema, a través de la computadora que se basa en los rangos asignados previamente para nivel y caudal.
- Digitalizar la señal, una vez obtenida la información de la variable que vaya a controlar, pasa a la digitalización de la misma para comprender de una mejor manera el sistema.
- Actualización de creencias, los anteriores elementos excepto por el entorno constituyen las creencias del sistema, en este caso es nivel o caudal pero que está relacionada directamente con una base de datos en la que se almacenará datos importantes como son los rangos de operación a los que debe trabajar el sensor para que opere de forma eficiente.
- Base de datos, la base de datos de las creencias contiene todos los datos importantes que debe saber el sensor antes de pasar a la ejecución de las tareas que es la segunda etapa de la misma.

3.2.2. Análisis de la segunda etapa de la Arquitectura del Sistema

La segunda etapa de la Arquitectura está compuesta por los deseos y las intenciones que debe cumplir el sistema. En la figura 3.5 se presenta esta segunda etapa.

- Deseos, dentro de esta categoría se encuentran dos objetivos que debe cumplir el programa, el primero es la función de selección de comportamiento la cual está destinada a elegir el despachador de eventos, en el cual se despliegan las intenciones de la arquitectura.
- Intenciones, en el despachador de eventos, se encuentran las intenciones de la arquitectura o los sub-objetivos que debe cumplir el sistema, que son las funciones a las que debe llegar el sensor junto con el microcontrolador. Se tiene la auto calibración, para que el sensor recuerde su Setpoint cada vez que lo necesite, el promedio de valores para tener una prevención en caso de fallos en el sistema por lo que también genera alarmas para saber de estos fallos y la extrapolación de datos, que también se la usa para los errores del programa.

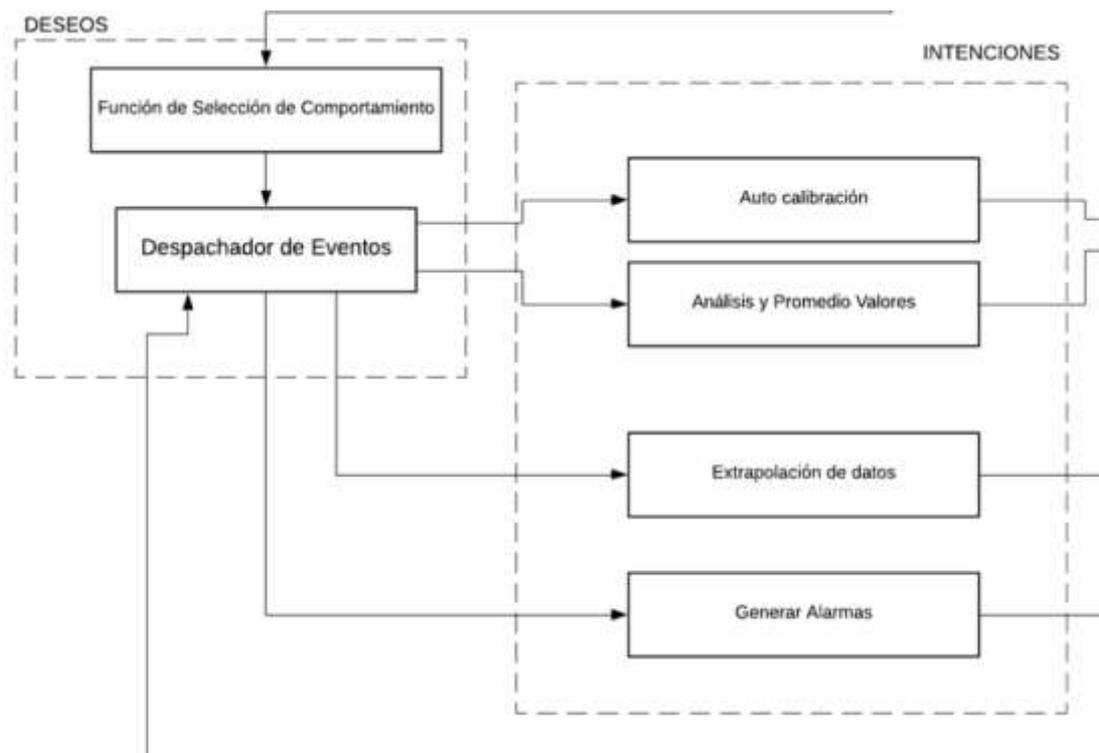


Figura 3. 5: Segunda etapa de la arquitectura del sistema
Elaborado por: Autor

3.2.3. Análisis de la tercera etapa de la Arquitectura del Sistema

La tercera parte de esta arquitectura está compuesta por los agentes externos e internos, es decir la comunicación final entre el autómata y el microcontrolador, para lo cual se establece una comunicación fiable para enviar /recibir datos del microcontrolador al autómata y viceversa.

En la imagen 3.6 se muestra la tercera parte de la arquitectura del sistema.

- Verificación y Validación, esta función está conectada a las intenciones de la arquitectura y pasa a la comunicación entre el microcontrolador y el autómata.
- Interprete de Comandos, sirve en caso de que el autómata envíe algún comando u orden al microcontrolador, en la que primero analiza si es un dato basura o no y en caso de no serlo pasa a los objetivos que posterior pasa a las intenciones y así cumple el bucle establecido, para salir de nuevo al autómata.
- Adaptación de comunicación, este representa al protocolo de comunicación que se establece entre el microcontrolador y el autómata.
- Autómata, representa al objetivo final a quien llegará todos los datos que el microcontrolador le envíe y en caso de que este dispositivo envíe al microcontrolador se establezca una comunicación fiable y fluida sin interferencias.

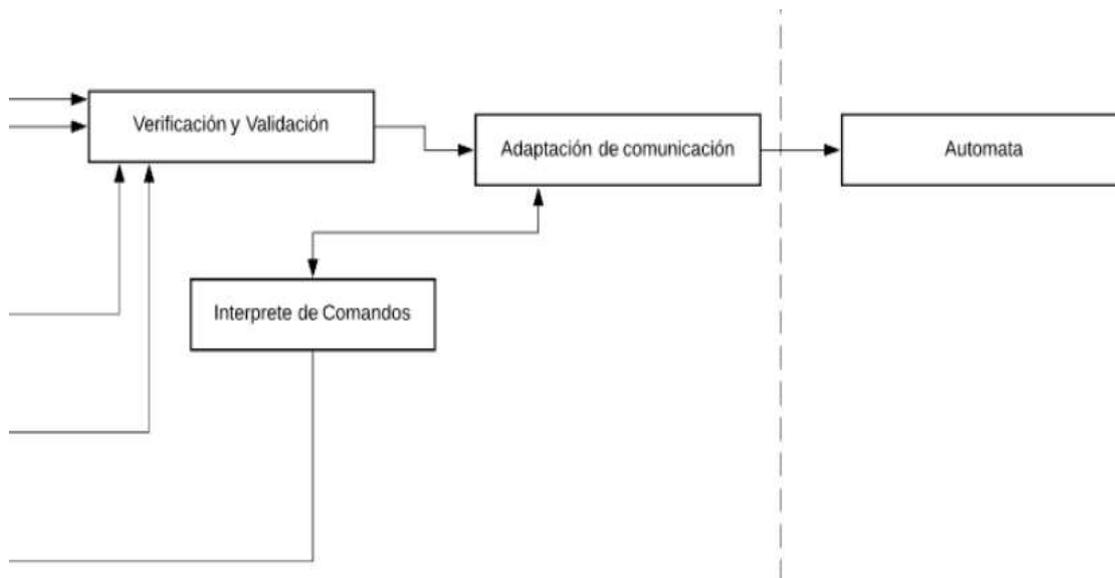


Figura 3. 6: Tercera etapa de la arquitectura del sistema
Elaborado por: Autor

3.2.4. Arquitectura Final del Sistema

Al integrar las tres etapas que se han presentado en los puntos anteriores, se tiene la arquitectura final del sistema. Se logró una arquitectura que se válida para cualquier tipo de variable que se vaya a controlar, a pesar de que las variables de estudio en este trabajo de investigación son caudal y nivel.

En la figura 3.7 se muestra la arquitectura final del sistema de sensado inteligente.

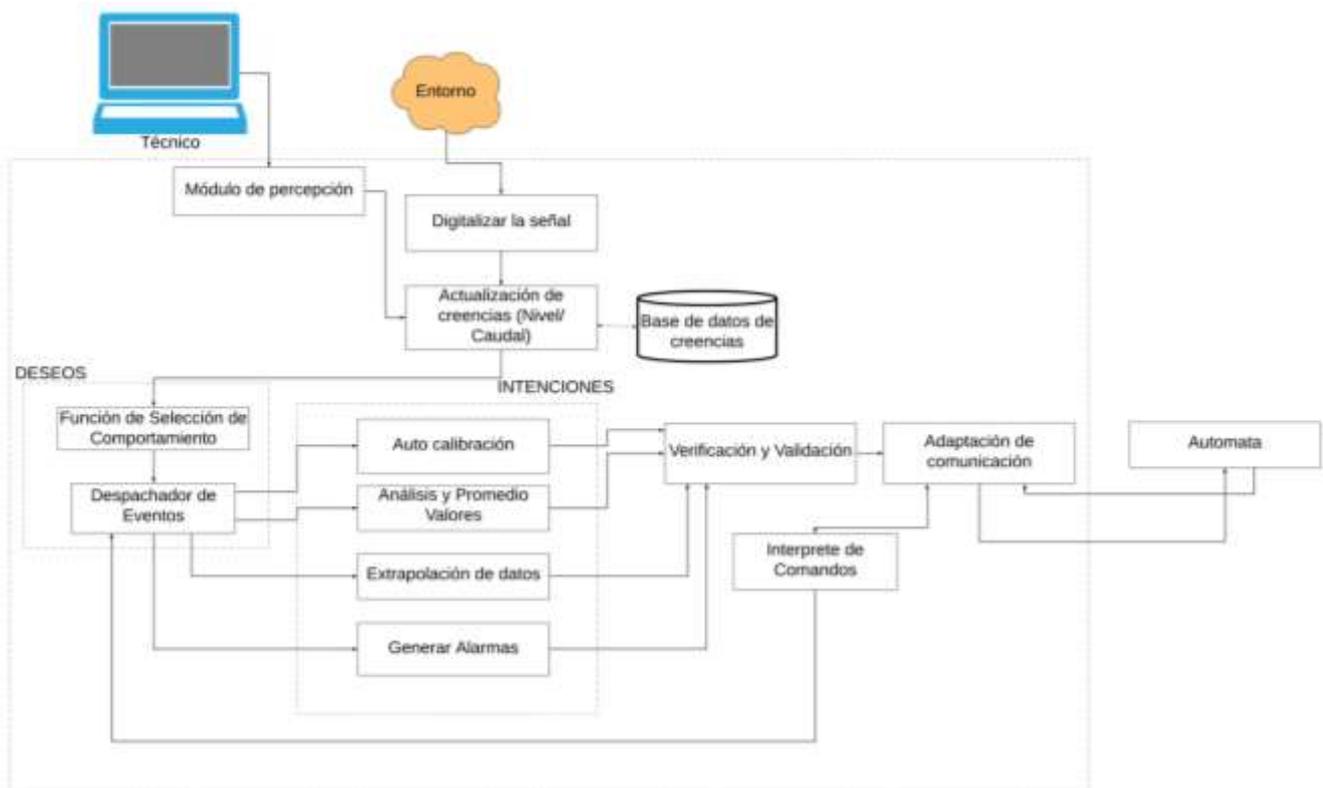


Figura 3. 7: Arquitectura final del sistema
Elaborado por: Autor

3.3. Modelado UML del Nodo Sensor/Actuador

UML es una forma de modelado más utilizado para desarrollo de software en la ingeniería, otra de las utilidades que se le da a este modelado es para estructurar sistemas de control industrial. Por tal motivo se presentan 3 de sus diagramas más comunes que son los que más se ajustan al caso de estudio de este proyecto de investigación. Se utiliza el diagrama de casos de uso para expresar los actores que intervienen en el sistema que son el microcontrolador o agente, el técnico, el sensor y el autómeta, y definir los casos de uso o actividades que va a tener cada uno. El diagrama de secuencia muestra de que forma interactúan los dispositivos del sistema y especifican los mensajes que se envían entre ellos. Por último, se tiene el diagrama de estados que muestra los lugares con las transiciones del sistema y como avanza

en su ciclo de vida, también muestra los estímulos externos o internos y como se desarrollan en cada estado del sistema.

3.3.1 Diagrama de Casos de Usos

El diagrama de Casos de Usos se utiliza para ejemplificar la acción de cada actor o componente que interviene en el sistema y las acciones o casos de uso que debe realizar. La figura 3.8 es el diagrama de casos de uso del sensor inteligente, que se describe a continuación.

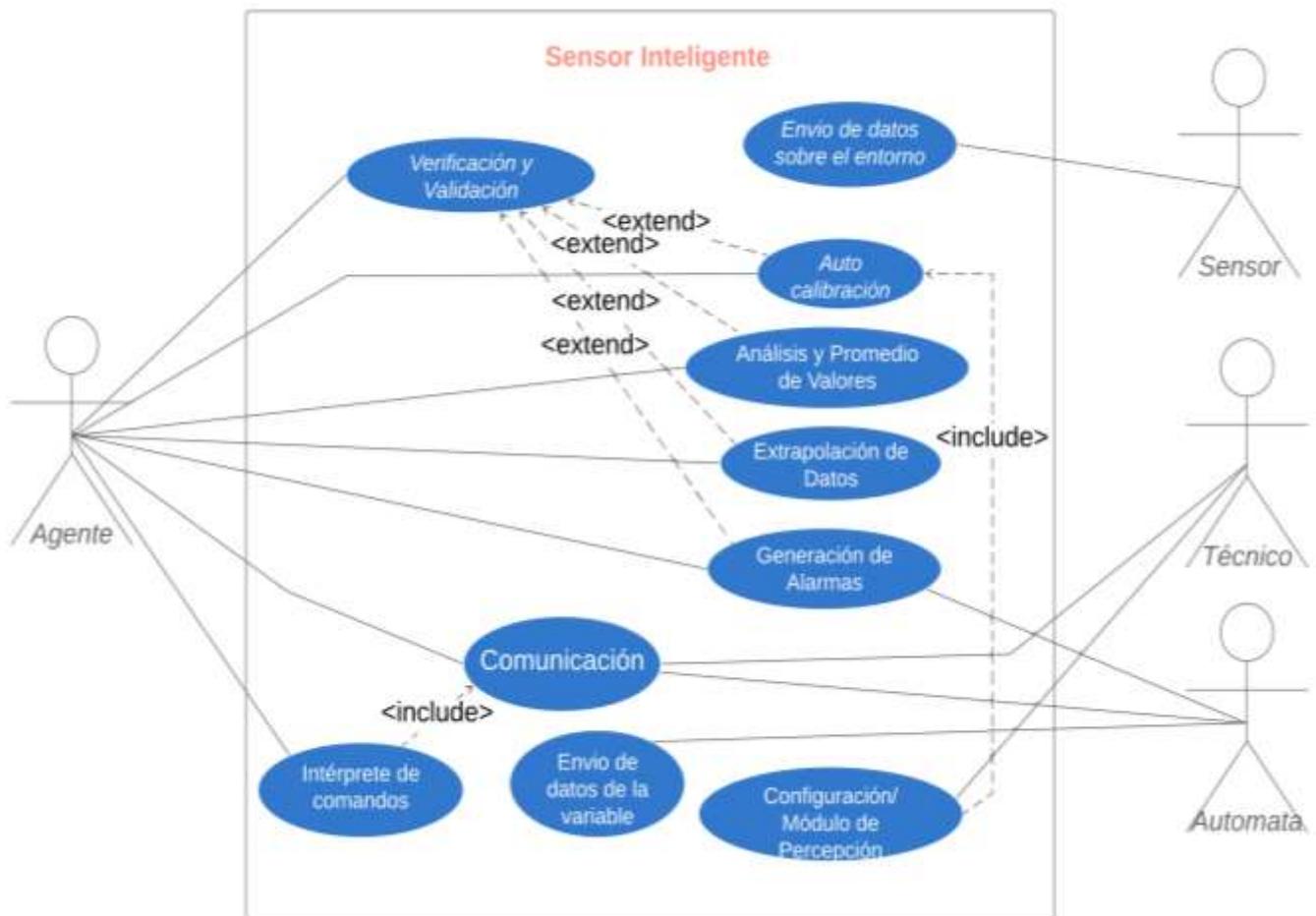


Figura 3. 8: Casos de Uso del Sensor Inteligente
Elaborado por: Autor

3.3.1.1 Casos de Uso del Agente

El agente inteligente en este caso es el microcontrolador, que tiene los siguientes casos de uso:

- Verificación y Validación, esta función se emplea para comprobar que los datos que se envían al autómata son correctos y no son datos basura, en caso de

que sean datos así los desecha, no permite el paso de esta información y pide una nueva para enviarle al autómata.

- Auto calibración, en este espacio se almacena los rangos de operación de cada variable, para que el agente recuerde estos valores constantemente con el fin de evitar fallos en el sistema.
- Análisis y Promedio de Valores, aquí se almacena los valores de operación del sensor, para realizar un análisis y promediar para verificar que su funcionamiento este correcto o en caso de no estarlo va a emitir una alarma.
- Extrapolación de Datos, esta función registra los últimos valores emitidos por el sensor, para que siga su flujo normal y genera una alarma para comunicar el fallo.
- Generación de Alarmas, función que permite emitir alarmas en caso de que sea necesario o de que alguna otra función lo solicite.
- Comunicación, le permite al agente establecer el protocolo adecuado de comunicación con el autómata para enviar o recibir datos.
- Intérprete de Comandos, esta función es un caso de uso incluido con la comunicación agente/autómata, debido a que este es un comando que el autómata le envía al agente, que procede a verificarse y pasa para cumplir todas las funciones anteriores.

3.3.1.2 Casos de Uso del Sensor

El sensor va a tener un único pero importante caso de uso, puesto que este va a informar al sistema sobre el entorno, es decir va indicar la variable que está sensado sea caudal o nivel. Debido a eso el caso de uso del sensor tiene por nombre “Envío de dato sobre el entorno”.

3.3.1.3 Casos de Uso del Técnico

El técnico es una función del usuario, cada vez que necesite configurar algún sensor nuevo para el sistema, lo conecta al PC y puede pedir información para monitorear o puede configurar algún sensor nuevo al sistema. Configuración del Módulo de Percepción, aquí es donde se configura los rangos de valores para nivel y caudal, con el fin del que el agente tenga conocimiento de la variable que este controlando en ese momento. Comunicación, también pertenece a este caso de uso

ya que necesita estar en contacto con el agente para informarle sobre cambios en la variable que este midiendo, o si hay alguna nueva variable que deba controlar.

3.3.1.4 Casos de Uso del Autómata

El autómata es el PLC del sistema, y este va a tener tres casos de usos que son los siguientes:

- Comunicación, es el protocolo MODBUS que permite establecer contacto agente/autómata.
- Generación de Alarmas, en caso de ser necesario el autómata también puede generar una alarma para informar al agente que hay fallos en el envío de la información.
- Envío de datos de la variable, se lo utiliza para que el autómata envíe información al agente, que sea fiable y segura.

3.3.2 Diagrama de Secuencia

El diagrama de secuencia representa como todos los casos de uso o actividades convergen entre sí. Se han elaborado tres diagramas de secuencia, el primero que representa al sistema en general y los otros dos que son subprocesos del sistema.

En la figura 3.9 se presenta el diagrama de secuencia general del sistema, el agente inicia verificando la comunicación con el autómata, en caso de no tener la misma se queda en el bucle esperando a que se establezca una comunicación fiable. Una vez que establece esta comunicación pasa a verificar si existe una entre el agente con el sensor, en caso de no existir una se queda esperando a que la señal se establezca para proceder a las siguientes secuencias y genera una alarma para comunicar el fallo. Una vez establecidas ambas comunicaciones el agente procede a realizar sus funciones de Auto calibración, Analizar y promediar los valores que le pide al sensor y la Extrapolación de datos en caso de ser necesaria. Para finalizar envía los datos al autómata y este responde con una señal de reconocimiento.

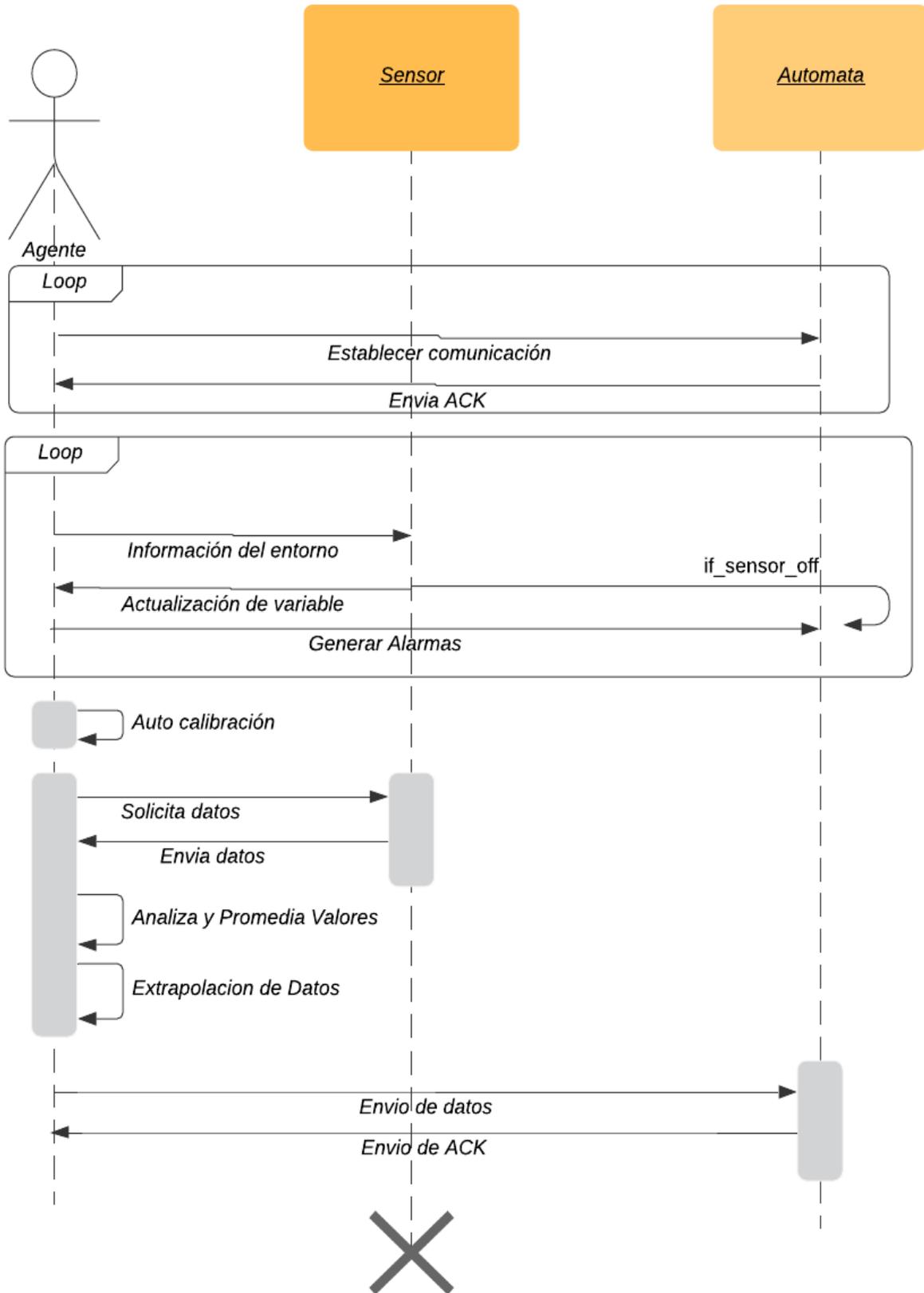


Figura 3. 9: Diagrama de Secuencia General del Sensor Inteligente
Elaborado por: Autor

3.3.2.1 Diagrama de Secuencia Autómata – Agente

El diagrama de secuencia Autómata – Agente es presentado en la figura 3.10, este subproceso ocurre una vez que se ha establecido la comunicación con el autómata, solo cuando este quiera enviarle información al agente. De tal modo que inicia enviando un comando, el cual llega al agente y este lo interpreta, si esta correcto pasa al despachador de eventos que lo lleva a realizar todas las funciones disponibles del sistema y por último una vez que haya cumplido con todas las funciones pasa a validación de la información para comprobar que todo este correcto y luego procesa estos datos para ser enviados como respuesta del agente al autómata.

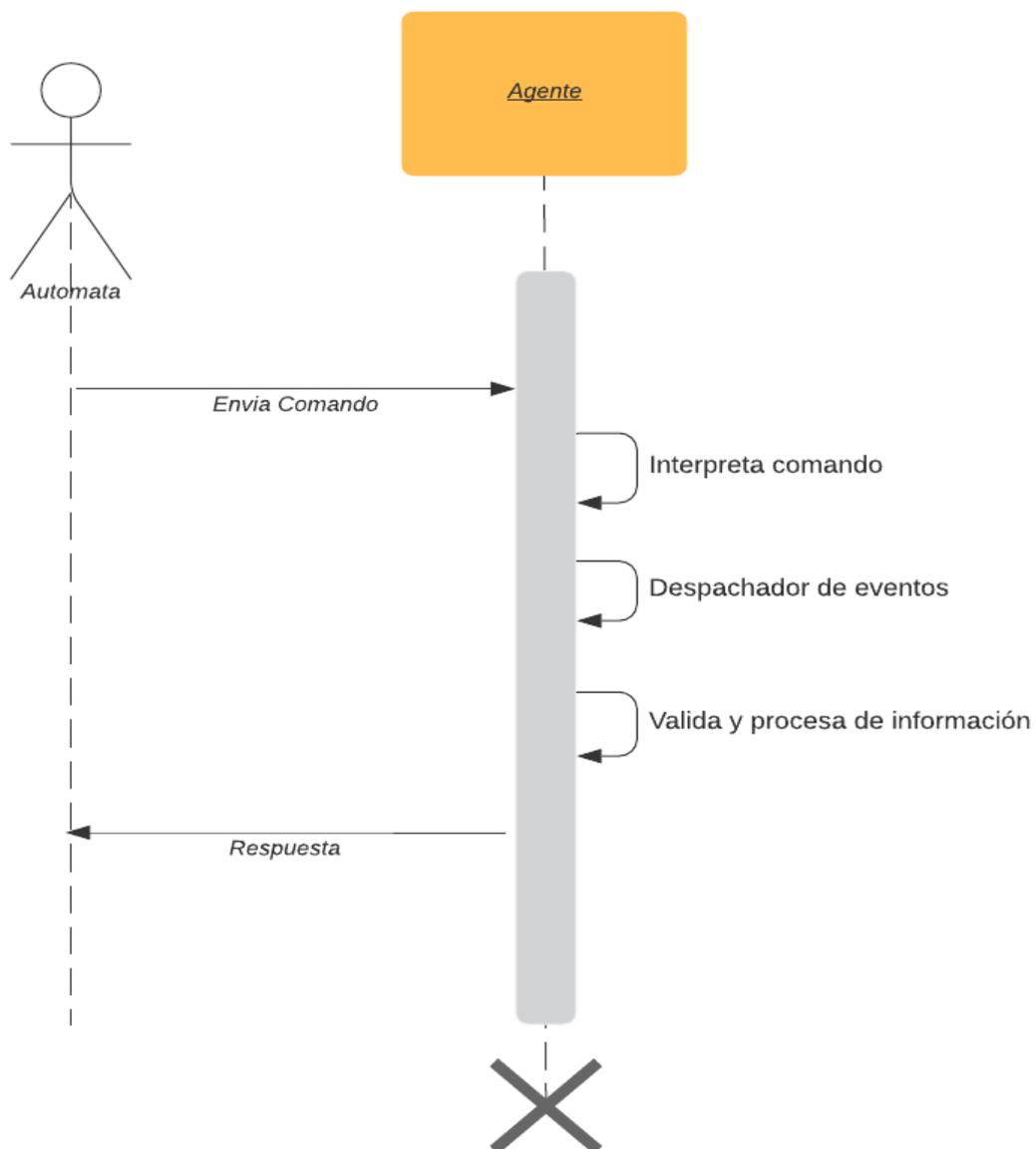


Figura 3. 10: Diagrama de Secuencia Autómata - Agente
Elaborado por: Autor

3.3.2.2 Diagrama de Secuencia Técnico – Agente

Este subproceso es utilizado siempre y cuando el técnico este activo, es decir si el usuario requiere de información sobre el sistema o tiene una nueva variable que desee agregar al sistema. También se lo utiliza para configurar el sensor, en este subproceso se incluyen los rangos de operación de cada uno de los sensores. En la figura 3.11 se muestra este diagrama, el cual inicia verificando si existe una comunicación, es decir si el técnico está activo, en caso de estarlo responde con un ACK y comienza la configuración del sensor, luego pasa al agente y este realiza las actividades de verificación en la base de datos de las creencias, la actualiza y confirma o niega la información en caso de que sean datos basura que se quiera enviar.

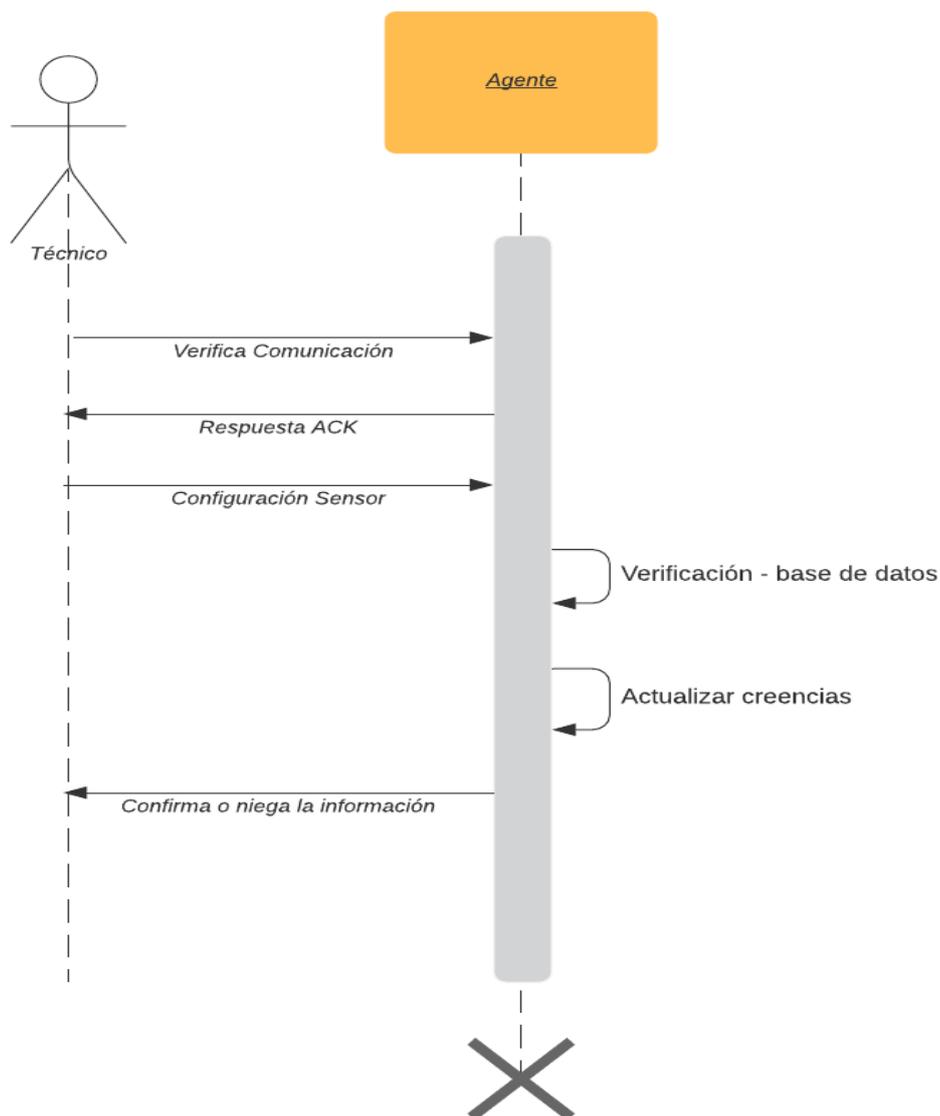


Figura 3. 11: Diagrama de Secuencia Técnico- Agente
Elaborado por: Autor

3.3.3 Diagrama de Estados

Este diagrama representa el conjunto de estados por los que tiene que pasar el sistema, para ejecutar las acciones correspondientes. Está conformado por estados y transiciones, que a su vez incluyen eventos, acciones y actividades que el agente debe realizar. El diagrama comienza con el primer estado que es configuración y actualización de creencias, posterior pasa a una transición donde verifica la disponibilidad del técnico, es decir que entra en una decisión donde si el técnico no está activo sigue su flujo normal, pero si el técnico está activo pasa al sub- estado de configuración del sensor. El siguiente estado se verifica la señal del sensor, y se comunica a una transición para verificar si el sensor está funcionando correctamente, si lo está haciendo sigue su flujo normal y en caso de no estarlo pasa al sub- estado de detección de errores.

El estado tres es el despachador de eventos que contiene todas las funciones disponibles que va a realizar el agente, si las funciones son correctas pasan al siguiente estado, en caso de no estarlo se quedan en un bucle que da apertura al siguiente sub- estado llamado Selección de Comportamiento. El cuarto estado es donde se establece la comunicación con el autómatas o PLC, si la comunicación fue exitosa, sigue el flujo normal y si no pasa al sub- estado de verificación de comunicación Agente/PLC. El último estado es en caso de que el PLC emita una orden pasa al estado 3 para desempeñar las funciones requeridas y si no existe orden alguna sigue preguntando hasta que aparezca una.

El diagrama general está compuesto por 5 sub-estados, los que son:

- Configuración del Sensor
- Detección de Errores
- Selección de Comportamiento
- Recibir Órdenes del PLC
- Verificación de Comunicación Agente/PLC

Cada uno realiza sub-funciones que permiten al sistema que se mantenga estable dentro de las perturbaciones que el sistema tenga, y sirven como prevención a fallos. En la figura 3.12 se muestra el diagrama de estados general del sistema con sus transiciones, sub-estados y decisiones.

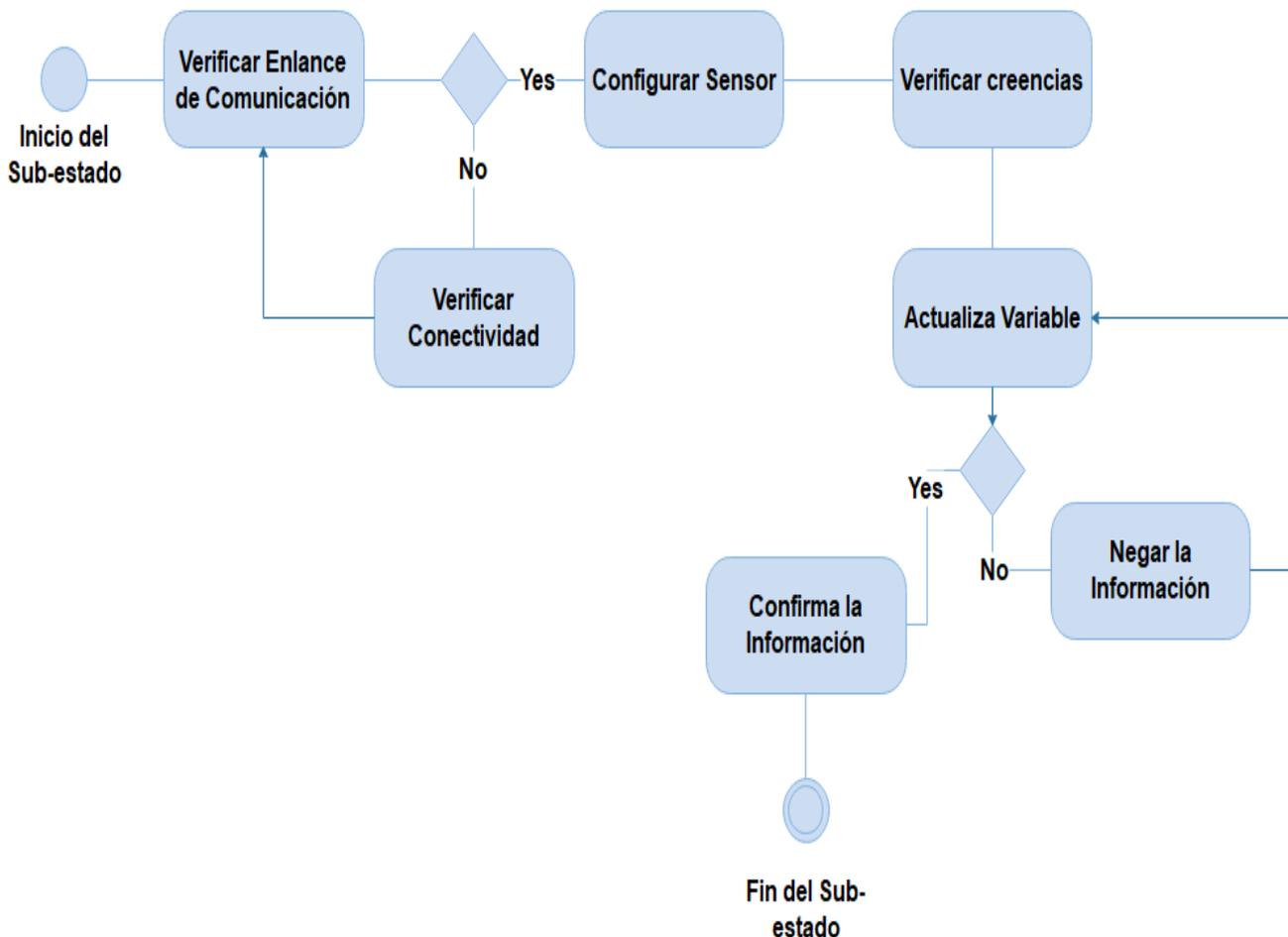


Figura 3. 13: Diagrama de Estados del Sub- proceso Configuración del Sensor
Elaborado por: Autor

3.3.3.2 Diagrama de Estados: Configuración Agente/ PLC

Este sub- estado inicia verificando el enlace de comunicación entre el Agente/ PLC, si existe esta conexión pasa a configurar el sensor, si no existe comunicación sigue en el bucle hasta que se establezca una conexión. En el siguiente estado se verifica la creencia es decir la variable que este controlando sea la correcta, se la actualiza y si todo está correcto confirma la información, pero si no está correcto el proceso niega la información, con lo que se repite todo el proceso hasta que sea correcto, finalizando este sub- estado.

En la figura 3.14 se muestra el diagrama de estados que se utiliza para modelar el sub- proceso de configuración Agente/ PLC. El cual se compone de 7 estados que son: Verificar Enlace de Comunicación, Verificar Comunicación, Enviar Comando, Interpretación del Comando, Despachador de Eventos, Validación y procesamiento de Información, Negar la Información, Enviar una respuesta, y dos decisiones.

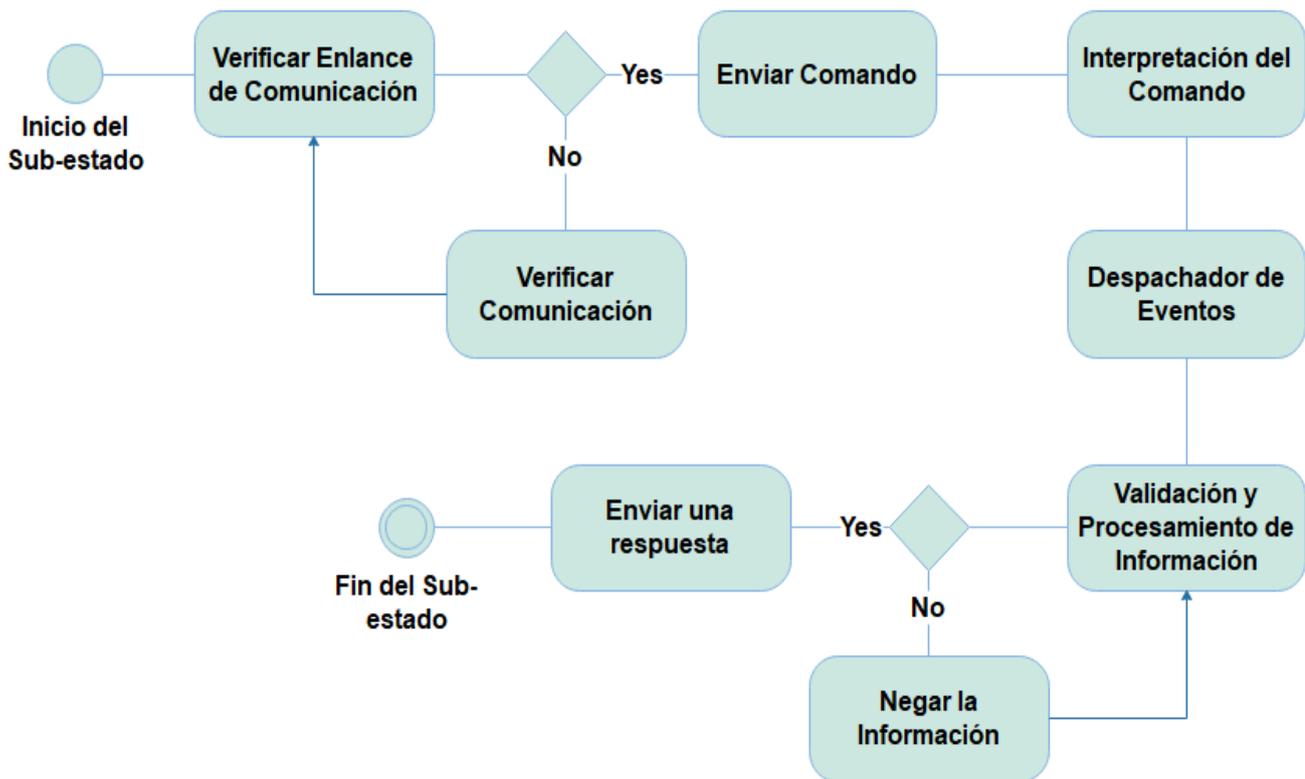


Figura 3. 14: Diagrama de Estados del Sub- proceso Configuración Agente/PLC
Elaborado por: Autor

3.3.3.3 Diagrama de Estados: Detección de Errores

Este sub- estado se lo utiliza en caso de que no se haya establecido correctamente la comunicación con el sensor, por lo que primero pide información del estado para actualizar variable, en caso de que no se haya actualizado bien la variable, regresa al inicio, caso contrario continua su flujo normal dando paso al despachador de eventos, del cual se despliegan las funciones del sistema, y pasa al último estado donde verifica todo lo que va a enviar al siguiente estado del diagrama general presentado en la figura 3.15.

3.3.3.4 Diagrama de Estados: Selección de Comportamiento

La selección de comportamiento tiene dos opciones:

- Según el tipo de Sensor activo, ya sea nivel, caudal, caudal, temperatura o la variable que este sensor proporcione.
- Según el funcionamiento que tiene, ya sea como sensor inteligente principal o sensor inteligente de respaldo.

Una vez que se elija el comportamiento que tiene este sensor pasa al despachador de eventos a cumplir con las funciones del sistema, luego pasa a verificar la

información, si todo es correcto finaliza el sub- estado para continuar con el flujo normal del sistema general.

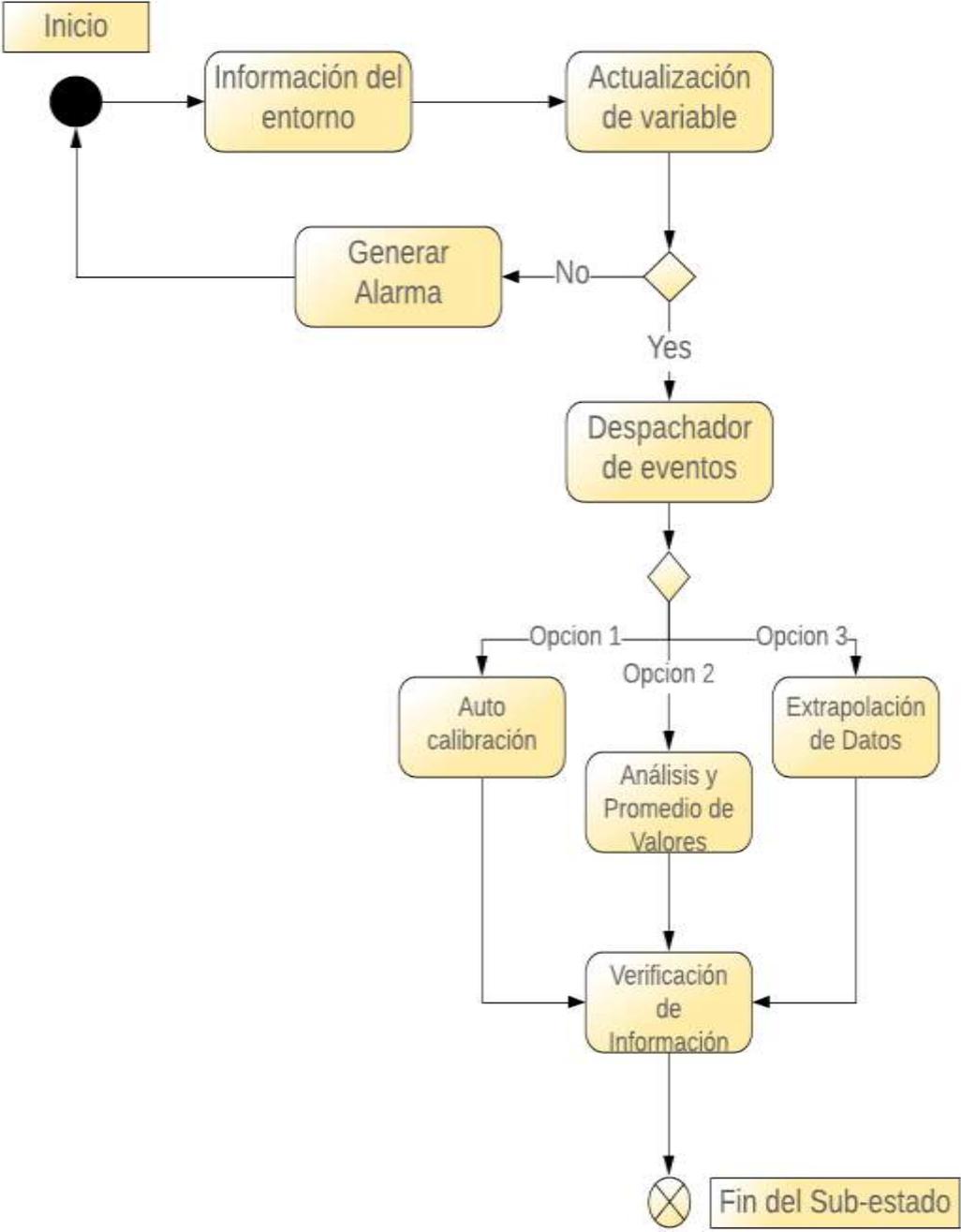


Figura 3. 15: Diagrama de Estados del Sub- proceso Detección de Errores
Elaborado por: Autor

En la figura 3.16 se muestra el diagrama de estado del sub- proceso de la función de selección de comportamiento.

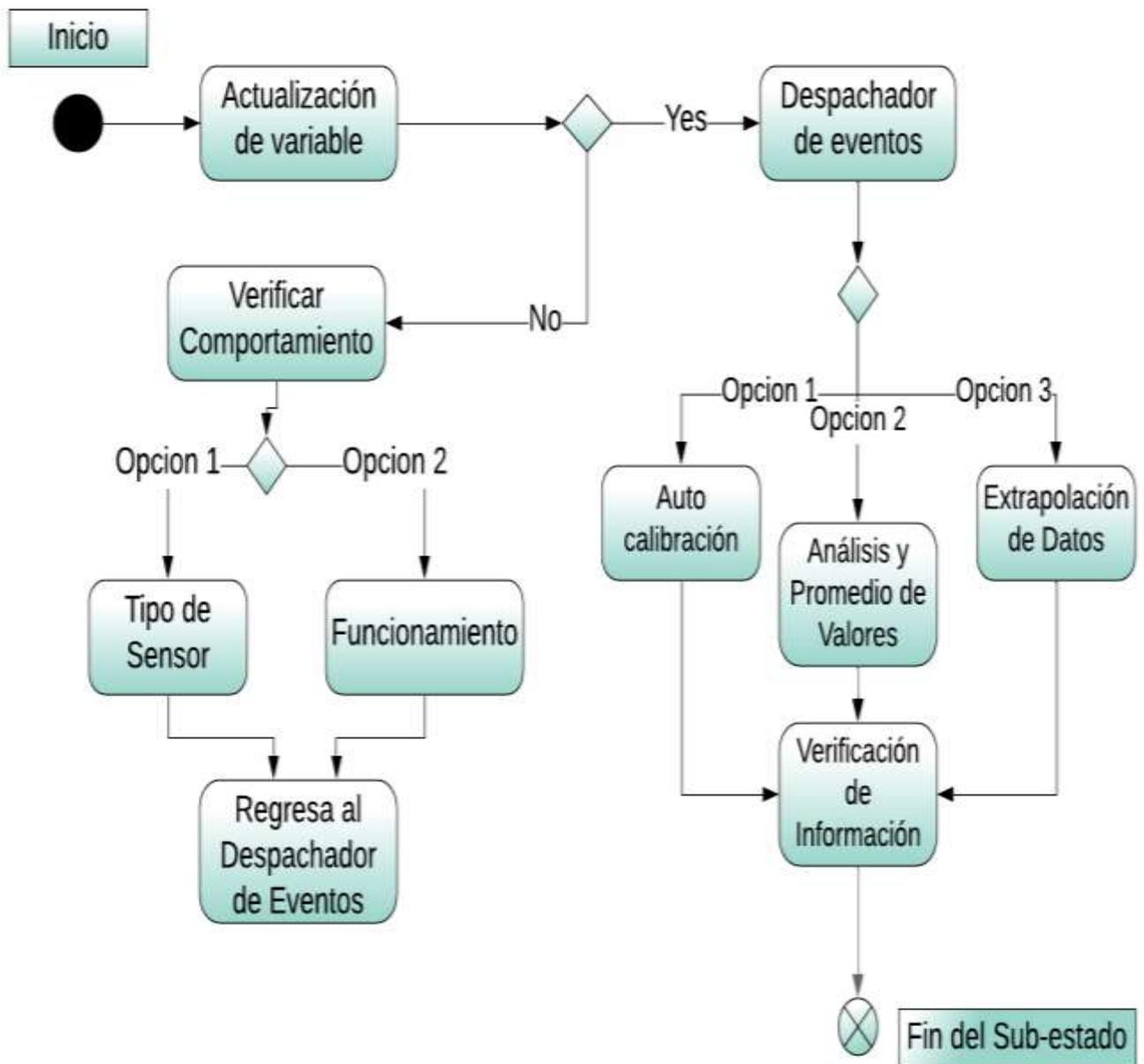


Figura 3. 16: Diagrama de Estados del Sub- proceso Selección de Comportamiento
Elaborado por: Autor

3.4. Diseño de la Red de Petri Nodo Sensor/ Actuador

Las Redes de Petri son utilizadas para representar de forma gráfica sistemas de eventos discretos, y para demostrar la topología de sistemas distribuidos, concurrentes o paralelos. Al diseñar la RP (Red de Petri) del funcionamiento del sensor inteligente, primero se realizó el diagrama de Flujo para tener más claro de la manera se va a comportar el sistema. Este diagrama presenta los estados que tiene el sistema, las decisiones que toma y los sub-estados que se generan a partir de estas preguntas que se encuentran ejemplificadas en la figura 3.17.

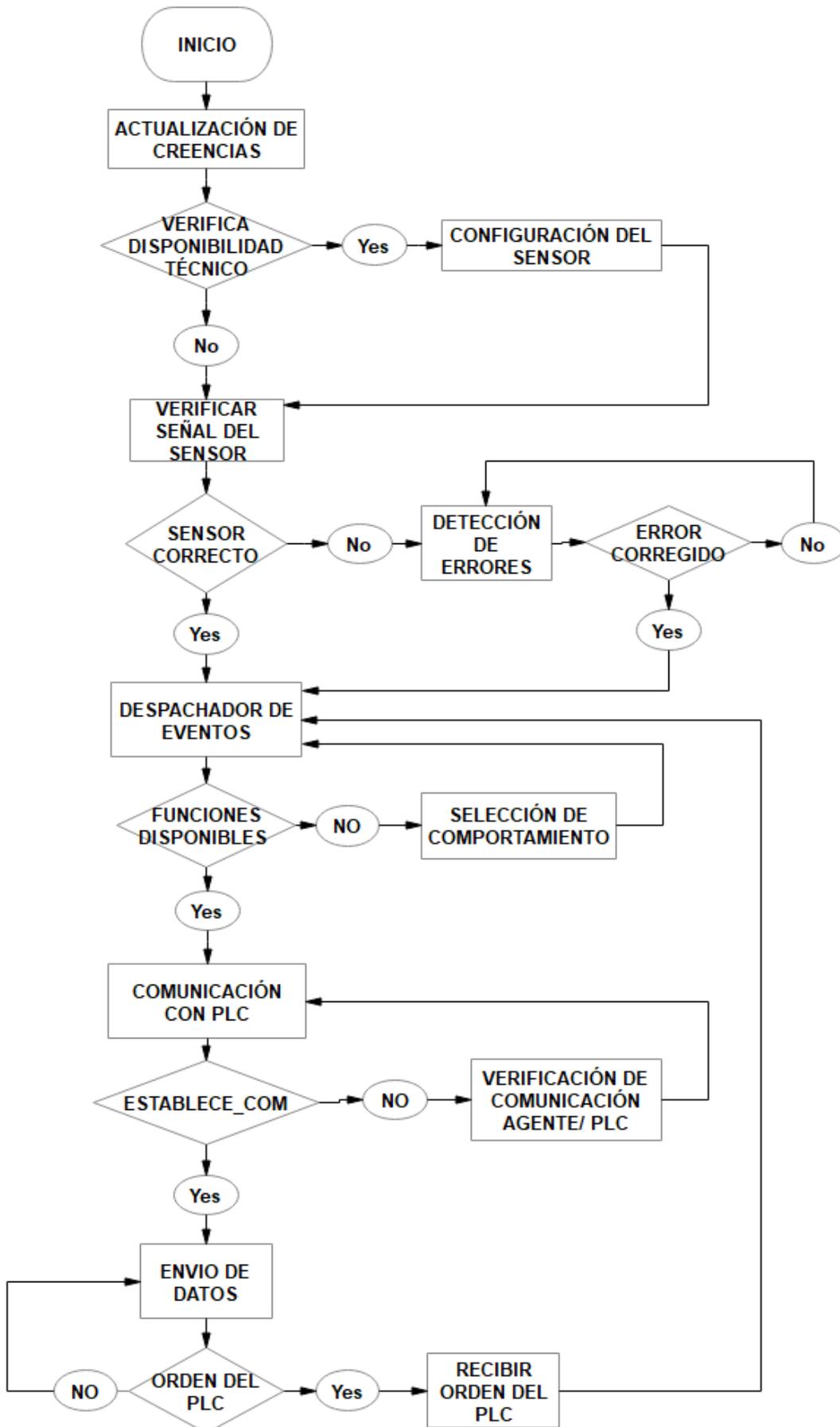


Figura 3. 17: Diagrama de Flujo del Sensor Inteligente
Elaborado por: Autor

3.4.1 Red de Petri del Sensor Inteligente

El diseño de la Red de Petri, se basa en dos diagramas el de flujo de la figura 3.18 y el diagrama de estados de UML de la figura 3.13, se los utiliza para determinar la forma de comportarse el sistema, y la secuencia que va a tener.

Esta red se compone de 6 lugares principales, que están representados en el diagrama de flujo como estados y en la red son los círculos de tamaño normal estos lugares son:

- Inicio, es el momento en que el sistema se energiza y comienza a estabilizarse, para pasar al siguiente estado.
- Actualización de Creencias, se define que variable es la que va a medir, ya sea caudal o nivel, solo será una a la vez debido a la arquitectura previamente diseñada.
- Verificar Señal del Sensor, establece un enlace de comunicación Agente/Sensor, si todo está correcto pasa al siguiente estado.
- Despachador de Eventos, en este estado se cumplen todas las funciones del sistema que ya están establecidas.
- Comunicación con PLC, establece un enlace de comunicación Agente/Autómata, a través del protocolo MODBUS.
- Envío de Datos, se envían los datos proporcionados por el agente inteligente, de forma fiable y segura.

En la figura 3.18 se muestra la red de Petri general del sistema, en la cual se muestra todos los elementos previamente descritos, todos sus estados y transiciones están vivas, por lo que genera invariantes del sistema que son analizadas, en la sección 3.4.2.

Se compone de 5 transiciones importantes que son representadas en el diagrama de flujo como decisiones del sistema. Estas transiciones son:

- Verificar Disponibilidad de Técnico, enlace de comunicación Agente/ Técnico, en caso de que este activo.
- Sensor Correcto, verifica si las señales del sensor están correctas, con esto continua su flujo normal.
- Funciones Disponibles, todas las funciones ya establecidas previamente son ejecutadas por el Agente inteligente.

- Establece Comunicación, establece comunicación con PLC, y verifica si es fiable el envío de datos.
- Orden del PLC, en caso de que el PLC emita una orden al Agente, aquí revisa que sea datos fiables y no basura.

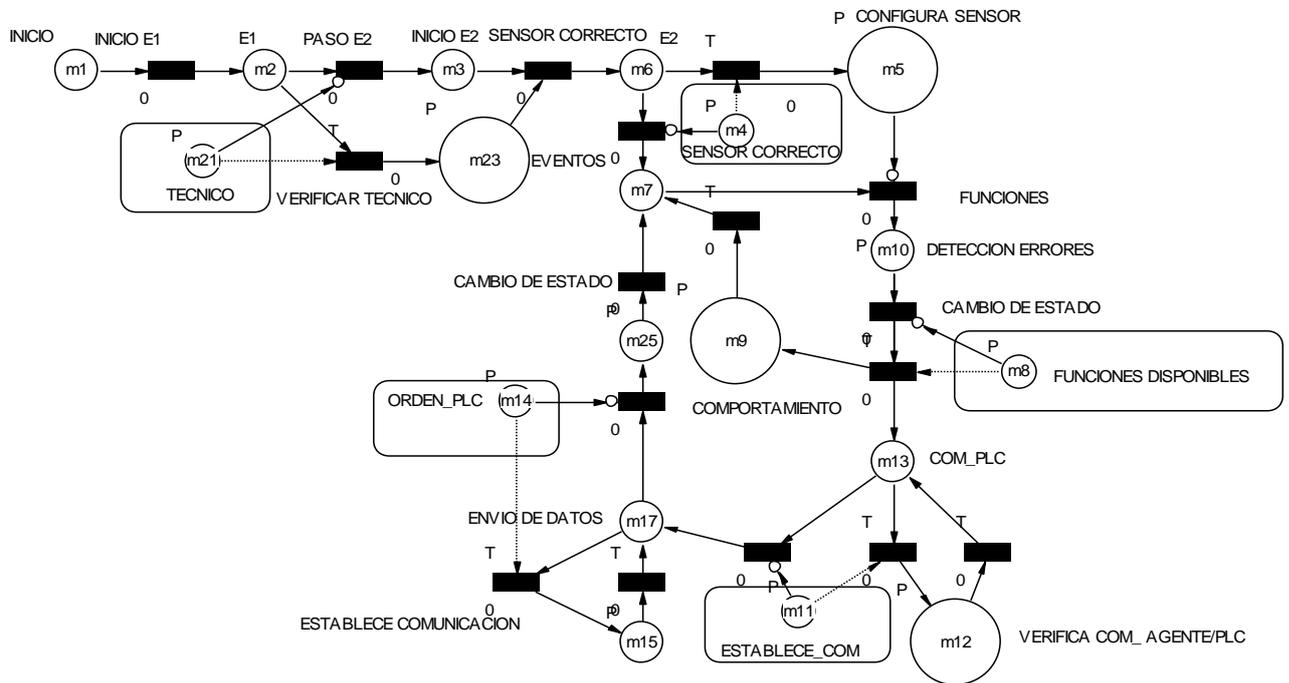


Figura 3. 18: Red de Petri del Sensor Inteligente
Elaborado por: Autor

Los 5 sub-estados de la Red de Petri, están representados por los círculos de mayor tamaño y son:

- Configuración del Sensor, en caso de que el técnico si esté disponible, se utiliza este sub-estado para configurar los datos del sensor y pasa al estado 2.
- Detección de Errores, si el sensor no está correcto, en este sub-estado se verifica la señal del sensor, si el error está corregido pasa al estado 3, si no lo está se queda en ese bucle hasta que se corrija.
- Selección de Comportamiento, se lo utiliza para escoger de que manera se va a comportar el sensor, ya sea por tipo del sensor o si es el sensor inteligente principal o el de respaldo, pasa al estado 3 una vez que se termine el sub-estado.
- Verificación de Comunicación Agente/PLC, en caso de que no se haya establecido un enlace de comunicación, se tiene este bucle que verifica que

esté conectado, si lo está pasa al estado 4, caso contrario se queda esperando que exista una comunicación.

- Recibir Orden del PLC, si existe una orden que haya emitido el PLC, pasa a ser procesada en el estado 3.

3.4.2 Redes de Petri de los Sub-estados del Flujo General

La red de Petri general contiene 4 sub-estados que son:

- Configuración del Sensor
- Configuración Agente/PLC
- Detección de Errores
- Selección de Comportamiento

Todos estos sub-estados tienen su Red de Petri, que fueron realizadas y simuladas en Visual Object Net.

3.4.2.1 Red de Petri del Sub-estado: Configuración del Sensor

La red de Petri del sub-estado Configuración del Sensor está compuesta por 9 sub-estados, los que son:

- Inicio, lugar del flujo general que da inicio a este sub-estado.
- Verificar Enlace de Comunicación, establece un enlace de comunicación con el sensor.
- Verificar Conectividad, comprueba si el sensor está activo, y preparado para enviar o recibir datos.
- Configurar Sensor, el agente carga las configuraciones del sensor para comenzar la operación.
- Verificar Creencia, verifica que variable mide el sensor, de acuerdo a los rangos de operación.
- Actualiza la Creencia, indica a todo el sistema cual es la nueva variable que va a medir.
- Niega la Información, en caso de no estar correcto la configuración del sensor, no permite el paso de datos y pregunta que verifiquen otra vez la creencia.
- Confirma la Información, si todo está bien, pasa los nuevos rangos de operación para que se ejecuten las operaciones del sistema.
- Paso al Estado 3.

En la figura 3.19 se muestra la red de Petri para el sub-estado Configuración del Sensor.

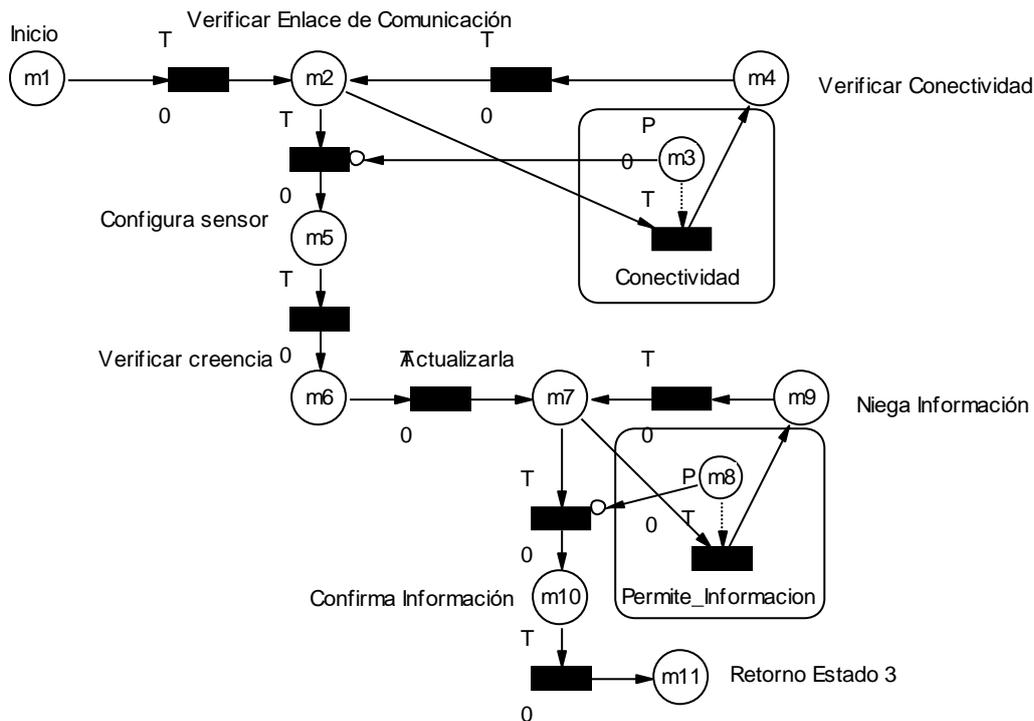


Figura 3. 19: Red de Petri del Sub-estado: Configuración del Sensor
Elaborado por: Autor

3.4.2.2 Red de Petri del Sub-estado: Configuración Agente/PLC

La red de Petri del sub-estado Configuración Agente/PLC está compuesta por 10 sub-estados, los que son:

- Inicio, lugar del flujo general donde comienza este sub-estado.
- Verificar Enlace de Comunicación, establece un enlace de comunicación con el PLC.
- Enviar Comando, si existe respuesta por parte del PLC, se envía un comando.
- Verificar Comunicación, si la comunicación con el PLC fue interrumpida, pregunta de nuevo y regresa a verificar nuevamente la comunicación.
- Interpretación del Comando, una vez que se envió el comando, pasa a interpretarlo para que se ejecuten las acciones del sistema.
- Despachador de Eventos, lugar donde se ejecutan las acciones del sistema.
- Validación y Procesamiento de Información, una vez se realicen las acciones del sistema, se debe validar estos datos para que ingresen nuevamente al PLC.

- Niega la Información, si la información está errónea prohíbe el paso de la misma, y pregunta por un nuevo comando.
- Enviar una Respuesta, una vez validada la información pasa al estado 4.

En la figura 3.20 se muestra la red de Petri para el sub-estado Configuración Agente/PLC.

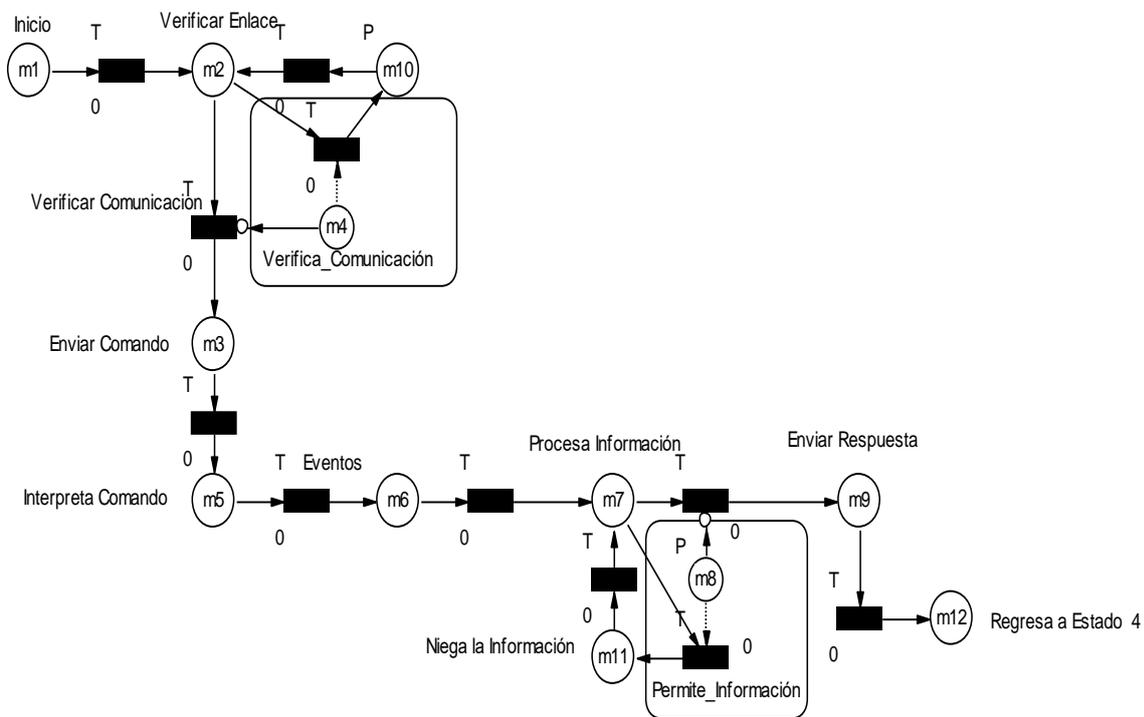


Figura 3. 20: Red de Petri del Sub-estado: Configuración Agente/PLC
Elaborado por: Autor

3.4.2.3 Red de Petri del Sub-estado: Detección de Errores

La red de Petri del sub-estado Detección de Errores está compuesta por 10 sub-estados, los que son:

- Inicio, lugar del flujo general que da paso a este sub-estado.
- Información del Entorno, pregunta que variable es la que va a medir el sensor.
- Actualización de variable, indica al sistema general que existe una nueva variable.
- Generar Alarma, en caso de que no se actualice la variable correctamente, genera un aviso de fallo.
- Despachador de Eventos, si la variable se actualizó correctamente cumple las siguientes funciones:

- Auto Calibración
- Análisis y Promedio de Valores
- Extrapolación de Datos
- Verificación y Validación, una vez terminada la ejecución de las funciones, verifica los datos y los envía al siguiente estado.
- Regreso al siguiente estado del flujo general.

En la figura 3.21 se muestra la red de Petri para el sub-estado Detección de Errores.

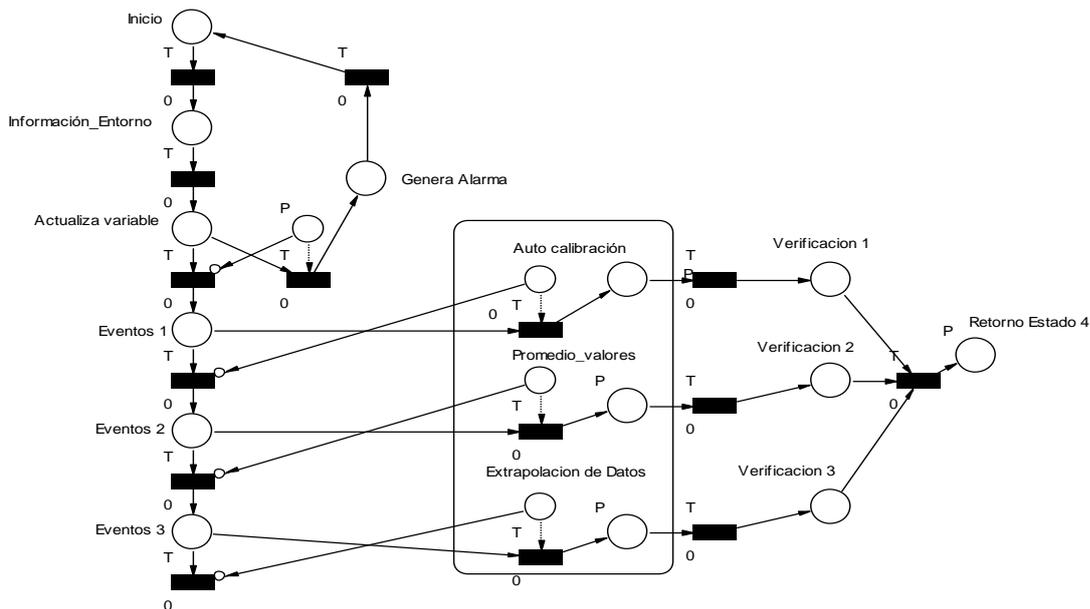


Figura 3. 21: Red de Petri del Sub-estado: Detección de errores
Elaborado por: Autor

3.4.2.4 Red de Petri del Sub-estado: Selección de Comportamiento

La red de Petri del sub-estado Selección de Comportamiento está compuesta por 12 sub-estados, los que son:

- Inicio, lugar del flujo general que da paso a este sub-estado.
- Actualización de variable, indica al sistema general que existe una nueva variable.
- Despachador de eventos, si la variable se actualizó correctamente cumple las siguientes funciones:
 - Auto calibración
 - Análisis y promedio de Valores
 - Extrapolación de Datos

- Verificar Comportamiento, en caso de que la variable no se haya actualizado, se escoge el tipo de comportamiento:
 - Tipo de Sensor
 - Funcionamiento
- Verificación de Información, comprueba que todo este correcto y pasa al siguiente estado.
- Regreso al estado 3

En la figura 3.22 se muestra la red de Petri para el sub- estado: Selección de Comportamiento.

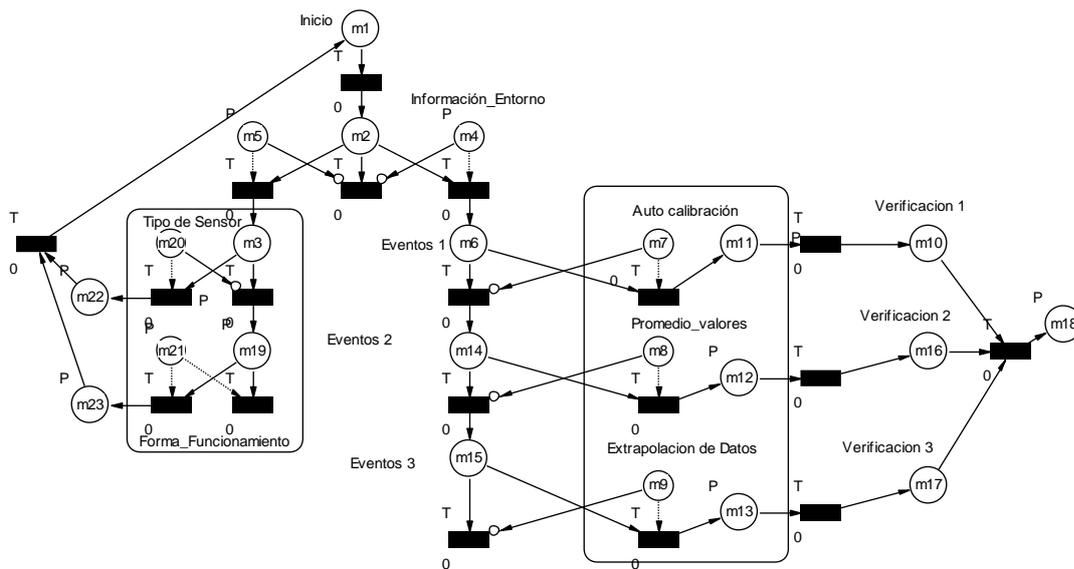


Figura 3. 22: Red de Petri del Sub-estado: Selección de Comportamiento
Elaborado por: Autor

3.4.3 Análisis de las Invariantes de la Red de Petri

Las ecuaciones de la red de Petri general se obtuvieron en el software PIPE 4.3.0, así también para cada sub- red conformada por los sub-estados del flujo general. En esta sección se realiza un análisis de las ecuaciones que se obtuvieron para cada red.

3.4.3.1 Análisis de las Invariantes de la Red de Petri General

Una vez realizada la red, el siguiente paso es la generación de invariantes de la red, para esto se establecen dos tablas diferentes, para las transiciones y lugares del sistema. En la tabla 3.1 se tiene los resultados de las invariantes para las transiciones.

Tabla 3. 1 Invariantes de las Transiciones: RdP General (Autor)

T0	T1	T10	T11	T12	T13	T14	T15	T16	T17	T2	T3	T4	T5	T6	T7	T8	T9	T18
0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1	0

En la tabla 3.2 se tiene los resultados de las invariantes para los lugares de la Red general.

Tabla 3. 2 Invariantes de los Lugares: RdP General (Autor)

P	P	P1	P	P	P	P	P	P	P	P								
0	1	0	1	2	3	4	5	6	7	8	2	3	4	5	6	7	8	9
1	1	0	1	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1

En la ecuación 5 se muestra el resultado de las invariantes:

$$M(P0) + M(P1) + M(P11) + M(P12) + M(P14) + M(P15) + M(P16) + M(P18) + M(P3) + M(P4) + M(P5) + M(P7) + M(P8) + M(P9) = 1 \quad (5)$$

La red de Petri general tiene diversas interrupciones que generan los sub-estados, pero la ecuación 5 demuestra que el sistema se mantiene estable, sea en la interrupción que se encuentre o no estén activadas ninguna de estas interrupciones. Para todos los lugares P0, P1, P11, P12, P14, P15, P16, P18, P3, P4, P5, P7, P8, P9, solo existe un único token que activa todo el sistema, lo que indica que controla una variable a la vez, y que solo existe un estado de operación para toda la red.

3.4.3.2 Análisis de las Invariantes del Sub-estado 1

Al realizar la red general, se tienen interrupciones que generan sub-estados, los que también generan invariantes para su respectiva red, para esto se tiene dos tablas diferentes, para las transiciones y lugares del sistema, respectivamente para cada sub-estado. En la tabla 3.3 se tiene los resultados de las invariantes para las transiciones del sub-estado 1.

Tabla 3. 3 Invariantes de las Transiciones: Sub-estado1 (Autor)

T0	T1	T10	T11	T2	T4	T5	T6	T8	T9	T12
1	0	0	0	1	1	1	1	1	0	1

En la tabla 3.4 se tiene los resultados de las invariantes para los lugares del Sub-estado 1.

Tabla 3. 4 Invariantes de los Lugares: Sub-estado 2 (Autor)

P0	P1	P10	P12	P2	P4	P5	P6	P7	P8	P9
1	1	1	1	0	1	1	1	1	0	1

En la ecuación 6 se muestra el resultado de las invariantes:

$$M(P0) + M(P1) + M(P10) + M(P12) + M(P4) + M(P5) + M(P6) + M(P7) + M(P9) = 1 \quad (6)$$

El sub-estado 1 tiene dos interrupciones o decisiones que el Agente debe tomar si se encuentra en este lugar, la ecuación 6 demuestra que el sistema se mantiene estable, sea en la interrupción que se encuentre o no estén activadas ninguna de estas interrupciones. Para todos los lugares P0, P1, P10, P12, P4, P5, P6, P7, P9, solo existe un único token que activa todo el sistema, lo que indica que controla una variable a la vez, y que solo existe un estado de operación para toda la red.

3.4.3.3 Análisis de las Invariantes del Sub-estado 2

Al realizar la red general, se tienen interrupciones que generan sub-estados, los que también generan invariantes para su respectiva red, para esto se tiene dos tablas diferentes, para las transiciones y lugares del sistema, respectivamente para cada sub-estado. En la tabla 3.5 se tiene los resultados de las invariantes para las transiciones del sub-estado 2.

Tabla 3. 5 Invariantes de las Transiciones: Sub- estado 2 (Autor)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
1	1	0	0	1	1	1	1	0	1	0	1

En la tabla 3.6 se tiene los resultados de las invariantes para los lugares del Sub-estado 2.

Tabla 3. 6 Invariantes de los Lugares: Sub-estado 2 (Autor)

P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
1	1	0	1	1	1	1	1	0	1	1	1

En la ecuación 7 se muestra el resultado de las invariantes:

$$M(P0) + M(P1) + M(P3) + M(P4) + M(P5) + M(P6) + M(P7) + M(P9) + M(P10) + M(P11) = 1 \quad (7)$$

El sub-estado 2 tiene dos interrupciones o decisiones que el Agente debe tomar si se encuentra en este lugar, la ecuación 7 demuestra que el sistema se mantiene

estable, sea en la interrupción que se encuentre o no estén activadas ninguna de estas interrupciones. Para todos los lugares P0, P1, P3, P4, P5, P6, P7, P9, P10, P11 solo existe un único token que activa todo el sistema, lo que indica que controla una variable a la vez, y que solo existe un estado de operación para toda la red.

3.4.3.4 Análisis de las Invariantes del Sub-estado 3

Al realizar la red general, se tienen interrupciones que generan sub-estados, los que también generan invariantes para su respectiva red, para esto se tiene una tabla para los lugares del sistema, respectivamente para este sub-estado. En la tabla 3.7 se tiene los resultados de las invariantes para los lugares del sub-estado 3.

Tabla 3. 7 Invariantes de los Lugares: Sub-estado 3 (Autor)

P0	P1	P2	P3	P6	P7	P8	P11	P14	P17	P16	P15	P13	P12	P10	P9	P5	P4	P18	P19
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0

El sub-estado 3 tiene una interrupción que despliega un menú donde el Agente debe realizar las funciones que se encuentran disponibles, las ecuaciones 8-9-10 demuestran que el sistema se mantiene estable, sea en la función que se encuentre. En las ecuaciones 8-9-10 se muestra el resultado de las invariantes:

$$M(P8) + M(P11) + M(P14) + M(P17) = 1 \quad (8)$$

$$M(P17) + M(P16) + M(P13) + M(P10) = 1 \quad (9)$$

$$M(P17) + M(P15) + M(P12) + M(P9) = 1 \quad (10)$$

3.4.3.5 Análisis de las Invariantes del Sub-estado 4

Al realizar la red general, se tienen interrupciones que generan sub-estados, los que también generan invariantes para su respectiva red, para esto se tiene una tabla para los lugares del sistema, respectivamente para este sub-estado. En la tabla 3.8 se tiene los resultados de las invariantes para los lugares del sub-estado 4.

Tabla 3. 8 Invariantes de los Lugares: Sub-estado 4 (Autor)

P0	P1	P10	P11	P12	P13	P14	P15	P16	P17	P20	P22	P23	P24	P25	P26	P27	P3	P4	P5	P8	P9	P28
0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

El sub-estado 4 tiene dos interrupciones que despliega un menú donde el Agente debe realizar las funciones que se encuentran disponibles, las ecuaciones 11-12-13 demuestran que el sistema se mantiene estable, sea en la función que se encuentre.

En las ecuaciones 11-12-13 se muestra el resultado de las invariantes:

$$M(P10) + M(P13) + M(P16) = 1 \quad (11)$$

$$M(P11) + M(P14) + M(P8) = 1 \quad (12)$$

$$M(P12) + M(P15) + M(P9) = 1 \quad (13)$$

3.5. Trama del Protocolo MODBUS

La trama ideal del protocolo MODBUS para este caso de estudio, es de tipo RTU ya que emite 8 bits de información, lo que se ajusta para la emisión y recibo de datos desde el Microcontrolador al PLC y viceversa.

La trama de MODBUS contiene la Unidad de Datos de Aplicación (ADU) y una Unidad de Datos de Protocolo (PDU), la que se conforma por:

- ADU= Dirección más PDU más la comprobación de errores.
- PDU= Código de función más los datos que se van a emitir.

Está comunicación es de tipo serial, y utiliza una compactación binaria de los datos que le permite el envío más rápido de la información. Por esto es el protocolo de comunicación que más le conviene al sistema para el envío y recibo de datos de forma fiable.

A continuación, en la tabla 3.9 se muestra la trama de MODBUS RTU:

Tabla 3. 9 Trama MODBUS RTU (Autor)

Con chequeo de paridad										
Inicio	1	2	3	4	5	6	7	8	Par	Paro
Sin chequeo de paridad										
Inicio	1	2	3	4	5	6	7	8	Paro	Paro

A continuación, en la tabla 3.10 se describe la trama de MODBUS RTU:

Tabla 3. 10 Trama MODBUS RTU (Autor)

Nombre	Longitud en bits	Funcionamiento
Inicio	28	Por lo menos 3 ½ tiempos de silencio
Direccionamiento	8	Dirección de Estado
Funcionamiento	8	Indica código de funcionamiento
Datos	N*8	Datos + longitud de que cada mensaje

		que se rellena automáticamente
CRC	16	Verificación de redundancia cíclica
Final	28	Por lo menos 3 ½ tiempos de silencio entre cada trama

3.6. Pruebas Experimentales del Sensor Inteligente

Las pruebas experimentales se las utiliza para comprobar si la arquitectura planteada es la correcta, y si el sistema cumple con sus funciones que son el conjunto de condiciones que debe cumplir el sensor para ser inteligente, en el control del nivel o caudal. En la Imagen 3.23 se muestra la estructura utilizada para controlar el nivel de agua y en la figura 3.24 está la estructura para controlar el caudal del agua.



Figura 3. 23: Pruebas con Nivel de Agua
Elaborado por: Autor

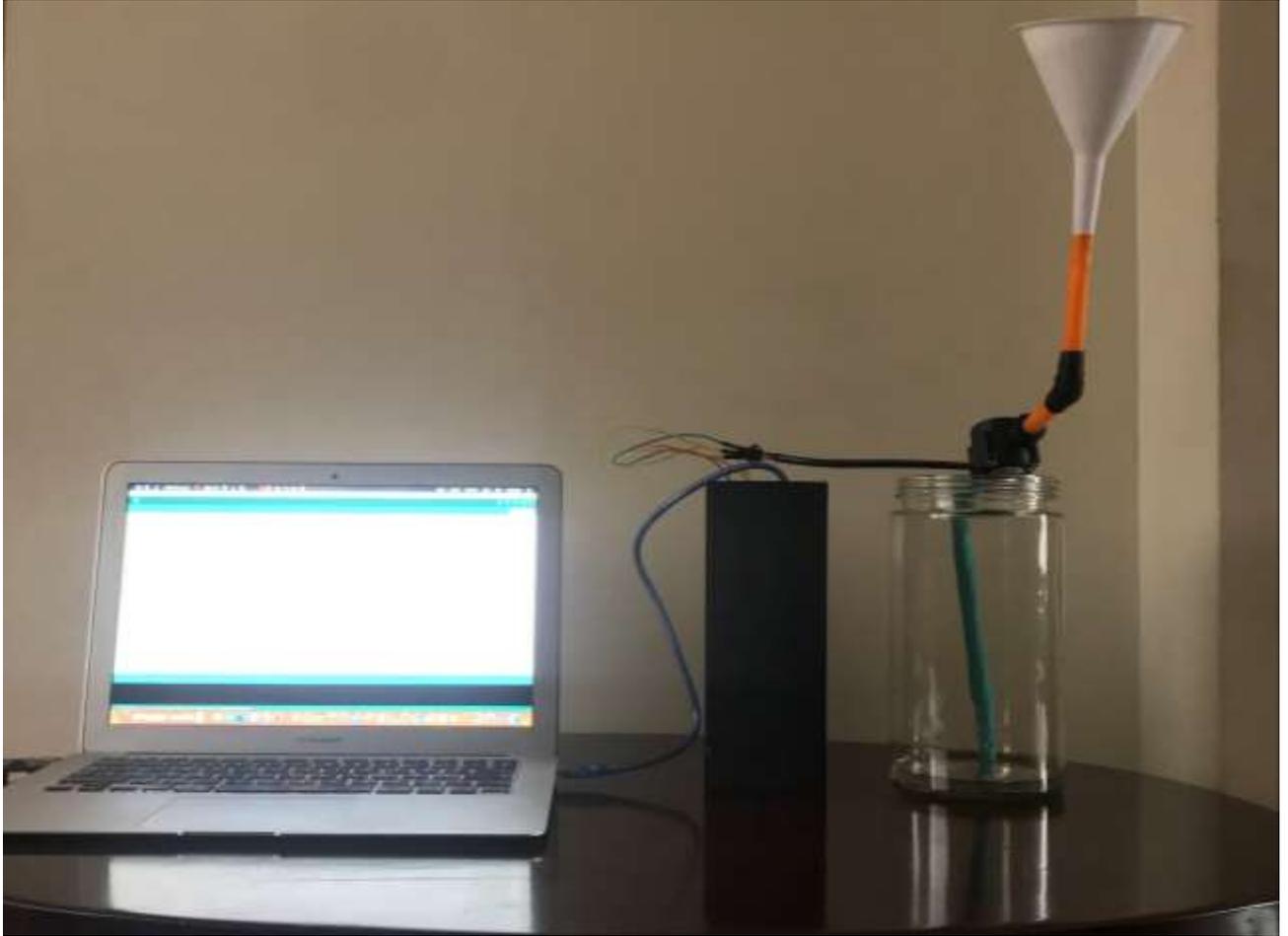


Figura 3. 24: Pruebas con Caudal de Agua
Elaborado por: Autor

3.5.1 Prueba Experimental de Nivel

Objetivo de la Prueba: Demostrar que el sistema se mantiene estable a medida que el agua incrementa, con tiempo de respuesta mínimo.

Variables que Intervienen: Nivel de Agua.

Desarrollo del Experimento: Para esta prueba se tiene un reservorio de 20cm de alto que es donde se va a ejecutar el control del nivel, un sensor ultrasónico HC-SR04, la tarjeta Arduino Mega 2560, una pantalla Oled 128*64 pixeles donde se aprecia los valores del nivel en tiempo real.

Para esta prueba se obtiene el tiempo que tarde en emitir una medida a la otra mientras el agua se emite constantemente. Para comprobar si el sistema se mantiene estable, y se adapta a la situación conforme el tiempo transcurre.

Análisis de los Resultados: En la siguiente tabla 3.11 se muestran los tiempos de estabilización del sistema para los diferentes niveles de agua. La media entre todos estos valores es de 47.2 milisegundos de estabilización entre una medida a la otra.

Tabla 3. 11 Tiempo de Estabilización del Nivel (Autor)

Nivel cm	Tiempo de Estabilización
18	16 mseg
12	20mseg
6	34mseg
3	40mseg
1	36mseg

Este tiempo de estabilización varía según el flujo constante del agua en el reservorio de control, lo que arroja un tiempo de respuesta del sistema óptimo y fiable. Este sistema es estable y trabaja de manera eficiente.

La fórmula 14 es la que se utiliza para hallar la media:

$$\sum \text{ los valores de muestra } / \text{ para el valor total de muestras } = \text{ media } \quad (14)$$

La media entre los valores del tiempo de estabilización es de 47.2 milisegundos, que es el tiempo de respuesta del sensor en el sistema.

En la imagen 3.25 se observa la gráfica obtenida en la primera y segunda prueba.

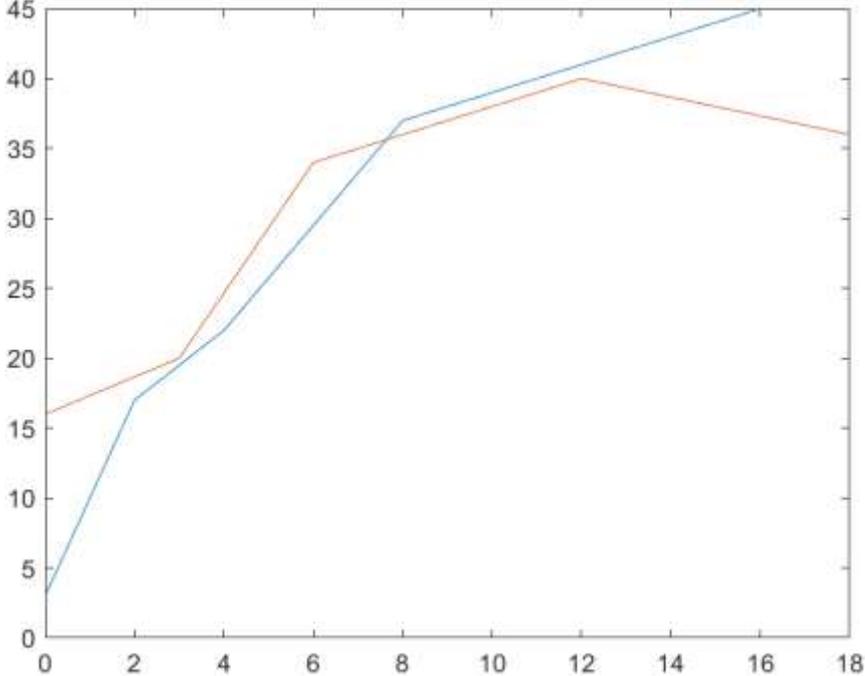


Figura 3. 25: Gráfica Nivel vs Tiempo
Elaborado por: Autor

En la imagen 3.26 se observa la gráfica obtenida de todas las pruebas.

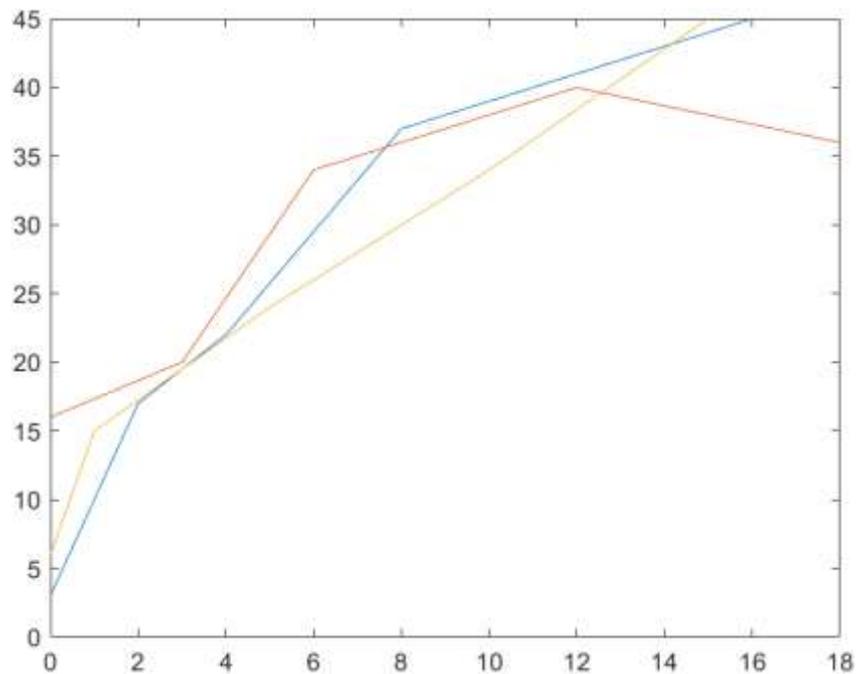


Figura 3. 26: Gráfica Nivel vs Tiempo
Elaborado por: Autor

En la siguiente tabla 3.12 se muestran los valores arrojados por el sensor inteligente versus los valores reales del reservorio de control. Esto se realiza con el fin de obtener un porcentaje de error del sensor, el cual es de $\pm 0.15\%$.

Tabla 3. 12 Porcentaje de Error Nivel (Autor)

	SENSOR INTELIGENTE	VALOR REAL
	12	11.99
	11	10.99
	10	9.99
	9	8.99
	8	7.99
	7	6.99
	6	5.99
	5	4.99
	4	3.99
	3	2.99
	2	1.99
	1	0.99
TOTAL	78	77.88
PROMEDIO	6.5	6.49
ERROR	0.15%	

3.5.2 Prueba Experimental de Caudal

Objetivo de la Prueba: Demostrar que el sistema se mantiene estable a medida que el agua incrementa, con tiempo de respuesta mínimo.

Variables que Intervienen: Caudal de Agua.

Desarrollo del Experimento: Para esta prueba se tiene un reservorio de 20cm de alto que es donde se va a ejecutar el control del caudal, un sensor de flujo YF-S201, la tarjeta Arduino Mega 2560, una pantalla Oled 128*64 pixeles donde se aprecia los valores del caudal en tiempo real.

Para esta prueba se obtiene el tiempo que tarde en emitir una medida a la otra mientras el agua se emite constantemente. Para comprobar si el sistema se mantiene estable, y se adapta a la situación conforme el tiempo transcurre.

Análisis de los Resultados: En la siguiente tabla 3.13 se muestran los tiempos de estabilización del sistema para las diferentes medidas de Caudal. La media entre todos estos valores es de 37 milisegundos de estabilización entre una medida a la otra.

Tabla 3. 13 Tiempo de Estabilización del Nivel (Autor)

Frecuencia Hz.	Caudal Lt/min
19	3.16
13	2.16
12	2
11	1.83
10	1.67
9	1.5
8	1.33
7	1.16

Este tiempo de estabilización varía según el flujo constante del agua en el reservorio de control, lo que arroja un tiempo de respuesta del sistema óptimo y fiable. Este sistema es estable y trabaja de manera eficiente.

La media entre los valores del tiempo de estabilización es de 37 milisegundos, que es el tiempo de respuesta del sensor en el sistema.

En la imagen 3.27 se observa la gráfica obtenida en la primera prueba que se realizaron al caudal de un fluido en este caso fue agua.

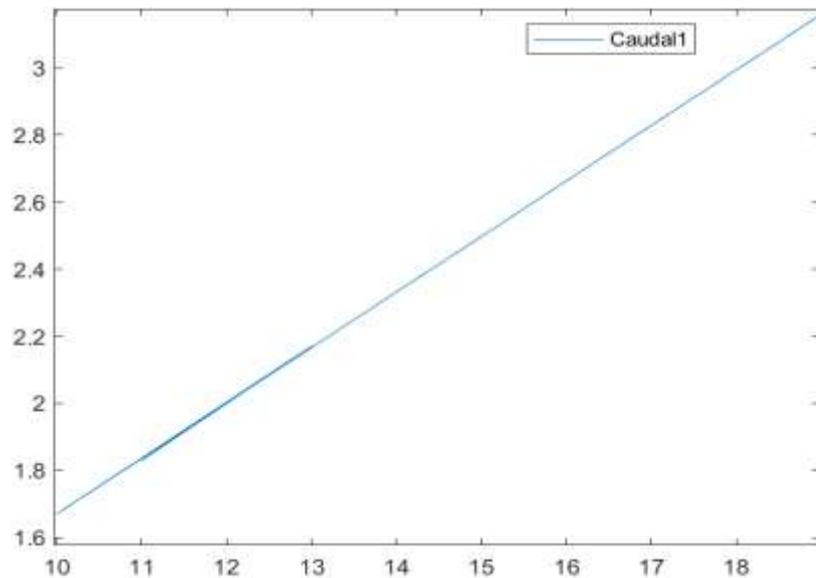


Figura 3. 27: Gráfica Caudal vs Frecuencia
Elaborado por: Autor

En la imagen 3.28 se observa la gráfica obtenida de todas las pruebas.

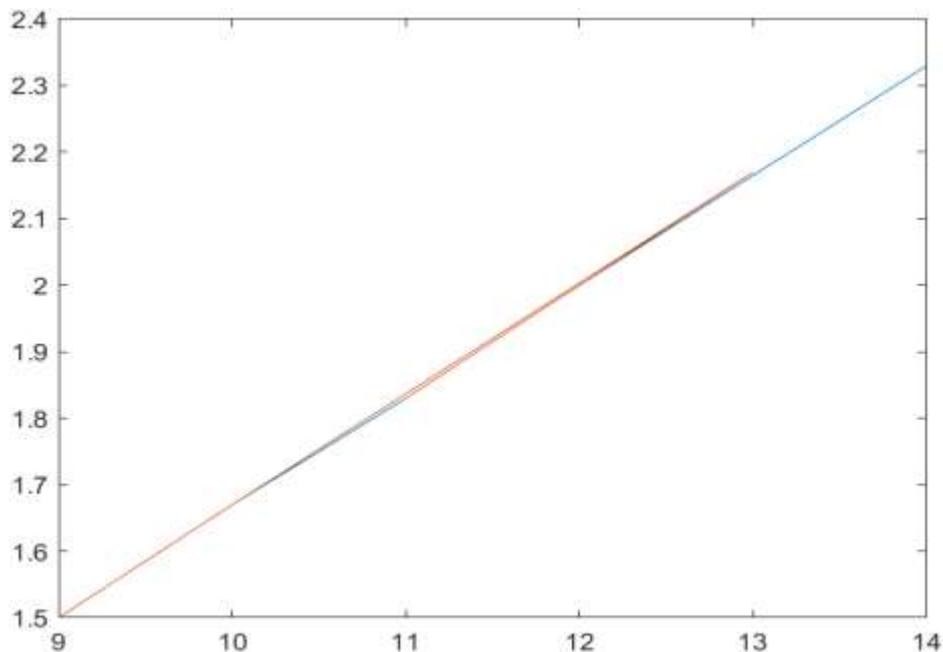


Figura 3. 28: Gráfica Caudal vs Frecuencia
Elaborado por: Autor

En la siguiente tabla 3.14 se muestran los valores arrojados por el sensor inteligente versus los valores reales del reservorio de control. Esto se realiza con el fin de obtener un porcentaje de error del sensor inteligente de caudal, el cual es de $\pm 2.5\%$.

Tabla 3. 14 Porcentaje de Error Caudal (Autor)

	Sensor Inteligente	Valor Real
	3,16	3,5
	2,16	3
	2,00	2,5
	1,83	2
	1,67	1,5
	1,50	1
	1,33	0,5
Total	13,65	14
Promedio	1,95	2
Error total	2,56%	

El porcentaje de error obtenido, es bajo en comparación con otros trabajos científico, ya que el porcentaje de precisión del sensor de caudal Yf-s201 es de $\pm 10\%$. En siguiente figura 3.29 se muestra una comparación del porcentaje de error obtenido en trabajos de la comunidad científica versus el porcentaje de error que se obtuvo en esta investigación.

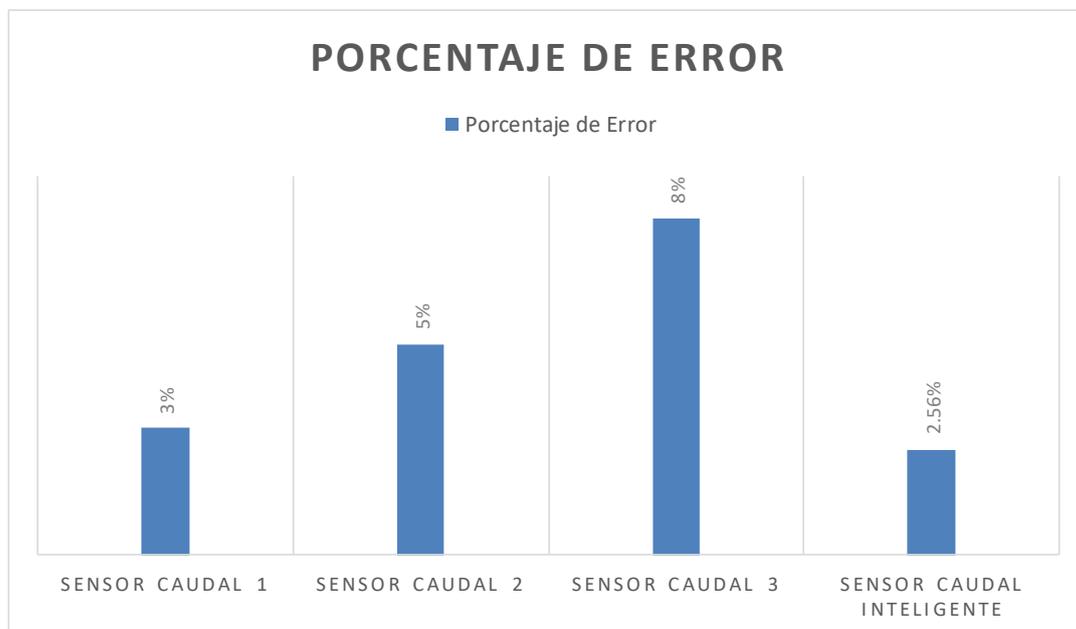


Figura 3. 29: Sensor No inteligente vs Sensor Inteligente
Elaborado por: Autor

CAPÍTULO IV: PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS

Este capítulo contiene todos los resultados obtenidos del presente trabajo de investigación, también se demuestra lo que se ha logrado en base a los objetivos planteados, se analiza las características que posee el sistema y se compara estos resultados versus otros trabajos de investigación.

4.1 Presentación de Resultados

En este sub capítulo se presentan los resultados obtenidos en el estado de arte, en la arquitectura planteada para el sensor inteligente, los diferentes diagramas de UML, la red de Petri planteada conforme a lo establecido previamente y las reglas de Lógica Difusa que se obtuvo para nivel y caudal.

4.1.1 Presentación de Resultados relacionados al Estado de Arte

El estado de arte está conformado por varios trabajos de investigación referentes al tema sensores inteligentes, que han servido como plataforma para el desarrollo de este proyecto de investigación.

Del estado de arte se obtiene que existen diferentes aplicaciones para los sensores inteligentes, pero ningún ofrece una arquitectura nodo sensor/ actuador, o alguna que utilice agentes inteligentes para modelar el sistema con sus funciones, entradas y salidas. Por otro lado, se tiene la diferencia entre un sistema de sensado inteligente y lo que realmente es un sensor inteligente, dicha diferencia radica en que el sensor junto con el microcontrolador son un solo dispositivo, es decir el microcontrolador se dedica a ejecutar tareas o acciones que son del sensor y que permiten que trabaje de forma eficiente. Un sistema de sensado inteligente es aquel que posee el sensor con el microcontrolador por separado y se dedica a ejecutar diversas tareas, sin darle prioridad al sensor.

Las funciones del sensor inteligente son diversas, existen varios trabajos en que este tipo de sensor ha sido utilizado para el control de la turbidez del agua o para redes de nodos con sensores para la producción del agua bebible, si bien es cierto el caso de estudio es un sub- sistema de filtrado que va a producir agua potable de servicio, pero si es factible para el control de nivel o caudal de acuerdo a los artículos que se encuentran mencionados en este trabajo.

Las ventajas de implementar un sistema industrial con sensores inteligentes, es la transmisión de datos de forma fiable, aporta con información en tiempo real, permite que el sistema tome decisiones que vayan acorde a los parámetros que tenga establecidos y permite procesar los datos de forma distribuida.

Los diagramas de UML han sido más utilizados en el desarrollo de software, ya que hay muy poca información acerca del uso de estos diagramas para el modelado de un sistema industrial.

El protocolo de comunicación MODBUS es la forma más fiable de comunicar Arduino con un PLC, es un protocolo amigable y el más utilizado en diversos trabajos que han decidido experimentar en este campo de la industria 4.0.

4.1.2 Resultados obtenidos en Hardware

Después de varias pruebas realizadas, se determinó que la tarjeta Arduino mega 2560 es una de las más indicadas para el procesamiento de los datos del sensor, el sensor que se utiliza para el control del nivel es el ultrasónico HC – SR04, es el más indicado para las pruebas experimentales que se realizaron. El sensor de caudal YF-S201 fue implementado para realizar las pruebas experimentales, debido a la arquitectura que posee este sistema, que es bastante flexible para la variable que se quiera controlar. Los rangos que se establecieron para el control de Nivel y Caudal son: 0-10 lt/seg y 0-100cm, estos valores van establecidos en el código de programación, que va embebido en la tarjeta Arduino. Debido a la diferencia de voltaje que tienen el transmisor de caudal se utiliza una fuente de 24 voltios, con un divisor de voltaje para conectar la tarjeta Arduino junto con el sensor.

Para mayor control del sistema en tiempo real se utiliza una pantalla Oled de 128 * 64 de 96", está pantalla transmite en tiempo real el nivel que este midiendo en ese momento. Las pruebas experimentales que se realizaron son solo del control de nivel.

4.1.3 Resultados obtenidos en Software

En el desarrollo del software del sistema, se realizó un estudio previo en el cual se la elaboró una arquitectura nueva, utilizando agentes inteligentes del tipo deliberativos BDI, esto permite que sea más flexible para las variables que se quiera controlar, solo que debe ser una por una, no todas a la vez debido al diseño de esta arquitectura. Una vez obtenida esta arquitectura se realiza el modelado del sistema en UML, con diferentes diagramas, comenzando por los casos de uso, donde se

ejemplifica cada uno de los actores o dispositivos que intervienen en el sistema, y que actividades o casos de uso tienen que realizar cada uno o entre ellos. El siguiente diagrama es el de secuencia, el cual da un panorama más amplio en cuanto a la forma de cómo va a responder o trabajar el sistema, que va a realizar primero o después, dando paso al tercer diagrama utilizado que es el de estados, este da una idea mucho más clara y completa del sistema. El sistema primero tiene que actualizar la variable que está controlando, para continuar con el flujo general, y se va a encontrar con decisiones que llevan a sub-procesos que tienen sus propios diagramas de estado, donde se encuentran todas las actividades que realizan hasta volver a uno de los estados del flujo general. Todo esto se utiliza para modelar la Red de Petri general del sensor inteligente, en esta red se encuentran los estados y como actúan cada uno de ellos al momento de darles una marca o token, esta es la primera simulación que se realiza al sistema para comprobar que la arquitectura y el modelado en UML sean los correctos. Con base a todo lo anterior planteado se procede a realizar las pruebas experimentales, con los resultados de la simulación de la Red de Petri, para el software de programación esto le permite tener más robustez ante perturbaciones, como prevención a fallos ya que el sistema sabe en que momento debe generar una alarma, por último, sirve para mantener los parámetros estables dentro de lo establecido previamente.

4.1.4 Resultados obtenidos en las Pruebas Experimentales

Se realizaron diferentes pruebas tanto para nivel y caudal, de las cuales se obtuvieron resultados varios, en cuanto al tiempo de respuesta de cada sensor para que se establezca el sistema general. Estos resultados son:

- Para el control de nivel el sistema es capaz de estabilizarse tras 47.2 milisegundos, después de ser energizado con el flujo de agua constante.
- El tiempo promedio entre cada una medida a la otra es de 15ms, con el flujo constante de agua.
- Para caudal el sistema se estabiliza en 37 milisegundos, después de ser energizado con la bomba de agua.
- El caudal se lo toma referente a la frecuencia y presión que se emite, comprobando que a mayor altura hay menor presión lo que produce un menor caudal y menor frecuencia. Pero a menor distancia mayor caudal y mayor frecuencia. Los valores del caudal tienden a ser constantes en el tiempo.

- El sistema para las dos variables, puestas a prueba en diferentes momentos se mantiene estable con respecto al tiempo y al flujo de agua que emite la bomba de agua en el caudal y el flujo de agua constante en el nivel.

4.2 Discusión de Resultados

En esta sección se muestran los resultados de forma que las funciones y características cumplan con los objetivos planteados al comienzo del proyecto de investigación, se comprueba si hubo una solución al problema de investigación y si se llevó a cabo la metodología previamente establecida. De esta manera se comprueba que el sistema si es viable al momento de ser implementado y como avance en el control de un proceso industrial.

4.2.1 Análisis de las características y funciones del sistema

Las características y funciones que posee el sistema son planteadas y analizadas en el Capítulo III.

La arquitectura que se plantea en este trabajo de investigación es un método nuevo en la industria de control de procesos por el uso de agentes inteligentes del tipo deliberativos BDI. Este tipo de arquitectura permite tener las siguientes funciones como: Auto calibración, Análisis y Promedio de Valores, Extrapolación de Datos, Generación de Alarmas, Comunicación con el Autómata, Interpretación de Comandos y Detección de la Creencia. Las funciones principales que realiza el agente para el sensor, se comprobó en la simulación de la Red de Petri.

Las características del sistema se modelaron en UML en tres diagramas diferentes los que son: de casos de uso, de secuencia y de estados. Los que ayudaron a comprender cuales son los dispositivos que van a intervenir en el sistema, las funciones que cada uno va a realizar y el tiempo que le va a tomar a cada uno llevar a cabo las funciones preestablecidas en la arquitectura. Después de modelar el sistema general en un diagrama de flujo, con los resultados obtenidos en UML, se puede modelar la Red de Petri general del sistema, de la cual se generaron 4 sub-estados o sub-procesos de operación, para esto se realizó el diagrama de estados de cada sub-proceso para comprender el funcionamiento y como intervienen en el sistema general. Al obtener estos diagramas se elaboró la red de Petri para cada sub-estado, de las cuales se obtuvieron diferentes tablas de invariantes para transiciones

y lugares respectivamente, también se obtuvo las ecuaciones para la red general, como para cada sub-red.

Las redes de Petri son una excelente herramienta que permite simular el sistema con sus sub-procesos, para así comprobar si la arquitectura y los diagramas de UML están correctamente planteado o necesitan cambios.

Para el software de programación se utilizó el IDE de Arduino, el cual contiene un ambiente más amigable para el usuario, el código fue embebido en una tarjeta Arduino Mega 2560 de 8 bits de transmisión, que es lo necesario para enviar o recibir información de forma fiable y segura.

La mayoría de los trabajos presentados a inicios del Capítulo II, en el Estado de Arte presentan los agentes inteligentes para diversos usos en el campo industrial, pero ninguno propone una arquitectura basada en sus principios de razonamiento. El uso de UML para modelado de sistemas de control industrial no es una opción para mucho de los trabajos presentados en capítulos posteriores. Lo que convierte al sistema en novedoso e innovador con respecto a otros trabajos ya presentados ante la comunidad científica.

4.2.2 Cumplimiento de los Objetivos

En esta sección se discuten si los resultados se han cumplido conforme a los objetivos planteados en el Capítulo I. Se divide en dos sub-secciones la primera para el Objetivo general y la otra para los objetivos específicos, esto se realiza para analizar de manera objetiva cada uno.

4.2.2.1 Objetivo General

El objetivo principal de este proyecto de investigación es plantear una arquitectura para sensores inteligentes con una comunicación para microcontroladores y autómatas. En base a estas condiciones se plantea una arquitectura basada en agentes inteligentes de tipo deliberativos, con la que se logra medir cualquier variable que el usuario desee, también una comunicación fiable con el autómata a través de un protocolo industrial MODBUS.

4.2.2.2 Objetivos Específicos

Este trabajo de investigación es innovador no solo por la arquitectura que se presenta, si no por el uso de herramientas que son utilizadas por la rama de Ingeniería

en Sistemas, la herramienta de modelado UML para un sistema de control industrial permite comprender con más detalle el funcionamiento del sistema y de que manera intervienen cada uno de los dispositivos junto con las actividades que le toca realizar respectivamente.

La práctica no estuvo tan alejada de la teoría, ya que siempre todo se basa en un estado ideal, pero el sistema al momento de ser implementado funcionó de excelente manera, se mantuvo estable y con tiempo de respuesta mínimo, lo que hace que sea robusto ante las perturbaciones que se pueden generar dentro del sistema general.

4.2.3 Análisis de la Metodología

El tipo de investigación utilizado para este proyecto de investigación es de tipo correlacional con un enfoque cuantitativo, esto se debe a los resultados que se obtienen de forma numérica, pero también se tiene una base bibliográfica para tener conocimiento en que ámbitos han sido empleados los sensores inteligentes y los métodos expuestos en los capítulos anteriores en la comunidad científica. Los resultados obtenidos en el modelado de UML y la arquitectura que se plantea son comprobados en la simulación de la Red de Petri general y de cada sub-red que se genera por los sub-estados que posee el sistema.

Se consiguió cumplir todas las tareas planteadas para el sistema, es decir que se escogió correctamente el tipo de metodología para el desarrollo de este trabajo de investigación.

4.2.4 Análisis de la Solución del Problema de Investigación

Los sensores inteligentes se basan en el uso de inteligencia artificial, pero la arquitectura propuesta en este trabajo de investigación científica se basa en el uso de agentes inteligentes de tipo deliberativos, lo que hace un método novedoso y muy flexible para la variable que el usuario desee controlar. Una función adicional de esta arquitectura es que permite al agente o microcontrolador que se comunique de forma fiable con el autómatas o PLC. El uso de técnicas nuevas para sensores inteligentes es totalmente fiable, previene al sistema de fallos ya que genera alarmas cada que lo necesite y se auto calibra en caso de existir una perturbación o fallos en el envío o recibo de datos, también almacena los últimos valores que el sensor emitió para continuar con su funcionamiento regular, si es que el sensor principal presenta algún error en la comunicación Agente/Sensor. Envía los datos al PLC de forma fiable a

través de un protocolo de comunicación industrial MODBUS. El modelado del sistema en UML permitió entender el funcionamiento del sistema con más detalle y a su vez comprobarlo en la simulación de las Redes de Petri planteadas en el Capítulo III.

4.2.5 Discusión Final

Los objetivos, las tareas de la metodología, y el planteamiento del problema de investigación se pudieron cumplir a cabalidad con la arquitectura planteada en este proyecto de investigación. Lo que permitió generar el modelado en UML de forma satisfactoria para cada dispositivo y sus actividades. El uso de Redes de Petri para tener los resultados de simulación de la arquitectura planteada, fue satisfactorio, con la que se pudo ver el comportamiento del mismo a través de la red general y de cada sub-red que se generan por los sub-estados que posee el sistema. El objetivo de este trabajo de investigación era proponer una arquitectura novedosa y que esté disponible para cualquier variable que se desee controlar, lo que se logró debido al uso de agentes inteligentes de tipo deliberativos.

Este sistema cuenta con métodos novedosos para la aplicación de sensores inteligentes en el control de procesos industriales.

Los resultados obtenidos fueron satisfactorios ya sea en lo simulado como en la implementación física del sensor. Los rangos de operación se mantuvieron estables conforme al tiempo, incluso teniendo un flujo constante de agua. El tiempo de respuesta fue muy favorable ya que fue mínimo a medida que incrementaba el agua para el caso del nivel.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

En esta sección se presentan las conclusiones y recomendaciones a las que se llegó después de realizar el análisis y estudio de las tareas planteadas para este trabajo de investigación.

5.1 CONCLUSIONES

- La arquitectura basada en agentes inteligentes de tipo deliberativos que se propone en este trabajo de investigación es un método novedoso para la automatización inteligente de sistemas industriales que requieran arquitecturas distribuidas.
- El modelado de UML es de gran utilidad para modelar con más detalle el funcionamiento del sensor y los dispositivos que intervienen. Para ello, se utilizó tres diagramas que componen este método, los que son casos de uso, secuencia y estados, con los que se define los dispositivos, las actividades y en que momento intervienen cada uno en conjunto en el sistema.
- El modelado desarrollado en UML Y RdP, permitió escribir el código de programación de una forma sencilla y ordenada y facilitó comprobar que la arquitectura planteada funciona correctamente en el sistema.
- En una valoración teórico – práctica se comprobó que la arquitectura establecida funciona de forma adecuada, para las dos variables de estudio de este trabajo de investigación, es decir nivel y caudal.
- El protocolo MODBUS, utilizado para establecer una comunicación entre el agente o microcontrolador con el autómeta o PLC, resultó seguro, fiable y eficaz, para enviar o recibir datos de forma bidireccional

5.2 RECOMENDACIONES

- Para trabajos futuros se recomienda analizar los dispositivos que el sistema va a utilizar, para que tenga un rendimiento óptimo y seguro. Es importante que la cantidad de transmisión de bits del microcontrolador vaya acorde a las necesidades del sistema, caso contrario este dispositivo no soportará la carga de ejecución de tareas del programa.
- Para establecer nuevas reglas de lógica difusa en caso de ser necesario, es preferible realizar un análisis de las variables nuevas, la forma en como se van comportar dentro del sistema, y si no influyen de manera negativa en la arquitectura planteada.
- La arquitectura que poseen estos sensores inteligentes, es flexible para cualquier variable, pero una a la vez, si se ponen dos en la misma arquitectura requiere de replantearse desde el inicio para evitar fallos significativos en el sistema.
- El uso del IDE de Arduino junto con sus librerías para lógica difusa son un recurso importante y muy útil en caso de querer utilizar herramientas de software sencillas.

Bibliografía

- Aguilar, J. (2013). *LB 2013 Sistemas MultiAgentes y sus Aplicaciones en Automatización Industrial_1.pdf* (2a ed.). Universidad de los Andes.
- Alarcon, R. (2000). *Diseño orientado a objetos con UML*. Grupo EIDOS Consultaría y Documentación Informática,.
- Ando, S. (1997). Intelligent sensor systems: Integrating advanced automatism and optimality into sensors. *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*, 1, 291–294 vol.1.
<https://doi.org/10.1109/SENSOR.1997.613641>
- Barastegui, J. J. R., & Castaño, Á. R. (2017). Diseño y desarrollo de una red MODBUS RTU basada en Arduino. *2017*, 66.
- Business, V. (2018, junio 29). Cursos de ingeniería industrial. *Vester Training Center*.
<https://vestertraining.com/redes-industriales-modbus/>
- Cárdenas, O. (2009). *Medicion de Nivel*.
http://webdelprofesor.ula.ve/ingenieria/oscaror/CursosDictados/web%20instrumentacion%20industrial/1%20transductores%20para%20procesos%20industriales/libro%20pdf/CAP%203%20Medicion_nivel_2009_n.pdf
- Casans, S., Navarro, A. E., Ramirez, D., Marco, B., & Prieto, J. (2006). Intelligent Sensor Network Applied to ISFETs Sensors: Measurement and Characterisation. *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*, 1553–1556.
<https://doi.org/10.1109/IMTC.2006.328675>
- Cohen, G. (2001). *ANALISIS Y CONTROL DE SISTEMAS DE EVENTOS DISCRETOS: DE REDES DE PETRI TEMPORIZADAS AL ALGEBRA*. Universidad Nacional de Rosario.

Cosco, J. (2013). *SENSORES*. Wordpress.

<https://coscomantauni.files.wordpress.com/2013/08/sensores-de-presion-nivel-flujo-temperatura.pdf>

Gerard Meijer, Michiel Pertijs, & Makinwa, K. (2014). *SMART SENSOR SYSTEMS: EMERGING TECHNOLOGIES AND APPLICATIONS* (first). John Wiley & Sons.

Inzunza, D. E. (2016). *Microcontroladores*. 438.

Jantunen, L., & Huuskonen, P. (2008). (PDF) *Smart sensor architecture for mobile-terminal-centric ambient intelligence*. ResearchGate.

<http://dx.doi.org/10.1016/j.sna.2007.04.014>

Joyanes, L., & Zahonero, I. (2014). *Programación en C, C++, Java y UML* (Segunda). McGraw - Hill Interamericana.

Kimmel, P. (2008). *Manual de UML* (Primera). McGraw - Hill Interamericana.

Lambrou, T. P., Anastasiou, C. C., Panayiotou, C. G., & Polycarpou, M. M. (2014). A Low-Cost Sensor Network for Real-Time Monitoring and Contamination Detection in Drinking Water Distribution Systems. *IEEE Sensors Journal*, 14(8), 2765–2772. <https://doi.org/10.1109/JSEN.2014.2316414>

Loarte, R., Quizhpe, B., & Paz, H. (2015). Desarrollo y simulación de un sistema multiagente para la comunicación de semáforos para encontrar la ruta óptima mediante grafos. *Revista Tecnológica ESPOL*, 28, 43–63.

Lucidchart. (2019). *UML*. Types of UML.

https://www.lucidchart.com/documents#docs?folder_id=home&browser=icon&sort=saved-desc

Luis, I., Melo, F. O., & Colombia, N. D. (2016). *Sistemas Multiagente y Fútbol de robots: Estado del*.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.5889&rep=rep1&type=pdf>

Mekid, S. (2006). Further Structural Intelligence for Sensors Cluster Technology in Manufacturing. *Sensors*, 6(6), 557–577. <https://doi.org/10.3390/s6060557>

Microcontrollers. (2019, agosto 1). Difference between RISC and CISC.

Microcontrollers Lab. <https://microcontrollerslab.com/difference-between-risc-and-cisc/>

Morillo, H. R., Pérez, F. M., & Jorquera, D. M. (2013). Redes Inalámbricas de Sensores Inteligentes. Aplicación a la Monitorización de Variables Fisiológicas. *Universidad de Alicante*, 10.

Naylamp, M. (2019). *Sensor de caudal HK3022 0.5MPa*. Naylamp Mechatronics - Perú. <https://naylampmechatronics.com/sensores-liquido/318-sensor-de-presion-hk3022-05mpa.html>

Pérez, C. (2017). *Comunicaciones industriales y en Tiempo Real*. 23.

Premkumardeepak, S., & Krishnan, M. B. M. (2017). Intelligent sensor based monitoring system for underwater pollution. *2017 International Conference on IoT and Application (ICIOT)*, 1–4. <https://doi.org/10.1109/ICIOTA.2017.8073626>

Rivera, J., Seáñez, C., Hernández, A., & López, A. (2011). Intelligent sensor with data fusion to improve the care and management of water. *2011 IEEE International Instrumentation and Measurement Technology Conference*, 1–5. <https://doi.org/10.1109/IMTC.2011.5944181>

Rodríguez, M. (2016). *Sensores*. Universidad de Valladolid.

Saab, C., Shahrour, I., & Chehade, F. H. (2017). Smart technology for water quality control: Feedback about use of water quality sensors. *2017 Sensors Networks*

Smart and Emerging Technologies (SENSET), 1–4.

<https://doi.org/10.1109/SENSET.2017.8125060>

Schmuller, J. (2000). *Aprendiendo UML en 24 horas* (Pearson Educacih de MCxico, Vol. 1). Pearson Educación.

Schonberg, T., Almtoft, K. P., Hansen, M. K., Christensen, B. H., Svensson, M., Ostbo, N. P., Ronkainen, H., & Nielsen, L. P. (2009). Intelligent machine parts with surface embedded sensors under wear resistant coatings.

TRANSDUCERS 2009 - 2009 International Solid-State Sensors, Actuators and Microsystems Conference, 1857–1860.

<https://doi.org/10.1109/SENSOR.2009.5285708>

Selvakumar, K., Karuppiah, M., SaiRamesh, L., Islam, S. H., Hassan, M. M., Fortino, G., & Choo, K.-K. R. (2019). Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in WSNs. *Information Sciences*, 497, 77–90. Scopus.

<https://doi.org/10.1016/j.ins.2019.05.040>

Shen, G. P., Qin, M., Dong, Z. Q., & Huang, Q. A. (2009). An intelligent wind sensor system with auto-zero function. *TRANSDUCERS 2009 - 2009 International Solid-State Sensors, Actuators and Microsystems Conference*, 256–259.

<https://doi.org/10.1109/SENSOR.2009.5285514>

SIEMENS. (2006). *AS-Interface—Introduccion y Nociones fundamentales*. 04, 57.

Torres, V. (2007). *Apuntes de Automatización Industrial*. 36.

Valdés, F., & Pallás, R. (2007). *Microncontroladores fundamentos y aplicaciones con PIC* (Marcombo). Alfaomega.

Vizcay, M. (2005). *Sensores de Presion*. Universidad de Chile. https://www.u-cursos.cl/ingenieria/2005/2/EL53A/1/material_docente/bajar?id_material=8162

9

Wang, P.-J. (2007). Design and Application of Soft Sensor Object in Water Treatment Plant. *2007 International Conference on Machine Learning and Cybernetics*, 5, 2678–2682. <https://doi.org/10.1109/ICMLC.2007.4370602>

Wei, Y., Hu, X., & An, D. (2018). Design of An intelligent pH sensor based on IEEE1451.2. *IFAC-PapersOnLine*, 51(17), 191–198. <https://doi.org/10.1016/j.ifacol.2018.08.139>

Yamasaki, H. (Ed.). (1996). *Intelligent sensors*. Elsevier.



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Andramuño Cando, Joselyne Paulette** con C.C: # 093102372-5 autor del Trabajo de Titulación: **Prototipos de sensores inteligentes de caudal y nivel para la operación de un subsistema de filtrado en una planta potabilizadora de agua**, previo a la obtención del título de **INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 27 de Febrero del 2020

f. _____

Nombre: Andramuño Cando, Joselyne Paulette

C.C: 093102372-5



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

REPOSITARIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Prototipos de sensores inteligentes de caudal y nivel para la operación de un subsistema de filtrado en una planta potabilizadora de agua		
AUTOR(ES)	ANDRAMUÑO CANDO, JOSELYNE PAULETTE		
REVISOR(ES)/TUTOR(ES)	M. Sc. EDUARDO VICENTE MENDOZA MERCHÁN		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería Electrónica en Control y Automatismo		
TITULO OBTENIDO:	Ingeniero Electrónico en Control y Automatismo		
FECHA DE PUBLICACIÓN:	27 de Febrero del 2020	No. DE PÁGINAS:	94
ÁREAS TEMÁTICAS:	Automatización Industrial , Industria 4.0, Control de Procesos		
PALABRAS CLAVES/ KEYWORDS:	Sensor Inteligente, Arquitectura Inteligente, UML, Redes de Petri, BDI.		

RESUMEN/ABSTRACT (150-250 palabras):

El uso de técnicas de inteligencia artificial en arquitecturas para procesos industriales, son de diferentes tipos, como es el uso de agentes inteligentes para el control de procesos, pero no se ha encontrado mucha base científica en el uso de agentes de tipo deliberativos para el diseño de una arquitectura inteligente que sea adaptable a cualquier variable que el usuario desee controlar. El presente trabajo de investigación propone una arquitectura para el control de un sub-sistema de filtrado para una planta potabilizadora de agua utilizando agentes inteligentes BDI para que sea apto a cualquier variable de control, para modelar este sistema se utiliza UML una herramienta muy utilizada por los informáticos y muy poco utilizado para la industria. Para establecer una comparación entre un estado ideal y la realidad se tiene las Redes de Petri, con las que se puede simular la arquitectura y el modelado planteado previamente. La investigación es de tipo correlacional con un enfoque cuantitativo, los métodos que se utilizan son: análisis y síntesis para valorar todas las fuentes de información que se obtengan, simulación para evaluar el desempeño del sistema y experimental para comprobar si funciona en el campo real.

Este trabajo de investigación tiene como resultados la arquitectura adaptable para cualquier variable, el modelado de UML y la RdP correctamente planteados, que se comprueba en la valoración teórico práctica con las pruebas experimentales que arrojan un tiempo de respuesta menor a los 100ms, y se mantiene estable en cuanto cambia de una medida a otra

ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO
CONTACTO CON AUTOR/ES:	Teléfono: +593995413471	E-mail: joselynepau067@gmail.com
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Mendoza Merchán, Eduardo Vicente	
	Teléfono: +593-9-85086815	
	E-mail: eduardo.mendoza@cu.ucsg.edu.ec	
SECCIÓN PARA USO DE BIBLIOTECA		
Nº. DE REGISTRO (en base a datos):		
Nº. DE CLASIFICACIÓN:		
DIRECCIÓN URL (tesis en la web):		