

Intelligent flow and level sensors for a filtering system in a water treatment plant

Eduardo Mendoza, Ms.C¹, Joselyne Andramuño, Ing², and Luis Córdova, Ms.C³

¹ Universidad Católica de Santiago de Guayaquil, Ecuador, eduardo.mendoza01@cu.ucsg.edu.ec, ³luis.cordova@cu.ucsg.edu.ec

²Universidad Católica de Santiago de Guayaquil, Ecuador, joselyneandramuno@gmail.com

Abstract— Industrial sensors play a fundamental role in distributed control systems, and their increase in number and quantity of parameters involved in industrial processes, makes it necessary for these elements to evolve towards intelligent entities, which in addition to fulfilling more complex tasks, have a certain autonomy, which allows reducing the load on the control equipment, increasing the overall performance of the system. This research work proposes an intelligent sensor architecture, based on the BDI methodology, aimed at controlling the filtering system in a water treatment plant, for level and flow variables. The system design was projected using UML tools, and for the modeling of the smart sensor, Petri nets were used, which allowed the simulation, validation and verification of the system. The Modbus protocol for communications was adopted, and for the implementation the ATmega2560 microcontroller was used, which allowed to test experimentally the behavior of the intelligent sensor; response times and the values of the measured variables. The proposed Architecture allowed to develop an adaptable and flexible intelligent sensor in the industrial environment, with adequate performance and behavior, generating values and response times, in many cases better than other works found in the scientific literature.

Keywords: *Sensor Inteligente, Arquitectura, Lenguaje de Modelado, Redes de Petri, BDI.*

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2020.1.1.424>

ISBN: 978-958-52071-4-1 ISSN: 2414-6390

Sensores Inteligentes de Nivel y Caudal para un sistema de filtrado en una planta potabilizadora de agua

Eduardo Mendoza, Ms.C¹, Joselyne Andramuño, Ing², and Luis Córdova, Ms.C³

¹Universidad Católica de Santiago de Guayaquil, Ecuador, eduardo.mendoza01@cu.ucsg.edu.ec, ³luis.cordova@cu.ucsg.edu.ec

²Universidad Católica de Santiago de Guayaquil, Ecuador, joselyneandramuno@gmail.com

Abstract— *Los sensores industriales desempeñan un papel fundamental en los sistemas de control distribuido, y su incremento en número y la cantidad de parámetros involucrados en los procesos industriales, hace que sea necesario que estos elementos evolucionen hacia entidades inteligentes, que además de cumplir tareas más complejas, tienen cierta autonomía lo que permite reducir la carga en el equipo de control, aumentando el rendimiento general del sistema. Este trabajo de investigación propone una arquitectura de sensor inteligente, basada en la metodología BDI, dirigida a controlar el sistema de filtrado en una planta de tratamiento de agua, para variables de nivel y flujo. El diseño del sistema se proyectó utilizando herramientas UML, y para el modelado del sensor inteligente, se utilizaron redes de Petri, lo que permitió la simulación, validación y verificación del sistema. Se adoptó el protocolo Modbus para comunicaciones, y para la implementación se utilizó el microcontrolador ATmega2560, que permitió probar experimentalmente el comportamiento del sensor inteligente; tiempos de respuesta y los valores de las variables medidas. La arquitectura propuesta permitió desarrollar un sensor inteligente flexible y adaptable en el entorno industrial, con un rendimiento y comportamiento adecuados, generando valores y tiempos de respuesta, en muchos casos mejores que otros trabajos encontrados en la literatura científica.*

Keywords: *Smart Sensor, Architecture, Modeling Language, Petri Net, BDI.*

I. INTRODUCCIÓN

La evolución tecnológica a través de los años ha dado paso a la Industria 4.0, que se basa en el uso de métodos nuevos para el control de procesos industriales a través de técnicas de inteligencia artificial. Los sensores, al ser una parte fundamental de un sistema industrial, requieren de un comportamiento inteligentes para acoplarse a los actuales sistemas de control distribuido en la automatización moderna [1][2][3][4]. Una metodología novedosa para el diseño de sensores inteligentes, es a través del uso de agentes inteligentes, los cuales son entidades autónomas que pueden tomar decisiones, sociabilizar y tienen capacidad de reacción y proactividad. Estas características son muy útiles en los procesos industriales para generar un control eficiente y fiable, que sea adaptable al control requerido en los sistemas industriales [3][5][6].

La arquitectura propuesta, está basada en agentes inteligentes de tipo deliberativos BDI, cuyo nombre del modelo es debido a sus componentes: creencias, deseos e intenciones. Las creencias es lo que el agente sabe de su entorno, es un componente informativo del agente. Los deseos están relacionados con lo que el que agente quiere conseguir: sus objetivos, convirtiéndose en un componente motivacional.

Finalmente, las intenciones que constituyen un compromiso del agente para conseguir los objetivos elegidos, constituyéndose en un componente deliberativo [7][8].

De la definición de los requisitos funcionales del sensor, está la función auto calibración. Esta permite al sensor actualizarse cuando sea requerido, con lo datos almacenados en una base de creencias o de información de parametrización. La arquitectura propuesta, permite el sentido de una sola variable por nodo, si se desea cambiar de variable, el sistema se interrumpe para darle paso al ingreso de los datos y parámetros de la nueva variable, actualizar la información, continuar cumpliendo sus tareas y comunicarse con el autómata programable asociado [9][10][11][12].

El modelado del sistema general, se caracteriza por el uso de un método novedoso para sistemas de control industrial, pero comúnmente utilizado en ingeniería de software. El lenguaje de Modelado Unificado (UML, del inglés Unified Modeling Language) permite proyectar el diseño de este sistema, definir los requisitos funcionales y no funcionales con mayor detalle, a través de la identificación del número actores o dispositivos que van a intervenir en el sistema, apoyado en los diagramas de casos de uso del sistema [13][14][15]. Los diagramas de secuencia y de estados de UML, permite determinar el orden de ejecución de tareas que cada actor debe realizar, describiendo el comportamiento dinámico del sistema de información. Otra herramienta utilizada en el modelado del sistema, son las Redes de Petri, que constituye un método gráfico y matemático para simular, verificar y validar el comportamiento de la arquitectura en el sensor inteligente [9][16]. Cuando el modelo crece demasiado, las redes de Petri permiten agrupar secciones en subredes, que se generan a partir del sistema general, donde cada subred pasa a ser una entidad con entradas y salidas. [17] Los resultados del modelado elaborado, permitieron escribir de forma rápida y sistemática, el código para la programación en C/C++, utilizado para el desarrollo del firmware de microcontrolador de 8 bits ATmega2560, utilizado para la implementación del sistema y la comunicación bidireccional con el PLC [18][19][20][21]. Las pruebas realizadas del funcionamiento de este programa, permitieron comprobar la arquitectura propuesta de forma experimental.

Durante la investigación, el análisis de cada uno de los componentes de estudio del sistema, determinó que la arquitectura propuesta, basada en agentes inteligentes BDI, es viable para ser implementada y en el control de procesos industriales. De esta forma, el sensor inteligente, cumple las funciones definidas al inicio de la investigación, reduciéndole carga al autómata programable, y ejecutando más tareas de forma autónoma.

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2020.1.1.424>

ISBN: 978-958-52071-4-1 ISSN: 2414-6390

II. MATERIALES Y MÉTODOS

En esta sección se describe los elementos que componen el sistema, y que fueron utilizados para realizar las pruebas experimentales del sensor inteligente. El sistema está compuesto por los siguientes elementos, que se muestran en la Tabla 1.

TABLA I
ELEMENTOS DEL SISTEMA GENERAL

Elemento	Tipo
Microcontrolador	AT Mega 2560
Tipo de Comunicación Agente/PLC	MODBUS RTU
Sensor de Nivel	Ultrasónico HC-SR04
Sensor de Caudal	YF-S201
Pantalla LCD gráfica	SSD1306
Tensión de trabajo	12 voltios

A. Trama MODBUS RTU

La trama del protocolo MODBUS seleccionada para este sistema, es de tipo RTU que se ajusta a la emisión y recepción de datos desde el microcontrolador al PLC y viceversa.

La trama de MODBUS está conformada por la Unidad de Datos de Aplicación (ADU) y la 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.

Esta 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 recepción de datos de forma fiable. En la tabla 2 se muestra la trama con chequeo de paridad de datos, donde el penúltimo bit es par.

TABLA II
TRAMA CON CHEQUEO DE PARIDAD

Inicio	1	2	3	4	5	6	7	8	PAR	Paro
--------	---	---	---	---	---	---	---	---	-----	------

En la tabla 3 se muestra la trama sin chequeo de paridad de datos, donde el penúltimo bit es paro.

TABLA III
TRAMA SIN CHEQUEO DE PARIDAD

Inicio	1	2	3	4	5	6	7	8	PARO	Paro
--------	---	---	---	---	---	---	---	---	------	------

En la tabla 4 se muestra el funcionamiento y longitud de bits de cada componente de la trama RTU de MODBUS.

TABLA IV
DESCRIPCIÓN TRAMA MODBUS

Nombre	Longitud de 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

III. MODELADO DEL SENSOR INTELIGENTE

En esta sección se describe, de forma general el funcionamiento del sensor a través de un diagrama de flujo, el diseño de la arquitectura basada en agentes inteligentes de tipo deliberativos BDI, el modelado en UML y el diseño de la red de Petri general para las variables de control industrial.

A. Descripción del Sistema

El programa del sensor inicia cargando la base de datos de las creencias, donde se encuentran los valores de parametrización de las variables del sistema. Luego, continúa su flujo verificando si existe un usuario. De haberlo, configura una nueva variable que no se encuentre registrada en la base de datos de creencias. De no existir presencia de usuario, pasa a verificar la señal con el sensor y el sistema verifica la comunicación con el PLC. Si existe dicha comunicación, recibe la señal, la procesa y la envía al PLC, el cual responde con un orden. La orden es recibida por el sensor a través del intérprete de comandos, donde se ejecutan las funciones solicitadas y repite el bucle anterior. En la figura 1 se muestra el diagrama de flujo general del sistema.

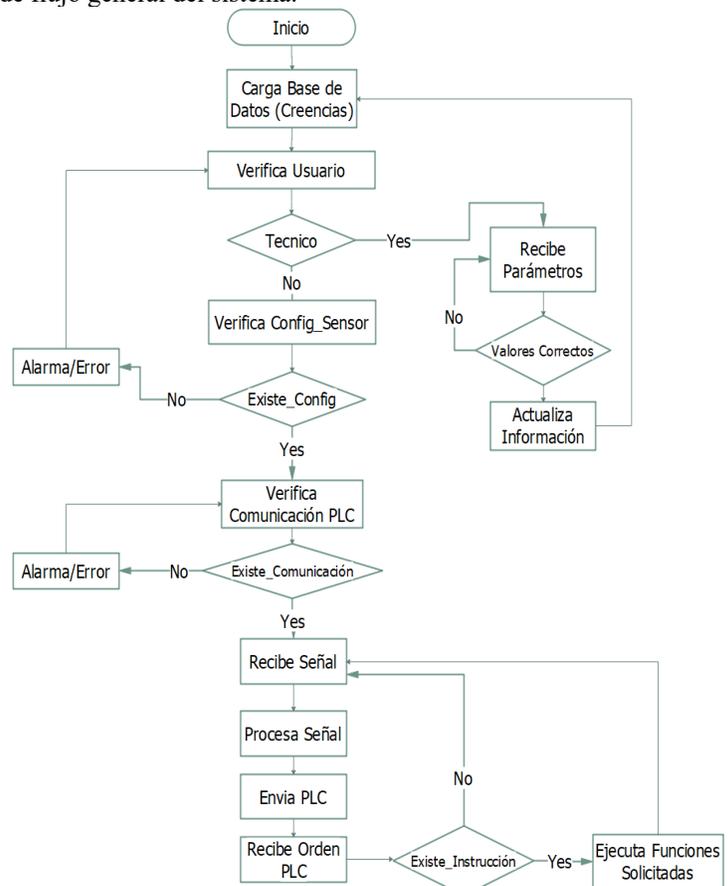


Fig. 1 Diagrama de Flujo General del Sistema

B. Diseño de la arquitectura nodo sensor/actuador

La arquitectura propuesta, está compuesta por cuatro etapas. En la primera etapa se encuentran los bloques donde se alojan las creencias, que corresponde a una pequeña base de datos que contiene algoritmos de funcionamiento y rangos de parametrización, señales de tensión, frecuencias de operación, entre otros parámetros que se desee agregar. En la figura 2 se aprecia la primera etapa de la arquitectura nodo sensor/actuador.

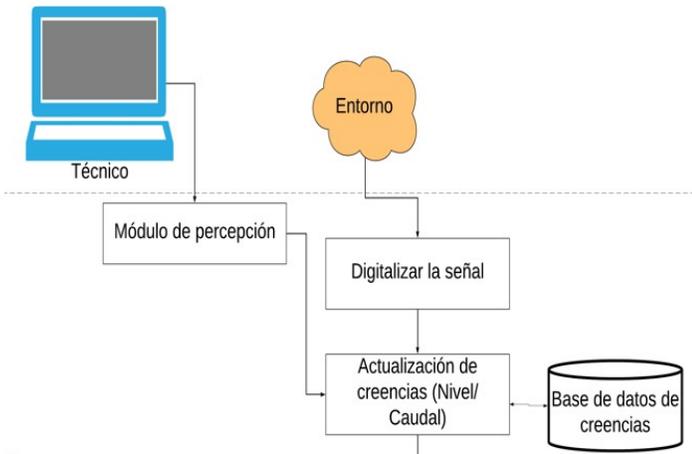


Fig. 2 Primera Etapa de la Arquitectura Inteligente

La segunda etapa de la arquitectura BDI, está compuesta por los deseos, los cuales son representados por los objetivos que debe cumplir el sensor inteligente. En esta etapa se encuentra la función de selección de comportamiento que se utiliza para que el sensor determine si la señal de entrada es de caudal o nivel, o si está funcionando como sensor principal o de respaldo. El despachador de eventos también se encuentra en esta etapa, la cual da paso a la tercera etapa de la arquitectura que son las intenciones. En la figura 3 se muestra la segunda etapa de esta arquitectura.



Fig. 3 Segunda Etapa de la Arquitectura Inteligente

La tercera etapa de la arquitectura BDI, está compuesta por las intenciones, las cuales representa las funciones que debe ejecutar el agente con los datos obtenidos por el sensor para la operación óptima del sistema general. Estas funciones son: auto

calibración, análisis y promedio de valores, extrapolación de datos, y generación de alarmas. En la figura 4 se muestra la tercera etapa de la arquitectura del sensor inteligente.



Fig. 4 Tercera Etapa de la Arquitectura Inteligente

La etapa final de la arquitectura, donde se encuentra la comunicación agente/PLC y viceversa. En esta parte se aloja el protocolo MODBUS de trama RTU, que permitirá enviar y recibir datos e instrucciones con el dispositivo de control, de forma fiable y segura. En la figura se muestra la última etapa de la arquitectura inteligente.

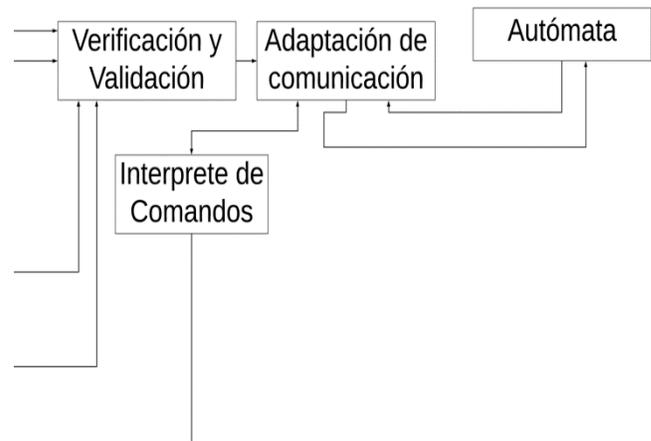


Fig. 5 Cuarta Etapa de la Arquitectura Inteligente

B. Modelado en UML del Sistema General

En esta sección, se presenta el modelado con UML, del sistema general, para lo cual se recurre a tres diagramas UML diferentes: casos de uso, secuencia y estados, previo al diseño de la Red de Petri.

El diagrama de Casos de Uso, se lo utiliza para identificar los actores o dispositivos que van a interactuar en el sistema, con sus respectivas funciones o casos de uso. En la figura 6 se muestra el diagrama general del sistema.

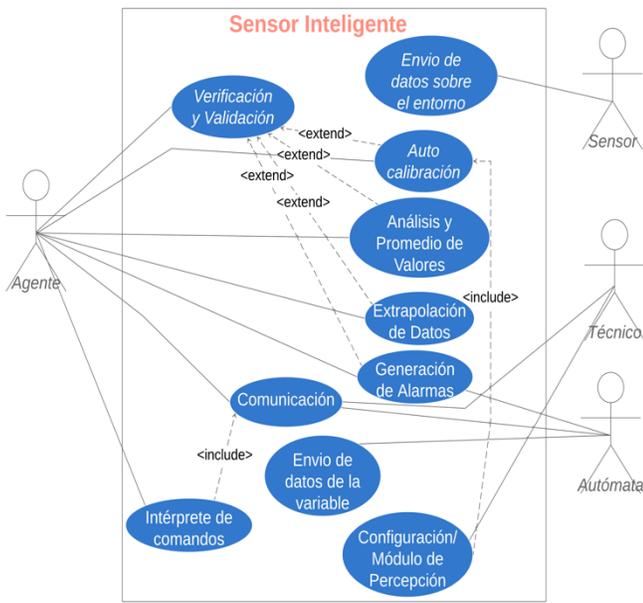


Fig. 6 Diagrama de Casos de Uso General del Sistema

El diagrama de Secuencia, es utilizado para representar y comprender la secuencia de interacción de estas actividades con los dispositivos o actores que dispone el sistema. En la figura 7 se muestra el diagrama general del sistema.

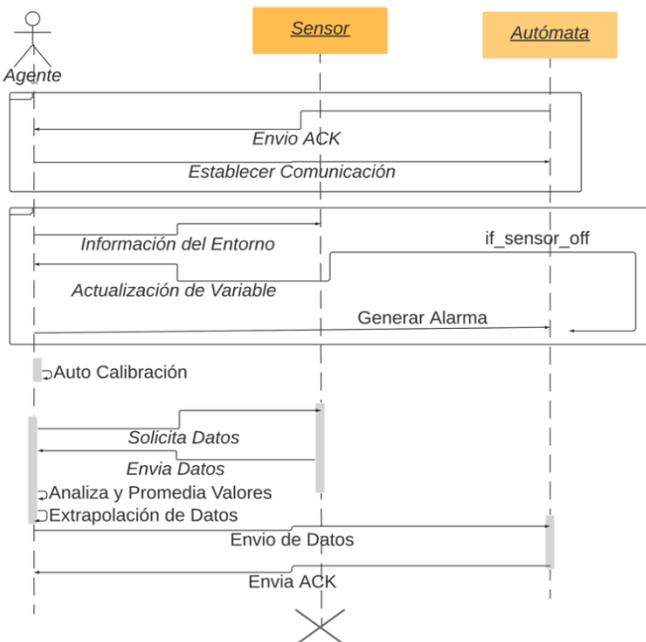


Fig. 7 Diagrama de Secuencia General del Sistema

El diagrama de Estados, es utilizado para definir y establecer como el conjunto de estado del sistema, pasa en determinado orden en respuesta a los eventos generados. Dentro de estos estados, pueden representarse sub-procesos que se generan a partir de las decisiones o interrupciones que el sistema maneja. En la figura 8 se muestra el diagrama de estados general del Sistema.

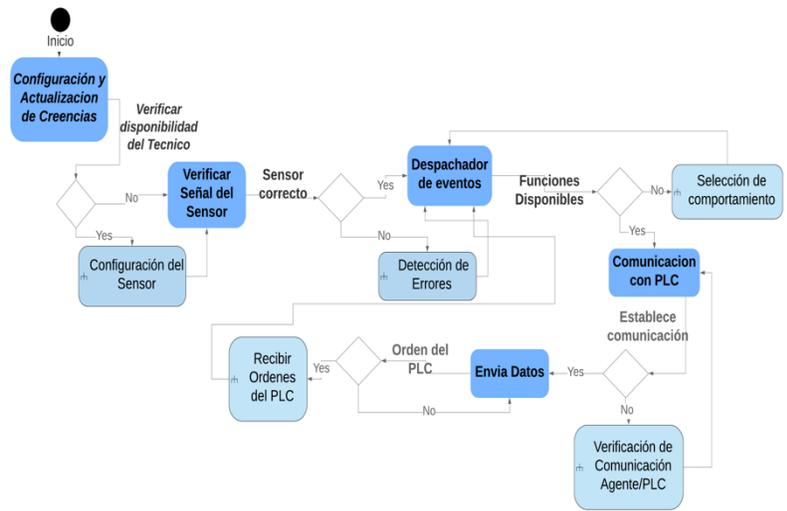


Fig. 8 Diagrama de Estados Estados del Sistema

C. Diseño de la Red de Petri General.

La red de Petri son grafos utilizados para representar sistemas concurrentes de eventos discretos, y que permite simular y obtener las ecuaciones matemáticas a partir del diseño de la red. Se utilizó este método para comprobar mediante simulación, la arquitectura planteada y el comportamiento del sensor inteligente. En la figura 9 se presenta la red de Petri general del sistema.

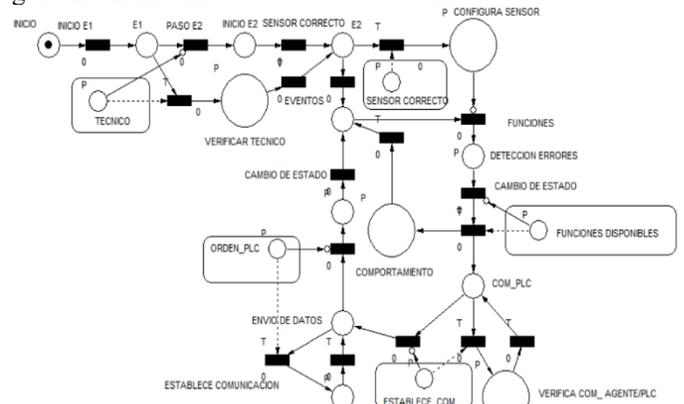


Fig. 9 Red de Petri General del Sistema

Esta Red contiene 4 sub-estados, que se van a activar por componentes de selección, que se deducen a partir de lo investigado. En la figura 10 se presenta este componente de selección.

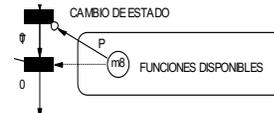


Fig. 10 Componente de Selección del Sistema

El componente de selección está compuesto por: un arco inhibitor, uno de prueba y un lugar que es donde se va a alojar el token que dará apertura a los sub-estados o interrupciones del sistema.

A partir de esta red se obtiene la ecuación 1, que corresponde al sistema general.

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

Esta ecuación demuestra que el sistema se mantiene estable en la interrupción que se encuentre, así 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 muestra, que controla una variable a la vez, y que solo existe un estado de operación para toda la red, comprobando la correcta distribución de la arquitectura planteada y el funcionamiento general del sistema simulado.

D. Sub-estados de la Red de Petri

A partir de la RdP general, se tiene cuatro sub-estados que operan en diferentes instancias en el sistema, pero todos cumplen el flujo general. Estos sub-estados son: configuración del sensor, configuración agente/PLC, detección de errores y selección de comportamiento. En la figura 11 se muestra el sub-estado de configuración del sensor.

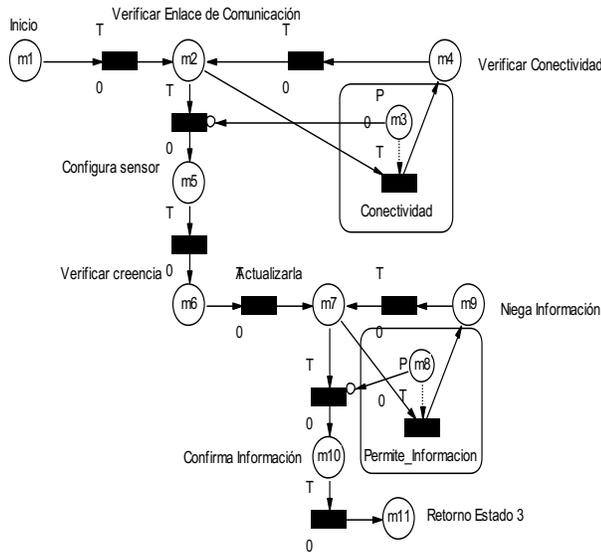


Fig. 11 Red de Petri del sub-estado: Configuración del Sensor

A partir de esta red se tiene la ecuación 2, que corresponde al comportamiento del sub-estado 1 o configuración del sensor.

$$M(P0) + M(P1) + M(P10) + M(P12) + M(P4) + M(P5) + M(P6) + M(P7) + M(P9) = 1. \quad (2)$$

Esta ecuación muestra todos los lugares posibles por donde pasa la marca o token y se mantiene en 1 a pesar de las dos decisiones internas que debe tomar, es decir que se mantiene estable. Este sub-estado se lo utiliza para configurar un sensor nuevo o, exista el requerimiento de cambiar de tipo de sensor, o parámetros que no estén declarados en la base de datos de las creencias. Otra función de este sub-estado es verifica si existe conectividad con el dispositivo de control, en caso de no responder, emite una alarma de error, caso contrario actualiza la información.

III. PRUEBAS EXPERIMENTALES

Las pruebas experimentales se realizaron en un prototipo basado en el microcontrolador Atmega2560, en el cual se comprueba de forma práctica, el correcto funcionamiento de la arquitectura para el sensor inteligente.

Para la prueba de nivel se ha provisto de un reservorio de 20cm de alto, un sensor ultrasónico HC-SR04, la tarjeta Arduino Mega 2560, una pantalla Oled SSD1306 donde se muestran los valores del sensor de nivel en tiempo real. Para esta prueba se obtiene el tiempo respuesta del sensor, desde que la señal es enviada al líquido, hasta que se muestra en la pantalla. También se analiza la estabilidad del sistema y su comportamiento durante la ejecución de la tarea. Con los datos obtenidos, contrastado con un patrón, se determina el porcentaje de error entre el valor real y el valor obtenido por el sensor. En la tabla 5 se muestra el tiempo de estabilización del sensor inteligente para la señal de nivel.

TABLA V
TIEMPO DE ESTABILIZACIÓN DEL NIVEL

Nivel (cm)	Tiempo de Estabilización (ms)
18	16
12	20
6	34
3	40
1	36

Se considera tiempos discretos durante la ejecución de la prueba, los cuales se encuentran a la izquierda, en la tabla V. Inicialmente los tiempos de respuestas son más pequeños que al final. Calculando el promedio de los tiempos de estabilización obtenidos, se tiene que el sensor se estabiliza en 29.2ms.

En la tabla 6 se muestra el porcentaje de error del sensor inteligente de nivel. En el lado derecho se muestran los valores obtenidos del sensor inteligente

TABLA VI
VALORES DE NIVEL DEL SENSOR INTELIGENTE

Nivel patrón (cm)	Nivel del sensor (cm)
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

A partir de la tabla VI, se obtienen los errores de los valores obtenidos del nivel, posteriormente se determina el porcentaje de error promedio entre el valor patrón y el valor entregado por el sensor inteligente, el cual da como resultado 0.15%

Para la prueba de caudal se tiene un reservorio de 20 cm de alto, un sensor de caudal YF- S201, la tarjeta Arduino Mega 2560, una pantalla Oled SSD1306, donde se aprecia los valores de caudal en tiempo real.

Para esta prueba se obtiene el tiempo de respuesta que tarda el sensor en procesar una medida y otra, mientras está encendida la bomba y emite un flujo de agua constante. Comparando de esta forma, el sistema se mantiene estable, y se comprueba el porcentaje de error entre el valor patrón y el obtenido por el sensor. En la tabla 7 se muestra el tiempo de estabilización del sensor de nivel, considerando la salida del sensor (Hz).

TABLA VII
TIEMPO DE ESTABILIZACIÓN DEL CAUDAL

Frecuencia (Hz)	Caudal (lt/min)	Tiempo (ms)
19	3.16	8
13	2.16	17
12	2	24
11	1.83	33
10	1.67	40
9	1.5	53
8	1.33	54
7	1.16	63

Realizando una media de todos los tiempos de estabilización obtenidos, se tiene que el sensor se estabiliza en 37ms.

En la tabla 8 se muestra los valores obtenidos de las medidas de patrón de caudal y del sensor inteligente. A partir de estos valores se obtiene el error en cada uno de ellos.

TABLA VIII
PORCENTAJE DE ERROR DEL CAUDAL

Nivel Patrón (lt)	Nivel del sensor inteligente (lt/min)
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
3,16	3,5
2,16	3
2,00	2,5
1,83	2
1,67	1,5

Al calcular el promedio de los errores de los valores de la tabla 8, se determina el porcentaje de error entre el valor patrón y el valor entregado por el sensor inteligente, el cual da como resultado 2.56%.

IV. RESULTADOS Y DISCUSIÓN

Se propone una metodología novedosa para el diseño de sensores inteligentes basado en una arquitectura deliberativa de agentes inteligentes para el sensado en un proceso industrial.

La gráfica obtenida en la prueba experimental de nivel, realizada con el software Matlab, que se muestra en la figura 12, permite determinar que el tiempo de estabilización de la respuesta del sensor es aceptable, debido a que se encuentra por debajo de los 10ms, permaneciendo estable ante alguna perturbación. El tiempo promedio entre cada respuesta del sensor inteligente es de 15ms, considerando un flujo constante de agua.

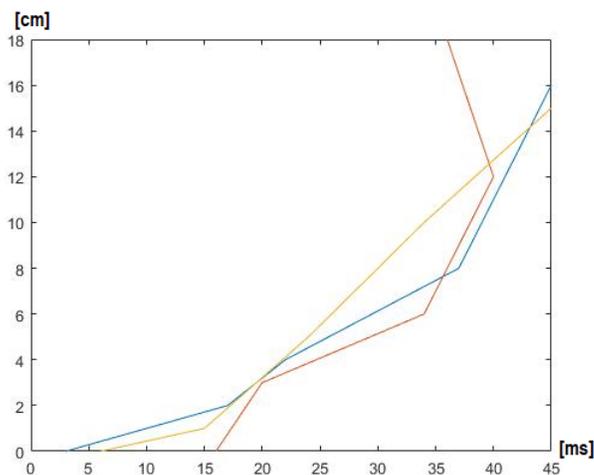


Fig. 12 Gráfica del Sensor Inteligente de Nivel

La gráfica obtenida en la prueba experimental de caudal, realizada en Matlab, y que se muestra en la figura 13, permite determinar que el tiempo de estabilización es aceptable debido a que se encuentra por debajo de los 50ms, permaneciendo el sistema estable ante cualquier perturbación. El caudal se lo toma con referencia a la frecuencia y presión que emite el sensor, 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.

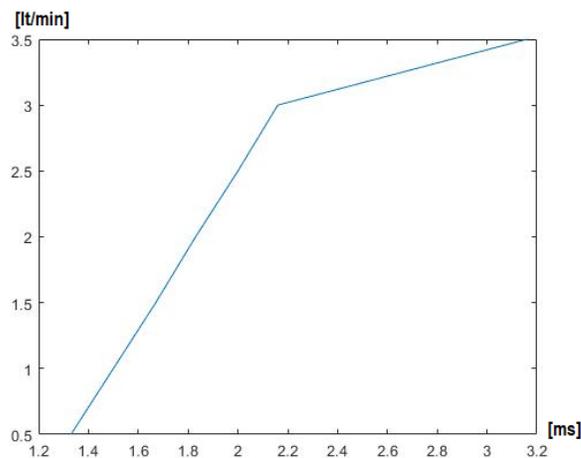


Fig. 13 Gráfica del sensor inteligente de caudal

El sistema propuesto, basada en una arquitectura deliberativa se mantiene estable para las dos variables de control, 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.

El porcentaje de error obtenido en nivel, es bajo en comparación con otros trabajos científico, con un porcentaje de precisión del sensor ultrasónico utilizado para el nivel HC-SR04 es de $\pm 0.3\text{cm}$. Los porcentajes de error para un sensor ultrasónico sin inteligencia son: 12%, 0.74% y 3.74% a diferencia si se le aplica la arquitectura planteada que se obtiene un error de 0.15% [22] [23][24].

El porcentaje de error obtenido en caudal, es bajo en comparación con otros trabajos científicos, teniendo en cuenta que el porcentaje de precisión del sensor de caudal Yf-s201 es de $\pm 10\%$. Los porcentajes de error para un sensor ultrasónico sin inteligencia son: 3%, 5% y 8% a diferencia si se le aplica la arquitectura planteada con la que se obtiene un error de 2.56% [25][26][27].

El tiempo de estabilización o respuesta en nivel es bajo con respecto a los que presentan en otros trabajos científicos, debido a que los valores son: 14ms, 58ms y 104ms a diferencia del tiempo de respuesta aplicando la arquitectura inteligente que se plantea, con la que se obtiene un valor de 29.2ms [27] [28][29].

El tiempo de estabilización o respuesta en caudal es bueno con respecto a los que presentan en la comunidad científica, debido a que los valores son: 12s, 5.04ms y 1s a diferencia del tiempo de respuesta aplicando la arquitectura inteligente que se plantea, con la que se obtiene un valor de 37ms[30][31][32].

IV. CONCLUSIONES

La arquitectura basada en agentes inteligentes a través de una metodología BDI deliberativa para el diseño de un sensor inteligente industrial, es una propuesta novedosa que aporta al diseño de sistemas de automatización inteligente distribuido, permitiendo que los sensores industriales, a más de realizar tareas más complejas, dispongan de un grado de autonomía, que les permita tomar ciertas decisiones de forma eficiente y segura, reduciendo la carga en los equipos de control y el tiempo de respuesta del sistema.

La metodología de modelado basado en UML y RdP, favorece considerablemente la proyección y diseño de sensores inteligentes industriales, permitiendo definir con mucho detalle el funcionamiento del sensor y los dispositivos que intervienen en el sistema. A través de la simulación y análisis de las ecuaciones de estado de las redes de Petri, permite una exploración del comportamiento dinámico del sensor, así como la verificación y validación de las propiedades del sistema, lo que favorece la estabilidad y robustez del dispositivo. Por otro lado, el modelado basado en UML y RdP, facilita el desarrollo del código de programación, permitiendo una escritura sistemática y sencilla.

Para una valoración teórico – práctica, se determinó que la arquitectura propuesta, permiten diseñar sistemas estables, apoyados en agentes inteligentes, dotando de la flexibilidad requerida para adaptarse a diferentes tipos de redes y variable que el usuario final desee controlar.

Finalmente, en trabajos futuros, es importante acercarse más a los sistemas multiagentes y las técnicas de inteligencia artificial disponibles, de tal forma que los sensores inteligentes, puedan comunicarse y socializar con otros dispositivos similares, en aras de generar un sistema industrial más robusto, eficiente, estable y confiable.

REFERENCIAS

[1] H. Yamasaki, Ed., *Intelligent sensors*. Amsterdam; New York: Elsevier, 1996.

[2] S. Premkumardeepak y M. B. M. Krishnan, “Intelligent sensor based monitoring system for underwater pollution”, en *2017 International Conference on IoT and Application (ICIOT)*, Nagapattinam, India, 2017, pp. 1–4, doi: 10.1109/ICIOTA.2017.8073626.

[3] S. Casans, A. E. Navarro, D. Ramirez, B. Marco, y J. Prieto, “Intelligent Sensor Network Applied to ISFETs Sensors: Measurement and Characterisation”, en *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*, 2006, pp. 1553–1556, doi: 10.1109/IMTC.2006.328675.

[4] S. Ando, “Intelligent sensor systems: integrating advanced automatism and optimality into sensors”, en *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*, 1997, vol. 1, pp. 291–294 vol.1, doi: 10.1109/SENSOR.1997.613641.

[5] K. Selvakumar *et al.*, “Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in WSNs”, *Inf. Sci.*, vol. 497, pp. 77–90, 2019, doi: 10.1016/j.ins.2019.05.040.

[6] J. Rivera, C. Seáñez, A. Hernández, y A. López, “Intelligent sensor with data fusion to improve the care and management of water”, en *2011 IEEE International Instrumentation and Measurement Technology Conference*, 2011, pp. 1–5, doi: 10.1109/IMTC.2011.5944181.

[7] J. Aguilar, *LB 2013 Sistemas MultiAgentes y sus Aplicaciones en Automatización Industrial_1.pdf*, 2a ed. Venezuela: Universidad de los Andes, 2013.

[8] T. Schonberg *et al.*, “Intelligent machine parts with surface embedded sensors under wear resistant coatings”, en *TRANSDUCERS 2009 - 2009 International Solid-State Sensors, Actuators and Microsystems Conference*, 2009, pp. 1857–1860, doi: 10.1109/SENSOR.2009.5285708.

[9] G. Cohen, *ANÁLISIS Y CONTROL DE SISTEMAS DE EVENTOS DISCRETOS: DE REDES DE PETRI TEMPORIZADAS AL ALGEBRA*. República Argentina: Universidad Nacional de Rosario, 2001.

[10] V. Torres, “Apuntes de Automatización Industrial”, p. 36, 2007.

[11] P.-J. Wang, “Design and Application of Soft Sensor Object in Water Treatment Plant”, en *2007 International Conference on Machine Learning and Cybernetics*, 2007, vol. 5, pp. 2678–2682, doi: 10.1109/ICMLC.2007.4370602.

[12] Y. Wei, X. Hu, y D. An, “Design of An intelligent pH sensor based on IEEE1451.2”, *IFAC-Pap.*, vol. 51, núm. 17, pp. 191–198, ene. 2018, doi: 10.1016/j.ifacol.2018.08.139.

[13] J. Schmuller, *Aprendiendo UML en 24 horas*, Pearson Educacih de MCxico., vol. 1, 1 vols. 53370 Naucalpan de Jufirez, Edo. de MCxico: Pearson Educación, 2000.

[14] P. Kimmel, *Manual de UML*, Primera. Mexico: McGraw - Hill Interamericana, 2008.

[15] R. Alarcon, *Diseño orientado a objetos con UML*. España: Grupo EIDOS Consultoría y Documentación Informática, 2000.

[16] H. R. Morillo, F. M. Pérez, y D. M. Jorquera, “Redes Inalámbricas de Sensores Inteligentes. Aplicación a la Monitorización de Variables Fisiológicas”, *Univ. Alicante*, p. 10, 2013.

[17] Gerard Meijer, Michiel Pertijs, y K. Makinwa, *SMART SENSOR SYSTEMS: EMERGING TECHNOLOGIES AND APPLICATIONS*, First. United Kingdom: John Wiley & Sons, 2014.

[18] L. Joyanes y I. Zahonero, *Programacion en C, C++, Java y UML*, Segunda. México, D. F.: McGraw - Hill Interamericana, 2014.

[19] J. J. R. Barastegui y Á. R. Castaño, “Diseño y desarrollo de una red MODBUS RTU basada en Arduino”, 2017, p. 66, 2017.

[20] F. Valdés y R. Pallás, *Microncontroladores fundamentos y aplicaciones con PIC*, Marcombo. Cuba: Alfaomega, 2007.

[21] D. E. Inzunza, “Microcontroladores”, p. 438, 2016.

[22] C. J. Jeswin, B. Marimuthu, y K. Chithra, “Ultrasonic water level indicator and controller using AVR microcontroller”, en *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, Chennai, 2017, pp. 1–6, doi: 10.1109/ICICES.2017.8070773.

[23] M. Saraswati, E. Kuantama, y P. Mardjoko, “Design and Construction of Water Level Measurement System Accessible through SMS”, en *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling*

- and Simulation*, Malta, Malta, 2012, pp. 48–53, doi: 10.1109/EMS.2012.60.
- [24] C. D. S. Mayorga, L. Morales, y L. Corrales, “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE VIDEO Y MEDICIÓN DE DISTANCIA DURANTE EL RETROCESO DE UN VEHÍCULO EN UN DISPOSITIVO MÓVIL”, p. 161.
- [25] A. C. Tasong y R. P. Abao, “Design and Development of an IoT Application with Visual Analytics for Water Consumption Monitoring”, *Procedia Comput. Sci.*, vol. 157, pp. 205–213, 2019, doi: 10.1016/j.procs.2019.08.159.
- [26] E. Nasimba y S. Ríos, “Diseño, Reconstrucción e Implementación del modular didáctico para el control de nivel, caudal y temperatura basado en autómatas programables usando software libre”. Espe, 11-2016.
- [27] C. F. C. Cevallos y S. A. R. Porras, “INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES”, p. 153, 2018.
- [28] N. Vega, P. Parra, L. Córdova, J. Andramuño, y V. Navarrete, “Adaptive Control Implemented with Embedded Systems”, en *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, 2018, pp. 1–6, doi: 10.1109/ICA-ACCA.2018.8609713.
- [29] N. Vega, P. Parra, L. Cordova, J. Andramuno, y J. Alvarez, “Cascade Control Algorithm developed with Embedded Systems”, en *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, Concepcion, 2018, pp. 1–6, doi: 10.1109/ICA-ACCA.2018.8609825.
- [30] I. J. C. Sosa y A. S. M. Paguay, “INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y REDES”, p. 104, 2017.
- [31] C. López-Montes, “Telemonitorización y control de un sistema de riego por goteo mediante bombeo solar”, sep. 2017.
- [32] J. Andramuño, N. Vega, y P. Parra, “Industry 4.0 Embedded Systems Network”, en *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 2019, pp. 1–6, doi: 10.1109/CHILECON47746.2019.8987620.