

DISEÑO DE UN SISTEMA DE COMUNICACIÓN IoT PARA EL MONITOREO Y CONTROL DE VARIABLES DE PROCESOS INDUSTRIALES

DESIGN OF IoT PLATFORM FOR MONITORING AND CONTROL OF INDUSTRIAL PROCESS

Ing. Azarquiel Díaz*, M.Sc. Hernando González*

* Universidad Autónoma de Bucaramanga, Grupo de Investigación de Control y
Mecatrónica - GICYM.

Av. 42 # 48 - 11, Bucaramanga, Santander, Colombia.

PBX (57) (7) 643 6111/643 6261

E-mail: { adiaz106, hgonzalez7 }@unab.edu.co.

Resumen: El artículo presenta el diseño e implementación de una plataforma IoT con fines académicos. Se modelan procesos industriales en el software Factory I/O, se diseña una interfaz para el monitoreo y control de las variables relevantes del proceso simulado. El valor de referencia de los lazos de control se puede definir a partir de un aplicativo web o a través de un aplicativo móvil, además se puede descargar históricos de las diferentes variables del sistema simulado.

Palabras clave: Industria 4.0, Software Factory I/O, Plataforma IoT, Heroku, Mongo DB.

Abstract: The paper presents the design and implementation of an IoT platform for academic purposes. Industrial processes are modeled in software Factory I/O, an interface is designed for the monitoring and control of the relevant variables of the simulated process. The reference value of the control loops can be defined from a web application or through a mobile application, also it allows download historical data of the different variables of the process.

Keywords: Industry 4.0, Software Factory I/O, IoT Platform, Heroku, Mongo DB

1. INTRODUCCIÓN

El sistema de control distribuido (DCS) es un sistema moderno que puede controlar diferentes procesos en paralelo, con un controlador central que funciona como el cerebro y unidades terminales remotas (RTU) que permiten tomar la información, procesarla y actuar sin importar si los sensores y actuadores no se encuentran cerca del controlador (Luz y Forero, 2015). Este tipo de sistema de control es empleado en plantas de generación eléctrica, sistemas de tratamiento de aguas servidas, procesos mineros, sistemas de

control ambiental y en industrias petroquímicas, mineras (Quispe 2016).

El reciente desarrollo tecnológico de varias aplicaciones para edificios, hogares inteligentes, industria, transporte, atención sanitaria y vigilancia ha sido proporcionado por el paradigma generalizado de Internet de las cosas. Este hecho ha permitido la aparición de un número creciente de dispositivos conectados en una infraestructura de red con protocolos de comunicación y software interoperables (Neve *et al.*, 2018). Internet de las cosas, incluye muchas oportunidades en toda la industria de procesos, abarcando sectores como la

producción de petróleo y gas, farmacéutica, alimentos y bebidas, energía, etc. Para muchas de estas industrias, un cambio en la calidad o productividad del 1% al 2% puede producir beneficios significativos a través del ahorro de energía, la reducción de desperdicios, la reducción del tiempo de fabricación, la mejora de la seguridad y la reducción del impacto en el medio ambiente. Algunos ejemplos de plataformas IoT en el sector industrial son: Emerson Automation Solutions ha desarrollado un plataforma IoT, que incluye dispositivos de campo con protocolo de comunicación Hart y WirelessHart, estos equipos transmiten la lectura del sensor y el estado del mismo. Los datos se almacenan en la nube mediante el servidor Microsoft Azure, utilizando el servicio *IoT edge gateways*, el cual habilita puertas de enlace para los dispositivos, permitiendo el almacenamiento y monitoreo de la información (Wang *et al.*, 2019). Mediante el uso del bus CAN (Controller Area Network) implementado en un sistema conformado por los microprocesadores NXP1pc1768, ATmega328 y MCP2515, se plantea una conexión entre una red de sensores y un aplicativo web. El sistema propuesto consta de tres módulos esclavos y un maestro NXP1pc1768. Cada módulo esclavo lo conforma un chip integrado ATmega328 que está conectado a dos sensores que miden diferentes parámetros. El microcontrolador entrega la información de los sensores al controlador CAN MCP2515. Este controlador CAN actúa como una interfaz entre el ATmega328 y el transceptor CAN MCP2551. El Master NXP1pc1768 tiene puertos CAN y puerto Ethernet incorporados. CAN H y CAN L forman el bus de red del área del controlador y permiten la comunicación entre el maestro y los tres módulos esclavos. El transceptor de los esclavos y el transceptor del maestro se comunican eficazmente mediante la conexión de bus. El maestro monitorea y muestra los valores analógicos de los sensores en una página web y también permite al usuario desde la ubicación remota controlar los sensores dependiendo de los valores de los sensores mostrados (Niveditha *et al.*, 2018).

2. PROCESO INDUSTRIAL

Una de las ramas de la industria 4.0, son los ambientes simulados. Con el fin de no invertir en una planta física que puede estar sub o sobredimensionada, empresas optan por realizar simulaciones de sus procesos industriales, de esta manera pueden corregir u optimizar el sistema. (Muñoz Moner, A. F., et al, 2005). La simulación se utiliza en muchas fases del ciclo de vida de un

producto: se utiliza dentro de la fase de adquisición para demostrar los beneficios de un producto para los clientes potenciales. En la fase de concepción, la simulación se utiliza para la verificación y se utiliza en las pruebas de hardware, programando bucles. Existen cuatro etapas de simulación: aplicación individual, herramientas de simulación, diseño de sistema basado en la simulación y gemelo digital.

Factory I/O es una simulación de fábrica 3D para tecnologías de automatización de aprendizaje. Es un software fácil de utilizar y permite construir rápidamente una fábrica virtual utilizando una selección de piezas industriales comunes. Factory I/O también incluye muchas escenas inspiradas en aplicaciones industriales típicas, que van desde niveles de dificultad principiantes hasta avanzados. Para el artículo se programó un caso común en la industria, que es el llenado de tanque o control de nivel. Para esto Factory IO posee una escena preestablecida de un tanque, como se observa en la figura 1, este proceso contiene dos actuadores, válvula de llenado y válvula de vaciado, cuenta con dos sensores, sensor de flujo de salida y sensor de nivel. Además de estas variables que existen por defecto se agregó un tablero de control desde el cual se pueda comandar el proceso. Para controlar el proceso se establece una red ethernet con un PLC Siemens S7 1200. El nivel del tanque se regula con un controlador PID.



Fig. 1. Proceso Industrial - Software Factory IO

3. PLATAFORMA IoT

El término IoT hace referencia a los sistemas físicos que reciben y transfieren datos a través de redes inalámbricas con poca intervención humana, lo cual es posible gracias a la integración de los dispositivos informáticos en todo tipo de objetos.

3.1 Servidor

El servidor web es el eje central, atiende todas las consultas, provenientes de la aplicación web. El servidor seleccionado para el diseño de la plataforma IoT es Heroku (Heroku, 2021). Si bien, existen más proveedores informáticos, tras evaluar los costos de uso y la facilidad en el despliegue de un aplicativo web, Heroku sobresale, ya que permite implementar, ejecutar y administrar aplicaciones web, en diferentes lenguajes de programación: Ruby, Node.js, Java, Python, Clojure, Scala, Go y PHP. Para la configuración de microsistemas del servidor se utilizó la librería *flask* de python, la cual permite definir y ejecutar funciones por medio de peticiones HTTP a una url. Para la comunicación del PLC con el servidor se implementó el diagrama de la figura 2. La función de cada dispositivo es:

- EC28J60. Este módulo agrega una interfaz de conexión RJ45 con el arduino uno, para interpretar los datos que este recibe del PLC mediante el protocolo modbus.
- Arduino Uno. Ejecuta la comunicación con el PLC cada determinado tiempo y transmite la información a la tarjeta ESP32s. El arduino debe incluir dentro de su código las librerías *modbus*, *modbusip* y *modbus_enc28j60*.
- ESP32s. Permite la conexión a red wifi y transmite los datos mediante protocolo HTTP al servidor.



Fig. 2. Esquema de comunicación de microprocesadores

3.2 Base de datos

Dado que la función que desempeñará la base de datos es almacenamiento y consulta de datos en tiempo real, requiere de una respuesta de consulta rápida y manejar un gran volumen de datos. Partiendo de la comparación realizada por (Trevisio *et al.*, 2018), en la cual se evaluaron las bases de datos MongoDB, ArangoDB y CouchBase, en dos escenarios: “*Solamente Lectura*” y “*Lectura Pesada*” para dos fuentes de datos. Como resultado de este estudio, MongoDB arrojó los mejores

resultados, en la evaluación del tiempo de respuesta y obtuvo la desviación estándar más baja. Por esta razón la base de datos seleccionada para el artículo fue MongoDB.

Los datos almacenados en MongoDB se dividen en documentos, los cuales tienen una estructura JSON, y un tamaño de almacenamiento límite de 2MB, de manera que el registro histórico de las variables se encuentra dividido en varios documentos, como se observa en la figura 3. El formato de archivos JSON tiene por defecto una variable “_id” para identificar este documento dentro de la base de datos, “varname” indica la numeración del documento, en este caso es “var0” porque es el primer documento creado para este registro. Dado que MongoDB también restringe el número de datos dentro de un vector, el documento está conformado por 16 vectores nombrados “varT” de 1000 espacios, lo que arroja un total de 16.000 datos por documento, esto es equivalente a 1,81 MB, de manera que cumple el tamaño límite. Las variables almacenadas dentro de cada vector, se concatenan en un solo dato tipo “string” separado por medio del símbolo “#”, cada vez que se actualiza el registro, se añade la fecha en la que se almacenó.

3.3 Aplicativo Web

El *back-end* de la página web se programó en JavaScript enlazando con los servicios dispuestos en el servidor Heroku, los cuales están desarrollados en Python. El *front-end*, que corresponde a la parte gráfica del sitio, se utilizan los ficheros de HTML para definir la estructura, agregar funcionalidades y el estilo de la interfaz (Castellanos, W. A., et al, 2018).

4. APLICATIVO MÓVIL

Con el fin de desarrollar una aplicación móvil que pueda ser desplegada en dispositivos Android y IOS, se escogió la biblioteca de C++ Xamarin Forms, esta permite programar todas las funciones de una aplicación móvil en un lenguaje C y exportarla a la estructura de código utilizada por IOS y Android. Para desarrollar una aplicación en Xamarin Forms, se debe descargar Visual Studio Community. Desde la página oficial se puede descargar el instalador, incluyendo los siguientes paquetes, “Desarrollo de ASP.NET”, “Desarrollo de escritorio de .NET” y “Desarrollo para dispositivos móviles con .NET”. Una vez iniciada la aplicación móvil se cargan las librerías y componentes requeridos. El menú de procesos de la aplicación permite seleccionar entre los

diferentes procesos establecidos, una vez seleccionado el proceso que se desea monitorear, se activa una función en segundo plano que realiza una petición al servidor cada dos segundos con el fin de actualizar los datos en tiempo real. En cada página de la aplicación se evalúa si el usuario desea salir con el botón atrás, de ser así la sesión en la aplicación se cierra, hasta que el usuario valide su identidad en la página de inicio. Para verificar el funcionamiento correcto de la aplicación, se emplea el emulador que tiene Visual Studio para Xamarin Forms.

```
{
  "id": {
    "$oid": "5fb1459e1983b0a397b0dc8d"
  },
  "varname": "var0",
  "varT0": [...],
  ],
  "varT1": [...],
  ],
  "varT2": [...],
  ],
  "varT3": [
    "29/9/2020/11/43/22#0#81#4077#17498",
    "29/9/2020/11/43/25#0#81#4076#17499",
    "29/9/2020/11/43/28#0#81#4076#17500",
    "29/9/2020/11/43/31#0#81#4075#17501",
    "29/9/2020/11/43/34#0#81#4074#17502",
  ]
}
```

Fig. 3. Estructura de almacenamiento de datos

En la Figura 4, se muestra la aplicación móvil, navegando de izquierda a derecha, como pantalla inicial se encuentra el inicio de sesión, una vez validado el usuario y contraseña con el servidor, se encuentra una pantalla con la información de proceso simulado (Factory IO).



Fig. 3. Estructura de almacenamiento de datos

5. CONCLUSIONES

El servidor web Heroku ofrece un plan de desarrollo gratuito, sin embargo, se encuentra limitado su tiempo de uso a un mes y la capacidad de trabajo, al desplegar un aplicativo con varias

funciones, hace que el servidor se vuelva lento, por lo que se recomienda ampliar la capacidad de trabajo y segmentar las funciones en microservicios.

El estilo de la interfaz web fue basado en una plantilla, por tal motivo algunas de las características agregadas no comparten el estilo y tiende a perderse la estética, a futuro es recomendado desarrollar una plantilla propia con el fin de agregar las características necesarias que requiere la interfaz.

Dado que la base de datos permite crear una estructura propia, se pudieron evaluar diferentes formas de almacenar los registros históricos de cada proceso, esto permitió mejorar los tiempos de consulta a la base de datos y reducir la carga de trabajo del servidor.

REFERENCIAS

- Castellanos, W. A., Suarez, O. J., & Garcia, A. P. (2018). Usability in virtual learning environments, an approach to the integrated grid (IG) application. LACCEI International Multi-Conference for Engineering, Education and Technology, , 2018-July doi:10.18687/LACCEI2018.1.1.497
- Heroku. (2021). The Heroku Platform. <https://www.heroku.com/platform> (Consultado: 4 de noviembre de 2011)
- Luz D. y Forero L. (2015). Estudio y Diseño de una Plataforma de Entrenamiento de Alto Nivel en Control Electrónico a partir de Sistemas de Control Distribuido (DCS), Universidad Distrital Francisco José De Caldas, Bogotá, Colombia. <https://repository.udistrital.edu.co/handle/11349/4200> (Consultado: 4 de noviembre de 2011)
- Muñoz Moner, A. F., Garcia, A. P., & Díaz Rodríguez, J. L. (2005). Artificial cloning applied to industrial process control. WSEAS Transactions on Systems, 4(7), 930-934.
- Neve C., Moreno U., Boava A. (2018). IoT-Based Distributed Networked Control Systems Architecture, IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turín, Italia. Electronic ISBN:978-1-5386-7108-5, DOI: 10.1109/ETFA.2018.8502500.
- Niveditha A. T., Nivetha M., Priyadharshini K. y Punithavathy K. (2018). IoT Based Distributed Control System Using CAN. Second International Conference on

- Computing Methodologies and Communication (ICCMC), Erode, India, pp. 967-971, doi: 10.1109/ICCMC.2018.8488003, 2018.
- Quispe O. G. (2016). Diseño e implementación de un módulo de pruebas para la simulación de operaciones del sistema de control de procesos (PCS) y del sistema instrumentado de seguridad (SIS) DeltaV Emerson para el área de proyectos y servicios de la empresa SEIN S.A., Quito. Ecuador: ESCUELA POLITÉCNICA NACIONAL. <https://bibdigital.epn.edu.ec/handle/15000/16518?mode=full> (Consultado: 4 de noviembre de 2011)
- Treviño M., Viquez L., Quirós R., Esquivel G. (2018). Una comparación de rendimiento entre MongoDB, ArangoDB y CouchBase para la operación lectura sobre bases de datos geográficas, 2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII), San Salvador, DOI:10.1109/CONCAPAN.2018.8596387
- Wang G., Nixon M., Boudreaux M. (2019). Toward Cloud-Assisted Industrial IoT Platform for Large-Scale Continuous Condition Monitoring, in Proceedings of the IEEE, vol. vol. 107, n°. 6, doi: 10.1109/JPROC.2019.2914021, pp. pp. 1193-1205.