

SISTEMA IoT PARA DETECCIÓN Y MONITOREO DE TICKETS EN LÍNEAS DE PRODUCCIÓN

Banda-Cortez, Jesús Eduardo ¹ Hernández-Maldonado, Nélide Edith ² Ortiz-Aguirre Yara Lizet ³
Flores-Reyes, Yves Ananías ⁴ Benítez-Cortés Myriam ⁵

RESUMEN

El presente artículo tiene como objetivo el desarrollo de un sistema de monitoreo que implemente tecnología IoT para máquinas PLC. En él se realiza el análisis de procesos dentro de los equipos productivos, los cuales se encuentran ubicados en el departamento de producción, con la finalidad de introducir un módulo que consiste en monitorear las máquinas asociadas a las líneas de producción establecidas. Los resultados de esta medida de variables en máquinas PLC provocan además una notificación en caso de que los datos se encuentren anormales, lo que ayuda a la mejora de eficiencia del proceso. El proyecto pretende comprobar que el uso de un sistema de monitorización de máquinas con tecnología IoT es beneficioso para reducir los errores que gestionan los equipos dentro del departamento, ya que de forma automática apertura una alerta (ticket) de mantenimiento si se encuentra con errores, además de mostrar aquellos tickets abiertos y cerrados para el monitoreo constante.

Palabras claves: tecnología de la información, producción, seguimiento y control, gestión de mantenimiento

IoT SYSTEM FOR DETECTION AND MONITORING OF TICKETS IN PRODUCTION LINES

ABSTRACT

The objective of this article is the development of a monitoring system that implements IoT technology for PLC machines. In it, the analysis of processes is carried out within the productive teams, which are located in the production department, with the purpose of introducing a module that consists of monitoring the machines associated with the established production lines. The results of this measurement of variables in PLC machines also cause a notification in case the data is abnormal, which helps to improve the efficiency of the process. The project aims to verify that the use of a machine monitoring system with IoT technology is beneficial to reduce the errors managed by the teams within the department, since it automatically opens a maintenance alert (ticket) if errors are found, in addition to showing those open and closed tickets for constant monitoring.

Keywords: information technology, production, monitoring and control, maintenance management

¹ Universidad Tecnológica de Tamaulipas Norte (México). Email: jesus.banda@uttm.mx .

² Universidad Tecnológica de Tamaulipas Norte (México). Email: nelida.hernandez@uttm.mx

³ Universidad Tecnológica de Tamaulipas Norte (México). Email: yara.ortiz@uttm.mx

⁴ Universidad Tecnológica de Tamaulipas Norte (México). Email: yves.flores@uttm.mx

⁵ Universidad Tecnológica de Tamaulipas Norte (México). Email: myriam.benitez@uttm.mx

Introducción

El uso de los Controladores Lógicos Programales (PLC) en las empresas ha representado una mejora sustancial de muchas tareas repetitivas con las que se contaba en los procesos. La adaptación a esta tecnología permitió en un principio efficientizar y optimizar los tiempos de los procesos para la obtención de mejores resultados gracias a la automatización que otorgaban los mismos. A partir de esto, el uso de Internet de las Cosas se puede considerar una mejora sustancial a ellos, ya que con esto se habilita la opción del monitoreo y control a distancia, además de la automatización del monitoreo mismo al enviar notificaciones en caso de errores.

En el ámbito industrial, el IoT es la columna vertebral para la implementación de la Industria 4.0, que describe cómo los procesos industriales pueden digitalizarse e interconectarse entre sí, incrementando la productividad de estos y mejorando los objetivos planteados. El Internet de las Cosas (IoT), tecnología emergente que permite interconectar en redes inalámbricas a dispositivos comunes para la comunicación de datos, se está integrando muy rápidamente en nuestra sociedad, basta con ver la cantidad de elementos que se pueden encontrar para solucionar los problemas cotidianos mediante Internet.

En el presente artículo se menciona que la consulta y el análisis de informes multidimensionales, las visualizaciones de gráficos y el análisis de big data proporcionan referencias para la buena toma de decisiones en las organizaciones. Este proyecto es un esfuerzo por implementar conceptos de IoT en el trabajo en entornos industriales; se ha diseñado y construido un sistema IoT utilizando herramientas de desarrollo de software para el control y medida de variables máquinas PLC que permita tomar decisiones inteligentes para mantener bajo control el proceso en cuestión. Los resultados de esta medida de variables en máquinas PLC representará además una notificación en caso de que los datos se encuentren anormales, lo que ayudará a la mejora de eficiencia del proceso.

Marco Teórico

La toma de decisiones representa un punto crítico en cada uno de los procesos existentes en las organizaciones debido al impacto que generan en la obtención de resultados. El desarrollo de software, por ejemplo, implementado en procesos y sistemas anteriormente manuales, ha permitido que estos alcancen objetivos de manera más eficaz y efectiva. La decisión de distintas organizaciones de emplear tecnologías emergentes como lo es el desarrollo de software ha significado la mejora de múltiples procesos.

Las tecnologías emergentes son herramientas ideales para mejorar muchos ciclos y sistemas que se siguen en las distintas empresas. El uso de Internet en máquinas permite controlar a distancia, mejorar errores y monitorear resultados de estos.

Los controladores lógicos programables (PLC), computadoras digitales que mejoran procesos en las industrias al trabajar en conjunto con actuadores y sensores, digitales y analógicos, son fundamentales en la automatización y el control de procesos industriales al otorgar métricas necesarias para la toma de decisiones. Estos PLC por lo general realizan operaciones repetitivas y también son por lo general manejados y monitoreados por los empleados.

En el último tiempo se ha visto un cambio tecnológico que sugiere que se está en proceso de transición hacia la cuarta revolución industrial o Industria 4.0. La Industria 4.0 no se basa en una sola tecnología, sino que en la idea de agrupar sistemas con tecnologías similares en torno al concepto de sistemas ciber físicos (CPSs) y en particular al de sistemas ciber físicos industriales (ICPSs), característica que diferencia a esta revolución tecnológica de las anteriores. (Mellado Aceitón, 2020)

El uso del IoT permite en la Industria 4.0 que los procesos y las máquinas industriales se interconecten en una red para la adquisición de datos, que generen estadísticas y métricas en tiempo real para su análisis y, en caso necesario, la modificación del proceso para la comprobación de errores y/o mejora de los resultados.

Desde 2015 se ha producido un avance considerable en las metodologías de adquisición y procesamiento de señales debido a la aparición de nuevas tecnologías como WSN (*wireless sensor network*), IoT y *machine learning*, lo que sin duda revoluciona la forma de monitoreo de datos sensibles para los procesos industriales. La clave para tener fabricas inteligentes está en cambiar las máquinas tradicionales con modernas.

Las metodologías de monitoreo de maquinaria de procesos se pueden catalogar de dos maneras: la pasada y la presente, esta última muestra la integración de 3S; sistema de detección, el sistema de decisiones y el sistema de control para el monitoreo de las condiciones de la maquinaria de proceso. WSN se integra a las máquinas tradicionales para permitir la recopilación y transferencia de datos (Iliyas et al., 2020).

Metodología

El proyecto se desarrolla en diferentes etapas: primero se realiza un análisis de todos los requerimientos, una vez identificadas las necesidades del proceso se analiza el llevar a cabo el desarrollo tecnológico en dos fases principalmente. Una vez que se prueban e implementan cada una de las fases de desarrollo se lleva cabo el análisis de los resultados. Este proceso se encuentra plasmado en la figura 1.

Figura 1.

Método del desarrollo



Análisis de la Problemática

En el sector industrial es común que las empresas hagan uso de maquinaria para realizar sus procesos de manufactura. Muchos de estos equipos cuentan con una serie de dispositivos para controlar, regular o hacer que funcione una máquina y en conjunto hacen que una organización puede fabricar productos en masa o gran escala. Esto permite diferentes tipos de productos de distintas formas, por lo que es necesaria diferente tipo de maquinaria para cada producto a elaborar.

Es común que en los procesos de ensamble en los que en su mayoría las máquinas son operadas por personal humano estén propensos a errores, pero también se pueden presentar ocasiones en los que un proceso se ve involucrado en un error no causado por un operador de producción. Por ejemplo, la falta de materiales o materia prima en general para la continuación de los procesos, la presencia de fallo de hardware o software de un producto en ensamble, el desgaste de las herramientas que usan las máquinas de ensamble en cada proceso.

Gestionar todos los procesos de ensamble de dichas máquinas es algo que aseguraría a la empresa conocer el desempeño de sus operaciones, permite realizar acciones de prevención ante situaciones o corrección y detección de errores en los procesos de ensamble, es por eso que se realiza el proyecto que consiste en el desarrollo de un sistema para la detección de errores dentro de los procesos de producción de la organización, consiste en un sistema de IoT para la detección de errores de forma automatizada en múltiples procesos de ensamble y se complementa con un sistema de generación de tickets que busca cerrar el ciclo de detección y corrección de errores.

Fase 1: Sistema de monitoreo

Al inicio se contaba con una aplicación basada en web para capturar el tiempo de inactividad de las líneas de producción, estos eventos se registraban manualmente por los supervisores de producción en función de sus actividades de monitoreo. El registro de estos tiempos de inactividad dependía de la disponibilidad del supervisor lo cual generaba un retraso en la exactitud con la que se generaban los reportes, el proceso

era el siguiente: una vez que se detectaba un tiempo de inactividad el supervisor debía notificarlo en la aplicación web de manera manual, escribiendo los parámetros requeridos como la línea en la que sucedió, el turno, el proceso en el que sucedió un incidente y una descripción de lo sucedido.

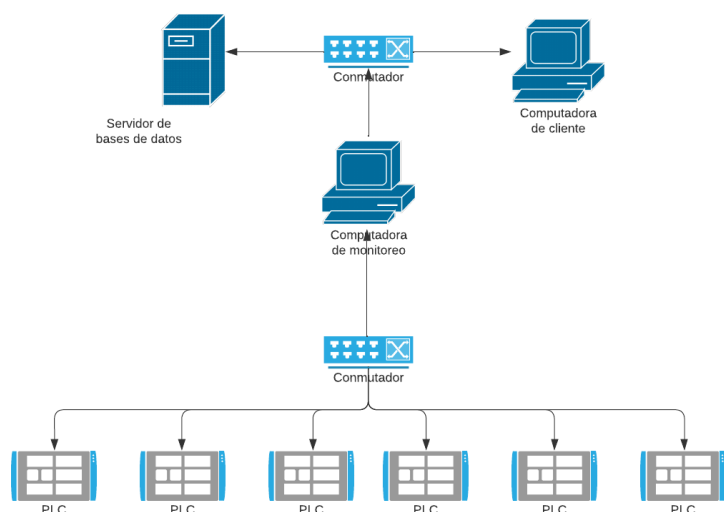
Para mejorar la situación anterior se desarrolla el proyecto de Sistema IoT para detección y monitoreo de tickets en líneas de producción el cual en su primera fase contempla una serie de etapas que se documentan como sigue.

Estructura de Red

La estructura de la red se muestra en la siguiente figura, contiene un computador que funciona como host encargado de realizar el monitoreo de los PLC's de una línea de ensamble. Se encuentra conectado a dos redes, una red en donde se encuentran los PLC's y a la red corporativa del cliente. Para realizar esta conexión se utilizan dos tarjetas de red en la computadora de monitoreo.

Figura 2.

Estructura de red



El servidor de base de datos se encuentra en la red corporativa y es el encargado de almacenar y proporcionar datos a los equipos conectados a la red. La computadora del cliente es cualquiera que requiera acceder a la información recolectada por la computadora de monitoreo que permite acceder a los datos proporcionados accediendo a ellos por el servidor de base de datos que se encuentra en la misma red.

Estructura de sitio

La estructura del sitio se muestra en la figura siguiente, en esta se puede apreciar parte de la estructura de la aplicación web. La navegación en el sistema web se divide en cuatro secciones, tres de ellas se relacionan al acceso de datos de monitoreo para

PLC's, hosts y líneas de ensamble. Para el acceso a las vistas de registro de equipos, líneas y errores es necesario acceder al sistema web con un usuario y contraseña válidos, es por ello que se incluye un inicio de sesión para poder realizar estas funciones.

Base de Datos

La estructura de la base de datos está conformada por trece tablas dedicadas al sistema de monitoreo y cinco tablas dedicadas al sistema de inicios de sesión y control de cuentas de usuario para la plataforma web, todas las tablas que excluyen el término de usuarios y roles de usuario son las que trabajan con el sistema de monitoreo de PLCs y todas las tablas en general trabajan con la aplicación web para la muestra de datos de errores, procesos, información de la línea y rastreo en tiempo real.

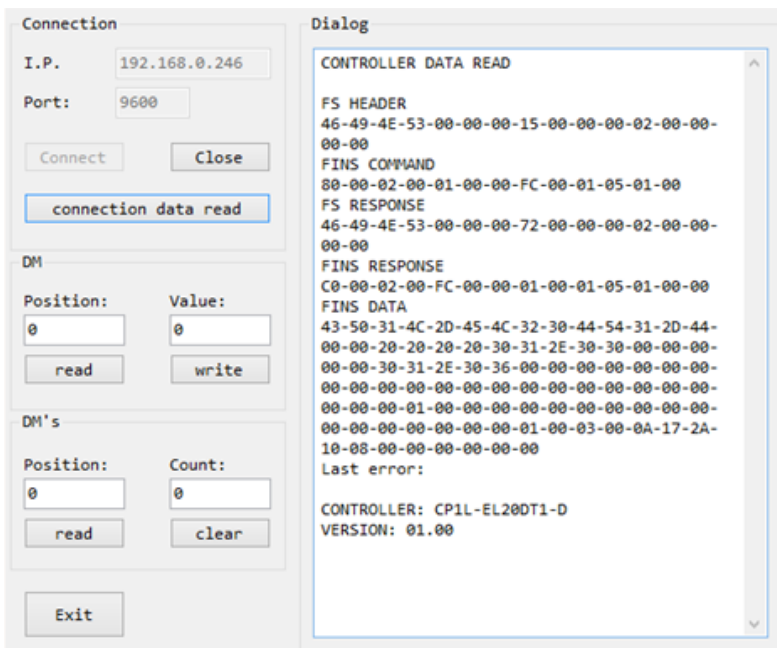
Desarrollo de API y aplicaciones

En las máquinas de ensamble se utilizan PLCs OMRON para la gestión de sus procesos, como primera fase inicia el desarrollo de una solución para obtener datos almacenados en los PLC, se opta por utilizar .Net Framework 4.5 porque esta versión viene instalada por defecto en todos los equipos de la empresa, para esto se escribe una clase que de varias clases, la clase principal, esta clase utiliza sockets síncronos para la comunicación con los PLC, una clase de capa de transporte (TCP) y una clase para administrar las ordenes FINS a través de TCP. La capa de transporte se basa en dos interfaces ITransport y IFINSCommand.

El primer prototipo del software se basa en una aplicación de escritorio la cual permite conectarse a un solo PLC ingresando en un campo de texto la dirección IP y el puerto del PLC al que se desea conectar, esta primera versión permite realizar una conexión al PLC, cerrar conexión y leer o escribir valores de las memorias DM. Cuenta con un cuadro de dialogo en donde se muestra la información recolectada del PLC, en este cuadro de dialogo se observa la respuesta del PLC, muestra información acerca de FINS Command y Response que se devuelve como datos binarios y son convertidos a hexadecimal, en caso de presentar un error en la comunicación indica cual es el error y también informa la que controlador de PLC es el que se está usando, así como su versión. Estos valores se muestran en la figura 3.

Figura 3.

Primer prototipo del software



Para el desarrollo del software de almacenamiento de datos se utiliza la clase base creada en el desarrollo de la API de comunicación. El desarrollo se divide en dos proyectos dentro de una solución de Visual Studio, el proyecto A se encarga de ejecutar algunos métodos como la conversión de direcciones de PLC, los protocolos de comunicación con el PLC, la descarga de información necesaria para la ejecución del programa, el almacenamiento de información del PLC a la base de datos, la conexión con la base de datos mientras que el proyecto B se encarga de ejecutar toda la lógica detrás de la interfaz gráfica del programa.

El prototipo posterior se basa en el desarrollo de una interfaz utilizando una interfaz multiproceso para la ejecución de múltiples tareas al mismo tiempo. Esto es necesario porque una PC puede estar monitoreando la actividad de uno o más PLCs, por lo que es necesario que el programa se adapte a las necesidades presentes en una línea de producción. El desarrollo inicial consiste en una interfaz desde la cual se puede acceder a información de múltiples PLCs para obtener principalmente información de trazabilidad y el modelo de producto de múltiples PLCs.

Finalmente, se desarrolla una aplicación web en ASP.NET, que consiste en un sistema tipo panel de control informativo en donde el usuario podrá acceder a la información almacenada por el sistema de monitoreo en la base de datos. La principal es la vista se puede apreciar en la figura siguiente.

Al hacer clic en la dirección IP del PLC en la vista anterior se abre una vista que ofrece información más detallada acerca de los procesos de ese PLC, como primera parte, muestra una tabla con todos los errores detectados por el software de monitoreo en los procesos de ensamble, la tabla muestra la dirección de bit o Bit Address del PLC

correspondiente al error, una breve descripción del error y la fecha con hora en la que fue detectado.

Pruebas y mejoras

Se realizan pruebas de funcionalidad, rendimiento e integridad de datos. La prueba de funcionalidad validó cómo funciona el sistema frente a algún requisito. En este caso se verificó la funcionalidad del sistema para la modificación de parámetros de PLC's. Para poder realizarlo el usuario debe estar autenticado en la aplicación web, en caso contrario no otorga acceso a esa función y pide credenciales válidas.

La prueba de rendimiento se realizan la página que más carga exige al servidor de base de datos que es la principal de la aplicación web porque constantemente se encuentra haciendo peticiones al servidor para mantener actualizada la información de las líneas de ensamble en tiempo real. Para esto se abrieron múltiples ventanas con la principal ventana de monitoreo cada una en una línea diferente sin presentar problema en el rendimiento.

La prueba de integridad de datos se realiza en la aplicación de monitoreo, esta prueba pretende evitar el duplicado de datos y garantizar la exactitud y fiabilidad de estos mismos, para esto se realiza una prueba abriendo nuevamente el software de monitoreo cuando este ya se encuentra trabajando, como resultado no permite abrir nuevamente el software de monitoreo, envía un mensaje al usuario de que solo se permite la ejecución del software una vez por PC, esto para evitar que dos software registren la misma información recolectada dos veces al servidor de base de datos.

Implementación

Esta implica la configuración de software y hardware para la puesta en marcha del proyecto. Por parte del sistema de monitoreo, es necesario registrar previamente los PC y PLC's que serán monitoreados por el software, ya que este no se ejecutará si no encuentra información almacenada en la base de datos vinculada a la dirección MAC del equipo que ejecuta el software de monitoreo, otra recomendación es verificar que la posición del PLC en una línea sea la correcta, esto se debe a que la posición del equipo influye en el funcionamiento del sistema de monitoreo y la plataforma web. También se recomienda que todas las PC ejecuten la misma versión del software.

Por parte de la plataforma web, es necesario revisar la validación de datos al momento de modificar o agregar valores, ya que un valor erróneo puede ocasionar la falla del software de monitoreo en uno o más PC's, esto es muy recomendable al momento de agregar nuevos errores a un PLC, es recomendable revisar la vista previa que arroja la plataforma web para asegurarse de que los datos serán confiables.

Fase 2. Sistema de tickets

Después de desarrollar el sistema de monitoreo, la fase dos se enfoca en la detección de errores y generación de Tickets para Mantenimiento Correctivo y así llevar una correlación en tiempos de producción y tiempos perdidos en las líneas de la empresa.

Para lograrlo se desarrolla una aplicación de IoT (Internet of Things) la cual recopila el estado de las estaciones de trabajo en tiempo real para detectar los problemas y generar un ticket automáticamente obteniendo un control más exacto y sin intervención humana. Este proceso permite llevar a cabo un control de los registros de tiempos perdidos y obtener promedios de errores reportados y así los técnicos ataquen los problemas creando un plan de seguimiento para realizar mantenimiento preventivo.

Requerimientos funcionales y no funcionales

Para el proyecto se toman en cuenta los siguientes requerimientos funcionales:

1. Detección de errores: El sistema de tickets debe detectar errores que generen las máquinas de manera automatizada.
2. Generación de tickets: El sistema deberá generar un ticket con la información pertinente a la falla que se presente en la máquina de forma automática.
3. Mostrar tickets en tiempo real: El sistema debe mostrar una interfaz que se actualice en tiempo real para mostrar los tickets generados en ambas aplicaciones.
4. Captura de soporte:
5. Sistema de notificaciones: El sistema debe alertar a los técnicos e ingenieros mediante notificaciones por correo electrónico cuando se genere un ticket.
6. Listado de tickets: El sistema debe contar con una interfaz web donde se muestre el historial de tickets y se puedan filtrar por línea, estatus, si el ticket está abierto o cerrado.
7. Detalles de los tickets: El sistema debe mostrar la información detallada del ticket como la información relacionada con los tiempos y soporte.

También se toman en cuenta los requerimientos no funcionales:

1. Bajo consumo de recursos: La aplicación no deberá de consumir una gran cantidad de memoria al momento de estar corriendo continuamente.

Estructura y configuración de la base de datos

La base de datos también se reutiliza, además se usa Linked Servers para poder acceder a otra tabla en un servidor distinto, esto porque producción necesita tener datos de los paros que se realicen para que tengan la posibilidad de realizar métricos independiente del sistema de IoT_Monitoring y que se utilizaran en el sistema de generación de tickets y monitoreo de las máquinas. Para que todos los sistemas trabajen en conjunto se agregan las siguientes tablas a la base de datos:

- IoT_ErrorEvents: en esta tabla son almacenados todos los eventos producidos por cada máquina, es requerido que existan estos registros debido a que se necesita llevar un monitoreo de cada máquina y del evento que está realizando, por otro lado, también es necesario este historial de eventos para realizar métricos respecto a tiempos en cada proceso realizado.

- **IoT_Notify_Reports:** en esta tabla se almacenan todos los eventos que fueron disparados por alguna bandera de notificación activa, esto para generar un historial de cada alerta generada.
- **IoT_Notify_Flags:** en esta tabla se almacenan todas las banderas de notificaciones generadas por los usuarios.
- **IoT_Notify_EventError:** esta tabla almacena las notificaciones generadas cuando se abre un ticket, esto para prevenir que se envíen múltiples notificaciones.
- **IoT_Machine_Notify:** esta tabla almacena todas las alertas sobre maquinas personalizadas.
- **IoT_AlarmName:** en esta tabla se encuentran los tipos de alertas que se pueden poner a la escucha en una bandera.

Desarrollo del sistema de tickets

Se hizo el uso de la API de monitoreo la cual contiene las clases necesarias para realizar la conexión a los PLC, de esta manera se hace más óptimo el desarrollo de esta implementación.

Al momento que se genera un nuevo evento, se identifica si el evento pertenece a un error que se encuentra ya previamente registrado en la tabla `IoT_ErrorTypes` si no se encuentra en esta tabla solo almacena el evento dentro de la tabla `IoT_EquipmentEvents`, de lo contrario se verifica si la máquina se encontraba operando anteriormente, esto para prevenir falsos eventos de error. Posterior a esto se identifican todos los errores presentes en la maquina reportados por el PLC y se obtiene su descripción desde la tabla `IoT_ErrorTypes` despues se corrobora si alguno de esos errores aplica para generar un ticket de mantenimiento correctivo. Después, al cumplirse esta condición se valida si este evento lleva más de un minuto para generar el ticket de mantenimiento correctivo dentro de la tabla `IoT_ErrorEvents`, esto debido a que un evento de error de cualquier tipo puede tener menos de un minuto, pero esto no se reconoce como un evento para generar un ticket y es omitido. Sin embargo, si el error no califica para que se genere un ticket entonces solo se almacena el evento en la tabla `IoT_EquipmentEvents`.

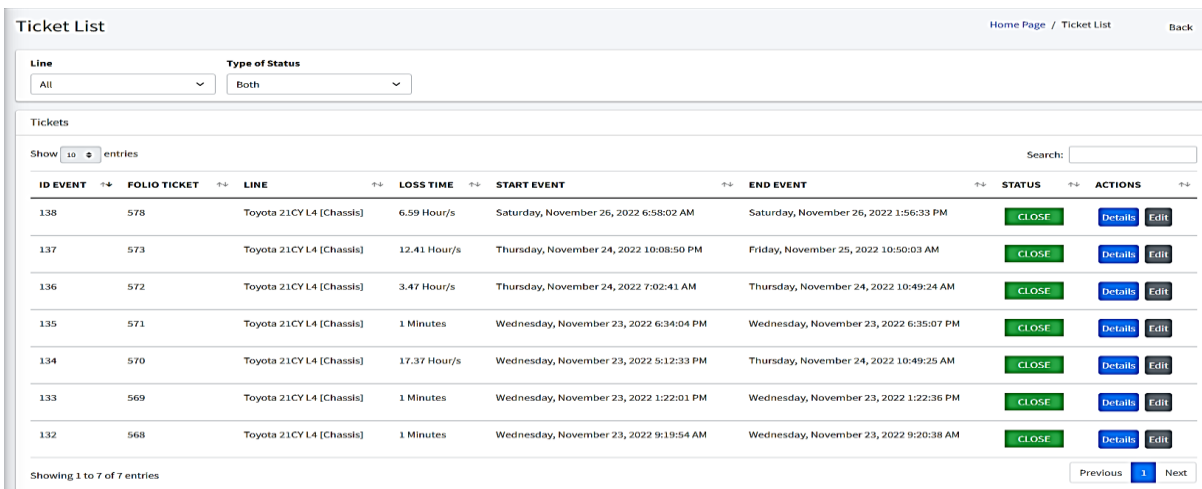
Todo este procedimiento se realiza paralelamente en el mismo programa, es decir, si una línea en producción tiene cinco máquinas trabajando, cualquiera de estas máquinas puede tener un problema o más bien puede estar corriendo correctamente. Para que el software no tenga problemas de rendimiento o de encolamiento de procesos, se tomó la decisión de crear un hilo para cada máquina, de esta manera, cada hilo repetirá este proceso de monitoreo y detección de eventos de error paralelamente a los demás por cada máquina que esté trabajando en la línea.

Después de analizar el nuevo flujo de la aplicación, se crea una nueva sección en la aplicación de escritorio para mostrar todos los tickets generados por el sistema en la línea en la que se encuentra instalado.

Por otro lado, en la aplicación web se muestra, de igual manera y en tiempo real, un listado de todos los tickets generados por las aplicaciones de monitoreo, como se observa en la siguiente figura 4.

Figura 4.

Listado de tickets generados



ID EVENT	FOLIO TICKET	LINE	LOSS TIME	START EVENT	END EVENT	STATUS	ACTIONS
138	578	Toyota 21CY L4 [Chassis]	6.59 Hour/s	Saturday, November 26, 2022 6:58:02 AM	Saturday, November 26, 2022 1:56:33 PM	CLOSE	Details Edit
137	573	Toyota 21CY L4 [Chassis]	12.41 Hour/s	Thursday, November 24, 2022 10:08:50 PM	Friday, November 25, 2022 10:50:03 AM	CLOSE	Details Edit
136	572	Toyota 21CY L4 [Chassis]	3.47 Hour/s	Thursday, November 24, 2022 7:02:41 AM	Thursday, November 24, 2022 10:49:24 AM	CLOSE	Details Edit
135	571	Toyota 21CY L4 [Chassis]	1 Minutes	Wednesday, November 23, 2022 6:34:04 PM	Wednesday, November 23, 2022 6:35:07 PM	CLOSE	Details Edit
134	570	Toyota 21CY L4 [Chassis]	17.37 Hour/s	Wednesday, November 23, 2022 5:12:33 PM	Thursday, November 24, 2022 10:49:25 AM	CLOSE	Details Edit
133	569	Toyota 21CY L4 [Chassis]	1 Minutes	Wednesday, November 23, 2022 1:22:01 PM	Wednesday, November 23, 2022 1:22:36 PM	CLOSE	Details Edit
132	568	Toyota 21CY L4 [Chassis]	1 Minutes	Wednesday, November 23, 2022 9:19:54 AM	Wednesday, November 23, 2022 9:20:38 AM	CLOSE	Details Edit

Sistema de notificaciones

El sistema de notificaciones es fundamental para este sistema de monitoreo, ya que su función principal es notificar a los técnicos cuando algún evento ha ocurrido. Este sistema cuenta con múltiples eventos de alerta, es decir, múltiples tipos de notificaciones, por ejemplo:

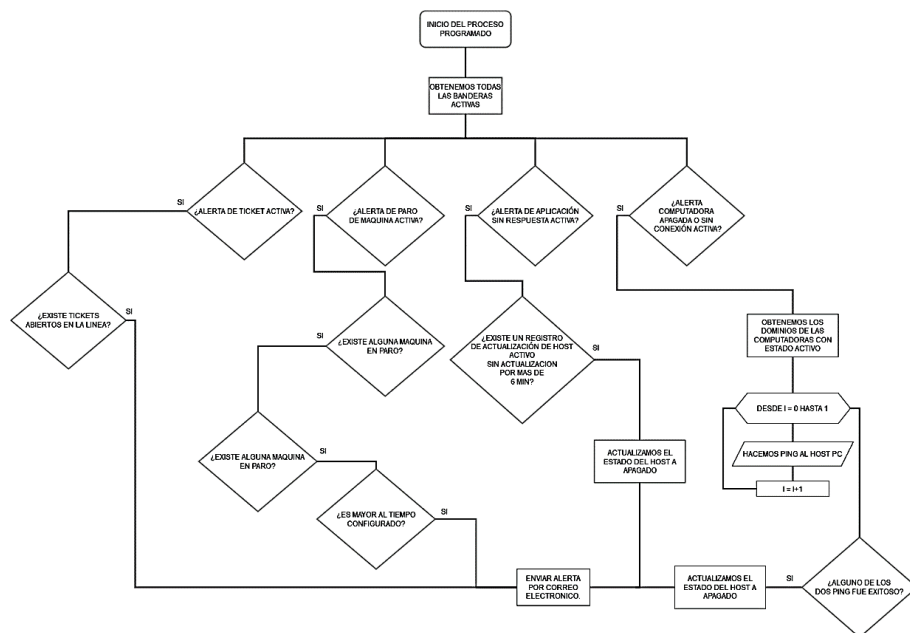
- Evento a la escucha de un ticket generado.
- Evento a la escucha de paro de máquina.
- Aplicación no responde.
- Computadora host esta apagada o sin conexión a internet.

El sistema de notificaciones genera alertas por correo electrónico; la idea principal es que los técnicos encargados de sus líneas o ingenieros sean alertados cuando se genere un ticket para mantenimiento correctivo, de esta manera no se pierde tiempo al momento en el que se presente la falla y se le informa al técnico. Por consiguiente, no habrá recurso humano encargado de notificarle a los técnicos cuando una falla se presente en alguna de las maquinas o se genere un ticket para mantenimiento correctivo.

Para cumplir con los requerimientos se diseñó el sistema de la siguiente manera como se muestra en la siguiente figura:

Figura 5.

Diagrama de flujo del sistema de notificaciones



Con el fin de lograr desarrollar este sistema se crea un trabajo programado el cual se ejecuta cada ocho segundos. Este trabajo programado se encuentra ejecutándose en la aplicación web, accediendo a la base de datos principal donde se encuentran todos los registros que son requeridos para hacer el proceso de notificaciones. Los técnicos podrán generar banderas personalizadas donde pueden crear alertas de eventos. Con este nivel de personalización los usuarios que generen alertas se aseguran solo de recibir alertas de las máquinas y de las líneas en las que solo están interesados.

En la vista principal del controlador de tickets hay una interfaz donde los usuarios pueden ver el listado de banderas que han creado; por otro lado, también pueden filtrarlo. En cada bandera se pueden activar 4 tipos de alertas por eventos: evento por ticket generado, evento por paro de máquina, aplicación de monitoreo no responde, computadora o host apagado o sin conexión a internet. Cada evento tiene una función, a continuación, serán explicados:

- Evento por ticket generado: Esta alerta se mantiene a la escucha de tickets que se hayan generado en alguna de las máquinas de la línea donde se haya configurado la bandera.
- Evento por paro de maquina: Esta alerta se mantiene a la escucha si alguna de las maquinas que han sido seleccionadas en la configuración haya entrado en modo de falla y si pasa de cierto umbral de tiempo configurado se envía un correo al usuario que haya configurado la alerta.
- Aplicación no responde: Esta alerta se mantiene a la escucha de cualquier aplicación donde se esté instalado el software de monitoreo y no haya realizado alguna actualización, pero se encuentre con estado activo.

- **Computadora/host apagado o sin conexión:** Esta alerta se mantiene a la escucha de todas las computadoras donde se encuentre instalado el sistema de monitoreo, el Job encargado de realizar este proceso realiza ping a todas las computadoras, si alguna de las computadoras rechaza el ping por dos intentos consecutivos se genera una alerta y se les notifica a todos los usuarios que tengan encendida esta alerta.

Pruebas y mejoras

En la fase de pruebas se inicia con una prueba funcional en el sistema de tickets donde se hace el uso de un usuario de control para mostrar la información del ticket y dependiendo del estado en el que se encuentra el ticket se mostrara un color que lo identifique. Se realiza además una prueba con el sistema de notificaciones donde se crean alertas en una máquina, generando un ticket de mantenimiento correctivo para recibir la notificación vía correo electrónico que informa sobre el tipo de alerta y por qué se produjo.

Mejoras

Durante todo el proceso de desarrollo se realizaron diversas modificaciones al sistema en general, ya que el programa de monitoreo aumentaba el consumo de memoria RAM exponencialmente, esto ocasionando que el sistema operativo Windows lo cierre de manera automática, viendo este inconveniente se dio a la tarea de utilizar un recolector de basura que está incluido en .Net Framework, esto ayuda a eliminar los objetos creados y que no sean utilizados por el programa, esto bajo en gran medida el consumo de memoria RAM de la aplicación. Es importante realizar el seguimiento correcto a este tipo de situaciones a esta aplicación ya que es el punto central de este sistema de monitoreo IoT.

Resultados

La plataforma implementada basada en el concepto de Industria 4.0, permite la integración completa de las unidades del proceso desarrollado, un enfoque aplicable e importante para la transformación industrial. Los protocolos utilizados para el proyecto son TCP (Transmission Control Protocol) y FINS (Factory Interface Network Service), ambos con características de comunicación necesarias para la implementación de este proyecto. Este proyecto puede ser beneficioso para empresas públicas y privadas que deseen implementar una plataforma industrial para la gestión de procesos. El proyecto está desarrollado para Windows. Consta de dos clientes, que es el proceso de control de un PLC y comunicación con la computadora Windows a través del puerto 9600.

Al realizar esta implementación se redujeron en horas un total de 9.5 Horas/semana, esto debido a que el operador ya no necesita notificarle al supervisor o al operador al momento en el que se detecta una falla, el supervisor ya no necesita realizar el procedimiento de registro de tickets y el técnico se le redujo un total de 71% de tiempo utilizado para darle seguimiento a tickets generados, ya que su tarea es a finalizar el turno seria detallar en cada ticket la información de soporte que se le dio al equipo.

Discusión

Debido a que el proyecto se realiza en dos fases con dos programadores distintos, una de las problemáticas encontradas durante la implementación fue la falta información y documentación que explicara el funcionamiento del sistema. En este caso fue necesario realizar ingeniería inversa al código fuente de la aplicación de escritorio y la aplicación web para poder ver su funcionamiento interno.

Los resultados del proyecto se obtuvieron al analizar el tiempo que tardaba el proceso antes de la implementación del proyecto, y una vez implementado el mismo. Esto permite realizar un cálculo costo beneficio, en el cual se toma en cuenta el salario de los empleados involucrados, y la diferencia del tiempo empleado por cada uno de ellos antes y después de la implementación.

Conclusiones

El proyecto al terminar sus fases de desarrollo logra hacer una contribución muy importante para monitorear, identificar y poder mejorar las fallas en el proceso en tiempo real, además que permite una toma de decisiones inteligente basada en información. A partir de esta mejora se reduce el tiempo de atención que los operadores y los supervisores invierten en el proceso de alertar y generar un ticket para mantenimiento correctivo; se lleva un control más exacto de los registros de los tickets de mantenimiento correctivo, permitiendo tomar estos datos para tener informes y métricos de tiempo perdido por línea y por máquina para generar reportes.

Se demuestra que el uso del Internet de las Cosas permite optimizar los procesos en la industria, automatizando el trabajo que anteriormente se debía manipular con recursos humanos.

Referencias

- Cortés, C. B. Y., Landeta, J. M. I., & Chacón, J. G. B. (2017). *El entorno de la industria 4.0: implicaciones y perspectivas futuras*. Conciencia Tecnológica [en línea] Nº 54, Recuperado el 24 de Febrero de 2022 de <https://dialnet.unirioja.es/servlet/articulo?codigo=6405835>
- Iliyas Ahmad, M., Yusof, Y., Daud, M. E., Latiff, K., Abdul Kadir, A. Z., & Saif, Y. (2020). *Machine monitoring system: a decade in review*. The International Journal of Advanced Manufacturing Technology [en línea] 108, 3645–3659 Recuperado el 23 de Febrero de 2022 de <https://doi.org/10.1007/s00170-020-05620-3>
- Mellado Aceitón, J. (2020). *El IOT-PLC: una nueva generación de controladores lógicos programables para la industria 4.0* [Tesis de maestría]. Pontificia Universidad Católica de Chile.
- Rana, Kuldeep. (2021). *Prototype Model*. Art Of Testing. Recuperado el 20 de Febrero de 2022 de <https://artoftesting.com/prototype-model>
- Schwalbe, Kathy. (2012). *Managing a project using an agile approach and the PMBOK® guide*. Recuperado el 25 de Febrero de 2022 de <https://kathyschwalbe.files.wordpress.com/2013/06/managing-a-project-using-an-agile-approach-and-the-pmbokc2ae-guide.pdf>