TeCEO
Revista Tecnología, Ciencia y Estudios Organizacionales

TeCEO, 6(12), 2024.
Revista Tecnología, Ciencia y Estudios Organizacionales / ISSN: 2594-1410
https://doi.org/10.56913/teceo.6.12.68-82

Mantenimiento Predictivo para Gestión de Servidores Utilizando Tecnología IoT

Predictive Maintenance for Server Management Using IoT Technology

- **Gloria Itzel Campuzano-Guillen** es profesora de la Universidad Tecnológica de Bahía de Banderas (México) (gcampuzano@utbb.edu.mx) (https://orcid.org/0009-0005-0704-6399), Ingeniero.
- **Erick Ramírez-Gil** desarrollador de software de la Universidad Tecnológica de Bahía de Banderas (México) (tabannie@gmail.com) (https://orcid.org/0009-0005-8843-3373), Ingeniero.
- **Alejandro Sánchez-Aguilar** desarrollador de software de la Universidad Tecnológica de Bahía de Banderas (México) (tabannie@gmail.com) (https://orcid.org/0009-0009-2978-915X), Ingeniero.

Resumen: En el presente estudio se centra en el desarrollo e implementación de un sistema de mantenimiento predictivo en la Universidad Tecnológica de Bahía de Banderas. Con el objetivo de llevar a cabo la gestión de los servidores en tiempo real y así optimizar las estrategias de mantenimiento predictivo, toma de decisiones y garantizar la continuidad de los servicios informáticos sin pérdida de datos, y mantener la continuidad de los procesos, desarrollando herramientas robustas que permitan prevenir fallas inesperadas, y así poder llegar a la toma de decisiones ante cualquier situación de riesgo de manera oportuna implementando las tecnologías IoT en servidores Raspberry Pi.

Palabras clave: Gestión de servidores, mantenimiento predictivo, monitoreo en tiempo real, internet de las cosas, computación en la nube.

Abstract: Bahía de Banderas University of Technology. The objective is to manage servers in real time and thus optimize predictive maintenance strategies, decision making and ensure the continuity of IT services without data loss, and maintain the continuity of processes, developing robust tools that allow preventing unexpected failures, and thus being able to make decisions in the face of any risk situation in a timely manner by implementing IoT technologies on Raspberry Pi servers.

Keywords: Server management, predictive maintenance, real-time monitoring, internet of things, cloud computing.

Cómo citar Campuzano-Guillen, G.I., Ramírez-Gil, E. y Sánchez-Aguilar, A. (2024). Mantenimiento Predictivo para Gestión de Servidores Utilizando Tecnología IoT**.** *Tecnología, Ciencia y Estudios Organizacionales*, 6(12), pp. 68–82. https://doi.org/10.56913/teceo.6.12.68-82

Recepción: 30-09-2024
Aprobación: 23-10-2024

Introducción

El almacenamiento de datos de las empresas es de suma importancia, ya que esto permite que la información esté disponible de manera inmediata para fines comerciales, esto permite la eficiencia y productividad de los empleados y mejora la experiencia de los usuarios, implementar el almacenamiento de información tiene como consecuencia diferentes beneficios, tales como el análisis de datos de manera oportuna, verificando que la información esté bien registrada para que pueda ser analizada, procesada y administrada con mayor eficiencia, para la toma de decisiones de manera oportuna, hoy en día una de las tecnologías de la actualidad para almacenar información es por medio de servidores, "en informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos. Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor" (Andrade, 2014).

Los servidores han revolucionado la forma en que las empresas almacenan y gestionan sus datos. Al centralizar la información en un único lugar, los servidores facilitan el acceso a los datos desde cualquier dispositivo con conexión a internet, lo que agiliza los procesos de trabajo y mejora la colaboración entre equipos. Además, los servidores ofrecen una mayor seguridad para los datos, ya que permiten implementar medidas de protección robustas, como sistemas de respaldo, firewalls y encriptación. Esta seguridad es fundamental para proteger la información confidencial de las empresas y cumplir con las normativas de privacidad de datos. Asimismo, los servidores escalables permiten a las empresas adaptarse a las crecientes demandas de almacenamiento de datos, lo que es esencial en un entorno empresarial en constante evolución.

El mantenimiento de los servidores es crucial para garantizar su correcto funcionamiento y prolongar su vida útil. Existen diferentes tipos de mantenimiento, cada uno con sus objetivos específicos. El mantenimiento preventivo se enfoca en realizar tareas de limpieza, actualización de software y hardware, y monitoreo constante para detectar posibles problemas antes de que se conviertan en fallas mayores. Por otro lado, el mantenimiento correctivo se lleva a cabo cuando ya ha ocurrido una falla y se requiere reparar o reemplazar componentes dañados. Además, el mantenimiento predictivo utiliza herramientas de análisis para predecir posibles fallas futuras y realizar acciones preventivas para evitarlas. Estos tipos de mantenimiento, combinados con una adecuada gestión de la infraestructura, permiten optimizar el rendimiento de los servidores y minimizar el tiempo de inactividad, la integración de un sistema web para mantenimiento predictivo en servidores Raspberry Pi ofrece múltiples ventajas. En primer lugar, permite monitorear en tiempo real el estado de los servidores, detectando anomalías y posibles fallas antes de que se produzcan, lo que minimiza el tiempo de inactividad y reduce los costos de reparación. Además, al centralizar la información en una plataforma web, se facilita el acceso a los datos desde cualquier dispositivo con conexión a internet, lo que agiliza la toma de decisiones y la colaboración entre equipos. Asimismo, los sistemas web permiten generar informes detallados sobre el estado de los servidores, lo que facilita la planificación del mantenimiento y la identificación de patrones que pueden ayudar a mejorar la eficiencia del sistema.

Método

El objetivo de este artículo fue desarrollar una un sistema de mantenimiento predictivo, el cual tiene la capacidad de analizar distintas variables para permitir la toma de decisiones, dicha integración del sistema se implementó para los servidores de la Universidad Tecnológica de Bahía de Banderas, la cual cuenta con una instalación de servidores Raspberry Pi para la administración del sistema escolar de expedientes digitales "SEED", el cual es una integración de todas las áreas de la institución en una sola plataforma.

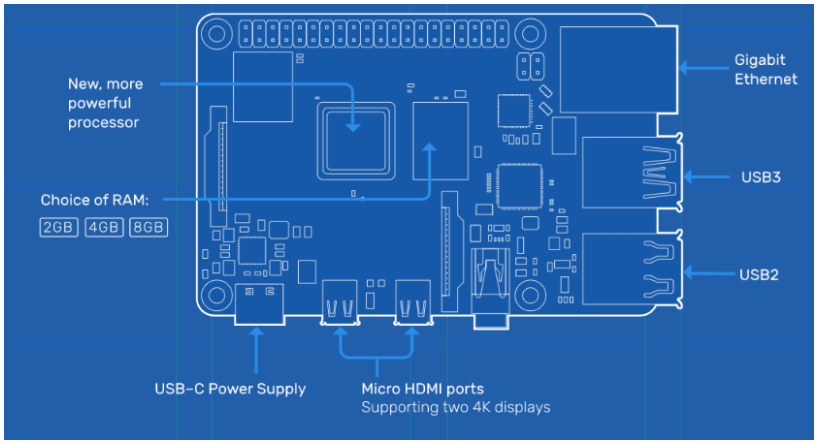
Materiales y dispositivos

Se tomaron como base para el desarrollo 4 Raspberry Pi 4, 1 sensor dht11, 1 Arduino UNO, 1 ESP8266, 1 protoboard mini, 1 buzzer, 1 pantalla LCD16x2 Ic2, 1 caja de registro impresa en 3D, sensor de temperatura DHT11.

Raspberry Pi 4

Raspberry Pi 4 se muestra en la Figura 1, es una computadora de placa reducida cuyas principales características son, Procesador ARM Cortex-A72, memoria RAM 1 GB / 2 GB / 4 GB LPDDR4 SDRAM, 1 GB, 2 GB, 4 GB u 8 GB LPDDR4-3200 SDRAM, conexión inalámbrica IEEE 802.11ac de 2,4 GHz y 5,0 GHz, bluetooth 5.0, BLE.

Figura 1. *Representa las especificaciones técnicas de Raspberry Pi 4.*

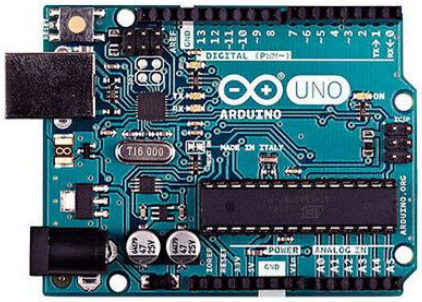


Arduino uno

Arduino uno, Figura 2, es una de las placas más utilizadas cuyas principales características son, microcontrolador ATMega328P, velocidad de reloj 16 MHz, voltaje de trabajo 5V, voltaje de entrada 7,5 a 12 voltios, pinout 14 pines digitales (6 PWM) y 6 pines analógicos, 1 puerto serie por hardware, memoria 32KB Flash (0,5 para bootloader), 2KB RAM y 1KB Eeprom.

 ISSN: 2594-1410

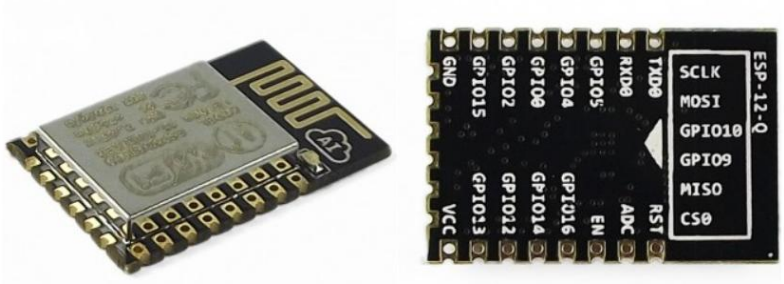
Figura 2. *Representa el Arduino UNO.*



ESP8266

ESP8266 es un SoC (System on Chip), mostrado en la Figura 3, con capacidades de 2.4 GHz Wi-Fi (802.11 b / g / n, soporte WPA / WPA2), 16 GPIO de propósito general (entrada / salida), I²C, convertidor analógico-digital (ADC de 10 bits), SPI, I²S, UART y modulación de ancho de pulso (PWM), emplea un CPU RISC de 32 bits basado en el Tensilica Xtensa LX106 funcionando a 80 MHz (o overclocked a 160 MHz), tiene una memoria ROM de inicio de 64 KB, memoria RAM de instrucciones de 64 KB y 96 KB de RAM de datos, memoria flash externa de 4MB, pero este último varía entre diferentes versiones de módulo.

Figura 3. *Representa el módulo ESP de autoría propia.*



Protoboard mini

Protoboard mini es compatible con Arduino, se muestra en la Figura 4, plástico ABS, apto para cables 21-26 AWG (0,4 - 0,7 mm), número de agujeros: 170 puntos de conexión, columnas: 2 con 17 carriles, filas: 2 con 5 puntos cada una, dimensiones 35 x 47 x 8,5 mm.

Figura 4. *Representa una protoboard mini.*

Buzzer

Buzzer es tipo zumbador, activo, material: PBT, voltaje: 12V, voltaje de trabajo 9V to 15V max, nivel de presión de sonido 85dB at 10cm, temperatura de trabajo -20°C to +70°C, peso: 2 gramos y se muestra en la Figura 5.

Figura 5. *Representa un buzzer.*

Pantalla LCD16x2 Ic2

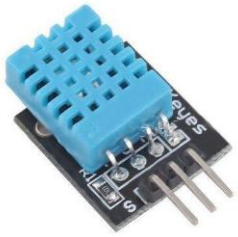
Dimensiones: 80mm x 35mm x 11mm, área visible 64.5mm x 16mm, voltaje de operación: 5V DC, corriente de operación: 90mA ~ 120 mA, pantalla de 2 líneas por 16 caracteres, protocolo I2C (Figura 6).

Figura 6. *Representa una pantalla LCD.*

Sensor de temperatura DHT11

Sensor de temperatura DHT11 V, Figura 7 es un sensor digital de temperatura y humedad relativa cuyas características son voltaje de operación 3V – 5V DC, rango de medición de temperatura 0 a 50 °C, precisión de medición de temperatura ±2.0 °C, resolución temperatura 0.1°C rango de medición de humedad 20% a 90% RH, precisión de medición de humedad 5% RH, resolución humedad 1% RH, tiempo de censado 1 seg.

Figura 7. *Representa un sensor de temperatura.*



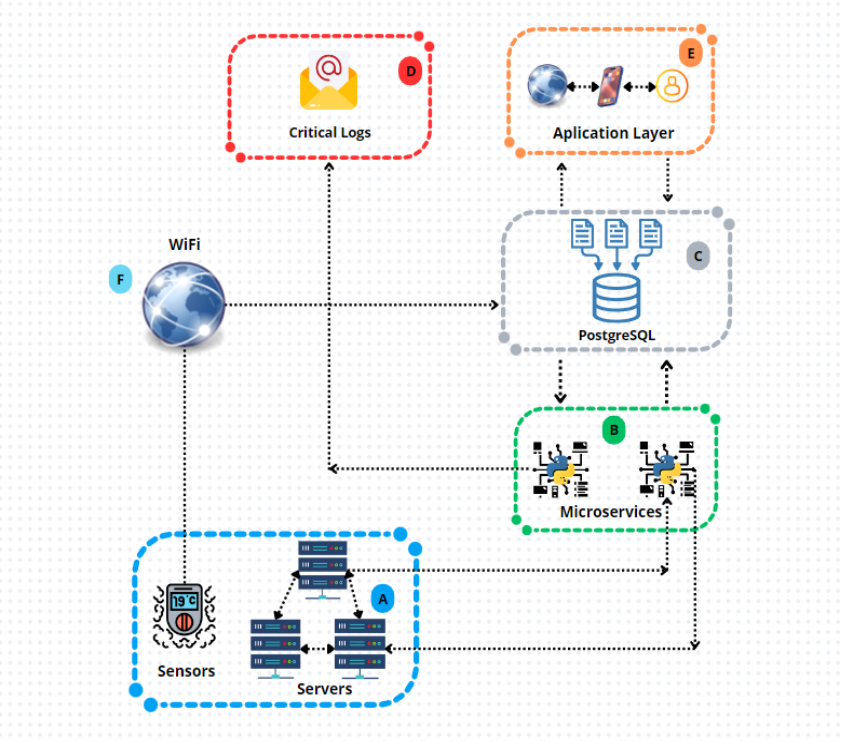
WSN para el monitoreo remoto y aplicación de IoT

Si bien la metodología de “Antonio García Cubas” servirá como base para este trabajo, se realizarán ciertas adaptaciones y se manejará distinto escenario para ajustarla a las particularidades del contexto actual y a los objetivos específicos de este estudio. Por ejemplo, la metodología en la que se basa es Sensor inalámbrico modular multipropósito para monitoreo remoto y aplicaciones

 ISSN: 2594-1410

de IoT y en este trabajo la adaptación es a un escenario distinto en las cuales, estas modificaciones permitirán obtener resultados más precisos y relevantes aplicables a este proyecto, el objetivo de implementar un diseño WSN (Wireless Sensor Networks o Redes Inalámbricas de Sensores) y tecnología IoT en la universidad, es para llevar a cabo el monitoreo constante de los datos proporcionados por los servidores y sensores instalados en su entorno, estos datos van a ser monitoreados desde una aplicación web desarrollada a lo largo de este trabajo, la arquitectura consta de 6 bloques representados en la Figura 8, el bloque A se refiere a los servidores encargados de almacenar la aplicación web que actualmente soporta todas las operaciones administrativas de la institución, los cuales se les monitorea la temperatura que reciben mediante los sensores ya integrados, así como también se monitorea la temperatura mediante un controlador localizado en el mismo bloque A, el bloque B se refiere a los microservicios ejecutados en segundo plano para control de recepción de datos del monitoreo, los cuales serán almacenados en el bloque C que contiene la base de datos para recopilación de los datos, para posteriormente enviarlos al bloque D en caso de haber una alerta se envían los datos críticos mediante correo electrónico, y de manera simultánea de envían los datos recopilados en la base de datos al bloque E, el cual es la interface gráfica en donde se gestionarán todos los datos para la toma de decisiones y poder llevar a cabo el mantenimiento predictivo y evitar daños críticos en los servidores, todo está integrado a una red de internet bloque F, para realizar la gestión de envío de alertas y datos a la aplicación web.

Figura 8. *Representa el diseño IoT.*



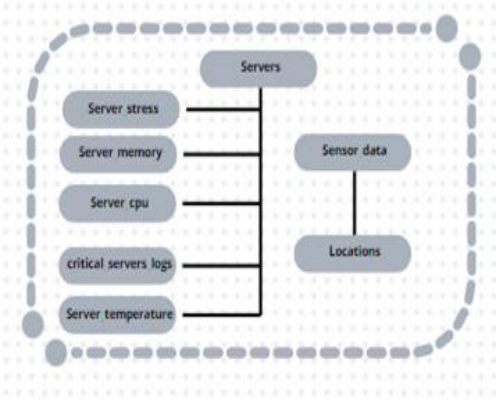
Servidores

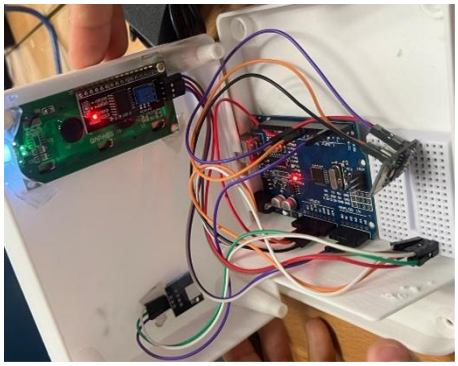
La familia de computadoras Raspberry Pi ha sido ampliamente utilizada por la comunidad académica para el prototipado de sistemas en robótica, visión artificial, teledetección, monitoreo de sensores, Internet de las cosas, entre otros, aprovechando su bajo costo y capacidad de funcionar con sistemas operativos basados en Linux con distribuciones como Raspbian. (Rivera, 2016), e utilizan para alojar la aplicación web, bases de datos, entornos de pruebas/desarrollo y microservicios. Todos los servidores operan sobre un sistema basado en Debian (Ubuntu Server) el servidor de microservicios ejecuta scripts desarrollados en Python, los cuales corren en segundo plano como demonios. El primer script consulta la base de datos para obtener la lista de servidores activos. Una vez obtenida esta información, el microservicio se conecta a cada servidor mediante SSH, recopila los datos necesarios y los utiliza para generar gráficas y actualizar la información en tiempo real en el sistema y base de datos, el segundo microservicio consulta la información registrada por el primer microservicio, para evaluar las condiciones de cada servidor y validar en qué estado se encuentran, si encuentra algún error crítico en algún servidor lo registra en la base de datos y además envía un correo informativo a todos los usuarios privilegiados del sistema.

Integración WSN

La arquitectura del WSN mostrada en la Figura 9 están integrados en una placa de desarrollo ESP para la conexión a la red y enviar los datos a la aplicación web, alimentada a 5V, la cual tiene integrado el sensor de temperatura para monitoreo de la habitación en los cuales enviará las alertas y sonará una alarma si detecta que la temperatura de la habitación sobrepasa los 27°, este se encuentra alimentado a 3.3V, cuyos datos monitoreados son mostrados en la pantalla LCD para poder visualizar la temperatura, cada uno de los dispositivos se encuentran integrados a un Arduino UNO para poder codificar las acciones de medición de temperatura y envío de datos previamente registrados en una base de datos localizada en los servidores Rasperry Pi para posteriormente ser enviados a la interfaz web donde el usuario podrá realizar acciones respecto a las distintas situaciones de monitoreo.

Figura 9. *Representa la estructura de los servidores.*



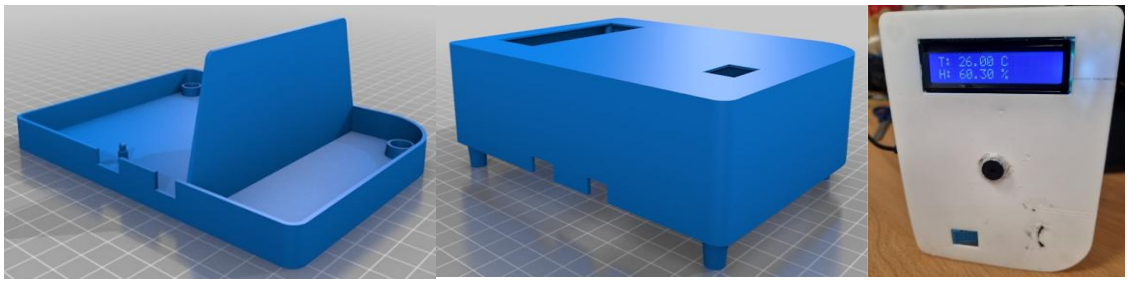


 ISSN: 2594-1410

Diseño de caja 3D

Se diseñó una caja para integrar el circuito y mantenerlo seguro y protegido (Figura 10), dado que lo que interesa es algo accesible y ya que no será una producción en volumen, se buscó el desarrollo integrando herramientas existentes, el diseño fue desarrollado en Flash Print (versión 5.5.1) de uso libre, dicho diseño fue impreso en una impresora 3D creator 3 pro, con un filamento PLA de 1.75mm.

Figura 10. *Representa diseño 3D de la caja contenedora del circuito.*

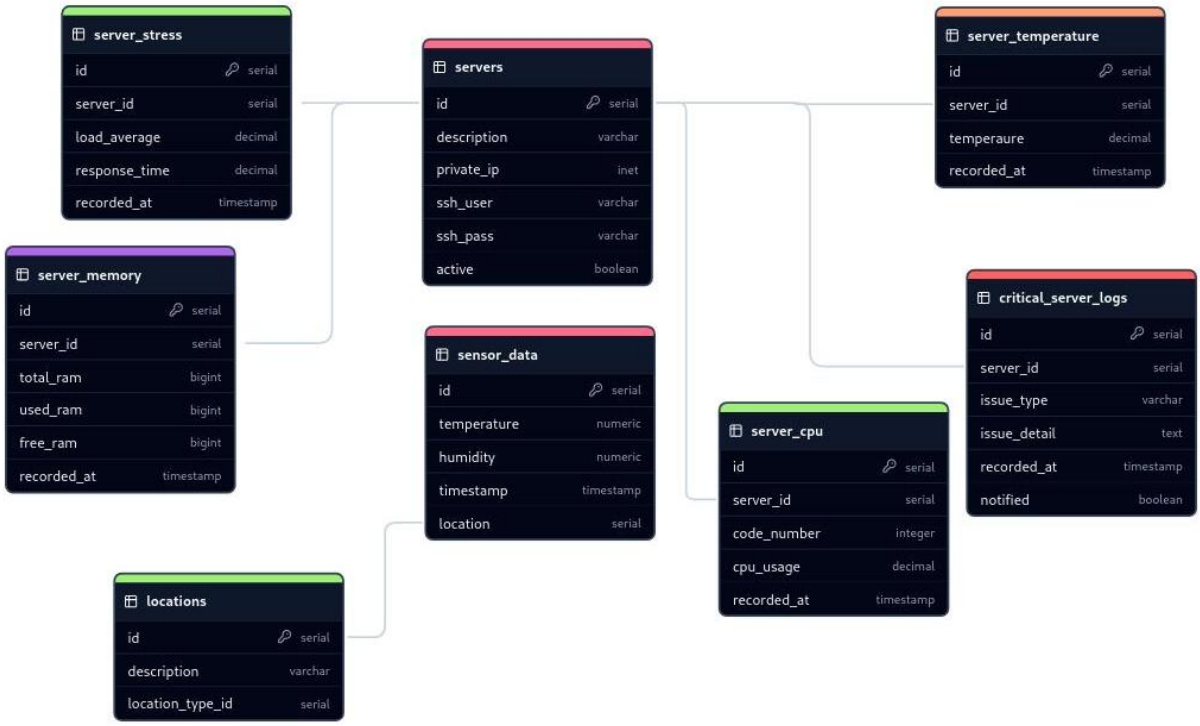


Base de datos

En la integración de este sistema, se desarrolló una base de datos en PostgreSQL versión 14. Esta base de datos está diseñada para almacenar y gestionar toda la información relevante relacionada con el monitoreo y control de servidores (Figura 11). La base de datos está organizada en torno a los servidores y sus diversos parámetros de operación, como temperatura, uso de CPU, memoria, y otros factores críticos que permiten una evaluación exhaustiva del estado de cada máquina. La estructura que conforma la base de datos es la siguiente: Servers: Es la tabla central que almacena información básica sobre los servidores, como su descripción, IP privada, credenciales SSH, y si el servidor está activo o no. Cada servidor tiene un identificador único (id) que se relaciona con las demás tablas para vincular los datos específicos de monitoreo, server_stress: Esta tabla almacena el estado de carga del servidor, con datos como la carga promedio (load_average) y el tiempo de respuesta (response_time), junto con una marca de tiempo (recorded_at) que indica cuándo se tomó la medición, server_memory: Aquí se registran los datos de la memoria de los servidores, como la cantidad total de RAM (total_ram), RAM usada (used_ram), y RAM libre (free_ram). Cada registro también está vinculado a un servidor específico mediante el server_id, server_cpu: Contiene la información sobre el uso del CPU de los servidores, representando el número de código del servidor (code_number) y el porcentaje de uso del CPU (cpu_usage). Estos datos son cruciales para monitorear el rendimiento y prevenir sobrecargas, server_temperature: Almacena los datos de temperatura de cada servidor, esencial para evitar el sobrecalentamiento. Los registros incluyen el valor de la temperatura (temperature) y la marca de tiempo de la medición (recorded_at), sensor_data: Esta tabla recoge información general de los sensores en las ubicaciones de los servidores, tales como la temperatura ambiente (temperature), la humedad (humidity), y la ubicación específica (location). Estos datos ayudan a contextualizar el entorno en el que opera cada servidor, critical_server_logs: Almacena los registros de eventos críticos ocurridos en los servidores, como tipos de problemas (issue_type), detalles del problema (issue_detail), y si se notificó al personal responsable (notified). Locations: Define las diferentes ubicaciones donde se encuentran los servidores, registrando su descripción y tipo de ubicación, este modelo de base de datos permite el monitoreo detallado de múltiples servidores, proporcionando datos en tiempo real que facilitan la toma de decisiones para la gestión eficiente

de la infraestructura tecnológica. La interconexión entre las tablas asegura que cada aspecto del rendimiento del servidor esté documentado y accesible para su análisis y mejora continua. Esté diseño modular y escalable de la base de datos es una pieza clave en la arquitectura del sistema, permitiendo un manejo centralizado y eficiente de los recursos del servidor, así como la identificación temprana de problemas potenciales.

Figura 11. *Representa el diagrama entidad relación de cada una de las tablas de la base de datos.*



Resultados

El sistema “Mantenimiento Predictivo”, es una aplicación que, empleando un sistema de información en tiempo real, recopila información para dar aviso por medio de correo electrónico cuando un equipo comience a fallar o a trabajar de forma inadecuada (Bolaño, 2005).

La implementación de sistemas de mantenimiento predictivo es clave para la evolución hacia la industria 4.0, ya que permite anticiparse a fallos y optimizar el uso de los recursos. A diferencia del mantenimiento reactivo o preventivo, el predictivo se basa en la recopilación de datos en tiempo real a través de sensores y dispositivos conectados, esto no solo reduce el tiempo de inactividad no planificado, sino que también mejora la eficiencia operativa y prolonga la vida útil de los equipos. Además, permite a las organizaciones tomar decisiones informadas, basadas en el estado real de los activos, lo que incrementa la confiabilidad y seguridad en los procesos productivos. El sistema desarrollado en este artículo, consiste en mostrar datos previamente almacenados en el servidor, dicho sistema será encargado de mostrar los datos al usuario, tales como, servidores activos, no activos, críticos y en observación, mostrarán los datos de los servidores Raspberry PI, como la temperatura interna de los servidores, humedad, el estrés de la memoria RAM, y el usuario podrá tomar decisiones referentes a las acciones a implementar en

 ISSN: 2594-1410

dichos dispositivos para prevenir fallos antes de que sucedan y evitar la pérdida de los servicios y la operatividad de dicha institución.

Instalación Raspberry PI4

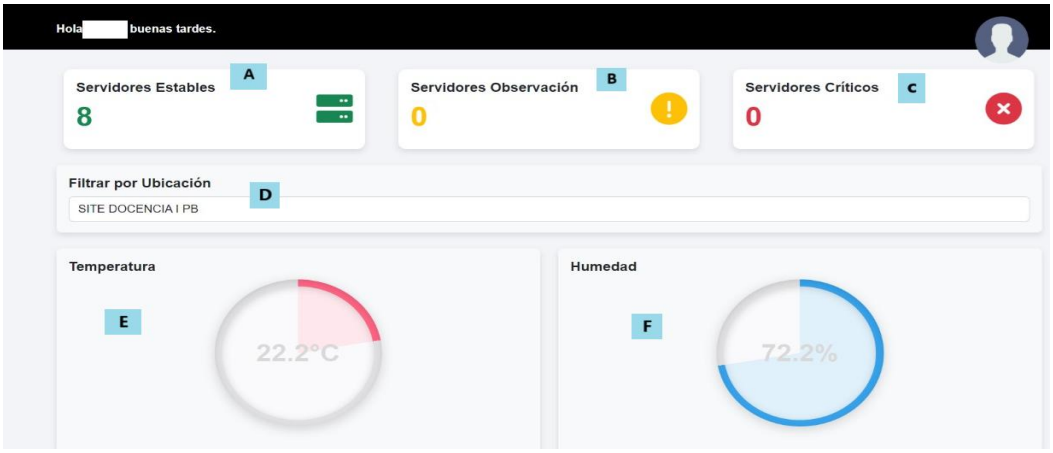
La instalación de los servidores Raspberry PI 4 (Figura 12) y en centro de control de temperatura, fue ejecutada una vez terminado el periodo de pruebas, dichos dispositivos fueron instalados en un site de uno de los edificios de docencia de la institución, los servidores ya se encuentran en modo producción ejecutando todos los servicios y enviando los datos al sistema de monitoreo de los diferentes tipos de variables.

Figura 12. *Representa instalación de servidores Raspberry y centro de control de temperatura.*

Sistema de monitoreo para mantenimiento predictivo

La interfaz de usuario es una aplicación Web, la cual está desarrollada en PHP y JavaScript, esta es la encargada de mostrar diferentes tipos de variables se muestra en la Figura 13, bloque A muestra todos los servidores activos y estables, estos son servidores que están funcionando de manera óptima, es decir, que cumplen con los siguientes criterios: Uso de CPU menos del 50%, uso de memoria menos del 70% de la memoria total, temperatura menos de 60°C, bloque B, muestra los servidores en observación, estos servidores están en una condición intermedia y deben ser monitoreados, ya que presentan indicadores que podrían empeorar, los criterios para que sea considerado en observación son: Uso de CPU: Entre 50% y 80%, uso de Memoria, entre 70% y 90% y una temperatura: Entre 60°C y 75°C, Bloque C muestra los servidores críticos estos deben de estar en un estado que requiere atención inmediata, estos servidores tienen al menos una métrica en niveles peligrosos, lo que podría indicar un fallo inminente, los criterios a considerados son: Uso de CPU más del 80%, uso de memoria más del 90% de la memoria total y una temperatura más de 75°C, bloque D, muestra la ubicación de los servidores y permite la selección de la misma, bloque E muestra en tiempo real los datos de la temperatura de los servidores, bloque F muestra los datos de humedad de los servidores.

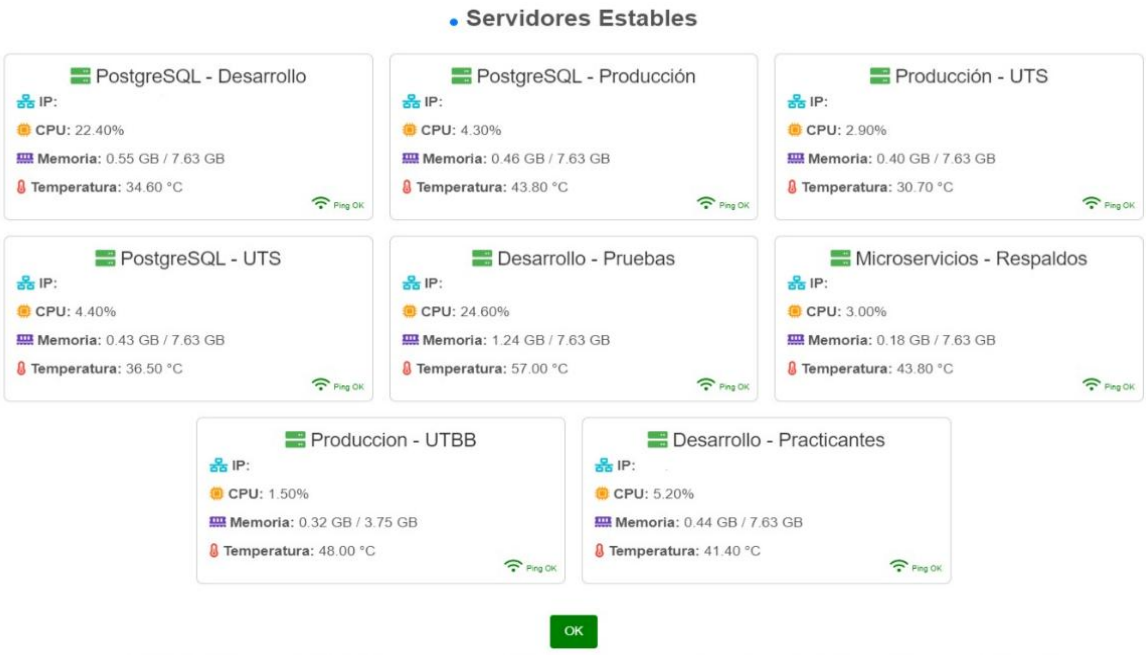
Figura 13. *Representa la interface de usuario del sistema.*



Servicios

Los servidores estables, en observación y críticos muestran las mismas variables con los datos correspondientes en cada uno de ellos según su estatus, a la IP, la cual fue borrada de la imagen para protección de los datos y seguridad de estos, CPU, memoria, y temperatura, como se muestra en la Figura 14.

Figura 14. *Representa la visualización de los servicios estables.*



 ISSN: 2594-1410

Gráficas de datos

Se muestran gráficas como en la Figura 15 con las variables enviadas por el servidor.

Figura 15. *Representa la visualización de los datos graficados.*

Envío de alertas

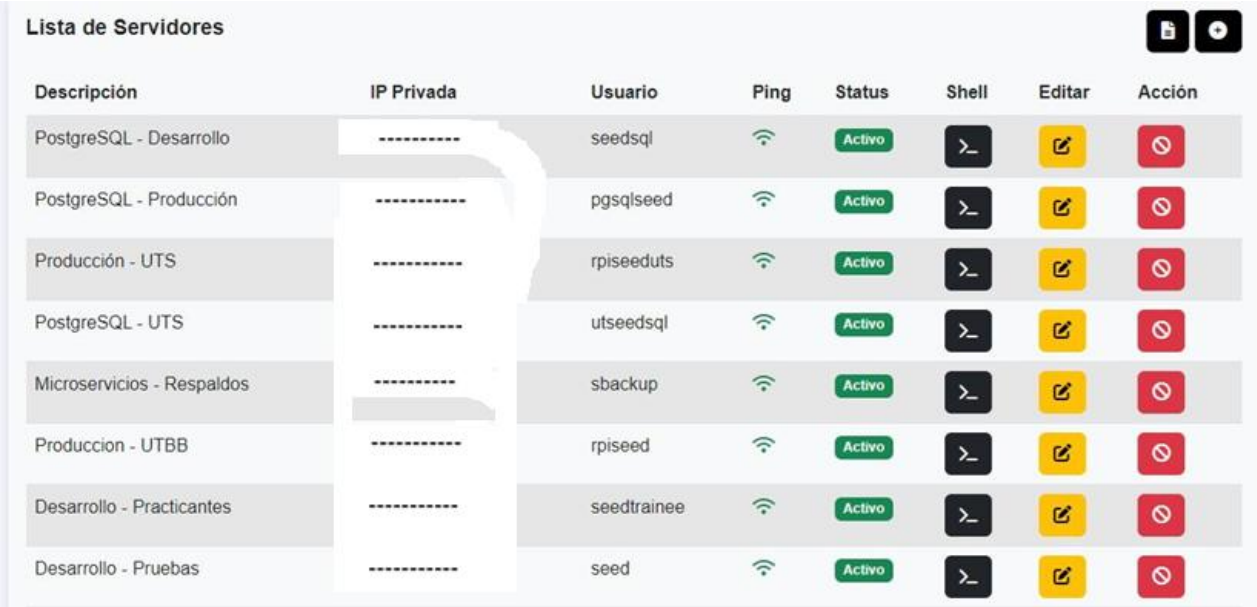
Las alertas son enviadas como se muestra en la Figura 16 solamente si un servidor se encuentra en estado crítico o de observación, dichas alertas son enviadas al correo del área encargada para tomar acciones inmediatas desde la interfaz de usuario del sistema de monitoreo para mantenimiento predictivo.

Figura 16. *Representa las alertas enviadas al correo electrónico.*

Datos de servidores

En la lista de los servidores se pueden realizar distintas acciones como muestra en la Figura 17, se puede observar la descripción del servidor, la IP (actualmente oculta para la protección de los datos), nombre de usuario, el ping, el estatus, la terminal para tomar acciones de manera inmediata, editar, el cual permite editar todos los parámetros previamente mencionados, incluyendo la contraseña, y por último se encuentra el de acción el cual permite inactivar el servidor de manera inmediata.

Figura 17. *Representa los datos de los servidores y las acciones a realizar.*



Discusión

Los resultados obtenidos en esta investigación sobre Mantenimiento Predictivo para Gestión de Servidores Utilizando Tecnología IoT revelan hallazgos significativos que tanto confirman como desafían las teorías existentes. Por un lado, dada la hipótesis del sobrecalentamiento de los cuatro servidores Raspberry Pi 4 fue provocado por una combinación de factores, incluyendo una inadecuada disipación de calor, una carga de trabajo excesiva y posiblemente fallas en los sistemas de enfriamiento, el sobrecalentamiento prolongado condujo a un fallo catastrófico en los componentes electrónicos de los servidores, resultando en daños irreparables y la necesidad de reemplazar completamente las unidades, Es importante destacar que el costo exacto de reparar cuatro servidores Raspberry Pi 4 dependerá de varios factores, como modelo específico, componentes dañados, mano de obra y partes de repuesto, un escenario seria el reemplazo total de los servidores, los rangos de precio podrían ubicarse entre $2,500.00 o $3,000.00 por cada uno de ellos, dando un total de $10,000.00 o $12,000.00 por adquirir los 4 servidores. Sin embargo, estos hallazgos sugieren que al contar con este sistema de mantenimiento predictivo se pueden prevenir fallas como el sobrecalentamiento en servidores Raspberry Pi 4, detectar anomalías y tendencias antes de que se conviertan en problemas graves.

Los resultados obtenidos respaldan de manera contundente la hipótesis inicial. Los datos muestran una clara relación entre reducción de costos y prevención de daños críticos, tal como se predijo. De igual manera al detener los servidores antes de que sufran un fallos catastrófico, se reduce

 ISSN: 2594-1410

significativamente el riesgo de pérdida de datos, esto es especialmente importante si no cuenta con copia de seguridad actualizadas, otro de los resultados obtenidos fue dado el acontecimiento ocurrido el 27 de septiembre de 2024, “Un apagón masivo sorprendió a los habitantes de Puerto Vallarta, Bahía de Banderas y Compostela la noche de este jueves, cuando a las 7:26 pm se interrumpió el suministro eléctrico en toda la región. La situación, que también afectó la señal de celulares e Internet, se prolongó hasta las 7:30 pm, causando alarma entre la ciudadanía”. (Baumgarten, 2024), al tener implementado dicho sistema se pudo recibir una aleta mediante correo electrónico en estatus critico indicando una pérdida de datos en los servidores, indicando que la situación sea revisada y se realice el encendido de los servidores, en otras circunstancias, en las cuales no se cuenta con un sistema de monitoreo, no hubiera sido posible saber que los servidores fueron inactivados por la pérdida de suministro de energía. Estos hallazgos fortalecen la evidencia a favor de la teoría del mantenimiento predictivo para gestión de servidores utilizando tecnología IoT, y mostrando resultados favorables dada su implementación. Cabe recalcar que dicha implementación es reciente entrando en modo producción el lunes 23 de septiembre 2024, por lo tanto, no hay un historial abastecido de datos para que se pueda llevar a cabo un análisis estadístico y poder sugerir propuestas de mejora, sin embargo, los datos que son almacenados en la base de datos serán sometidos a análisis para futuras mejoras y análisis para próximos artículos de investigación.

Referencias

Arzate-Rivas O, Sámano-Ortega V, Martínez-Nolasco J, Santoyo-Mora M, Martínez-Nolasco C, De León-Lomelí R. Sistema de gestión de energía IoT basado en una red inalámbrica de sensores/actuadores. Tecnologías. 2024; 12(9):140. https://doi.org/10.3390/technologies12090140

Durani, H., Sheth, M., Vaghasia, M. y Kotech, S. (abril de 2018). Aplicación de automatización inteligente para el hogar mediante IoT con la aplicación Blynk. Segunda conferencia internacional sobre tecnologías de computación y comunicación inventivas (ICICCT) de 2018 (págs. 393-397). IEEE.

Gulati, K., Boddu, RSK, Kapila, D., Bangare, SL, Chandnani, N. y Saravanan, G. (2022). Un artículo de revisión sobre técnicas de redes de sensores inalámbricos en Internet de las cosas (IoT). Materials Today: Proceedings, 51, 161-165.

Jimena-Baumgarten. (27 de Septiembre de 2024). Tribuna de la bahía. Recuperado el 2024 de Septiembre de 2024, de https://tribunadelabahia.com.mx/apagon-puerto-vallarta-por-que-se-fue-luz/

Mora-Andrade, S. E. (2014). Implementación de un servidor web cache bajo la plataforma GNU/LINUX, automatizando scritps para óptiminizar el mantenimiento del servidor.

Mudaliar-Mani Dheeraj."Sistema de monitoreo de energía en tiempo real basado en IoT que utiliza Raspberry Pi". *Internet of Things* 12 (2020): 100292.

Panigrahi, C. R., Sarkar, J. L., Pati, B., Buyya, R., Mohapatra, P., & Majumder, A. (2021). Mobile Cloud Computing and Wireless Sensor Networks: A review, integration architecture, and future directions. Iet Networks, 10(4), 141-161.

Pimentel-Victoria y Bradford G. Nickerson. "Comunicación y visualización de datos en tiempo real con WebSocket". *IEEE Internet Computing* 16.4 (2012): 45-53. De Volder, K. (2006, January). JQuery: A generic code browser with declarative configuration language. In International Symposium on Practical Aspects of Declarative Languages (pp. 88-102). Springer Berlin Heidelberg.

Quiñonez-Yadira, Carmen-Lizarraga, Juan-Peraza & Oscar-Zatarain1. (2019). Sistema inteligente para el monitoreo automatizado del transporte público en tiempo real. Revista Ibérica de Sistemas e Tecnologias de Informação, (31), 94-105.

Rivera, Andrés Felipe Gómez, and Fabián Velásquez Clavijo. "Monitoreo de sensores en aplicaciones web embebidas." *International Congress of Basic Sciences and Engineering CICI*. No. 105. 2016.

Sámano-Ortega V, Arzate-Rivas O, Martínez-Nolasco J, Aguilera-Álvarez J, Martínez-Nolasco C, Santoyo-Mora M. Sensor inalámbrico modular multipropósito para monitoreo remoto y aplicaciones de IoT. Sensores. 2024; 24(4):1277. https://doi.org/10.3390/s24041277

Shelby-Zach. (2010). Embedded web services. IEEE Wireless Communications, 17(6), 52-57.