

# Aplicación web adaptativa para la monitorización y graficación de datos en tiempo real



## Colaboración

Erica María Lara Muñoz; Israel Viveros Torres,  
Tecnológico Nacional de México / Instituto Tecnológico Superior de Alvarado

Fecha de recepción: 07 de mayo de 2024

Fecha de aceptación: 22 de noviembre de 2024

**RESUMEN:** El presente artículo tiene como objetivo mostrar el desarrollo de una aplicación web adaptativa, realizada con la finalidad de interpretar y graficar datos en tiempo real de un prototipo mecánico rotativo. Esto se logró a través de la creación de una base de datos que almacena los registros de las vibraciones generadas por el prototipo, la creación de una API para establecer la comunicación entre la base de datos y la aplicación web, y la conexión de la API con un microcontrolador Atmel 328 para la transmisión de los datos en tiempo real. Los resultados obtenidos fue una aplicación web interactiva, responsiva y accesible que permite a los usuarios visualizar estadísticas, gráficas y datos recopilados en tiempo real de un banco de vibraciones mecánicas, lo que significa que se puede obtener un monitoreo continuo del comportamiento del prototipo, un análisis predictivo para llevar el prototipo mecánico a su máximo potencial y conocer cuando realizar un mantenimiento preventivo o correctivo en caso de ser necesario.

**PALABRAS CLAVE:** Análisis predictivo, API, aplicación web adaptativa, Arduino, graficar datos, monitorización, prototipo mecánico, tiempo real.

**ABSTRACT:** The objective of this paper is to show the development of an adaptive web application, with the purpose of interpreting and graphing data in real time of a rotating mechanical prototype. This was achieved through the creation of a database that stores the vibrations records generated by the prototype, creating an API to establish communication between the database and the web application and connecting the API with an Atmel 328 microcontroller for real time data transmission. The results were an interactive, responsive and accessible web application that allows users to visualize statistics, graphs and data collected in real time from the mechanical vibration bank, which means that you can obtain a continuous monitoring of the behavior of the prototype, a predictive analysis to bring the mechanical prototype to its full potential and know when to perform preventive or corrective maintenance if necessary.

**KEYWORDS:** Predictive analysis, API, responsive web application, Arduino, graph data, monitoring, mechanical prototype, real time.

## INTRODUCCIÓN

Los avances tecnológicos, han propiciado que las personas cada vez se hagan más dependientes de las tecnologías, éstas se vuelven más indispensables en cualquier ámbito de la vida, ya sea para uso personal [1], educativo [2][3], profesional [4] o empresarial [5][6]. Dentro de la industria, éstas pueden automatizar procesos [7], mejorar la interacción, apoyar en la comunicación y la colaboración entre trabajadores, pero, sobre todo, se puede tener mayor impacto, al utilizar las tecnologías para desarrollar aplicaciones o sistemas, que permitan analizar grandes cantida-

des de información, datos para la toma de decisiones y mejorar incluso, los procesos de una organización o industria y el rendimiento de maquinaria y equipos.

Sin embargo, dentro de la industria aún hay cosas por hacer. Un aspecto muy común que ocurre es el desbalance en los sistemas de tipo rotatorio, esto es, una afectación muy recurrente en el sector industrial que genera índices muy altos de vibraciones mecánicas en las máquinas y sistemas de potencia, por lo que es muy importante realizar un diagnóstico adecuado y su corrección cuando las vibraciones se detecten para evitar una falla grave [8].

Debido a este fenómeno que ocurre en la industria, en los últimos tiempos, se ha observado el interés en utilizar técnicas de mantenimiento predictivo para el cuidado de maquinaria y equipos, con la finalidad de que éstas operen de forma más eficiente, y se disminuyan fallas que eviten tiempos muertos en la industria [9] o fallas, incluso fatales [8].

Este mantenimiento predice cuando una maquinaria o equipo, requiere de una reparación o cambio de pieza. Para ello, dentro de la industria, se utiliza el monitoreo y análisis de vibraciones, con la finalidad de conocer el estado de la salud mecánica de la maquinaria y prevenir algún fallo antes de que se origine [10], ya que con el análisis de vibraciones se puede anticipar a un mal funcionamiento, extendiendo de esta manera la vida útil y productividad de la maquinaria o equipo.

La industria 4.0 definida como la integración de maquinaria y dispositivos físicos complejos con sensores y software en red, utilizados para predecir, controlar y planificar mejores resultados comerciales y sociales [11], apoya el monitoreo y análisis de vibraciones, debido a que permite entre otras cosas, la comunicación en tiempo real entre dispositivos, sistemas y usuarios haciendo uso de tecnologías y servicios de internet [12].

Por estas razones, la importancia de realizar sistemas o aplicaciones que impulsen esta industria inteligente y permitan eficientar los procesos y la toma de decisiones dentro de la empresa e incluso prevenir alguna falla en la maquinaria o equipo utilizado.

Por lo que a continuación, se muestra un sistema que permite monitorear datos en tiempo real de un banco de vibraciones mecánicas, con la finalidad de garantizar que opere eficientemente, minimice sus tiempos de paro y se le pueda realizar un mantenimiento preventivo o correctivo de manera oportuna.

## MATERIAL Y MÉTODOS

### Recursos materiales

Para la elaboración de la aplicación web, se hizo uso de hardware, software e infraestructura tecnológica, las cuales, en su conjunto, se combinan de manera sinérgica permitiendo la creación de la aplicación dinámica para la

interpretación y graficación de datos en tiempo real para un banco de vibraciones mecánicas. Cada uno de estos recursos, fue seleccionado de manera cuidadosa para garantizar un óptimo rendimiento y la confiabilidad del sistema.

Para el caso del hardware se utilizó un prototipo de banco de vibraciones, un sensor MPU6050, un ESP32 y una computadora de alto rendimiento.

Prototipo de banco de vibraciones. Construido en aluminio anodizado, material aplicable por su rigidez y ligereza apropiada para este tipo de aplicaciones, aportando así también movilidad en el dispositivo. Este sistema está dotado como unidad de potencia motriz, un motor de corriente alterna, símil a un equipo industrial. El prototipo [8] busca emular una maquinaria rotatoria con elementos de fijación por medio de chumaceras en dos puntos específicos del eje, a fin de evaluar las vibraciones en dos planos. El dispositivo antes descrito se utiliza como el componente físico central y generador de las señales dinámicas que serán adquiridas y mostradas por la aplicación en tiempo real, así como implementar la simulación de un desbalanceo dinámico para la generación de datos anómalos de funcionamiento, proporcionando una base realista para la evaluación y monitoreo.

Sensor MPU6050. Este sensor [8], aplicado como un transductor realiza la conversión de parámetros físicos a señales eléctricas normalizadas de 5VCC, este elemento es estrictamente un acelerómetro de tipo análogo de simple implementación para la medición de vibraciones, desempeña un papel fundamental al capturar datos precisos para su posterior procesamiento y análisis.

ESP32. Es un protocolo de comunicación altamente utilizado como estándar de comunicación para microcontroladores, el cual resulta en desempeño altamente eficiente para esta aplicación por su transmisión de datos en "paquetes" de bytes, que permite un manejo dinámico y robusto de la información, sin demandar altos costos de procesamiento de capacidad de cómputo por parte del microcontrolador y por ende de la aplicación en sí misma. Este protocolo resulta más que adecuado para la transmisión en tiempo real de datos de vibración a través de la API, lo anterior con base en las consideraciones técnicas antes citadas. El ESP32 asegura la conectividad eficiente y confiable entre el hardware físico y la infraestructura digital [13].

Computadora de alto rendimiento. Una computadora equipada con un procesador i5 de octava generación, 32 GB de RAM y 1 TB de SSD con la finalidad de que proporcione el entorno de desarrollo robusto necesario para ejecutar las aplicaciones y procesos intensivos de software.

El software utilizado fue React, Node.js, PostgreSQL y Socket.io, las razones por las que se utilizaron estas interfaces son las siguientes.

React. Es una biblioteca de Javascript de código abierto [14], que fue diseñada para generar interfaces de usuario con la finalidad de que el desarrollo de las aplicaciones sea más fácil. Fue utilizado en el desarrollo del frontend de la aplicación web, ya que ofrece una estructura modular y eficiente para crear interfaces de usuario interactivas y receptivas.

Node.js. Es un entorno de código abierto multiplataforma de JavaScript en tiempo de ejecución del lado del servidor, esencial para la implementación de la API, ya que es basada en eventos, permitiendo una gestión eficiente de las operaciones del servidor y facilitando la comunicación con la base de datos [15].

PostgreSQL. Es el sistema de código abierto y orientado a objetos de gestión de base de datos relacional que garantiza la integridad y consistencia de los datos, es fundamental para el almacenamiento y recuperación eficiente de la información generada por el banco de vibraciones [16].

Socket.io. Utilizado para la integración de WebSockets, facilitando la transmisión en tiempo real de datos entre el ESP32 y la aplicación web, asegurando de esta manera, una actualización instantánea de la información [15].

La infraestructura tecnológica utilizada fue un servidor VPS y un dominio.

Servidor VPS. Es un servidor privado virtual que se implementa para alojar la API, aprovechando la capacidad de ejecutar Node.js y garantizando un acceso constante a los servicios ofrecidos por la aplicación.

Dominio. Sistema de nombre de dominio, utilizado para la identificación clara y accesible de la aplicación web, proporcionando una interfaz intuitiva para los usuarios finales.

**Metodología**

La aplicación web se realizó siguiendo una serie de iteraciones definidas que permitieron una gestión flexible y eficiente en el desarrollo de la misma. La primera iteración consistió en el análisis de los requerimientos y la planificación inicial, posteriormente se hace la configuración del entorno y el prototipado, como tercera iteración se desarrolla la base de datos y se crea la API, en la iteración cuatro se realiza la integración del hardware y la configuración del WebSockets, la iteración 5 realiza la validación de notificaciones y funcionalidades, en la iteración 6 se realizó el desarrollo de la página web y la validación del diseño responsivo, finalmente para la iteración 7, se hacen las pruebas finales y la preparación para la implementación. Estas iteraciones las puede observar en la Figura 1. Para la parte de adaptabilidad, se hace uso del React.js ya que proporciona bibliotecas y componentes diseñados para sitios web, mismos que permiten la adaptación de los diversos dispositivos móviles, pudiendo visualizar correctamente en pantallas de todos los tamaños. Un com-

ponente responsivo utilizado fue Tailwind CSS para darle los estilos al administrador, este es un framework de CSS de código abierto que permite crear diseños adaptables y personalizados para sitios web.



Figura 1. Iteraciones para el desarrollo de la aplicación web adaptativa.

Fuente: Elaboración propia

Análisis de requerimientos y planificación inicial. Se realiza el análisis detallado de los requerimientos y la planificación inicial de las actividades, definiendo roles y actividades.

Configuración del entorno y prototipado. Se configuran los entornos React, Node.js y PostgreSQL quedando listos para la implementación (ver Figura 2), y se desarrolla el prototipo de interfaz de usuario válido.

```

BACK_NODE > src > JS app.js > ...
You, hace 2 meses | 1 author (You)
1 // app.js You, hace 2 meses + base del proyecto
2 const express = require("express");
3 const cors = require("cors");
4 const app = express();
5 const swaggerUI = require("swagger-ui-express");
6 const openApiConfiguration = require("./docs/swagger");
7 const morganBody = require("morgan-body");
8 const customHeader = require("./middleware/customHeader");
9 const { dbConnectPostgresql } = require("./database/postgres");
10 const path = require('path');
11
12 if (process.env.NODE_ENV === 'production') {
13   require('dotenv').config({ path: 'production.env' });
14 } else {
15   require('dotenv').config({ path: '.env.dev' });
16 }
17
18 app.use("/api/storage/users", express.static(path.join(__dirname, 'storage/users')));
19 app.use("/api/storage/logos", express.static(path.join(__dirname, 'storage/logos')));
20 app.use(cors());
21 app.use(express.json());
22 dbConnectPostgresql();
23 morganBody(app, {
24   noColors: true,

```

Figura 2. Entorno de desarrollo.

Fuente: Elaboración propia.

Desarrollo de la base de datos y creación de la API. La base de datos se desarrolla haciendo uso de PostgreSQL y se crea la API utilizando Node.js para la comunicación con la base de datos.

Integración del hardware y configuración del WebSocket. Se integra el ESP32 para la transmisión de datos en tiempo real y se configura el WebSocket para la comunicación bidireccional.

Validación de notificación y funcionalidades. Se implementa el envío de notificaciones y se validan haciendo pruebas de funcionalidades clave en la base de datos. Figura 3.

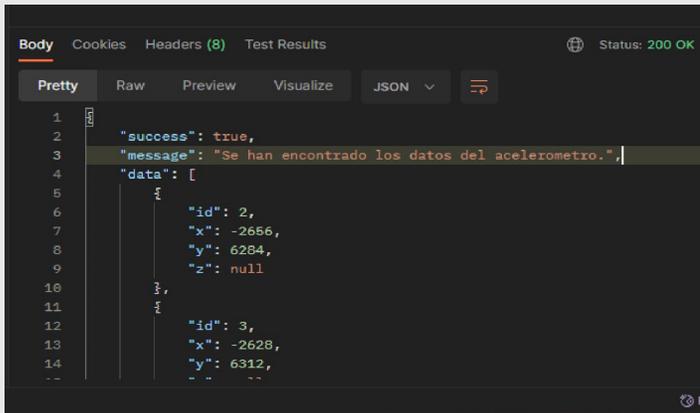


Figura 3. Validación de datos recibidos en la base de datos. Fuente: Elaboración propia.

Desarrollo de página web y validación del sistema responsivo. Se desarrolla la interfaz de usuario utilizando React y se valida el diseño responsivo en diferentes dispositivos y pantallas. Esto se puede observar en las Figuras 4 y 5.



Figura 4. Dashboards de la aplicación web. Fuente: Elaboración propia.

Pruebas finales y preparación para la implementación. Se realizaron las pruebas exhaustivas de todas las funcionalidades y la preparación de la aplicación e infraestructura para la implementación.



Figura 5. Vista de dashboard de dispositivo móvil. Fuente: Elaboración propia.

## RESULTADOS

Ejecutando todas las iteraciones para el desarrollo de la aplicación web adaptativa, se obtuvo una aplicación completamente funcional para la interpretación y graficación de datos en tiempo real de un banco de vibraciones mecánicas, la implementación exitosa de cada iteración garantiza la eficiencia operativa, la monitorización constante y la capacidad de respuesta a cambios en la infraestructura.

La interfaz de la aplicación web muestra de inicio, una pantalla para que el usuario abra su sesión, la cual proporciona un acceso seguro a la aplicación, en donde los usuarios pueden ingresar sus credenciales para autenticarse. La pantalla de inicio está diseñada con una disposición intuitiva, facilitando una experiencia de usuario amigable al guiar al usuario a través de este proceso de autenticación de manera clara y efectiva. La elección de elementos visuales y su disposición estratégica contribuye a la usabilidad de la aplicación web, asegurando una interacción inicial sin contratiempos para los usuarios. Esto se puede observar en la Figura 6.

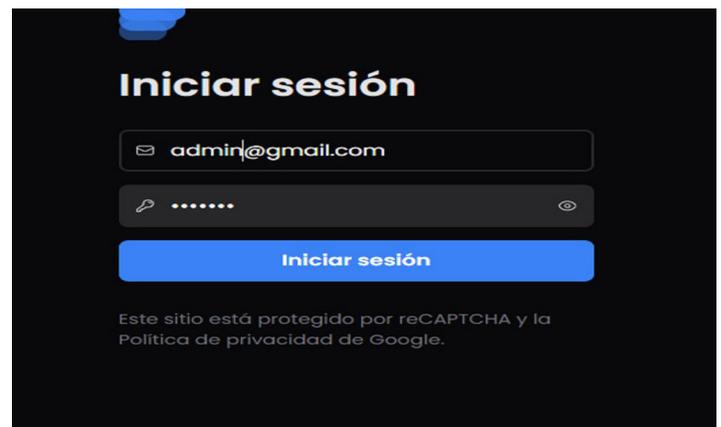


Figura 6. Vista del login. Fuente: Elaboración propia.

Como se observa en la Figura 7, con la finalidad de que el usuario personalice su experiencia y agregue un toque estético y albergue opciones esenciales relacionadas con su perfil de usuario, la aplicación web permite ajustar el modo de visualización y el tamaño de la letra, asegurando de esta manera, una adaptabilidad óptima a las preferencias individuales de los usuarios. Esta funcionalidad amplía la accesibilidad de la aplicación web, además de reflejar un enfoque centrado en el usuario, que proporciona herramientas para personalizar la interfaz de acuerdo con sus preferencias. La inclusión del pie de página del navegador ofrece una navegación conveniente que permite al usuario ajustar su configuración y explorar su perfil sin interrumpir la visualización principal de los datos que se estén analizando de las vibraciones mecánicas del equipo rotatorio.

sistema, como nombres, roles y detalles de contacto. Su diseño permite una fácil búsqueda y clasificación, proporcionando a los administradores un acceso rápido a la información necesaria para la gestión de usuarios. Así mismo, tener la facilidad de crear más usuarios para que puedan acceder al panel con diferentes permisos.

La advertencia relacionada con eventos de inicio de sesión se realiza a través de un correo electrónico, el cual, es una capa adicional de seguridad, notificando a los usuarios sobre cualquier actividad sospechosa en sus cuentas. La información detallada en el correo electrónico proporciona pistas sobre la ubicación, hora y dispositivo asociado, permitiendo una rápida identificación y respuesta a posibles amenazas de seguridad en el acceso a la aplicación.

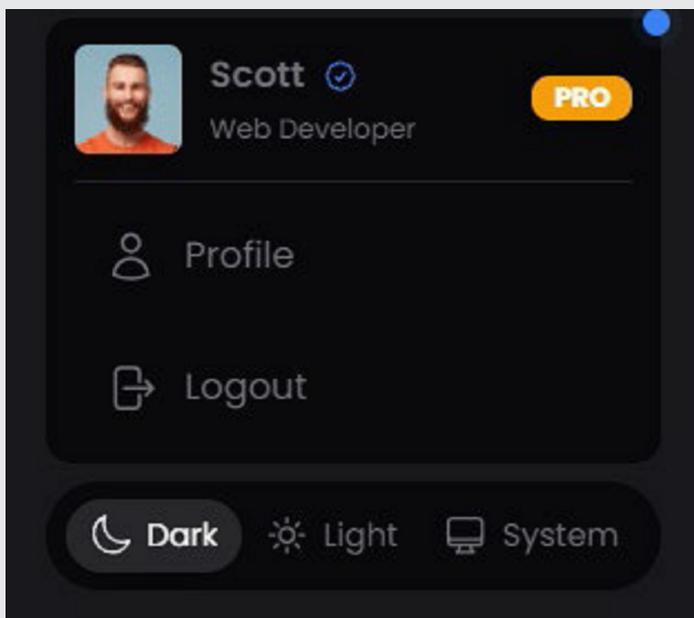


Figura 7. Pie de página del navegador.  
Fuente: Elaboración propia.

ID	x	y	z
2	-2656	6284	
3	-2628	6312	
4	-2600	6276	
5	-2600	6276	
6	-2668	6312	

Figura 8. Nivelación de actividades.  
Fuente: Elaboración propia.

La Figura 8 muestra el módulo del acelerómetro, donde se observa una Tabla detallada del historial del acelerómetro. Esta Tabla integral para el monitoreo, presenta de manera ordenada y cronológica las lecturas del acelerómetro. Cada entrada en la Tabla representa un punto temporal, proporcionando a los usuarios una referencia clara de cómo las vibraciones mecánicas han evolucionado. Además, el diseño de la Tabla facilita la identificación de los valores de las coordenadas y eventos destacados en el historial, contribuyendo así, a un análisis más profundo.

En la Figura 9 se puede observar la consola en serie del entorno de desarrollo Arduino IDE, proporcionando una visión detallada de la conexión del ESP32. Esta herramienta es fundamental para el desarrollo y la depuración, mostrando mensajes y detalles importantes sobre la configuración y funcionamiento del microcontrolador durante la conexión, de esta manera, tener la certeza, que todo está funcionando correctamente en la conexión de los componentes físicos con la API y el Socket, para poder visualizar la información en tiempo real.

Además de la Tabla del historial del acelerómetro, se tiene una Tabla que muestra a los usuarios del sistema, esta Tabla sirve como una herramienta central para la administración eficiente de cuentas. La Tabla presenta información clave sobre los usuarios registrados en el

La notificación de advertencia relacionada con la actividad del acelerómetro, se realiza a través de una notificación de correo electrónico. Este mecanismo de notificación es esencial para alertar a los usuarios o administradores sobre posibles problemas o eventos inesperados relacionados con las vibraciones mecánicas. El contenido del correo electrónico proporciona detalles contextuales, permitiendo una respuesta rápida y adecuada a situaciones críticas.

```

sketch_novo3a.ino
71  pinMode(LED_BUILTIN, OUTPUT);
72  digitalWrite(LED_BUILTIN, LOW);
73  sensor.initialize();
74
75  // Verificar si el sensor MPU6050 se ha iniciado correctamente
76  if (Sensor.testConnection()) Serial.println("Sensor iniciado correctamente");
77  else Serial.println("Error al iniciar el sensor");
78
Monitor Serie x
Mensaje (fíjate para mandar el mensaje de 'ESP32 Dev Modul' a 'COM5')
-----Conectado a WiFi
La IP asignada es: 192.168.3.144
Conexión exitosa con MPU-6050
Connecting to Socket.io...
[Socket] add packet 42 {"client:accelerometerdata", {"axg":-0.075683594,"ayg":-0.486083984,"x":-1240,"y":-7964}}
Respuesta de la API: {"success":true,"message":"Se ha guardado el dato del acelerometro.", "data":{"id":7906,"x":-1240,"y":-7964,"z":null}}
[Socket] disconnected
Acelerómetro - X: -1324g | Y: -7932g
[Socket] add packet 42 {"client:accelerometerdata", {"axg":-0.08010547,"ayg":-0.484130859,"x":-1324,"y":-7932}}
Respuesta de la API: {"success":true,"message":"Se ha guardado el dato del acelerometro.", "data":{"id":7907,"x":-1324,"y":-7932,"z":null}}
[Socket] disconnected
Acelerómetro - X: -1220g | Y: -8012g
[Socket] add packet 42 {"client:accelerometerdata", {"axg":-0.074951172,"ayg":-0.489018972,"x":-1220,"y":-8012}}
Respuesta de la API: {"success":true,"message":"Se ha guardado el dato del acelerometro.", "data":{"id":7908,"x":-1220,"y":-8012,"z":null}}
[Socket] disconnected

```

Figura 9. Consola en serie de la conexión del ESP32 en Arduino IDE. Fuente: Elaboración propia.

Por otro lado, el sistema de monitoreo y graficación de los datos, se puede observar en la Figura 10, en la cual se muestra tanto el total de registros, como los valores en las coordenadas X e Y, en los valores mínimos, máximos y medias, así como el acelerómetro en tiempo real y las coordenadas del RMS en X e Y con la finalidad de monitorear las vibraciones mecánicas de manera eficiente y clara.

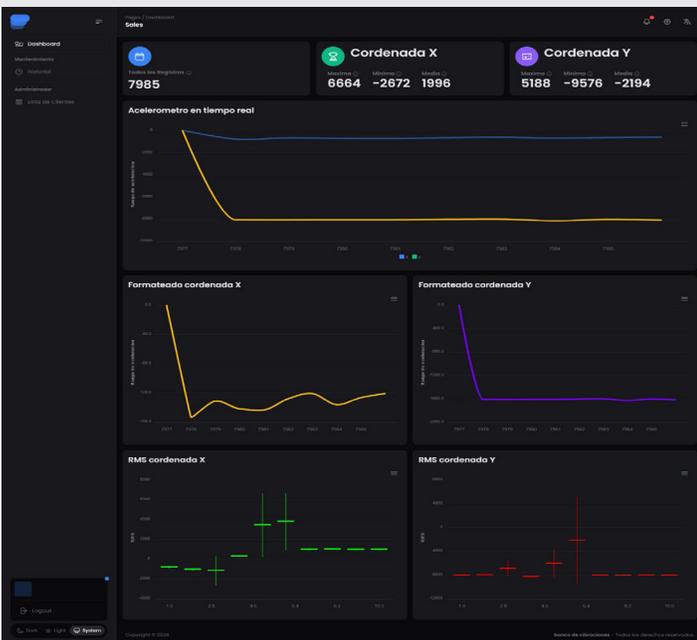


Figura 10. Monitoreo y graficación de datos. Fuente: Elaboración propia.

## CONCLUSIONES

El desarrollo e implementación de la aplicación web adaptativa para la monitorización, interpretación y graficación de datos en tiempo real de un banco de vibraciones mecánicas, ha tenido un proceso metodológico que permitió asegurar la calidad de la aplicación. Durante el desarrollo del proyecto, se buscó abordar la necesidad de monitorear de manera constante el comportamiento de sistemas mecánicos, destacando la importancia de la interpretación eficiente de datos vibratorios.

La aplicación web, construida sobre tecnologías modernas como React para el frontend, Node.js para la API, PostgreSQL para la base de datos, y Socket.io para la comunicación en tiempo real, ha sido un reto muy importante en este proyecto ya que son tecnologías relativamente nuevas en la industria, sin embargo, se observa que son herramientas poderosas y adaptables.

La interfaz interactiva y accesible de la aplicación web, permite a los usuarios visualizar de manera efectiva las vibraciones en tiempo real, facilitando la comprensión y análisis de los datos generados por el prototipo mecánico.

La implementación y el uso de un sensor MPU6050 y un ESP32 para la captura y transmisión de datos, ha validado la viabilidad de la aplicación en un entorno experimental, ya que con tan pocos sensores se pudo llegar a medir, además de otros, muchos datos de vibración, de esta forma, se pudo sacar el máximo provecho para poder hacer prevenciones en los mantenimientos.

Este proyecto contribuye al avance del conocimiento dentro del área de las vibraciones mecánicas, así como también a la esfera tecnológica, ya que la aplicación web proporciona una solución práctica y eficiente para la recolección y análisis de datos, beneficiando además de esta manera, a estudiantes y profesionales interesados en conocer y comprender la forma en cómo se comportan los sistemas mecánicos en tiempo real.

Para la implementación del proyecto a nivel práctico y la expansión de su uso, se considera la necesidad de adquirir un dominio y un servidor VPS, como Digital Ocean, esto con el propósito de alojar la API en un entorno de producción, lo que garantiza la disponibilidad y accesibilidad continua de la aplicación, así como también resalta la escalabilidad para operar esta solución dentro de la industria.

Este proyecto ha cumplido su propósito para el monitoreo y graficación de datos en tiempo real para un equipo rotatorio, pero, además, ha sentado las bases para futuras investigaciones y aplicaciones en el campo de las vibraciones mecánicas. Por lo que la combinación de la tecnología moderna con la experimentación práctica, ha resultado en una herramienta valiosa que, sin duda, contribuirá al conocimiento de las personas que hagan uso de la aplicación y la comprensión de los sistemas mecánicos en movimiento, con la ventaja de que se utilizaron tecnologías modernas para que no quede descontinuada en un corto plazo.

## BIBLIOGRAFÍA

[1] Rodríguez-Gómez, D., Casto, D., y Meneses, J. (2018). Usos problemáticos de las TIC entre jóvenes en su vida personal y escolar. *Comunicar*, 26(56), 91-107. doi: <https://doi.org/10.3916/C56-2018-09>.

- [2] Otero, A., Rivera, W., Pedraza, C. y Canay, J. (2019). Tic para la educación: sistema adaptativo basado en mecanismos de aprendizaje automático para la apropiación de tecnologías en estudiantes de educación media. *Telos*, 21(3), 526-537. doi: <https://doi.org/10.36390/telos213.03>.
- [3] Cruz, M., Pozo, M., Aushay, H., y Arias, A. (2019). Las Tecnologías de la Información y de la Comunicación (TIC) como forma investigativa interdisciplinaria con un enfoque intercultural para el proceso de formación estudiantil. *E-Ciencia de la información*, 9(1), 44-59. doi: <https://doi.org/10.15517/eci.v1i1.33052>.
- [4] Belandria, G. (2019). El concepto de trabajo y las TIC. *Sapientia organizacional*, 6(12), 21-56.
- [5] Buenrostro, H., y Hernández, M. (2019). La incorporación de las TIC en las empresas. Factores de la brecha digital en las Mipymes de Aguascalientes. *Economía: teoría y práctica*, 50, 101-124. doi: <http://dx.doi.org/10.24275/ETYPUAM/NE/502019/Buenrostro>.
- [6] Pacheco, D., y Rodríguez, R. (2019). Las TIC como estrategia competitiva en la gestión empresarial. *Enfoque*, 3(12), 286-298.
- [7] Ibarra, M., Eusebio, R., y Julio, L. (2023). Red neuronal convencional para la clasificación de piezas mecánicas usando un sistema de visión artificial. *Ingeniantes*, 2(2), 67-72.
- [8] Viveros, I., Muñoz, J., y Aguirre, J. (2021). Diseño de sistema de análisis de vibraciones por adquisición de datos con interfaz gráfica. *INNO-DOCT 2021*. 561-568. Editorial Universitat Politècnica de València. doi: <http://dx.doi.org/10.4995/INN2021.2021.13432>.
- [9] Maigua, C., Quitiaquez, W., Simbaña, I., y Quitiaquez, R., Toapanta, L., y Isaza-Roldan, C. (2020). Diseño de un sistema de monitoreo de vibraciones mecánicas en generadores hidroeléctricos de media potencia. *Técnica energía*, 17(1), 92-102. doi: [10.37116/revistaenergia.v17.n1.2020.397](https://doi.org/10.37116/revistaenergia.v17.n1.2020.397).
- [10] Flores, E., Albornoz, A., López, E., y Romero, J. (2020). Evaluación mediante vibraciones de los rodamientos de la transmisión de un prototipo mini Baja SAE. *Ingeniería UC*, 27(1), 41-53.
- [11] Dinardo, G., Fabbiano, L., y Vacca, G. (2018). A smart and intuitive machine condition monitoring in the Industry 4.0 scenario. *Measurement*, 126, 1-12, doi: <https://doi.org/10.1016/j.measurement.2018.05.041>
- [12] Camargo-Amado, R., Mosquera-Ayala, A. (2023). La revolución industrial 4.0. *Ingeniería y competitividad*, 25(2), 1-2. doi: <https://doi.org/10.25100/iyv.v25i2.13294>.
- [13] ElectronicWings (14 noviembre, 2024). MPU6050 Gyroscope Interfacing with ESP32. [En línea]. Disponible: <https://www.electronicwings.com/esp32/mpu6050-gyroscope-interfacing-with-esp32>.
- [14] React (14 noviembre, 2024). React, la biblioteca para interfaces de usuario web y nativas. [En línea]. Disponible: <https://es.react.dev/>.
- [15] Ochoa, A., Cangrejo L. y Delgado, T. (2018). Alternativa Open Source en la implementación de un sistema IoT para la medición de la calidad del aire. *Revista cubana de ciencias informáticas*, 12(1), 189-204.
- [16] Timarán-Pereira, R., Chaves-Torres, A. y Ordoñez-Erazo, H. (2023). Decision Tree Algorithm Moderately Coupled to PostgreSQL DBMS. *Revista Facultad de Ingeniería*, 32(66), 1-17. doi: <https://doi.org/10.19053/01211129.v32.n66.2023.16777>.

