

Solución IoT para el Registro y Visualización del Estado de la Producción en la Industria 4.0

Silva Pablo Eduardo ^{a *}, Iurinic Gerardo Mathias ^{a b}, Krzyzanowski Lucas Nicolás ^a

^a *Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.*

^b *GIDE, FI-UNaM, Oberá, Misiones, Argentina.*

e-mails: pablosilva.ten@gmail.com, gerardo_m_iurinic@hotmail.com, lucasclark997@gmail.com

Resumen

Se presenta una solución integral de IoT para el monitoreo en tiempo real del estado de producción en plantas industriales, orientada a identificar periodos improductivos de las máquinas. A través de módulos de sensado distribuidos, se captura información sobre los tiempos de encendido y actividad productiva, la cual es centralizada en un servidor local. Una plataforma *web* interactiva procesa estos datos, ofreciendo gráficos porcentuales y de evolución temporal, controles para filtrar por turno o rangos personalizados, y la posibilidad de exportar informes en formato Excel. Esta combinación de sensado embebido y visualización dinámica mejora la visibilidad de los procesos productivos y apoya la toma de decisiones basadas en estos datos.

Palabras Clave – *Dashboard; Docker; Flask; Industria 4.0; Internet de las Cosas; Producción; Raspberry Pi; Servidor; Sensado; Visualización de datos;*

1 Introducción

La Industria 4.0 representa un cambio de paradigma en los sistemas productivos globales. También conocida como la cuarta revolución industrial [1], esta transformación se caracteriza por la integración de tecnologías digitales, físicas y biológicas que modifican no solo los procesos de manufactura, sino también los modelos de negocio y la interacción entre actores económicos. Según el Foro Económico Mundial, esta revolución no se limita a la automatización, sino que introduce sistemas ciberfísicos, Internet de las Cosas (IoT), inteligencia artificial (IA) y análisis de datos avanzados en todas las etapas de la cadena de valor. La mencionada convergencia tecnológica habilita la creación de entornos de producción más conectados, capaces de recolectar, procesar y compartir información en tiempo real, redefiniendo la manera en que los bienes son diseñados, fabricados, monitoreados y mantenidos.

El Internet de las Cosas es uno de los ejes de la Industria 4.0. Básicamente consiste en conectar sensores y dispositivos a las máquinas para recopilar datos en tiempo real sobre su estado y rendimiento. Esto habilita nuevas formas de gestión de la producción, ya que las empresas pueden monitorear cada máquina desde cualquier lugar y reaccionar rápidamente ante anomalías. En la práctica, la información provista por el IoT optimiza la eficiencia operativa: al detectar ineficiencias o paros tempranos, se reduce el tiempo improductivo y se maximizan los recursos [2].

La adopción del IoT en fábricas ha permitido la creación de paneles de control y *dashboards* donde se visualiza el desempeño de cada línea. La analítica de datos en tiempo real, mejora significativamente la toma de decisiones sobre la marcha y también a largo plazo [2].

En Latinoamérica la creciente adopción del IoT está impulsando la transformación digital al permitir que empresas manufactureras, agrícolas y logísticas recojan y analicen sus datos, pudiendo reducir costos y mejorar gradualmente la calidad de sus productos o servicios [3]. Por ejemplo, se documenta el caso de Tecpetrol (sector energético), donde 200 sensores envían 3 millones de datos por hora desde yacimientos remotos, permitiendo a la empresa estandarizar operaciones y tomar decisiones desde centros urbanos [4].

La digitalización de la planta productiva conlleva una capacidad inédita del control de la producción. Con el IoT surgen el Sistema de Ejecución de la Manufactura (MES, por sus siglas en inglés) o uno más completo el cual se trata de la Gestión de Operaciones de Manufactura (MOM), con estos integrados las empresas pueden medir en tiempo real indicadores clave de fabricación como el *Overall Equipment Effectiveness* (OEE), tiempos de parada de máquina, unidades defectuosas, entre otros. Por ejemplo, al disponer de terminales HMI (interfaz hombre-máquina) conectadas, el personal puede reportar automáticamente cada incidencia o finalización de orden, alimentando el sistema con datos precisos. De esta forma, el equipo de mantenimiento puede anticipar paradas antes de que ocurran; además, los sistemas inteligentes de análisis predictivo pueden alertar sobre tendencias anómalas y programar ya intervenciones preventivas [5]. Todo esto contribuye a reducir los periodos improductivos —casos donde las máquinas permanecen encendidas sin producir— y a incrementar la productividad general. Estas soluciones IoT para el piso de fábrica facilitan reportes personalizados y alarmas, mejorando notablemente la supervisión y gestión de la producción.

En Argentina y la región, el interés por Industria 4.0 e IoT crece rápidamente y se encuentra en pleno desarrollo. El informe del Banco Interamericano de Desarrollo (BID) destaca que tecnologías como IoT, *big data* e IA “refuerzan la importancia de la industria manufacturera a partir de la fabricación de productos personalizados e inteligentes” [6]. En la práctica, muchas empresas argentinas ya incorporan componentes 4.0 en sus procesos. Un estudio con PyMEs argentinas señala que diversas compañías están aplicando Tecnología de la Información y las Comunicaciones (TIC), servicios *cloud* e IA para automatizar su producción de principio a fin [7]. Este fenómeno no se limita a sectores avanzados: la manufactura básica, los alimentos y otros rubros también integran IoT y automatización para mejorar sus operaciones.

Sin embargo, persisten los desafíos locales. En la región falta infraestructura de conectividad robusta y presupuestos amplios, lo que retrasa la adopción masiva del IoT [3] [4]. En Argentina en particular, sólo un tercio de las empresas afirma planear la incorporación total de tecnologías 4.0 en los próximos años y el 70% identifica la escasez de personal calificado como la principal limitante [4]. A pesar de ello, organismos públicos e iniciativas de colaboración regional (Argentina-Paraguay-Brasil) están impulsando la transformación digital mediante programas de formación, incentivos e investigación conjunta [6].

A pesar de todos los indicadores y sistemas mencionados anteriormente, muchas plantas industriales en la región aún carecen de herramientas que midan con precisión los periodos improductivos de sus equipos. Con frecuencia, las máquinas permanecen encendidas sin estar efectivamente produciendo, lo cual genera consumos de energía y costos de mantenimiento

innecesarios que pasan desapercibidos. La falta de visibilidad y de análisis de estos eventos limita la capacidad de los responsables de planta para cuantificar pérdidas y tomar decisiones informadas, dado que los sistemas tradicionales de registro suelen ser manuales, esporádicos y de baja granularidad temporal.

En este contexto, surge la necesidad de una solución que combine la simplicidad de un sistema de sensado local con la potencia de una plataforma *web* para el registro y la visualización continua del estado de producción. El presente trabajo describe una propuesta basada en nodos de adquisición distribuidos, capaces de detectar periódicamente el estado de cada máquina. Estos datos se transmiten a un concentrador central, que los almacena en una base de datos para su posterior consulta. A partir de allí, se implementa un *dashboard* interactivo que permite explorar, filtrar y comparar el comportamiento de todas las máquinas, tanto por turnos en días específicos como en rangos de tiempo arbitrarios. Se establecen dos tipos principales de visualizaciones:

- **Gráficos de torta** que muestran de forma porcentual el tiempo durante el cual la máquina estuvo encendida y, de ese conjunto, el tiempo que realmente estuvo produciendo.
- **Gráficas temporales** donde el eje horizontal representa la secuencia de muestras y permite observar con exactitud los instantes en que cambian los estados de la máquina.

Además, la interfaz facilita la comparación simultánea de varias máquinas, para detectar si un evento corresponde a una falla general en la planta o a un problema puntual de un solo equipo.

2 Arquitectura General del Sistema

La solución propuesta en este trabajo tiene como fin el registro y visualización de los tiempos en los cuales las máquinas, dentro de una industria, se encuentran encendidas/apagadas y además si las mismas están produciendo o no. En la *Fig. 1* se muestra la visión global del sistema, a la izquierda, cada máquina dispone de un módulo de adquisición embebido que, mediante señalización eléctrica, comunica su estado.

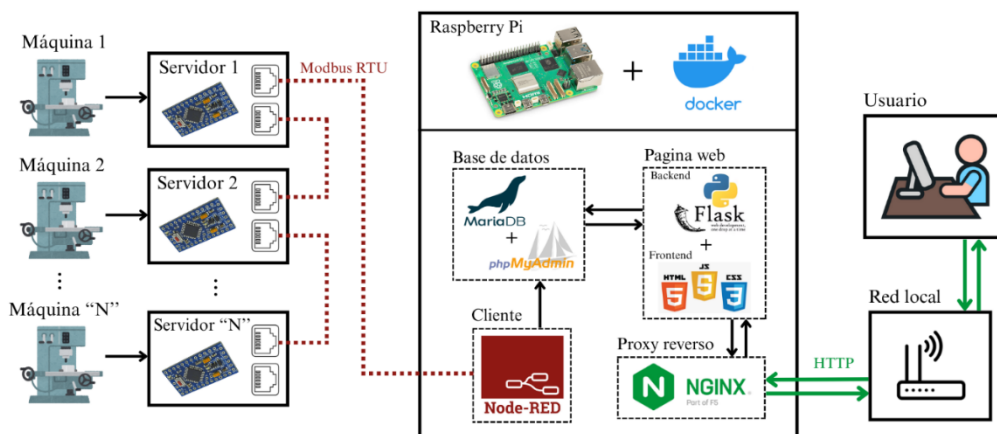


Fig. 1. Diagrama en bloques de la solución IoT completa.

Estos módulos forman una topología en cadena (*Daisy-chain*) y emplean Modbus RTU, sobre RS-485 en la capa física, para transmitir datos hasta la unidad central. En el centro del diagrama, la Raspberry Pi orquesta todo el sistema dentro de contenedores Docker, donde:

- **Node-RED** recibe las tramas Modbus RTU de cada servidor, las decodifica y las inserta en la base de datos.

- **MariaDB** almacena los registros de estado y mantiene vistas optimizadas para su consulta. Se ofrece administración de esta mediante el servicio **phpMyAdmin**.

- **Nginx** se configura como proxy reverso, donde según la ruta solicitada, encamina el tráfico hacia el backend Flask o las páginas estáticas del frontend.

- **Flask** implementa la API REST que atiende los diferentes endpoints procesando consultas a la base de datos y devolviendo un JSON con los datos necesarios para los dashboards.

A la derecha, desde cualquier navegador en la red local, el usuario accede al frontend mediante peticiones HTTP al proxy Nginx y la API Flask, puede alternar entre gráficos de tortas y series de tiempo, así como descargar reportes Excel sobre los periodos de interés. Esta arquitectura modular aprovecha el uso de contenedores Docker para asegurar que cada componente sea fácil de desplegar, escalar y actualizar sin afectar al resto del sistema.

3 Módulo de Adquisición de Datos

3.1 Plataforma de sensado en las máquinas

El módulo de sensado instalado en cada máquina integra el muestreo directo de canales de detección independientes los cuales, mediante un puente rectificador y un optoacoplador por cada uno, convierten la presencia de tensión en cierta parte la máquina en una señal digitalizada apta para un microcontrolador. Su principio de funcionamiento consiste en detectar alimentaciones específicas dentro de la máquina, que indiquen el estado de esta, por ejemplo, una bobina que solo se energiza cuando un operario esta activamente trabajando o una resistencia por la cual circula corriente solo ante cierta situación. Esta señal de entrada es rectificada y filtrada, para luego excitar el diodo LED interno del optoacoplador cuando existe voltaje, esta configuración se puede observar con mas detalle en la *Fig. 2*. La malla de colector del fototransistor a la salida del optoacoplador se conecta a un pin de un Arduino Pro Mini, el cual es el corazón de este sistema embebido. El firmware del microcontrolador interpreta cada canal como un parametro clave de la máquina a registrar. El diseño se realizó pensando en máxima flexibilidad, aunque hoy sólo se usan dos entradas (alimentación de la máquina y estado de producción), el tercer canal queda libre para futuras expansiones o para algún caso donde se requiera la medición de 3 parámetros.

En el esquema eléctrico también puede verse el puente MB8S, el condensador de suavizado y la resistencia de aislación que limitan corriente al LED del 4N25, así como la salida protegida para el GPIO del Arduino donde nuevamente se tienen LEDs indicadores de presencia de la señal. Que se encuentren instalados antes y después del optoacoplador facilitan la verificación visual de la señal de

entrada y salida, y aportan una capa de diagnóstico inmediato. Gracias al aislamiento óptico del 4N25, cualquier sobretensión o fallo en la alimentación de la máquina no repercute en el microcontrolador ni en el resto del sistema.

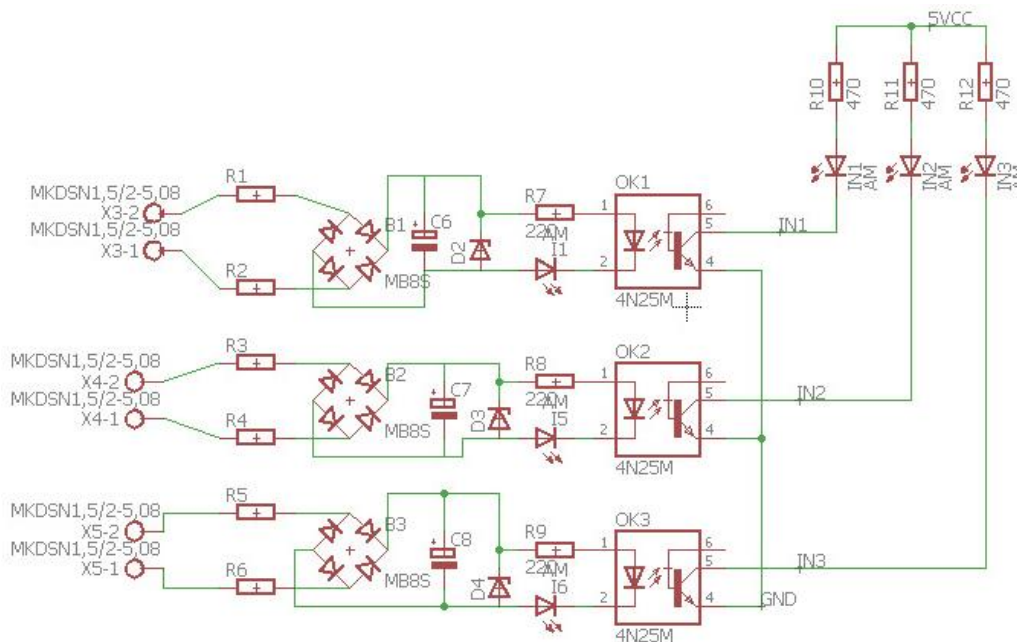


Fig. 2. Esquema eléctrico del circuito que realiza el sensado.

3.2 Comunicación y Registro de Datos

Para garantizar la integridad en entornos industriales con alto ruido electromagnético, la comunicación entre cada módulo y el concentrador de información se emplea el protocolo Modbus RTU encapsulado sobre RS-485, donde cada uno actúa como un servidor que responde al cliente. Para esto se utiliza un transceptor MAX485 el cual convierte los niveles TTL del microcontrolador en líneas diferenciales balanceadas, robustas ante interferencias.

Físicamente, cada módulo dispone de dos conectores RJ-45, dispuestos en configuración de “*Daisy-chain*” abierto (*open-loop*), lo que permite ampliar la red sin necesidad de cableados extra y evitando redundancia o bucle cerrado para una implementación más sencilla, el mismo par recibe tanto la alimentación para la placa como los pares de datos A/B de RS-485. Con este diseño, los módulos pueden interconectarse en serie, finalizando el bus en el último nodo con una resistencia de terminación para evitar reflexiones. De este modo, se logra una topología sencilla de escalera, donde cada servidor de sensado alimenta y comunica a su vez con el siguiente en la línea, antes de alcanzar el concentrador Node-RED que reside en la Raspberry Pi.

La tabla principal registra, para cada evento, un *id* auto incremental como clave primaria del registro, la marca de tiempo en GMT-3 (hora estándar de Argentina), el identificador de la máquina y dos indicadores booleanos: mOn, que refleja si la máquina está encendida (1) o apagada (0); y

mWo, que señala si está produciendo (1) o no (0). Como ejemplo, la ilustra la estructura de la base de datos.

Tabla 1 - Estructura de los registros almacenados en la base de datos.

Id	Tiempo	N° de Máquina	mOn	mWo
1	2025-07-06 07:00:00	3	1	0
2	2025-07-06 07:03:00	3	1	1

En esta solución, el intervalo de muestreo es configurable en el código del microcontrolador, en esta implementación se estableció en 3 min, lo que permite un equilibrio entre granularidad de datos y carga de la red.

4 Plataforma Web de Consulta y Visualización

4.1 Backend y servicios Web

La capa de backend se implementó en Python utilizando el micro-*framework* Flask, orquestado junto a Nginx, MariaDB y phpMyAdmin dentro de contenedores Docker. Esta decisión permite un despliegue modular, escalable y desacoplado, donde cada componente puede actualizarse sin afectar al resto.

4.1.1 Contenedores y orquestación: MariaDB almacena los datos de los sensores en tablas optimizadas para consulta. Y phpMyAdmin ofrece una interfaz administrativa para verificar esquemas, índices y ejecutar consultas *ad-hoc*; también implementa una view o vista en español, la cual expone un *endpoint* REST (/api/vista) que, según los filtros recibidos (fecha/turno o rango personalizado), calcula internamente los instantes de inicio y fin y agrupa en la base de datos los registros de cada máquina para obtener el total de muestras, tiempo encendida, tiempo produciendo y tiempo encendida sin producir. Devuelve estos valores ya agregados en un JSON que el *frontend* consume para construir dinámicamente los gráficos y permitir al usuario explorar el comportamiento de las máquinas por turnos o periodos libres, sin preocuparse por detalles de zonas horarias, consultas SQL ni procesamiento de datos. Luego Flask expone una API REST en el puerto interno 5000. Atiende varios endpoints que procesan parámetros de consulta (fecha, turno o rango libre) y ejecutan sentencias SQL para la mencionada “vista” para recuperar series de datos agregados o individuales. Y Nginx funciona como proxy reverso. Recibe todas las peticiones HTTP en el puerto 80, y enruta los paquetes según la ruta solicitada:

- Rutas estáticas (/static, /) hacia los archivos HTML/CSS/JS del frontend.
- Prefijo /api/ hacia el contenedor Flask.
- /phpmyadmin hacia el contenedor de administración de la base de datos.

4.1.2 Endpoints principales del backend:

- **GET /api/vista?date=<YYYY-MM-DD>&turno=<TM|TT|TN>**

Recupera, para cada máquina, los totales de muestras (N), encendido (On), apagado (Off), produciendo (Prod) y sin producir (NoProd) en el intervalo correspondiente al turno seleccionado.

- **GET /api/vista?start=<ISO>&end=<ISO>**

Permite rangos arbitrarios mediante parámetros start y end en formato ISO-8601 (por ej. 2025-07-06T07:00).

- **GET /api/maquina/<m>?date=...&turno=...**
- **GET /api/maquina/<m>?start=...&end=...**

Devuelven la serie temporal completa de un único módulo: timestamp, mOn, mWo para cada muestra. Para un turno seleccionado o para un rango arbitrario respectivamente

4.1.3 *Gestión de Parámetros y validaciones:* Flask valida que siempre se provean “date+turno” o “start+end”; en caso contrario, redirige a la página de inicio (/). Las consultas SQL utilizan marcadores de posición y vistas materializadas para mejorar el rendimiento bajo alta concurrencia. Todos los timestamps se convierten a la zona GMT 3 antes de la presentación.

4.1.4 *Seguridad y despliegue local:* Aunque la red se limita al ámbito local (no se exponen puertos al exterior), Nginx incluye reglas de firewall y también puede integrarse con certificados de *Let's Encrypt* para implementar HTTPS. En este caso considerando la existencia de una red segura y únicamente de uso interno de la empresa, se optó por el uso de HTTP sin cifrado de datos.

4.2 Interfaz de Usuario y Visualización de Datos

La interfaz gráfica se ha desarrollado con una plantilla de Bootstrap 5 (tema *Cerulean*) y los gráficos con la librería Chart.js, ofreciendo al usuario controles intuitivos para seleccionar fechas, turnos de trabajo o intervalos arbitrarios mediante campos de tipo *date* y *datetime* local. Al enviar estos parámetros, el *frontend* realiza consultas asíncronas (*fetch*) al *endpoint* “/api/vista”, que devuelve para cada máquina los totales de tiempo encendida, produciendo y sin producir. A partir de esta información, la página genera dinámicamente gráficos de torta superpuestos (*Fig. 3*). La capa interna del gráfico de pastel muestra el porcentaje de encendido frente al tiempo apagada, mientras que la capa externa, dentro del porcentaje correspondiente al tiempo encendido, que porcentaje de este tiempo fue de producción efectiva frente al tiempo en que la máquina estuvo encendida sin producir. Con la implementación de leyendas personalizadas y la capacidad de alternar la visibilidad de cada capa de forma dinámica, el usuario puede explorar en detalle los datos sin recargar la página.

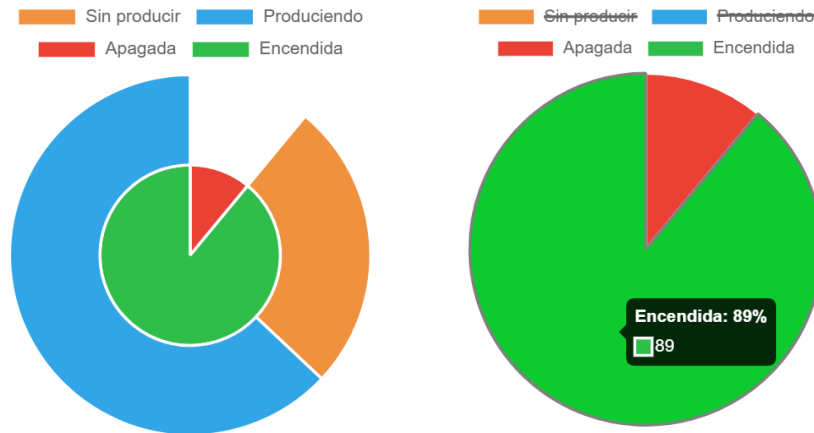


Fig. 3. Gráfico dinámico de torta doble representando el tiempo en porcentajes.

Cuando se requiere un análisis más fino sobre la evolución temporal de una sola máquina, la aplicación invoca a “/api/maquina/<m>” con los mismos parámetros de fecha, turno o rango libre, y representa la serie de muestras como un gráfico de líneas escalonado (Fig. 4). El eje horizontal recoge la marca de tiempo de cada registro, y el eje vertical distingue tres niveles fijos: “Apagada o encendida sin producir” (valor 0), “Produciendo” (valor 0,8) y “Encendida” (valor 1).

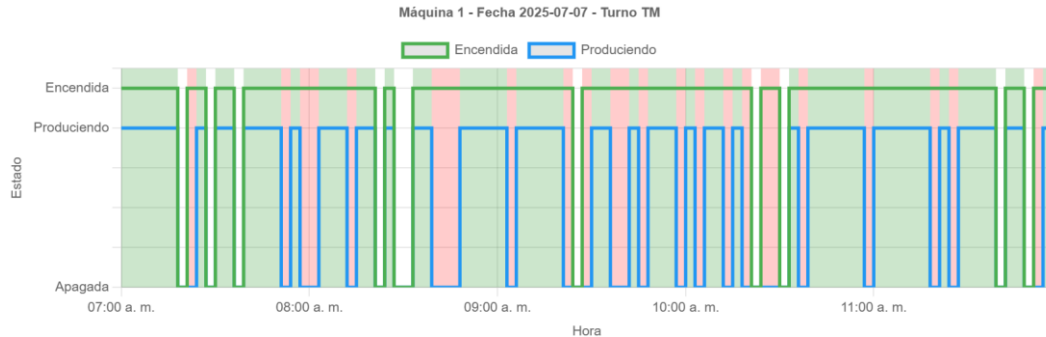


Fig. 4. Gráfico de muestras a lo largo del tiempo.

Nótese también que se pintan de verde los intervalos donde la máquina se encuentra efectivamente encendida produciendo y de rojo los tiempos que son de especial importancia reconocer, donde la máquina se encuentra encendida sin producir. Cada vez que cambia la selección de máquina o de intervalo, la instancia de Chart.js correspondiente se destruye y vuelve a crearse para reflejar con exactitud los nuevos datos. Cabe mencionar que, si no se proporcionan parámetros al cargar la página, el sistema automáticamente consulta el último turno con datos disponibles y ajusta los controles para mostrar siempre la información más reciente.

5 Resultados

5.1 Hardware ensamblado

El diseño del módulo de sensado y comunicación de estado se envió a producción en masa a un fabricante en China, obteniéndose placas profesionales que incorporan todos los componentes críticos y algunas mejoras para futuras aplicaciones. En la *Fig. 5* se puede observar que cada placa incluye, además de los puentes rectificadores y optoacopladores 4N25, el lugar para colocar un circuito de potencia con relé electromagnético que abre la puerta a integraciones de automatización (por ejemplo, activar rampas de enfriamiento o alarmas), manteniendo la máxima flexibilidad en la aplicación como se mencionó anteriormente. El transceptor RS-485 modelo MAX485 se seleccionó con un encapsulado DIP (*Dual In-line Package*) de 8 pines para facilitar su eventual reemplazo en caso de una falla en la línea de comunicación, luego están los dos conectores RJ-45 industriales que suministran alimentación y datos en un único cable, manteniendo la topología “*daisy-chain*” sin redundancia. También en la parte inferior se dejó un área destinada al rotulado manual del número correspondiente de servidor (o esclavo siguiendo la nomenclatura anterior) que se le asigna cada módulo por el cliente (o maestro) para la comunicación, pudiendo tener así una rápida visualización de que placa física se corresponde a cada servidor.

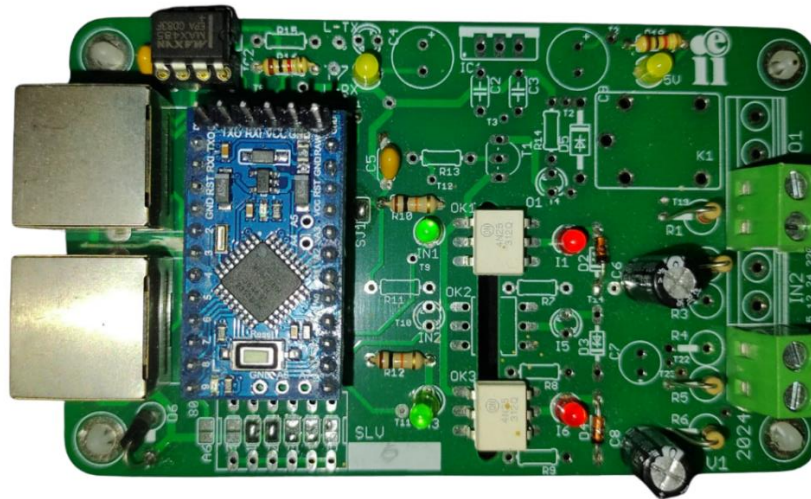


Fig. 5. Circuito confeccionado para sensado y envío del estado de las máquinas.

5.2 Pantallas del Sistema

Todo el sistema ha sido desarrollado utilizando herramientas y tecnologías de código abierto, lo que facilita su adopción, personalización y escalabilidad en entornos industriales diversos. Además de las funcionalidades específicas, la plataforma incorpora elementos transversales que enriquecen la experiencia del usuario y refuerzan su usabilidad. En la barra de navegación superior se encuentra el botón de cambio de tema (claro/oscuro), el cual es persistente mediante la sesión del navegador. Esto

permite que, al navegar en el sistema, la interfaz mantenga automáticamente la preferencia seleccionada anteriormente, garantizando coherencia visual. Junto a él se sitúa el botón de descarga de registros, que invoca al endpoint “/descargar”, generando un archivo Excel con los datos del período consultado. Esta función resulta especialmente útil para análisis offline, informes periódicos o auditorías internas, y está implementada usando el paquete pandas de Python en el *backend*. Estas funcionalidades complementarias convierten al sistema no solo en una herramienta de monitoreo en tiempo real, sino también en una plataforma de consulta histórica adaptable a distintos entornos operativos.

5.2.1 *Pantalla de Login:* Al llegar a la URL de acceso, el usuario se encuentra con un formulario como se observa en la Fig. 6. Adoptando el tono azul característico del tema *Cerulean*. Justo encima del título “Iniciar sesión” aparece un icono de persona de *Bootstrap Icons*, reforzando el propósito de la página. En esta pantalla se dejó la barra de navegación únicamente con el botón para cambiar de tema, de modo que toda la atención se centra en la validación de credenciales. Tras pulsar “Aceptar”, si la autenticación es correcta, el usuario es redirigido al *dashboard*; de lo contrario, se muestra un mensaje de error en rojo justo debajo del formulario, indicando “Usuario o contraseña inválidos”. También en la parte inferior de la carta se incluye un hipervínculo a la pantalla de registro para usuarios nuevos la cual es prácticamente similar a esta de inicio de sesión.

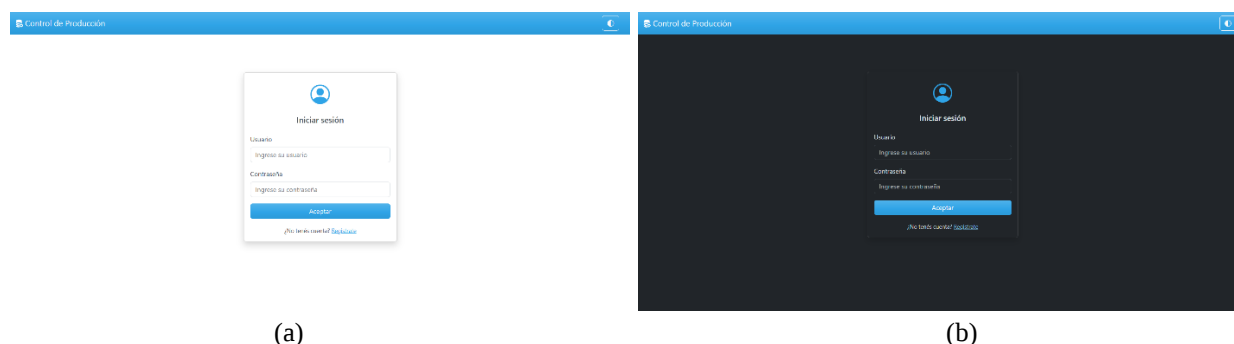


Fig. 6. Pantalla de login: (a) Tema claro; (b) Tema oscuro.

5.2.2 *Dashboard de Gráficos de Torta:* Una vez dentro, la barra de navegación reaparece con el logo del proyecto a la izquierda, el selector de vista temporal (ícono de *Kanban*) —que nos lleva a la página de gráficos temporales—, el cambio de tema y los botones de descarga y cierre de sesión a la derecha, Fig. 7. Bajo la barra, un bloque de controles ocupa toda la anchura: a la izquierda un *picker* de fecha y junto a él un *dropdown* de turno, y a la derecha un campo de fecha-hora de inicio y otro de fin para rangos libres. El botón “Filtrar” ejecuta una consulta al endpoint “/api/vista”, y “Consultar rango libre” recarga con parámetros personalizados. En cuanto llegan los datos, la sección de tarjetas se llena con una por máquina; cada una incorpora un gráfico de pastel doble —detallado en la sección 4.2—, una leyenda interactiva y un pequeño texto con totales numéricos. En la parte inferior de cada tarjeta aparece un icono de información que, al pasar el ratón, muestra un *tooltip* con la cantidad porcentual de minutos encendida y produciendo, para el periodo seleccionado.

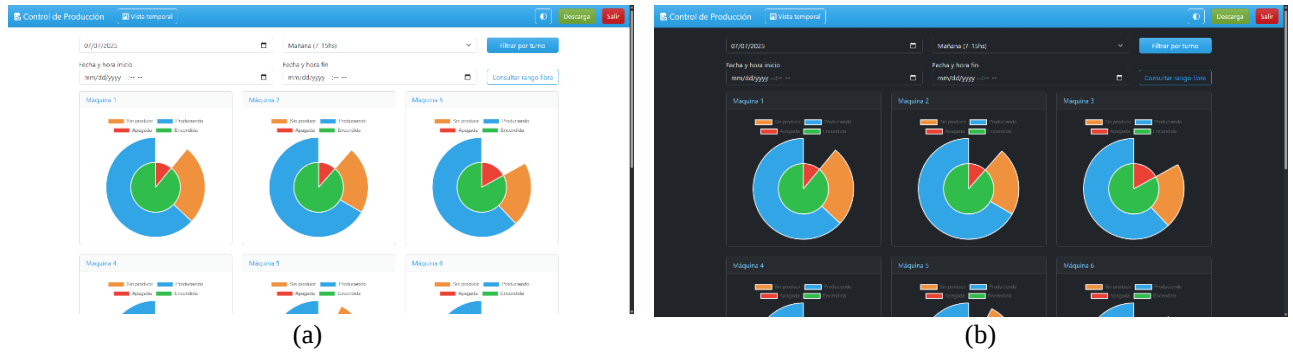


Fig. 7. Pantalla principal con gráficos de torta para todas las máquinas: (a) Tema claro; (b) Tema oscuro.

5.2.3 *Vista Temporal de Máquinas:* Al pinchar el icono de *kanban* o el botón “Vista temporal” mencionado en la pantalla anterior, accedemos al *endpoint* “/maquinas” donde podemos ver todas las muestras para un período seleccionado como se muestra en la Fig. 8. Aquí, el *navbar* y el selector de tema permanecen fijos, pero los controles de fecha/turno desaparecen. Inmediatamente debajo, una hilera de botones *checkbox* de color secundario listan todas las máquinas; por defecto todas están activas. Cada bloque de máquina ocupa una fila con dos columnas: a la izquierda su gráfico de líneas, en el que el eje vertical discrimina tres estratos (“Apagada”, “Encendida”, “Produciendo”) y el horizontal muestra la hora con marcas cada 1 hora; a la derecha un recuadro con un resumen numérico y porcentual del período visualizado. Cuando el usuario desmarca una máquina, su gráfico desaparece al instante, sin recargar la página. Si cambia nuevamente fecha o rango libre, los gráficos se destruyen y recrean usando Chart.js para evitar acumulación de memoria.

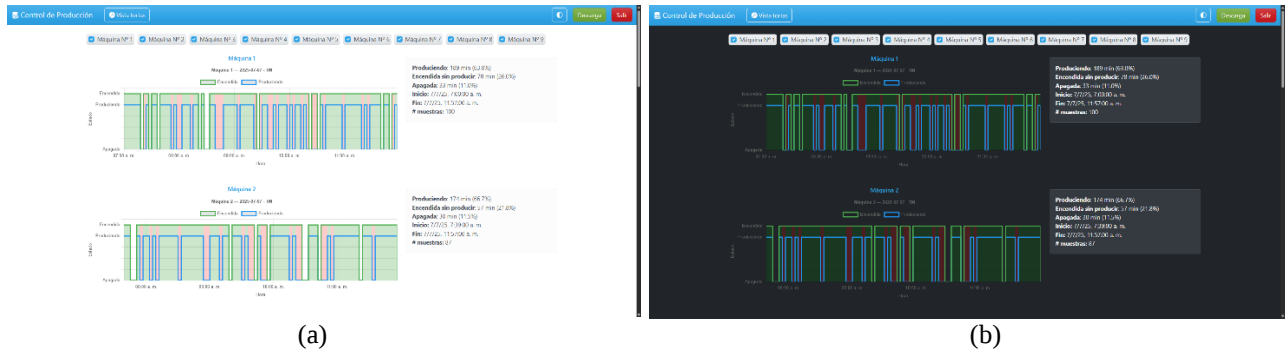


Fig. 8. Gráficos de muestras a lo largo del tiempo con selección de máquinas: (a) Tema claro; (b) Tema oscuro.

6 Conclusiones

Este trabajo presenta el diseño e implementación de una solución integral de monitoreo para plantas industriales basada en tecnologías de bajo costo, código abierto e interfaces web modernas. La arquitectura propuesta permite registrar, almacenar y visualizar el estado de producción de múltiples máquinas en tiempo real, proporcionando a los responsables de planta una herramienta confiable para la supervisión operativa.

Mediante el uso de módulos de adquisición distribuidos conectados mediante un bus de paquetes RS-485 y la posterior centralización de datos en una plataforma web desarrollada con Flask, MariaDB y Node-Red, se logra una trazabilidad completa de los eventos productivos con alta granularidad temporal. El sistema contempla dos modalidades principales de visualización: gráficos porcentuales que resumen la eficiencia operativa por turno, y gráficos temporales que permiten estudiar el comportamiento histórico de cada equipo.

Si bien el sistema no interviene directamente en los procesos productivos, constituye un paso inicial fundamental para la digitalización de las plantas. Al estructurar de forma ordenada y estandarizada la información sobre los estados de producción, se abren nuevas oportunidades para aplicar modelos de análisis de datos, detección de anomalías y predicción de ineficiencias operativas. En este sentido, la solución desarrollada puede entenderse como una base flexible y escalable hacia esquemas más avanzados de mantenimiento predictivo y control basado en datos.

Finalmente, se destaca que la implementación real de este sistema en entornos industriales de la región puede contribuir de manera significativa a reducir pérdidas ocultas de eficiencia, fomentar una cultura de mejora continua, y acercar a las pymes manufactureras al paradigma de la Industria 4.0.

Referencias

- [1] K. Schwab, «The Fourth Industrial Revolution: what it means and how to respond,» World Economic Forum, 16 Enero 2016. [En línea]. Available: <https://www.weforum.org/stories/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>. [Último acceso: 21 Julio 2025].
- [2] «Cómo el IoT está transformando la industria,» 9altitudes, 2023. [En línea]. Available: <https://9altitudes.com/es/aprende-y-conecta/articulos/como-el-iot-esta-transformando-la-industria>. [Último acceso: 21 Julio 2025].
- [3] M. O. Orquera, «Industria 4.0 en Latinoamérica: un camino hacia la transformación digital,» La comunidad logística, 14 Junio 2024. [En línea]. Available: <https://logistica.enfasis.com/ultimas-noticias/industria-4-0-en-latinoamerica-un-camino-hacia-la-transformacion-digital/>. [Último acceso: 21 Julio 2025].
- [4] «IoT en la industria argentina: casos, avances y desafíos,» InnovaciónDigital360, 27 Mayo 2025. [En línea]. Available: <https://www.innovaciondigital360.com/industria-4-0/iot-en-la-industria-argentina-entre-la-transformacion-y-el-desafio/>. [Último acceso: 21 Julio 2025].
- [5] «Industria 4.0, la fábrica digital, flexible y optimizada,» doeet.com, 3 Noviembre 2022. [En línea]. Available: <https://doeet.com/industria-4-0/>. [Último acceso: 21 Julio 2025].
- [6] A. I. Basco, G. Beliz, D. Coatz y P. Garnero, «Industria 4.0: fabricando el futuro,» 2018.
- [7] F. Aleman, D. G. Cortés, E. Hurtado, J. M. Paz Portilla and E. S. Tokashiki Etcheverry, "Industria 4.0 en Pequeñas y Medianas Empresas argentinas," *9º Congreso Nacional de ingeniería Informática/Sistemas de Información*, 2022.