



“Implementación de un Pipeline automatizado de descarga y limpieza de datos para reportes operacionales y una mejora en el control operacional en Eleccon Maquinarias SpA.”

POR

Luciano Mauricio Yáñez Carrillo

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Industrial

Profesor Guía
Juan Carlos Caro Seguel

Profesional Supervisor
Jose Luis Alcayaga

Abril 2026
Concepción (Chile)

© 2026 Luciano Mauricio Yáñez Carrillo

© 2026 Luciano Mauricio Yañez Carrillo

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

El/la autor declara que cualquier uso de herramientas de Inteligencia Artificial en este trabajo fue exclusivamente de apoyo y que todo el contenido ha sido revisado, validado y es de su autoría.

Agradecimientos

Sumario

Las empresas del sector minero-industrial que operen en múltiples frentes de trabajo se ven expuestos a un problema frecuente: convertir grandes volúmenes de datos operacionales desordenados en información confiable y útil para la toma de decisiones. Elecon Maquinarias SpA., empresa dedicada a obras civiles, movimientos de tierra y arriendo de maquinaria para el sector minero, no es la excepción. Antes del presente proyecto, dos personas intervenían entre 16 y 18 horas por archivo para descargar, limpiar y ordenar reportes operacionales desde la plataforma Procore, archivos que podían contener hasta 15.000 filas con errores producidos en la digitación, formatos inconsistentes y registros inválidos.

Con el propósito de resolver este problema, el presente trabajo muestra el proceso de diseño e implementación de un pipeline automatizado de extracción, transformación y carga de datos (ETL) desarrollado en Python, usando Pandas, Selenium y openpyxl. La solución tiene dos partes: un script que descarga reportes desde Procore de forma automática, y un módulo de validación que aplica nueve reglas de negocio para detectar inconsistencias o errores críticos en los registros.

Los resultados evidencian una reducción del tiempo de procesamiento desde 16 horas a aproximadamente 1,5 minutos por archivo, lo cual equivale a una disminución de 99% en el tiempo operativo, con la mitad del personal requerido. Además, en el archivo de validación utilizado se detectaron y clasificaron automáticamente 413 registros con errores, generando trazabilidad por tipo de regla. Los Dashboards generados en Power BI a partir de los datos procesados mostraron una mejora notable en la consistencia y confiabilidad respecto del estado inicial.

En conclusión, implementación el pipeline no solo resuelve una necesidad operativa concertar en Elecon Maquinarias SpA., sino, además, sienta las bases para una estrategia de automatización aplicable en otras áreas relacionadas de la empresa, tales como: asistencia de las personas trabajadoras, control de flota y seguimiento del consumo de combustible, entre otros.

ABSTRACT

Companies in the mining and industrial sector operating across multiple work fronts frequently face a common problem: transforming large volumes of disorganized operational data into reliable and useful information for decision-making. Eleccon Maquinarias SpA., a company dedicated to civil works, earthmoving, and machinery rental for the mining sector, is no exception. Before the intervention described in this project, two people spent between 16 and 18 hours per file downloading, cleaning, and organizing operational reports from the Procore platform. These files could contain up to 15,000 rows with data entry errors, formatting inconsistencies, and invalid records.

To address this problem, this work presents the design and implementation process of an automated data extraction, transformation, and loading (ETL) pipeline developed in Python, using Pandas, Selenium, and openpyxl. The solution has two parts: a script that automatically downloads reports from Procore, and a validation module that applies nine business rules to detect inconsistencies or critical errors in the records.

The results show a reduction in processing time from 16 hours to approximately 1.5 minutes per file, which is equivalent to a 99% decrease in operating time, with half the required personnel. Furthermore, the validation file used automatically detected and classified 413 records with errors, generating traceability by rule type. The dashboards generated in Power BI from the processed data showed a significant improvement in consistency and reliability compared to the initial state.

In conclusion, implementing the pipeline not only resolves a critical operational need at Eleccon Maquinarias SpA., but also lays the foundation for an automation strategy applicable to other areas related to company management, such as employee attendance, fleet management, and fuel consumption tracking, among others.



Índice de Contenidos

1	Introducción.....	10
1.1	Antecedentes generales del tema.....	10
1.2	Objetivos del Tema.....	12
1.2.1	Objetivo General.....	12
1.2.2	Objetivos Específicos	12
1.3	Justificación del Tema	13
2	Marco Teórico	14
2.1	Proceso ETL (Extract, Transform, Load).....	14
2.2	Automatización robótica de procesos (RPA)	15
2.3	Procesamiento de Datos con Python y Pandas	16
2.4	Limpieza y estandarización de los datos	16
3	Metodología.....	18
3.1	Análisis del problema actual.....	18
3.2	Diseño de la solución automatizada	19
3.3	Validación de piloto implementación.....	20
3.4	Escalabilidad y replicabilidad.....	21
4	Resultados.....	22
4.1	Estado Inicial del proceso.....	22
4.2	Reglas de validación implementadas.....	23
4.3	Comparación de tiempos de procesamiento.	24
4.4	Errores detectados y corregidos.....	25
4.5	Integración con Power BI	25
4.6	Impacto Organizacional y validación interna.....	27
4.7	Escalabilidad y extensión de la solución	34
5	Discusión	35
5.1	Escalabilidad y replicabilidad.....	36

5.2	Interpretación de los hallazgos	37
5.3	Relación con los objetivos	37
5.4	Fortalezas	38
5.5	Limitaciones	39
5.6	Amenazas a la validez y mitigaciones	39
5.7	Implicancias y generalización	39
6	Conclusiones	40
7	Recomendaciones	41
7.1	Operación y monitoreo	41
7.2	Gobierno y calidad de datos	41
7.3	Adopción y soporte	41
7.4	Escalabilidad	42
8	Trabajos futuros	42
9	Referencias	43
10	ANEXOS	45
10.1	Anexo 1: Automatización de descarga desde Procore (Selenium)	45
10.2	Anexo 2 Validador y reporte de errores (pandas)	47

Índice de Figuras

Figura A 1: Librerías e imports de Selenium.....	45
Figura A 2: Configuración del driver y espera explícita.	45
Figura A 3: Inicio de sesión en Procore (email/clave y botón “login”).....	45
Figura A 4: Selección del proyecto desde el “picker”	45
Figura A 5: Apertura del panel de herramientas.	46
Figura A 6: Ingreso a “Reportes 360.....	46
Figura A 7: Selección del reporte “Reporte Equipos (Nuevo)”	46
Figura A 8: Exportación a Excel del reporte.	46
Figura A 9: Importación y carga del archivo.....	47
Figura A 10: Normalización de columnas y contenedor de errores	47
Figura A 11: Regla 1: Horómetro final < inicial	47
Figura A 12: Regla 2: Conversión a datetime y falla antes de 08:00	47
Figura A 13: Regla 3: Retorno después de 20:00.....	47
Figura A 14: Regla 4: Falla posterior al retorno	48
Figura A 15: Regla 4: Falla posterior al retorno	48
Figura A 16: Consolidación de hallazgos y export del reporte de errores	48
Figura A 17: Generación de nombre/ruta de salida y export	48
Figura A 18: Ruta final del archivo limpio y guardado	48

Índice de Tablas

Tabla 1: Tabla comparación tiempo antes y después de la implementación.	24
Tabla 2: Tabla de impacto en tiempo luego de la implementación.....	28

1 Introducción

1.1 Antecedentes generales del tema

En las organizaciones del sector industrial que operan con múltiples frentes de trabajo, la gestión de información operacional suele depender de diversos frentes de trabajo, la gestión de información operacional suele depender de diversos sistemas digitales y registros administrativos que siempre se encuentran completamente integrados. Esta situación genera dificultades para consolidar información, controlar procesos y obtener indicadores confiables en tiempo adecuado. En este contexto, el control de gestión requiere disponer de mecanismos que permitan transformar grandes volúmenes de datos operacionales en información estructurada, pertinente y disponible para el análisis. La automatización del procesamiento de datos y estandarización de registros se ha transformado en herramientas clave para mejorar la eficiencia operativa y mejorar la toma de decisiones basada en datos dentro de las organizaciones.

Respecto de lo anterior, Elecon Maquinarias SpA es una empresa especializada en la prestación de servicios de obras civiles, movimientos de tierras, construcción de caminos y arriendo de maquinaria para el sector minero. Actualmente, la empresa enfrenta desafíos en la gestión y análisis de datos operativos, debido a que muchos de sus procesos dependen de registros manuales o herramientas poco optimizadas para la automatización, principalmente data cruda descargada a través de hoja de cálculo. La empresa enfrenta una carga operativa significativa por la gestión manual de reportes mineros.

Uno de los principales problemas detectados es la descarga y procesamiento manual de archivos hoja de cálculo con gran volumen de datos, provenientes de distintas plataformas utilizadas por la empresa, como Talana y Procore. Estos archivos contienen información clave para el análisis de asistencia, control de maquinaria, y reportes diarios de equipos en faena, pero llegan generalmente con errores de digitación, nombres mal escritos, formatos desordenados o estructuras inconsistentes. Esto genera una alta carga administrativa y retrasa la construcción de reportes consolidados, afectando la toma de decisiones operativas.

En la actualidad, el análisis de asistencia se realiza de forma manual. La información descargada desde las aplicaciones debe ser trasladada a una planilla de hoja de cálculo que se diseñó, donde se organiza para generar reportes detallados. Esta forma de trabajo permite un análisis más completo, sigue siendo un proceso que requiere intervención manual, lo que implica pérdida de tiempo y posibles errores humanos.

Ante esta situación, el directorio necesita una solución más automatizada que facilite el acceso y análisis de los datos de asistencia sin la necesidad de manipulación manual en hoja de cálculo y que permita descargar de forma masiva estos archivos, corregir sus errores y generar versiones limpias y estandarizadas, listas para el análisis para poder tomar decisiones óptimas de forma más rápida y accesible.

Este trabajo propone e implementa un pipeline automatizado que descarga y estandariza archivos hoja de cálculo, detectando detecta errores con trazabilidad por regla. La solución pretende disminuir la intervención manual y mejorar la oportunidad y calidad de la información para la toma de decisiones operacionales.

1.2 Objetivos del Tema

1.2.1 Objetivo General

Diseñar y validar un pipeline automatizado para la extracción, transformación y preparación de datos operacionales provenientes de plataformas utilizadas por la Elecon Maquinarias SpA., con el propósito de mejorar la calidad, consistencia y confiabilidad de la información disponible para el control operacional y el análisis de indicadores mediante herramientas de visualización.

1.2.2 Objetivos Específicos

- Diseñar un script de automatización que permita la descarga estructurada de reportes operacionales desde las plataformas digitales utilizadas por la empresa.
- Desarrollar un módulo de validación de datos basado en reglas de negocio que permita detectar diferencias en los registros operacionales.
- Implementar un proceso de transformación y estandarización de datos que permita preparar los archivos descargados para su posterior análisis.
- Generar conjunto de datos detectados que puedan ser utilizados directamente en herramientas de visualización y análisis de información.
- Validar el desempeño de la solución propuesta mediante la comparación entre el proceso automatizado y el método manual previamente utilizado por Elecon Maquinarias SpA.

1.3 Justificación del Tema

Elecon Maquinarias SpA., como empresa del rubro minero-industrial, genera diariamente una gran cantidad de información operacional proveniente de sus diferentes frentes de trabajo. Esta información es registrada a través de reportes en formato hoja de cálculo descargados desde plataformas como Procore, los cuales reflejan el estado, rendimiento y uso de equipos en faena. Sin embargo, estos reportes presentan errores frecuentes: nombres mal escritos, datos duplicados, formatos desordenados y estructuras variables entre proyectos, lo que dificulta su análisis y requiere una limpieza manual laboriosa.

Este problema genera una pérdida considerable de tiempo para el equipo encargado del análisis de datos, retrasa la toma de decisiones operativas y aumenta el riesgo de inconsistencias en los informes. La necesidad de intervenir manualmente cientos o miles de filas por archivo impide que el equipo directivo acceda rápidamente a información confiable y actualizada para el control de proyectos.

Frente a este escenario, la implementación de una solución automatizada basada en scripts permite abordar el problema desde su origen: automatizando la descarga de archivos desde las plataformas digitales, y posteriormente limpiando y estandarizando la información para dejarla lista para su análisis en dashboards como Power BI. Esta automatización elimina tareas repetitivas, mejora la calidad de los datos y permite al personal enfocar sus esfuerzos en tareas de mayor valor.

Desde una perspectiva académica, el presente trabajo representa un caso real de aplicación de automatización robótica de procesos (Robotic process automation, RPA en inglés) y tratamiento de datos masivos mediante tecnologías como Python y Pandas. La solución desarrollada responde a una necesidad concreta de la empresa, es replicable en otros procesos con estructuras similares, y constituye un avance en la digitalización y eficiencia operativa de Elecon Maquinarias SpA.

2 Marco Teórico

La automatización de procesos operacionales en empresas con alta carga administrativa, como Elecon Maquinarias SpA, representa una tendencia creciente en la búsqueda de eficiencia y reducción de errores humanos. Específicamente, la gestión de datos ha sido tradicionalmente abordada mediante métodos manuales o herramientas no integradas, generando demoras, duplicidad de datos y limitaciones en el análisis. Este marco teórico revisa los conceptos claves que sustentan el desarrollo de scripts, utilizando tecnologías de ciencia de datos como Python, pandas y bases de datos estructuradas.

2.1 Proceso ETL (Extract, Transform, Load)

El procesamiento de grandes volúmenes de datos en las organizaciones suele ejecutarse mediante el enfoque conocido como ETL (Extract, Transform, Load). Este modelo describe el proceso mediante el cual los datos son extraídos desde distintas fuentes, posteriormente, son transformados para asegurar su consistencia y calidad, y finalmente, ser cargados en sistemas que permiten su análisis y visualización y uso.

La etapa de extracción (Extract) corresponde al proceso mediante el cual los datos son obtenidos desde sus fuentes originales, tales como, plataformas digitales, bases de datos o archivos estructurados. En entornos empresariales; esta etapa puede realizarse manualmente o por medio de mecanismos automatizados que permiten descargar la información de forma sistemática.

La fase de transformación (Transform) corresponde al procesamiento de los datos extraídos, donde se realizan actividades como depuración de registros, detección de inconsistencias, estandarización de formatos y aplicación de reglas de validación. Esta etapa es fundamental para asegurar que los datos utilizados en los análisis sean consistentes y confiables,

Finalmente, la etapa de carga (Load) corresponde en la integración de los datos transformados en sistemas de almacenamiento o herramientas de análisis, tales como: sistemas de inteligencia de negocios o plataformas de visualización de datos. Esta fase permite que la

información procesada sea utilizada para la generación de reportes, indicadores y dashboards que apoyen la toma de decisiones al interior de la organización.

2.2 Automatización robótica de procesos (RPA)

Automation Anywhere (s. f.), señala que la RPA es una tecnología que permite automatizar tareas repetitivas mediante el uso de scripts de software que imitan acciones humanas en entornos digitales. En lugar de ejecutar instrucciones manualmente, los scripts interactúan con interfaces gráficas, plataformas web y archivos para realizar operaciones como extracción de datos, descarga de archivos, envío de reportes o validaciones automáticas

De acuerdo con Smagul (2023), en una revisión sistemática reciente se describe a la RPA como el uso de scripts de software que imitan acciones humanas en interfaces digitales para ejecutar tareas repetitivas, y se constata su fuerte crecimiento de adopción en la última década. Además, Según Alfandi (2022), la RPA consiste en scripts de software que replican acciones humanas en sistemas digitales para automatizar tareas repetitivas y basadas en reglas, reduciendo la intervención manual y los errores asociados.

Esta tecnología ha ganado popularidad en los últimos años debido a su efectividad para mejorar la eficiencia y reducir errores humanos en procesos administrativos, como sintetizan Ivančić y Vugec (2019) en su revisión, los beneficios más reportados de RPA son eficiencia, precisión y menor variabilidad, junto con desafíos recurrentes en selección de procesos, seguridad y cambio organizacional. Según PwC (2024), la automatización podría disminuir los errores en los procesos hasta en un 68%, favoreciendo una mayor exactitud y calidad en los productos o servicios ofrecidos.

En el contexto de empresas industriales como Elecon Maquinarias SpA, la automatización de flujos operacionales resulta especialmente útil para tareas como la recopilación diaria de reportes desde plataformas digitales, lo que permite estandarizar procesos y asegurar una mayor trazabilidad en la gestión de proyectos.

También, la implementación de una tecnología que automatice procesos posibilita a las empresas utilizar sus recursos de manera más efectiva, distribuyendo tareas según la capacidad y disponibilidad, lo que optimiza el rendimiento y reduce los gastos.

2.3 Procesamiento de Datos con Python y Pandas

Python es uno de los lenguajes de programación más utilizados en ciencia de datos y automatización por su sintaxis clara y su ecosistema de librerías especializadas. La librería Pandas permite el manejo eficiente de estructuras tipo DataFrame, la capacidad de pandas para trabajar con grandes volúmenes de información en formatos como hoja de cálculo o CSV lo convierte en una herramienta ideal para procesos de automatización de limpieza de datos como explica McKinney (2022), pandas provee estructuras DataFrame y operaciones vectorizadas que facilitan filtrado, agregación y transformación eficientes sobre hoja de cálculo/CSV a escala, ideal para representar tablas de datos como las extraídas de sistemas como Talana o procore.

De acuerdo con McKinney (2022), creador de Pandas (librería de Python), “la manipulación de datos tabulares con estructuras robustas como DataFrame permite ejecutar operaciones de filtrado, agrupamiento y generación de estadísticas con gran eficiencia”.

2.4 Limpieza y estandarización de los datos

El procesamiento de datos operacionales provenientes de múltiples fuentes presenta desafíos importantes, especialmente cuando la información se encuentra en hojas de cálculo generadas manualmente. Problemas como nombres mal escritos, estructuras de columnas variables, registros duplicados o formatos inconsistentes son comunes y dificultan el análisis eficiente de la información.

La limpieza de datos es una etapa fundamental del proceso de análisis, ya que garantiza que los datos sean precisos, completos y uniformes antes de su uso en reportes o visualizaciones.

En muchos casos, la estandarización también implica normalizar formatos de fechas, corregir errores de tipeo, validar rangos numéricos y reestructurar los archivos para que se integren fácilmente con herramientas de dashboard interactivos. Como señalan Rahm y Do (2000), la limpieza de datos aborda errores e inconsistencias para mejorar la calidad, y es una fase crítica del ETL en entornos con múltiples fuentes.

3 Metodología

3.1 Análisis del problema actual

Antes de implementar la automatización, el proceso de obtención y validación de los reportes diarios de equipos en Eleccon Maquinarias SpA. era completamente manual. Cada día, un trabajador debía acceder a la plataforma Procore, identificar el proyecto correspondiente dentro de una extensa lista que incluía proyectos antiguos y activos, ingresar a la sección de reportes, localizar el reporte específico de equipos y, finalmente, proceder a la descarga del archivo en formato hoja de cálculo.

Los archivos descargados contenían entre 10.000 y 15.000 filas, con registros desordenados y errores frecuentes como:

- Nombres de equipos escritos de forma distinta en cada fila.
- Códigos mal digitados.
- Nombres de operadores con faltas de ortografía o sin estandarización.
- Formatos numéricos inconsistentes (uso de coma, punto, ceros adicionales, etc.).
- Inconsistencias en fechas y horas (ej: horómetro final menor al inicial, horas fuera de rango, etc.).

Estos errores hacían que el análisis de datos y la posterior generación de reportes fuese altamente ineficiente. Dos personas dedicaban hasta dos jornadas laborales completas para limpiar un solo archivo, lo que impactaba negativamente en la productividad y agilidad del equipo.

Este trabajo, además de consumir tiempo, interfería con otras funciones críticas del equipo, generando cuellos de scriptella y retrasos en la entrega de reportes operacionales.

3.2 Diseño de la solución automatizada

La solución propuesta consiste en la creación de dos scripts desarrollados en Python:

- **Script de descarga automática:** Utilizando Selenium WebDriver, este script automatiza el acceso a Procore, selecciona el proyecto correcto, navega hasta la sección de reportes y descarga el archivo correspondiente de manera autónoma, sin intervención humana. Realiza las siguientes acciones:

- Inicia sesión en el portal.
- Selecciona el proyecto específico desde una lista extensa.
- Accede al módulo de reportes.
- Localiza el reporte de equipos.
- Ejecuta la descarga del archivo en hojas de cálculo.

Esta automatización elimina la necesidad de realizar el proceso manualmente cada día, estandarizando la operación y liberando tiempo del equipo.

- **Script de validación y limpieza de datos:** Este segundo script lee la hoja de cálculo descargado mediante la librería pandas y aplica múltiples reglas de validación para detectar errores comunes, tales como:

- El horómetro final no puede ser menor a la inicial.
- La hora de falla no debe ser antes de las 08:00.
- La hora de retorno no debe ser después de las 20:00.
- Un equipo no puede estar en ambas mantenciones el mismo día.
- El equipo no puede tener horómetros iguales y estar en operación.
- La fecha de falla no puede ser posterior a la de retorno.
- No puede tener simultáneamente estado "en operación" y ubicación "taller".
- El período debe coincidir con la fecha de registro.
- Siempre que haya mantención, deben estar indicadas las fechas y horas.

Estos errores son extraídos y guardados automáticamente en un archivo aparte, mientras que los registros limpios se exportan a una nueva hoja de cálculo estandarizado, listo para su análisis o carga en Power BI.

3.3 Validación de piloto implementación

El desarrollo de los scripts implicó una etapa de pruebas intensiva, donde se validó el correcto funcionamiento de cada regla de limpieza y la navegación automatizada por Procore. Para ello, se utilizó un entorno de pruebas con proyectos reales y se replicaron distintos escenarios con errores frecuentes.

Las tecnologías utilizadas fueron:

- Python como lenguaje base de automatización.
- Selenium para controlar el navegador y realizar descargas automáticas.
- Pandas para la limpieza estructurada de datos.
- openpyxl para la manipulación avanzada de hojas de cálculo.
- Visual Studio Code como entorno de desarrollo.
- GitHub como sistema de control de versiones interno.
- Microsoft Excel para la verificación visual de los resultados.

La validación de la propuesta se realizó mediante la comparación entre el mecanismo tradicional utilizado por la empresa para el análisis de datos y el pipeline automatizado desarrollado en el presente trabajo.

Para estos efectos se seleccionaron dos archivos operacionales reales provenientes de la plataforma usada por Elecon Maquinarias SpA.; este proceso considera los siguientes indicadores de desempeño:

- a. Tiempo total requerido para preparar los archivos que se van a analizar.
- b. Número de errores detectados en los registros operacionales utilizados.
- c. Nivel de estandarización de los datos generados.

- d. Estatus de preparación de los archivos que serán utilizados en las herramientas de visualización.

Lo anterior permitió evaluar el impacto de la automatización en términos de eficiencia operativa y calidad de la información dispuesta para el análisis.

Además, se contempla retroalimentación directa de los usuarios que revisan o cargan los datos al sistema final. La estrategia busca garantizar que la automatización no solo reduzca tiempos, sino que mejore la calidad y confiabilidad de los datos.

3.4 Escalabilidad y replicabilidad

Debido a que Elecon Maquinarias SpA. utiliza diversas plataformas digitales en diferentes unidades operativas, la solución diseñada es completamente escalable. El modelo de scripts puede adaptarse fácilmente para:

- Automatizar la descarga de otros tipos de reportes.
- Validar datos en áreas como recursos humanos, gestión de flota o seguridad.
- Replicarse al iniciar nuevos contratos o proyectos, reduciendo la curva de entrada operativa.

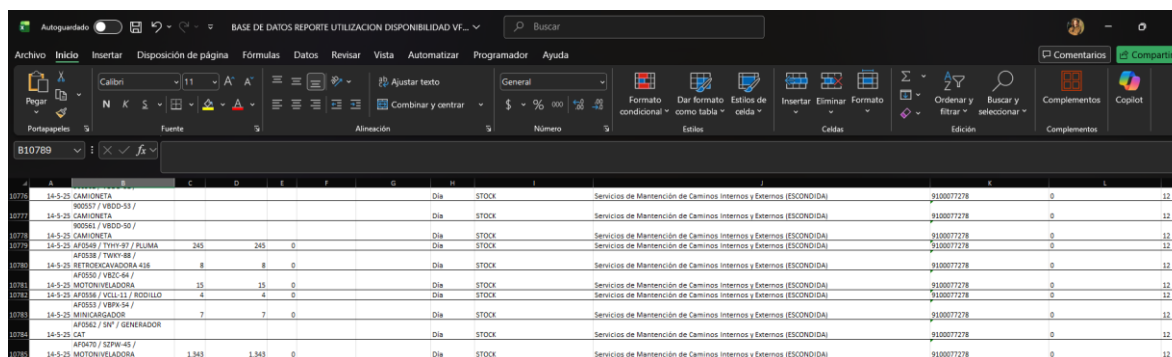
Esta solución, además de resolver un problema puntual, sienta las bases para un enfoque más automatizado y eficiente en el manejo de datos masivos dentro de la organización.

4 Resultados

4.1 Estado Inicial del proceso

Previo a la automatización, la descarga, orden y validación de reportes operacionales se realizaba manualmente a partir de hojas de cálculo con entre 10.000 y 15.000 filas, específicamente el que se trabaja en este informe tiene 19 columnas y 10785 filas (Ver figura 1), todos los archivos descargados desde Procore. Dos personas dedicaban entre **1 y 2 días laborales** por archivo para dejarlos utilizables. Los errores más frecuentes eran: horómetro final menor a la inicial, horas fuera de rango (antes de 08:00 y después de 20:00), fechas de falla posteriores a retorno, diferencias de horómetro iguales a cero en “EN OPERACIÓN”, doble mantención en el día, y nombres mal escritos que impedían consolidar dashboards en Power BI. Esta situación retrasaba la disponibilidad de información y aumentaba el riesgo de inconsistencias en informes.

Figura 1: Captura de la hoja de cálculo correspondiente al reporte de disponibilidad de equipo.



Equipo	Fecha	Estado	Métrica 1	Métrica 2	Métrica 3	Métrica 4	Métrica 5	Métrica 6	Métrica 7	Métrica 8	Métrica 9	Métrica 10	Métrica 11	Métrica 12	Métrica 13	Métrica 14	Métrica 15	Métrica 16	Métrica 17	Métrica 18	Métrica 19
10774	14-5-20 CAMIONETA																				
10775	14-5-20 CAMIONETA																				
10776	14-5-20 CAMIONETA																				
10777	14-5-20 CAMIONETA																				
10778	14-5-20 CAMIONETA																				
10779	14-5-20 AFOSSE / TWIN 97 / PLUMA																				
10780	14-5-20 AFOSSE / TWIN 97 / PLUMA																				
10781	14-5-20 AFOSSE / TWIN 97 / PLUMA																				
10782	14-5-20 AFOSSE / TWIN 97 / PLUMA																				
10783	14-5-20 AFOSSE / TWIN 97 / PLUMA																				
10784	14-5-20 AFOSSE / TWIN 97 / PLUMA																				
10785	14-5-20 AFOSSE / TWIN 97 / PLUMA																				

Fuente: Elaboración propia

4.2 Reglas de validación implementadas

Se implementó un **validador automático** en Python (pandas) que identifica y etiqueta filas infractoras según reglas operativas del tipo de contrato en Elecon Maquinarias SpA.:

- a) Leer el archivo de datos descargado desde la plataforma
- b) Para cada registro contenido en el archivo, hacer lo siguiente:
 1. Verificar si el horómetro final es menor que el horómetro inicial.
 2. Verificar si la hora de falla ocurre antes de la hora mínima permitida.
 3. Verificar si la hora de retorno excede el rango horario permitido.
 4. Verificar si la fecha de falla es posterior a la fecha de retorno.
 5. Verificar si un equipo presenta estado de operación con diferencia de horómetro igual a cero.
 6. Verificar inconsistencias entre estado operativo y ubicación del equipo.
- c) Si alguna de las condiciones anteriores se cumple:
 1. Registrar el error en un archivo de reporte de inconsistencias.
- d) Si el registro no presenta inconsistencias:
 1. Mantener el registro dentro del conjunto de datos limpio.
- e) Exportar el archivo de errores detectados

La figura 2 presenta un ejemplo de la hoja de cálculo con un error común destacado en rojo, en la cual se observa que el equipo AF0362 no presenta en cambios en el horómetro.

Figura 2: Captura de un reporte con los datos ingresados correctamente.

Fecha	Nombre De Equipo	Horómetro Inicia	Horómetro Fin
8-5-25	AF0481 / TFCV-14 / EXCAVADORA PC210LC 10M0	3.150	3.157
8-5-25	AF0482 / TFCW-28 / EXCAVADORA PC210LC 10M0	3.413	3.420
8-5-25	AF0303 / JKCY-15 / EXCAVADORA PC200 LC -8	9.641	9.648
8-5-25	AF0362 / LKVT-35 / EXCAVADORA PC210 C/Cuarta función	7.830	7.830
8-5-25	AF0364 / LKVT-36 / EXCAVADORA PC210	7.830	7.838

Fuente: Elaboración propia

Respecto a la figura 3 que se subscribe, presenta un ejemplo de una hoja de cálculo descargada desde procore, la cual, tiene destacado en amarillo una fila bien escrita sin errores.

Figura 3 Captura de un reporte con los datos ingresados de forma errónea.

Fecha	Nombre De Equipo	Horón	Horómetro Fin	Hora	Fecha Y Hora	Fecha Y Hora De R	Turno	Estado	Nombre del Proyecto
3-6-25	AF0351 / LFKL-71 / TOLVA	10.482	10.489	7			Día	EN OPERACION	Construcción Canal de Contorno 2 y Defensa Fluvial (ENAMI)
3-6-25	AF0361 / LIWY-30 / TOLVA	8.417	8.424	7			Día	EN OPERACION	Construcción Canal de Contorno 2 y Defensa Fluvial (ENAMI)
3-6-25	AF0201 / DHVZ-23 / TOLVA	476	481	5	03/06/25 08:00	03/06/25 18:00	Día	MANTENCIÓN CORRECTIVA	Construcción Canal de Contorno 2 y Defensa Fluvial (ENAMI)
3-6-25	900437 / TKJW-43 / ALIBE	1.855	1.859	4			Día	EN OPERACION	Construcción Canal de Contorno 2 y Defensa Fluvial (ENAMI)
3-6-25	AF0481 / TFCV-14 /	3.259	3.264	5			Día	EN OPERACION	Construcción Canal de Contorno 2 y Defensa Fluvial (ENAMI)

Fuente: Elaboración propia

4.3 Comparación de tiempos de procesamiento.

La automatización reduce el tiempo de 1–2 días o jornadas laborales a segundos/minutos como lo muestra la tabla 1, dependiendo del tamaño del archivo. Se presenta la comparación para dos escenarios típicos. Los tiempos demorados fueron tomados con cronometro del celular y fue probado en distintos computadores, entregando distintos tiempos y se selecciona el menor para el resultado

Tabla 1: Tabla comparación tiempo antes y después de la implementación.

Tipo de archivo	Proceso	Manual	Bot	Ahorro
Liviano (5400 FILAS)	Descarga Limpieza	+ 9h	30 s	-99%
Pesado (10785 FILAS) Utilizado en este informe	Descarga Limpieza	+ 18h	87s	-99%

Fuente: Elaboración propia

4.4 Errores detectados y corregidos

El validador genera una **hoja de cálculo de errores** con todas las filas infractoras que en este archivo fueron 413 filas (ver Figura 4), etiquetadas por tipo de error, permitiendo trazabilidad.

Figura 4: Hoja de cálculo generado por el script con errores detectados.

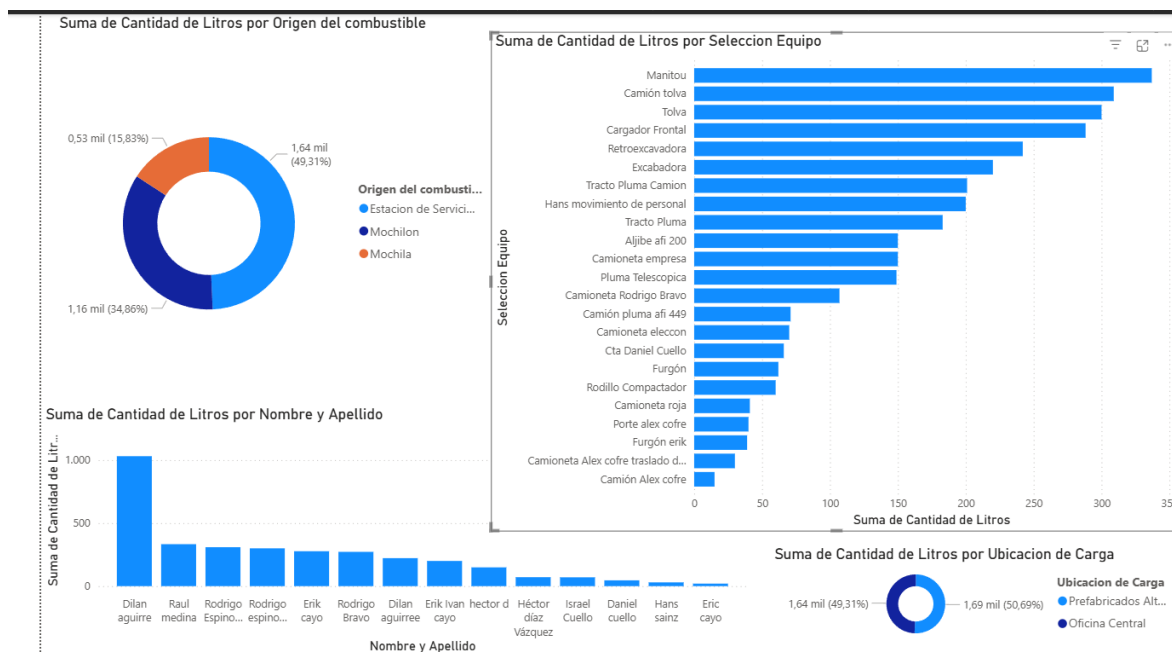
	Nombre De Equipo	Km (Camión)	Km (Camión)	Operación	Turno	Estado	bre del Projeo del Pro	Fuera de S	mpo Nomi	mpo Dispon	check duplica	de no ope	Error	Falla_dt	Retorno_dt
D25	AF0381 / LPPH-63 / RODILLO	4618	4618	0	Día	EN OPERAC	Construcci	45000240	0	12	10 OK	Interno	En operación con horómetro sin cambio		
D25	AF0362 / LKVT-35 / EXCAVADORA PC210 C	7830	7830	0	Día	EN OPERAC	Construcci	45000240	0	12	10 OK		En operación con horómetro sin cambio		
D25	900031 / RLTC-96 / CAMIONETA	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	900444 / TKKP-77 / CAMIONETA	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	900448 / TKKP-76 / CAMIONETA	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	900440 / TKKR-19 / CAMIONETA	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	900163 / SHKW-91 / CAMIONETA	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	AF0472 / TBWV-39 / PETROLERO	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	900435 / KKYH-80 / MINI BUS MERCEDES BE	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	900434 / RSYF-63 / MINI BUS FORD	0	0	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		
D25	AF0366 / LKVT-34 / EXCAVADORA PC210	10750	10750	0	Día	EN OPERAC	Servicio de	45000660	0	12	10 OK		En operación con horómetro sin cambio		

Fuente: Elaboración propia

4.5 Integración con Power BI

La Figura 5 muestra el estado inicial del dashboard generado a partir de los archivos operacionales descargados desde procore. En esta etapa, los datos aún no habían sido sometidos a un proceso de validación ni limpieza automatizada, por lo que las visualizaciones reflejaban directamente las inconsistencias presentes en los registros originales. Entre los problemas de mayor recurrencia se observaba: los nombres de equipos, registros duplicados y diferencias en los formatos de ingreso de algunos datos operacionales. Estas inconsistencias provocaban distorsiones en los gráficos y perjudican la correcta interpretación de los indicadores presentados en el dashboard. Considerando que los archivos analizados pueden contener más de diez mil registros, como se menciona en el capítulo 4.1; la identificación manual de estos errores resultaba compleja y demandaba una revisión extensa de la información, situación que, dejó en evidencia la necesidad de implementar un proceso automatizado que permitiera detectar y pormenorizar estas inconsistencias antes de hacer uso de los datos en la etapa de análisis.

Figura 5: Estado Inicial de dashboard.



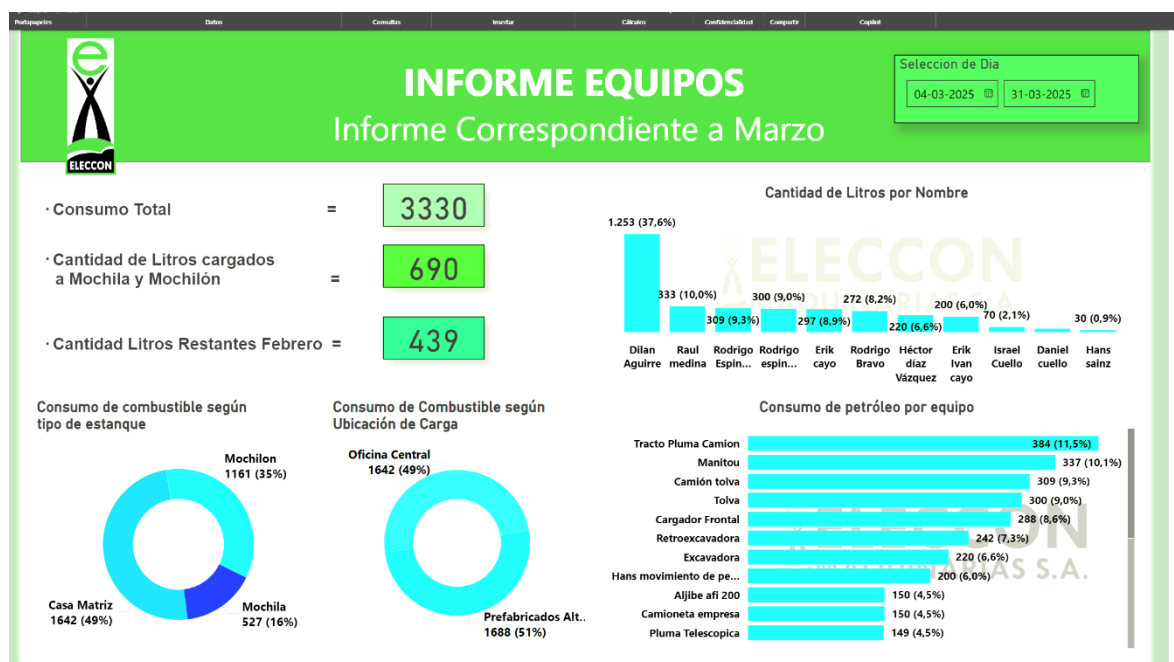
Fuente: Elaboración propia

Por otra parte, la Figura 6 presenta el dashboard generado posteriormente a la aplicación del proceso automatizado de depuración y limpieza de datos, etapa en la cual, los archivos operacionales son procesados mediante el script desarrollado, el cual, revisa automáticamente los registros contenidos en la hoja de cálculo descargados, donde se aplica una serie de reglas para detectar inconsistencias en la información.

La relevancia de lo antes mencionado es que este script permite analizar miles de registros en segundos (15 segundos aproximadamente), identificando errores como duplicidad de equipos, inconsistencias en nombres y registros con formatos incorrectos. Posteriormente, los registros que presentan fallas son extraídos automáticamente y almacenados en un archivo independiente, permitiendo al analista revisar únicamente las filas que contiene los errores identificados.

Finalmente, y como resultado de este proceso, se genera un conjunto de datos limpios y estandarizados que pueden ser utilizados directamente en la herramienta de visualización. Esto permite obtener un dashboard sin duplicidades ni inconsistencias, mejorando la claridad de la información y reduciendo significativamente el tiempo de preparación de los datos.

Figura 6: Dashboard sin errores en power-bi.



Fuente: Elaboración propia

4.6 Impacto Organizacional y validación interna

La Tabla 2 muestra el impacto organizacional generado tras la implementación del proceso automatizado de validación y limpieza de datos. Antes de la automatización, el análisis de los reportes requería la participación de dos personas y un tiempo aproximado de hasta 16 horas por archivo, además de, presentar un alto nivel de reproceso debido a errores en los registros. Posteriormente, con la utilización del sistema automatizado, el proceso puede realizarse en aproximadamente 1,5 minutos por archivo y con la participación de una sola persona, lo que se traduce como una reducción significativa en los tiempos de procesamiento, una disminución del reproceso por errores y una mejora en la confiabilidad de los dashboards generados para el análisis operativo.

Tabla 2: Tabla de impacto en tiempo luego de la implementación.

Indicador	Antes	Después
Personas dedicadas	2	1
Tiempo por archivo	16 h	1.5 min
Reproceso por errores	Alto	Mínimo
Confiabilidad de dashboards	Baja/Variable	Alta/Estable

Fuente: Elaboración propia

Respecto de la validación interna del script, se aplica una prueba piloto a 3 colaboradores de Elecon Maquinarias SpA, pertenecientes a distintos cargos de la organización: (1) un representante de la línea de gerencia, (2) uno de la línea de supervisión y (3) uno de la línea de proyectos. Esta selección intencionada busca asegurar que el instrumento fuera comprensible y pertinente para los diferentes perfiles de usuario que interactúan con los reportes automatizados.

El instrumento estuvo compuesto por seis preguntas estructuradas, orientadas a medir el impacto percibido y la usabilidad de la solución automatizada, y una séptima pregunta de carácter abierta destinada a recoger comentarios y sugerencias de mejora. Durante la aplicación del piloto, se constata que ninguno de los tres participantes generó consultas ni requirió aclaraciones adicionales para responder; lo que evidencia que las consultas estaban correctamente formuladas y alineadas con el propósito de la medición. Asimismo, no se registraron dudas en la interpretación de los ítems, lo que permite validar la claridad y pertinencia del instrumento antes de su aplicación definitiva.

Por otra parte, y con el fin de completar la validación técnica del sistema, se aplica una encuesta de percepción a tres colaboradores de Elecon Maquinarias SpA. que habían utilizado el pipeline al menos una vez. Los encuestados pertenecen a distintos niveles de la organización: dos de ellos forman parte de la Oficina Técnica o PMO, con uso diario y mensual respectivamente, mientras que el tercero pertenece al área de Operaciones y Terreno con uso diario. Esta diversidad de perfiles permitió obtener una visión transversal sobre el impacto percibido de la solución.

Los resultados muestran un nivel de satisfacción consistentemente alto. En la mayoría de los ítems evaluados, los tres encuestados respondieron "Muy de acuerdo", con dos excepciones puntuales: el encuestado del área de Operaciones valoró la mejora en tiempos y la facilidad de uso como "De acuerdo" en lugar de "Muy de acuerdo", lo que sugiere que desde el terreno la percepción de mejora existe pero es levemente más moderada que desde la PMO. Por su parte, el encuestado mensual evaluó la estabilidad de la solución también con "De acuerdo", lo que podría explicarse por una menor frecuencia de uso y, por tanto, menos oportunidades de observar el comportamiento del sistema en distintas condiciones. En cuanto a la disposición a recomendar la solución, los tres encuestados asignaron un puntaje de 9 sobre 10, lo que en términos de NPS refleja una percepción ampliamente positiva y una alta probabilidad de adopción por parte de otros usuarios.

En cuanto a los comentarios cualitativos entregaron antecedentes clave para la proyección de la solución, destacando la necesidad de contar con material de apoyo – instructivos, videos o simulaciones - que permita su uso autónomo y asegure la transferencia de conocimiento. Si bien la herramienta presenta un buen funcionamiento, su sostenibilidad depende de completar lo técnico con una estrategia formativa. Asimismo, se propuso incorporar validaciones en el origen de los datos para mejorar su calidad desde el ingreso.

Finalmente, la mención espontánea de ser aplicado en nuevas áreas, tales como: Casino, Alimentación y otros módulos de reportes 360, confirma la escalabilidad de la solución tanto técnica como funcionalmente.

A continuación, se muestra la estructura que conforma la encuesta precedentemente, como sigue:

Figura 7: Portada y descripción de la encuesta.

Evaluar el impacto de la automatización en "Eleccon Maquinarias"

El objetivo de la siguiente encuesta es medir el impacto de la automatización (descarga, validación y limpieza) en rapidez, calidad de datos, usabilidad y estabilidad, para mejorar los reportes operacionales y Power BI

Descripción de evaluación

Este formulario busca medir la percepción sobre la automatización de reportes en ELECCON (descarga, validación y limpieza). Se presentan afirmaciones que debes calificar según tu experiencia. La encuesta es anónima y toma 3–5 minutos.

Criterios que se evaluarán

- Rapidez: reducción del tiempo de preparación de datos frente al proceso manual.
- Facilidad de uso: simplicidad del flujo y de los archivos generados.
- Calidad de datos: consistencia, estandarización y menor presencia de errores.
- Auditoría: utilidad del archivo "errores_detectados.xlsx" para identificar y corregir problemas.
- Power BI: funcionamiento sin quiebres (medidas/segmentaciones) tras la automatización.
- Valor agregado: tiempo liberado para tareas de mayor valor.
- Estabilidad: frecuencia con que el proceso falla o requiere intervención.
- Satisfacción general: evaluación global de la solución.

Escala de evaluación (Likert 1–5)

1. Muy en desacuerdo / Inaceptable: no cumple con el criterio.
2. En desacuerdo / Deficiente: cumple mínimamente; presenta múltiples problemas.
3. Ni de acuerdo ni en desacuerdo / Aceptable: cumple lo básico; puede mejorar.
4. De acuerdo / Bueno: buen desempeño con oportunidades de mejora.
5. Muy de acuerdo / Excelente: desempeño destacado y consistente.

Pregunta de recomendación (NPS): Se incluye una pregunta de 0 a 10: **0 = "nada probable"**, **10 = "muy probable"**.

Instrucciones: Responde según tu experiencia real en el proyecto. Si NO has utilizado los reportes del bot, indícalo y continúa a la sección de comentarios. No se registran datos personales; los resultados se usarán para mejora interna y para la Memoria de Título.

Fuente: Elaboración propia

Figura 8: Sección 1: Perfil del encuestado – P1 Área o rol.

Perfil del encuestado

1. Area o rol en el que trabaja.

☐ Operaciones / Terreno

☐ Oficina Técnica / PMO

☐ Administración

☐ Otras

Fuente: Elaboración propia

Figura 9: Sección 1: Perfil del encuestado – P2 Frecuencia de uso.

2. Frecuencia del uso de reportes

☐ Diario

☐ Semanal

☐ Mensual

☐ Esporádico

Fuente: Elaboración propia

Figura 10:— Sección 1: Perfil del encuestado – P3 Uso de reportes del script.

3. ¿Ha usado reportes generados por el bot (descarga + validación/limpieza) al menos una vez?

☐ Sí

☐ No

☐ No lo sé

Fuente: Elaboración propia

Figura 11: Sección 2: Percepción – Escala Likert (filas 1 a 4)

4. Escala de opinion

	Muy en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Muy de acuerdo
El tiempo de preparación de datos mejoró con la automatización.	☐	☐	☐	☐	☐
El proceso automatizado es fácil de usar.	☐	☐	☐	☐	☐
Los datos son más consistentes y con menos errores.	☐	☐	☐	☐	☐
El archivo "errores_detectados.xlsx" facilita identificar y corregir problemas.	☐	☐	☐	☐	☐
Los dashboards en Power BI funcionan sin quiebres (medidas/segmentaciones).	☐	☐	☐	☐	☐
Puedo dedicar más tiempo a tareas de mayor valor.	☐	☐	☐	☐	☐
La solución es estable (rara vez falla).	☐	☐	☐	☐	☐
Satisfacción general con la solución.	☐	☐	☐	☐	☐

Fuente: Elaboración propia

Figura 12: Sección 3: Recomendación (NPS 0–10)

Recomendacion

5. ¿Qué tan probable es que recomiende esta solución a un colega? (0 = Nada probable, 10 = Muy probable)

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Fuente: Elaboración propia

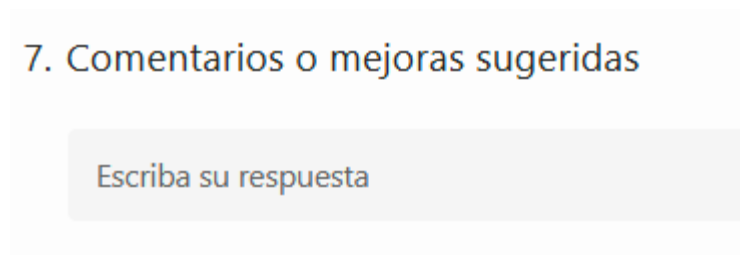
Figura 13: Sección 4: Próximos pasos – Áreas a automatizar

6. ¿Qué otras áreas le gustaría automatizar?

- ☐ Asistencia
- ☐ GPS / Telemática
- ☐ Casino / Alimentación
- ☐ Combustible
- ☐ Reportes 360 (otros)

Fuente: Elaboración propia

Figura 14: Sección 4: Comentarios o mejoras sugeridas.



7. Comentarios o mejoras sugeridas

Escriba su respuesta

Fuente: Elaboración propia

4.7 Escalabilidad y extensión de la solución

La arquitectura demostró ser replicable a procesos con estructura similar (descarga masiva + limpieza + estandarización), por ejemplo: asistencia, GPS, casino, combustible. La propuesta considera que cada nuevo proyecto inicie con un script asociado desde el día 1, asegurando datos listos para análisis sin carga manual.

5 Discusión

El punto de partida de esta memoria no fue una revisión bibliográfica ni un problema identificado desde la teoría, sino algo mucho más concreto: trabajar directamente con los datos. Durante la práctica profesional, la tarea diaria de revisar hojas de cálculo y armar dashboards deja en evidencia algo que cualquier persona que haya pasado por ese proceso reconocería de inmediato – demasiado tiempo invertido en ordenar información que llegaba mal desde el origen. Esa observación, que un momento parecía sol una incomodidad operativa, termina siendo el punto de partida del desarrollo del presente trabajo.

Los resultados muestran que la solución funciona. El Pipeline logra reducir el tiempo de procesamiento de manera significativa y mejora la calidad de la información con la que trabaja el equipo. Sin embargo, desarrollar e implementar el sistema también deja una conclusión que no estaba en los objetivos originales, pero vale la pena mencionar; El pipeline limpia bien lo que llega mal, pero no puede impedir que llegue mal.

El validador detecta errores y los clasifica con trazabilidad, pero lo hace después de que ya fueron cometidos. El problema de fondo – el error humano en el momento de ingresar el registro – sigue ocurriendo en Procore, antes de que cualquier script entre en acción. En ese sentido, la automatización resuelve el procesamiento, peor no ataca la causa raíz.

Pensando en eso, una mejora concreta que complementaría lo desarrollado sería intervenir directamente en el formulario de ingreso de datos. Si el sistema no permitiera guardar un horómetro final menor al inicial, o si el estado del equipo solo pudiera seleccionarse desde una lista predefinida, una parte importante de ellos errores detectados por el validador simplemente no existiría. Eso no reemplaza el pipeline, sino que lo hace más eficiente porque le llega información de mejor calidad desde el inicio.

Otra mejora posible, más simple de implementar, sería retroalimentar periódicamente los operadores con los reportes de errores generados por el sistema. Mostrarles qué reglas se rompen con más frecuencia y en qué frentes de trabajo ocurre, convierte al validador en algo más que una herramienta de limpieza: lo transforma en un instrumento de aprendizaje para

el equipo. Con el tiempo, esa retroalimentación debería reducir naturalmente la tasa de error en el ingreso.

En definitiva, lo que comenzó como una propuesta informal en un almuerzo termina siendo una solución que la empresa adoptó y que demostró resultados concretos. Pero también dejó claro que la automatización tiene un límite: puede hacer mucho con los datos malos, pero haría mucho más con datos buenos. Esa es, probablemente, la principal lección que deja este trabajo para quien quiera seguir desarrollando la solución

5.1 Escalabilidad y replicabilidad

Debido a que Elecon Maquinarias SpA. utiliza diversas plataformas digitales en diferentes unidades operativas, la solución diseñada es completamente escalable. El modelo de scripts puede adaptarse fácilmente para:

- Automatizar la descarga de otros tipos de reportes.
- Validar datos en áreas como recursos humanos, gestión de flota o seguridad.
- Replicarse al iniciar nuevos contratos o proyectos, reduciendo la curva de entrada operativa.

Esta solución, además de resolver un problema puntual, sienta las bases para un enfoque más automatizado y eficiente en el manejo de datos masivos dentro de la organización.

Aunque el caso de Elecon Maquinarias SpA. es particular por su uso de aplicaciones específicas, existen estudios similares que abordan la automatización de procesos administrativos mediante herramientas de ciencia de datos. Por ejemplo, el trabajo de Ganesan (2024) detalla cómo crear aplicaciones de dashboard en Python para monitorear indicadores de negocio, incluyendo paneles de ventas, asistencia y cumplimiento de metas.

En Elecon Maquinarias SpA., se evalúa que la implementación de dashboards para monitorear los GPS de la flota de vehículos es una herramienta eficiente para identificar ineficiencias logísticas y optimizar el uso de los recursos. De acuerdo con Geotab (2025), la

supervisión GPS y el análisis de datos en tiempo real: mejoran la seguridad y la eficiencia de las flotas, permitiendo la optimización de rutas, la prevención de riesgos y la mejora del desempeño operativo de la empresa. Gracias al uso de Big Data en logística, es posible acceder a reportes de flota en tiempo real ayudando a optimizar el mantenimiento de activos, lo que se significa que habrá una mayor seguridad de los datos y reducción de costos para las flotillas.

Además, Elecon Maquinarias SpA. utiliza un conjunto diverso de plataformas digitales en sus distintos frentes de trabajo, por lo que la creación de scripts reutilizables representa una herramienta estratégica para estandarizar procesos desde el inicio de cada nuevo proyecto. Esta escalabilidad permite adaptar rápidamente la automatización a diferentes equipos, ubicaciones o tipos de reporte, entregando una solución flexible y de alto impacto para la gestión operacional.

5.2 Interpretación de los hallazgos

Primero, la comparación de tiempos (Tablas 4.2) evidencia un salto de 1–2 días a segundos/minutos por archivo, lo que impacta directamente la oportunidad de información para jefaturas. Segundo, la distribución de errores (Tabla 4.3) confirma que las reglas abordan los focos críticos: horómetro final < inicial, ventanas de tiempo inválidas y casos de “EN OPERACIÓN” con $\Delta=0$, evitando la propagación de inconsistencias silenciosas hacia el BI. Tercero, el archivo errores_detectados.xlsx introduce trazabilidad por fila y por regla, habilitando auditoría y aprendizaje operacional.

5.3 Relación con los objetivos

Los hallazgos están alineados con los objetivos específicos: (i) automatizar la descarga (script 1), (ii) validar/limpiar y estandarizar (script 2), (iii) generar datasets listos para BI y (iv) reducir drásticamente la intervención manual. Asimismo, aportan a la escalabilidad, pues el patrón “descarga + validación + estandarización” es replicable a asistencia, GPS, casino y combustible.

5.4 Fortalezas

Entre las principales fortalezas de la solución presentada se destaca: en primera instancia, la mejora que destaca se relaciona con la eficiencia operativa del proceso de gestión de datos. Adicionalmente, la automatización del pipeline permitió reducir notablemente el tiempo requerido para la preparación de archivos, generando así, un ahorro sostenido de horas humanas, y sumado a ello, el tiempo de ciclo necesario para disponer de información requeridas para estos efectos.

Por otra parte, cabe señalar que, la implementación de reglas para la validación contribuyó a mejorar la calidad y fidelización de los datos operacionales obtenidos, reduciendo así, la variabilidad generada por: errores de digitación, inconsistencias en los registros y diferencias en los formatos de los archivos.

Otra fortaleza que destaca corresponde a la trazabilidad incorporada en el proceso de validación, puesto que, el sistema genera automáticamente un registro de los errores significativos, los cuales, eran clasificados según el tipo de regla infringida y el registro específico donde se producía el error, permitiendo así, facilitar la auditoría de los datos e identificar patrones de error que pueden ser abordados más tarde desde la línea operativo de Eleccon Maquinarias SpA.

Por último, la arquitectura de la solución se destaca por grado de replicabilidad, puesto que, el diseño del pipeline automatizado permite parametrizar su funcionamiento para distintos proyectos o tipos de reportes.

5.5 Limitaciones

- Cambios en la UI de la plataforma pueden afectar la descarga automatizada (Selenium).
- Cobertura de reglas: la detección se limita a lo normado; anomalías no previstas pueden pasar.
- Normalización de nombres: requiere mantener diccionarios y revisión de outliers.
- Gestión de credenciales: demandas políticas formales para ambientes productivos.

5.6 Amenazas a la validez y mitigaciones

- Muestra limitada de archivos/proyectos en las pruebas iniciales → ampliar la base de validación.
- Medición parcial del impacto (foco en tiempos) → incorporar lead time de decisión y reprocesos evitados en BI.
- Falsos positivos/negativos por umbrales → pruebas unitarias por regla y revisión mensual del catálogo.

5.7 Implicancias y generalización

La evidencia sugiere una adopción institucional en modalidad “script por proyecto”, iniciando cada contrato con un pipeline preconfigurado. Este enfoque promueve gobernanza de datos (reglas versionadas, diccionarios vivos) y monitoreo continuo (tiempos, % filas limpias, top reglas infringidas), facilitando la extensión hacia otras áreas con mínima fricción técnica.

6 Conclusiones

A partir de los resultados obtenidos y de la validación práctica en Elecon Maquinarias SpA., es posible sintetizar el impacto de la automatización propuesta en términos de eficiencia, calidad de datos y escalabilidad. En esta sección se presentan las conclusiones más relevantes del estudio, destacando cómo el pipeline de descarga, validación y limpieza permitió mejorar la oportunidad y confiabilidad de la información operativa, y qué aprendizajes dejan estos hallazgos para la gestión y la expansión de la solución en futuros proyectos.

1. Eficiencia operacional: La automatización redujo el tiempo de preparación de datos desde 1–2 días por archivo a segundos/minutos, habilitando información oportuna para la toma de decisiones y liberando carga administrativa significativa.
2. Calidad y trazabilidad: El validador automático basado en reglas de negocio (horómetros, ventanas horarias, coherencia temporal, estados/mantenciones) eliminó errores críticos y aportó trazabilidad mediante el archivo errores_detectados.xlsx.
3. Estabilidad analítica en BI: La estandarización de columnas, formatos y nombres permitió que los dashboards de Power BI funcionen sin quiebres (medidas y segmentaciones), fortaleciendo la confiabilidad del monitoreo operativo.
4. Cumplimiento de objetivos: Se cumplieron los objetivos específicos: descarga automatizada, validación/limpieza sistemática y generación de datasets listos para BI, con reducción drástica de intervención manual.
5. Alineación con usuarios: La encuesta a usuarios evidenció [alta/media] satisfacción en rapidez, calidad y facilidad de uso, además de insumos concretos para priorizar mejoras.
6. Escalabilidad: La arquitectura es replicable a procesos de patrón similar (asistencia, GPS, casino, combustible) y puede institucionalizarse como práctica base en cada proyecto nuevo (“script por proyecto”).

7 Recomendaciones

7.1 Operación y monitoreo

- Programar el pipeline (descarga → validación → limpieza → exportación) como job recurrente (Task Scheduler/CI) con logs de tiempo y conteos por regla.
- Implementar alertas automáticas (correo/Teams) ante umbrales críticos (p. ej., % de filas con error > X% o activación de reglas críticas).
- Ejecutar Selenium en modo headless y usar selectores resilientes con fallback ante cambios de la UI.

7.2 Gobierno y calidad de datos

- Mantener un catálogo versionado de reglas y un diccionario vivo de estandarización (equipos/operadores), con revisión mensual de outliers.
- Incorporar pruebas unitarias por regla y una muestra de regresión de archivos para asegurar no-recaídas.
- Gestionar credenciales con variables de entorno o secret manager corporativo.

7.3 Adopción y soporte

- Elaborar un manual breve (1–2 páginas) para interpretar errores_detectados.xlsx y el dataset limpio.
- Realizar micro-capacitaciones a usuarios clave (PM/BI/Operaciones) y establecer un canal de soporte para incidencias y sugerencias.

7.4 Escalabilidad

- Formalizar una plantilla de “script por proyecto” (parámetros, rutas, reglas mínimas) que se active desde el inicio de cada contrato.
- Priorizar la extensión a asistencia, control de petróleo, GPS y casino reutilizando el mismo patrón técnico y de gobierno.

8 Trabajos futuros

1. Corrección automática: además de detectar, corregir ciertos errores de forma automática (normalización de nombres con diccionarios/reglas y bitácora de cambios).
2. Detección de anomalías (ML): complementar las reglas con modelos de anomalía para capturar casos no contemplados explícitamente.
3. API interna y catálogo: exponer datasets limpios vía API y registrar un catálogo de versiones para consumo estándar por BI u otros sistemas.
4. Panel de monitoreo del pipeline: dashboard con tiempos de ejecución, % de filas limpias, top reglas infringidas y evolución semanal/mensual.
5. Evaluación de impacto ampliada: medir efectos en lead time de decisión, reprocesos evitados en BI y horas-hombre liberadas a nivel mensual/trimestral.
6. Pruebas A/B de reglas: iterar umbrales y criterios con A/B testing para optimizar sensibilidad/especificidad y reducir falsos positivos/negativos.

9 Referencias

Smagul, D., & Daineko, Y. (2024). Robotic process automation: A comprehensive systematic literature review across industries, with a focus on healthcare. CEUR Workshop Proceedings, 3966. <https://ceur-ws.org/Vol-3966/W2Paper2.pdf>

Alfandi, M., & Seçkiner, S. U. (2022). Robotic process automation: A literature review on quantitative benefits. En Proceedings of the International Conference on Industrial Engineering and Operations Management (pp. [verificar]). IEOM Society International. <https://ieomsociety.org/proceedings/2022istanbul/846.pdf>

Ivančić, L., Vugec, D. S., & Vukšić, V. B. (2019). Robotic process automation: Systematic literature review. En C. Di Ciccio et al. (Eds.), Business Process Management: Blockchain and Central and Eastern Europe Forum (Lecture Notes in Business Information Processing, Vol. 361, pp. 280–295). Springer. https://doi.org/10.1007/978-3-030-30429-4_19

PwC. (2018). Is China ready for intelligent automation? PwC China. <https://www.pwccn.com/en/issues/technology/emerging-technologies/is-china-ready-for-intelligent-automation.pdf>

McKinney, W. (2022). Python for data analysis (3.^a ed.). O'Reilly Media. <https://wesmckinney.com/book/>

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin, 23(4), 3–13. <https://dbis.uni-leipzig.de/files/research/publications/2000-1/pdf/TBDE2000.pdf>

Ganesan, C. (2024). Building a dashboard app in Python using Streamlit. Streamlit. <https://blog.streamlit.io/crafting-a-dashboard-app-in-python-using-streamlit/>

Geotab. (2025). Monitoreo GPS y tecnología avanzada para flotas más seguras y eficientes. Geotab. <https://www.geotab.com/es-latam/blog/como-beneficia-gps-tracking-seguridad-flotas/>

Automation Anywhere. (s. f.). Robotic process automation (RPA).

<https://www.automationanywhere.com/la/rpa/robotic-process-automation>

10 ANEXOS

10.1 Anexo 1: Automatización de descarga desde Procore (Selenium)

Figura A 1: Librerías e imports de Selenium.

```
1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  from selenium.webdriver.chrome.service import Service
4  from selenium.webdriver.support.ui import WebDriverWait
5  from selenium.webdriver.support import expected_conditions as EC
6  import time
```

Fuente: Elaboración propia

Figura A 2: Configuración del driver y espera explícita.

```
ruta_driver = "C:/Selenium/chromedriver-win64/chromedriver.exe"
servicio = Service(ruta_driver)
driver = webdriver.Chrome(service=servicio)
wait = WebDriverWait(driver, 20)
```

Fuente: Elaboración propia

Figura A 3: Inicio de sesión en Procore (email/clave y botón "login")

```
driver.get("https://login.procore.com")
wait.until(EC.presence_of_element_located((By.ID, "session_email"))).send_keys("luciano.yanez@eleccon.cl")
wait.until(EC.presence_of_element_located((By.ID, "session_password"))).send_keys("XXXXXX.")
wait.until(EC.element_to_be_clickable((By.ID, "login-btn"))).click()
time.sleep(5)
```

Fuente: Elaboración propia

Figura A 4: Selección del proyecto desde el "picker"

```
selector_proyectos = wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, 'div[data-header="picker-company-project"]')))
selector_proyectos.click()
wait.until(EC.element_to_be_clickable(
    (By.LINK_TEXT, "4500066042 - Servicio de Construcción Muro de Arena (MANTO VERDE)"))
).click()
time.sleep(5)
```

Fuente: Elaboración propia

Figura A 5: Apertura del panel de herramientas.

```
boton_herramientas = wait.until(EC.element_to_be_clickable((By.ID, "tool-picker-target")))
driver.execute_script("arguments[0].scrollIntoView();", boton_herramientas)
boton_herramientas.click()
```

Fuente: Elaboración propia

Figura A 6: Ingreso a “Reportes 360

```
reportes_360 = wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, 'a[data-header="tools-menu-reports"]')))
reportes_360.click()
time.sleep(8)
```

Fuente: Elaboración propia

Figura A 7: Selección del reporte “Reporte Equipos (Nuevo)”

```
reporte = wait.until(EC.presence_of_element_located((By.XPATH, '//div[text()="Reporte Equipos (Nuevo)"]')))
driver.execute_script("arguments[0].scrollIntoView(true);", reporte)
wait.until(EC.visibility_of(reporte))
driver.execute_script("arguments[0].click();", reporte)
```

Fuente: Elaboración propia

Figura A 8: Exportación a Excel del reporte.

```
# Paso 6: Esperar que se abra el reporte y hacer clic en el botón de exportar
time.sleep(5)
boton_exportar = wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, 'div.dropdown-toggle.export')))
boton_exportar.click()

# Paso 7: Hacer clic en "Excel" del menú desplegable
excel_btn = wait.until(EC.presence_of_element_located((By.XPATH, '//button[normalize-space()="Excel"]')))
wait.until(EC.element_to_be_clickable((By.XPATH, '//button[normalize-space()="Excel"]')))
driver.execute_script("arguments[0].click();", excel_btn)
time.sleep(10)
```

Fuente: Elaboración propia

10.2 Anexo 2 Validador y reporte de errores (pandas)

Figura A 9: Importación y carga del archivo

```
import pandas as pd

# Cargar archivo
archivo = r"BASE DE DATOS REPORTE UTILIZACION DISPONIBILIDAD VF.xlsx"
df = pd.read_excel(archivo)
```

Fuente: Elaboración propia

Figura A 10: Normalización de columnas y contenedor de errores

```
df.columns = df.columns.str.strip()
errores = []
```

Fuente: Elaboración propia

Figura A 11: Regla 1: Horómetro final < inicial

```
filtro1 = df["Horómetro 0 Km (Camionetas) Final"] < df["Horómetro 0 Km (Camionetas) Inicial"]
errores.append(df[filtro1].assign(Error="Horómetro final menor al inicial"))
```

Fuente: Elaboración propia

Figura A 12: Regla 2: Conversión a datetime y falla antes de 08:00

```
# 2. La hora de falla no debe ser antes de las 8:00 am
# Convertir a datetime
df["Falla_dt"] = pd.to_datetime(df["Fecha Y Hora De Falla (Preventiva O Correctiva)"], errors="coerce", dayfirst=True)
filtro2 = df["Falla_dt"].dt.hour < 8
errores.append(df[filtro2].assign(Error="Falla antes de las 8:00 am"))
```

Fuente: Elaboración propia

Figura A 13: Regla 3: Retorno después de 20:00

```
# 3. La hora de retorno no debe ser después de las 20:00
df["Retorno_dt"] = pd.to_datetime(df["Fecha Y Hora De Retorno (Preventiva O Correctiva)"], errors="coerce", dayfirst=True)
filtro3 = df["Retorno_dt"].dt.hour > 20
errores.append(df[filtro3].assign(Error="Retorno después de las 20:00"))
```

Fuente: Elaboración propia

Figura A 14: Regla 4: Falla posterior al retorno

```
# 4. La fecha/hora de falla no puede ser posterior al retorno
filtro4 = df["Falla_dt"] > df["Retorno_dt"]
errores.append(df[filtro4].assign(Error="Falla posterior a retorno"))
```

Fuente: Elaboración propia

Figura A 15: Regla 4: Falla posterior al retorno

```
# 5. No puede estar en operación si horómetros son iguales
filtro5 = (df["Estado"] == "EN OPERACION") & (
    df["Horómetro 0 Km (Camionetas) Final"] - df["Horómetro 0 Km (Camionetas) Inicial"] == 0)
errores.append(df[filtro5].assign(Error="En operación con horómetro sin cambio"))
```

Fuente: Elaboración propia

Figura A 16: Consolidación de hallazgos y export del reporte de errores

```
# Unir todos los errores
df_errores = pd.concat(errores)
# Guardar como Excel para revisión
df_errores.to_excel("errores_detectados.xlsx", index=False)
print("✅ Validaciones realizadas. Archivo generado: errores_detectados.xlsx")
import os
```

Fuente: Elaboración propia

Figura A 17: Generación de nombre/ruta de salida y export

```
# Crear nombre del nuevo archivo con mismo nombre pero anteponiendo "nuevo_"
nombre_archivo_original = os.path.basename(archivo)
nuevo_nombre = "nuevo_" + nombre_archivo_original

# Ruta final del archivo limpio
ruta_salida = os.path.join(r"C:\Users\Luciano\Desktop\Automatizaciones\Limpios", nuevo_nombre)
```

Fuente: Elaboración propia

Figura A 18: Ruta final del archivo limpio y guardado

```
ruta_salida = os.path.join(r"C:\Users\Luciano\Desktop\Automatizaciones\Limpios", nuevo_nombre)
df_errores.to_excel(ruta_salida, index=False)
```

Fuente: Elaboración propia