

Automatización Robótica de Procesos (RPA) para el seguimiento analítico de activos y generación de reportes de excedentes en clientes AGPE de la Electrificadora de Santander (ESSA)

Fabian Ernesto Hernández Ramírez

Brayan Arley Rojas Mayorga

Trabajo de grado para optar por el título de Ingeniero de Sistemas

Director

Carlos Adolfo Beltrán Castro

MSc. Ingeniería de Sistemas e Informática

Tutor

Román Alexis Suárez Cáliz

MSc. Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

Facultad de ingenierías fisicomecánicas

Escuela de ingeniería de sistemas e informática

Bucaramanga

2025

Dedicatoria

A todas las personas de la Universidad Industrial de Santander que, en el desarrollo de sus prácticas académicas, utilicen tecnologías como Python o plataformas de bajo código de Microsoft, que puedan beneficiarse de los aprendizajes y resultados de este proyecto.

Agradecimientos

Quiero agradecer profundamente a mi mamá, quien fue la única persona que me apoyó incondicionalmente durante toda mi carrera universitaria, incluso cuando yo mismo perdía la fe en mí. A mi hermana, que me ha ayudado a ver los errores que he cometido y que puedo ser una mejor persona. Agradezco también a mis amigos, quienes fueron parte fundamental de este proceso y sin los cuales probablemente me habría quedado atascado a mitad del camino. Al BPG y Monkey Swag, que han sido mis mejores amistades a lo largo de la vida.

Y, por último, me agradezco a mí mismo, porque reconozco que sin el esfuerzo que puse en las últimas etapas, no estaría donde estoy ahora.

Fabian Ernesto Hernández Ramírez.

Agradezco a Dios por permitirme estar donde estoy, por nunca abandonarme, por darme fortaleza y siempre ser tan bueno conmigo.

De todo corazón, le agradezco por haberme dado a mis padres.

Gracias, Padre y Madre, por brindarme aquello que quizá no tuvieron: una vida llena de comodidades, apoyo incondicional, tiempo, comprensión y, sobre todo, mucho amor y motivación para seguir adelante. Su paciencia, consejos, enseñanzas y todo el sacrificio que realizan para verme sonreír son invaluable.

Simplemente, las palabras no me alcanzan para expresar mi gratitud.

También agradezco a mi hermano, quien, a pesar de estar tan lejos, siempre está pendiente y me ha motivado con sus palabras y forma de ser a vivir de verdad un poco más.

Sin él, no sería quién soy.

Gracias a ustedes por confiar y estar conmigo.

A mi familia, que siempre ha sido incondicional; hemos compartido muchas experiencias que también han forjado quién soy ahora. A todos mis compañeros, amigos y conocidos, que, de alguna manera, confían en mí, gracias por los momentos compartidos, los conocimientos intercambiados y el crecimiento mutuo que hemos experimentado.

Sin ustedes, todo sería mucho más gris y no sé si hoy estaría aquí.

Por último, a Molly, mi gatica, que, aunque hace poco llegó a mi vida, ha sido una acompañante constante con sus caricias, ronroneos, arañazos y mordiscos. Y a mí mismo, porque a pesar de ser demasiado imperfecto, he logrado cosas y, aunque no sean para nada perfectas, son mis logros; los aprecio y significan lo que soy.

Nunca dejes de mejorar.

Brayan Arley Rojas Mayorga.

Agradecemos especialmente a Kuni que hicieron de este camino algo mucho más agradable, a la Electrificadora de Santander por habernos brindado su apoyo en la realización de este proyecto, facilitarnos las herramientas necesarias y ofrecernos un espacio donde pudimos crecer tanto personal como profesionalmente. También, a la Universidad Industrial de Santander, a sus docentes que nos brindaron gran parte del conocimiento que ahora poseemos y por el que este proyecto es una realidad, a sus instalaciones donde, a pesar de la pandemia, cada uno vivió incontables momentos de risa, alegría, angustia, miedo y demás emociones, indicando que estamos viviendo.

Tabla de Contenido

Introducción	14
1. Planteamiento del problema y justificación	15
2. Objetivos.....	16
2.1. Objetivo general	16
2.2. Objetivos específicos.....	16
3. Estado del arte.....	17
3.1. Antecedentes de automatización del proceso	18
4. Marco de referencia.....	19
4.1. Autogeneradores a pequeña escala (AGPE)	19
4.2. Herramientas y enfoques tecnológicos.....	20
4.3. Automatización robótica de procesos (RPA).....	22
5. Metodología	24
5.1. Análisis del sistema	25
5.2. Técnicas de recolección de información	29
5.3. Procedimientos de validación y tratamiento de datos.....	31
5.4. Desarrollo e implementación del informe de Power BI.....	32
5.5. Estrategia de evaluación económica del impacto	34

6. Desarrollo	34
6.1. Desarrollo de la solución RPA.....	34
6.2. Técnicas de recolección de información.....	74
6.3. Procedimientos de validación y tratamiento de datos.....	93
6.4. Desarrollo e implementación del informe de Power BI.....	97
6.5. Estrategia de evaluación económica del impacto	113
7. Conclusiones	116
8. Trabajo Futuro	118
Referencias bibliográficas	119

Lista de Figuras

Figura 1	Arquitectura del proyecto	28
Figura 2	Esquema de indicadores clave para los dashboards	33
Figura 3	Diagrama BPMN del RPA	36
Figura 4	Arquitectura de la librería OracleDB	38
Figura 5	Estructura del proyecto Reporte de Excedentes.	39
Figura 6	Diagrama BPMN: Fase de exportación de archivo csv	52
Figura 7	Diagrama BPMN: Fase de generación del reporte.....	53
Figura 8	Diagrama BPMN: Fase de cargue del reporte	53
Figura 9	Flujo de nube de power automate.....	55
Figura 10	Conexión de máquina local.....	56
Figura 11	Flujo principal de escritorio en power automate	58
Figura 12	Tarea ejecutar comando de DOS	59
Figura 13	Subflujo envío de correo	60
Figura 14	Subflujo error de adjunto.....	61
Figura 15	Visualización del histórico de procesos	69
Figura 16	Configuración de actualización del modelo semántico	72
Figura 17	Configuración del gateway para conexiones a los orígenes de datos.....	73

Figura 18 Elementos estructurados como registros	95
Figura 19 Diagrama BPMN del Seguimiento AGPE en operación (AS-IS).....	97
Figura 20 Diagrama BPMN del Seguimiento de visitas (AS-IS)	99
Figura 21 Diagrama BPMN para Seguimiento de los AGPE en Power BI.....	100
Figura 22 Modelo relacional en Power BI.....	101
Figura 23 Dashboard de los excedentes generados por AGPE	106
Figura 24 Menú principal del informe de Power BI.....	108
Figura 25 Panel de autogeneradores en operación.....	109
Figura 26 Tabla informativa accedida mediante Drill Through.....	111
Figura 27 Horas consumidas antes y después de la solución.	114
Figura 28 Ahorro y Retorno sobre la Inversión de los próximos años.	115

Lista de Tablas

Tabla 1 Tipos de energía en el sistema.....	44
Tabla 2 Campos de Excedentes.....	75
Tabla 3 Campos de relaciones técnicas.....	76
Tabla 4 Campos de clientes y atributos descriptivos.....	76
Tabla 5 Campos de detalles administrativos y técnicos de los autogeneradores.....	78
Tabla 6 Campos de trazabilidad de los procesos.....	79
Tabla 7 Campos del estado del proceso.....	80
Tabla 8 Campos complementarios de clientes.....	81
Tabla 9 Campos de nombre de encargados.....	83
Tabla 10 Campos de procesos históricos de los autogeneradores.....	83
Tabla 11 Campos de excedentes generados por los autogeneradores.....	84
Tabla 12 Campos de fechas de festivos por año.....	85
Tabla 13 Columnas calculadas del informe de Power BI.....	102
Tabla 14 Medidas en DAX del informe de Power BI.....	104
Tabla 15 Tabla Calendario.....	105

Glosario

Autogeneradores a Pequeña Escala (AGPE): Clientes que generan su propia energía eléctrica, generalmente mediante fuentes renovables como solar o hidráulica, con una capacidad instalada inferior a 1 MW. Los excedentes de energía generados pueden ser inyectados al Sistema Interconectado Nacional (SIN) o vendidos a la red comercializadora.

Automatización Robótica de Procesos (RPA): Tecnología que permite automatizar tareas operativas repetitivas mediante la configuración de flujos de trabajo sin intervención humana. En este proyecto se utilizó Power Automate para ejecutar scripts, validar reportes y enviarlos automáticamente.

PrimeRead: Base de datos utilizada en ESSA que contiene registros de consumo y generación horaria de energía por parte de los clientes AGPE. Es fuente clave para el cálculo de excedentes energéticos.

SAC (Sistema de Administración Comercial): Sistema interno de ESSA que almacena información administrativa, técnica y de trazabilidad de los clientes AGPE y sus procesos de conexión.

Power BI: Herramienta de Microsoft para la visualización de datos. Permite construir dashboards interactivos con indicadores clave sobre el estado de los AGPE, sus procesos y excedentes.

Power Automate: Plataforma RPA de Microsoft utilizada para orquestar la ejecución automática del módulo de Python, validar resultados y enviar reportes a los responsables designados.

Python: Lenguaje de programación usado para extraer, procesar y transformar los datos provenientes de las bases PrimeRead y SAC, generando el archivo de reporte en formato CSV

exigido por XM.

Modo Grueso (Thick Mode): Modo de conexión de la librería oracledb de Python que requiere la instalación local del cliente Oracle. Fue necesario debido a restricciones de red y políticas de seguridad interna en ESSA.

DataLake: Almacenamiento en la nube de ESSA donde reside un conjunto de datos históricos de los procesos AGPE. Se conectó a Power BI para enriquecer el análisis visual y temporal de las solicitudes.

Excedente de Energía: Cantidad de energía generada por un cliente AGPE que excede su consumo. Este valor es reportado a XM mensualmente y requiere validaciones técnicas antes de su consolidación.

XM: Administrador del mercado de energía mayorista en Colombia. Recibe los reportes mensuales de excedentes generados por los AGPE, bajo formatos y plazos establecidos.

Workflow (Flujo de trabajo): Secuencia automatizada de actividades implementada para ejecutar el proceso de generación, validación y envío del reporte de excedentes.

Validación de Datos: Conjunto de reglas y condiciones aplicadas a los datos extraídos, para asegurar su integridad, consistencia y cumplimiento con los lineamientos de XM.

Modelo Semántico: Estructura de datos implementada en Power BI que permite analizar información de forma eficiente, manteniendo relaciones entre tablas y filtros contextuales.

Resumen

Título: Automatización Robótica de Procesos (RPA) para el seguimiento analítico de activos y generación de reportes de excedentes en clientes AGPE de la Electrificadora de Santander (ESSA)*

Autores: Fabian Ernesto Hernández Ramírez, Brayan Arley Rojas Mayorga **

Palabras clave: RPA, Python, Microsoft Power Platform, ETL, AGPE, Eficiencia Operativa, ROI.

Descripción:

La Electrificadora de Santander (ESSA) se encontraba con ineficiencias y riesgos regulatorios debido a la gestión manual de la información de excedentes generados por Autogeneradores a Pequeña Escala (AGPE) para reportes a XM. Este proyecto transformó esos procedimientos manuales en un proceso automatizado, mediante una solución de Automatización Robótica de Procesos (RPA) empleando Python para la extracción y transformación de datos (ETL) de Oracle, y Microsoft Power Platform (Power Automate y Power BI). La duración de estas tareas se redujo de 38 a 3,33 horas mensuales, lo que resultó en un ahorro anual de 416 horas y alcanzando un ROI del 60,17%. La solución puso fin a los fallos humanos, aseguró la exactitud de los informes para XM y reunió dicha información en Power BI, incrementando su accesibilidad y facilitando el monitoreo. Demostrando así el valor de la automatización en los procesos repetitivos y periódicos, liberando recursos tanto humanos como operativos y mejorando la eficiencia operativa.

* Trabajo de investigación

** Facultad de Ingenierías Físicomecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: Carlos Adolfo Beltrán Castro, Ph.D.

Abstract

Title: Robotic Process Automation (RPA) for Analytical Asset Tracking and Surplus Reporting in AGPE Clients of Electrificadora de Santander (ESSA) *

Authors: Fabian Ernesto Hernández Ramírez, Brayan Arley Rojas Mayorga **

Keywords: RPA, Python, Microsoft Power Platform, ETL, AGPE, Operational Efficiency, ROI.

Description:

The Electrificadora de Santander (ESSA) was facing inefficiencies and regulatory risks due to the manual management of surplus information generated by Small-Scale Self-Generators (AGPE) for reporting to XM. This project transformed those manual procedures into an automated process through a Robotic Process Automation (RPA) solution, using Python for data extraction and transformation (ETL) from Oracle, and Microsoft Power Platform (Power Automate and Power BI). The duration of these tasks was reduced from 38 to 3.33 monthly hours, resulting in an annual saving of 416 hours and achieving a ROI of 60.17%. The solution eliminated human errors, ensured the accuracy of the reports for XM, and centralized the information in Power BI, increasing its accessibility and facilitating monitoring. Thus demonstrating the value of automation in repetitive and periodic processes, freeing up both human and operational resources and improving operational efficiency.

* Research work

** Faculty of Physics-Mechanics Engineering. School of Systems Engineering and Informatics. Advisor: Carlos Adolfo Beltrán Castro, Ph.D.

Introducción

Dentro del contexto de la transición energética, las empresas del sector eléctrico enfrentan el reto de gestionar y optimizar procesos relacionados con la autogeneración de energía por parte de los usuarios. En particular, la Electrificadora de Santander (ESSA) debe garantizar la correcta gestión de los Autogeneradores a Pequeña Escala (AGPE), cumpliendo con las normativas regulatorias y asegurando la calidad en la entrega de reportes a entidades como XM, entidad responsable del mercado mayorista de energía en Colombia y fundamental para coordinar las operaciones del sistema eléctrico nacional, así como para proporcionar información precisa sobre capacidad, generación y demanda energética en el país (XM S.A. E.S.P., s.f.).

Este proyecto propone el diseño de una solución de Automatización Robótica de Procesos (RPA) que integre tecnologías como Microsoft Power Platform y Python, con el objetivo de mejorar la gestión de los datos de los AGPE. Esta metodología incluye la automatización de procesos importantes como la elaboración de informes para XM y la creación de un informe de Power BI, facilitando el análisis y monitoreo de indicadores estratégicos.

Con esta solución, ESSA optimizará procesos como la identificación de clientes AGPE con excedentes, la generación de reportes para XM y el cálculo de KPI's sobre las solicitudes de los autogeneradores. Esto reducirá los tiempos de ejecución manual, eliminará errores como la inclusión o exclusión incorrecta de clientes en los reportes, y garantizará el cumplimiento normativo mediante la entrega precisa y oportuna de datos a XM. Este documento especifica la metodología utilizada, los principios teóricos y los recursos requeridos para la realización de este proyecto.

1. Planteamiento del problema y justificación

En la Electrificadora de Santander (ESSA), la falta de un sistema robusto y automatizado para gestionar de manera integrada los datos de los AGPE y sus excedentes causa retrasos y errores en la creación de informes, dado que los datos están sujetos a la dependencia de terceros, lo que ocasiona demoras y fallos en la digitalización de dichos datos. Como resultado, decisiones críticas en áreas financieras y regulatorias se ven afectadas como la correcta facturación a los clientes AGPE y el cumplimiento de obligaciones regulatorias, especialmente en la precisión de los reportes entregados a la empresa XM, entidad encargada de administrar el mercado eléctrico colombiano. La inexactitud en estos informes puede generar multas y sanciones para la empresa impactando negativamente en sus finanzas y reputación.

Adicionalmente, el procedimiento actual exige el uso de herramientas que, con el transcurso del tiempo, se han vuelto anticuadas cuya funcionalidad no es adecuada para las demandas actuales. Esto lleva a decisiones menos precisas como el ajuste de la oferta y demanda de energía, afectando tanto a la compañía como a los clientes que buscan optimizar su producción energética, lo cual reduce la competitividad en un mercado energético cada vez más dinámico.

Se propone implementar una solución de Automatización Robótica de Procesos (RPA), desarrollado con Microsoft Power Platform y Python, que facilite el seguimiento de los clientes AGPE y el reporte de sus excedentes. Estas herramientas permitieron incrementar la exactitud de la información al automatizar la recolección y procesamiento de datos, minimizando errores humanos y estableciendo protocolos estandarizados. También, garantizará que la ESSA mejore la eficiencia operativa al reducir los tiempos de procesamiento de datos y liberar recursos humanos para que puedan enfocarse en otros procesos que generen mayor valor a la empresa. Al emplear

aplicaciones como Power BI para el análisis de datos, ESSA podrá optimizar la toma de decisiones en la gestión de los AGPE en base al estudio de la información visualizada, dando paso a la mejora de capacidad de respuesta, optimizando la gestión de sus recursos y reforzando la toma de decisiones financieras y regulatorias con información precisa y oportuna permitiendo a la empresa adaptarse a las exigencias del mercado y fortalecer su posición competitiva, garantizando el cumplimiento normativo y la satisfacción de sus clientes.

2. Objetivos

2.1. Objetivo general

- Implementar una solución de Automatización Robótica de Procesos (RPA) mediante herramientas de Power Platform y Python para automatizar el seguimiento de solicitudes y la generación de reportes de excedentes, de clientes Autogeneradores a Pequeña Escala (AGPE) en ESSA, facilitando el monitoreo de la gestión y reduciendo errores manuales.

2.2. Objetivos específicos

- Desarrollar un módulo de Python que realice la conexión a la base de datos que contiene la información de los AGPE para su extracción, tratamiento de los datos y generación del reporte de excedentes.

- Crear un flujo RPA utilizando la herramienta Microsoft Power Automate que ejecute mensualmente el módulo Python, enviando el reporte de los excedentes generados al usuario para su validación antes del quinto día hábil de cada mes.

- Diseñar el informe en Microsoft Power BI integrado al sistema SAC y un conjunto

de datos dentro del DataLake de la ESSA, permitiendo la visualización de información actualizada de los procesos finalizados, procesos en trámite y autogeneradores activos de los clientes de autogeneradores, facilitando la identificación de tendencias y la gestión proactiva de solicitudes.

- Publicar el informe en Power BI Service, con actualizaciones automáticas, permitiendo el acceso centralizado de los equipos involucrados en la gestión de los AGPE y sus excedentes.

3. Estado del arte

El proceso actual para generar y publicar reportes sobre los excedentes energéticos de los clientes autogeneradores a pequeña escala (AGPE) en la Electricadora de Santander S.A. (ESSA) ha estado tradicionalmente marcado por una alta carga manual y operativa. Este proceso inicia con la ejecución mensual de consultas mediante el módulo interno de ESSA llamado Business PrimeRead, generando así archivos CSV con los datos crudos de consumo y generación energética.

Una vez generados estos archivos, el usuario responsable realizaba manualmente una serie de transformaciones en hojas de cálculo Excel, lo que incrementaba significativamente la probabilidad de errores operativos y retrasos. Adicionalmente, era necesario esperar información complementaria proporcionada por el área Comercial T&D, consistente en datos descriptivos clave de los clientes AGPE para completar los reportes solicitados por la entidad reguladora XM.

Debido a la estructura fragmentada y manual de este proceso, se enfrentaban frecuentemente dificultades operativas, incluyendo la necesidad de realizar ajustes fuera de los horarios laborales, retrasos recurrentes en la entrega de información y errores en los datos finales enviados a XM.

3.1. Antecedentes de automatización del proceso

En entrevistas realizadas con usuarios clave del proceso en ESSA, se evidenció que, si bien previamente existieron esfuerzos internos orientados a automatizar parcialmente las tareas involucradas, estos intentos no lograron alcanzar la integración tecnológica ni la automatización completa que exigía el área técnica encargada de estos reportes.

En específico, se realizaron esfuerzos aislados mediante scripts en Python y consultas SQL ejecutadas directamente en las bases de datos PrimeRead y SAC. Sin embargo, estos desarrollos tuvieron dificultades operativas significativas, relacionadas especialmente con limitaciones en seguridad institucional, dificultad en la integración y escaso mantenimiento del código desarrollado.

Internamente en ESSA, también se han llevado a cabo proyectos relacionados con automatización y optimización de procesos utilizando herramientas tecnológicas como Power Automate y Python, con resultados exitosos. No obstante, ninguno de estos proyectos abordaba específicamente las particularidades técnicas y regulatorias de la generación y reporte de excedentes energéticos.

En este contexto, el presente trabajo de grado aborda de forma integral las limitaciones previamente identificadas mediante el desarrollo e implementación de una solución automatizada. Esta solución combina flujos de trabajo en Power Automate Cloud y Power Automate Desktop, scripts desarrollados en Python y tableros interactivos diseñados en Power BI. Esta última herramienta fue implementada específicamente en este proyecto para ofrecer visualizaciones interactivas que permitieran un seguimiento preciso y dinámico de indicadores operativos claves relacionados con los AGPE.

La solución implementada no solo mejoró notablemente la eficiencia operativa del proceso, sino que también mitigó riesgos operativos, garantizó una mayor calidad en la información entregada, y fortaleció significativamente la trazabilidad de los datos críticos para la toma de decisiones en ESSA.

4. Marco de referencia

4.1. Autogeneradores a pequeña escala (AGPE)

En la Electrificadora de Santander (ESSA), se manejan varios tipos de conexiones de autogeneración que se clasifican según su capacidad de producción energética y en base a normativas como la Ley 1715 de 2014 y la Resolución CREG 030 de 2018 (Congreso de Colombia, 2014; Comisión de Regulación de Energía y Gas [CREG], 2018) . En estas conexiones se incluyen los autogeneradores a pequeña escala (AGPE), los autogeneradores a gran escala (AGGE) y los generadores distribuidos (GD).

Un autogenerador a pequeña escala (AGPE) es un sistema implementado por un usuario o cliente que quiere producir su propia energía haciendo uso de fuentes de generación no convencionales de energía renovable (FNCER), como la solar o hidráulica. Estos clientes buscan producir energía eléctrica para satisfacer sus necesidades, según la regulación el tamaño de generación de energía debe ser inferior a 1 MW, con la posibilidad de vender sus excedentes o inyectarlos al Sistema Interconectado Nacional (SIN). En base a la Resolución CREG 030 de 2018 hay dos tipos de AGPE, los que tienen una capacidad o potencia instalada menor o igual a 100kW y los que tienen una capacidad mayor a 100kW y menor o igual a 1MW (Comisión de Regulación de Energía y Gas [CREG], 2018).

Los excedentes de energía corresponden a la cantidad generada por estos usuarios que

exceden su propio consumo. Según la ley 1715 de 2014 y la Resolución GREG 030 de 2018, estos excedentes pueden ser vendidos a los comercializadores de energía (en el caso de ESSA, a XM) o inyectarlos al SIN, bajo requisitos técnicos ya que deben contar con equipos de medición bidireccional que registren el consumo y el exceso de esta energía producida, y condiciones comerciales específicas.

En este contexto, ESSA desempeña el rol de supervisar que estos sistemas cumplan con los estándares técnicos y comerciales establecidos en las normativas. Además, ESSA garantiza la correcta conexión y operación de los AGPE, y asegura el reporte adecuado de los excedentes generados al administrador del mercado de energía mayorista de Colombia llamado XM (Electrificadora de Santander S.A., 2020). Por su parte, XM establece plazos específicos para la entrega del reporte de estos excedentes por parte de ESSA cuyo incumplimiento genera multas para la empresa y afecta la conexión y operación de estos clientes AGPE.

4.2. Herramientas y enfoques tecnológicos

4.2.1. Microsoft Power Platform

Microsoft Power Platform es una solución integral diseñada para optimizar la transformación digital en las organizaciones a través de la integración de datos, procesos y equipos (Dataprise, 2022). Combina herramientas como Power BI, Power Automate y Power Apps para optimizar flujos de trabajo, generar análisis avanzados y construir soluciones personalizadas. Estas herramientas permiten tanto a empleados técnicos como no técnicos que participen en la mejora de procesos y la toma de decisiones basadas en datos. Se hizo uso de las siguientes herramientas pertenecientes a Power Platform:

Power Automate: Una herramienta que automatiza la ejecución de tareas manuales y

optimiza los procesos. En este proyecto, Power Automate será empleado para validar, procesar y cargar automáticamente los reportes de excedentes en la plataforma XM, asegurando el cumplimiento de los plazos regulados y minimizando errores humanos.

Power BI: Una solución de análisis de datos y visualización interactiva que transforma la información compleja en gráficos e informes accesibles. En este proyecto, Power BI será usado para crear dashboards dinámicos que representen indicadores clave de clientes AGPE, tales como excedentes generados, estado de procesos y métricas operativas.

4.2.2. Python

Python se ha consolidado como una herramienta versátil y eficiente para el análisis de grandes volúmenes de datos. Esto se debe en gran parte a bibliotecas como Pandas y Numpy, las cuales permiten realizar operaciones complejas de manera eficiente y optimizada.

Pandas: es una biblioteca que proporciona estructuras de datos avanzadas como los dataframes, que facilitan la manipulación de datos tabulares y temporales. Gracias a las funciones que integra esta biblioteca, es posible realizar tareas de filtrado, agrupamiento y transformación, lo que es ideal para manejar los datos de excedentes de los clientes AGPE.

Numpy: esta biblioteca está diseñada para cálculos numéricos. Ofrece soporte para operaciones vectorizadas lo que mejora significativamente el rendimiento en comparación a métodos tradicionales. En este proyecto, numpy será utilizado para realizar cálculos que sigan las métricas mensuales exigidas por XM.

4.2.3. Caso práctico

La implementación de Microsoft Power Platform y Python ha mostrado resultados exitosos

en la automatización de procesos empresariales. Un ejemplo es el proyecto realizado por Julián León en la Electrificadora de Santander S.A. (ESSA), que abordó la generación automática de reportes contables mediante la integración de herramientas de Power Platform como Power Apps y Power Automate.

Aunque este proyecto se centró en procesos contables y el desarrollo de interfaces con Power Apps, permitió reducir significativamente el tiempo de generación de reportes gracias a la eficiencia de combinar Power Automate para la automatización de flujos de trabajo con Python para el tratamiento y validación de datos. En contraste, el presente proyecto se enfoca en el reporte técnico y regulatorio de excedentes energéticos generados por los AGPE, sin incluir Power Apps, pero con una integración más robusta entre Python, Power Automate y Power BI. Estos resultados, en ambos casos, demuestran la capacidad de estas herramientas para manejar grandes volúmenes de datos, mejorar la productividad operativa y minimizar errores humanos (León, 2024).

4.3. Automatización robótica de procesos (RPA)

La automatización robótica de procesos (RPA) ha revolucionado el ámbito empresarial mediante la optimización de los flujos de trabajos manuales repetitivos y complejos. Sus características clave incluyen:

- **Integración sin APIs:** Opera sobre la capa de interfaz de usuario de aplicaciones existentes, sin requerir modificaciones en los sistemas legacy.
- **Escalabilidad:** Permite desplegar múltiples bots en paralelo para manejar grandes volúmenes de datos.
- **Precisión:** Elimina errores humanos en procesos como entrada de datos o cálculos

numéricos.

Según Ilo (2018), el uso de RPA en procesos de reportes contables permite centralizar datos provenientes de múltiples fuentes, mejorar la eficiencia operativa y garantiza la precisión de la información generada. Es ideal para automatizar soluciones como:

- **Procesos Repetitivos Basados en Reglas:** Tareas que siguen pasos predefinidos y no requieren toma de decisiones complejas.
- **Procesos de Alto Volumen y Frecuencia:** Aquellos que se ejecutan muchas veces al día, semana o mes.
- **Procesos con alta propensión a Errores Humanos:** Tareas manuales donde los errores pueden generar costos o retrabajos.
- **Procesos que Involucran Múltiples Sistemas sin Integración Nativa:** Tareas que requieren copiar y pegar información entre diferentes aplicaciones que no están conectadas directamente.

En este proyecto, se utilizarán las capacidades de RPA mediante la herramienta de Microsoft Power Automate, que permitieron automatizar la validación, el procesamiento y el cargue de reportes regulatorios relacionados a los clientes AGPE. Para garantizar una implementación exitosa, existen algunas buenas prácticas recomendadas del ciclo de vida del RPA (ND Marketing Digital, 2024):

- **Establecimiento de objetivos:** Definir claramente los objetivos que se buscan con la implementación del RPA.
- **Identificación de procesos candidatos:** Priorizar tareas repetitivas, que se basan

en reglas e implican grandes volúmenes de datos.

- **Pruebas de viabilidad:** Evaluar la factibilidad técnica y económica de automatizar los procesos seleccionados, asegurando que la inversión en RPA genere un retorno positivo.
- **Diseño y desarrollo:** crear el flujo del proceso en la herramienta RPA seleccionada, asegurando que cada etapa esté bien documentada y que sigan las mejores prácticas del desarrollo.
- **Pruebas y validación:** realizar pruebas piloto con subconjuntos de datos antes del despliegue total para asegurar que el bot funcione correctamente y cumpla con los objetivos establecidos.
- **Implementación y despliegue:** Poner en producción el bot RPA, asegurando que esté integrado correctamente con los sistemas existentes y los usuarios estén capacitados para interactuar con él. A su vez, entregar la documentación necesaria para su mantenimiento y escalabilidad.
- **Monitoreo continuo:** Auditorías periódicas para asegurar la calidad de los datos y el rendimiento del sistema de RPA.

Esta automatización no solo garantiza el cumplimiento de los plazos establecidos por XM, sino también una reducción en los tiempos de ejecución. Ilo destaca que la integración de RPA en procesos críticos no solo optimiza recursos, sino que refuerza la trazabilidad y la calidad de los resultados, beneficios aplicables al contexto de la Electrificadora de Santander S.A.

5. Metodología

Para el desarrollo y cumplimiento de los objetivos, se siguió una serie de cuatro pasos que consta en implementar la solución RPA de la siguiente manera:

5.1. Análisis del sistema

Con el fin de lograr los objetivos establecidos, se planea diseñar un RPA que integre diferentes tecnologías de Microsoft Power Platform y una herramienta de código interpretado, en este caso Python, para garantizar la extracción, tratamiento, validación y finalmente visualización de los datos de los clientes AGPE. Este diseño busca agilizar el proceso de generación del reporte actual, reduciendo los tiempos de ejecución y cumpliendo los plazos establecidos por las normas regulatorias constando de los siguientes componentes:

5.1.1. Bases de datos Oracle

El RPA se conectó a las bases de datos Oracle PrimeRead y SAC, donde se encuentra la información clave de consumo y excedentes de los clientes AGPE. Para más detalles, consultar el apartado 5.2.1 Origen de los datos.

5.1.2. Módulo Python

El módulo fue la parte fundamental del inicio del RPA debido a que incluye partes fundamentales como la conexión, tratamiento de datos y generación del reporte en formato CSV (exigido por XM). Teniendo esto en cuenta, se construyeron varios scripts que se integren con un script main que es el que ejecuta paso a paso el módulo de la siguiente manera:

- **Conexión a la base de datos:** haciendo uso de la librería oracledb en modo thick, se garantizó una conexión estable a las bases PrimeRead y SAC.
- **Extracción de datos:** se ejecutaron las consultas SQL optimizadas que permitieron reducir significativamente los tiempos actuales.
- **Tratamiento de datos:** se usaron las librerías pandas, numpy y openpyxl para

filtrar los datos relevantes, realizar los cálculos y validaciones según el formato exigido por XM y exportar este archivo en formato CSV para su cargue.

- **Alimentación del Power BI:** Power BI admite varios orígenes de datos y entre estos está “Ejecutar un script de Python” el cual se maneja en base a un dataframe, por medio de un .csv, que retornó el módulo y que se usó directamente por Power BI para actualizar los dashboards.

5.1.3. Power Automate (RPA)

Power Automate se empleó como herramienta RPA para implementar una solución encargada de ejecutar el flujo de trabajo relacionado con la validación y entrega del reporte generado por el módulo Python descrito anteriormente. El desarrollo de la solución siguió una serie de pasos clave permitió optimizar su implementación y funcionamiento. Estos pasos fueron (Microsoft Corporation, 2023):

1. **Definir el alcance:** Identificar claramente cuáles son las tareas posibles que se desean automatizar con el RPA y establecer lo que se espera alcanzar con su implementación.
2. **Diseñar los procesos a automatizar:** Realizar un análisis de los procesos a automatizar, identificar las etapas involucradas y diseñar un flujo de trabajo que pueda ser replicado por el RPA.
3. **Desarrollar las configuraciones en la herramienta:** Crear scripts y configuraciones necesarias en la herramienta de RPA para que pueda ejecutar las tareas definidas de manera automatizada y precisa.
4. **Realizar pruebas:** Llevar a cabo pruebas para verificar su correcto

funcionamiento, identificar posibles errores y realizar los ajustes necesarios.

5. **Implementar el RPA en producción:** Implementar el RPA en el entorno de producción, asegurando la integración adecuada con los sistemas existentes.

6. **Entrega de documentación:** Se entregó un manual técnico que detalle la arquitectura de la solución, configuración y procedimientos en caso de fallos. A su vez, se entregó una guía operativa con instrucciones para los usuarios finales.

Las principales funciones que integraron este flujo fueron:

- **Ejecución del módulo de Python:** se iniciará el flujo de trabajo de acuerdo con los plazos establecidos en los que se configurará esta ejecución del módulo Python para la generación del reporte.
- **Validación del reporte generado:** al generar el archivo CSV, este flujo envió una aprobación al encargado actual de este proceso para validar que el formato y los datos sean consistentes según el formato exigido por XM.

5.1.4. Power BI

En esta aplicación se hizo la visualización interactiva de la información haciendo uso de los siguientes orígenes de datos:

- **Datos generados por el módulo Python:** Excedentes generados por los clientes AGPE
- **Base de Datos del SAC:** Detalles de los autogeneradores, Clientes, Solicitudes de conexión AGPE.

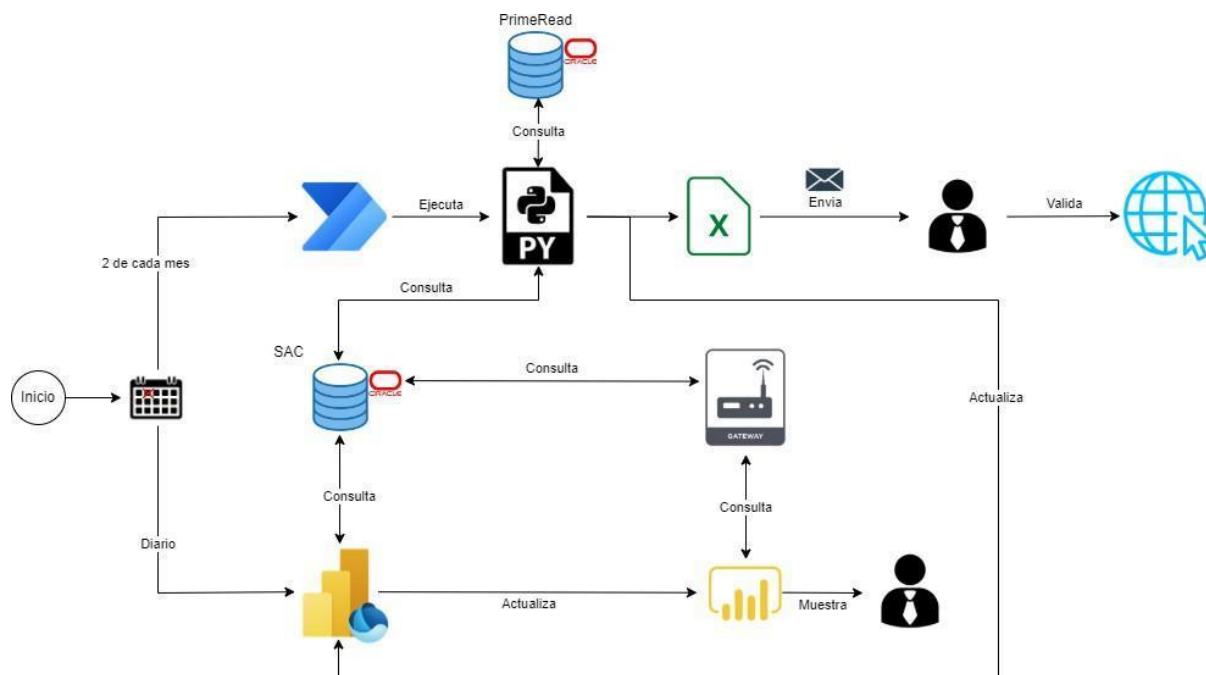
- **Datalake de ESSA:** Dataset completo de procesos de autogeneradores, desde la solicitud inicial hasta la respuesta final.

Para la visualización gráfica de los clientes AGPE activos, sus excedentes y demás información relacionada con los autogeneradores. Este informe será actualizado automáticamente y estará disponible a los encargados de recibir la información, facilitando el análisis de los procesos de los autogeneradores.

La Figura 1 presenta la estructura básica del RPA diseñado para la recolección, procesamiento, verificación y representación visual de la información de los clientes AGPE. El flujo muestra las interacciones entre las bases de datos Oracle (PrimeRead y SAC), el módulo Python responsable de procesar datos, y las herramientas de la plataforma Power Automate y Power BI de Microsoft. Cada elemento juega un papel crucial en la automatización y mejora del proceso (Stela.ai, 2024).

Figura 1

Arquitectura del proyecto



Nota. Figura elaborada por los autores. Algunos íconos fueron tomados de Vecteezy (2024), recuperado de <https://es.vecteezy.com/>. Licencia gratuita con atribución.

5.2. Técnicas de recolección de información

Para que el RPA opere adecuadamente, se establecieron unas técnicas de recolección y validación de la información. Estos datos se obtendrán directamente de las bases de datos Oracle de ESSA: PrimeRead y SAC. Estas bases contienen la información crítica sobre los clientes AGPE y sus excedentes que fueron la fuente para el desarrollo del proyecto.

5.2.1. Origen de los datos

PrimeRead:

Esta base de datos contiene la información respecto al consumo horario y los excedentes generados por los clientes AGPE. Los datos son necesarios para:

- Calcular los excedentes mensuales que se reportarán a XM.

- Validar los valores generados por los clientes para cumplir con las normativas regulatorias.

Sistema de Administración Comercial (SAC):

Esta base de datos almacena toda la información administrativa de los clientes y de los procesos de los AGPE como, por ejemplo:

- Identificación y estado del cliente (si se encuentra en operación)
- Fecha y tipo de conexión
- Procesos asociados a la conexión de los AGPE

Ambas bases de datos se conectaron utilizando la librería oracledb de Python, garantizando una conexión estable y eficiente.

Datalake de ESSA:

En un Storage de este datalake se encuentra un dataset que abarca todo el ciclo de vida de los procesos de autogeneradores, desde la solicitud inicial hasta la conexión final, junto con información detallada de los promotores y clientes de cada autogenerador. Estos datos permiten realizar análisis estratégicos como:

- Cálculo del crecimiento mensual y anual de clientes AGPE.
- Identificación de municipios con mayor concentración de autogeneradores.
- Ranking de promotores según volumen de instalaciones gestionadas.

Esta fuente se integra exclusivamente en el informe de Power BI mediante conexiones directas al storage, donde se transforman los datos crudos en visualizaciones interactivas.

Facilitando la identificación de patrones y toma de decisiones basada en comportamientos.

5.2.2. Validación de los datos

Para garantizar la calidad de los datos, se implementaron unas estrategias de validación y manejo de errores:

Validación de consistencia:

- Realizar una verificación de datos duplicados y valores inconsistentes (por ejemplo, excedentes negativos o fuera de rango normativo).
- Comparar que los datos del PrimeRead correspondan a los registros administrativos del SAC (los que se tengan).

Manejo de errores:

- Usar bloques try-except en los scripts de conexión para detectar fallas en la conexión de la base de datos y reintentar la conexión en caso de errores.

Pruebas de integridad:

- Realizar pruebas de las consultas a las bases de datos antes del proceso de extracción asegurando que los datos fueran accesibles y consistentes.

5.2.3. Estrategias de extracción

Se construyeron las consultas SQL optimizadas para reducir tiempos de ejecución agregando filtros para extraer únicamente los datos del mes a reportar en lugar de consultar toda la base de datos.

5.3. Procedimientos de validación y tratamiento de datos

Una vez llevada a cabo la extracción de los datos desde PrimeRead y SAC, fue necesario realizar un proceso de validación y tratamiento de datos para garantizar la fiabilidad de los datos, que se cumplan los requisitos establecidos por XM y sea adecuada para la generación del reporte y la visualización de la información en Power BI. Este proceso se llevó a cabo mediante herramientas de procesamiento de datos en Python, como las librerías numpy, oracledb y openpyxl, las cuales permiten filtrar, calcular, y transformar estos datos.

Limpieza, transformación y validación de los datos:

Los datos extraídos fueron transformados y validados para adaptarse al formato requerido por XM y alimentar los dashboards en Power BI de la siguiente manera:

- **Eliminación de duplicados:** usando la librería pandas, se identificaron y eliminaron los registros repetidos.
- **Cálculo de excedentes mensuales:** haciendo uso de la librería numpy, se agruparon los datos de excedentes por hora de cada cliente y se calculará el total mensual.
- **Manejo de valores faltantes:** se rellenan estos datos faltantes con valores calculados según el cliente o se eliminaron según su relevancia.
- **Reformateo de columnas:** las columnas se estructuraron de acuerdo con las especificaciones de XM, incluyendo fechas y excedente horario.

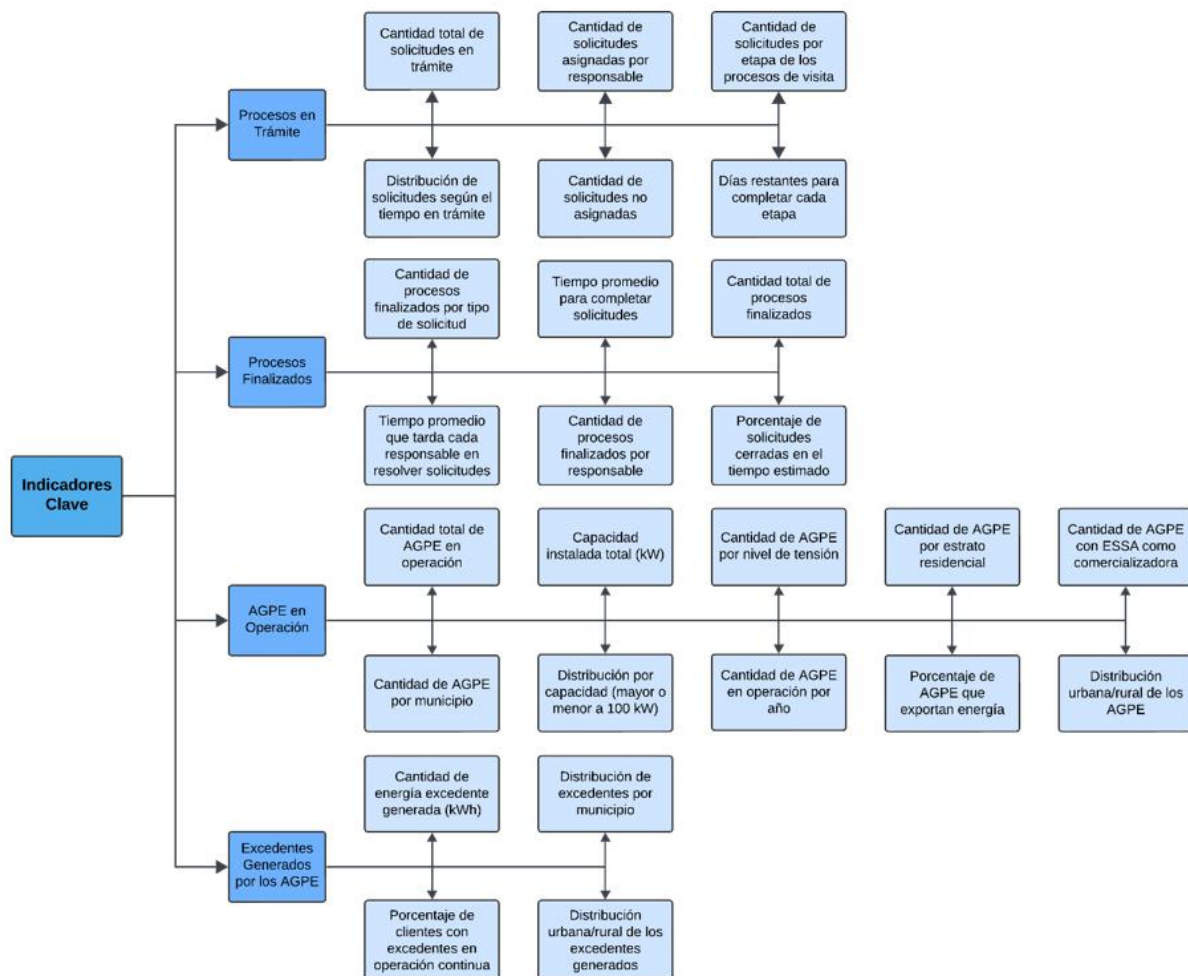
5.4. Desarrollo e implementación del informe de Power BI

El informe en Power BI permite a los responsables del sistema visualizar y analizar los datos de los clientes AGPE de manera interactiva. Este informe se diseñará con unos indicadores claves requeridos por los responsables del sistema.

En la Figura 2 se presentan de forma visual los indicadores clave definidos para cada una de las secciones a analizar: procesos en trámite, procesos finalizados, AGPE en operación y excedentes generados por los AGPE. Esta representación tiene como objetivo sintetizar y organizar los indicadores que pueden dar valor a ESSA en el contexto de los AGPE.

Figura 2

Esquema de indicadores clave para los dashboards



Nota. Figura elaborada por los autores. Representa los indicadores clave definidos para los dashboards del proyecto.

5.5. Estrategia de evaluación económica del impacto

Como parte del proyecto, se decidió calcular el impacto económico que tendría la automatización desarrollada. Para esto, se tomó como base el tiempo que antes tomaba realizar manualmente todo el proceso de generación, consolidación y envío del reporte, comparándolo con el nuevo tiempo de ejecución una vez implementado el RPA.

El cálculo se basó principalmente en el ahorro de horas operativas, considerando que el proceso antes se hacía todos los días de manera manual y ahora se ejecuta automáticamente dos veces al día. Con ese ahorro estimado en horas, se proyectó un valor anual aproximado usando como referencia el costo promedio de una hora operativa.

Para expresar este impacto de forma más clara, se utilizó la fórmula del retorno de inversión (ROI), que se calcula como:

$$ROI = \left(\frac{\textit{Beneficio neto}}{\textit{Costo de inversión}} \right) \times 100$$

Este análisis permite presentar de forma cuantitativa el impacto generado por la solución, y servirá como base para justificar futuras iniciativas de automatización dentro de ESSA.

6. Desarrollo

6.1. Desarrollo de la solución RPA

El desarrollo de esta solución RPA se llevó a cabo mediante la metodología del ciclo de vida de un RPA, descrita en la metodología. Desde el inicio del desarrollo llevamos a cabo reuniones con los usuarios finales con el fin de establecer unos objetivos, definiendo así los límites del alcance que tuvo la solución en el tiempo que fue creada.

Durante esta etapa de análisis, se identificaron las tareas que serían entregadas, quiénes eran los actores involucrados, qué se esperaba entregar y las herramientas tecnológicas necesarias para su ejecución. Gracias a esto se diseñó la solución basada en los siguientes componentes:

- Bases de datos Oracle, donde reside la información crítica de consumo y excedentes de los clientes AGPE.
- Módulo Python, encargado del tratamiento de datos, ejecución de consultas y generación del reporte.
- Power Automate, utilizado para orquestar la ejecución automatizada del módulo mediante flujos en la nube y de escritorio.
- Power BI, plataforma que permite visualizar los resultados procesados y facilitar el análisis de estos.

Además, como parte del análisis inicial del proyecto, se realizó una breve evaluación económica preliminar estimando el retorno de la inversión (ROI), basado en la reducción proyectada de horas operativas manuales y los costos asociados al procedimiento actual. Esta evaluación sirvió para justificar la viabilidad económica del proyecto y respaldar la decisión estratégica de implementar la solución propuesta.

Posteriormente, se establecieron diagramas técnicos y modelos BPMN que facilitaron la comunicación entre el usuario final y el equipo desarrollador, permitiendo orientar claramente el flujo del proceso y asegurar el cumplimiento de los objetivos establecidos.

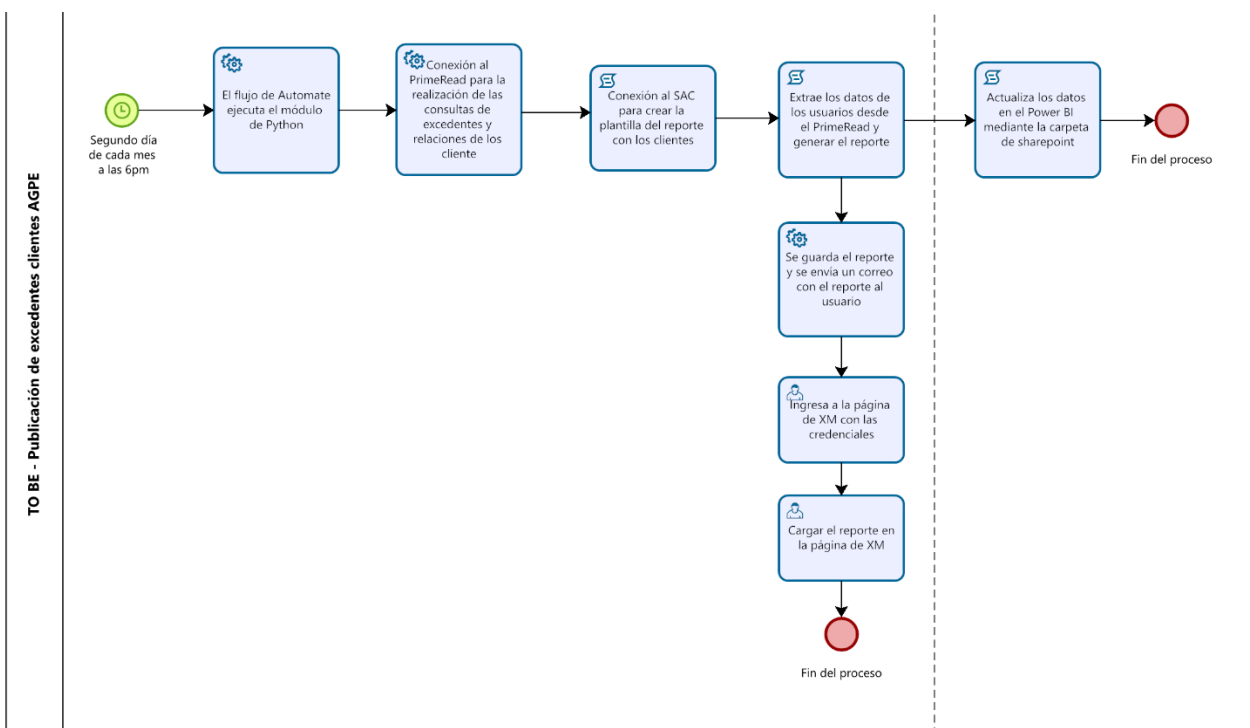
Como se muestra en la figura de la arquitectura de la solución (ver Figura 1, capítulo 5), esta se basa en la interacción entre los flujos de Power Automate, el módulo Python, las bases de

datos PrimeRead y SAC, y las herramientas de visualización en Power BI. Además de la arquitectura, se diseñó el siguiente diagrama BPMN para modelar el flujo del proceso automatizado.

En la Figura 3 se presenta el flujo BPMN del proceso automatizado correspondiente al RPA desarrollado. Se reflejan una serie de secuencias de actividades desarrolladas por el RPA, iniciando la ejecución del módulo Python por parte de Power Automate, junto a las conexiones de las bases de datos PrimeRead y SAC para la recolección de datos, generando así el reporte final de los excedentes.

Figura 3

Diagrama BPMN del RPA



Nota. Figura elaborada por los autores. Representa el flujo optimizado del proceso automatizado

(TO-BE) para la publicación de excedentes de clientes AGPE.

Posteriormente, el reporte es enviado al usuario responsable para su revisión y cargue en la plataforma XM. En paralelo, el script retorna el Dataframe en formato .csv para la carpeta Sharepoint que alimenta informes desarrollados en Power BI, permitiendo la actualización de los tableros.

6.1.1. Bases de datos Oracle

El proyecto establece conexión con dos bases de datos Oracle internas de ESSA: PrimeRead y SAC, las cuales contienen la información crítica relacionada con el consumo y los excedentes de los autogeneradores a pequeña escala (AGPE). Esta conexión se realiza utilizando la librería oracledb en su modo grueso (thick mode), el cual permite establecer una conexión robusta mediante bibliotecas cliente de Oracle instaladas localmente en el equipo.

El uso del modo grueso fue necesario debido a las restricciones de red y políticas internas de seguridad de la ESSA, las cuales limitan el acceso directo mediante conexiones ligeras (thin mode). Según la documentación oficial de Oracle, el modo grueso es recomendado cuando se requiere un mayor control sobre la configuración de red, autenticación y acceso a funcionalidades avanzadas del cliente Oracle, como el soporte para wallets, archivos sqlnet.ora y tnsnames.ora (Oracle, s.f.).

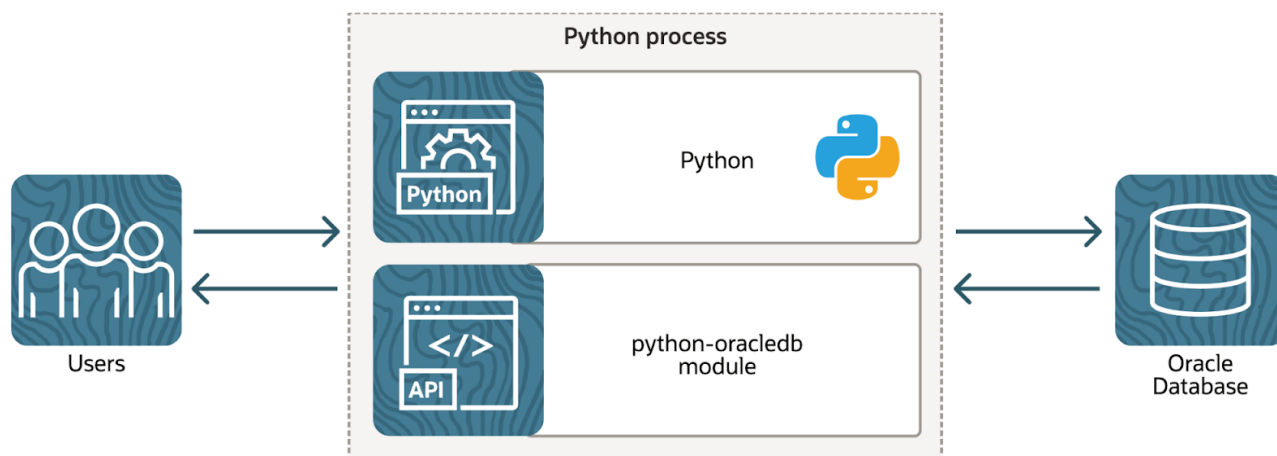
La configuración de las conexiones, incluyendo los parámetros de host, puerto, service_name, credenciales y ruta a las bibliotecas (lib_dir), se encuentra definida en archivos cifrados dentro del módulo Python. Esta información se mantiene en un archivo .json que contiene la cadena de conexión completa, el cual es cifrado por motivos de seguridad. El proceso de cifrado y descifrado se gestiona mediante la librería Fernet, perteneciente al paquete cryptography de

Python, lo que permite garantizar la confidencialidad de las credenciales sensibles durante la ejecución del proceso.

En la Figura 4 se presenta un esquema general del proceso de conexión entre el módulo Python y las bases de datos Oracle, a través del uso del cliente python-oracledb, el cual intermedia las consultas requeridas por el usuario (Oracle, s.f.).

Figura 4

Arquitectura de la librería OracleDB



Nota. Imagen adaptada de User Guide: Enabling Thick Mode, por Oracle (s.f.). Recuperado de https://python-oracledb.readthedocs.io/en/latest/user_guide/installation.html#enabling-thick-mode. Licencia gratuita para uso con atribución.

Cabe aclarar que este apartado se enfoca en los aspectos técnicos de conexión y configuración. Para una explicación detallada sobre las consultas SQL y los datos extraídos de estas bases de datos, se recomienda consultar la sección 6.2.1 Origen de los datos módulo.

6.1.2. Módulo Python

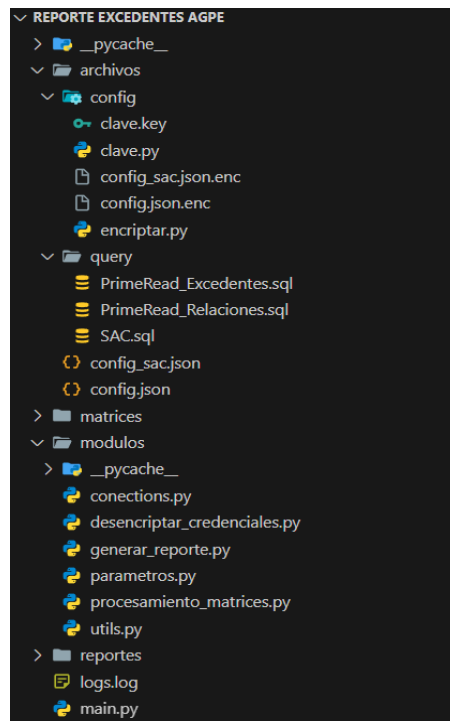
El módulo de Python es responsable de la conexión a las bases de datos de la empresa para extraer estos datos, transformarlos y generar el reporte bajo el formato trabajado con XM. Finalmente, este reporte será el que alimente parte de los tableros de Power BI para el análisis correspondiente.

Estructura de carpetas:

Con el fin de facilitar el mantenimiento y tener una buena administración del proceso, el módulo implementa una estructura modular organizada por carpetas y subcarpetas que funcionan entre sí, asegurando una correcta ejecución del módulo:

Figura 5

Estructura del proyecto Reporte de Excedentes.



Nota. Figura elaborada por los autores. Captura de pantalla generada desde el entorno de desarrollo

Visual Studio Code, correspondiente a la estructura del proyecto Reporte Excedentes AGPE.

- **config:** esta carpeta alberga los archivos cifrados con las credenciales necesarias para la conexión a las bases de datos de la ESSA (PrimeRead y SAC). Además, contiene los archivos de cifrado .enc junto a su respectiva clave.key que almacena la información de la cadena de conexión de las bases de datos.
- **query:** contiene las consultas SQL para extraer únicamente los datos necesarios para la generación del reporte mensual desde cada una de las bases de datos. Estas consultas están almacenadas externamente en archivos .sql, lo cual simplifica su modificación y facilita su mantenimiento.
- **matrices:** carpeta destinada a almacenar archivos .xlsx temporales generados durante la ejecución del módulo.
- **módulos:** esta carpeta agrupa los scripts de Python que cumplen funciones específicas, incluyendo la conexión segura a Oracle, el descifrado de credenciales, la gestión dinámica de parámetros de ejecución, funciones auxiliares, procesamiento de matrices y generación del reporte final.
- **reportes:** se encarga de almacenar el resultado final de la ejecución, el cual es un archivo en formato .csv.

Adicionalmente, dentro del módulo se encuentra un archivo main.py, encargado de realizar la ejecución de cada uno de los módulos en orden correspondiente, y un archivo logs.log para facilitar la gestión de errores que puedan generarse durante la ejecución del proceso.

Conexión a la base de datos:

Para establecer la conexión con las bases de datos Oracle, se utiliza la librería `oracledb`, un módulo de extensión de Python que permite a las aplicaciones en este lenguaje interactuar con Oracle Database. Esta librería es la sucesora de `cx_Oracle` y ofrece dos modos de operación: el modo `thin` (delgado) y el modo `thick` (grueso). En este proyecto, se optó por el modo `thick`, que requiere las bibliotecas cliente de Oracle instaladas localmente, proporcionando funcionalidades avanzadas y un mayor control sobre la conexión. Según la documentación oficial de Oracle, el modo `thick` es recomendado cuando se necesita acceso a características como la configuración personalizada de red y autenticación, soporte para wallets y el uso de archivos `sqlnet.ora` y `tnsnames.ora` (Oracle, s.f.).

Con lo anterior, se hace uso de la subcarpeta `config/` y del módulo `connections.py`, el cual contiene una clase llamada `OracleConnection`, que espera un diccionario en formato JSON como parámetro de configuración:

Args:

`config` (dict): Diccionario con la configuración de la conexión.

- `'username'`: Usuario de la base de datos.
- `'password'`: Contraseña del usuario.
- `'host'`: Dirección del servidor Oracle.
- `'port'`: Puerto del servidor Oracle.
- `'sid'` o `'service_name'`: Identificador del servicio o SID de la base de datos.
- `'thick_mode'` (opcional): Activa el modo grueso si es `True`.

- 'lib_dir' (opcional): Ruta a las librerías cliente para el modo grueso.

Esta clase determina, mediante el diccionario JSON, si se hará uso de un sid o de un service_name. En el caso de la ESSA, las bases de datos en producción se conectaban mediante service_name. La clase establece la conexión con la base de datos y crea un cursor para ejecutar la consulta SQL, según los parámetros obtenidos desde los archivos .sql. Con este cursor se recuperan las filas requeridas, las cuales se almacenan en un Dataframe que será utilizado posteriormente en el flujo del proceso.

Adicionalmente, para proteger los datos de conexión, se utilizó la librería Fernet de Python, que permite cifrar el archivo config.json donde están guardadas las credenciales de acceso a las bases de datos. Este archivo contiene la información sensible como el usuario, contraseña, host y nombre del servicio, por lo que se optó por cifrarlo una vez finalizada la configuración inicial.

El proceso de encriptación se realiza desde un script independiente que genera dos archivos: uno cifrado con los datos (config.json) y otro que contiene la clave (key.key). Esta clave se guarda en una carpeta interna del proyecto con acceso restringido. Luego, al momento de ejecutar el proceso, el script principal descripta el archivo en tiempo real para poder establecer la conexión sin exponer las credenciales en texto plano.

Este mecanismo fue implementado con el fin de mejorar la seguridad del módulo, sobre todo considerando que el script debe ejecutarse desde Power Automate sin intervención del usuario. Toda esta lógica quedó documentada para que, en caso de que se deban cambiar las credenciales en el futuro, pueda repetirse el procedimiento de forma rápida.

Extracción de datos:

De acuerdo con las instrucciones contenidas en el script principal main.py, se ejecutan tres consultas SQL que extraen los datos necesarios para el tratamiento posterior en los distintos módulos del proyecto. Las dos primeras consultas provienen de la base de datos PrimeRead y acceden a las tablas internas de la ESSA, las cuales contienen la información relacionada con los excedentes generados.

La consulta PrimeRead_Excedentes retorna los datos de identificación del cliente, la descripción correspondiente, y los valores de excedente por hora y por día. En este caso, cada cliente tiene un valor registrado desde la hora 0 hasta la 23 por cada día del mes. Dado que los contadores de ESSA registran lecturas en intervalos de 15 minutos, se realizó una agrupación horaria para optimizar la consulta, de acuerdo con las recomendaciones del equipo de Tecnología de la Información de ESSA.

Por otro lado, la consulta PrimeRead_Relaciones es de menor complejidad y permite recuperar información descriptiva del medidor del cliente, incluyendo la marca del equipo, su identificador, número de cuenta y descripción asociada al autogenerador. Ambas consultas implementan una cláusula WHERE de filtrado por fecha, que utiliza una variable definida en el módulo de utilidades. Esta variable actualiza automáticamente las fechas al mes vencido (inicio y fin del mes anterior a la ejecución del proceso), evitando así la intervención manual sobre el código.

Las consultas extraídas de PrimeRead contienen información energética desagregada por tipo, siendo estas: energía activa consumida (kWhR), energía activa generada (kWhD), energía reactiva consumida (kVarhR) y energía reactiva generada (kVarhD).

De acuerdo con la Resolución CREG 038 de 2014, la energía activa corresponde a la

energía que realiza trabajo útil en los equipos eléctricos, medida en kilovatios-hora (kWh), mientras que la energía reactiva es aquella que no genera trabajo directo, pero es necesaria para mantener los campos eléctricos y magnéticos en dispositivos como transformadores o motores, y se mide en kilovoltio-amperio reactivo-hora (kVarh). Esta clasificación técnica es indispensable para cumplir con los lineamientos regulatorios relacionados con la medición, control y reporte de excedentes de energía generada o consumida por los autogeneradores a pequeña escala (AGPE).

Cada uno de estos tipos de energía es representado en el sistema con los siguientes códigos:

Tabla 1

Tipos de energía en el sistema

Tipo de energía	Código	Descripción técnica
Energía activa consumida	kWhR	Energía real que es consumida desde la red por el cliente.
Energía activa generada	kWhD	Energía real que es producida por el cliente e inyectada a la red.
Energía reactiva consumida	kVarhR	Energía no útil absorbida por cargas inductivas, como motores o transformadores.
Energía reactiva generada	kVarhD	Energía no útil devuelta al sistema, generalmente por capacitores u otras fuentes.

Nota. Tabla elaborada por autores. Descripción de cada tipo de energía con su código

correspondiente en el sistema.

Esta desagregación es fundamental para el correcto tratamiento de los datos que posteriormente se consolidarán en el archivo de reporte exigido por la comercializadora XM [(Comisión de Regulación de Energía y Gas [CREG], 2014).

Tratamiento de datos:

El tratamiento de datos se realiza principalmente a través de dos módulos: `procesamiento_matrices.py`, encargado de generar las matrices individuales por cliente autogenerador, y `utils.py`, que contiene funciones auxiliares de soporte.

Módulo utilidades (`utils.py`):

Este módulo contiene funciones auxiliares como la creación del archivo de logs y la función de renombramiento de columnas, necesaria para ajustar los nombres de las columnas al formato exigido por XM en el archivo Excel mensual.

También incluye una función que limpia caracteres especiales que provienen de la base de datos en la columna del identificador del cliente usando una expresión regular. La expresión utilizada es:

$$[^\x20 -\x7E]$$

Esta expresión permite identificar y reemplazar cualquier carácter fuera del rango ASCII imprimible (desde el espacio hasta la tilde) por un espacio vacío, proviniendo errores durante la ejecución del proceso.

Módulo procesamiento de matrices:

Este módulo es el encargado de procesar las matrices de consumo y de generación de energía de los clientes autogeneradores. La función principal de este módulo consiste en generar un archivo excel por cliente. Se optó por generar un archivo individual por cliente debido a que, en pruebas iniciales, el manejo por medio de estructuras como diccionarios o dataframes generaba conflictos al cruzarse la información entre clientes, dificultando el tratamiento posterior. Además, esta decisión permitió una mejor integración con el módulo de generación del reporte final, el cual será descrito más adelante.

La función identifica la ruta de las matrices de todos los clientes autogeneradores, ubicada en la subcarpeta matrices/. Inicialmente, agrega una columna con la suma total diaria, calculada por fila, para obtener un acumulado de cada día. Luego, aplica un filtro para excluir clientes con datos incompletos, utilizando la siguiente condición:

$$\text{Cantidad de días nulos} \geq 4 \times \text{días del mes}$$

Esta fórmula asegura que se incluya únicamente a los clientes que tienen registros suficientes para los cuatro tipos de energía requeridos, eliminando aquellos con ausencia de datos significativa.

Posteriormente, el módulo realiza la lectura y relación de los datos provenientes de las dos consultas ejecutadas sobre la base de datos PrimeRead. Se busca la cuenta asociada a cada cliente, y si esta contiene valores nulos, el cliente es descartado del procesamiento. En caso contrario, se genera un archivo Excel estructurado en cuatro pestañas, donde se clasifica la información según el tipo de energía:

- **kWhR:** Energía activa consumida

- **kWhD:** Energía activa generada
- **kVarhR:** Energía reactiva consumida
- **kVarhD:** Energía reactiva generada

Esta estructura permite mantener un control claro sobre los consumos y excedentes registrados, facilitando la validación y consolidación del reporte final. Además, se calcula un total acumulado por tipo de energía, que será utilizado en la generación del archivo que se entrega mensualmente a XM.

Módulo generar reporte:

Este módulo es el encargado de consolidar toda la información procesada previamente y generar el archivo final en el formato requerido por la comercializadora XM. Para ello, lee todas las matrices de consumo y generación de energía creadas por el módulo anterior y las carga en una plantilla prediseñada, que se complementa con datos provenientes de una consulta SQL ejecutada sobre la base de datos SAC y funciones auxiliares del módulo de utilidades.

Esta plantilla representa el formato oficial que debe ser entregado a XM mensualmente y debe cumplir con una estructura específica definida por la entidad. Dentro de los requisitos técnicos se establece que cada cliente debe tener dos registros por día, correspondientes a dos conceptos fundamentales: crédito de energía y total excedente, acompañados por los valores de energía por fecha y hora. Esto genera aproximadamente 60 registros por cliente para cada mes. Además, existe una relación técnica obligatoria entre ambos valores: el crédito de energía no puede superar el total del excedente generado. En este contexto, es importante aclarar que el excedente de energía se define como la diferencia entre la energía generada por el cliente y su energía

consumida en un periodo de tiempo determinado.

Cuando este valor es positivo, representa la energía adicional que el cliente puede inyectar al Sistema Interconectado Nacional (SIN) y que, por lo tanto, puede ser reportada para efectos de compensación.

Durante el procesamiento de este módulo, se cargan y analizan las hojas de energía activa consumida y generada desde cada matriz. Se calculan los totales diarios de cada una para posteriormente aplicar una lógica de ajuste basada en la comparación entre estos valores:

- Si el total consumido es mayor al total generado , se duplican los datos de generación para compensar el desfase.
- Si el total consumido es menor al total generado, se ajustan los valores de generación para no exceder el consumo real.

Una vez realizados estos ajustes, se procede a cruzar los registros de la plantilla con los datos provenientes de las matrices para cada cliente. Cada archivo de matriz temporal se elimina una vez es procesado correctamente, con el fin de evitar sobreescrituras en futuras ejecuciones.

Posteriormente, se realiza un último ajuste técnico. En ocasiones, algunos clientes superan la capacidad instalada de generación eléctrica declarada ante la empresa, lo cual constituye un error atribuible al sistema de medición y control. Para corregir esta inconsistencia, se aplicó una condición que limita el valor máximo del excedente por hora de acuerdo con la capacidad del cliente.

Sea E_x el excedente de energía generado por el cliente en la hora x , y C la capacidad instalada declarada por dicho cliente. Entonces, se establece la siguiente regla:

$$E_x^{ajustado} = \begin{cases} E_x, & \text{si } E_x \leq C \\ C, & \text{si } E_x > C \end{cases}$$

Donde $E_x^{ajustado}$ representa el valor final del excedente reportado en dicha hora. Esta fórmula garantiza que los valores reportados no excedan la capacidad técnica del cliente, conforme a los límites contractuales establecidos por la empresa.

Una vez completado todo el proceso, el reporte final queda listo para su revisión y posterior envío automatizado al usuario final mediante Power Automate. Paralelamente, este archivo será utilizado como fuente para alimentar el dashboard de Power BI, donde los encargados del área de autogeneradores de ESSA podrán realizar el análisis y seguimiento de los datos reportados.

Alimentación del Power BI:

Además, el módulo de generación del reporte retorna un DataFrame con los excedentes generados por los clientes AGPE. Este conjunto de datos es exportado automáticamente como archivo .csv y almacenado en una carpeta designada dentro de SharePoint, la cual fue configurada como fuente de datos en Power BI. Esta estrategia permite que el informe se mantenga sincronizado con los resultados más recientes del procesamiento sin requerir intervención manual por parte del usuario final, facilitando así una integración eficiente entre el análisis en Python y la visualización en Power BI.

6.1.3. Power Automate (RPA)

Según lo expuesto en la metodología, la aplicación de Power Automate se utilizó para ejecutar el módulo Python con una programación mensual y para el envío del reporte a los usuarios finales, con el fin de que estos realicen su revisión correspondiente. Esta herramienta fue seleccionada por dos motivos principales: primero, por su facilidad de aprendizaje y su interfaz

amigable, lo que la hace accesible incluso para usuarios sin experiencia previa en automatización; y segundo, porque el equipo de Tecnología de la Información se encuentra impulsando el uso de Power Automate debido a su bajo costo en licenciamiento, en comparación con otras herramientas RPA utilizadas anteriormente como UIPath.

1. Definir alcance:

Se evaluaron los procesos susceptibles de automatización para determinar cuáles podrían ser incluidos en la solución implementada con Power Automate. Entre las actividades seleccionadas como automatizables se encuentran la ejecución del módulo Python de forma programada y sin intervención manual, así como el envío del reporte final generado al usuario encargado mediante un correo electrónico con el archivo adjunto.

Durante la etapa de desarrollo surgieron limitantes asociadas a las políticas de ciberseguridad de la organización. En particular, se identificó que por motivos de seguridad está restringida la apertura de ventanas de PowerShell en los equipos corporativos, ya que estas pueden representar posibles brechas que comprometan la integridad de la información gestionada por la empresa. Esta restricción impedía el uso directo de comandos para ejecutar el módulo desde Power Automate Desktop.

Ante esta situación, una alternativa planteada fue la programación de tareas en el sistema operativo Windows para ejecutar el módulo de forma desatendida. Estas tareas programadas permiten definir la ejecución de scripts o programas específicos en momentos determinados, sin intervención humana. Según la documentación oficial de Microsoft, el Programador de tareas de Windows (Task Scheduler) permite configurar acciones automáticas asociadas a condiciones específicas como fecha, hora, inicio de sesión, entre otros eventos del sistema operativo

(Microsoft, s.f.-a). Esta opción ofrecía una solución viable para generar el archivo del reporte automáticamente, el cual luego sería detectado y enviado por el flujo desarrollado en Power Automate Cloud.

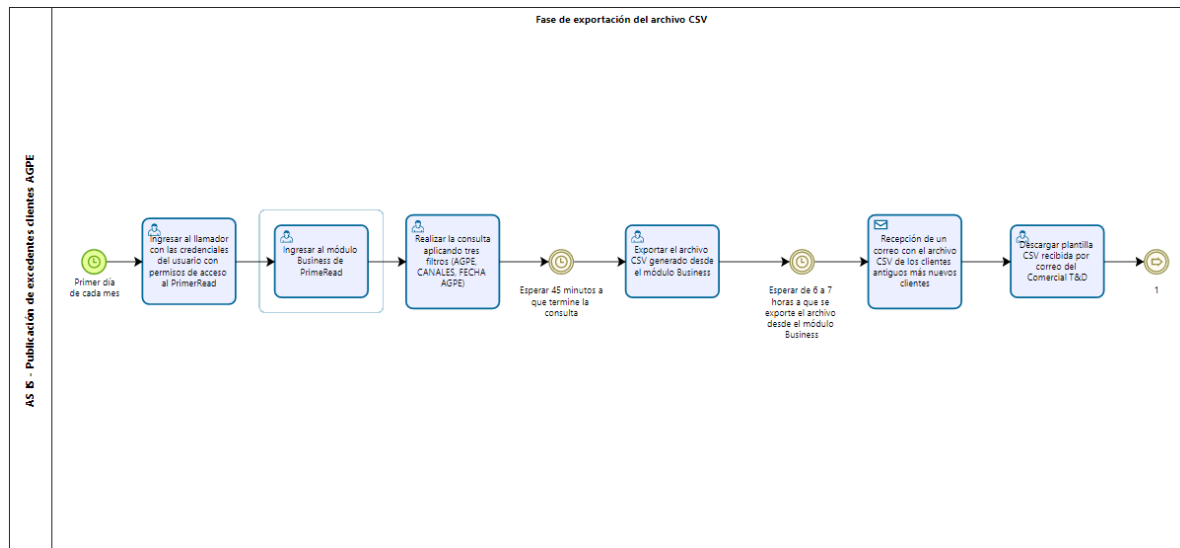
2. Diseñar los procesos a automatizar

Los procesos a automatizar están asociados con las tareas repetitivas que realizaba el usuario cada mes. Entre estas tareas se encontraba la ejecución de la consulta mediante un módulo interno de la empresa para la recolección de los datos, la transformación manual de esta información utilizando copiado y pegado entre archivos Excel, la espera del archivo de plantilla con la información de todos los clientes enviada por un tercero, y los ajustes manuales en horarios no laborales para cumplir con los tiempos establecidos por XM para la entrega del reporte. Con el fin de analizar la mejora del proceso, se construyeron dos diagramas BPMN para comparar el flujo anterior y el nuevo proceso automatizado.

En la Figura 6 se representa la fase inicial del proceso anterior, centrada en la generación del archivo CSV con los datos base para la construcción del reporte mensual. Este proceso comienza el primer día de cada mes y requiere que el usuario ingrese manualmente al módulo Business de PrimeRead utilizando sus credenciales. Posteriormente, se realiza una consulta que aplica filtros por AGPE, canal y fecha, cuya ejecución puede tardar hasta 45 minutos. Tras este paso, el archivo CSV generado debe exportarse, lo cual implica una espera adicional de 6 a 7 horas. Finalmente, el archivo se complementa con una plantilla recibida por correo electrónico desde el área Comercial T&D, que contiene información actualizada de los clientes AGPE. Este conjunto de tareas refleja una alta dependencia del usuario, demoras operativas y un riesgo elevado de errores por el manejo manual de archivos.

Figura 6

Diagrama BPMN: Fase de exportación de archivo csv

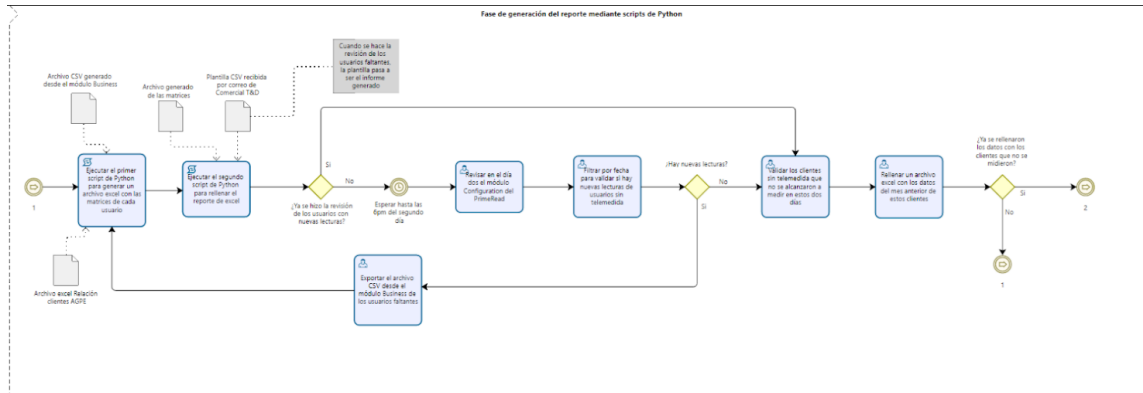


Nota. Figura elaborada por los autores. Representación del subproceso de exportación del archivo en formato CSV dentro del flujo de automatización del RPA.

La Figura 7 corresponde a la segunda fase del proceso AS-IS, en la cual se consolida y transforma la información para la construcción del reporte. Esta etapa comienza con la ejecución de scripts desarrollados en Python que generan matrices por cliente, las cuales son posteriormente integradas en la plantilla proporcionada por T&D. Un punto crítico de esta fase es la revisión manual de usuarios con nuevas lecturas, proceso que requiere validar si existen datos adicionales provenientes del módulo Configuration de PrimeRead. En caso afirmativo, se deben extraer y tratar manualmente los datos de clientes que no fueron medidos en los dos primeros días del mes. El objetivo es completar sus registros con datos del mes anterior, lo que introduce un alto grado de intervención humana y posibles errores en el reporte final, además de demoras en la validación.

Figura 7

Diagrama BPMN: Fase de generación del reporte

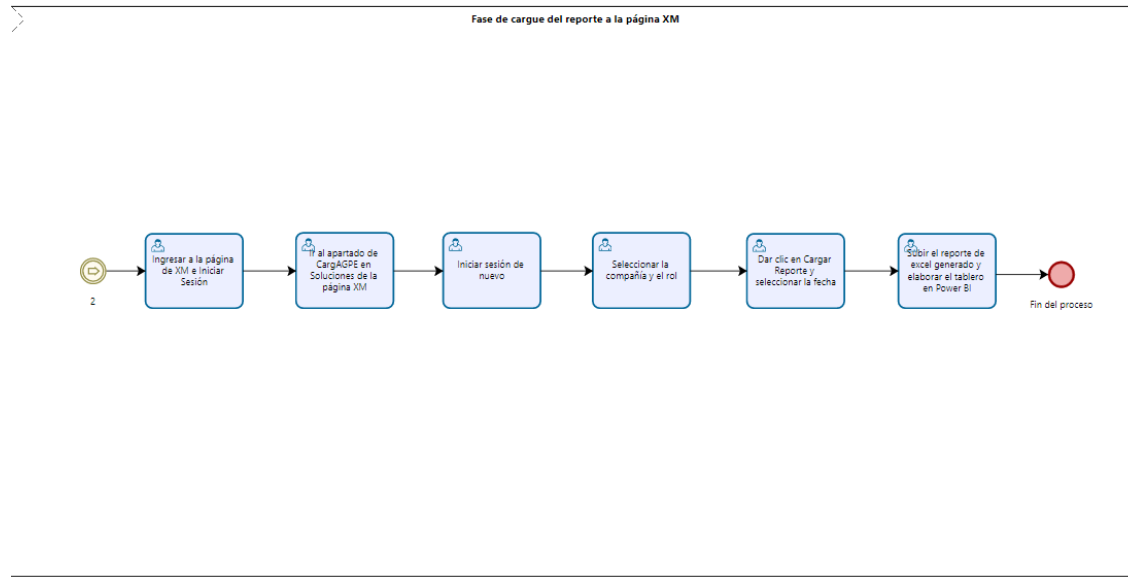


Nota. Figura elaborada por los autores. Representación de la fase de generación del archivo de reporte desde los datos procesados mediante Python en el flujo de automatización.

En la Figura 8 se ilustra la última etapa del proceso anterior, relacionada con la entrega del reporte a XM. Esta fase también depende completamente del usuario, quien debe ingresar a la plataforma de XM, iniciar sesión dos veces, seleccionar la compañía y el rol correspondiente, y proceder con la carga manual del archivo Excel generado. Una vez realizado este cargue, se utiliza el mismo archivo para alimentar los tableros de Power BI de manera externa. Esta etapa, además de ser repetitiva, conlleva riesgos de error humano al seleccionar fechas, archivos o compañías incorrectas, y dificulta la trazabilidad al no estar automatizada la integración con el sistema de visualización.

Figura 8

Diagrama BPMN: Fase de cargue del reporte



Nota. Figura elaborada por los autores. Representación del proceso automatizado que realiza el cargue del archivo generado en la plataforma XM como parte del flujo del RPA.

Posteriormente, como se mostró previamente en la Figura del diagrama BPMN del RPA (ver sección 6.1), se presenta el flujo BPMN automatizado correspondiente al nuevo proceso implementado con Power Automate. En este se destaca el uso de Power Automate Cloud para programar la ejecución del flujo de Power Automate Desktop, que es el encargado de abrir la terminal de comandos y ejecutar el script principal en Python, con toda la estructura del módulo. Una vez ejecutado el módulo y generado el reporte final, este es capturado y enviado automáticamente al usuario final por medio del flujo de escritorio.

Con esta automatización también se optimiza la etapa de visualización, ya que el archivo generado está estructurado para alimentar directamente el tablero de Power BI, eliminando la necesidad de construcción manual del informe, como se hacía en el proceso anterior.

3. Desarrollar las configuraciones en la herramienta:

Se desarrollaron dos flujos que se conectan entre sí: un flujo en Power Automate Cloud y otro en Power Automate Desktop, los cuales permiten la ejecución programada del módulo Python y el envío del archivo generado. La conexión entre estos dos flujos es clave para el funcionamiento de la solución y su implementación requiere configuraciones específicas en ambos entornos.

Power Automate Cloud:

Este es un flujo simple con solo dos tareas, se encarga de activar el flujo de escritorio debido a que esta tarea no está incorporada en Power Automate Desktop a fecha de hoy, siguiendo esta estructura:

En la Figura 9 se puede observar cómo se configura el intervalo de ejecución (en este caso mensual) mediante la tarea Recurrence, y la invocación del flujo de escritorio con sus respectivos campos:

- **Conectar:** mediante que método se quiere hacer la conexión.
- **Máquina o grupo de máquinas:** la máquina que contiene el flujo de escritorio.
- **Dominio y contraseña:** aunque son campos opcionales, se recomienda su uso para

garantizar una conexión más estable.

Figura 9

Flujo de nube de power automate

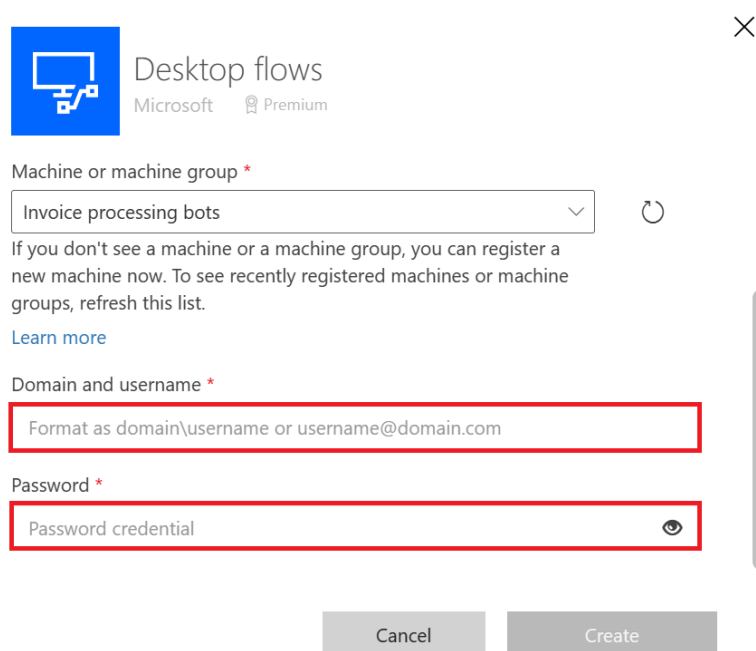
The image shows two screenshots from the Power Automate interface. The top screenshot is titled 'Recurrence' and shows the configuration for a recurring flow. It has two input fields: 'Intervalo' (Interval) with the value '1' and 'Frecuencia' (Frequency) with the value 'Mes' (Month). Below these fields is a link 'Mostrar opciones avanzadas' (Show advanced options). An arrow points down to the second screenshot, which is titled 'Flujos de escritorio' (Desktop flows). This screenshot shows the configuration for a desktop flow. It has several input fields: 'Conectar' (Connect) with the value 'Conectar con el nombre de usuario y contraseña' (Connect with username and password), 'Máquina o grupo de máquinas' (Machine or group of machines) with the value 'Seleccionar un elemento' (Select an item), 'Dominio y nombre de usuario' (Domain and username) with the value 'Formato de dominio\nnombre de usuario o' (Domain\nusername or), and 'Contraseña' (Password) with the value 'Credencial de contraseña' (Password credential). At the bottom of the form is a 'Crear' (Create) button.

Nota. Captura de pantalla elaborada por los autores. Representación del flujo de nube configurado en Power Automate para desencadenar la ejecución del proceso automatizado desde la nube.

Adicionalmente, en la Figura 10 se muestra la vista del asistente para registrar una nueva máquina desde Power Automate Cloud. Para establecer la conexión entre ambos entornos, es necesario proporcionar las credenciales de un usuario del equipo en el cual se encuentra Power Automate Desktop, siguiendo el formato dominio\usuario o usuario@dominio.com, junto con la contraseña correspondiente.

Figura 10

Conexión de máquina local



Desktop flows
Microsoft Premium

Machine or machine group *

Invoice processing bots

If you don't see a machine or a machine group, you can register a new machine now. To see recently registered machines or machine groups, refresh this list.

[Learn more](#)

Domain and username *

Format as domain\username or username@domain.com

Password *

Password credential

Cancel Create

Nota. Captura de pantalla elaborada por los autores. Ilustra el paso requerido para establecer la conexión entre el flujo de nube y la máquina local mediante credenciales de dominio en Power Automate.

Estas configuraciones permiten ejecutar flujos en modo atendido o desatendido, dependiendo de si se requiere la intervención de un usuario para iniciar sesión en la máquina o si la ejecución puede hacerse sin intervención humana. La ejecución en modo desatendido es ideal para automatizaciones programadas como la del presente proyecto, ya que no depende de que un usuario esté presente físicamente en el equipo (Microsoft, s.f.-b).

Power Automate Desktop:

Este flujo contiene las tareas necesarias para la ejecución del proyecto en la máquina local asociada al flujo en Power Automate Cloud. Aquí se realizan las actividades críticas, como ejecutar la aplicación Python mediante un comando DOS y enviar automáticamente el reporte generado

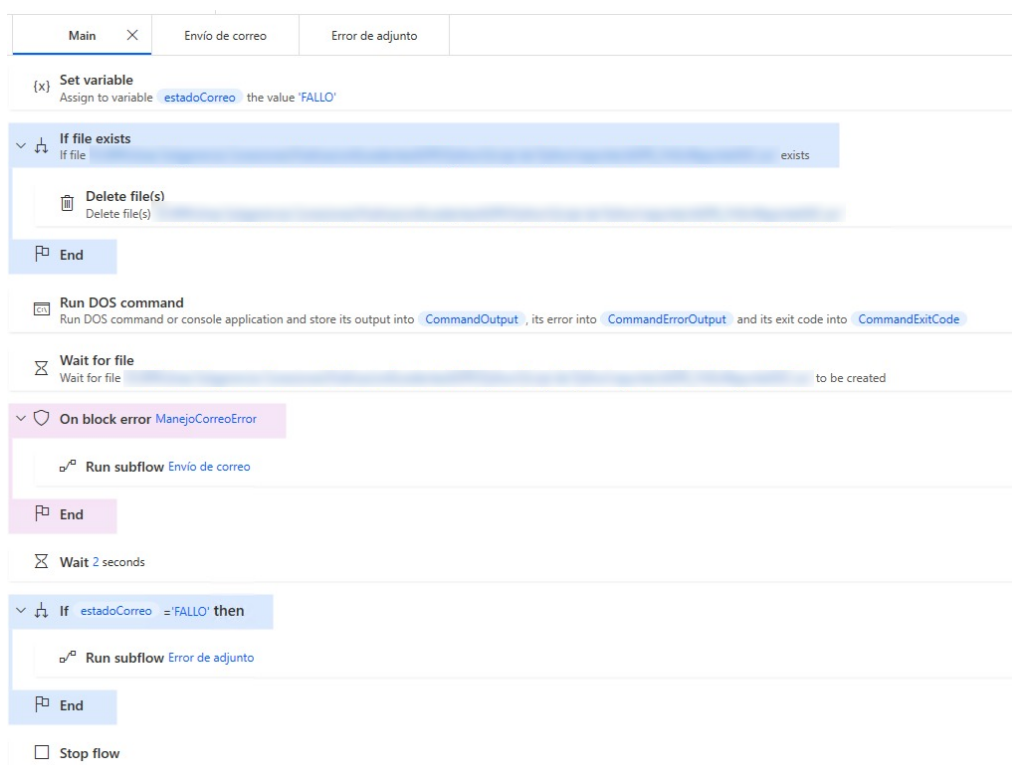
por correo electrónico al usuario encargado.

En la Figura 11 se muestra el flujo general desarrollado en Power Automate Desktop, el cual comienza configurando una variable inicial llamada estadoCorreo en el valor 'FALLO'. Esto permite verificar más adelante si el correo fue enviado exitosamente o no.

Luego, verifica si existe un archivo con un nombre específico (resultado de ejecuciones anteriores). De existir, lo elimina para evitar conflictos. Posteriormente ejecuta el script principal de Python mediante un comando DOS.

Figura 11

Flujo principal de escritorio en power automate



Nota. Captura de pantalla elaborada por los autores. Muestra el flujo principal de escritorio implementado en Power Automate, encargado de ejecutar el módulo Python, validar el reporte y

controlar posibles errores durante el proceso automatizado.

En la Figura 11 se observa la tarea mediante la cual se ejecuta el comando de DOS. Esta tarea permite abrir una ventana del símbolo del sistema de Windows para iniciar el script principal del módulo.

Figura 12

Tarea ejecutar comando de DOS

The screenshot shows a configuration window titled "Ejecutar comando de DOS". At the top, there is a description: "Ejecuta un comando de DOS o una aplicación de consola en modo invisible y recupera su resultado tras la finalización [Más información](#)". Below this, there are several input fields and controls:

- Comando DOS o aplicación:** A text input field with an information icon (i) to its right. Below it is a button labeled "Generar script con Copilot".
- Carpeta de trabajo:** A text input field with an information icon (i) to its right.
- Error tras tiempo de espera:** A toggle switch currently turned off, with an information icon (i) to its right.
- Avanzada:** A section header with a right-pointing chevron (>).
- Variables producidas:** A section header with a right-pointing chevron (>) followed by three buttons: "CommandOutput2", "CommandErrorOutput2", and "CommandExitCode2".

At the bottom of the window, there are three buttons: "Al producirse error" (with a shield icon), "Guardar" (in blue), and "Cancelar".

Nota. Captura de pantalla elaborada por los autores. Corresponde al componente “Ejecutar comando de DOS” en Power Automate Desktop, utilizado para ejecutar el módulo Python desde consola.

Son necesarios los siguientes campos:

- Comando DOS o aplicación: en este campo se especifica la ruta del ejecutable de Python y la del script main.py correspondiente al módulo del proyecto. Por ejemplo: "C:\Program Files\Python311\python.exe" "C:\RutaESSAProducción\Reporte Excedentes AGPE\main.py".
- Carpeta de trabajo: ruta de la carpeta raíz del módulo, necesaria para evitar errores al importar módulos internos durante la ejecución del script.

Una vez completada la ejecución del comando, el flujo continúa con una tarea de obtención de archivos desde una carpeta específica, donde se espera encontrar el reporte final generado. Este reporte es almacenado en una variable del entorno llamada %reporte%.

Subflujos para manejo de correo electrónico:

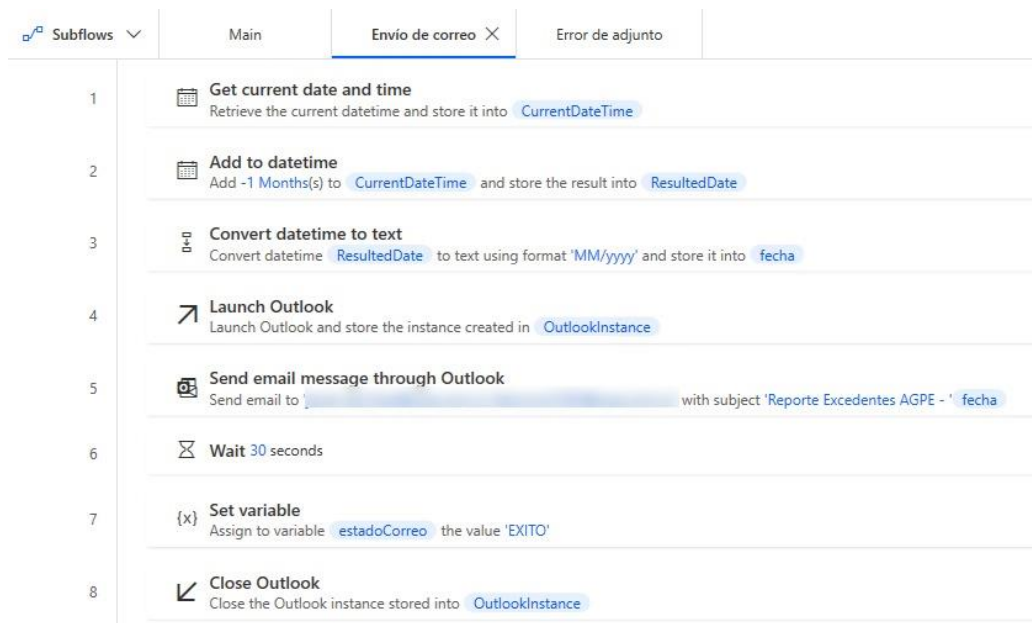
El flujo cuenta con dos subflujos específicos para manejar los correos electrónicos según el resultado obtenido durante la ejecución:

1. Envío de correo exitoso

Si el reporte se genera correctamente, el flujo ejecuta este subflujo. Inicialmente, obtiene la fecha actual, resta un mes y convierte esta fecha en texto para incluirla en el asunto del correo electrónico. Posteriormente, lanza una instancia de Outlook, envía el correo con el archivo adjunto y finalmente actualiza la variable estadoCorreo a 'EXITO'.

Figura 13

Subflujo envío de correo



Nota. Captura de pantalla elaborada por los autores. Corresponde al subflujo de Power Automate Desktop para el envío automático de correo electrónico en caso de ejecución exitosa.

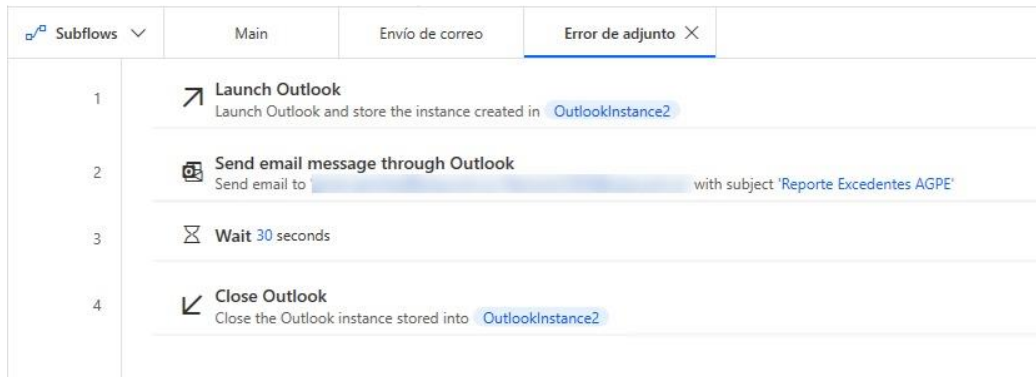
2. Error en adjunto

En caso de que el archivo adjunto no se genere adecuadamente, se ejecuta el subflujo para reportar este inconveniente. Aquí simplemente se lanza Outlook, se envía una alerta por correo al usuario indicando que el adjunto falló y posteriormente se cierra Outlook.

En la Figura 13, se detallan las siguientes tres tareas relacionadas con el uso de la aplicación Outlook: inicializar la instancia de la aplicación, enviar el correo electrónico con el archivo adjunto y cerrar la aplicación.

Figura 14

Subflujo error de adjunto



Nota. Captura de pantalla elaborada por los autores. Corresponde al subflujo de Power Automate Desktop utilizado para enviar una alerta por correo electrónico en caso de error al generar el archivo adjunto.

Algunos de los campos requeridos son:

- **Cuenta:** dirección del remitente (usuario técnico ESSA).
- **A:** destinatarios del reporte (usuarios finales del proceso en ESSA).
- **Asunto:** texto del asunto, usualmente incluyendo el nombre del reporte y la fecha (ejemplo: “Reporte Marzo 2025”).
- **Datos adjuntos:** se utiliza la variable %reporte% para adjuntar el archivo final generado.

Adicionalmente, el flujo contiene dos subflujos dedicados al manejo de errores críticos que pueden surgir durante la ejecución:

- El primer subflujo actúa como mecanismo de recuperación en caso de que la tarea “Obtener archivos de una carpeta” no encuentre el reporte. En ese caso, se reintenta ejecutar el módulo mediante el comando DOS y se notifica al encargado de soporte mediante correo

electrónico.

- El segundo subflujo se activa cuando existen problemas de conexión a las bases de datos (PrimeRead o SAC), ya sea porque estas estén caídas o porque hayan cambiado las cadenas de conexión. En esta situación, se envía una notificación al usuario final y al soporte técnico, indicando que la extracción deberá realizarse de forma manual para continuar con la generación del reporte.

Cabe aclarar que el manejo de errores relacionados con la disponibilidad de las bases de datos fue excluido del alcance del proyecto, dado que su resolución depende de la gestión y mantenimiento interno de la ESSA sobre sus sistemas de información.

4. Realizar pruebas

Se realizaron una serie de pruebas en el entorno de desarrollo de ESSA con el fin de verificar el correcto funcionamiento de cada uno de los flujos diseñados, tanto de forma individual como integrados entre sí.

Durante las primeras ejecuciones del flujo de Power Automate Cloud, se presentó un error de tipo timeout, el cual indicaba que la nube no lograba establecer conexión con la máquina que contenía el flujo de escritorio. Este error arrojaba el siguiente mensaje:

“ConnectionNotEstablished: ninguno de los agentes de escucha conectados aceptó las conexiones dentro del tiempo de espera permitido. Compruebe que la máquina está en línea.”

Este error fue diagnosticado como un problema de conectividad directa entre la nube de Power Automate y la máquina local, específicamente por restricciones en el firewall corporativo. Tal como lo señala la documentación oficial de Microsoft, este tipo de error puede surgir cuando

el servicio UIFlowService no puede establecer conexión con los servicios de retransmisión de Azure debido a bloqueos en las conexiones salientes necesarias para su operación (Microsoft, s.f.-c).

Este inconveniente fue resuelto con el apoyo del equipo de ciberseguridad de ESSA, quienes habilitaron las conexiones requeridas para que Power Automate Cloud pudiera comunicarse correctamente con el agente local, restableciendo así la conectividad.

Por otro lado, en el flujo de escritorio se identificó un fallo en la tarea llamada Enviar correo electrónico, ya que utilizaba una versión obsoleta de la aplicación Correo de Windows, la cual no procesaba correctamente los archivos adjuntos. Por este motivo, se optó por utilizar directamente la aplicación de Outlook para realizar el envío del reporte final, iniciando una instancia desde el escritorio.

Finalmente, una vez solucionados los errores individuales de ambos flujos, se procedió a realizar pruebas de integración, obteniendo resultados positivos en los que todo el RPA se ejecutó correctamente, generando el reporte final y enviándolo al destinatario designado.

5. Implementar el RPA en producción

Una vez finalizadas las pruebas en el entorno de desarrollo, se procedió con la migración del RPA desarrollado en Power Automate al entorno de Producción.

En el caso del flujo de Power Automate Desktop, el proceso de migración no presentó mayores inconvenientes. Fue suficiente con compartir el flujo desde el entorno de desarrollo hacia el entorno de producción, conservando su configuración original.

Para el flujo en Power Automate Cloud, se optó por crear una solución que incluyera tanto

el flujo como la conexión previamente establecida con la máquina de escritorio. Esta solución fue exportada como un archivo comprimido (.zip) para ser transferido al usuario técnico encargado del entorno productivo. Este procedimiento fue necesario debido a que el usuario de desarrollo no contaba con las licencias requeridas para ejecutar flujos que contienen acciones premium. Una vez cargado el archivo en el entorno de producción mediante la opción de importar soluciones que ofrece Power Automate, se modificaron las credenciales de ejecución (usuario y contraseña) para adaptarlas al perfil técnico habilitado en dicho entorno.

Durante el proceso de despliegue, se identificó un inconveniente relacionado con el tipo de ejecución configurada. Inicialmente, el flujo estaba configurado en modo Atendido, lo cual impedía que la ejecución ocurriera de forma automática sin que el usuario iniciara sesión previamente en la máquina. Aunque este comportamiento se había identificado desde etapas previas, la recomendación del equipo de TI de ESSA fue mantener inicialmente el modo Atendido, debido a la disponibilidad limitada de recursos asignados a usuarios con licencias para ejecuciones Desatendidas.

Sin embargo, considerando que el proceso debía ejecutarse de forma autónoma una vez al mes, se solicitó autorización para migrar la ejecución al usuario técnico con capacidad para correr flujos en modo Desatendido. Una vez aprobada esta solicitud, se desplegó la solución en dicho usuario, realizando las respectivas pruebas en el entorno de Producción, las cuales arrojaron resultados satisfactorios, cumpliendo con los objetivos establecidos tanto por el equipo de desarrollo como por el usuario final.

6. Entrega de documentación

Como parte del proceso de implementación de la solución, se elaboró y entregó la

documentación necesaria para orientar la comprensión técnica y facilitar el mantenimiento a largo plazo del desarrollo. Esta documentación incluyó dos manuales y un video explicativo, en los cuales se describe el funcionamiento general del sistema, los módulos implementados en Python y las acciones requeridas en caso de que se deban modificar las credenciales de las bases de datos.

El primer documento correspondió al manual técnico y de operación del proyecto, donde se detalla la arquitectura general de la solución, el flujo de trabajo, los recursos empleados, los accesos necesarios para su ejecución, y las condiciones requeridas para su correcto funcionamiento en el entorno de producción.

El segundo documento corresponde al manual técnico de encriptación de credenciales, el cual especifica el procedimiento utilizado para el cifrado de las credenciales y de la cadena de conexión a las bases de datos de ESSA. En este se describe el uso de la librería Fernet de Python y se explican los pasos necesarios para la generación de claves aleatorias, el proceso de encriptación de las credenciales y las recomendaciones sobre la estructura de carpetas y archivos que se deben mantener para su correcto uso.

Adicionalmente, se grabó y entregó un video explicativo en el que se describe de manera detallada el funcionamiento de los principales módulos del proyecto, explicando su código fuente y las funcionalidades desarrolladas, con el fin de reforzar el entendimiento y apoyar futuras labores de mantenimiento.

Cabe destacar que el código fuente desarrollado incluye documentación interna embebida, una práctica recomendada por la especificación PEP 257 de Python, que consiste en escribir explicaciones directamente en las funciones y clases del código para facilitar su comprensión y uso posterior por parte de otros desarrolladores o usuarios técnicos (Python Software Foundation,

s.f.):

6.1.4. Power BI

Para facilitar el análisis de la gestión de los clientes de autogeneradores, solicitudes y excedentes, se utilizó el servicio de Power BI, desarrollando un informe interactivo que integra múltiples fuentes de datos para cumplir los objetivos planteados. Este informe permite la visualización actualizada de indicadores relevantes sobre los procesos de los clientes de autogeneradores, los excedentes generados, los autogeneradores activos, entre otros aspectos del seguimiento de los AGPE, facilitando la toma de decisiones y mejorando el control sobre la gestión de los autogeneradores en ESSA.

Orígenes de datos integrados:

- **CSV generado por el módulo de Python:** Este archivo contiene el consolidado mensual de excedentes generados por los AGPE, el cual es calculado a partir de los datos de consumo extraídos desde la base de datos PrimeRead. Sirve como insumo principal para el monitoreo de energía inyectada al sistema.
- **Base de datos del SAC:** Se conecta directamente a esta base de datos para consultar cinco (5) tablas clave. A partir de ellas, se obtiene información administrativa y técnica de los autogeneradores, incluyendo información del cliente, el estado de vinculación, la trazabilidad a detalle de cada solicitud asociada a un proceso, características técnicas del autogenerador, y el nombre asociado al usuario responsable dentro de ESSA de gestionar la solicitud.
- **DataLake de ESSA:** El tablero establece conexión con un conjunto de datos estructurados en formato .parquet, alojado dentro de un contenedor de almacenamiento (blob

container) en la cuenta de almacenamiento que respalda el DataLake corporativo de ESSA. Este conjunto de datos agrupa la información de procesos de autogeneradores, incluyendo solicitudes en trámite o finalizadas. La información disponible abarca aspectos como el municipio, promotor, responsable técnico y la evolución temporal de procesos, permitiendo realizar análisis históricos y geográficos sobre la expansión y gestión de los autogeneradores en el territorio.

Contenido y visualizaciones del informe:

El informe fue diseñado con una estructura de navegación que permite acceder fácilmente a las diferentes páginas que lo componen: solicitudes en trámite, solicitudes finalizadas y autogeneradores en operación. Desde la pantalla de inicio, los usuarios pueden seleccionar la sección de su interés mediante botones de acceso rápido, lo que ofrece una buena experiencia de uso.

Además, en la parte izquierda del informe se incorporó un menú lateral, que permite desplazarse entre información relacionada con la sección como en "Finalizados" que se encuentra "Indicadores" y "Promotores". Cada una de estas secciones cuenta con su propio submenú horizontal, ubicado en la parte superior del panel principal, el cual organiza la información en pestañas temáticas, facilitando la segmentación y el análisis detallado.

En cuanto a las visualizaciones, se implementaron distintos tipos de indicadores permitiendo el análisis de datos:

- **Indicadores KPI (Key Performance Indicators):** Utilizados para resaltar cifras clave como el número total de solicitudes, tiempo promedio de respuesta, porcentaje de cumplimiento del tiempo meta y cantidad de autogeneradores en operación.

- **Gráficos de barras:** Permiten visualizar distribuciones como el número de clientes por municipio, solicitudes por tipo de proceso, y procesos por promotor.
- **Gráficos de líneas:** Empleados para mostrar la evolución mensual o anual de indicadores como solicitudes recibidas, procesos finalizados o número de clientes activos, facilitando el análisis de tendencias.
- **Gráficos circulares:** Representan proporciones como el cumplimiento por tipo de proceso, distribución de entrega de excedentes y distribución de clientes por zona.
- **Segmentadores (slicers):** Se habilitaron filtros dinámicos por periodo, municipio, estado del proceso, nombre del cliente, ID del proceso, comercializador, entre otros, lo que permite ajustar las visualizaciones según las necesidades específicas de análisis.

En la Figura 15 se presenta una vista general de una de las páginas del informe, allí se observan múltiples visualizaciones integradas junto con los menús interactivos, lo que permite una exploración ágil y eficiente de los indicadores más relevantes para la gestión de los autogeneradores.

Figura 15

Visualización del histórico de procesos



Nota. Visualización desarrollada por los autores en Power BI. Muestra una de las páginas del informe interactivo construido como parte del proyecto. Incluye gráficos de líneas, pastel y barras sobre los procesos finalizados de los clientes AGPE.

Funcionalidades de filtrado

El informe fue diseñado con múltiples filtros, permitiendo así al usuario analizar la información de forma dinámica, intuitiva y ajustada a las necesidades del proceso.

Para ello, como se mencionó anteriormente, se implementaron los segmentadores interactivos (slicers) en cada página, los cuales permiten filtrar los datos según criterios específicos del contexto de análisis.

En la sección de los autogeneradores en operación, se incluyeron slicers para el número de cuenta, nombre del cliente, comercializador, municipio, mes y año, permitiendo segmentar rápidamente a los AGPE activos por ubicación o agente comercial. En la sección de procesos finalizados, se habilitaron filtros por grupo de procesos semana, mes, año, número de proceso y

nombre del promotor, facilitando el análisis del desempeño y cumplimiento según el tipo de proceso y facilitando el reconocimiento de los mayores promotores de autogeneradores. Por último, en la sección de procesos en trámite, los filtros disponibles permiten seleccionar por número de proceso, ID del cliente, tipo de proceso, acción en la que se encuentra el proceso y municipio.

Los slicers permiten selección múltiple y los que permiten filtrar por nombre, ID, o códigos están configurados con la opción de búsqueda activa, esto facilita la identificación de clientes o procesos específicos sin necesidad de recorrer toda la lista que suele ser bastante extensa.

Además, Power BI brinda la interacción entre visualizaciones (cross-filtering), de modo que, al seleccionar un elemento en un gráfico, el resto de las visualizaciones se actualiza automáticamente para reflejar los datos correspondientes. Esta funcionalidad mejora la exploración de comportamientos dentro del conjunto de datos.

También se aplicaron filtros a nivel de visualización y de página para asegurar la coherencia contextual. Por ejemplo, en la página de “En operación” se muestra únicamente información de los autogeneradores con estado activo; en la página de “Finalizados”, se limita información a procesos en los que su estado sea finalizado y se aplican filtros por acción en la que se encuentre el proceso según el tipo de revisión (documental o técnica); y en la sección de “Trámite”, los datos corresponden exclusivamente a procesos en curso.

Adicionalmente, cada sección cuenta con submenús horizontales que organizan la información en pestañas temáticas. Esto permite visualizar subconjuntos específicos de datos sin salir de la página, brindando una mayor fluidez en la navegación.

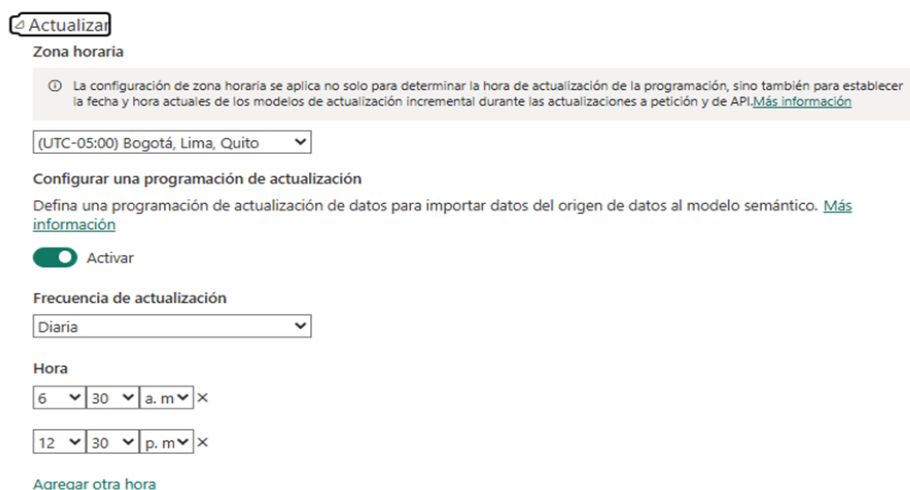
Publicación y actualización automática

El informe fue publicado en el servicio de Power BI Service, dentro del área de trabajo Seguimiento Autogeneradores, lo que permite su disponibilidad en línea para los usuarios autorizados dentro de la organización con quienes se haya compartido. Esta publicación garantiza que los responsables del seguimiento de los procesos de autogeneradores puedan acceder al informe desde cualquier dispositivo con conexión a internet, sin necesidad de manejar archivos locales.

Para asegurar que la información del tablero esté siempre actualizada, se configuró una actualización automática del modelo semántico, que es la estructura lógica de datos utilizada en el Power BI para representar relaciones y medidas de forma eficiente. Esta actualización se ejecuta dos veces al día, a las 6:30 a.m. y 12:30 p.m. (hora local, UTC-5), garantizando que los datos reflejados en el informe correspondan a un estado bastante reciente de los procesos.

Figura 16

Configuración de actualización del modelo semántico



The screenshot shows the 'Actualizar' (Update) configuration page in Power BI. It includes a 'Zona horaria' (Time zone) dropdown set to '(UTC-05:00) Bogotá, Lima, Quito'. A note explains that the time zone configuration applies to incremental update models. Below is the 'Configurar una programación de actualización' (Configure an update schedule) section, which is active. The 'Frecuencia de actualización' (Update frequency) is set to 'Diaria' (Daily). The 'Hora' (Time) section shows two scheduled times: 6:30 a.m. and 12:30 p.m. There is also a link to 'Agregar otra hora' (Add another time).

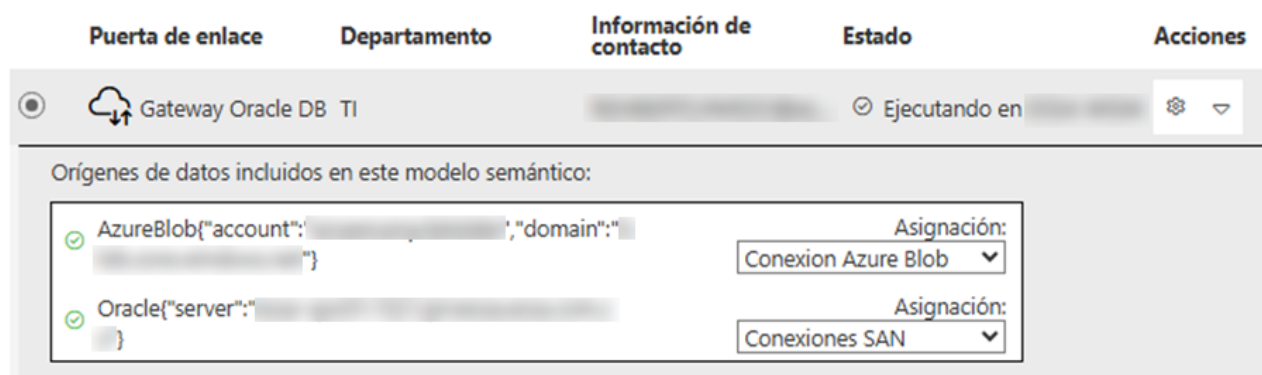
Nota. Captura de pantalla elaborada por los autores. Ilustra la configuración de programación de actualización del modelo semántico en Power BI Service como parte del prototipo implementado.

Como parte de la arquitectura de actualización, fue necesario integrar dos conexiones en el gateway corporativo:

- Uno para la conexión con la DataLake de ESSA, específicamente al contenedor en Azure Blob donde se encuentra el conjunto de datos histórico de solicitudes de autogeneradores.
- Otro para la conexión con la base de datos Oracle del SAC, desde donde se consulta información técnica y administrativa sobre los clientes de autogeneradores y sus procesos.

Figura 17

Configuración del gateway para conexiones a los orígenes de datos



Nota. Captura de pantalla elaborada por los autores. Muestra la configuración del gateway en Power BI Service utilizada para establecer las conexiones con los orígenes de datos implementados en el prototipo.

En cuanto a los datos procesados en Python, se implementó una solución en la cual el módulo de generación del reporte exporta automáticamente el DataFrame con los excedentes generados a un archivo .csv, el cual se guarda en una carpeta de SharePoint. Esta carpeta fue configurada como origen de datos en Power BI, permitiendo la integración automática del archivo sin necesidad de intervención manual, y asegurando así la sincronización entre el procesamiento

externo y la visualización dentro del informe.

Adicionalmente, se habilitó el envío de notificaciones automáticas en caso de errores durante la actualización. Estas alertas son remitidas al propietario del modelo y a los contactos técnicos responsables del monitoreo, permitiendo una respuesta oportuna ante cualquier error y manteniendo la confiabilidad operativa del sistema.

6.2. Técnicas de recolección de información

6.2.1. Origen de los datos

Para la construcción del reporte automatizado y la visualización en Power BI, se accedió a diferentes fuentes de datos internas de ESSA: PrimeRead y SAC, así como un conjunto de datos alojado en el DataLake corporativo, que contienen información técnica, operativa y descriptiva relacionada con los clientes AGPE y sus excedentes.

Esta información fue utilizada en dos frentes paralelos del proyecto:

- Por parte del módulo Python, con consultas SQL ejecutadas mediante el uso de la librería `oracledb` en modo grueso (ver sección 6.1.1).
- Por parte del Power BI, mediante conectores nativos en Power BI para acceder directamente a las fuentes internas (SAC y DataLake), además del archivo `.csv` generado desde Python, para la construcción de un modelo de datos visual, filtrable y actualizado.

Consultas implementadas en la automatización de Python y Power Automate

En total, se desarrollaron tres consultas SQL dentro de la automatización de Python y Power Automate: dos dirigidas a la base de datos PrimeRead y una a la base de datos SAC. A

continuación, se presenta una descripción de los datos extraídos en cada una de estas:

PrimeRead – Consulta de Excedentes

Consulta enfocada en la recuperación de los valores energéticos registrados por hora para cada cliente en el mes vencido. Los datos extraídos son transformados mediante una operación PIVOT para estructurarlos por hora (0 a 23) y se filtran por fecha de inicio y fin.

Tabla 2

Campos de Excedentes

Campo	Descripción
Device ID	Identificador del medidor
Channel ID	Canal de medición
Customer ID	Identificador del cliente
Date	Fecha de la lectura
Hora 0 a 23	Valores por cada hora del día
Tipo de energía	kWhR, kWhD, kVarhR, kVarhD

Nota. Tabla elaborada por autores. Campos consultados más relevantes de la tabla donde se encuentra la información de los excedentes de los AGPE.

PrimeRead – Consulta de Relaciones Técnicas

Consulta diseñada para complementar el reporte con atributos técnicos del sistema de

medición de cada cliente.

Tabla 3

Campos de relaciones técnicas

Campo	Descripción
Frontera	Punto de conexión del cliente
Código SIC	Código de identificación del cliente
Medidor	Número del medidor
Cuenta	Número de cuenta asociado
Marca del medidor	Marca del equipo
IP	Dirección IP del dispositivo

Nota. Tabla elaborada por autores. Campos consultados más relevantes de la tabla donde se encuentran las relaciones técnicas.

SAC – Consulta de Clientes y Atributos Descriptivos

Consulta desarrollada para obtener información descriptiva de los clientes AGPE, con fines tanto analíticos como de filtrado en el proceso automatizado.

Tabla 4

Campos de clientes y atributos descriptivos

Campo	Descripción
--------------	--------------------

Identificador	Identificador interno del cliente AGPE
Nombre	Nombre del cliente
Comercializador	Empresa comercializadora asociada
Nivel de tensión	Nivel de tensión eléctrica del cliente
Mercado destino	Segmento de mercado
Tipo de excedente	Tipo de energía excedente generada
Capacidad instalada	Potencia instalada (kW)
Fuente de energía	Tecnología de generación (solar, eólica, etc)
Fecha de entrada	Fecha de ingreso al sistema

Nota. Tabla elaborada por autores. Campos consultados más relevantes de la tabla donde se encuentran los detalles de los clientes de los AGPE.

En esta última consulta se implementó también un filtro por tarifas específicas (919 a 924), para segmentar únicamente los clientes autogeneradores a pequeña escala (AGPE) y validar su estado activo al momento de la ejecución.

Consultas utilizadas en la construcción del informe en Power BI

Para realizar el informe de Power BI se desarrollaron seis (6) consultas a la base de datos SAC, cinco de ellas se realizaban directamente del Power BI a la base de datos y una al del DataLake, pero este consultaba al SAC de manera periódica. Adicionalmente, se realizó una consulta a una carpeta de SharePoint y también se consumió una API pública. A continuación, se presenta una descripción de los datos extraídos en cada una de estas:

SAC - Consulta detalles administrativos y técnicos de los autogeneradores

Se utilizó para extraer atributos técnicos y administrativos de los autogeneradores, filtrando únicamente aquellos cuyo estado correspondía a “Generador Activo”. Esta información permitió caracterizar los AGPE activos, incluyendo su configuración y estado operativo.

Tabla 5

Campos de detalles administrativos y técnicos de los autogeneradores

Campo	Descripción
ID del cliente	Identificador interno del cliente AGPE
Tipo de generador	Clasificación si es AGPE con Potencia menor o igual que 0.1 MW o mayor
Tipo de tecnología	Clasificación del tipo de tecnología del autogenerador para generar energía
Entrega de excedentes	Indica si ese autogenerador presenta o no excedentes
Almacena energía	Indica si ese autogenerador almacena o no energía
Número del proceso	Identificador interno del proceso del autogenerador
Comercializador	Indica quién es el comercializador de ese

AGPE

Capacidad

Capacidad del AGPE

Fecha de conexión

Fecha de conexión al sistema

Nota. Tabla elaborada por autores. Campos consultados más relevantes de la tabla donde se encuentran los detalles administrativos y técnicos de los AGPE.

SAC - Consulta de la trazabilidad de los procesos

Contiene el historial detallado de acciones asociadas a cada solicitud de conexión. La consulta a esta tabla permitió construir la trazabilidad temporal de cada proceso, incluyendo fechas clave y responsables de cada solicitud del proceso.

Tabla 6

Campos de trazabilidad de los procesos

Campo	Descripción
Número del proceso	Identificador interno del proceso del autogenerador
Descripción de Acción	Nombre de la acción que se realiza para responder a solicitud
Tipo de Proceso	Describe que proceso es el solicitado
Fecha inicial	Fecha que entró la solicitud al sistema

Fecha final	Fecha en que se dio solución a la solicitud
Número de acción	Corresponde al orden secuencial de la acción dentro de un mismo proceso

Nota. Tabla elaborada por autores. Campos más relevantes consultados de la tabla donde se encuentran la trazabilidad de los procesos de los AGPE.

Con el fin de evitar la inclusión de registros irrelevantes para el análisis, la consulta fue filtrada cuidadosamente según el tipo de proceso, la acción ejecutada y la etapa del trámite, permitiendo así recuperar únicamente las acciones relevantes para el cálculo de tiempos y trazabilidad de los procesos AGPE.

SAC - Consulta estado en que se encuentra un proceso

Proporciona el estado de vinculación del proceso, clasificando cada caso en categorías como aprobado, no aprobado, pendiente, cancelado o en estudio.

Tabla 7

Campos del estado del proceso

Campo	Descripción
Número de proceso	Identificador interno del proceso del autogenerador
Estado de vinculación	Describe el estado de vinculación del proceso

Tipo de proceso Describe qué proceso es el solicitado

Nota. Tabla elaborada por autores. Campos consultados de la tabla donde se indica el estado de los procesos de los AGPE.

La inclusión de esta consulta fue necesaria ya que la información sobre el estado actual del proceso de vinculación no estaba disponible en las demás consultas realizadas. Este campo resultó esencial para segmentar los procesos y asegurar la coherencia contextual en el análisis de solicitudes activas o finalizadas.

SAC - Consulta datos complementarios de clientes

Contiene los datos generales del cliente asociado al autogenerador, tales como el nombre, número de cuenta y municipio, los cuales se utilizaron tanto para segmentación como para visualización personalizada.

Tabla 8

Campos complementarios de clientes

Campo	Descripción
ID de cliente	Identificador interno del cliente
Nombre	Nombre del cliente
Tarifa	Código del tipo de tarifa que maneja el cliente (necesario para conocer el comercializador)

Tipo de persona	Describe si es persona natural o jurídica
Clase de servicio	Describe qué clase de servicio está asociada al cliente
Estrato	Describe el estrato del cliente
Área	El área del cliente (urbano o rural)
Municipio	Municipio al que pertenece el cliente
Departamento	Departamento al que pertenece el cliente
Teléfono	Teléfono de contacto del cliente
Correo	Correo de contacto del cliente

Nota. Tabla elaborada por autores. Campos consultados complementarios de los clientes de los AGPE.

Para reducir el volumen de datos y asegurar la relevancia de los registros consultados, se aplicó un filtro que permitió recuperar únicamente los clientes que contaban con al menos un generador activo y cuya información estaba presente en la tabla de detalles administrativos y técnicos de los autogeneradores. Esta selección tuvo como propósito complementar la sección de análisis de los autogeneradores en operación dentro del informe.

SAC - Consulta nombre de encargados de la ESSA

Se utilizó para asociar a cada solicitud el responsable interno de ESSA encargado de su gestión.

Tabla 9*Campos de nombre de encargados*

Campo	Descripción
Usuario	Identificador interno del usuario ESSA
Nombre	Nombre del usuario ESSA

Nota. Tabla elaborada por autores. Campos consultados de la tabla donde se encuentran los encargados ESSA de los procesos AGPE.

DataLake - Consulta de los procesos históricos de los autogeneradores

Contiene información consolidada sobre los procesos de conexión de autogeneradores, tanto en trámite como finalizados.

Tabla 10*Campos de procesos históricos de los autogeneradores*

Campo	Descripción
Número de proceso	Identificador interno del proceso del autogenerador
ID de cliente	Identificador interno del cliente.
Tipo de proceso	Describe qué proceso es el solicitado
Fecha de inicio	Fecha en que el proceso entró al sistema
Fecha de vencimiento	Fecha en la que el proceso vence

Fecha final	Fecha en qué se dio respuesta al proceso
Nombre del promotor	Nombre del promotor asociado al autogenerador
Teléfono del promotor	Teléfono del promotor asociado al autogenerador
Municipio del promotor	Municipio del promotor asociado al autogenerador
Estado	Estado en que se encuentra el proceso
Nombre del cliente	Nombre del cliente
Municipio del cliente	Municipio donde se encuentra el cliente
Área del cliente	Área donde se encuentra el cliente (urbano o rural)
Usuario inicial	Usuario ESSA responsable del proceso
Usuario revisión	Usuario ESSA responsable si se requiere de técnico para responder el proceso

Nota. Tabla elaborada por autores. Campos consultados de la tabla donde se encuentran todos los procesos de los AGPE.

SharePoint - Excedentes generados por los autogeneradores

Este conjunto de datos corresponde al archivo .csv generado automáticamente por el módulo Python, a partir de los datos extraídos desde PrimeRead y procesados para obtener el consolidado mensual de excedentes energéticos de los AGPE.

Tabla 11

Campos de excedentes generados por los autogeneradores

Campo	Descripción
ID del cliente	Identificador interno del cliente AGPE
Tipo de excedente	Indica el tipo de excedente generado
Fecha	Mes y año del registro
Nombre	Nombre del Cliente
Nivel de tensión	Indica el nivel de tensión del AGPE
Capacidad	Capacidad instalada del AGPE
Total	Total de energía generada en ese periodo por el AGPE

Nota. Tabla elaborada por autores. Campos consultados del archivo csv generado por el módulo de Python donde se encuentran los excedentes generados por los AGPE.

API pública de Nager.Date - Fechas de Festivos por año en Colombia

Para el cálculo de indicadores relacionados con la duración de los procesos (en días hábiles), se realizó una conexión mediante Power Query a la URL <https://date.nager.at/api/v3/publicholidays/{año}/CO>, para obtener los días festivos oficiales de Colombia entre los años 2018 y 2035. Estos datos fueron utilizados para construir la tabla de festivos y alimentar la lógica de calendario laboral del informe.

Tabla 12

Campos de fechas de festivos por año

Campo	Descripción
date	Fecha
localName	Nombre del festivo
countryCode	Indicador del país

Nota. Tabla elaborada por autores. Campos consultados del API donde se encuentran todas las fechas festivas de Colombia.

La recopilación y estructuración de estas consultas permitió consolidar un modelo de datos robusto y alineado con los objetivos del informe, facilitando el análisis segmentado de los procesos, clientes y excedentes de los autogeneradores. Cada consulta fue diseñada para aportar información específica y complementaria, asegurando consistencia, trazabilidad y contextualización en la visualización de resultados.

Nota. Por motivos de confidencialidad, el contenido completo de las sentencias SQL, como nombres de tablas internas y de atributos de estas, no se presenta en este documento. No obstante, su estructura y propósito fueron revisados y validados por el equipo técnico de ESSA.

6.2.2. Validación de los datos

Durante la ejecución del proceso automatizado se implementaron varias validaciones para asegurar la calidad y coherencia de los datos utilizados en el reporte final. Estas validaciones fueron integradas directamente en el módulo Python, específicamente en la etapa de tratamiento inicial de los datos, garantizando así que la información cumpliera con las condiciones técnicas necesarias antes de avanzar en el proceso.

Una de las validaciones principales se realiza en el módulo de procesamiento de matrices, donde se verifica que cada cliente autogenerador tenga información suficiente para los cuatro tipos de energía requeridos. Para esto, se aplica la siguiente condición:

$$\text{Cantidad de días nulos} \geq 4 \times \text{días del mes}$$

Con esta validación, se asegura que solo sean considerados aquellos clientes que cuenten con datos completos y consistentes durante el mes analizado.

En el módulo que genera el reporte se validan adicionalmente los totales diarios de energía consumida y generada para cada cliente. Si se encuentran inconsistencias, como situaciones donde un cliente genera más energía de la que consume, se aplica automáticamente una lógica de ajuste. Además, se utiliza una regla específica para comparar el excedente generado por hora contra la capacidad instalada del cliente, lo que impide reportar valores que excedan la capacidad real del equipo:

$$\text{Si } E_x > C, \text{ entonces } E_x^{\text{ajustado}} = C$$

donde E_x es el excedente generado por hora, y C la capacidad instalada declarada por el cliente.

Otra validación clave fue la consistencia entre los datos técnicos provenientes de PrimeRead y los datos descriptivos del SAC. Al realizar este cruce, si no se encontraba una correspondencia directa (por ejemplo, un número de cuenta sin coincidencia en SAC), automáticamente se descartaba dicho cliente para prevenir errores en las etapas posteriores del procesamiento.

Todas estas validaciones fueron integradas directamente en los módulos del proyecto con

el objetivo de garantizar que el archivo final presentado a XM sea coherente, confiable y cumpla con los criterios técnicos definidos. Así mismo, esta estrategia permitió reducir considerablemente la intervención manual en el proceso y agilizó significativamente la ejecución mensual del reporte.

Durante el desarrollo del informe también se implementaron validaciones adicionales orientadas a asegurar la integridad y coherencia de los datos utilizados en las visualizaciones de Power BI. Estas revisiones se realizaron tanto en el entorno de Power Query como en etapas previas de procesamiento, incluyendo revisiones manuales y controles de calidad.

Al relacionar la tabla de usuarios de ESSA con otras entidades mediante un campo clave, se detectaron problemas en la cardinalidad de las relaciones. Aunque visualmente los valores coincidían, algunos registros tenían espacios invisibles al final del texto, causando conflictos al establecer relaciones "uno a muchos". Para corregir esta situación, se utilizó la función `Text.Trim` en Power Query seguida de la operación `Distinct`, eliminando los espacios no visibles y estandarizando los valores.

En la revisión de la trazabilidad de los procesos se identificaron casos en los que la fecha final de una acción era anterior a la fecha inicial registrada, afectando cálculos de días hábiles y el cumplimiento del proceso. Para resolverlo, se ajustó la consulta a la tabla de trazabilidad para tomar como fecha inicial la fecha final de la acción anterior, determinada por el número secuencial dentro del proceso.

Durante la comparación de procesos históricos del DataLake y procesos activos se descubrió que algunos AGPE activos no tenían correspondencia en los históricos. Esto se debía a que antes del año 2023, ciertos procesos no incluían el prefijo "46" como identificador, lo que ocasionaba que fueran omitidos en consultas iniciales. Este problema fue resuelto aplicando un

join entre las tablas utilizando el ID del cliente, lo que permitió integrar correctamente los registros faltantes.

También se verificó la integridad referencial de campos clave, como el ID del cliente, para asegurar que no hubiera valores vacíos ni inconsistencias relevantes. Asimismo, se hicieron validaciones directas con el usuario experto encargado del proceso para confirmar la exactitud de campos clave como estados de vinculación, fechas de inicio y finalización, responsables asignados y comercializadores correspondientes a cada AGPE.

Para mejorar la calidad del análisis, se aplicaron filtros desde Power Query excluyendo procesos en estado “Cancelado”, puesto que el interés se centraba exclusivamente en procesos en trámite o finalizados, evitando incluir procesos que ESSA utilizaba como pruebas.

Al exportar datos desde el DataLake en formato .csv, se encontró que el campo “Observación”, que contenía textos extensos y caracteres especiales, generaba un desplazamiento de columnas afectando la integridad del archivo. Este inconveniente fue resuelto cambiando la configuración en Azure Data Factory para exportar los datos en formato .parquet, un formato que mantiene una estructura definida y soporta campos complejos, además de optimizar la velocidad de carga en Power BI.

Adicionalmente, se validó manualmente la información obtenida de la API Nager.Date sobre los días festivos de Colombia desde 2018 hasta 2035. Esto permitió construir una tabla precisa de días festivos, la cual se utilizó en la lógica para el cálculo de días hábiles por proceso, corroborada también mediante casos reales.

Finalmente, se realizaron validaciones específicas a las visualizaciones generadas para asegurar que tuvieran sentido lógico y contexto adecuado. Por ejemplo, en la sección de procesos

finalizados del informe se verificó que solo aparecieran estados como “Aprobado”, “No aprobado” o “Información de estudio entregada”.

Todas estas validaciones fueron determinantes para asegurar la calidad de los datos presentados en el reporte final, garantizando que los indicadores y visualizaciones se basaran en información coherente, actualizada y representativa de los procesos reales, fortaleciendo así la confiabilidad del análisis realizado.

6.2.3. Estrategias de extracción

Al momento de construir el módulo, se definieron estrategias específicas para extraer solamente la información necesaria de las bases de datos, evitando traer volúmenes innecesarios que pudieran afectar el rendimiento del proceso. Las consultas SQL se diseñaron de forma que permitieran filtrar los datos del mes vencido, utilizando parámetros de fecha definidos dinámicamente desde el script de Python.

En el caso de PrimeRead, se configuraron filtros en las cláusulas WHERE para limitar la consulta al rango de fechas correspondiente al mes anterior. Esto ayudó a reducir el tiempo de ejecución y optimizar el uso de recursos del sistema. Además, se aplicó una transformación con PIVOT sobre los registros horarios, lo que permitió reorganizar los datos por hora (de 0 a 23) y facilitar su posterior tratamiento en el archivo de reporte.

La consulta de relaciones técnicas también fue ajustada para recuperar únicamente los registros relevantes de ese mismo periodo, sin necesidad de recorrer toda la base histórica. Esta decisión ayudó a reducir la carga sobre el servidor de Oracle, especialmente teniendo en cuenta que se trabajaba en entornos de producción.

Para la base de datos SAC, se incorporó un filtro adicional que permitiera seleccionar únicamente a los clientes asociados a un grupo específico de tarifas (919 a 924), correspondiente a los autogeneradores a pequeña escala (AGPE). Gracias a este filtro, fue posible reducir la muestra directamente desde el origen, evitando realizar ese filtrado posteriormente en memoria.

En lo que respecta a la consulta de los procesos históricos de autogeneradores, esta información se obtiene desde un archivo estructurado en formato .parquet, alojado en el Azure Blob Storage que respalda el DataLake corporativo de ESSA. La conexión con Power BI se estableció mediante el conector nativo de Azure Blob Storage, accediendo a un conjunto de datos previamente generado desde Azure Data Factory.

La consulta sobre este archivo fue diseñada para combinar dos subconjuntos de registros mediante UNION. El primero recupera todos los procesos cuyo tipo tiene el prefijo “46”, correspondiente a procesos AGPE convencionales. El segundo conjunto extrae procesos asociados a los autogeneradores activos presentes en la vista de detalles administrativos y técnicos, con el objetivo de complementar aquellos registros que no contenían información completa, como el nombre del cliente, municipio, contacto, entre otros. Esta estrategia permitió asegurar que ningún proceso relevante quedara fuera del análisis.

Posteriormente, en Power Query se aplicaron transformaciones adicionales para depurar los datos, eliminando columnas que no aportaban valor analítico o que se encontraban completamente vacías, con el fin de optimizar el modelo semántico y mejorar el desempeño general del informe.

Adicionalmente, en la consulta de detalles administrativos y técnicos de los autogeneradores, se incluyó un filtro por estado para recuperar únicamente aquellos con estado

“Generador activo”, descartando registros inactivos o en configuración. En la consulta de clientes, se implementó una cláusula JOIN entre la vista de clientes y detalles administrativos y técnicos de los autogeneradores, limitando el resultado a los clientes que efectivamente tienen un generador asociado. De este modo, se evitó traer todo el volumen de clientes de la base y se garantizó la relevancia de los datos extraídos.

Para la tabla de trazabilidad de acciones del proceso, se desarrolló una consulta compuesta con múltiples CTEs (Common Table Expressions), mediante la cual se filtraron únicamente las acciones relevantes para el análisis. Entre las funciones utilizadas destacan ROW_NUMBER() para numerar secuencialmente las acciones por proceso y tipo, y cláusulas condicionales para incluir solo las acciones clave, como “Respuesta Solicitud Insumo”, “Respuesta formulario AGPE” o “Ingreso visita conexión AGPE”. Además, se implementó una lógica adicional para recuperar la acción inmediatamente anterior a cada una de estas, lo cual permitió calcular correctamente los intervalos de tiempo entre acciones dentro del proceso.

El archivo .csv generado por el módulo Python ya contiene un conjunto de datos procesados y filtrados, correspondiente al consolidado mensual de excedentes de los AGPE. Se estableció una carpeta en SharePoint como ubicación de carga automática del archivo, donde la conexión en Power BI se realizó utilizando el conector SharePoint Folder, permitiendo la integración sin necesidad de intervención manual ni configuración adicional de gateways.

Finalmente, la información de festivos oficiales en Colombia fue obtenida directamente desde la API pública de Nager.Date. La extracción se realizó en Power Query mediante la función Web.Contents, combinada con Json.Document, para importar la estructura de días no laborables entre 2018 y 2035, los cuales fueron utilizados en la lógica de cálculo de días hábiles aplicada al

análisis de tiempos de respuesta.

Todas las consultas fueron probadas previamente en el entorno de desarrollo, verificando su rendimiento y validando que los resultados cumplieran con los criterios establecidos para el reporte mensual. Gracias a esta estrategia, el proceso automatizado logró un tiempo de ejecución razonable y se garantizó que los datos extraídos fueran exactamente los necesarios, sin cargas innecesarias ni reprocesos.

6.3. Procedimientos de validación y tratamiento de datos

Durante la ejecución del proceso se implementaron diversos procedimientos específicos para asegurar la calidad y coherencia de los datos desde el momento mismo en que se extraen y durante su posterior tratamiento. Estas validaciones quedaron integradas directamente en los módulos desarrollados en Python, facilitando la detección y corrección automática de cualquier inconsistencia.

En primer lugar, el módulo de procesamiento de matrices realiza una validación clave, verificando que cada cliente cuente con datos suficientes para los cuatro tipos de energía durante todo el mes. Aunque este procedimiento se mencionó previamente, aquí cobra especial relevancia porque forma parte integral del conjunto de filtros que aseguran la integridad de los datos antes de avanzar en el procesamiento.

Adicionalmente, se implementó una validación cruzada entre los datos técnicos obtenidos de PrimeRead y la información descriptiva proveniente del SAC. Si al realizar este cruce faltaba información esencial, como la cuenta asociada a un cliente, dicho cliente se descartaba inmediatamente. Esto previene posibles errores en las siguientes fases y garantiza que solo se consoliden datos completos y precisos en el reporte final.

En el módulo encargado de generar el reporte también se realizaron validaciones específicas. Una de ellas es la comparación automática entre la energía consumida y generada por cada cliente, permitiendo detectar inconsistencias rápidamente. En aquellos casos donde la energía generada superaba lo que el cliente consumía o excedía la capacidad instalada declarada, el módulo ejecutaba ajustes automáticos. Esto garantiza que los valores informados estén siempre dentro de límites realistas y permitidos por la empresa.

Paralelamente, se implementó un mecanismo adicional mediante la generación de un archivo llamado logs.log. Este archivo documenta de forma detallada cualquier error o anomalía detectada durante la ejecución del proceso, incluyendo problemas en la conexión a bases de datos, errores en estructuras o falta de archivos requeridos. Este registro ha resultado particularmente útil para realizar trazabilidad y aplicar correcciones rápidas, especialmente durante las etapas de pruebas o ejecución en ambiente productivo.

En cuanto a la preparación y validación adicional de datos en Power Query, se llevaron a cabo procedimientos específicos para garantizar la integridad y usabilidad del modelo de datos en Power BI. Inicialmente se realizó una estandarización de tipos de datos en campos clave. Aquellos utilizados para filtrado en segmentadores interactivos —como número de cuenta, número de proceso o ID de cliente— fueron convertidos de tipo numérico a texto, habilitando así la funcionalidad de búsqueda directa. Del mismo modo, las columnas que inicialmente contenían marcas de tiempo fueron transformadas al formato fecha, simplificando posteriores análisis y segmentaciones por día, mes o año.

También se realizó un tratamiento cuidadoso de columnas innecesarias o con valores nulos que no aportaban valor analítico al modelo. En particular, se reemplazaron los valores nulos en

campos booleanos o de tipo "sí/no", como "Obligado a facturar" o "Almacena energía", asignando "NO" como respuesta predeterminada. Este procedimiento proporcionó consistencia adicional y mejoró la claridad del análisis posterior.

Además, se aplicaron técnicas de normalización y categorización para facilitar la visualización de los datos en Power BI. Por ejemplo, mediante la función Replace Values, etiquetas como “Estudio no aprobado” o “Solicitud incompleta” fueron unificadas bajo categorías generales como “No aprobado”, simplificando la comprensión visual de los resultados.

Respecto a la depuración de duplicados, se identificaron y corrigieron casos puntuales en la tabla de usuarios de ESSA, causados por espacios invisibles al final del texto. La solución implementada consistió en utilizar Text.Trim seguido de la operación Distinct, logrando así una correcta configuración de las relaciones necesarias en el modelo de datos.

Para los datos provenientes de la API utilizada en la obtención de días festivos en Colombia, se realizó un proceso específico desde Power Query. Los resultados de la consulta (mediante la función Web.Contents) se recibían en formato lista, que posteriormente se convertía en tabla usando la función Table.FromList. Seguidamente, esta tabla era expandida para extraer columnas relevantes como la fecha, nombre local y país. Las columnas no necesarias se eliminaron y se ajustaron los tipos de datos finales, garantizando su adecuada integración con el calendario laboral del informe.

Figura 18

Elementos estructurados como registros



The image shows a screenshot of a Power Query table. The table has a header row with a green background. The header row contains the text 'ABC 123' in the first column and 'Column1' in the second column. Below the header, there are 18 rows of data, each containing the word 'Record' in the second column. The rows are numbered 1 through 18 in the first column.

ABC 123	Column1
1	Record
2	Record
3	Record
4	Record
5	Record
6	Record
7	Record
8	Record
9	Record
10	Record
11	Record
12	Record
13	Record
14	Record
15	Record
16	Record
17	Record
18	Record

Nota. Captura de pantalla elaborada por los autores en Power Query. Representa la estructura inicial de los datos devueltos por la API, en forma de registros que deben ser transformados para su integración en el informe.

En relación con el calendario laboral, se consolidaron múltiples tablas anuales de días festivos en una única tabla llamada "Festivos", mediante la función Append Queries. Esta tabla sirvió posteriormente como base para el análisis preciso del calendario laboral en los informes generados. Adicionalmente, se realizaron otras clasificaciones relevantes en Power Query, como la creación de un campo específico que agrupa tipos de proceso según si eran técnicos o documentales, permitiendo segmentaciones más eficientes y relevantes.

Finalmente, todos estos procedimientos permitieron que los datos llegaran al modelo de forma limpia, estructurada y sin redundancias, asegurando que la información usada en las visualizaciones fuera confiable, coherente y lista para análisis precisos y efectivos dentro del

informe final.

6.4. Desarrollo e implementación del informe de Power BI

El informe en Power BI permitirá a los responsables del sistema visualizar y analizar los datos de los clientes AGPE de manera interactiva. Este informe se diseñará con unos indicadores claves requeridos por los responsables del sistema:

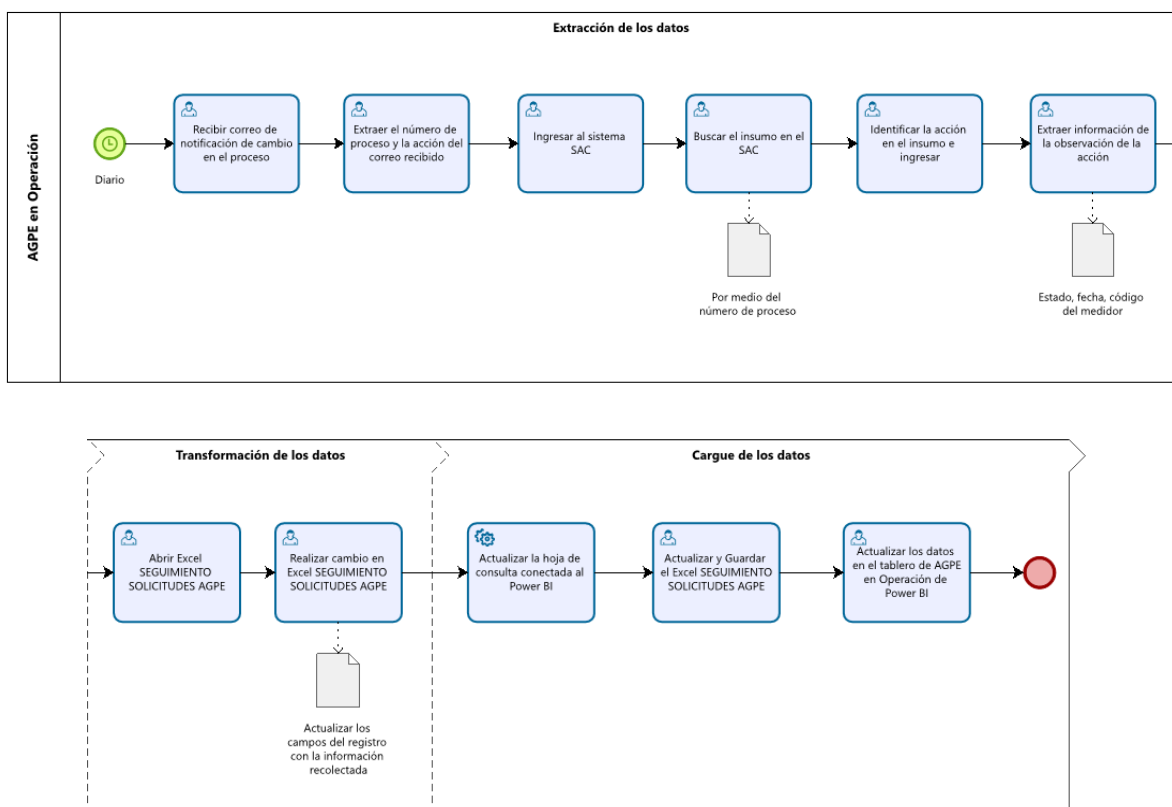
6.4.1. Comparación del proceso previo y proceso automatizado

Previo al desarrollo del informe automatizado en Power BI, el seguimiento a los autogeneradores a pequeña escala (AGPE) en operación y los procesos en trámite se realizaba mediante procedimientos manuales, los cuales requerían una intervención diaria por parte del personal operativo. Estos procedimientos implicaban la consulta constante de reportes en el sistema SAC, la edición de archivos Excel locales y la actualización manual del tablero, lo que no solo consumía tiempo, sino que también aumentaba el riesgo de errores humanos, afectando la trazabilidad y la eficiencia del seguimiento.

En la Figura 18 se presenta el diagrama BPMN del proceso anterior (AS-IS) correspondiente al seguimiento de AGPE en operación. Como puede observarse, el flujo dependía de notificaciones por correo, identificación manual de los procesos y modificación directa de los registros en la hoja Append3 del archivo conectado al informe. Todo este flujo debía ser ejecutado manualmente cada día.

Figura 19

Diagrama BPMN del Seguimiento AGPE en operación (AS-IS)



Nota. Diagrama BPMN elaborado por los autores. Representa el modelo de datos utilizado en el informe de Power BI, correspondiente al flujo anterior (AS-IS) de seguimiento de AGPE en operación.

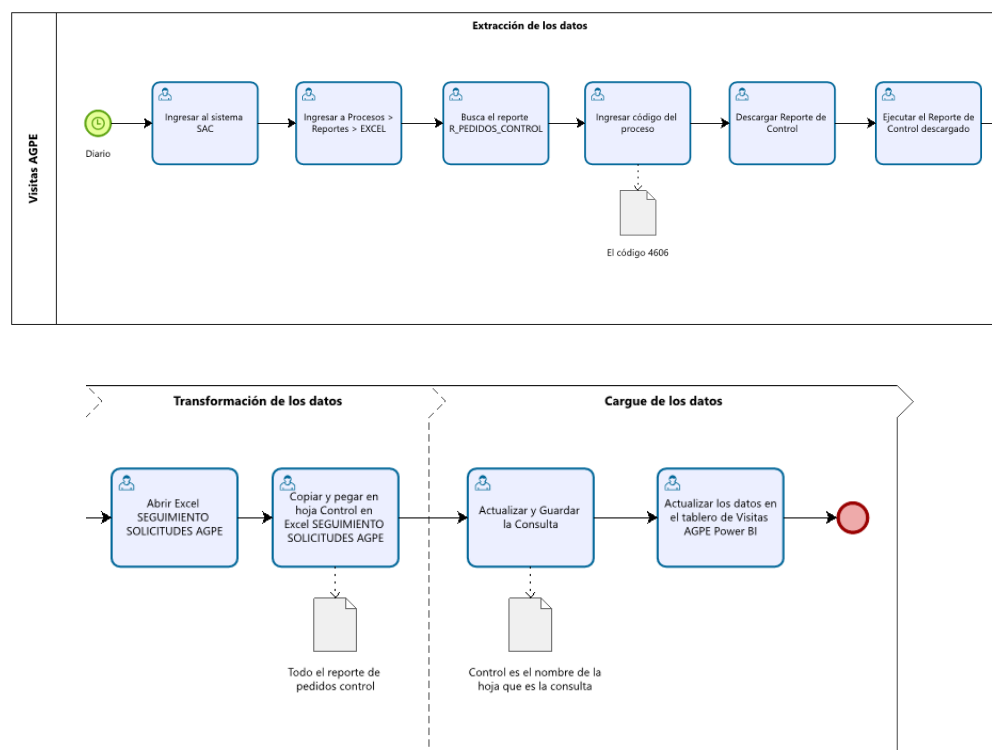
De forma similar, la Figura 19 muestra el proceso anterior para el seguimiento de visitas AGPE, en donde el usuario debía ingresar al sistema SAC, descargar manualmente el reporte “R_PEDIDOS_CONTROL”, copiar la información en el archivo “SEGUIMIENTO SOLICITUDES AGPE” y actualizar el informe. Este flujo era también completamente manual y susceptible a omisiones o demoras.

Con la implementación del informe automatizado, ambos procesos fueron consolidados y optimizados en un único flujo automatizado. El nuevo proceso realiza de forma automática la extracción de datos desde las bases de datos y orígenes configurados, el tratamiento y

transformación de los registros, y la visualización de indicadores a través de Power BI. Este flujo se ejecuta sin intervención humana, lo que ha mejorado significativamente la oportunidad de la información, la confiabilidad de los datos y la disponibilidad del informe.

Figura 20

Diagrama BPMN del Seguimiento de visitas (AS-IS)



Nota. Diagrama BPMN elaborado por los autores. Representa el modelo de datos utilizado en el informe de Power BI para el seguimiento previo (AS-IS) de visitas AGPE, antes de la automatización del proceso.

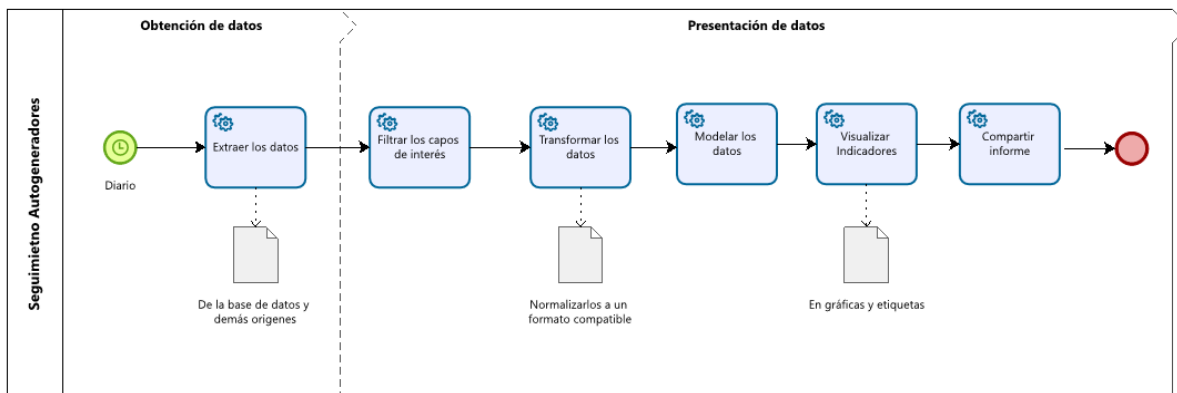
La Figura 20 muestra el diagrama BPMN del proceso optimizado (TO-BE). El nuevo flujo se limita a una secuencia automática de obtención, filtrado, modelado y visualización de datos. Esto permite al usuario final acceder en tiempo real a los resultados del análisis sin necesidad de

realizar tareas operativas.

Este cambio no solo eliminó múltiples tareas operativas, sino que también permitió incorporar nuevos indicadores que antes no podían ser calculados fácilmente, como: el histórico de solicitudes, los tiempos de respuesta por tipo de proceso, los responsables de cada etapa, y un resumen de desempeño por promotor.

Figura 21

Diagrama BPMN para Seguimiento de los AGPE en Power BI.



Nota. Diagrama elaborado por los autores. Representa el modelo de datos en Power BI para el seguimiento de visitas de los AGPE dentro del prototipo.

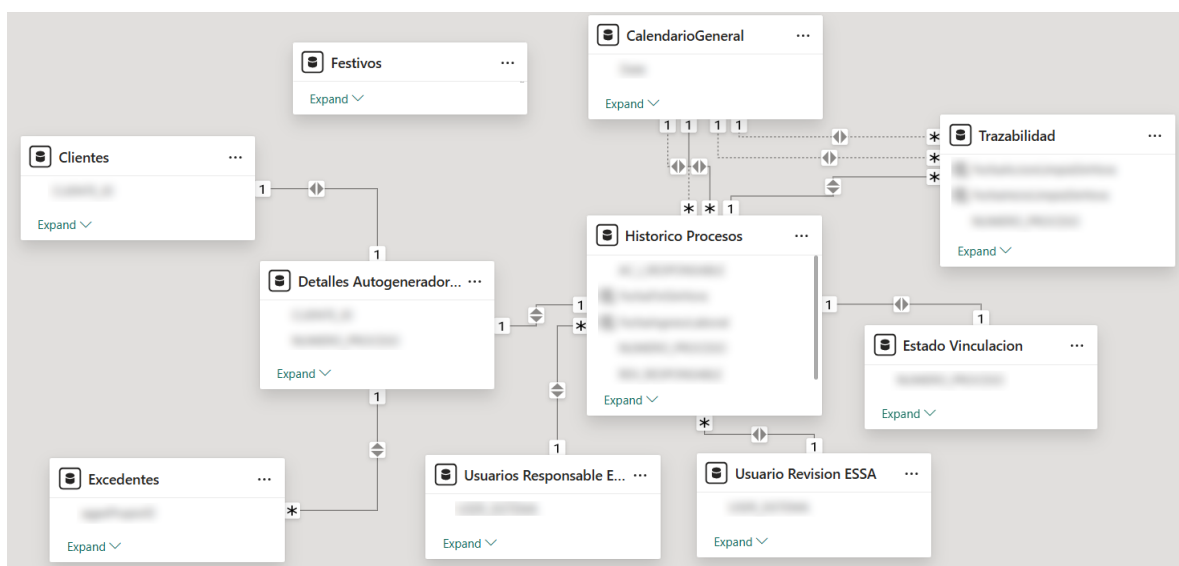
6.4.2. Modelo de datos

El modelo de datos del informe fue diseñado siguiendo principalmente una estructura en estrella, donde la tabla central “Histórico Procesos” actúa como eje principal del análisis. Esta tabla contiene el conjunto más amplio de registros y se relaciona directamente con múltiples dimensiones auxiliares que enriquecen la visualización, como son: el estado de vinculación, los usuarios responsables, la trazabilidad, el calendario general y los detalles del autogenerador.

No obstante, el modelo presenta algunas características de diseño mixto, dado que existen relaciones entre tablas de dimensiones (por ejemplo, “Clientes” y “Excedentes” están conectadas a “Detalles Autogeneradores”), lo que permite una trazabilidad más precisa desde el cliente hasta los excedentes generados, pasando por el proceso y su estado técnico.

Figura 22

Modelo relacional en Power BI



Nota. Captura elaborada por los autores. Representa la estructura relacional entre las tablas utilizadas en Power BI para consolidar la información de los autogeneradores.

Para garantizar la integridad y funcionalidad del modelo, las relaciones entre tablas se definieron principalmente bajo una cardinalidad uno a muchos, asegurando la correcta propagación de filtros entre entidades. Además, se configuraron relaciones inactivas específicas, como la establecida entre la tabla “Festivos” y “CalendarioGeneral”, la cual se activa mediante funciones DAX solo cuando se requiere calcular días hábiles. Esta lógica permitió una evaluación más precisa de los plazos de gestión de cada solicitud.

Además, se utilizó una relación inactiva entre la tabla “CalendarioGeneral” y la tabla de festivos, con el propósito de construir una lógica de días hábiles más precisa. Para ello, se creó una columna calculada en “CalendarioGeneral” que verifica si cada fecha está presente en la tabla de festivos y la clasifica como “Laboral” o “Festivo”. Esta lógica también contempla los días de la semana, marcando como laborales únicamente de lunes a viernes, lo cual fue fundamental para el cálculo exacto del cumplimiento de plazos en los procesos.

Esta estructura logra mantener una lógica clara para los usuarios del informe, optimizar el rendimiento en la carga de datos y facilitar el mantenimiento del modelo.

6.4.3. Columnas calculadas, medidas DAX y diseño de navegación

Columnas calculadas

Durante la construcción del modelo semántico se implementaron diversas columnas calculadas con el objetivo de enriquecer el análisis de los datos, estandarizar campos de entrada y facilitar la visualización. Las columnas más relevantes fueron:

Tabla 13

Columnas calculadas del informe de Power BI

Columna	Propósito principal
Días hábiles del proceso	Calcula los días hábiles entre la fecha de inicio y fin de una solicitud
Días hábiles meta	Establece el tiempo objetivo de respuesta según el tipo de proceso

Cumplimiento	Indica si el proceso fue resuelto dentro del tiempo meta (Sí/No)
Clasificación del tipo de proceso (GrupoProceso)	Agrupar los procesos en Visitas, Conexión o Estudios.
Clasificación ESSA vs otros	Determina si el cliente pertenece a ESSA según su tarifa.
D_Area / D_ObligaFactura / D_Comodato / D_EntregaExcedentes / D_AlmacenaEnergia	Traducción de códigos a valores comprensibles para visualización.
GrupoPrograma	Clasifica el proceso según la acción en la que se encuentra.
Fecha de registro laboral	Ajusta la fecha de inicio a la siguiente fecha laboral más próxima.
Vencido	Evalúa si el proceso excedió los días hábiles permitidos.

Nota. Tabla elaborada por autores. Columnas calculadas utilizadas en el informe de Power BI para visualización de indicadores.

Estas columnas no solo facilitaron el análisis contextual, sino que también permitieron una mejor segmentación e interpretación de los datos por parte del usuario final.

Medidas DAX

Para complementar las visualizaciones, se desarrollaron múltiples medidas DAX que permiten analizar tendencias, calcular indicadores clave de rendimiento (KPI) y ofrecer comparativos dinámicos. Entre ellas destacan:

Tabla 14

Medidas en DAX del informe de Power BI

Medida	Descripción
AGPE_Acumulado	Acumula la cantidad de AGPE registrados por año.
Porcentaje acumulado de solución por días	Indica el porcentaje acumulado de procesos resueltos por día hábil.
Sin responsable	Cuenta los procesos que no tienen un usuario asignado.
Cantidad de procesos a tiempo	Total de procesos que cumplieron con los días establecidos como meta.
Cantidad procesos finalizados/iniciados	Cuenta dinámica de procesos por estado, usando USERRELATIONSHIP para sincronizar fechas.
Procesos año pasado/este año	Comparación dinámica por periodo anual.
Promedio días excedidos	Calcula el promedio de días de retraso en

solicitudes fuera de tiempo.

Cantidad de procesos totalmente finalizados	Total de procesos completados exitosamente.
Cantidad por tipo de tarea (Doc/Tec/Visita)	Cuenta las solicitudes finalizadas por tipo de acción.

Nota. Tabla elaborada por autores. Medidas en Dax utilizadas en el informe de Power BI para visualización de indicadores y KPI's.

Se emplearon funciones como CALCULATE, COUNT, FILTER, ALL, ALLSELECTED, DIVIDE y USERRELATIONSHIP, permitiendo realizar cálculos avanzados, filtrar por contexto dinámico y adaptar los resultados a distintas visualizaciones. También se utilizaron cálculos como promedios y totales gracias a las funciones integradas de Power BI en las visualizaciones.

Tabla de calendario

Se utilizó una tabla de calendario personalizada que incluye las siguientes columnas:

Tabla 15

Tabla Calendario

Columna	Descripción
Date	Fecha completa
Año	Año de la Fecha
Mes	Mes de la Fecha
Día	Día de la Fecha

RangoSemana	Rango de semana en que se encuentra esa fecha.
EsDiaLaboral	Indica si el día es laboral o festivo, según día de la semana y la tabla Festivos.

Nota. Tabla elaborada por autores. Tabla calendario utilizada en el informe de Power BI para segmentación y cálculo de tiempos de respuesta.

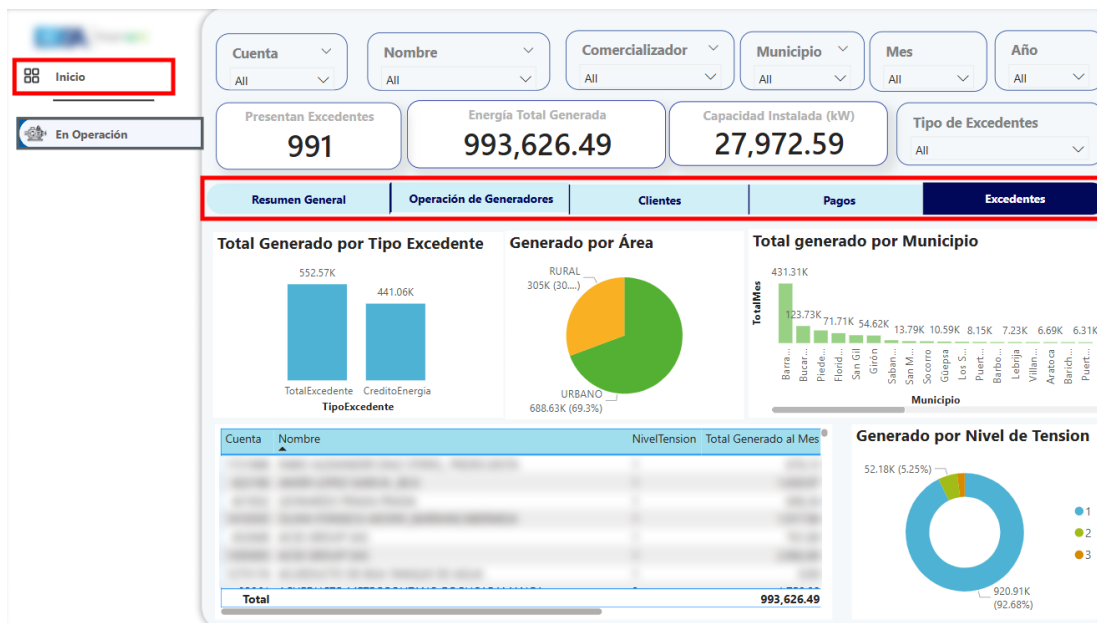
Esta tabla fue relacionada con las tablas de Trazabilidad y del Histórico de Procesos, y fue clave para filtrar por periodos permitiendo análisis temporales, agrupar datos por semanas y validar el cumplimiento de tiempos.

Navegación y submenús

El informe cuenta con una navegación por submenús visuales que permiten al usuario alternar entre vistas sin afectar los datos cargados. Estos submenús fueron implementados con marcadores y botones de acción, ofreciendo una experiencia fluida dentro de cada página. Tanto el submenú de la izquierda como el submenú central cuenta con identificador de la posición en donde se encuentra dentro del informe, facilitando la orientación dentro del informe. La lógica de navegación permite que los filtros seleccionados se mantengan dentro de la misma página, pero se reinicien al cambiar de página, lo cual mejora la exploración sin comprometer la coherencia de los análisis. Para cambiar de página se requiere ir al inicio y seleccionar la página a la deseé desplazarse.

Figura 23

Dashboard de los excedentes generados por AGPE



Nota. Captura elaborada por los autores. Representa la visualización en Power BI del total de excedentes generados, clasificados por área, municipio, tipo de excedente y nivel de tensión.

6.4.4. Estructura visual del informe y validación del funcionamiento

El informe fue desarrollado con una estructura de navegación clara y segmentada, distribuida en cinco páginas principales, diseñadas para facilitar el acceso intuitivo y el análisis visual por parte de los usuarios:

- **Menú principal:** Página inicial del informe que actúa como punto de entrada. Contiene botones interactivos diseñados con Power BI, que enlazan directamente a cada una de las secciones del informe. Estos botones utilizan la funcionalidad de navegación entre páginas (page navigation) y permiten una experiencia fluida y ordenada. Además, incluye información introductoria sobre el objetivo del reporte.
- **En trámite:** Aquí se muestran en tiempo real los procesos de autogeneradores que aún no han sido finalizados. Se incluyen visualizaciones con filtros dinámicos que permiten

identificar rápidamente los responsables del trámite, el estado actual de cada solicitud, la fase del proceso y las próximas fechas de vencimiento. Este tablero facilita la gestión operativa del equipo encargado del seguimiento.

- **Finalizados:** Agrupa las solicitudes que han culminado su proceso. Se incluyen indicadores de cumplimiento, tiempos de ejecución y desempeño por tipo de proceso. Segmentado según el tipo de proceso y de acción por la que pasó el proceso.

- **En operación:** Muestra información técnica detallada como capacidad instalada, ubicación geográfica, tipo de fuente energética, tipo de conexión y estado comercial. Además, se visualizan gráficos de producción mensual, histórico de excedentes y alertas sobre posibles inconsistencias en los datos operativos.

- **Tabla informativa (oculta):** Página auxiliar habilitada como drill through, la cual permite profundizar en la información detallada de procesos con día hábil específico. Incluye número de proceso, tipo, acción, fechas de inicio y fin, y días hábiles. Es accesible desde visualizaciones ubicadas en la página “Finalizados”.

Figura 24

Menú principal del informe de Power BI



Nota. Captura elaborada por los autores. Muestra la interfaz inicial del informe, desde donde se accede a las diferentes secciones del seguimiento de autogeneradores (trámite, finalizados y en operación).

En cuanto a los tipos de visualizaciones utilizadas, predominan los gráficos de barras, seguidos por gráficos de líneas y gráficos de torta, los cuales permiten representar la evolución y comparación de datos de forma clara y accesible. Estas visualizaciones son clave para identificar tendencias y comportamientos por tipo de proceso o estado. Además, se utilizaron numerosas tarjetas de resumen diseñadas para mostrar indicadores clave como la cantidad total de procesos gestionados, el porcentaje de cumplimiento, el tiempo promedio de respuesta, y otras métricas relevantes para la toma de decisiones operativas y estratégica.

Figura 25

Panel de autogeneradores en operación



Nota. Captura elaborada por los autores. Se muestran visualizaciones del informe de Power BI relacionadas con los clientes AGPE en operación, incluyendo capacidad instalada, entregas de excedentes y tipo de generador.

El diseño visual del informe se ajustó a la plantilla institucional proporcionada por ESSA, utilizando los colores corporativos azul, verde y naranja. Este aspecto fue clave para mantener la coherencia gráfica con otros informes corporativos.

Durante el desarrollo del informe se realizaron pruebas funcionales constantes para validar el comportamiento de los datos y su correspondencia con el contexto de cada sección. Posteriormente, con acceso a los datos en ambiente de producción, se realizaron pruebas finales con los usuarios expertos, quienes confirmaron la validez tanto de los valores presentados como de la lógica de segmentación y filtrado aplicada.

Aunque no se utilizaron tooltips personalizados, el informe cuenta con interacciones

Este diseño integral logró balancear la visualización resumida y el acceso a detalle, facilitando tanto el análisis ejecutivo como la consulta operativa.

Nota: Por motivos de confidencialidad y protección de datos sensibles, algunos elementos técnicos del proyecto han sido omitidos o censurados, tales como nombres exactos de tablas, estructuras internas, o detalles que comprometan la identidad de clientes. No obstante, su estructura, propósito y funcionalidad fueron revisados y validados por el equipo técnico de ESSA y cumplen con los lineamientos del área de análisis y datos.

6.4.5. Entrega de documentación técnica

Como parte de las actividades requeridas para la culminación del proyecto y su correcta adopción por parte de la ESSA, se realizó la entrega formal de los documentos técnicos asociados a la solución implementada en Power BI. Esta documentación tuvo como propósito facilitar el entendimiento, mantenimiento y escalabilidad del informe desarrollado en Power BI, especialmente al tratarse de un proyecto ejecutado bajo la modalidad de práctica empresarial y con alcance estratégico dentro de la organización.

Se entregaron tres documentos principales:

Acta de definición de la solución (PDD): Documento inicial donde se establece el objetivo de automatización, los actores involucrados, los prerequisites técnicos, el detalle de los procesos manuales previos (AS-IS), los procesos optimizados (TO-BE), los nuevos indicadores generados y la descripción general del alcance del proyecto. Este acta sirvió como guía para el desarrollo progresivo de la solución y permitió documentar los cambios introducidos a lo largo del proyecto.

Manual técnico y de operación: Documento que describe el objetivo de la solución, su infraestructura técnica, los recursos utilizados, accesos requeridos y el repositorio donde se encuentra almacenado el archivo del informe. Este manual también especifica la lógica del modelado de datos, las fuentes de origen, el detalle de las vistas desarrolladas y el procedimiento para actualizar el informe o modificarlo desde Power BI Desktop.

Acta de entrega de solución: Registro formal de la entrega de la solución, que valida la conformidad del área solicitante frente a los entregables desarrollados y el cumplimiento de los requerimientos pactados. En este documento se resumen los componentes entregados, su alcance y los medios de acceso al informe publicado.

La entrega de estos documentos garantiza la sostenibilidad de la solución a largo plazo, y también respalda la transferencia de conocimiento hacia otros equipos que puedan beneficiarse del modelo implementado. Adicionalmente, representa una práctica profesional dentro del desarrollo de soluciones en ambientes corporativos, alineando el trabajo técnico con las buenas prácticas de documentación en ingeniería de sistemas.

6.5. Estrategia de evaluación económica del impacto

Uno de los logros más relevantes del proyecto fue la reducción significativa en el tiempo de ejecución del proceso completo, el cual anteriormente combinaba múltiples tareas manuales como la consulta de datos en PrimeRead, el tratamiento en hojas de cálculo Excel, la consolidación del archivo de cargue para XM y la actualización de reportes internos. Este flujo tradicional demandaba en promedio unas 15 horas por cada ejecución mensual, lo que representaba aproximadamente 180 horas al año exclusivamente dedicadas a esta labor operativa. Estas tareas, además de repetitivas, estaban expuestas a errores humanos y variabilidad en los tiempos de

entrega.

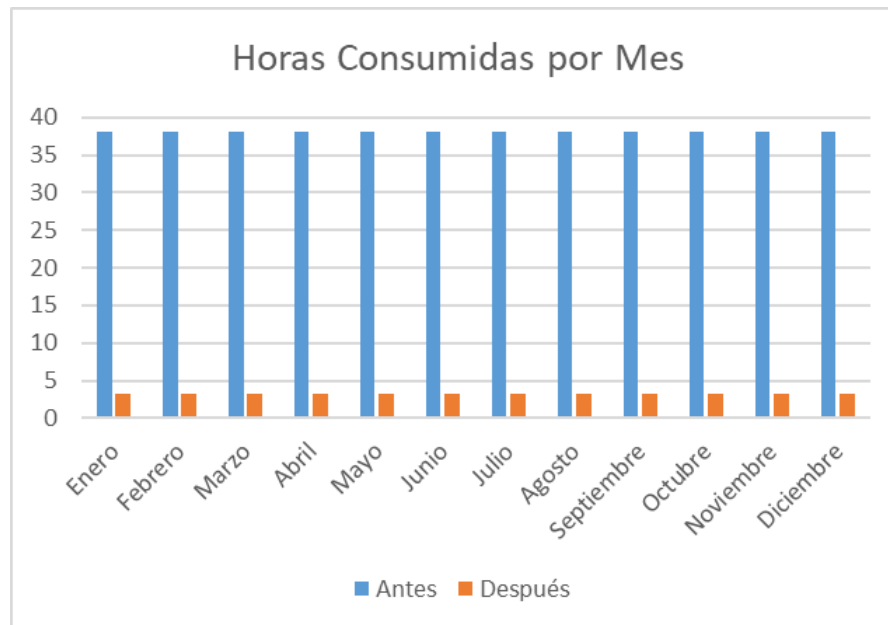
Con la implementación de la solución automatizada —que integró de forma secuencial la ejecución del script en Python, la conexión segura con las bases de datos, el procesamiento automático de los datos, la generación del reporte en formato validado y el envío automático del archivo por correo electrónico—, el tiempo necesario para completar todo el proceso se redujo a tan solo 20 minutos al mes, es decir, alrededor de 240 minutos al año.

Adicionalmente, se eliminaron otras tareas operativas vinculadas a la alimentación manual de datos en los tableros de Power BI, actividad que anteriormente consumía más de 23 horas mensuales (276 horas anuales) debido a la manipulación de archivos Excel con múltiples transformaciones. Actualmente, estas acciones se ejecutan de forma programada, dos veces al día, como se indica en la Figura 16. La automatización permite que los tableros se actualicen sin intervención humana, tardando aproximadamente 3 minutos por cada actualización, es decir, 6 minutos diarios, 180 minutos mensuales o 2160 minutos anuales de procesamiento automático.

Teniendo en cuenta ambos frentes —el reporte mensual y la actualización de tableros— como parte de un flujo integrado y automatizado, se concluye que antes de la solución implementada se consumían 38 horas mensuales (456 horas anuales), mientras que después de la automatización ese tiempo se redujo a 3,33 horas mensuales (40 horas anuales). Esto representa un ahorro total de 416 horas anuales, lo que no solo mejora la eficiencia operativa, sino que también libera al equipo técnico para enfocarse en actividades de mayor valor agregado, como el análisis de datos, la toma de decisiones estratégicas y la mejora continua del servicio.

Figura 27

Horas consumidas antes y después de la solución.

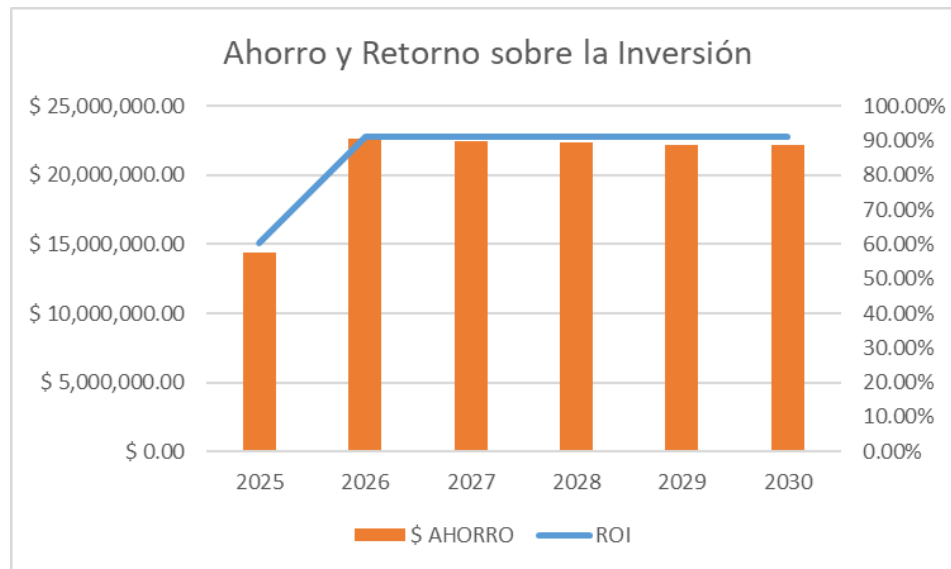


Nota. Gráfico elaborado por los autores. Visualización de horas consumidas antes y después de la implementación de la solución.

La métrica utilizada para medir la rentabilidad para ESSA fue el retorno sobre la inversión (ROI) con un valor estimado del 60,17%, con un ahorro aproximado de \$14.378.668,07 COP para el año actual y un valor estimado del 90,93%, con un ahorro aproximado de \$22.352.241,64 COP para los siguientes 5 años, lo que respalda de manera cuantitativa los beneficios obtenidos con la implementación de la solución.

Figura 28

Ahorro y Retorno sobre la Inversión de los próximos años.



Nota. Gráfico elaborado por los autores. Visualización del ahorro en COP y el Retorno sobre la Inversión (ROI) estimados para 2025-2030.

Para el cálculo tanto del ahorro como del ROI se tuvieron en cuenta: costo de la tarea sin automatizar, costo de tarea con automatización y costo de la automatización, en donde se tuvieron en cuenta los aspectos como: costo anual de personas que realizan el desarrollo del bot, costo anual de la máquina virtual usada por el bot, costo anual del Orquestador usado por el bot, costo anual de la licencia usada por el bot y costos de mantenimiento.

Esta automatización no solo resolvió las limitaciones operativas del proceso original, sino que también dejó sentadas las bases para futuras mejoras en otras áreas de la empresa, demostrando que con herramientas adecuadas se pueden transformar flujos complejos en sistemas sostenibles, confiables y con impacto real en la toma de decisiones.

7. Conclusiones

El desarrollo de este proyecto permitió automatizar por completo un proceso que

anteriormente dependía de múltiples tareas manuales, como la recolección de datos, el cruce de información entre bases de datos, la validación visual de registros y la generación del reporte final exigido por XM. Gracias a la integración de Python, Power Automate Desktop, Power Automate Cloud y Power BI, se logró construir una solución que funciona de manera secuencial, organizada y sin necesidad de intervención del usuario una vez programado el flujo mensual.

Desde el tratamiento de los datos hasta su consolidación en la plantilla oficial, cada etapa del proceso fue diseñada para resolver una necesidad específica: desde asegurar la calidad de los registros, hasta permitir el análisis visual en un tablero dinámico que muestra el estado de los AGPE. Además, se implementó un sistema de conexión segura mediante cifrado, un archivo de logs para trazabilidad de errores, y se documentaron todos los pasos para garantizar el mantenimiento futuro de la solución.

Uno de los cambios más importantes fue el ahorro de tiempo. Lo que antes tardaba horas, ahora se ejecuta en cuestión de minutos. Esta mejora liberó carga operativa al equipo y permitió que el reporte esté disponible el mismo día en que se ejecuta el proceso, incluso en horarios no laborales. También fue posible asegurar que los datos sean válidos antes de consolidarse, lo que redujo significativamente el margen de error y mejoró la confiabilidad de la información entregada.

Toda la solución fue desarrollada considerando las condiciones reales de producción dentro de ESSA, utilizando sus entornos, sus bases de datos y sus políticas de seguridad. Por eso, más allá del funcionamiento técnico, el proyecto demostró que es posible implementar automatizaciones robustas dentro del entorno corporativo, sin depender de licencias adicionales o herramientas externas.

Con esto, se logró cumplir con el objetivo del proyecto, automatizando de principio a fin la generación y entrega del reporte de excedentes, fortaleciendo la trazabilidad de los datos, optimizando el tiempo de ejecución y dejando implementado un sistema que puede mantenerse operativo a largo plazo dentro de la infraestructura de ESSA.

8. Trabajo Futuro

El proyecto abre la puerta a múltiples oportunidades de mejora y expansión que permitirían escalar la solución actual y consolidar una estrategia de automatización más robusta dentro de la organización. Una de las líneas más prometedoras consiste en extender el modelo de automatización a otros procesos técnicos o comerciales de la ESSA, particularmente aquellos que actualmente dependen de reportes manuales y seguimiento por Excel, lo que permitiría replicar los beneficios obtenidos en términos de eficiencia y trazabilidad.

Por otro lado, la incorporación de mecanismos de analítica avanzada, como el aprendizaje automático, permitiría no solo describir el estado actual de los procesos, sino anticiparse a comportamientos futuros. Por ejemplo, el análisis de tendencias podría utilizarse para predecir la carga de trabajo operativa o estimar el tiempo esperado de respuesta para solicitudes nuevas. Estas proyecciones podrían integrarse al informe existente o a soluciones complementarias de inteligencia operacional.

Finalmente, se considera relevante fortalecer los esquemas de gobernanza y seguridad sobre los datos, evaluando la implementación de controles como seguridad a nivel de fila (Row-Level Security) y trazabilidad completa de las consultas ejecutadas, garantizando así una mayor transparencia y cumplimiento normativo.

Referencias bibliográficas

- Comisión de Regulación de Energía y Gas (CREG). (2014). *Resolución CREG 038 de 2014: Parámetros para la medición y reporte de energía activa y reactiva.*
https://gestornormativo.creg.gov.co/gestor/entorno/docs/resolucion_creg_0038_2014.htm
- Comisión de Regulación de Energía y Gas (CREG). (2018). *Resolución CREG 030 de 2018: Condiciones de conexión y operación de los Autogeneradores a Pequeña Escala.*
https://gestornormativo.creg.gov.co/gestor/entorno/docs/resolucion_creg_0030_2018.htm
- Congreso de Colombia. (2014). *Ley 1715 de 2014: Por medio de la cual se regula la integración de las energías renovables no convencionales al Sistema Energético Nacional.*
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=57353>
- Dataprise. (2022). *Microsoft Power Platform: Empowering Organizations to Achieve More.*
https://www.dataprise.com/wp-content/uploads/2022/11/Microsoft_Power_Platform.pdf
- Electrificadora de Santander S.A. (ESSA). (2020). *Procedimiento para la conexión de autogeneradores.*
<https://www.essa.com.co/site/Portals/clientes/Documentos/procedimientos-conexion-red/conexion-autogenerador.pdf?ver=2020-04-20-120948-793>
- Ilo, N. (2018). *Robotic Process Automation implementation in record-to-report process.*
Theseus.fi.
https://www.theseus.fi/bitstream/handle/10024/158271/Ilo_Nina.pdf?sequence=1
- León, J. (2024). *Automatización de procesos de gestión contable en la Electrificadora de Santander S.A. (ESSA) a través de herramientas de Microsoft y Scripts de Python.*

- Universidad Industrial de Santander*. <https://noesis.uis.edu.co/items/b70c28df-ec93-4907-b447-8bd558a09005>
- Microsoft Corporation. (2023). *Power BI, Power Automate*. <https://www.microsoft.com>
- Microsoft. (s.f.-a). *Task Scheduler Start Page*. Documentación de Windows. <https://learn.microsoft.com/en-us/windows/win32/taskschd/task-scheduler-start-page>
- Microsoft. (s.f.-b). *Conexiones de flujo de escritorio en Power Automate*. <https://learn.microsoft.com/es-es/power-automate/desktop-flows/desktop-flow-connections>
- Microsoft. (s.f.-c). *Solución de problemas de conectividad directa de flujos de escritorio*. <https://learn.microsoft.com/es-es/troubleshoot/power-platform/power-automate/desktop-flows/troubleshoot-direct-connectivity-issues>
- ND Marketing Digital. (2024). *Cómo se elabora un RPA: Guía paso a paso para elaborar un RPA de manera efectiva*. <https://ndmarketingdigital.com/como-se-elabora-un-rpa/>
- Oracle. (s.f.). *Enabling Thick Mode*. Documentación oficial de oracledb. https://python-oracledb.readthedocs.io/en/latest/user_guide/installation.html#enabling-thick-mode
- Python Software Foundation. (s.f.). *PEP 257 – Docstring Conventions*. <https://peps.python.org/pep-0257/>
- Stela.ai. (2024). *Implantar RPA: Guía Completa*. <https://www.stela.ai/implantar-rpa-guia-completa/>
- Vecteezy. (2024). *Iconos de arquitectura*. <https://es.vecteezy.com/>

XM S.A. E.S.P. (s.f.). *Quiénes somos*. <https://www.xm.com.co/nuestra-empresa/nosotros/quienes-somos>