Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Automatización del Análisis y Resumen de Logs del Sistema Windows mediante RPA para Monitorización y Soporte en Análisis Forense

Proyecto Fin de Grado

Grado en Sistemas de Información

Curso académico 2024-2025

Autor:
Julen Blázquez Álvarez

Tutores:
Juan Manuel Castelo Gómez

En primer lugar, quiero agradecer a Juan Manuel por su apoyo, su interés y su cercanía como tutor, ha sido un placer.
Gracias a mis amigos por estar siempre a mi lado, y a Omar por su ayuda.
A mi novia, por apoyarme incondicionalmente y hacer que cada día sea especial, sin duda lo mejor que me llevo de esta etapa.
Y por último y más importante, gracias a mi hermano y mis padres. Gracias por vuestro esfuerzo diario. Todo lo que he conseguido y lo que consiga en el futuro es, en gran parte, gracias a vosotros.

Resumen

En este trabajo de fin de grado se desarrolla una herramienta para el análisis y filtrado de logs mediante la automatización robótica de procesos (RPA) utilizando el marco REFramework, siendo capaz de extraer información relevante de los logs en Windows mediante el visor de eventos del sistema.
A lo largo de esta memoria, se detallará el contexto histórico y teórico de la automatización robótica de procesos, siendo una tecnología extremadamente útil y novedosa con múltiples aplicaciones. También se explica la utilidad y funcionamiento del visor de eventos de Windows, diferenciando los distintos tipos de logs que podemos identificar, incluidos logs de aplicaciones, de seguridad y de sistema.
Posteriormente, se detallará el funcionamiento y desarrollo de la herramienta, desde el planteamiento hasta la implementación. Se detallarán los diferentes desafíos y problemas que han surgido durante el proyecto y las soluciones empleadas, incluyendo un caso de estudio probando la eficacia de la herramienta con un paquete de logs predefinido. Finalmente, se proponen nuevas líneas de actualización, mejora y complementación de la herramienta, demostrando su potencial en el análisis forense, además de una conclusión sobre el proyecto.

Palabras clave: Automatización robótica de procesos, RPA, analisis, Windows, logs, visor de eventos.

Abstract

This thesis develops a tool for log analysis and filtering through robotic process automation (RPA) using the REFramework structure, being able to extract relevant information from Windows logs through the system's event viewer.
Throughout this report, we will detail the historical and theoretical context of robotic process automation, being an extremely useful and novel technology with many applications. We will also explain the usefulness and operation of the Windows Event Viewer, differentiating the different types of logs that can be identified, including application, security, and system logs.
Next, the operation and development of the tool will be detailed, from the approach followed to the implementation. We will detail the different challenges and problems that have arisen during the project and the solutions used, including a case study testing the effectiveness of the tool with a predefined log package.
Finally, new lines of improvement and complementation of the tool will be listed, demonstrating its potential in forensic analysis, as well as describing the conclusions that can be drawn from the project.

Key words: Robotic process, automation, RPA, analysis, Windows, logs, event viewer.

Índice

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

1.1. Introducción

Tanto la ciberseguridad como la seguridad de la información se han convertido en un pilar esencial para proteger a las organizaciones, los sistemas y a los usuarios. Cada día surgen nuevas amenazas que ponen en peligro la integridad y la confidencialidad de la información. Es por eso que resulta fundamental recurrir al análisis forense cuando ocurre un incidente de seguridad y así ser capaces de reconstruir los hechos e identificar las amenazas y su alcance.

El análisis forense digital implica la recolección, preservación y análisis de evidencias electrónicas. Sin embargo, esta tarea suele ser compleja y laboriosa, debido al enorme volumen de datos que generan los sistemas. Para abordar esta dificultad, cada vez se recurre más a la automatización de tareas repetitivas y sistemáticas, facilitando así una investigación más rápida y fiable.

Es en este contexto, es donde la automatización robótica de procesos (RPA), capaz de ejecutar acciones humanas en sistemas digitales de forma autónoma, ofrece una forma eficaz de analizar y resumir los registros de eventos, reduciendo los tiempos de respuesta y mejorando la capacidad de análisis de los expertos.

1.2. Motivación

El determinar **qué ha ocurrido en un sistema**, no solo nos permite proteger nuestros sistemas informáticos, sino también prevenir futuros incidentes, entender el impacto de los mismos e incluso obtener pruebas válidas en procedimientos legales.

El volumen de datos de un sistema informático es uno de los retos dentro del análisis forense, ya que el tiempo del que disponemos para realizar la investigación puede ser limitado. Es por ello que es tan relevante el uso de

herramientas forenses que faciliten y asistan a las personas en el proceso de investigación y extracción de información en un sistema que haya podido ser comprometido.
En este trabajo, se propone el desarrollo de una solución basada en RPA que contribuye a este objetivo, automatizando la exploración y resumen de anomalías en logs de sistemas. Esta propuesta no solo responde a una necesidad técnica evidente, sino también a un interés personal en el campo de la ciberseguridad.

1.3. Objetivos

El objetivo principal de este proyecto es plantear y desarrollar una herramienta que resulte fácil de utilizar, pero que a la vez sea de gran utilidad para monitorizar de manera sencilla y rápida la salud de los sistemas operativos Windows, usando como información base los datos relativos a los eventos del sistema, resultando útil tanto para usuarios que no cuenten con conocimientos técnicos como para administradores de sistemas en entornos empresariales. Se espera que el resultado proporcione una base sólida con gran escalabilidad hacia diferentes enfoques para aumentar su utilidad y rendimiento.

1.4. Fases

Las diferentes fases en las que se divide este proyecto son las siguientes:

1. Planteamiento de la idea. teniendo en cuenta las prestaciones y las capacidades de la automatización robótica de procesos (RPA), especialmente en lo que respecta a la automatización de tareas repetitivas y de bajo valor añadido, es interesante explorar cómo esta tecnología podía aplicarse en un ámbito de mayor complejidad y relevancia como es el de la seguridad informática. Dado el interés por la ciberseguridad, unido al potencial del RPA para operar de forma rápida y sin errores humanos, es interesante la posibilidad de combinar ambos mundos en un proyecto que no solo automatice procesos, sino que también contribuya a la detección de posibles amenazas o comportamientos anómalos en entornos digitales.

2. Fase de lectura y aprendizaje. Una vez definido el enfoque general del proyecto, una parte importante del tiempo se dedica a la fase de documentación, con el objetivo de identificar qué tipo de tareas podría asumir un robot RPA dentro del ámbito de la ciberseguridad. Al tener en cuenta distintos escenarios, como la monitorización de eventos, la gestión de alertas o la verificación de accesos, finalmente, el análisis de logs se estima como una tarea adecuada tanto por su valor en la detección de anomalías como por su naturaleza repetitiva y estructurada, lo que la la hace idónea para ser automatizada. El aprendizaje se dividie en dos líneas de estudio: por un lado, las tecnologías RPA disponibles, sus capacidades y limitaciones; y por otro, el funcionamiento de los registros de eventos (logs), su estructura, los diferentes tipos existentes (sistemas, aplicaciones, seguridad, etc.), y las metodologías comunes utilizadas para analizarlos en busca de comportamientos anómalos o indicios de incidentes de seguridad.

3. Fase de diseño. Se llevan a cabo pruebas con el visor de eventos para entender cómo se generan los logs y cómo interactuaría el robot con ellos. Es importante familiarizarse con el formato y la estructura de los registros para poder definir de manera adecuada cómo el robot debe procesarlos. Durante este proceso, se establece la estructura del robot, tomando en cuenta los tipos de logs que debe analizar, las fuentes de los mismos y las herramientas disponibles para procesarlos. El objetivo es construir un diseño flexible que pueda adaptarse a diferentes tipos de registros, manteniendo una operación fluida y eficiente. Se evalúan diversas formas de integrar el robot con los registros, buscando una solución que sea tanto eficiente como flexible.

4. Fase de implementación. El robot se programa mediante un proceso de prueba y error, ajustando el código progresivamente para mejorar su funcionamiento y su rendimiento. El desarrollo se realiza de manera independiente en cada uno de los estados de la *state machine* de la solución.

5. Una vez programado el robot, se inicia la fase de pruebas, la cual

es crucial para ajustar el funcionamiento del robot y corregir errores menores, como la configuración de las columnas del archivo Excel donde el robot registra los resultados del análisis, o el tiempo que debe esperar entre cada uno de los *click* que realiza durante la navegación en el visor de eventos. Es un proceso iterativo, donde cada ejecución permite corregir y mejorar el resultado.

6. Una vez finalizado el desarrollo del robot, se consideran posibles mejoras para incrementar su utilidad, tanto para empresas como para usuarios comunes. En el caso de las empresas, este tipo de robot podría ser muy valioso para automatizar la monitorización de seguridad, ayudando a detectar incidentes de manera rápida y eficiente. Para usuarios comunes, podría ofrecer una capa extra de protección al analizar logs de sus dispositivos o aplicaciones, mejorando así su seguridad personal.

1.5. Estructura de la memoria

- **Capítulo 1: Introducción.** Se presenta el contexto general del trabajo, la motivación detrás del mismo, los objetivos planteados y las diferentes fases que componen el proyecto.
- **Capítulo 2: Fundamentos técnicos.** Se describen los conceptos esenciales del análisis forense, logs de sistema y Robotic Process Automation (RPA), sirviendo de base para comprender el desarrollo del proyecto.
- **Capítulo 3: Desarrollo del proyecto.** Se expone el proceso de construcción del robot, las tecnologías empleadas y la lógica implementada.
- **Capítulo 4: Resultados y ejecución.** Se analizan los resultados obtenidos tras la ejecución de la herramienta, incluyendo el ejemplo de una ejecución. También se detallan diferentes retos y problemas encontrados durante el desarrollo y las pruebas.
- **Capítulo 5: Impactos sociales y ambientales.** Se analiza la contribución del proyecto a los Objetivos de Desarrollo Sostenible

(ODS), valorando tanto los beneficios sociales como los posibles efectos ambientales de la implantación de soluciones de automatización en el ámbito de la ciberseguridad.

- **Capítulo 6: Conclusiones y trabajos futuros.** Se presentan las principales conclusiones extraídas del trabajo, se reflexiona sobre los logros y limitaciones del proyecto y se proponen líneas de mejora y posibles extensiones para investigaciones o desarrollos posteriores.

Capítulo 2

Fundamentos técnicos

En este capítulo, se describirán en profundidad las diferentes tecnologías y herramientas empleadas en este proyecto. Además, se aporta un contexto histórico y empresarial de las mismas para ofrecer una mejor comprensión de la solución implementada.

2.1. Automatización Robótica de Procesos

Las siglas RPA significan *robotic process automation*, [4] y hacen referencia a una tecnología novedosa cada vez más implementada por las empresas, dado que mediante robots software facilita y automatiza las tareas manuales, sencillas o repetitivas que tienen lugar en los diferentes procesos de negocio, simulando acciones humanas.
La mejora de la eficacia que aportan estos robots es sobre todo notable en el aumento en la velocidad y la constancia frente a las personas en labores y tareas que tienen que ver con entrada de datos, envío y recepción de correos electrónicos, procesamiento de documentos, extracción de datos, actualizaciones de bases de datos, navegación por diferentes aplicaciones, gestión documental, y un largo etcétera [5].
Estos robots se basan en scripts integrables con múltiples aplicaciones empresariales y son combinables con tecnologías que cuentan con inteligencia artificial, como pueden ser el procesamiento de lenguaje natural o el aprendizaje autónomo.
Otra de las ventajas de la implementación de RPA dentro de los procesos de negocio de una empresa es la significativa reducción de costos operativos y la posibilidad de liberar a los empleados de invertir gran parte de su tiempo de trabajo en estas acciones más repetitivas, permitiendo así a los trabajadores dedicarse a actividades mucho más estratégicas, creativas y relevantes.

2.1.1. Marco histórico de la tecnología

La automatización como término, nació en 1946 en una fábrica de automóviles Ford, cuando el proceso de producción empezó a estar mecanizado y controlado por dispositivos autónomos.
Las empresas generalmente buscaban externalizar sus procesos empresariales subcontratando trabajo manual a otras empresas, a menudo en países más lejanos. Sin embargo, el aumento en los costes impulsó a estas empresas a buscar formas más baratas de realizar ese tipo de trabajos, como eran por ejemplo el procesamiento de pedidos y la gestión de inventarios. Los primeros casos de uso dónde se buscó aplicar tecnologías de automatización fueron la creación de copias de seguridad, el rellenado automático de formularios, tramitación de facturas, y gestión y extracción de datos, entre otros [6]. En la década de los 2000 surgieron las primeras tecnologías de *screen scraping* (automatización de la extracción de datos generados por una aplicación mediante un programa informático), pero no fue hasta 2012 que se usaron las siglas RPA por primera vez. No comenzó a ser una tecnología popular hasta 2014, cuando las empresas tecnológicas, de banca y finanzas, que eran las únicas que contaban con recursos suficientes, empezaron a adoptar enfoques mucho más ágiles dando más valor al tiempo de las personas. El RPA era una solución a la necesidad de acelerar las operaciones empresariales reduciendo costos y mejorando la eficiencia operativa imitando la cognición humana con mecánicas de automatización desatendidas [7].
En 2018, la conocida empresa KPMG redactó un informe [8] destacando la posibilidad de reducción de costos de bancos e instituciones financieras mediante el uso de RPA.

2.1.2. Actualidad del RPA y su sinergia con la IA

Actualmente, el RPA representa una de las más significativas e importantes herramientas para la transformación digital de las empresas, haciéndolas más productivas y eficientes.
Gracias al aumento del uso de estas tecnologías, en los últimos años han surgido muchas empresas que ofrecen soluciones RPA como servicio, permitiendo que estas innovaciones lleguen a empresas más pequeñas. Por ejemplo, el 80 % de las *PYMES* hoy en día considera la automatización parte fundamental de sus procesos de negocio [9].
Uno de los principales problemas de la automatización robótica de procesos es su limitación a la hora de manejar situaciones complejas o dinámicas, su falta de adaptabilidad ante cualquier variación de los datos de entrada (que deben ser siempre estructurados), o de los procesos, significaba un fallo que podía salir realmente caro a las empresas que implementaban la tecnología, siendo preciso el tener siempre personal pendiente de la ejecución de estos robots, capacitada para lidiar con excepciones o problemas de ejecución. Otro gran obstáculo para la RPA es la constante necesidad de actualización en su programación ante cualquier cambio en los procesos de negocio de las empresas, limitando en gran medida su escalabilidad.
Estos problemas se solucionan con la reciente implementación de la inteligencia artificial dentro de los procesos automatizados, esta combinación de ambas tecnologías se denomina como: **Automatización Inteligente** [10], que se basa en no solo imitar las acciones humanas, sino en replicar también su inteligencia y su capacidad de aprendizaje, pudiendo así tomar decisiones en tiempo real basadas en datos, reduciendo significativamente la incidencia de los problemas mencionados anteriormente, permitiendo la entrada de datos no estructurados, y dejando de ser necesario personal supervisando la ejecución de los procesos.
En un futuro cercano, los robots RPA serán capaces de recopilar y analizar información sobre la tarea que van a desempeñar, utilizando esos conocimientos para elevar exponencialmente su eficacia y su independencia

dentro de sus funciones [11].

2.2. Entornos de desarrollo y gestión de las tecnologías RPA

A lo largo de esta sección, se describirá en detalle los dos principales entornos utilizados por las empresas para el desarrollo y gestión de las tecnologías RPA: UiPath y Blue Prism.

2.2.1. Blue Prism

Blue Prism [12], que da nombre tanto a la tecnología como a la empresa que la desarrolla, es pionera y líder mundial en la automatización de procesos. Se destaca frente a sus competidoras por su enfoque en la seguridad y la gobernanza, siendo así una de las opciones preferidas por las empresas relacionadas con la banca, los seguros y la salud.
Su plataforma permite a los usuarios desarrollar y gestionar robots software de manera accesible, facilitando su escalabilidad e implementación. Además, ofrece herramientas para el análisis y monitoreo de sus robots para supervisar su rendimiento.
Blue Prism destaca por incluir funcionalidades como la integración con la inteligencia artificial, el reconocimiento óptico de caracteres (OCR) y el procesamiento de lenguaje natrual (NLP), permitiendo a sus robots el manejo de datos no estructurados y la realización de tareas complejas.
Blue Prism cuenta con dos herramientas fundamentales, **Blue Prism Process Studio**, que es el entorno de desarrollo donde los usuarios construyen sus automatizaciones y definen cómo se comportarán los robots mediante una interfaz gráfica basada en arrastrar y soltar bloques de código que definen diferentes acciones, conectando esos bloques mediante lógica y estructuras de programación básicas. Por otro lado, existe **Blue Prism Control Room**, que actúa como una plataforma para la gestión y monitorización de los robots creados en Process Studio, pudiendo controlar las ejecuciones y gestionar colas de trabajo.
La plataforma cuenta con una gran comunidad de miembros activos que contribuyen a su crecimiento, dado que tiene una gran oferta de formaciones

y recursos educativos, así como certificacioes y bibliotecas [13].

2.2.2. UiPath

UiPath [14], también da nombre tanto a la empresa, que fue fundada en el año 2005 en Bucarest, Rumanía, como a la tecnología en sí. UiPath ha sido reconocida como empresa líder en el sector de las tecnologías de automatización de procesos durante varios años consecutivos en el *Cuadrante Mágico de Gartner* [1] (herramienta que evalúa proveedores de tecnología en funcion de dos criterios: **Capacidad de ejecución** e **Integridad de la visión**), como podemos observar en la Figura 2.1, destacándonse en el mercado por su facilidad de uso, su escalabilidad y su integración con otras tecnologías como la Inteligencia Artificial.
Su principal desventaja frente a sus competidores en el mercado de proveedores de servicios RPA es el precio, ya que las soluciones de UiPath pueden ser especialmente costosas para pequeñas y medianas empresas que buscan implementar estas soluciones automatizadas a gran escala.

Figura 2.1: Magic Quadrant for Robotic Process Automation. [1] Fuente: Gartner, Arthur Villa et al, 7 de agosto de 2024.

Al igual que Blue Prism, UiPath cuenta con su propio ecosistema de entornos y tecnologías para ofrecer sus soluciones RPA:

- **UiPath Studio**: Se trata de un entorno de desarrollo integrado donde los usuarios y programadores diseñan sus automatizaciones mediante una interfaz visual muy intuitiva y amigable, arrastrando bloques conocidos como *actividades* que, junto a una lógica similar a la de cualquier lenguaje de programación, permite crear flujos de trabajo.

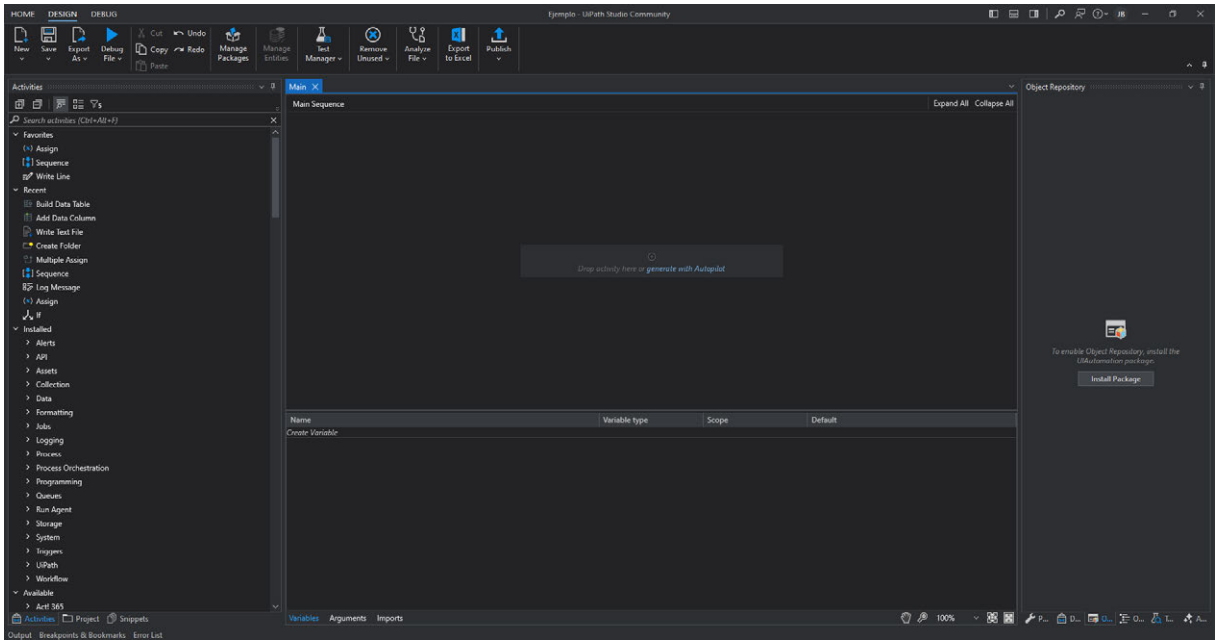

Figura 2.2: UiPath Studio

- **UiPath Orchestrator**: Es una plataforma de gestión que permite a organizaciones y usuarios gestionar y monitorizar sus automatizaciones. Esta plataforma también permite crear y manejar *Assets* y colas de trabajo útiles para las automatizaciones, además de programar *Triggers* para que se ejecuten los robots software de manera desatendida.

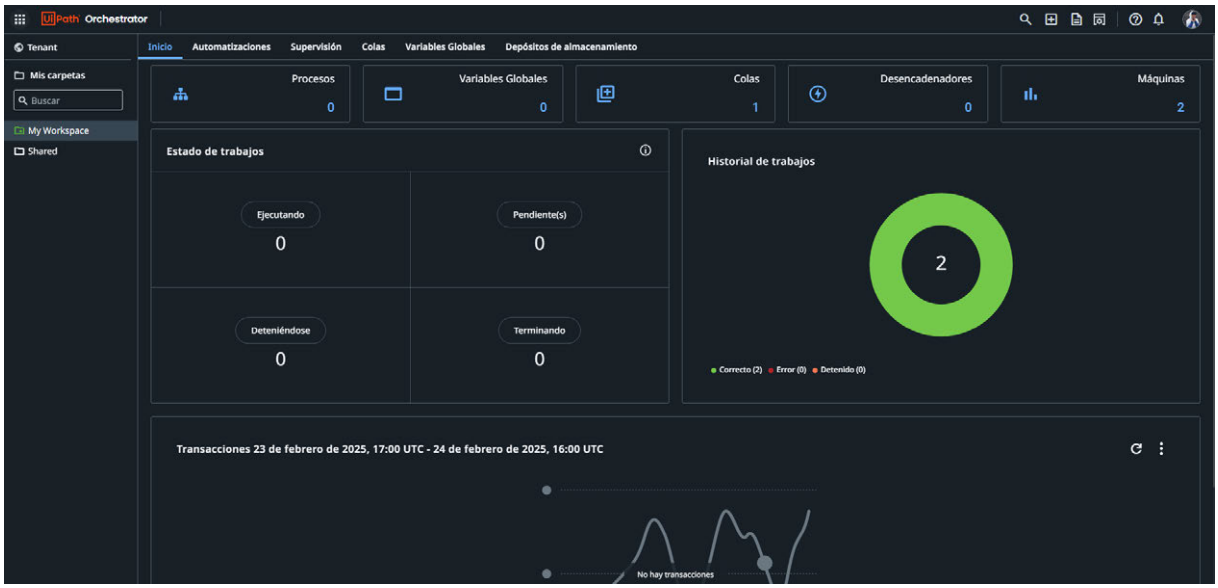

Figura 2.3: UiPath Orchestrator

- **UiPath Robots**: Estos robots software pueden ser de dos tipos, atendidos o desatendidos [15], los robots atendidos son monitorizados y ejecutados por una persona, necesitando de acciones de un empleado. Los robots desatendidos operan de manera autónoma sin necesidad de acciones, participación, ni intervención humana.

Dentro de la propia web de UiPath podemos encontrar un apartado llamado **UiPath Academy** [2], que ofrece una amplia gama de recursos y de diferentes formaciones gratuitas diseñadas para enseñar a los usuarios a desarrollar sus habilidades en la automatización de procesos en el ecosistema de UiPath, en función de las habilidades previas de los mismos. Hay desde formaciones básicas hasta niveles expertos de diversos temas como el propio desarrollo o la gestión de los mismos en Orchestrator.
Esta academia también ofrece certificaciones y exámenes propios muy bien reconocidas y valoradas dentro de la industria como puede ser la *UiPath Certified Advanced RPA Developer (UiARD)*. La plataforma también fomenta, mediante foros y comunidades de automatización, la interacción entre sus usuarios para resolver dudas y colaborar en proyectos.

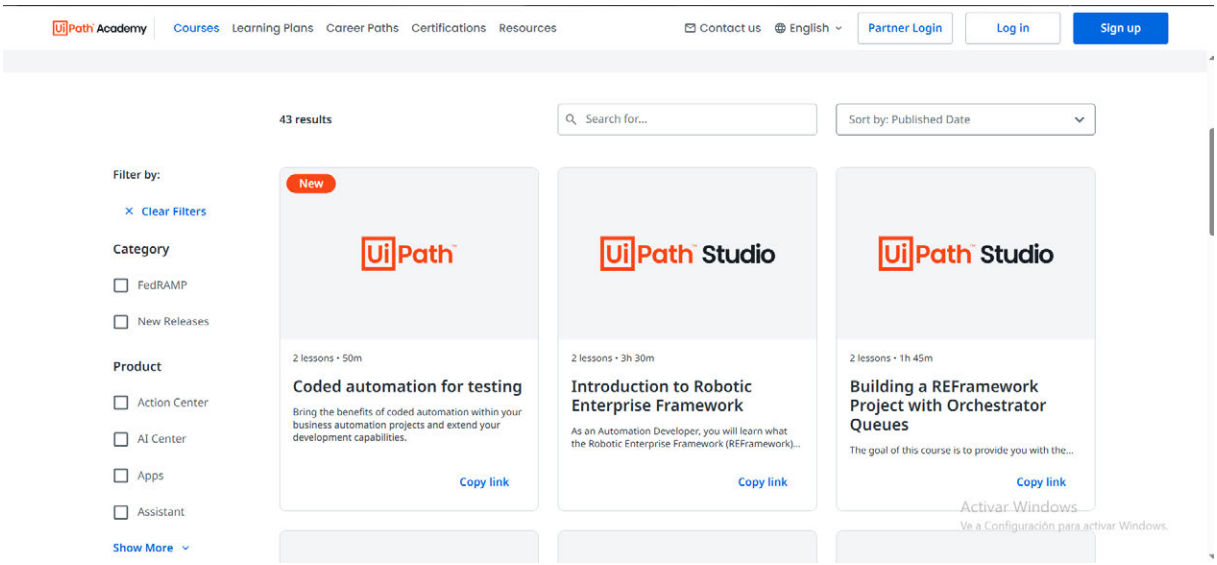

Figura 2.4: UiPath Academy [2]

2.2.3. REFramework

El *Robotic Enterprise Framework* o REFramework de UiPath es un marco de desarrollo de robots software **basado en *state machines*** [16], diseñado para automatizar procesos de negocio complejos. Este modelo ofrece múltiples ventajas, dado que proporciona una estructura muy robusta y en extremo escalable para desarrollar soluciones de automatización muy personalizables, además de contar con un sólido **sistema de manejo de excepciones** y la implementación de **buenas prácticas** (ambos conceptos serán abordados en la descripción de la solución implementada).

Este marco utiliza una serie de estados predefinidos y transiciones entre los mismos para gestionar el flujo de un proceso automatizado complejo de negocio. Los estados que podemos distinguir dentro de esta ***state machine*** son los siguientes:

- ***Inicialization State***: Este estado es el encargado de la inicialización de todas las variables y aplicaciones que van a formar parte del flujo del proceso de negocio, incluye también la lectura de las configuraciones desde el **archivo de configuración**. Abre todas las aplicaciones que vayan a utilizarse durante el proceso.

- ***Get Transaction Data State***: Durante la ejecución de este estado son extraídos o recuperados los datos que van a ser procesados en el siguiente estado, pueden proceder de distintas fuentes, como bases de datos, archivos de cálculo, aplicaciones externas o **colas en UiPath Orchestrator**.
- ***Process Transaction State***: Es en este estado donde los datos extraídos son realmente procesados, incluyendo la lógica específica del proceso de negocio concreto que se esté automatizando. Maneja cualquier excepción que pueda suceder durante el procesamiento. Dentro de este estado puede haber variaciones dependiendo de si se trata de un robot que cumple un rol de ***Dispatcher*** o ***Performer*** (estos roles se encuentran detallados más adelante en este mismo subapartado). Funciona formando un bucle con el estado anterior, donde se extrae un conjunto de datos que pasa a llamarse ***Transaction***, se procesa (*Process State*) y al acabar comprueba si hay mas datos para procesar (*Get Transaction Data State*). En caso contrario, avanza al *End Process State*.
- ***End Process State***: Finaliza el proceso, realiza cualquier proceso de limpieza o vaciado necesario y cierra las aplicaciones anteriormente inicializadas.

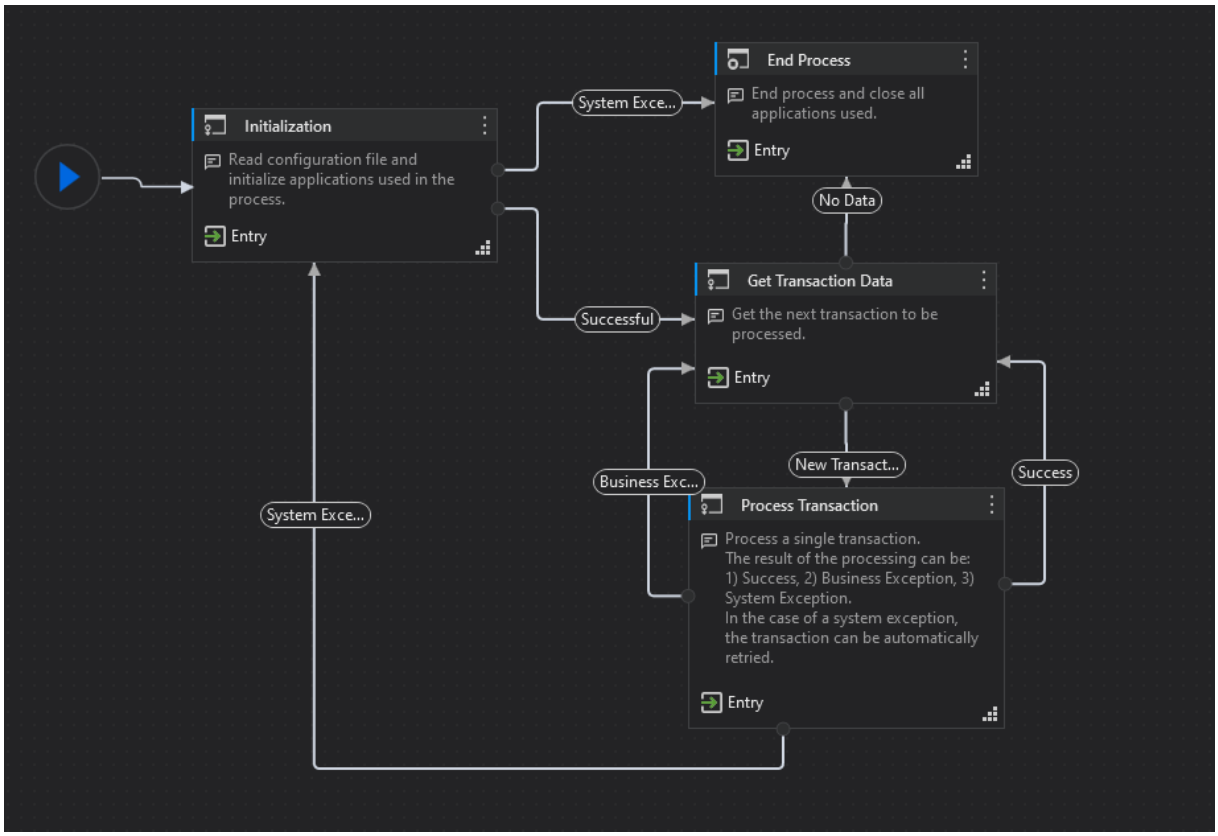

Figura 2.5: Robotic Enterprise Framework

El modelo ***Dispatcher-Performer*** es un tipo de arquitectura que divide el proceso de automatización en dos partes distinguidas:

1. ***Dispatcher***: Robot software que se ejecuta en primer lugar encargado de extraer y almacenar los datos brutos que se quieren procesar en una cola en **UiPath Orchestrator**. Separa la extracción de datos de su procesamiento. En un *Dispatcher*, el estado *Get Transaction Data*, extrae los datos en bruto de la fuente de datos y, el estado *Process Transaction* los almacena en la cola en UiPath Orchestrator.

2. ***Performer***: La ejecución de este robot consiste en extraer los datos almacenados por el *Dispatcher* en la cola de **UiPath Orchestrator** mediante el estado *Get Transaction Data* y realizar el procesamiento necesario, en el estado *Process Transaction*, en función del proceso de negocio para el cúal ha sido diseñado el robot.

Las ventajas de este modelo son variadas:

- Separación de responsabilidades, lo que permite una mejor gestión de los errores.
- Facilidad de escalabilidad y evolución de los procesos.
- Posibilidad de ejecución de varios *Performer* en paralelo en caso de haber muchos ítems a procesar, mejorando eficacia y reduciendo tiempo de procesamiento.
- Mayor flexibilidad en la gestión de cargas de trabajo.

Cada uno de los términos resaltados en negrita será explicado con más detenimiento en el apartado: Desarrollo

2.3. Visor de Eventos

El visor de eventos en Windows 11 es una herramienta esencial que monitorea y registra los eventos (*logs*) del sistema operativo. Permite tanto a los administradores del sistema como a los usuarios avanzados que cuenten con los permisos necesarios diagnosticar problemas, buscar incidencias, realizar auditorías de seguridad, analizar el rendimiento del sistema o buscar incidentes y eventos concretos [17].
Se trata de una herramienta ya integrada de forma nativa en los sistemas Windows, se puede acceder a ella mediante una interfaz gráfica, tal y como se ve en la Figura 2.6).

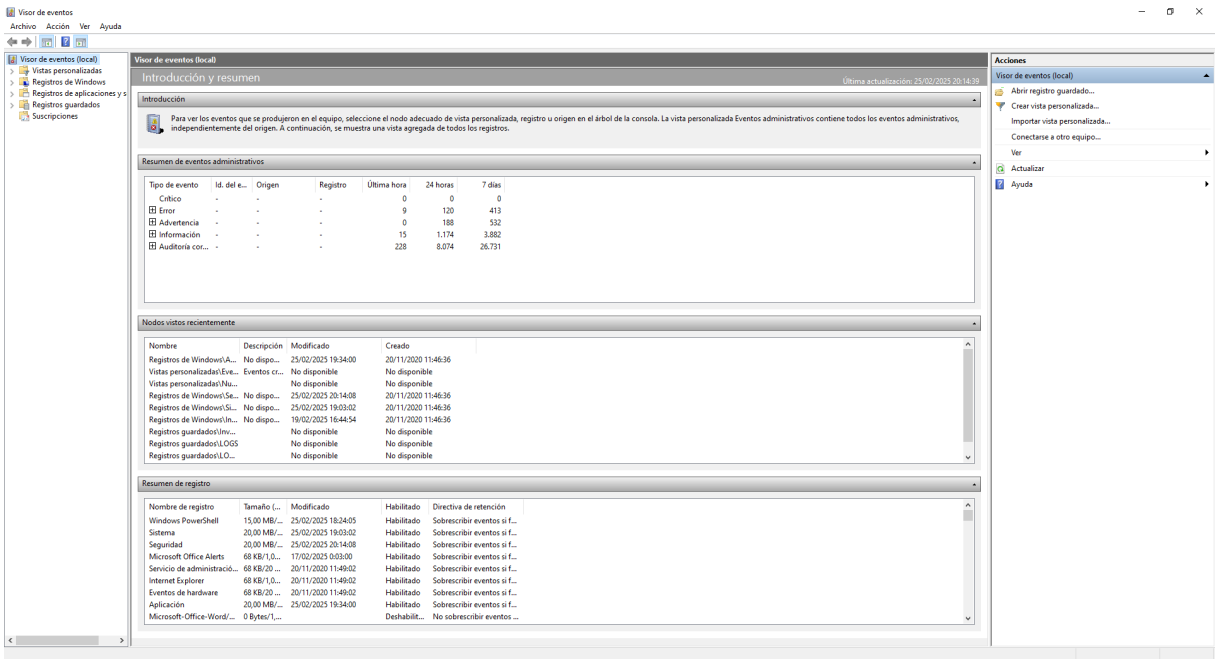

Figura 2.6: Visor de Eventos en Windows 11

2.3.1. Evolución del visor de eventos

La primera versión del sistema operativo Windows que introdujo por primera vez una herramienta de monitorización de eventos fue Windows NT 3.5 en el año 1995. Sin embargo, contaba con muchas desventajas frente al visor de eventos con el que cuenta hoy en día Windows, ya que ese servicio solo permitía registrar procesos locales y tampoco era posible acceder de manera remota.

El visor de eventos de Windows utiliza un formato de archivo propio: ***EVTX***, introducido con la llegada de *Windows Vista* y *Windows Server 2008*. La llegada de este nuevo formato de eventos mejoró significativamente la eficacia en el uso de recursos del sistema, así como la estructura de los registros de eventos. A día de hoy, este formato prevalece y se sigue actualizando hasta las versiones actuales de Windows.

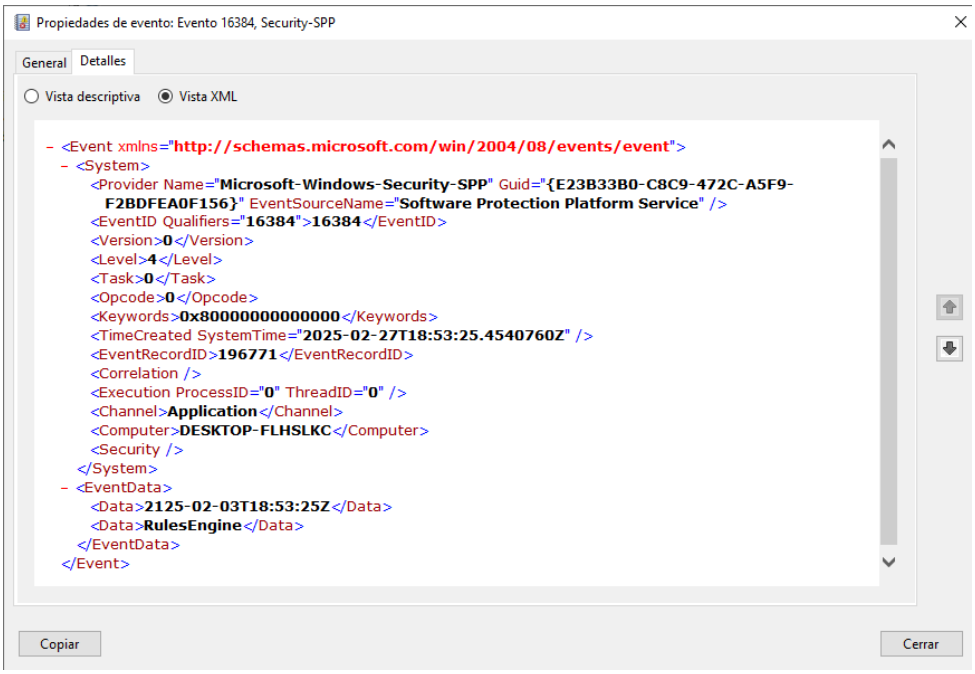

Figura 2.7: Ejemplo de detalle de un log de aplicación en formato XML

Este formato emplea un *XML* binario (Figura 2.7) para registrar los datos, facilitando la interpretación y el análisis. Dentro de un archivo con el formato ***EVTX*** podemos distinguir tres componentes principales [18]:

1. **Encabezado del archivo**: Este encabezado siempre comienza con la cadena *ElfChnk*, proporcionando información básica como el número de fragmentos o el tamaño del archivo. Representado con el nombre ELF_LOGFILE_HEADER en la Figura 2.8.

2. **Fragmentos**: Almacena registros de los eventos y tiene un tamaño de *64 KB*. Cada uno de los fragmentos contiene plantillas *XML* que ayudan a definir cada uno de los objetos.

3. **Registros de eventos**: Contienen información detallada de cada uno de los eventos, como la fecha y la hora de creación, una cadena de identificación, indicadores de longitud para permitir la navegación bidireccional entre los registros, el tipo de proceso que generó el evento y demás datos relevantes, almacenados en formato *XML*. Es representado como EVENTLOGRECORD en la Figura 2.8, ELF _EOF_RECORD se refiere a un registro de fin de archivo.

Windows utiliza una función llamada *ReportEvent*, encargada de escribir una nueva entrada al final del registro de eventos especificado. Cuando una aplicación llama a esta función para escribir esa nueva entrada, este servicio de registros usa esa información para construir la estructura del evento, como se muestra en el diagrama de la Figura 2.8.

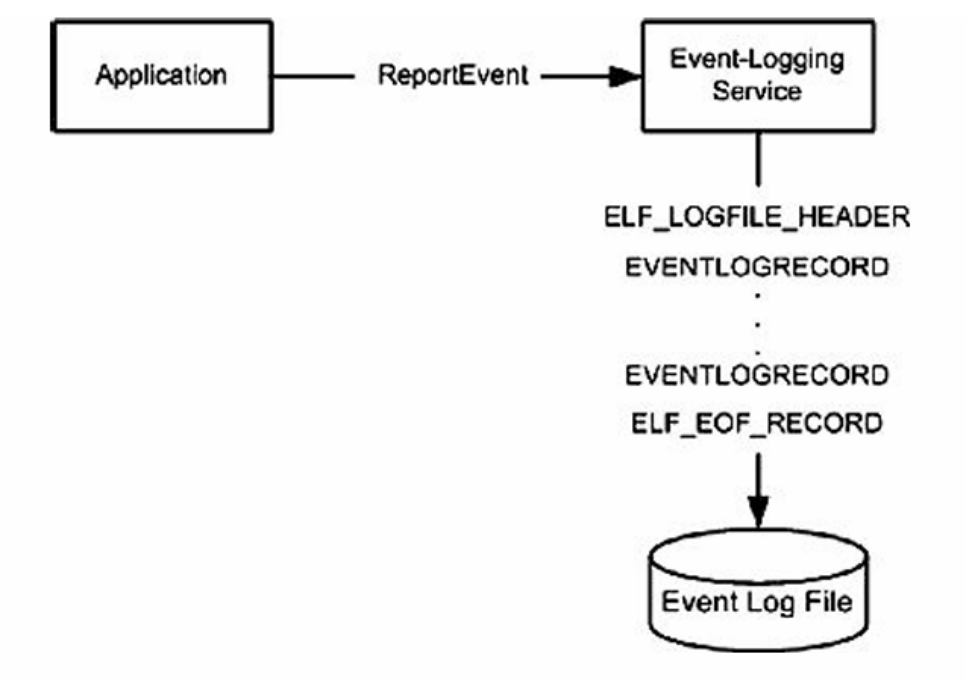

Figura 2.8: Componentes EVTX Adaptado de Microsoft, Event Log File Format

2.3.2. Acceso y utilidades del visor de eventos

El visor de eventos en Windows 11 puede ser accedido de diversas maneras, una de ellas es mediante el menú de inicio, la búsqueda de aplicaciones, el panel de control y **PowerShell** ejecutando el comando ***eventvwr***.

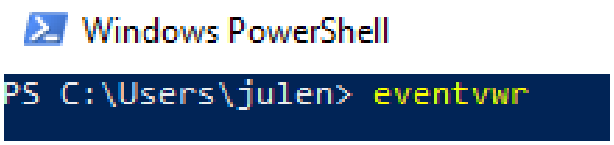

Figura 2.9: Comando *eventvwr* ejecutado en PowerShell de Windows 11

Gracias a sus actualizaciones, desde su incorporación en *Windows*

Vista y *Windows Server 2008*, la interfaz del visor de eventos ha sido actualizada, proporcionando una experiencia mucho más intuitiva de cara al usuario, siendo ahora posible ver diferentes eventos de múltiples registros, guardar filtros para diferentes conjuntos de eventos, crear vistas personalizadas y programar diferentes tareas en respuesta a eventos específicos.

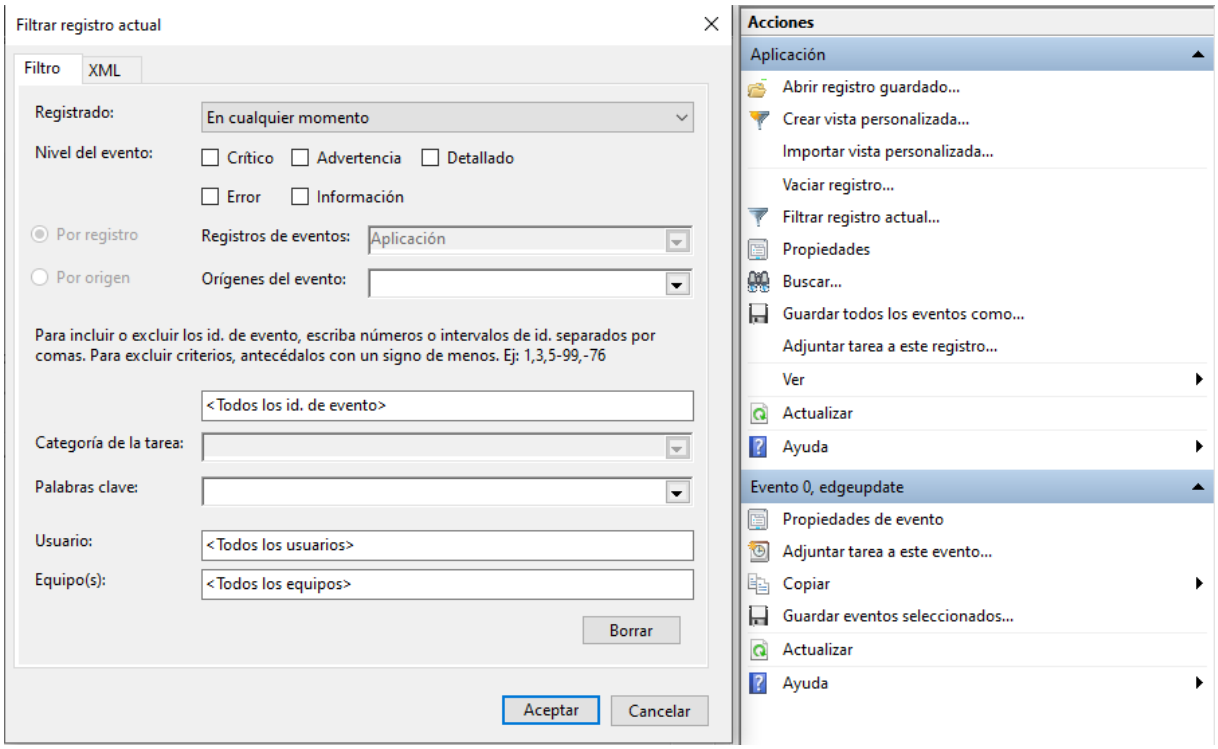

Figura 2.10: Diferentes Opciones de filtrado y visualización desde el visor de eventos en Windows 11

2.3.3. Tipos de Logs

Existen diferentes formas para clasificar los logs dependiendo de a qué característica se quiera hacer referecia [3].

- **Clasificación por Formato**: Se refiere a la manera en la que los logs estructuran, almacenan y organizan la información.

  1. Logs Estructurados: Siguen formatos definidos, por ejemplo, pares *key-value*, facilitando su almacenamiento y análisis. Consistentes y fáciles de procesar. Un log estructurado puede incluir campos como

`timestamp`, `cluster`, `node`, `rackId`, `API` y `version`.

2. Logs No Estructurados: Escritos en texto libre, legible por las personas. No siguen formatos predeterminados, proporcionando una alta flexibilidad pero dificultando su análisis y procesado automático. Un log semiestructurado puede tener un `timestamp`, `cluster`, `node` y un `message` en formato *JSON*.
3. Logs Semiestructurados: Combinación de logs estructurados y no estructurados. Mantienen un formato consistente permitiendo cierta flexibilidad en la estructura. Un log no estructurado puede incluir un `timestamp` y un `message` en texto libre, como: "2025-02-27 19:50:00 COM Receive request 3754984 through GetMessag".

Structured Log	**Timestamp**	**Cluster**	**Node**	**RackId**	**API**	**Version**
	2021-01-01 02:51:26	C01	N01	2_17	GetMessage	**v1**
	2021-01-01 02:51:28	C01	N03	2_18	GetMessage	**v2**

Unstructured Log	**Message**
	2021-01-01 02:51:26 COM Receive request 3754984 through GetMessage
	2021-01-01 02:51:28 COM Executing the request takes 0.47356277 seconds

Semi-Structured Log	**Timestamp**	**Cluster**	**Node**	**Message**
	2021-01-01 02:51:26	C01	N01	{"Version": "v1", "Message": Exception when processing request 89953}
	2021-01-01 02:51:26	C01	N03	{"Version": "v2", "Message": Exception when processing request 46352}

Figura 2.11: Muestras de registros estructurados, no estructurados y semiestructurados. Adaptado de An Empirical Study of Log Analysis at Microsoft. Microsoft Research [3].

- **Clasificación por Nivel de Evento**: La clasificación por nivel de evento se refiere a la gravedad o importancia para el sistema del evento registrado. Los diferentes niveles de severidad que podemos distinguir son:

  1. Crítico: Este nivel indica que hay un problema grave en el sistema que requiere de atención inmediata. Es prioritario solucionar la

fuente del problema y pueden afectar significativamente al sistema.

2. Error: Alerta de problemas significativos que pueden suponer la pérdida de información o la incapacidad para completar alguna funcionalidad del sistema. Son importantes a la hora de identificar y solucionar problemas graves que puedan ocasionarse.
3. Advertencia: Señala eventos que puedan suponer posibles problemas futuros. Son especialmente útiles para tomar medidas preventivas y eludir errores en el sistema.
4. Detallado: Proporciona información detallada sobre el funcionamiento del sistema, especialmente útil para el diagnóstico y para la resolución de problemas, ya que ofrece un análisis exhaustivo de las operaciones.
5. Información: Registra eventos generales que no representan problemas. Proporciona datos sobre el funcionamiento normal del sistema que resultan especialmente útiles en auditorías.

- **Clasificación por tipo de registro**: Este modo de clasificar los eventos hace referencia a su origen o propósito, ayudando a los administradores del sistema a encontrar registros en función de su fuente y permitiendo una monitorización más detallada y específica. Los tipos de registro son:

  1. *Application Logs*: Eventos específicos de aplicaciones y programas, como confirmaciones de descarga, errores en aplicaciones o actualizaciones de bases de datos. Sirven sobre todo para llevar un seguimiento correcto de las aplicaciones del sistema.
  2. *Security Logs*: Registran eventos relacionados con la seguridad del sistema y son utilizados para asegurar la integridad del mismo. Estos logs pueden registrar eventos como intentos de inicio de sesión, accesos a recursos y cambios en los privilegios del sistema.
  3. *System Logs*: Este tipo de registro contiene eventos que son registrados por componentes del sistema operativo, como pueden ser eventos hardware o fallos en los *drivers*, especialmente útiles para diagnosticar problemas del sistema operativo o del hardware.

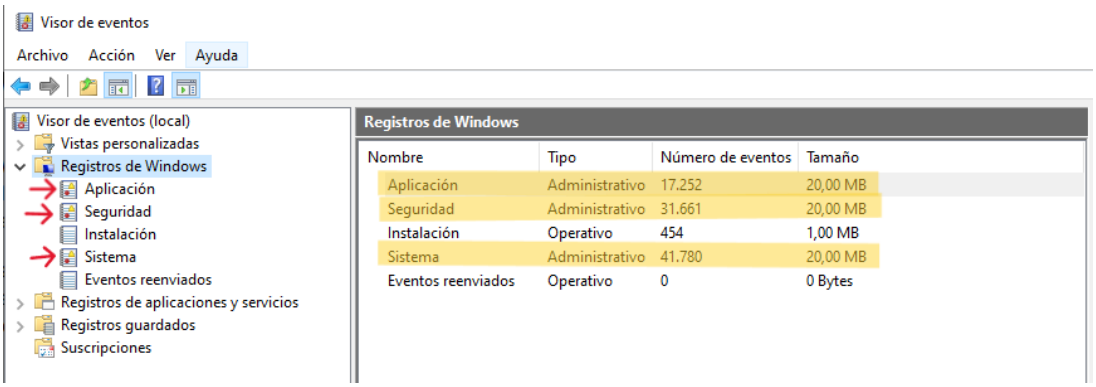

Figura 2.12: Resumen de eventos clasificados por tipo de registro en el Visor de Eventos de Windows 11

2.4. Análisis Forense

El análisis forense es una ciencia o disciplina que se encarga de recolectar, mantener y analizar pruebas y evidencias digitales que puedan aportar información útil para la investigación de delitos informáticos, de manera que mantengan su integridad y puedan ser admitidas y evaluadas en un tribunal de justicia. El análisis forense informático cuenta con varias fases [19]:

- Recopilación de evidencias: Que consiste en recopilar e identificar datos relevantes de distintas fuentes digitales, empleando diferentes técnicas y herramientas especializadas que permitan realizar copias exactas de los datos originales.
- Preservación de evidencia: Se trata de preservar la integridad de los datos recopilados como evidencias mediante la creación de copias de seguridad forenses, realizadas replicando bit a bit los datos originales, y el establecimiento de procedimientos de protección, como por ejemplo la llamada ´´Cadena de Custodia", que es un procedimiento documentado de identificación, recolección, transporte, registro y garantía de integridad mediante funciones resumen, como **SHA-256** [20], de las evidencias. Estas copias garantizan su validez en procedimientos legales.
- Análisis: Fase donde los investigadores forenses examinan en detalle la

evidencia digital, en busca de actividad maliciosa o fraudulenta que pueda ser de utilidad para la investigación. El análisis puede incluir recuperación de datos eliminados, estudio de los logs o identificación de archivos ocultos, entre otros, para conocer en detalle un incidente de seguridad.

- Documentación y reporte del incidente: Es fundamental documentar cada paso realizado de manera detallada, las conclusiones alcanzadas y cualquier hallazgo relevante. Esta información debe ser presentada como informe válido en procesos legales. El informe debe ser claro y comprensible para personas que no cuenten con un conocimiento profundo de la materia.
- Cumplimiento legal: Cualquier proceso de análisis forense debe realizarse conforme a las leyes y regulaciones aplicables, tanto internacionales (como puede ser el estándar **ISO/IEC 27037:2012** [**21**]), como marcos nacionales (por ejemplo la **Familia UNE 71505:2013** [**22**]). En consecuencia, los investigadores deben estar familiarizados con los marcos legales y regulatorios así como con las leyes de protección de datos y actuar de manera ética durante su investigación.

2.4.1. Análisis Forense de Logs

El análisis de logs y eventos es una parte crucial en el ámbito del análisis forense, centrada en el examen y estudio de los registros de actividad generados por los sistemas informáticos y las aplicaciones. Los logs almacenan información detallada sobre las acciones o modificaciones realizadas en un sistema, lo que permite a los investigadores reconstruir eventos y detectar con exactitud anomalías o actividades sospechosas.
Durante la fase de la realización del análisis forense de los logs, es importante tener en cuenta la correlación de los eventos del sistema, que consiste en combinar la información de múltiples logs para obtener una visión completa del entorno en el que trabajamos. El visor de eventos es una herramienta extremadamente útil a la hora de realizar estos análisis en sistemas operativos Windows, dado que permite conocer información

valiosa de los logs, como podemos observar en la Figura 2.13, ejemplo de un log de seguridad y de la información que podemos extraer del mismo:

- Nombre del servicio: SecurityCenter.
- ID del evento: 4625 .
- Nivel de gravedad: 2 = Error.
- Hora a la que se generó el evento: 2024-03-12T14:35:22.000Z .
- Equipo (útil en caso de querer diferenciar logs entre varias máquinas): PC-Usuario .
- Usuario afectado: admin.
- IP Origen del acceso: 192.168.1.10 .
- KeyWords: etiquetas en hexadecimal para clasificar algunos tipos de eventos, algunas etiquetas clave comunes son:
  - `0x40000000000000` → Diagnóstico.
  - `0x80000000000000` → Información genérica.
  - `0x20000000000000` → Eventos de auditoría.

```xml
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Security-Auditing"/>
    <EventID>4625</EventID>
    <Level>2</Level>
    <TimeCreated SystemTime="2024-03-12T14:35:22.000Z"/>
    <Computer>PC-Usuario</Computer>
  </System>
  <EventData>
    <Data Name="TargetUserName">admin</Data>
    <Data Name="IpAddress">192.168.1.10</Data>
    <Data Name="LogonType">3</Data>
  </EventData>
</Event>
```

Figura 2.13: Ejemplo de log XML de un intento de sesión fallido en (Event ID 4625 - Security Log)

2.5. RPA vs. SIEM

Un SIEM (*Security Information and Event Management*) es una solución de seguridad que permite a las empresas gestionar y responder a amenazas a su seguridad en tiempo real. Funciona recopilando datos de múltiples fuentes y transformándolos de manera que resulten fáciles de analizar de manera automatizada [23]. Un SIEM analiza los datos en busca de patrones anómalos mediante la incorporación de inteligencia artificial.
Una vez que una amenaza es detectada, se generan alertas a los equipos correspondientes y se proporcionan herramientas para poder responder a la misma. Además, proporciona apoyo en el cumplimiento de auditorías de seguridad.
Las herramientas SIEM más utilizadas por las empresas, dado su eficacia y sus amplias prestaciones, son:

1. Splunk [24].
2. IBM QRadar [25].
3. LogRhythm [26].

4. Cortex XSIAM [27].

A continuación, vamos a comparar las ventajas en lo referente al análisis de logs que nos proporciona la solución RPA propuesta en este proyecto con las prestaciones que ofrece un SIEM, resumiendo la comparación en la Tabla 2.1.

La solución RPA propuesta será descrita en detalle más adelante. Sin embargo, al tratarse de un robot que analiza los logs del sistema mediante un simple *click* desde UiPath Orchestrator y resume el resultado del análisis en un Excel, es muy fácil de configurar, usar y es accesible para usuarios que no cuenten con conocimientos de ninguna de las materias descritas anteriormente. Sin embargo, un SIEM está desarrollado para usarse en el ámbito empresarial y para ser usado por profesionales de la seguridad de los sistemas informáticos. Por ese mismo motivo, la solución RPA tiene un consumo de recursos mucho menor. Además, la licencia de UiPath es más barata (e incluso gratis durante 2 meses) que la contratación de una licencia para aplicar un software SIEM.

Las desventajas del RPA propuesto frente al SIEM residen principalmente en la eficacia del análisis y la cantidad de eventos y registros que pueden analizar cada uno, siendo el SIEM ampliamente superior en términos de análisis y seguridad.

En conclusión, esta solución RPA es especialmente útil para personas o usuarios comunes que quieran analizar la salud de su sistema, mientras que en un contexto profesional sería significativamente superada en todos los sentidos por un SIEM.

	Solución RPA	SIEM
Facilidad de Uso y Accesibilidad	✓	✗
Instalación y Configuración Sencilla	✓	✗
Menor Consumo de recursos	✓	✗
Menor Coste y Mantenimiento	✓	✗
Enfoque Específico y Claro	✓	✗
Análisis Avanzado y Correlación de Eventos	✗	✓
Monitorización Continua y en Tiempo Real	✗	✓
Detección de Amenazas y Respuesta Automatizada	✗	✓
Escalabilidad y Capacidad de Manejo de Grandes Volúmenes de Datos	✗	✓

Tabla 2.1: Tabla comparativa entre la solución RPA propuesta y un SIEM

Capítulo 3

Desarrollo del proyecto

A lo largo de esta sección, se describirá en detalle la solución RPA implementada, así como sus características y su programación. Se explicará con detenimiento cada uno de los componentes de la solución.

3.1. Planteamiento de la herramienta

La idea principal de esta herramienta es facilitar un análisis cómodo y rápido, pero a la vez de gran utilidad de los logs del sistema Windows, para que podamos ser capaces de tener una perspectiva general de la salud de nuestro sistema. Para poder utilizar esta herramienta, simplemente es necesario tener el software de **UiPath** instalado, así como una cuenta en la plataforma.

3.2. Desarrollo y especificaciones

En este apartado se describen cada uno de los componentes de la solución:

Dentro de la carpeta que compone la solución, encontraremos las diferentes subcarpetas que componen cualquier código implementado en UiPath, junto al archivo **main.xaml**, encontraremos adicionalmente una subcarpeta */Data/Input* (Ver Figura 3.1). Dentro de esta subcarpeta, encontraremos dos archivos Excel: **diccionarioEvents** y **templateAnalisisLogs**.

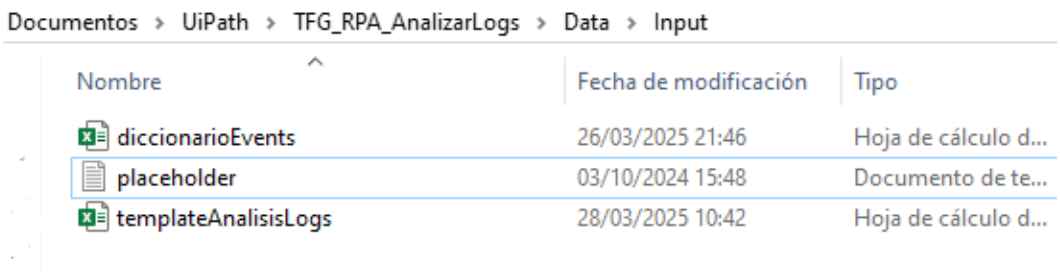

Figura 3.1: Carpeta *Data/Input* que almacena las plantillas Excel

- diccionarioEvents: Este Excel contiene un diccionario con los *EventID* considerados más relevantes [18] en materia de seguridad y salud del sistema. Este diccionario, con estructura *key-value*, relaciona cada uno de los ID de evento con una breve descripción del mismo (Ver Figura 3.2. Una de las principales ventajas de utilizar este diccionario reside en que podemos añadir o retirar logs en función del tipo y profundidad de análisis que queramos realizar.

	A	B
1	EventID	Description
2	4624	Inicio de sesión exitoso
3	4634	Cierre de sesión
4	4625	Intento de sesión fallido
5	1	Modificación de la hora del sistema (posible intento de manipulación de logs)
6	4723	Un usuario intentó cambiar su contraseña en un sistema Windows.
7	4724	Se intentó restablecer la contraseña de una cuenta en un sistema Windows
8	4720	Se creó una cuenta de usuario
9	4725	Una cuenta de usuario fué deshabilitada
10	4726	Una cuenta de usuario ha sido eliminada.
11	4727	Se ha creado un grupo global de seguridad.
12	4728	Se ha añadido un miembro a un grupo global de seguridad.
13	4729	Se ha eliminado un miembro de un grupo global de seguridad.
14	4730	Se ha eliminado un grupo global de seguridad.
15	4731	Se ha creado un grupo local de seguridad.
16	4732	Se ha añadido un miembro a un grupo local de seguridad.
17	4733	Se ha eliminado un miembro de un grupo local de seguridad.
18	4734	Se ha eliminado un grupo local de seguridad.
19	4735	Se ha modificado un grupo local de seguridad.
20	4737	Se ha modificado un grupo global de seguridad.
21	4755	Se ha modificado un grupo universal de seguridad.
22	4756	Se ha añadido un miembro a un grupo universal de seguridad.

DiccionarioEvents

Figura 3.2: Diccionario de Eventos

- templateAnalisisLogs: Este Excel conforma una plantilla para lo que será el futuro Excel resultado (Ver Figura 3.3. El robot copiará esta plantilla durante su ejecución en las carpetas que haya creado y será donde quede plasmado el resumen y análisis de los logs.

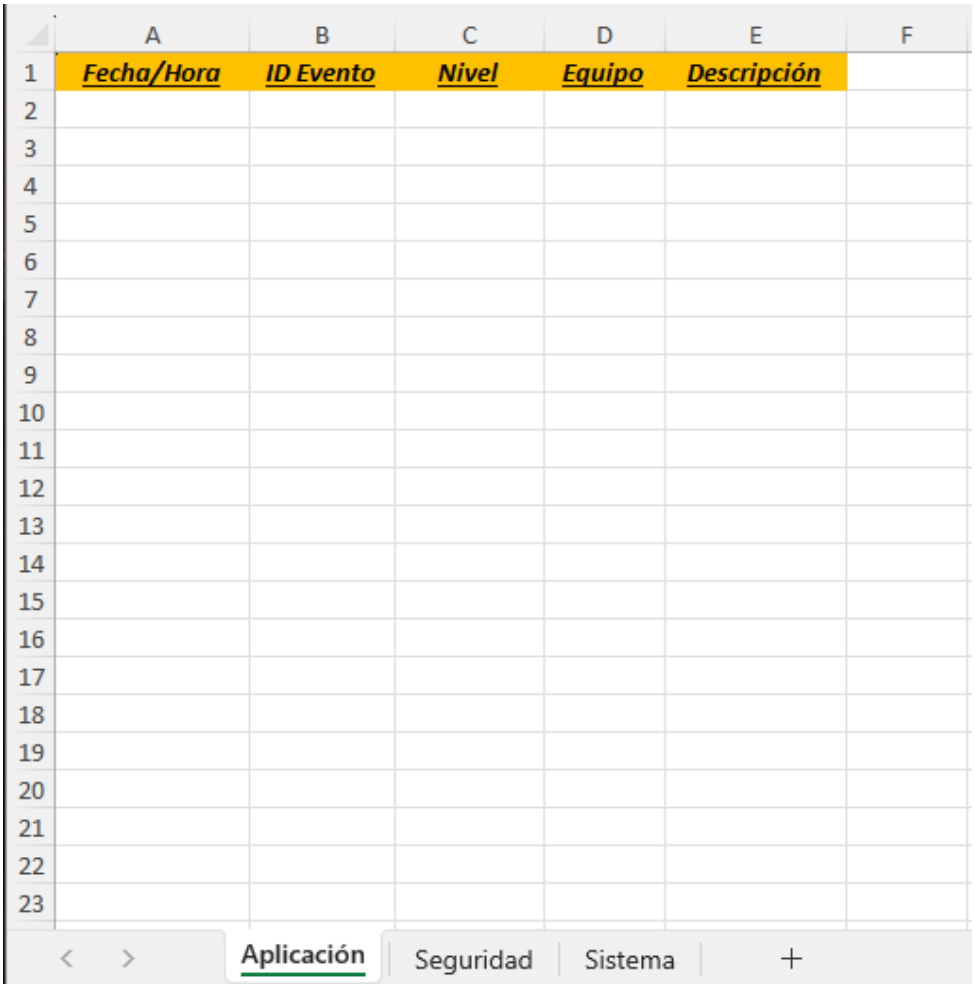

Figura 3.3: Estructura de la plantilla

- Config: Este archivo Excel se encuentra en la ruta: */Data*. Se trata del archivo de configuración que viene por defecto en todos los proyectos que utilizan REFramework. Este archivo contiene el nombre de las variables, el valor que tendrán durante la ejecución y una breve descripción. Además, cuenta con una hoja para configuración, otra para constantes y una última para *Assets* que se gestionan desde **UiPath Orchestrator**. Sin embargo, en el caso de esta solución implementada, solo utilizaremos la hoja de Configuración (Ver Figura 3.4).
  - *tipoLogs*: Variable que contiene el nombre de los 3 tipos de registro de logs separados por comas, en un único *String*.
  - *nombreXml*: *String* creado de manera que el robot sustituye el tipo de logs y la fecha según conveniencia y da formato al archivo en

xml.

- *PathInConfig*: Determina la ruta donde se van a crear las carpetas resultantes. En este caso, en la ruta */Downloads* del equipo donde se ejecute.

	A	B	C
1	**Name**	**Value**	**Description**
2	OrchestratorQueueName	ProcessABCQueue	Orchestrator queue Name. The value must match with the queue name defined on Orchestrator.
3	OrchestratorQueueFolder		Folder name. The value must match a folder defined in Orchestrator and queue specified as OrchestratorQueueName should be created in this folder. For classic folders leave the value field empty.
4			
5	logF_BusinessProcessName	Framework	Logging field which allows grouping of log data of two or more subprocesses under the same business process name
6			
7	tipoLogs	Aplicación,Sistema,Seguridad	
8	nombreXml	{tipoLog}_{fecha}.xml	
9	PathInConfig	C:\Users\{usuario}\Downloads	
10			

Figura 3.4: Archivo de Configuración de la solución implementada

Como ya se ha mencionado anteriormente, esta solución está implementada utilizando como base el modelo *REFramework*; sin embargo, ha sido adaptada a las necesidades de la misma (Ver Figura 3.5).

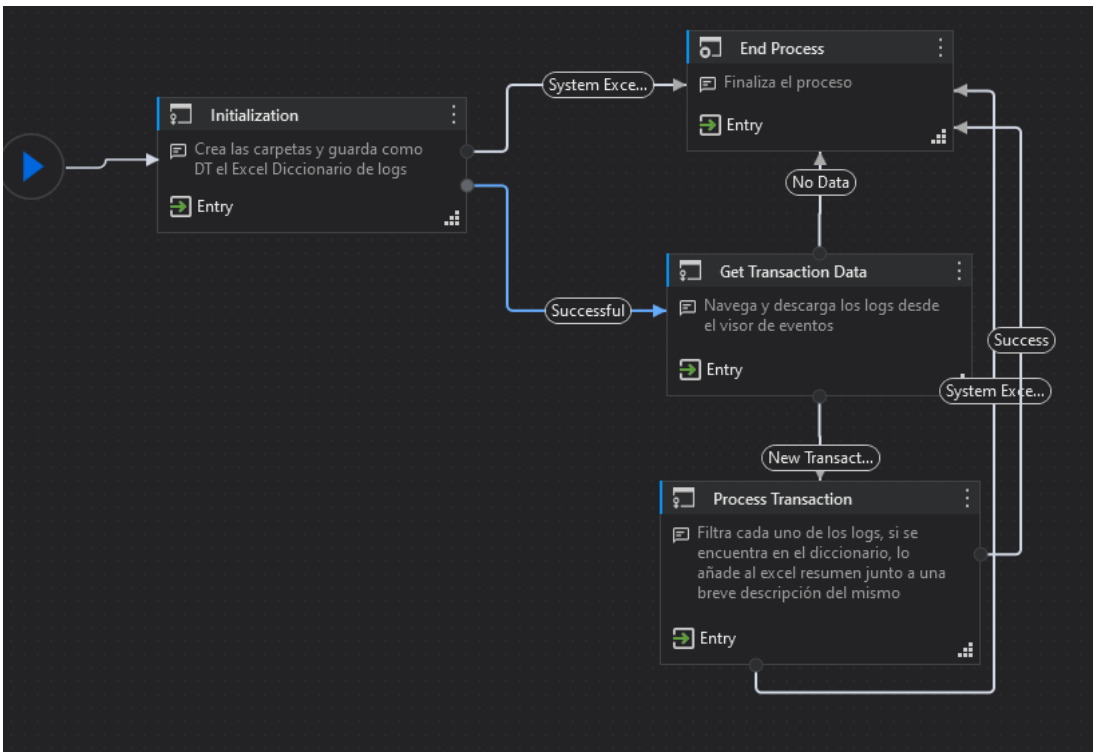

Figura 3.5: REFramework State Machine de la Solución implementada

A lo largo de la descripción de cada uno de los estados, se omitirán

Figuras de los componentes y actividades que ya formen parte del propio REFramework. Los diferentes estados de la *State Machine* que conforman la solución son:

- Inicialization: Se trata de un estado formado por un *Try-Catch*, donde el *Try* inicializa la variable *SystemException* en *Nothing* y continúa leyendo el archivo de configuración anteriormente descrito para inicializar todas sus variables. Después, realiza las siguientes acciones:

  1. Invoca al *Workflow: CrearCarpetas* (Ver Figuras 3.6 y 3.7), encargado de establecer las rutas donde se almacenarán los logs descargados y el Excel resultante, comprobando si, en caso de que hubiera existido una ejecución previa, las carpetas no están ya creadas.

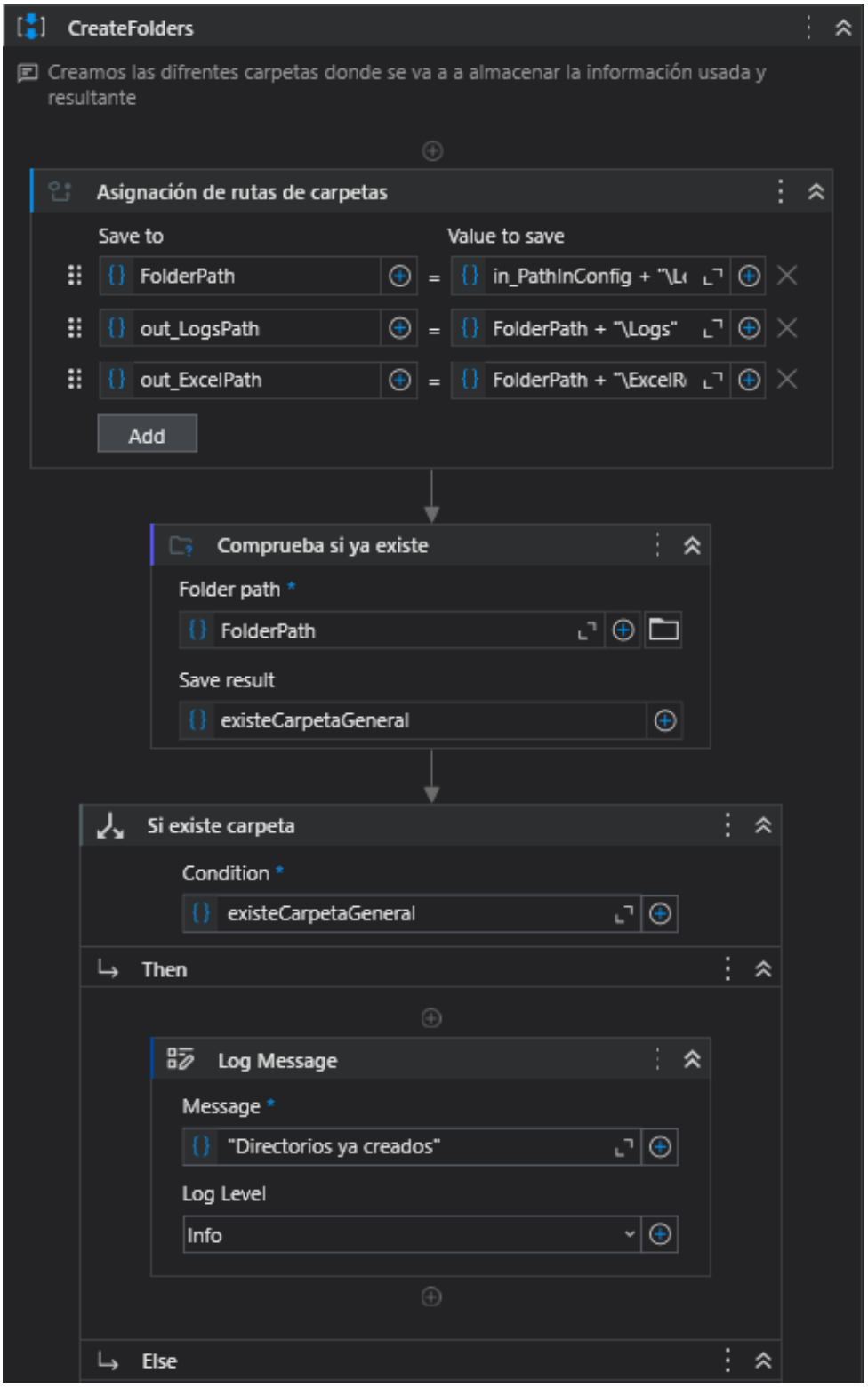

Figura 3.6: WorkFlow Crear Carpetas Parte 1

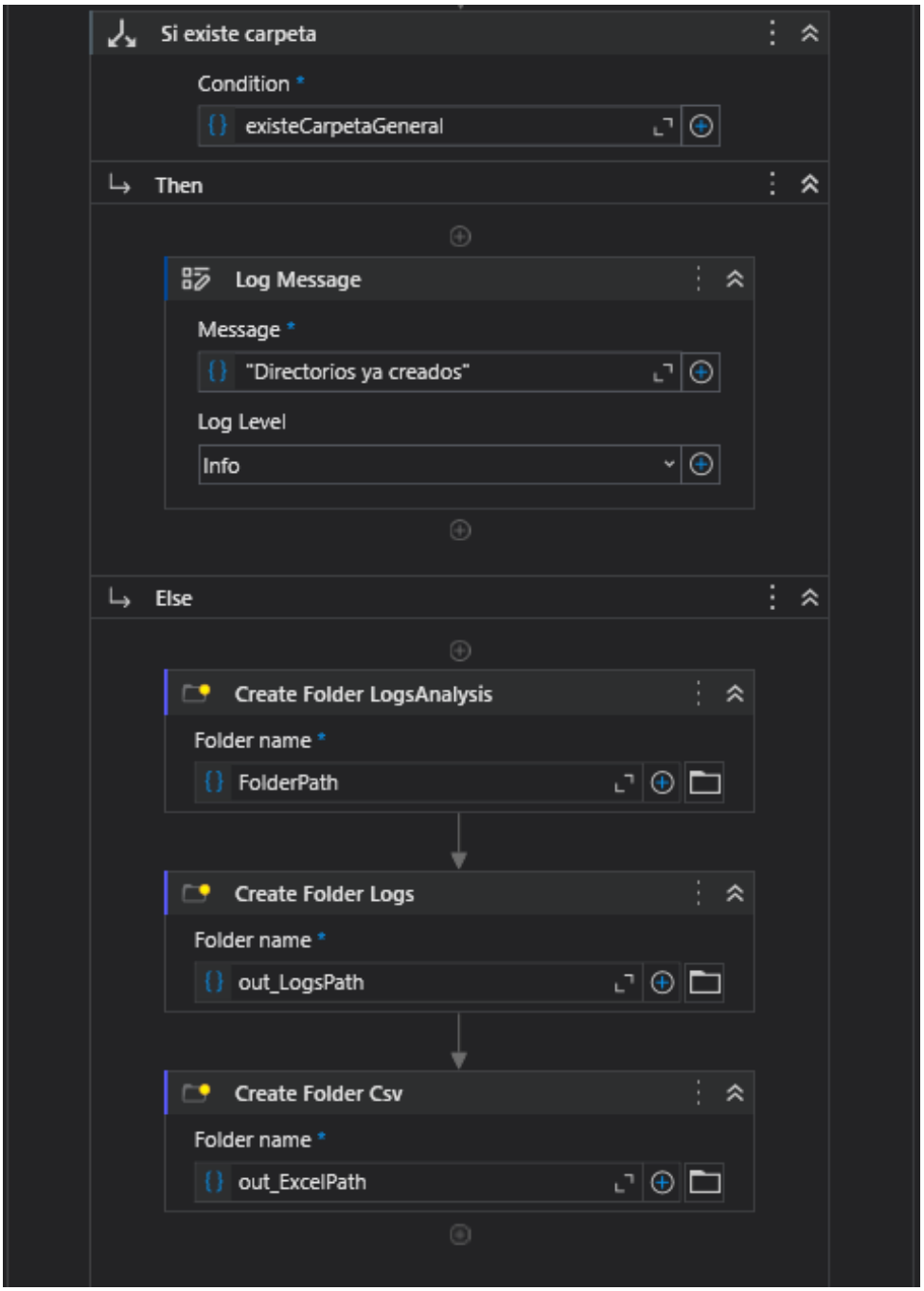

Figura 3.7: WorkFlow Crear Carpetas Parte 2

2. Crea el Excel resultado mediante la concatenación de la ruta de almacenaje y la fecha actual, usando la función *Now* (Ver Figura 3.8).

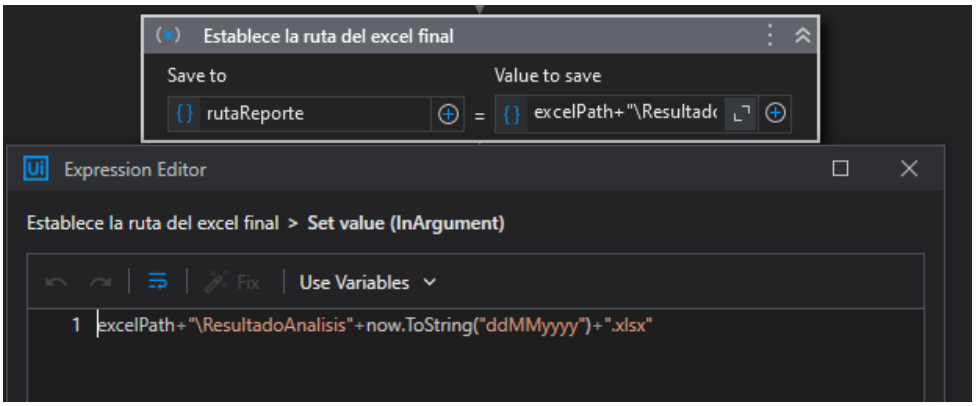

Figura 3.8: Definición de *rutaReporte*

3. Crea la *DateTable* con el diccionario de eventos anteriormente mencionado (Ver Figura 3.9).

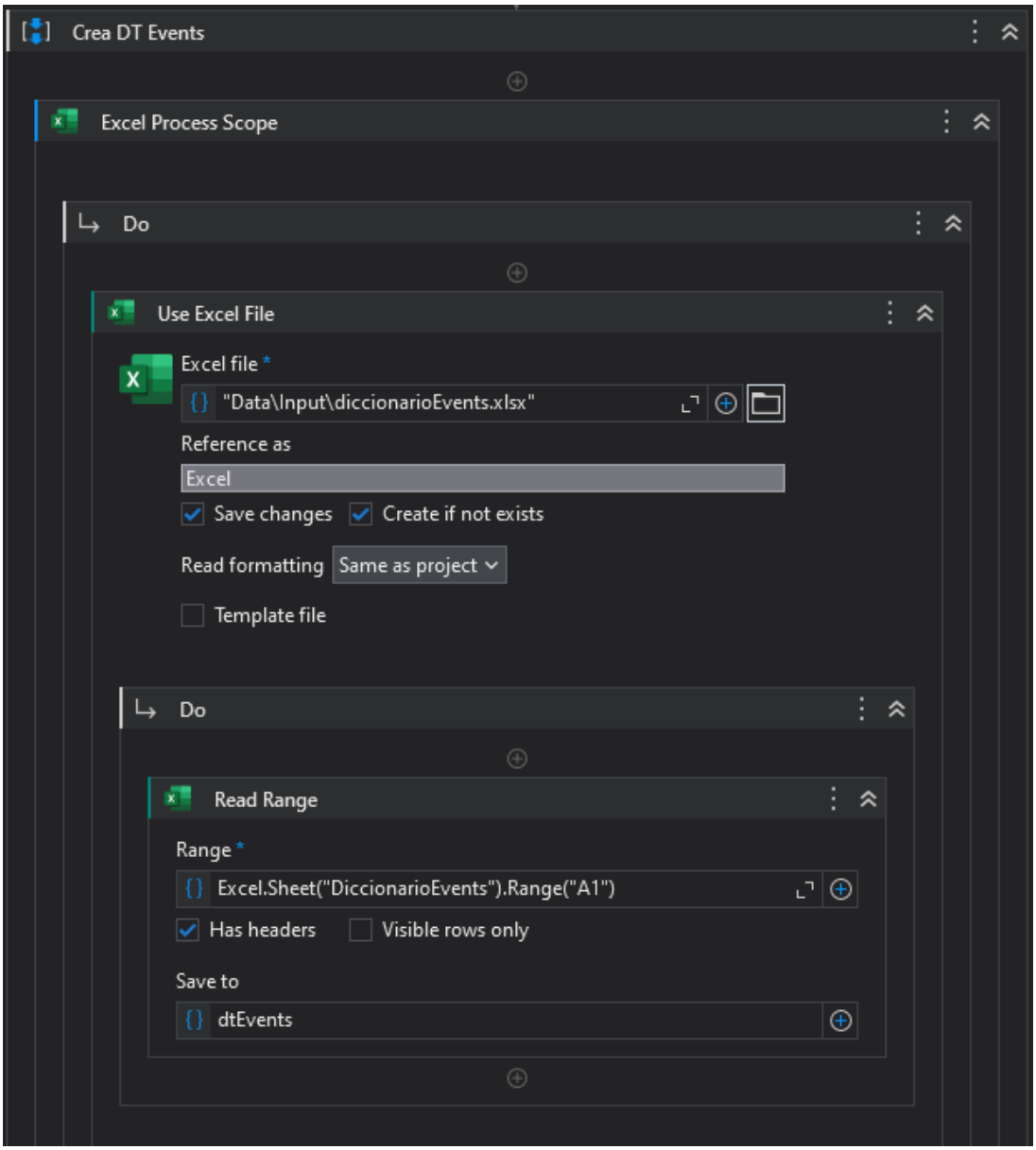

Figura 3.9: Crea la *DataTable* del diccionario de eventos mediante actividades Excel

4. Las salidas de este estado siguen la siguiente estructura:
   - Get Transaction Data: En caso de ejecución exitosa.
   - End Process: En caso de error, donde la variable *SystemException != Nothing*.

- Get Transaction Data: Este estado se encarga de preguntar al usuario la antigüedad de los logs que quiere resumir y analizar (Ver Figura 3.10).

Una vez obtenido el filtro de fecha por parte del usuario, comienza con la secuencia de extracción de los logs, donde mediante un bucle recorre los tres tipos de logs y descarga el XML correspondiente a cada uno de ellos.

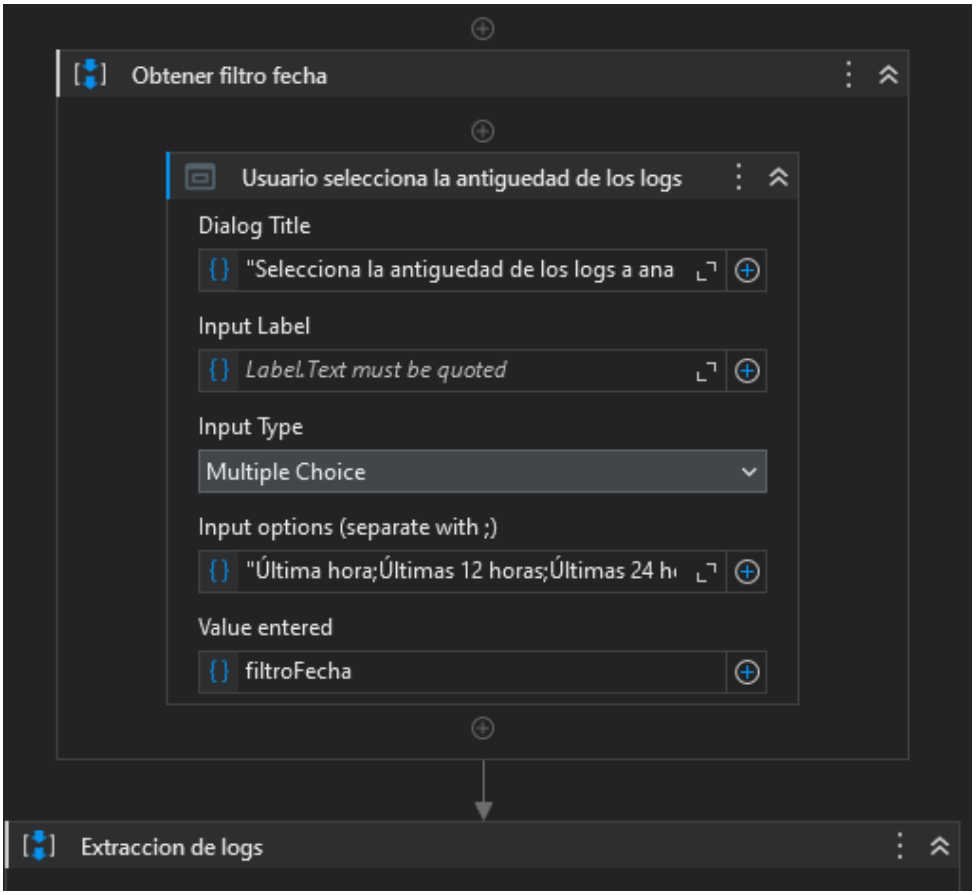

Figura 3.10: Actividad para que el usuario introduzca la antigüedad de los logs que prefiere mediante un desplegable

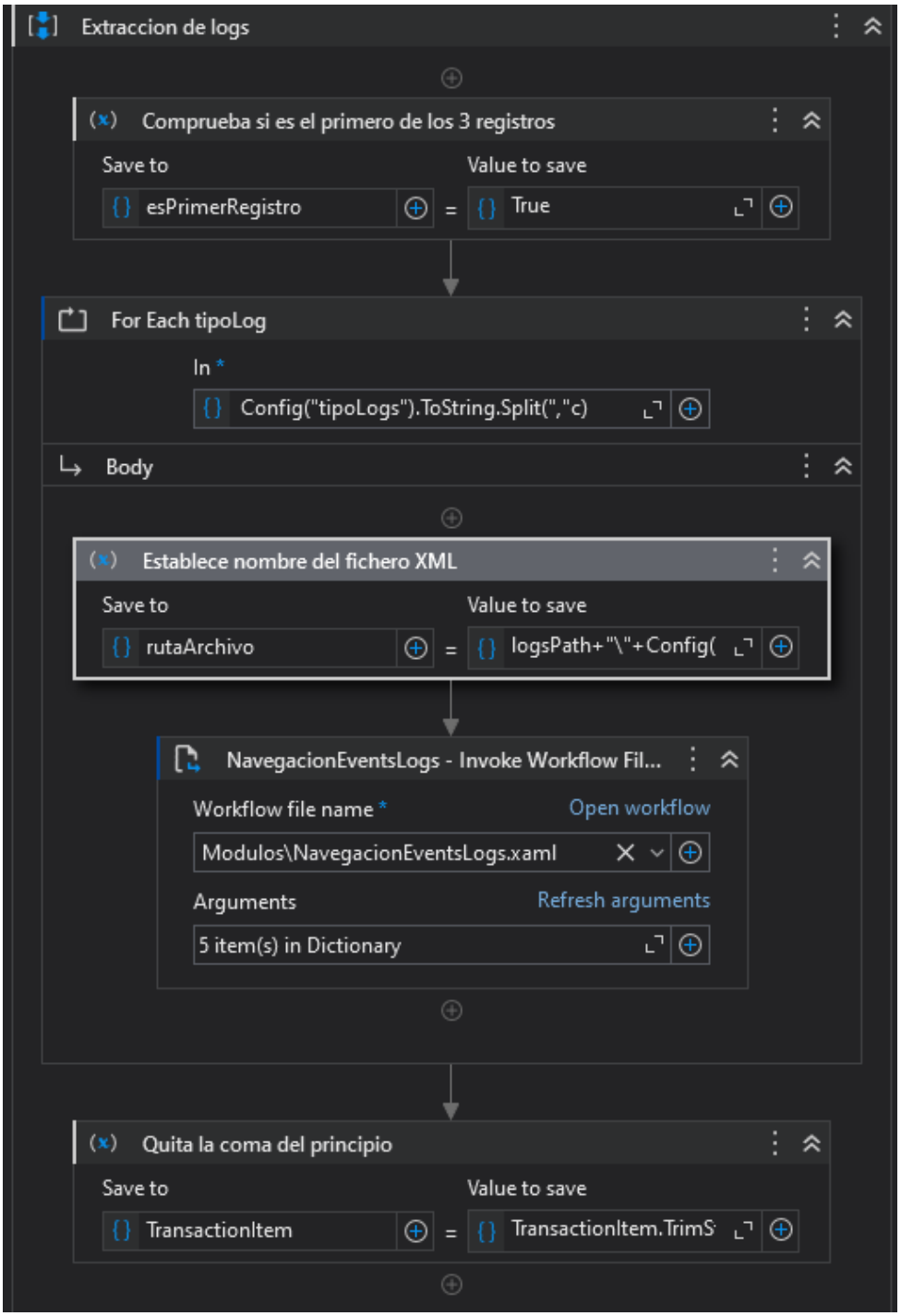

Figura 3.11: Secuencia de extracción de logs

1. NavegaciónEventsLogs (Ver Figura 3.12): Este *workflow* se encarga de abrir el Visor de Eventos y navegar mediante actividades que simulan los *clicks* de un usuario hasta el registro de logs pertinente a su iteración del bucle anterior, aplicar el filtro de fecha establecido

por el usuario, y finalmente descargar los archivos XML en la ruta correspondiente, con su nombre asignado anteriormente. Cada una de las actividades que simulan *clicks* del usuario cuentan con una configuración independiente, llamada selector, para identificar y seleccionar el elemento o botón pertinente dentro de la interfaz de usuario (Ver Figura 3.13). En este caso, usamos el *Strict Selector* que aportan precisión a la hora de encontrar el elemento concreto, independientemente de en qué lugar de la pantalla este situado, o de la resolución de la misma.

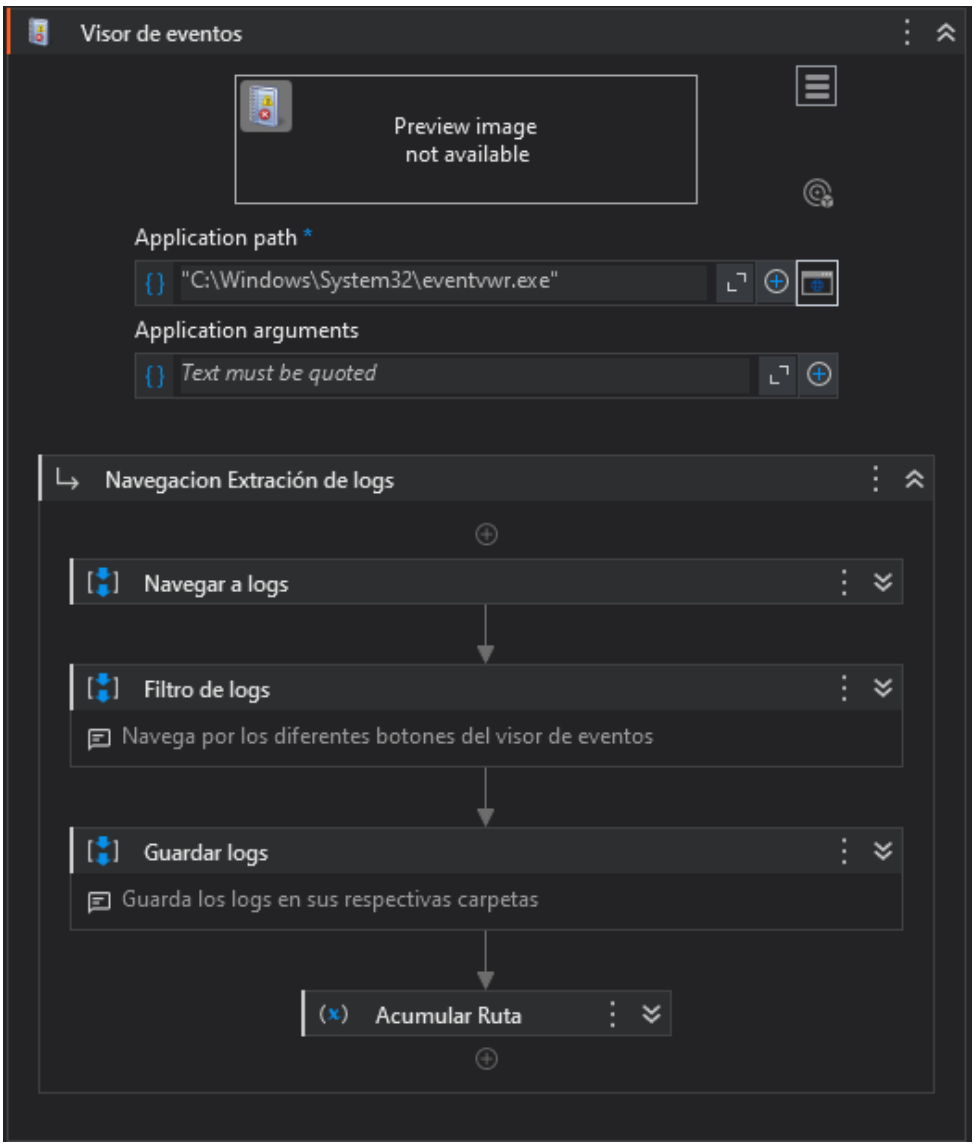

Figura 3.12: Workflow NavegacionEventsLogs con sus distintos componentes encampsulados

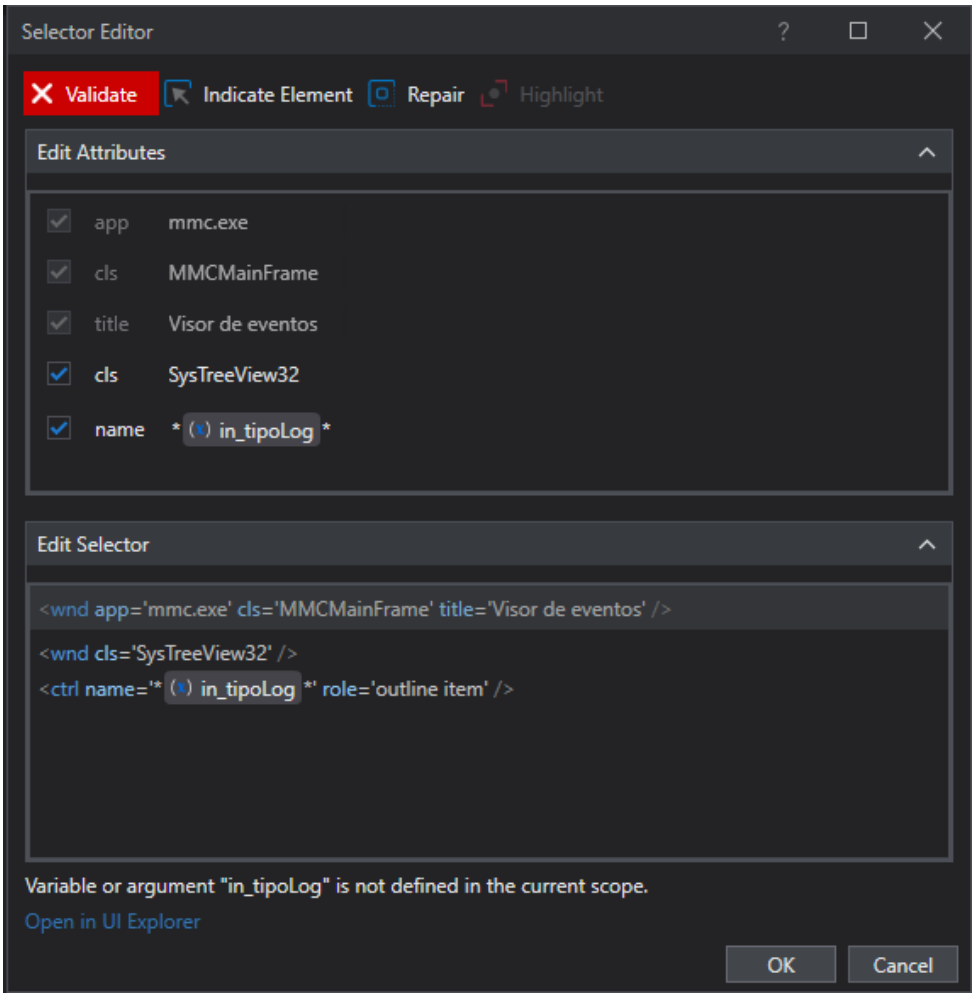

Figura 3.13: Configuración del *Strict Selector* para que busque el botón del Visor de eventos que coincida con el registro actual almacenado en el argumento *in_tipoLog*.

2. Las salidas de este estado siguen la siguiente estructura:
   - Process Transaction: En caso de ejecución exitosa.
   - End Process: En caso de error, donde la variable *SystemException != Nothing*.

- Process state: Este estado invoca al *Workflow: Process*. Durante su ejecución, construye la *Date Table* final, que será trasladada al Excel Resumen. Después, itera entre los tres tipos de registros de logs y, por cada uno de ellos, lleva a cabo las siguientes acciones:

1. Lee el archivo XML referente al tipo de registro correspondiente a esa iteración descargado en el estado anterior, y almacena todos los logs en un mismo *String* llamado *xmlLogsBruto*.
2. Carga el *String* anterior en un tipo de variable llamado **XDocument**, utilizado para crear diccionarios a los que se les pueda aplicar *Linq* [28].
3. Aplica una extensa consulta mediante *Linq* (Ver Figura 3.14), para comparar entre el diccionario de logs *diccionarioEvents* y los logs del registro que esté analizando en esa iteración. En el momento que encuentra una coincidencia en el campo *EventID*, extrae los datos de los diferentes campos de interes de ese log y lo agrega a la *Data Table* final.

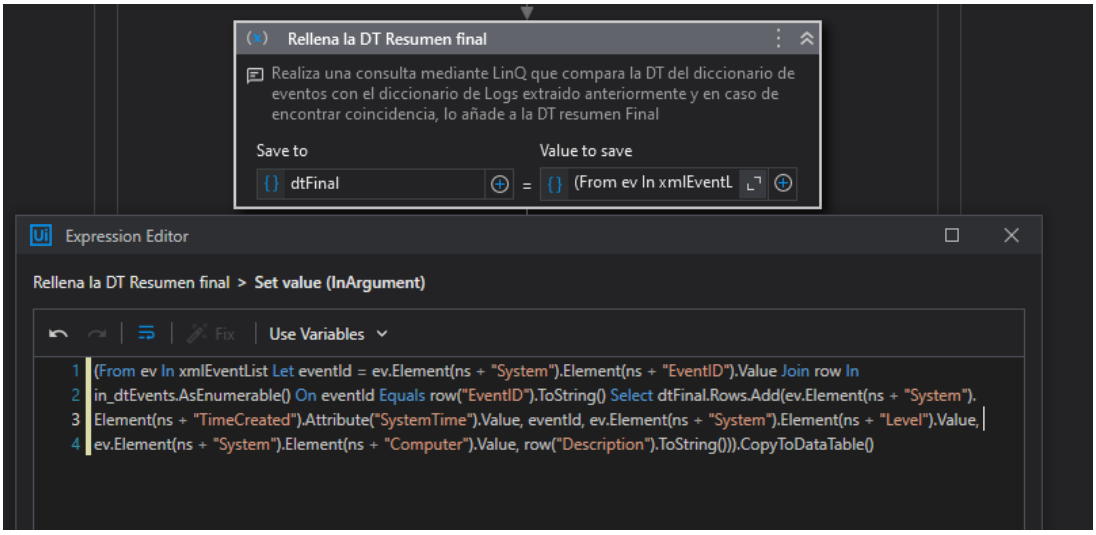

Figura 3.14: Linq que compara el diccionario de eventos con cada log extraído del sistema.

4. Por último, utilizando actividades propias de UiPath para escribir en Excel, copia la *Data Table* final a la hoja correspondiente a la iteración actual, del Excel *templateAnalisisLogs*, finalizando así su ejecución.

Capítulo 4

Resultados y ejecución

En esta sección, se expondrán los resultados obtenidos y un ejemplo de ejecución, así como los diferentes desafíos encontrados a la hora de diseñar e implementar la solución.

4.1. Resultados y ejecución

El tiempo de ejecución medio del robot, con la configuración descrita en esta memoria, es de aproximadamente dos minutos, dependiendo de la antigüedad de los logs que queramos analizar. En primer lugar, el robot genera las carpetas en la ruta mencionada, así como el Excel donde se imprimirán los resultados (Ver Figura 4.1).

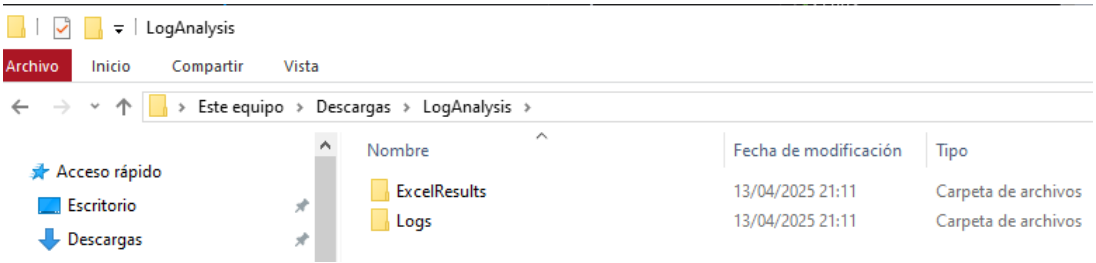

Figura 4.1: Carpeta *LogAnalysis* creada por el robot durante los primeros segundos de su ejecución

Inmediatamente después de crear las carpetas, el robot pregunta al usuario mediante un desplegable las diferentes posibilidades referentes a la antigüedad de los logs, para almacenar ese valor y poder aplicarlo como filtro durante la navegación por el visor de eventos (Ver Figura 4.2).

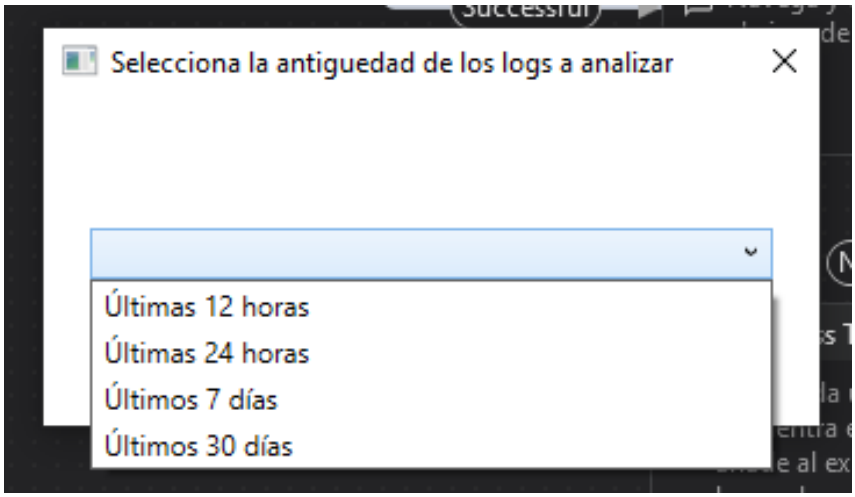

Figura 4.2: Desplegable que ofrece al usuario las diferentes opciones de filtrado referentes a la antigüedad de los logs.

Una vez seleccionado el filtro, el robot continúa su ejecución, dando inicio a la navegación a través del visor de eventos. Durante este proceso, el robot abre el visor de eventos y navega a través de las diferentes pestañas hasta que localiza los registros pertinentes mediante los *clicks* simulados. Después, reconoce la opción **Filtrar registro actual**, que abre un desplegable en el cual el robot establece el filtro seleccionado por el usuario (Ver Figura 4.3).

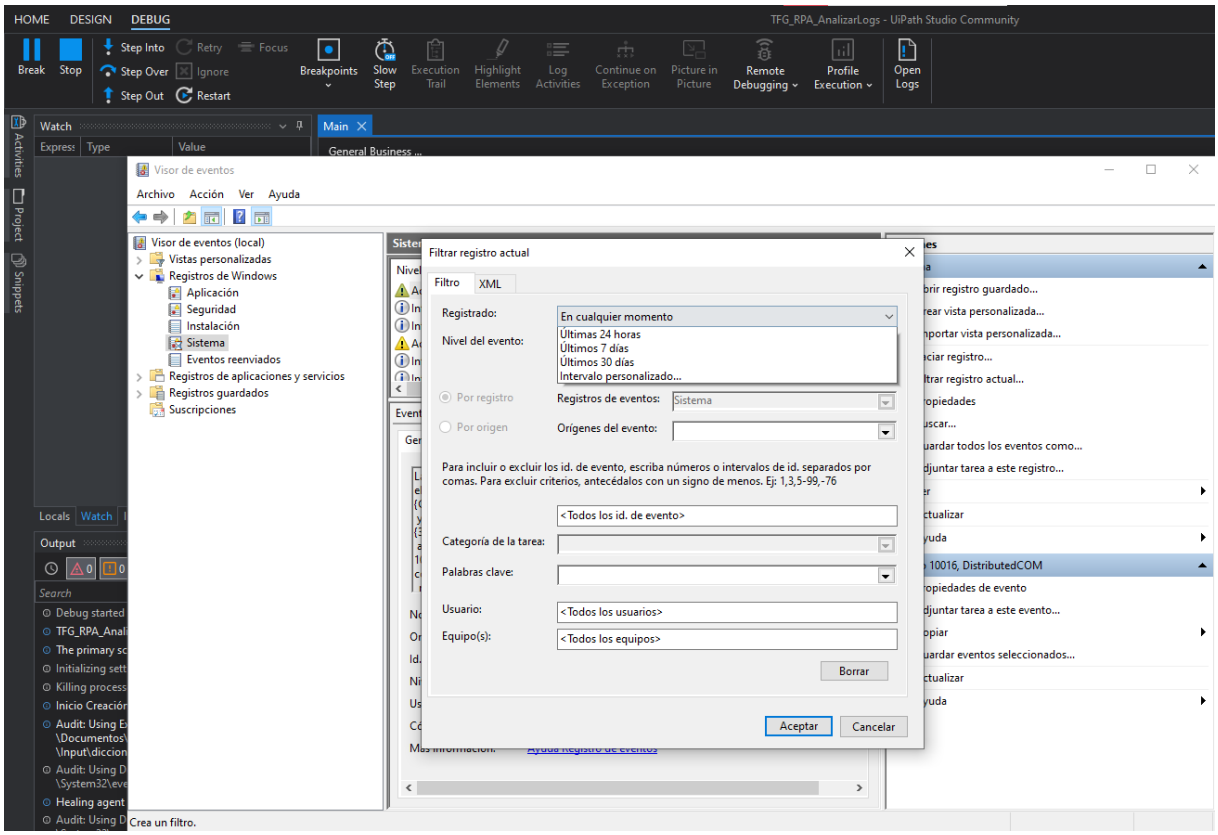

Figura 4.3: Captura de la ejecución del robot durante la fase de navegación a través del visor de eventos

Los logs descargados de los tres registros del sistema quedan almacenados en las carpetas creadas al inicio de la ejecución, lo que permite a un analista tener acceso a la fuente de evidencia original (Ver Figura 4.4).

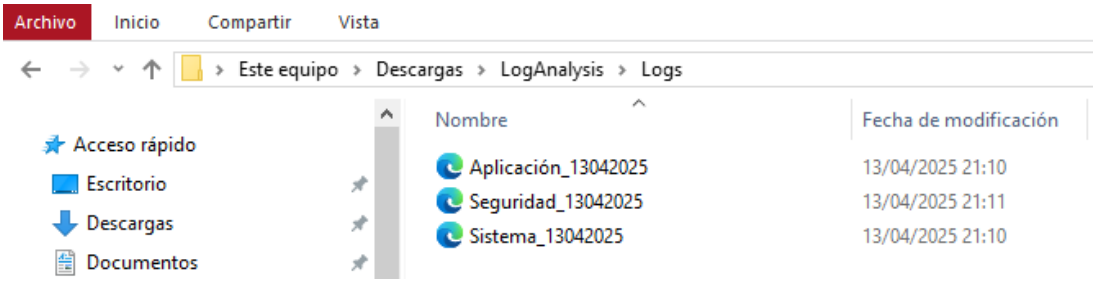

Figura 4.4: Logs almacenados en formato XML con un nombre descriptivo formado por el tipo de registro y la fecha actual en formato *dd-MM-aaaa*.

Finalmente, la ejecución termina con la lectura y comparación descrita anteriormente de cada uno de los eventos extraídos, sobreescribiendo el

Excel inicial con cada uno de los logs y una breve descripción de los mismos (Ver Figura 4.5).

	A	B	C	D	E
1	Fecha/Hora	ID Evento	Nivel	Equipo	Descripción
2	2025-04-26T14:26:21.1307004Z	1	4	DESKTOP-FLHSLKC	Modificación de la hora del sistema (posible intento de manipulación de logs)
3	2025-04-26T14:24:10.6639588Z	4625	4	DESKTOP-FLHSLKC	Intento de sesión fallido
4	2025-04-26T12:01:47.8317931Z	4625	4	DESKTOP-FLHSLKC	Intento de sesión fallido
5	2025-04-26T11:59:14.9076799Z	1	4	DESKTOP-FLHSLKC	Modificación de la hora del sistema (posible intento de manipulación de logs)
6	2025-04-26T11:57:04.8106219Z	4625	4	DESKTOP-FLHSLKC	Intento de sesión fallido
7	2025-04-26T11:55:04.4390942Z	4625	4	DESKTOP-FLHSLKC	Intento de sesión fallido
8	2025-04-26T11:46:49.7941625Z	4625	4	DESKTOP-FLHSLKC	Intento de sesión fallido
9	2025-04-26T11:44:27.0147472Z	4625	4	DESKTOP-FLHSLKC	Intento de sesión fallido

Figura 4.5: Excel resultante del análisis y el resumen tras la ejecución.

4.2. Problemas encontrados

Durante este apartado, se van a detallar los diferentes retos y problemas encontrados durante el desarrollo del proyecto y sus respectivas soluciones:

- Los selectores utilizados por el robot para detectar e interactuar mediante navegación con los diferentes botones y desplegables de las aplicaciones son muy flexibles, siendo capaces de identificar el elemento con el que tienen que interactuar de manera extremadamente precisa y eficaz. Normalmente, en las diferentes aplicaciones y webs, los robots se programan con *Fuzzy selectors*, que son especialmente flexibles e identifican cambios dinámicos en las webs y aplicaciones. También es común utilizar *Computer Vision*, otro tipo de selector que utiliza técnicas avanzadas de reconocimiento de imágenes para identificar e interactuar con los elementos. Permite a los robots de RPA interactuar con la UI de manera similar a como lo haría un humano, reconociendo elementos visualmente en lugar de basarse en sus propiedades.
  Sin embargo, en el caso de esta solución, dado la antigüedad del visor de eventos, la interfaz es difícil de utilizar para el robot, además de que es muy poco personalizable. El principal problema fue a la hora de aplicar filtros, puesto que el menú de filtrado se abre en una ventana diferente, costosa de identificar para el robot.
  Este problema fue resuelto mediante el empleo de únicamente *Strict Selectors*, que requieren una coincidencia exacta con los atributos del elemento de la interfaz de usuario. Esto incluye propiedades como el

ID, nombre, etc., que son ideales para entornos donde la estructura de la aplicación es estable y no cambia con frecuencia, ya que garantizan una alta precisión en la identificación de elementos.

- La idea inicial del proyecto era utilizar dos robots diferentes, siguiendo el modelo *Consumer-Dispatcher*, ambos con **REFramework** implementado. Sin embargo, resultaba en una solución mucho más compleja de lo necesario, ya que la manera más óptima y eficiente de llevar a cabo el propósito del robot resultó ser el tratar cada uno de los eventos como un elemento de un diccionario, en vez de como un *Item*, los cuales son extraídos por el *Consumer*, almacenados en una cola en **Orchestrator** y posteriormente procesados por el segundo robot *Dispatcher* en procesos más complejos y con conjuntos de datos más elavorados.

- Durante la ejecución del estado *Process state*, el robot compara el campo **EventID** de cada uno de los logs que componen el diccionario de logs anteriormente construido, con los ID de eventos más relevantes recogidos dentro del Excel *diccionarioEvents*. Para llevar esta comparación a cabo, la primera opción resultó en dos bucles anidados que iteraban dentro de ambos diccionarios, y, si encontraba coincidencia, almacenaba el log en cuestión. De esta manera, la eficacia y el consumo de recursos del robot, así como el tiempo de ejecución, daban un resultado mucho peor que el de la versión final.
  La forma de agilizar la ejecución y reducir los tiempos fue empleando una *Linq* (Ver Figura 3.14), que realiza todo el proceso descrito anteriormente de manera mucho más rápida, eficaz e implementado en una sola actividad.

Capítulo 5

Impactos sociales y ambientales

El proyecto desarrollado, centrado en el desarrollo de un robot RPA dedicado al análisis y procesamiento de logs en sistemas Windows, presenta diversos impactos a nivel social, ético y ambiental que resultan de relevancia, especialmente en el ámbito de la ciberseguridad.
Garantizar la salud y seguridad de los sistemas informáticos se ha convertido en un aspecto clave para proteger tanto a organizaciones como a usuarios particulares.
Desde una perspectiva social, el robot facilita la monitorización de la salud del sistema de los equipos, algo que puede ayudar significativamente a la detección temprana de incidentes y a la prevención de ciberataques. Esto es especialmente relevante en el contexto de los **Objetivos de Desarrollo Sostenible (ODS)** [29]. En particular, podemos hacer referencia a los siguientes **ODS**:

- **Educación de calidad (ODS 4):** Siendo útil para facilitar la enseñanza de conocimientos y conceptos relacionados con la ciberseguridad y el análisis forense de sistemas.
- **Trabajo decente y crecimiento económico (ODS 8):** Dado que la automatización de las tareas repetitivas y tediosas permite a los profesionales centrarse en actividades de mayor valor, mejorando la eficiencia y calidad del trabajo.
- **Industria, innovación e infraestructura (ODS 9):** Promoviendo soluciones tecnológicas que fortalezcan la seguridad y mantenimiento de la infraestructura digital.
- **Producción y consumo responsables (ODS 12):** Ya que el uso del RPA contribuye a una mejor gestión de los recursos digitales, reduciendo tiempos de ejecución y consumo energético.
- **Paz, justicia e instituciones sólidas (ODS 16):** Al contribuir en

la seguridad y salud de los sistemas, permite fortalecer la confianza en instituciones públicas y privadas.

- **Alianza para lograr objetivos (ODS 17):** Mediante la estandarización de herramientas como el robot propuesto en este proyecto, facilitando la colaboración entre diversos actores para el desarrollo de la seguridad de los sistemas informáticos.

Por otro lado, la solución RPA propuesta proporciona accesibilidad en el análisis de logs, comúnmente reservado a perfiles técnicos, permitiendo así a usuarios sin conocimientos especializados puedan mantener conocimiento y control de la salud y seguridad de sus sistemas.
Desde el punto de vista ético y profesional, el desarrollo del robot se ha realizado respetando los principios de transparencia, fiabilidad y utilidad, tratando de evitar en todo momento la recopilación innecesaria de información sensible de los usuarios. La automatización de tareas de supervisión debe aplicarse con responsabilidad, asegurando que no se utilice para la vigilancia masiva sin consentimiento ni para prácticas invasivas [30].
En cuanto al impacto ambiental, el uso de este robot RPA puede contribuir de manera indirecta a la sostenibilidad mediante la optimización de recursos. En conjunto con otras iniciativas de automatización, puede aportar un enfoque más amplio hacia una digitalización más eficiente y sostenible.

Capítulo 6

Conclusiones y trabajos futuros

Como resultado final de este proyecto, hemos conseguido diseñar una herramienta que es capaz de analizar los diferentes registros de eventos del sistema, facilitando su interpretación en comparación con el uso del visor de eventos tradicional. Este enfoque permite una gestión más eficiente y comprensible de los eventos del sistema, mejorando la capacidad de los administradores de sistemas y de los usuarios comunes para identificar y resolver problemas.
Los resultados obtenidos demuestran que el robot RPA es capaz de procesar grandes volúmenes de datos de manera rápida y precisa. Al filtrar y resumir los eventos, el robot reduce significativamente el tiempo necesario para analizar los registros, proporcionando descripciones claras y concisas que destacan los eventos más relevantes. Esto no solo optimiza el proceso de monitoreo del sistema, sino que también mejora la toma de decisiones al ofrecer información más accesible y fácil de entender para los usuarios.
En definitiva, la implementación de este robot RPA reduce la complejidad de extraer información de los logs del sistema Windows y facilita la identificación de anomalías de manera sencilla.

6.1. Líneas futuras

El diseño de este robot RPA tiene una gran escalabilidad. Actualmente, el robot puede descargar y analizar los logs del visor de eventos de Windows, resumiéndolos en un archivo Excel. Sin embargo, su capacidad puede ampliarse significativamente mediante la implementación de triggers en UiPath que ejecuten el robot a intervalos regulares. Esto permitiría la recopilación y análisis de logs de múltiples máquinas virtuales simultáneamente, proporcionando resúmenes periódicos de los logs de un conjunto de máquinas que estén ejecutando servicios críticos para una empresa, siendo enviados esos resúmenes de manera periódica y automática a una misma dirección de correo. Esta funcionalidad no

solo optimiza la eficiencia operativa, sino que también ofrece una solución robusta para la gestión de grandes volúmenes de datos en entornos empresariales, facilitando así su análisis para los administradores de sistemas.

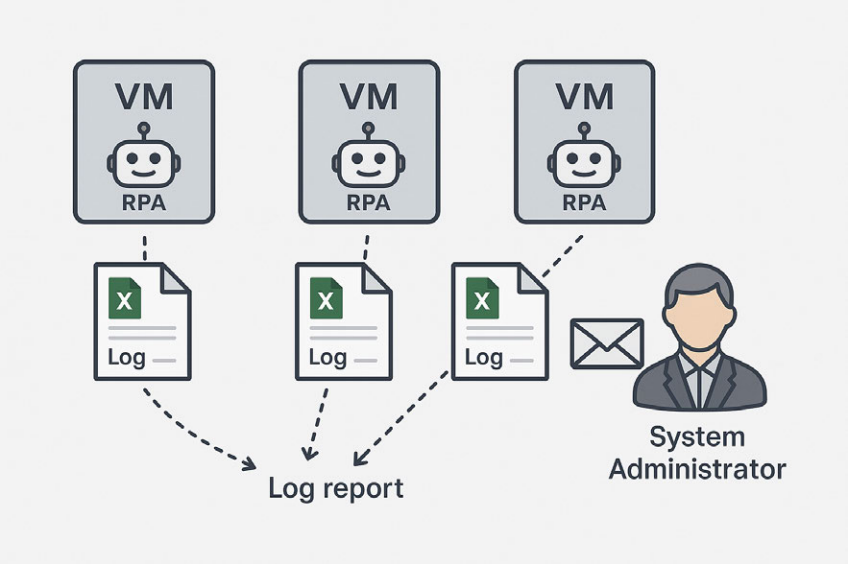

Figura 6.1: Reporte de resúmenes de logs programados en máquinas virtuales

Además, el proyecto puede evolucionar para incluir análisis predictivo y alertas proactivas. El robot podría identificar patrones y tendencias en los logs, anticipando posibles problemas antes de que ocurran. Esto permitiría a las empresas tomar medidas preventivas, mejorando la estabilidad y seguridad de sus sistemas. La capacidad de enviar alertas automáticas a los administradores de sistemas en caso de detección de eventos críticos es otra línea de desarrollo que podría aumentar significativamente el valor del robot.

Un aspecto destacado de este proyecto es su facilidad de uso para personas sin conocimientos técnicos. El diseño intuitivo del robot permite que cualquier usuario, independientemente de su experiencia en tecnología, pueda descargarlo y utilizarlo sin complicaciones. Además, la automatización de tareas complejas, como el análisis de logs y la generación de resúmenes en Excel, elimina la necesidad de intervención manual,

haciendo que el sistema sea ideal para pequeñas y medianas empresas que no cuentan con equipos técnicos especializados. La posibilidad de configurar *triggers* en UiPath para ejecutar el robot automáticamente también contribuye a su facilidad de uso, permitiendo a los usuarios recibir informes periódicos sin necesidad de supervisión constante.

Bibliografía

[1] Gartner, "Magic quadrant," 2025, accessed: 2025-02-24. [Online]. Available: https://www.gartner.es/es/metodologias/magic-quadrant#:~:text=Un%20Magic%20Quadrant%20de%20Gartner,posiciones%20relativas%20de%20sus%20competidores.

[2] UiPath, "Uipath academy," 2025, accessed: 2025-02-24. [Online]. Available: https://www.uipath.com/rpa/academy

[3] S. He, X. Zhang, P. He, Y. Xu, L. Li, Y. Kang, M. Ma, Y. Wei, Y. Dang, S. Rajmohan, and Q. Lin, "An empirical study of log analysis at microsoft," 2022, accessed: 2025-02-27. [Online]. Available: https://doi.org/10.1145/3540250.3558963

[4] UiPath, "Robotic process automation," 2025, accedido: 19-02-2025. [Online]. Available: https://www.uipath.com/rpa/robotic-process-automation

[5] C. M. Rivera Picado, L. Latorre, E. Rego, L. De Leo, and M. Gutierrez, "Reporte de tecnología: Rpa," 2024, accedido: 19-02-2025. [Online]. Available: https://publications.iadb.org/es/reporte-de-tecnologia-rpa

[6] ZAPTEST. (2023) Tecnología rpa: Repaso al pasado, presente y futuro. Accedido: 21-02-2025. [Online]. Available: https://www.zaptest.com/es/tecnologia-rpa-repaso-al-pasado-presente-y-futuro

[7] O. Doguc, "Robot process automation (rpa) and its future," *ResearchGate*, 2020, accedido: 21-02-2025. [Online]. Available: https://www.researchgate.net/publication/338302068_Robot_Process_Automation_RPA_and_Its_Future

[8] K. Insights. (2018) Audit quality and the historic potential of robotic process automation. Accedido: 21-02-2025. [Online]. Available: https://www.forbes.com/sites/insights-kpmg/2018/09/19/audit-quality-and-the-historic-potential-of-robotic-process-automation/

[9] Xerox, "Xerox smb white paper," 2021, accedido: 21-02-2025. [Online]. Available: https://s3.amazonaws.com/cms.ipressroom.com/89/files/20214/Xerox+SMB+White+Paper.pdf

[10] itSMF España. (2020) Inteligencia artificial y rpa. aclarando conceptos. Accedido: 21-02-2025. [Online]. Available: https://news.itsmf.es/inteligencia-artificial-y-rpa-aclarando-conceptos/

[11] O. Doguc, *Robot Process Automation (RPA) and Its Future*, 2020, accedido: 21-02-2025. [Online]. Available: https://www.researchgate.net/publication/338302068_Robot_Process_Automation_RPA_and_Its_Future

[12] S. B. Prism. (2025) Ss&c blue prism. Accedido: 21-02-2025. [Online]. Available: https://www.blueprism.com/es/

[13] ——. (2025) Recursos. Accedido: 21-02-2025. [Online]. Available: https://www.blueprism.com/es/resources/

[14] UiPath, "Automation platform," 2025, accessed: 2025-02-24. [Online]. Available: https://www.uipath.com/

[15] ——, "Attended vs unattended robots," 2022, accessed: 2025-02-24. [Online]. Available: https://docs.uipath.com/es/robot/standalone/2022.10/user-guide/attended-vs-unattended-robots

[16] ——, "Robotic enterprise framework," 2025, accessed: 2025-02-24. [Online]. Available: https://docs.uipath.com/es/studio/standalone/2023.10/user-guide/robotic-enterprise-framework

[17] J. Talebi, A. Dehghantanha, and R. Mahmoud, "Introducing and analysis of the windows 8 event log for forensic purposes," in *IWCF 2012 and 2014*. Springer International Publishing, 2015, pp. 145–162.

[18] Microsoft, "Event log file format," n.d., accessed: 2025-02-27. [Online]. Available: https://learn.microsoft.com/es-es/windows/win32/eventlog/event-log-file-format

[19] WeLiveSecurity, "5 fases fundamentales del análisis forense digital," 2023, accessed: 2025-03-15. [Online]. Available:

https://www.welivesecurity.com/es/recursos-herramientas/5-fases-fundamentales-del-analisis-forense-digital/

[20] Simplilearn, "Sha-256 algorithm: A complete guide," 2025, accedido: 21-03-2025. [Online]. Available: https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm

[21] ISO, "Iso/iec 27037:2012 - directrices para la identificación, recolección, adquisición y preservación de evidencia digital," 2023, accessed: 2025-03-15. [Online]. Available: https://www.iso.org/es/contents/data/standard/04/43/44381.html?browse=tc

[22] UNE, "Une 71505-1:2013 - tecnologías de la información (ti). sistema de gestión de evidencias electrónicas (sgee). parte 1: Vocabulario y principios generales," 2013, accessed: 2025-03-15. [Online]. Available: https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0051411

[23] Microsoft, "¿qué es siem?" 2025, accedido: 15-03-2025. [Online]. Available: https://www.microsoft.com/es-es/security/business/security-101/what-is-siem

[24] Splunk, "Splunk enterprise security," 2025, accedido: 21-03-2025. [Online]. Available: https://www.splunk.com/en_us/products/enterprise-security.html

[25] IBM, "Ibm qradar siem," 2025, accedido: 21-03-2025. [Online]. Available: https://www.ibm.com/products/qradar-siem

[26] LogRhythm, "Logrhythm siem documentation," 2025, accedido: 21-03-2025. [Online]. Available: https://docs.logrhythm.com/lrsiem/7.17.0/

[27] P. A. Networks, "What is extended security intelligence and automation management (xsiam)?" 2025, accedido: 21-03-2025. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-extended-security-intelligence-and-automation-management-xsiam

[28] Microsoft, "Linq en c#," 2025, accedido: 30-03-2025. [Online]. Available: https://learn.microsoft.com/es-es/dotnet/csharp/linq/

[29] G. de Costa Rica, "Objetivos de desarrollo sostenible - costa rica," 2025, accedido: 18-04-2025. [Online]. Available: https://www.ods.cr/

[30] I. O. for Standardization, "Iso/iec 27001:2013 - information security management systems — requirements," 2025, accedido: 18-04-2025. [Online]. Available: https://www.iso.org/contents/data/standard/05/45/54534.html

[31] V. M. Dominguez Rivas, *Plantilla TFG ETSISI UPM*. ETSISI, 2020.

[32] R. Cyber, "Comparison between uipath vs blue prism," 2025, accessed: 2025-02-21. [Online]. Available: https://www.royalcyber.com/blogs/rpa/comparison-between-uipath-vs-blue-prism/