

**IMPLEMENTACIÓN DE SOLUCIÓN DE SOFTWARE PARA LA
AUTOMATIZACIÓN DE CAPTURA DE EVIDENCIAS DURANTE
VENTANAS DE MANTENIMIENTO ON-PREMISE EN EL ENTORNO
SITESCOPE DE MICROFOCUS.**

JUAN DIEGO TORO PERILLA

**UNIVERSIDAD SANTO TOMAS
FACULTAD DE INGENIERÍA DE TELECOMUNICACIONES
BOGOTA
2023**

**IMPLEMENTACIÓN DE SOLUCIÓN DE SOFTWARE PARA LA
AUTOMATIZACIÓN DE CAPTURA DE EVIDENCIAS DURANTE
VENTANAS DE MANTENIMIENTO ON-PREMISE EN EL ENTORNO
SITESCOPE DE MICROFOCUS.**

Presentado Por:

**JUAN DIEGO TORO PERILLA
CÓDIGO: 2218979**

Proyecto de Pasantía para optar el Título de Ingeniero de telecomunicaciones

Dirigido por:

ING. JESUS AUGUSTO GUZMAN LOZANO

**UNIVERSIDAD SANTO TOMAS
FACULTAD DE INGENIERÍA DE TELECOMUNICACIONES
BOGOTA
2023**

Tabla de Contenidos

Capítulo 1	2
1.1. Introducción.....	2
1.2. Resumen	4
1.3. Abstract.....	5
1.4. Planteamiento y justificación del trabajo.....	6
1.4.1. Definición de problema.....	6
1.4.2. Justificación del trabajo	8
1.4.3. Funciones asignadas	11
1.5. Objetivos	12
1.5.1. Objetivo general:.....	12
1.5.2. Objetivos específicos:	13
1.6. Alcance.....	14
1.7. Metodología de la investigación.....	17
1.7.1. Investigación	17
1.7.2. Levantamiento de requerimientos	19
1.7.3. Diseño lógico de solución.....	21
1.7.4. Desarrollo de solución	24
Capítulo 2.....	26
2.1. Definición conceptos clave	27
2.1.1. Frontend.....	27
2.1.2. Backend.....	28
2.1.3. Monitoreo Sitescope.....	29
2.1.4. Ventanas de mantenimiento	30
2.1.5. OBM (Operations Bridge Manager)	31
2.2. Soluciones implementadas en monitoreo	32
Capítulo 3.....	35
3.1. Desarrollo de la pasantía	36
3.1.1. Proyectos	38
3.2. Implementación de solución software para automatizar toma de evidencias	43
3.2.1. Desarrollo scripts	43

3.2.2. Frontend interfaz gráfica	52
3.2.3. Backend solución consumo API	55
3.2.4. Notificación por email.....	56
3.3. Resultados	57
3.3.1. Archivo CSV.....	57
3.3.2. Archivo Excel	59
3.3.3. Capturas de pantalla PDF	61
Capítulo 4.....	71
Conocimientos adquiridos y aportes como pasante.....	72
Conclusiones	74
Recomendaciones	76
Bibliografía	77
Anexos.	79

Tabla de Figuras

Ilustración 1. Diagrama BPMN solución software. Autoría propia.....	22
Ilustración 2. Diseño técnico Ikono. Autoría BPS Consultores.....	38
Ilustración 3. Diseño técnico ADL. Autoría BPS Consultores.....	39
Ilustración 4. Diseño técnico Siebel. Autoría BPS Consultores.....	40
Ilustración 5. Diseño técnico APC. Autoría BPS Consultores.....	41
Ilustración 6. Diseño técnico Diebold. Autoría BPS Consultores.....	42
Ilustración 7. Diseño técnico SBL. Autoría BPS Consultores.....	42
Ilustración 8. Script login. Autoría propia.....	44
Ilustración 9. Script Interfaz_evidencias.py. Autoría propia.....	45
Ilustración 10. Schedule.py. Autoría propia.....	46
Ilustración 11. Sitescope_rest.py. Autoría propia.....	48
Ilustración 12. Auto_report.py. Autoría propia.....	49
Ilustración 13. Script send_email.....	51
Ilustración 14. Interfaz gráfica. Autoría propia.....	52
Ilustración 15. Diseño resumen evidencias. Autoría propia.....	53
Ilustración 16. Vista csv con información de la programación. Autoría propia.....	57
Ilustración 17. Archivo csv dentro del directorio db_evidencias_programadas. Autoría propia.....	58
Ilustración 18. Excel evidencia PRE y POST. Autoría propia.....	59

Ilustración 19. Evidencias guardadas en directorio. Autoría propia.	60
Ilustración 20. Capturas de pantalla en el directorio screens. Autoría propia.....	61
Ilustración 21. Reporte PDF en directorio. Autoría propia.	62
Ilustración 22. Login aplicación. Autoría propia.	63
Ilustración 23. Interfaz gráfica resultante.....	64
Ilustración 24. Comparación de aumento en productividad. Autoría propia.	65
Ilustración 25. Comparación eficiencia operativa. Autoría propia.	67
Ilustración 26. Comparación mejora de colecta de datos. Autoría propia.	68

PARTE I

Introducción a la Investigación

Capítulo 1

1.1. Introducción

En el mundo actual, donde la tecnología desempeña un papel fundamental en las operaciones empresariales, la automatización de tareas se ha vuelto esencial para mejorar la eficiencia y la productividad. En este contexto, el presente trabajo de monografía se centra en una implementación exitosa de automatización realizada para un cliente específico, con el objetivo de automatizar la toma de evidencias en ventanas de mantenimiento para su entorno de monitoreo on-premise.

El monitoreo de sistemas y aplicaciones en tiempo real es crucial para garantizar un rendimiento óptimo y prevenir posibles fallas que podrían impactar negativamente en el negocio [1]. En este sentido, el cliente se enfrentaba al desafío de gestionar y coordinar periódicamente la captura de evidencias antes y después de las ventanas de mantenimiento, necesarias para realizar actualizaciones y cambios en su infraestructura de monitoreo on-premise [2]. Anteriormente, este proceso de captura

de evidencias en ventanas de mantenimiento se realizaba de forma manual, lo que implicaba una cantidad de tiempo y esfuerzo significativas por parte del equipo de operaciones del cliente. Además, existía un riesgo inherente de errores humanos en cuanto a la ejecución de las tareas programadas como se describe en el capítulo 1.4. Planteamiento y justificación del trabajo.

Con el fin de superar estos desafíos, se implementó una solución de automatización personalizada. Esta solución permite la programación automatizada de toma de evidencias, optimizando los recursos y minimizando los riesgos asociados con la intervención manual. En este documento de monografía, se describirá en detalle el desarrollo e implementación de la automatización de toma de evidencias en entornos SiteScope mediante una interfaz desarrollada en Python, abordando los aspectos técnicos y operativos, así como los beneficios obtenidos a través de esta automatización.

1.2. Resumen

Durante la pasantía, existió la oportunidad de participar en el desarrollo e implementación de una solución de software, para la automatización de toma de evidencias antes y después de las ventanas de mantenimiento on-premise en el entorno SiteScope de Microfocus.

El objetivo de este proyecto fue desarrollar una solución que permitiera automatizar la toma de evidencias del estado de los ambientes SiteScope, antes y después del proceso de ejecución de ventanas de mantenimiento en los sistemas on-premise del cliente. Estas ventanas de mantenimiento como se discute en 2.1.4. Ventanas de mantenimiento, son períodos de tiempo predefinidos en los que se realizan tareas de mantenimiento y actualizaciones en los sistemas sin afectar la disponibilidad del servicio. Al finalizar las ventanas, era necesario tomar las evidencias manualmente para verificar el resultado de su ejecución.

En colaboración con el equipo de proyectos de la empresa BPS Consultores, se generó el diseño de la arquitectura de solución (véase Ilustración 1 para más detalles). Esto incluyó la definición de los requisitos funcionales y no funcionales, así como la identificación de los componentes clave del sistema. Además, se trabajó en la creación de una interfaz de usuario usable, que permitiera al cliente programar y gestionar la toma de evidencias en las ventanas de mantenimiento de manera eficiente [2].

Una vez completado el diseño, se procede a la etapa de desarrollo de la solución usando tecnologías como Python y Microfocus SiteScope para crear las funcionalidades necesarias. Se da la funcionalidad de programar toma de evidencias, enviar notificaciones a los usuarios involucrados y llevar un registro detallado del estado de los ambientes SiteScope antes y después de cada ventana (consulte sección 3.3. Resultados para más detalles).

Finalmente, tras completar un exhaustivo proceso de pruebas, se asegura la correcta funcionalidad de la solución para garantizar su despliegue en el entorno operativo (consulte sección 3.3. Resultados para más detalles).

1.3. Abstract

During the internship, there was the opportunity to participate in the development and implementation of a software solution for the automation of evidence collection before and after on-premise maintenance windows in the microfocus SiteScope platform environment.

the objective of this project was to develop a solution that would automate the taking of evidence of the status of SiteScope environments, before and after the process of executing maintenance windows in the client's on-premise systems. These maintenance windows, as discussed in 2.1.4. Ventanas de mantenimiento, are predefined periods of time in which maintenance tasks and updates are performed on the systems without affecting the availability of the service. At the end of the windows, it was necessary to take the evidence manually to verify the result of its execution.

In collaboration with the project team, at the company BPS Consultores, the design of the solution architecture was generated (see Ilustración 1 for more details). This included the definition of the functional and non-functional requirements, as well as the identification of the key components of the system. in addition, work was done on the creation of a usable user interface, which would allow the client to schedule and manage the collection of evidence in maintenance windows efficiently [2].

Once the design is complete, we proceed to the development stage of the solution using technologies such as python and microfocus SiteScope to create the necessary functionalities. It provides the functionality of scheduling evidence taking, sending notifications to the users involved and keeping a detailed record of the status of the SiteScope environments before and after each window.

Finally, after completing an exhaustive testing process, the correct functionality of the solution is ensured to guarantee its deployment in the operating environment (consult section 3.3. Resultados for more details).

1.4. Planteamiento y justificación del trabajo

1.4.1. Definición de problema

Este proyecto surge a partir de la necesidad de automatizar la gestión de toma de evidencias en el proceso de ventanas de mantenimiento en un entorno de monitoreo on-premise. De este modo, la coordinación y ejecución de toma de evidencias de manera manual implicaba que el equipo de producción asignara tiempo y esfuerzo adicional para documentar el estado de cada monitor antes y después de las ventanas de mantenimiento, y verificar que todo volviera a funcionar correctamente [2]. Considerando lo anterior a fin de orientar la investigación, se plantean las siguientes interrogantes: ¿Cuáles serán las tecnologías empleadas en la implementación de la automatización? (véase 1.7.1. Investigación), ¿Cuáles son los requisitos esenciales para la construcción de esta solución? (véase 1.7.2. Levantamiento de requerimientos), ¿De qué manera se llevará a cabo la automatización del proceso? (véase 1.7.3. Diseño lógico de solución), y finalmente, ¿cómo se desarrollará la solución? (véase 1.7.4. Desarrollo de solución). Estas interrogantes se resolverán a medida que se avance el documento.

En este contexto, la automatización surge como una solución eficiente y efectiva para mejorar la gestión de evidencias del monitoreo en SiteScope (véase sección 3.3.2. Archivo Excel). La implementación de esta solución automatizada permite programar de manera precisa y confiable esta toma de evidencias, mejorar la verificación del funcionamiento y documentación tanto antes como después de la ventana de

mantenimiento del monitoreo, además de minimizar los riesgos asociados con la intervención manual ().

1.4.2. Justificación del trabajo

Ante la necesidad de automatizar el proceso de evidencias en las ventanas de mantenimiento, se realiza la automatización que busca proporcionar precisión en la información y mejorar el proceso de evidencias del monitoreo on-premise. A través de este documento, se pretende proporcionar una guía práctica de cómo abordar y superar los desafíos relacionados con la gestión de evidencias en ventanas de mantenimiento en ambientes SiteScope de MicroFocus.

Dado el valor que este documento puede aportar, sus beneficios se dirigen no solo al cliente que se beneficia de la automatización implementada, sino también a otras organizaciones que buscan mejorar sus operaciones de monitoreo, como se describe más adelante. Al compartir los detalles técnicos, los enfoques metodológicos y los resultados obtenidos, se espera inspirar y guiar a otros ingenieros y equipos de proyectos a adoptar soluciones automatizadas para mejorar la gestión de evidencias de ventanas de mantenimiento en sus respectivos entornos de monitoreo.

Por otra parte, La automatización de este proceso conlleva repercusiones multidimensionales que pueden afectar a la sociedad, la empresa y el cliente en aspectos económicos, sociales, ambientales y legales. Económicamente, en primer lugar, la automatización puede reducir los costos operativos al eliminar la necesidad de personal dedicado a tareas en ventanas de mantenimiento ejecutadas en monitoreos como la recopilación de evidencias (véase 3.3.7. Mejora de datos para más detalles). En segundo lugar, al automatizar procesos, la empresa puede aumentar su eficiencia operativa, lo que se traduce en una mayor productividad (consulte) y, potencialmente,

en mayores ingresos [3]. En tercer lugar, al liberar al personal de tareas repetitivas, la automatización permite que los recursos humanos se centren en actividades más estratégicas y de mayor valor [4]. Finalmente, la automatización por software de la toma de evidencias en entornos SiteScope puede tener un impacto económico significativo para la empresa desarrolladora. Esto se refleja en la posibilidad de generar ingresos adicionales y mantener contratos a largo plazo con los clientes [2]. Además, la satisfacción del cliente con la automatización puede abrir oportunidades para ofrecer servicios de mantenimiento y abordar futuras automatizaciones, lo que contribuye a la estabilidad financiera de la empresa [2].

En relación al impacto ambiental, en primer lugar, al mejorar procesos y reducir la intervención humana, la automatización puede llevar a una gestión más eficiente de los recursos, lo que a su vez podría reducir el consumo energético en los centros de datos o servidores on-premise [5]. Finalmente, la implementación de prácticas de automatización puede incluir políticas de reciclaje y reutilización de componentes electrónicos obsoletos, promoviendo así una gestión más sostenible de los residuos electrónicos [6].

Por lo que se refiere al impacto social, si bien la automatización puede reducir la necesidad de personal para tareas manuales específicas, también puede generar empleo en tecnología, ya que se requieren profesionales para desarrollar, implementar y mantener sistemas automatizados [7]. Por otra parte, la disponibilidad de datos precisos y en tiempo real a través de la automatización puede ayudar a la empresa a tomar decisiones más informadas sobre la infraestructura, lo que puede resultar en una mejor gestión de recursos y una mejora en la calidad del trabajo [8].

En cuanto al impacto legal, inicialmente, La automatización puede llevar a errores o malentendidos en la recopilación de evidencias. Esto podría dar lugar a disputas legales entre la empresa y sus clientes o entre la empresa y terceros. Es importante contar con políticas claras de responsabilidad y documentación adecuada para tener bajo control cualquier imprevisto [9]. De igual forma, al ser software personalizado en la automatización, es esencial asegurarse de que se cumplan los derechos de propiedad intelectual y que no se infrinja ningún derecho de autor o patente [9].

1.4.3. Funciones asignadas

Durante la pasantía como Ingeniero de Telecomunicaciones, hubo la responsabilidad de trabajar en la configuración del monitoreo on-premise y nube para los clientes, utilizando las herramientas de Microfocus, específicamente SiteScope y OBM (consulte 1.7.2. Levantamiento de requerimientos).

En el contexto del monitoreo, el trabajo consistió en configurar el entorno de monitoreo en SiteScope, el cual estaba estrechamente relacionado con la infraestructura on-premise del cliente [2]. Se utilizó SiteScope para establecer umbrales y configurar las métricas de monitoreo de acuerdo con el rendimiento y funcionamiento de dicha infraestructura. Esto implica definir parámetros críticos y establecer alertas cuando se superaran ciertos umbrales (véase 3.1.1. Proyectos).

Una vez configurado el monitoreo en SiteScope, el siguiente paso fue trabajar en OBM (Operations Bridge Manager), también de Microfocus [10]. En OBM, se configuraron las políticas de monitoreo para recibir las alertas generadas por SiteScope (on-premise) y Cloud. Estas políticas permitían visualizar de manera clara el motivo de la alerta y definir las acciones que debería tomar el personal encargado para abordar la situación [10].

Durante la pasantía, estas fueron las principales funciones relacionadas con la configuración del monitoreo utilizando las herramientas de Microfocus. A través de este trabajo, el pasante adquirió experiencia en la implementación y configuración de soluciones de monitoreo para ayudar a los clientes a mejorar el control, rendimiento y mantenimiento de su infraestructura Cloud y On-premise.

1.5. Objetivos

1.5.1. Objetivo general:

Implementar una solución de software para automatizar la toma de evidencias antes y después de las ventanas de mantenimiento de monitoreo on-premise, utilizando principalmente Python y SiteScope-Microfocus.

1.5.2. Objetivos específicos:

1. Desarrollar scripts en Python que permitan seleccionar monitores del entorno SiteScope, programar y tomar evidencias automáticamente y llevar un registro sistemático de las evidencias programadas.
2. Elaborar una interfaz gráfica en Python destinada a la gestión de los scripts implementados y programación de captura de evidencias, que se caracterice por la usabilidad para el cliente.
3. Crear una funcionalidad que emplee la API pública de SiteScope. Para capturar evidencias textuales en formato Excel, acompañadas por la captura de screenshots en formato pdf del monitoreo en entornos SiteScope, a fin de ser enviadas automáticamente finalizada la ventana de mantenimiento al correo corporativo.

1.6. Alcance

El alcance del proyecto consiste en desarrollar una solución de automatización para la toma de evidencias antes y después de las ventanas de mantenimiento en el entorno de monitoreo SiteScope [11]. La solución se implementará como un complemento que se integra con las ventanas de mantenimiento, lo que implica ciertas limitaciones en cuanto a funciones y acceso.

La funcionalidad principal del proyecto es permitir la captura automatizada de evidencias antes y después de cada ventana de mantenimiento [2]. Para lograr esto, se crearán scripts que cumplan con los objetivos específicos mencionados en la sección 1.3.

El alcance del proyecto incluye:

1. Interfaz Gráfica: Se desarrollará una interfaz gráfica de fácil uso que permita al personal programar la toma de evidencias para cada monitor durante la ventana de mantenimiento (consulte sección 2.1.1. Frontend para más detalles). Esta interfaz cumple con:
 - Programar y llevar registro de las evidencias que se van tomando en cada ventana de mantenimiento.
 - No requerir instalación.
2. Toma de Evidencias: Se implementará la funcionalidad para capturar evidencias antes y después de cada ventana de mantenimiento (véase sección 2.1.2. Backend para más detalles). Estas evidencias serán tomadas

de los monitores gestionados en el entorno seleccionado. Para este propósito se implementará la siguiente funcionalidad:

- Seleccionar los hosts de cada monitoreo SiteScope y los monitores que serán incluidos en las ventanas de mantenimiento.
 - Capturar antes y después de cada ventana de mantenimiento para todos los monitores en los entornos SiteScope correspondientes.
 - Usar el programador de tareas de Windows, de manera que la toma de evidencias se ejecute según la programación establecida.
3. Notificación: Se enviará un correo electrónico al personal autorizado para informar el estado del monitoreo en la ventana de mantenimiento. El correo incluirá las evidencias adjuntas para su revisión (consulte sección 3.2.4. Notificación por email para más información).

El proyecto se enfocará en la funcionalidad mencionada anteriormente y no incluirá lo siguiente dentro de su alcance:

1. Programación de Activación/Desactivación de Monitores: No se incluirá la funcionalidad para programar la activación o desactivación de los monitores durante las ventanas de mantenimiento.
2. Configuraciones Adicionales: No se implementarán configuraciones adicionales más allá de la programación de la toma de evidencias.

En resumen, el proyecto se centrará en proporcionar una solución eficiente y efectiva para la toma de evidencias automatizada antes y después de las ventanas de mantenimiento, mejorando así la gestión del monitoreo y asegurando la detección temprana de posibles problemas. La solución final permitirá a los usuarios contar con la información necesaria para verificar el estado del monitoreo y así tomar acciones correctivas de manera oportuna, además, mejora sustancialmente la productividad del equipo de operaciones (consulte 3.3.5. Productividad para más detalles).

1.7. Metodología de la investigación

1.7.1. Investigación

Se empleó una metodología de investigación aplicada que culminó en la automatización de toma de evidencias en ventanas de mantenimiento. El enfoque de la investigación se orientó hacia la búsqueda de una solución eficiente de software para la automatización. Durante la fase exploratoria, se procedió a un análisis detallado del problema relacionado con la captura de evidencias que afectaba la productividad de la operación. Posteriormente, se definieron los objetivos de la investigación con el propósito de desarrollar scripts e integrarlos con herramientas para gestionar datos como Excel y pdf, que cumplieran con los requisitos establecidos en el alcance y objetivos específicos (consulte 1.5. Objetivos para más detalle). Como resultado, las herramientas seleccionadas fueron:

1. Python

Lenguaje de programación que permite crear los scripts [12] encargados en automatizar la toma de evidencias de monitores en entornos SiteScope.

2. SiteScope

Herramienta de monitoreo de infraestructura y aplicaciones que ayuda a los administradores de sistemas a supervisar y gestionar de forma proactiva el rendimiento y la disponibilidad de sistemas [11].

3. API público SiteScope

Interfaz de programación de aplicaciones diseñada para permitir la interacción y la automatización de las funcionalidades de SiteScope. Proporciona una forma flexible y potente de interactuar con SiteScope y aprovechar sus capacidades de monitoreo y gestión de forma automatizada y personalizada, facilitando la administración eficiente y efectiva de los sistemas y aplicaciones supervisados [11].

4. Excel

Herramienta Office usada para el manejo de la información. Se usa su capacidad para importar y exportar csv, tratando la información separada por comas, recibida por la interfaz, como una hoja de cálculo con columnas y filas. Adicionalmente, las hojas de cálculo se utilizan como un repositorio de datos en el cual el equipo de producción verifica modificaciones en el estado de los monitores [13].

5. TKinter

Biblioteca gráfica Tci/Tk diseñada para Python. Es la biblioteca usada para la creación de la interfaz gráfica [12].

6. Pdf

Es un estándar de archivo que permite presentar y compartir documentos de manera precisa e independiente del software y hardware utilizados para su creación o visualización. Herramienta utilizada para almacenar las capturas de pantalla.

1.7.2. Levantamiento de requerimientos

Se llevo a cabo una sesión con el cliente para definir los requisitos y expectativas de la automatización [2]. Además, se realizó un análisis del inventario en producción, el cual proporcionó la información necesaria para configurar y desarrollar los scripts de automatización de la toma de evidencia en las ventanas de mantenimiento (consulte sección 3.2. Implementación de solución software para automatizar toma de evidencias). Este proceso identificó los siguientes aspectos clave para el desarrollo del sistema de automatización de la toma de evidencias:

1. Diseñar e implementar una interfaz gráfica que permita a los usuarios programar la ejecución de la toma de evidencias de manera eficiente.
2. Desarrollar scripts que automaticen la toma de evidencias antes y después de la hora de inicio y fin respectivamente, de la ventana de mantenimiento, garantizando que se capturen los datos necesarios para su análisis.
3. Implementar una función para generar un reporte en formato Excel que contenga el estado de los monitores y otra información relevante, brindando una visión detallada del monitoreo.
4. Crear una funcionalidad para generar un reporte en formato PDF con pantallazos de todos los monitores funcionando, que permita una revisión visual rápida y precisa del estado del monitoreo.

5. Integrar notificación vía correo electrónico, que envíe de manera automática los reportes generados en los archivos Excel y PDF a los destinatarios asignados, asegurando una entrega oportuna de la información recopilada durante las ventanas de mantenimiento.

1.7.3. Diseño lógico de solución

Tras la definición de los requisitos del script y las herramientas, se realiza el diseño lógico de la solución, enfocado en identificar las actividades y aspectos técnicos específicos en el contexto de Python para lograr el objetivo establecido. Se debe aplicar conocimientos en desarrollo backend y frontend para desarrollar el script y se planea el desarrollo por métodos que separen cada proceso dividido en 4 diferentes scripts para mayor control del mismo (consulte sección 3.2. Implementación de solución software para automatizar toma de evidencias). Además, se elaboró un diagrama BPMN que representa de manera clara el funcionamiento integral de la solución de software para la automatización (véase Ilustración 1). Este diagrama visual permite identificar los eventos iniciales, tareas intermedias, decisiones clave y eventos finales involucrados en el proceso, asegurando una secuencia lógica y coherente para la automatización [14].

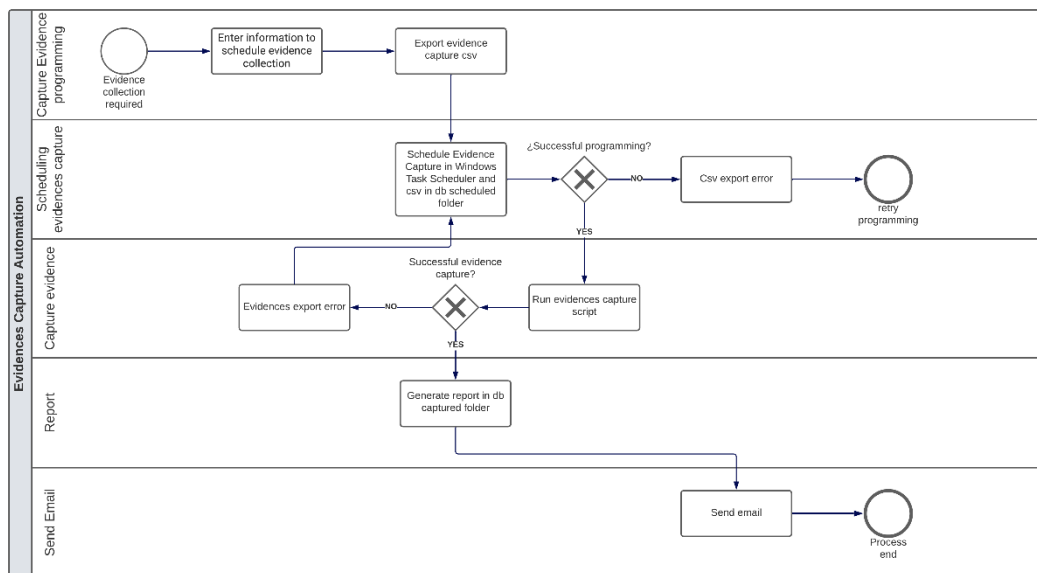


Ilustración 1. Diagrama BPMN solución software. Autoría propia.

La ilustración 1 tiene como propósito la elucidación del procedimiento que se lleva a cabo para la automatización de la captura de evidencias. El esquema contempla la presencia de cinco agrupaciones denominadas "pools", las cuales representan los diferentes pasos necesarios para la ejecución de la mencionada automatización.

El primer paso conlleva la programación de la obtención de evidencias por parte del usuario, quien debe llevar a cabo esta acción a través de la interfaz gráfica. El resultado de este proceso se traduce en la generación de un archivo en formato CSV que almacena la información requerida. Este paso inicial establece una conexión con el segundo paso, que consiste en la exitosa incorporación de la programación de la captura de evidencias en el programador de tareas de Windows y el subsiguiente almacenamiento del archivo CSV en la carpeta " db_evidencias_programadas" la cual se encuentra localizada en el directorio principal de la automatización. Finalmente, el programador de tareas de Windows, se encarga de activar el script que captura las

evidencias. En caso de que el segundo paso no sea ejecutado exitosamente, se hace necesario retomar la programación a través de la interfaz gráfica.

La activación del script responsable de la obtención de evidencias, de acuerdo al horario predeterminado, da inicio al tercer paso, que se encarga de conectarse al entorno SiteScope y capturar las evidencias. En caso de que la programación no haya sido efectuada con precisión, se requiere una nueva programación a través de la interfaz gráfica. La captura de evidencias propicia el inicio del cuarto paso, que se encarga de generar un archivo Excel, junto con un archivo PDF, ambos destinados a la carpeta titulada "db_evidencias_capturadas". Estos archivos contienen información relativa al monitoreo, así como capturas de pantalla del entorno SiteScope. En el evento de que la obtención de evidencias no sea posible, se procederá a documentar el error correspondiente, y se tornará necesario revisar la programación de la captura de evidencias.

Una vez que las evidencias antes y después de la ventana de mantenimiento hayan sido capturadas, se desencadena el quinto y último paso, que se enfoca en la transmisión de dichas capturas y la descripción incluida en la programación, a través del correo electrónico hacia una cuenta corporativa previamente configurada (véase 2.1.2. Backend para más detalles).

1.7.4. Desarrollo de solución

Durante la etapa de desarrollo de los scripts, se priorizo la legibilidad y comprensión del código por parte del cliente y su equipo de producción. Por ende, se incluyó una documentación que detalla cada proceso realizado [2]. La documentación proporciona entre otras cosas, instrucciones detalladas sobre cómo consumir las APIs que son fundamentales para el funcionamiento del núcleo del script. Además, se gestionó la creación de un usuario específico en el entorno de producción de SiteScope, con permisos adecuados para llevar a cabo las tareas requeridas.

Adicionalmente en la etapa de desarrollo, se exploró la opción de crear una interfaz basada en un formulario, donde el equipo de producción pudiera completar campos específicos para identificar los monitores o grupos de monitores de un host en particular que debían ser tomados para evidencia en un determinado tiempo. Se desarrollo inicialmente una versión con una estructura jerárquica, pero posteriormente se optó por desarrollar una interfaz con este enfoque, con el fin de facilitar la programación de la captura de evidencias [2].

PARTE II

Estado del Arte

Capítulo 2

Estado del Arte

En la última década, el monitoreo de sistemas y redes ha experimentado un crecimiento significativo, impulsado por la necesidad de mantener la estabilidad y rendimiento óptimo de infraestructuras tecnológicas complejas. En este contexto, se han desarrollado diversos proyectos que buscan automatizar y mejorar el funcionamiento del monitoreo a través de soluciones innovadoras [15]. Estas iniciativas se enfocan en mejorar la recolección y análisis de datos, implementar alertas proactivas y facilitar la generación de informes precisos y detallado. En esta revisión del estado del arte, se explorarán algunos avances y logros alcanzados por estos proyectos, así como su impacto en la eficiencia operativa y la toma de decisiones en entornos tecnológicos [15].

2.1. Definición conceptos clave

2.1.1. Frontend

Frontend se refiere a la parte visible y accesible de una aplicación con la que interactúan directamente. Es la interfaz gráfica y funcionalidades que se presentan en el dispositivo del cliente. El desarrollo frontend implica el uso de tecnologías como TKinter de Python para estructurar el contenido, diseñar el aspecto visual y brindar la interactividad necesaria. Los desarrollos frontend se centran en crear una experiencia de usuario intuitiva y atractiva, asegurando la usabilidad y la adaptabilidad a diferentes dispositivos. Además, se ocupan de mejorar el rendimiento y la velocidad para ofrecer una navegación fluida y eficiente [16].

2.1.2. Backend

Backend se refiere a la parte del sistema informático que no es visible para el usuario final, pero que es esencial para el funcionamiento de una aplicación. Es la parte encargada de procesar y almacenar datos, así como de gestionar la lógica y la lógica de negocio. Los desarrolladores backend trabajan con lenguajes de programación como Python, Java, PHP o Node.js para crear y mantener los servidores, las bases de datos y las API que permiten la comunicación entre el frontend y el servidor. También se encargan de la seguridad y el rendimiento del sistema, asegurando que la aplicación funcione de manera eficiente y segura [17].

2.1.3. Monitoreo SiteScope

SiteScope es una herramienta de monitoreo on-premise utilizada para supervisar y controlar el rendimiento de aplicaciones, servidores y dispositivos en entornos locales. Con SiteScope, las organizaciones pueden obtener información en tiempo real sobre la disponibilidad y el rendimiento de sus sistemas críticos, lo que les permite detectar y resolver problemas rápidamente. La solución ofrece una amplia gama de métricas y alertas personalizables, lo que permite a los equipos de operaciones y TI mantener un control total sobre la infraestructura. SiteScope es altamente configurable y compatible con diversos tipos de tecnologías y plataformas, lo que lo convierte en una herramienta versátil para la monitorización exhaustiva de entornos empresariales complejos [11].

2.1.4. Ventanas de mantenimiento

Las ventanas de mantenimiento en el monitoreo on-premise son periodos programados en los que se realiza mantenimiento, actualizaciones o tareas de configuración en los sistemas y servicios. Durante estas ventanas, se pueden interrumpir temporalmente algunas funciones del monitoreo para evitar falsas alarmas o afectaciones en los registros [18].

2.1.5. OBM (Operations Bridge Manager)

Micro Focus Operations Bridge Manager (OBM) es una solución integral de administración y monitoreo en tiempo real que consolida datos operativos, detecta alertas y permite respuestas ágiles tanto en entornos de nube mediante políticas personalizadas como en sistemas on-premise, incluido SiteScope. Su función principal es brindar una visión completa del estado de sistemas y aplicaciones para mejorar la eficiencia operativa y garantizar un rendimiento óptimo [10].

2.2. Soluciones implementadas en monitoreo

Existen diversas propuestas e implementaciones para monitoreo, pero este proyecto se centrará específicamente en algunas soluciones que busquen automatizar la captura de información. Se abordarán las siguientes soluciones con el objetivo de evaluar su viabilidad y efectividad en la automatización del proceso de recolección de datos.

“Plataforma que integra los monitoreos del área de seguridad de la información y ciberseguridad de una compañía vigilada por la superintendencia financiera de Colombia” Carla Vanessa Sayago [19].

El propósito de este proyecto fue desarrollar una plataforma local en Python para integrar 10 monitoreos de seguridad de la Información en el área de Riesgos de una compañía supervisada por la SFC, siguiendo las directrices de la norma ISO/IEC 27001:2013. El alcance abarca la creación de una interfaz gráfica intuitiva en Python que permitirá al usuario seleccionar y ejecutar los monitoreos deseados, garantizando la protección de la integridad, disponibilidad y confidencialidad de los datos. Además, la creación de diez interfaces gráficas individuales, una para cada monitoreo, donde se solicitan documentación específica, monitoreo y se brinda la opción de enviar los resultados al coordinador de seguridad de la Información y ciberseguridad a través de correos electrónicos [19]. Además, se desarrolló diez códigos fuente en R para recibir y procesar la información proporcionada por Python.

“Diseño e implementación back-end de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y medianas empresas” Jesús David Arias [20].

El propósito intrínseco de esta investigación radicó en el desarrollo de una solución de software altamente innovadora, destinada a empoderar a las MiPymes mediante una efectiva recolección y gestión de información que resulte propicia para una toma de decisiones informada. El sistema desarrollado ofrece a los usuarios la posibilidad de establecer conexiones con bases de datos pertinentes, lo cual les faculta para llevar a cabo consultas altamente personalizadas y la posterior visualización de datos en dashboards especialmente diseñados para un análisis efectivo. Un rasgo distintivo y relevante de esta solución es su capacidad para operar en entornos Cloud, lo que provee una dimensión adicional de flexibilidad y eficiencia a su implementación. En última instancia, el propósito principal es que las Micro, Pequeñas y Medianas Empresas (MiPymes) puedan aprovechar plenamente la riqueza de su información, con el objetivo de proyectar una planificación sólida que fomente un crecimiento sostenible y exitoso [20].

PARTE III

Pasantía e implementación de solución

Capítulo 3

Implementación, solución y resultados

En el presente capítulo se expondrá detalladamente el desarrollo e implementación de la automatización de captura de evidencias en ventanas de mantenimiento, así como los proyectos entregados en base a las funciones asignadas en el periodo de pasantía.

3.1. Desarrollo de la pasantía

Durante el periodo de pasantía en la empresa BPS Consultores, el pasante tuvo la valiosa oportunidad de asumir una serie de funciones que fueron detalladas en el apartado Levantamiento de requerimientos. En este sentido, la participación también se extendió hacia la sección de proyectos de monitoreo on-premise y Cloud, empleando las herramientas SiteScope y OBM de Microfocus.

A lo largo de los 6 meses de pasantía, el trabajo se centró en la configuración y optimización del entorno SiteScope, enfocado en el monitoreo de infraestructuras on-premise. Esta tarea implicó la meticulosa configuración de métricas y umbrales en los monitores para garantizar la detección temprana y precisa de situaciones relevantes para nuestros clientes [2]. La configuración y afinación precisa de estos monitores permite alertar oportunamente sobre cualquier evento crítico que pueda afectar la integridad o el rendimiento de la infraestructura on-premise de los clientes. Asimismo, en el entorno OBM, el pasante fue responsable de la creación y configuración de dashboards personalizados que desempeñan un papel crucial en la visualización clara y concisa de los eventos provenientes de SiteScope. Adicionalmente, fue responsabilidad la creación de políticas que reciben los eventos nube para la visualización en los dashboards [10]. Estos dashboards permiten una supervisión proactiva y una toma de decisiones fundamentada (véase 3.3.6. Eficiencia operativa).

Durante la participación en estos proyectos de monitoreo on-premise y Cloud con SiteScope y OBM de Microfocus, el pasante pudo adquirir una sólida experiencia en la

configuración de políticas y alertas, lo que garantizó el óptimo desarrollo del proyecto y posteriores (véase Ilustración 6, Ilustración 7).

La pasantía en BPS Consultores brindó una invaluable oportunidad para desarrollar y mejorar las habilidades técnicas, así como para comprender la importancia y el impacto del monitoreo efectivo en el entorno empresarial. Durante este período, el pasante pudo contribuir significativamente al éxito y rendimiento del equipo de proyectos de monitoreo (más detalles a continuación), quedando agradecido por la experiencia que obtuvo y el conocimiento que adquirió en el campo del monitoreo on-premise y Cloud con SiteScope y OBM de Microfocus.

3.1.1. Proyectos

- Monitoreo Cloud IKONO:

En este proyecto se implementó la política en OBM, la cual se encarga de la recepción de los eventos problema y solución de una nube Oracle. Durante el proyecto se llevaron a cabo las tareas de:

- Creación de política para parseo de eventos.
- Creación de filtros de visualización.
- Creación de página de visualización.

La realización de las demás tareas como diseño e implementación de arquitectura y árbol de servicio, fue desarrollada por otra parte del equipo de proyectos [2].

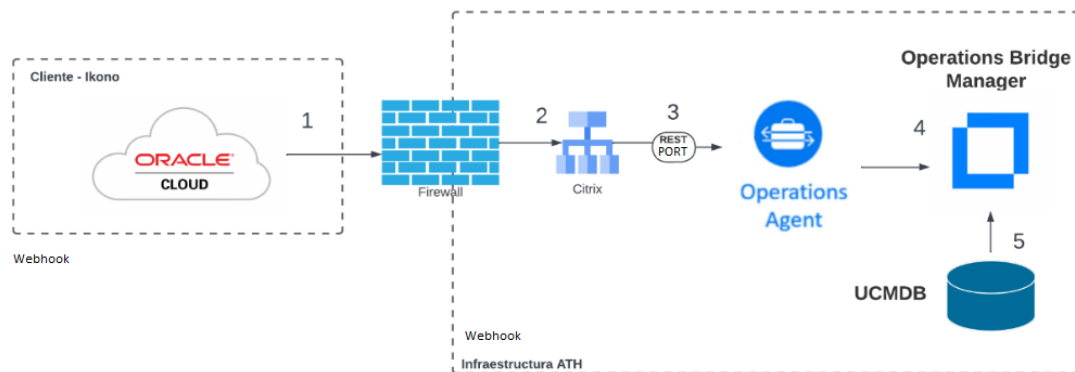


Ilustración 2. Diseño técnico Ikono. Autoría BPS Consultores.

- Monitoreo Cloud ADL:

En este proyecto se implementó la política en OBM, la cual se encarga de la recepción de los eventos problema y solución de una nube NewRelic. Durante el proyecto se llevaron a cabo las tareas de:

- Creación de política para parseo de eventos.
- Creación de filtros de visualización.
- Creación de página de visualización.

La realización de las demás tareas como diseño e implementación de arquitectura y árbol de servicio, fue desarrollada por otra parte del equipo de proyectos [2].

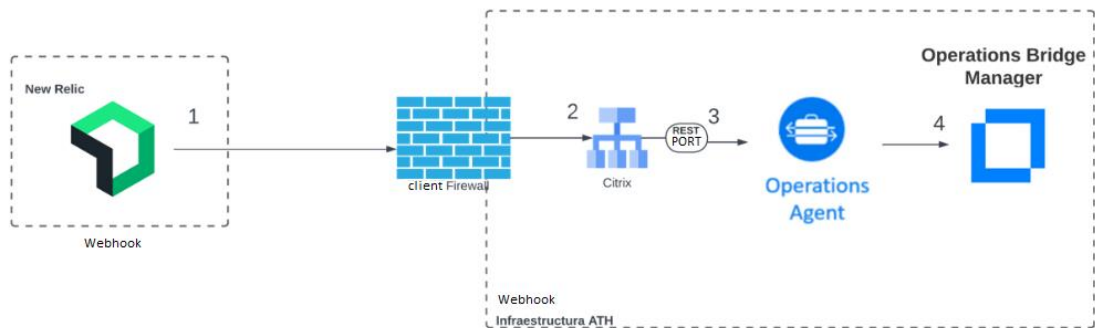


Ilustración 3. Diseño técnico ADL. Autoría BPS Consultores.

- Monitoreo Cloud SIEBEL:

En este proyecto se implementó la política en OBM, la cual se encarga de la recepción de los eventos problema y solución de una nube Oracle. Durante el proyecto se llevaron a cabo las tareas de:

- Creación de política para parseo de eventos.
- Creación de filtros de visualización.

- Creación de página de visualización.

La realización de las demás tareas como diseño e implementación de arquitectura y árbol de servicio, fue desarrollada por otra parte del equipo de proyectos [2].

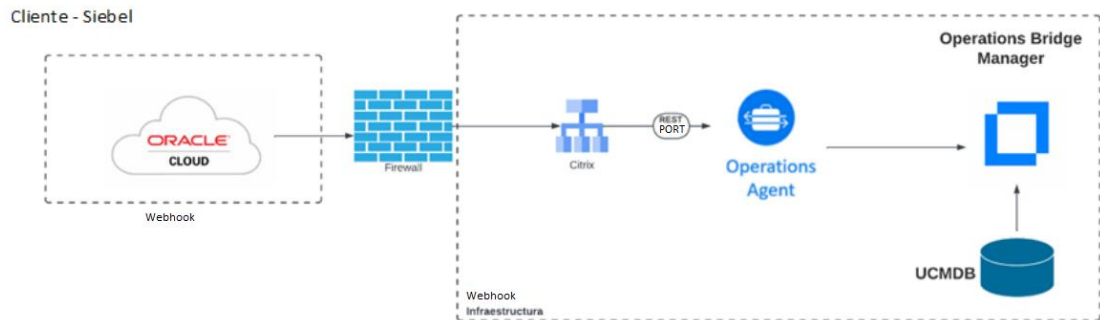


Ilustración 4. Diseño técnico Siebel. Autoría BPS Consultores.

- Monitoreo Cloud APC:

En este proyecto se implementó la política en OBM, la cual se encarga de la recepción de los eventos problema y solución de una nube AWS. Durante el proyecto se llevaron a cabo las tareas de:

- Creación de política para parseo de eventos.
- Creación de filtros de visualización.
- Creación de página de visualización.

La realización de las demás tareas como diseño e implementación de arquitectura y árbol de servicio, fue desarrollada por otra parte del equipo de proyectos [2].

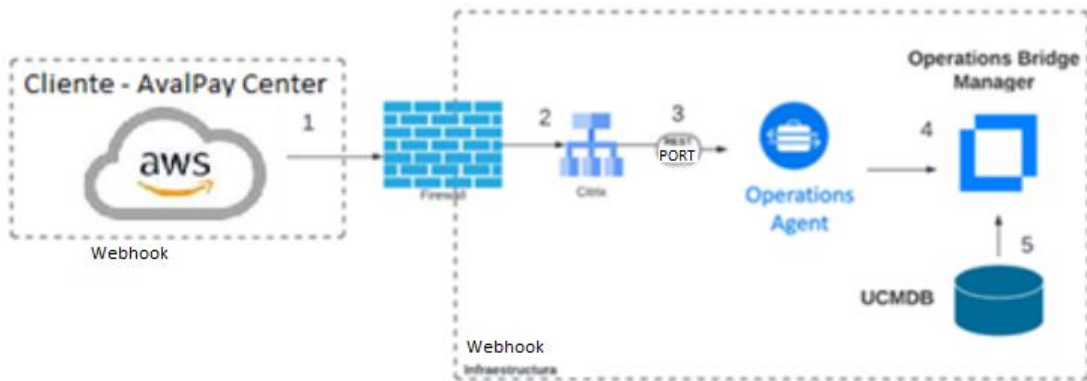


Ilustración 5. Diseño técnico APC. Autoría BPS Consultores.

- **Monitoreo On-premise DIEBOLD:**

En este proyecto se implementó la configuración de monitores en SiteScope, los cuales se encargan de monitorear el rendimiento operacional de la infraestructura On-premise del cliente, con el fin de alertar según corresponda. Durante el proyecto se llevaron a cabo las tareas de:

- Creación de ambiente SiteScope y monitores.
- Configuración de métricas y umbrales para los monitores.
- Integración SiteScope OBM para proyecto.
- Creación de página de visualización OBM.

La realización de las demás tareas como diseño e implementación de arquitectura y árbol de servicio, fue desarrollada por otra parte del equipo de proyectos [2].

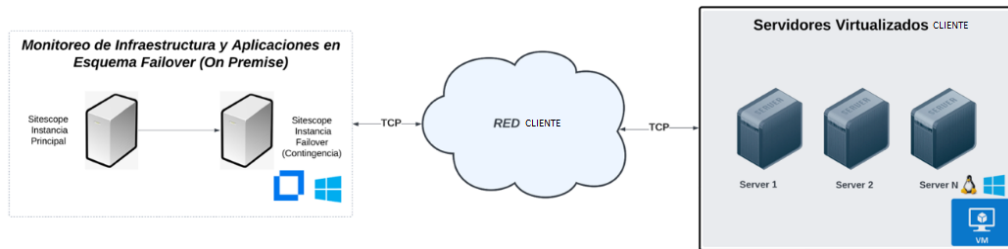


Ilustración 6. Diseño técnico Diebold. Autoría BPS Consultores.

- **Monitoreo On-premise SBL:**

En este proyecto se implementó la configuración de monitores en SiteScope, los cuales se encargan de monitorear el rendimiento operacional de la infraestructura On-premise del cliente, con el fin de alertar según corresponda. Durante el proyecto se llevaron a cabo las tareas de:

- Creación de ambiente SiteScope y monitores.
- Configuración de métricas y umbrales para los monitores.
- Integración SiteScope OBM para proyecto.
- Creación de página de visualización OBM.

La realización de las demás tareas como diseño e implementación de arquitectura y árbol de servicio, fue desarrollada por otra parte del equipo de proyectos [2].

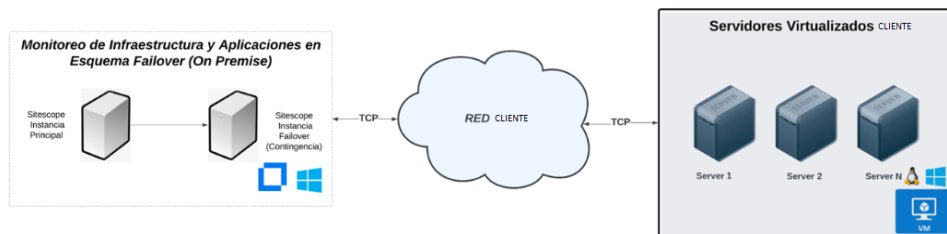


Ilustración 7. Diseño técnico SBL. Autoría BPS Consultores.

3.2. Implementación de solución software para automatizar toma de evidencias

3.2.1. Desarrollo scripts

Según el esquema BPMN que detalla el procedimiento para la automatización de la captura de evidencias, se han concebido y elaborado seis guiones operativos en lenguaje Python, destinados a regir la lógica subyacente a la automatización (consultar Ilustración 1 para una exposición más detallada). Los guiones o scripts creados desempeñan las siguientes funciones:

3.2.1.1. Login.py

El presente guion operativo integra la secuencia lógica que habilita el acceso al sistema por parte del operador encargado de programar la captura de evidencias antes y después de las ventanas de mantenimiento. El mencionado guion, tiene como objetivo principal la preservación de las credenciales necesarias para establecer una conexión directa con la API de SiteScope. Asimismo, almacena la dirección de correo electrónico corporativo del operador que posteriormente, será añadido en el contenido del mensaje de notificación por correo electrónico.

```

> def validar_correo(correo): ...
> def validar_campos(correo, user, password): ...
> def guardar_informacion(correo, user, password): ...
> def ejecutar_bat(): ...
> def validar_ingreso(): ...

# Ventana principal
root = tk.Tk()
root.title("Gestor captura de evidencias")
root.geometry("450x450")
root.resizable(False, False)
fontSubTitle = font.Font(family='Pier Sans')
estilo = ttk.Style()
estilo.configure("BordesRedondos.TEntry", borderwidth=1, relief="groove", padding=(10, 5))
frame_titulo = tk.Frame(root, bg="#59c5d3")
frame_titulo.grid(row=0, column=0, sticky="ew")
label_title = tk.Label(frame_titulo, text="LOGIN", bg="#59c5d3", fg="white", font=(fontSubTitle, 28, "bold"))
label_title.pack(pady=(25))
frame_contenido = tk.Frame(root, bg="white")
frame_contenido.grid(row=1, column=0, sticky="nsew")
label_correo = tk.Label(frame_contenido, text="Correo Electrónico:", bg="white", fg="#59c5d3", font=(fontSubTitle, 14, "bold"))
label_correo.pack(pady=10)
entry_correo = ttk.Entry(frame_contenido, style="BordesRedondos.TEntry", width=30, font=(fontSubTitle, 10))
entry_correo.pack(side="top", pady=(5, 5))
label_user = tk.Label(frame_contenido, text="Usuario ATH:", bg="white", fg="#59c5d3", font=(fontSubTitle, 14, "bold"))
label_user.pack(pady=10)
entry_user = ttk.Entry(frame_contenido, style="BordesRedondos.TEntry", width=30, font=(fontSubTitle, 10))
entry_user.pack(side="top", pady=(5, 5))
label_password = tk.Label(frame_contenido, text="Contraseña:", bg="white", fg="#59c5d3", font=(fontSubTitle, 14, "bold"))
label_password.pack(pady=10)
entry_password = ttk.Entry(frame_contenido, show="*", style="BordesRedondos.TEntry", width=30, font=(fontSubTitle, 10))
entry_password.pack(side="top", pady=(5, 5))
> def on_enter(event): ...
> def on_leave(event): ...
btn_ingresar = tk.Button(frame_contenido, text="Ingresar", borderwidth=0, highlightthickness=0, fg="white", bg="#59c5d3", font

```

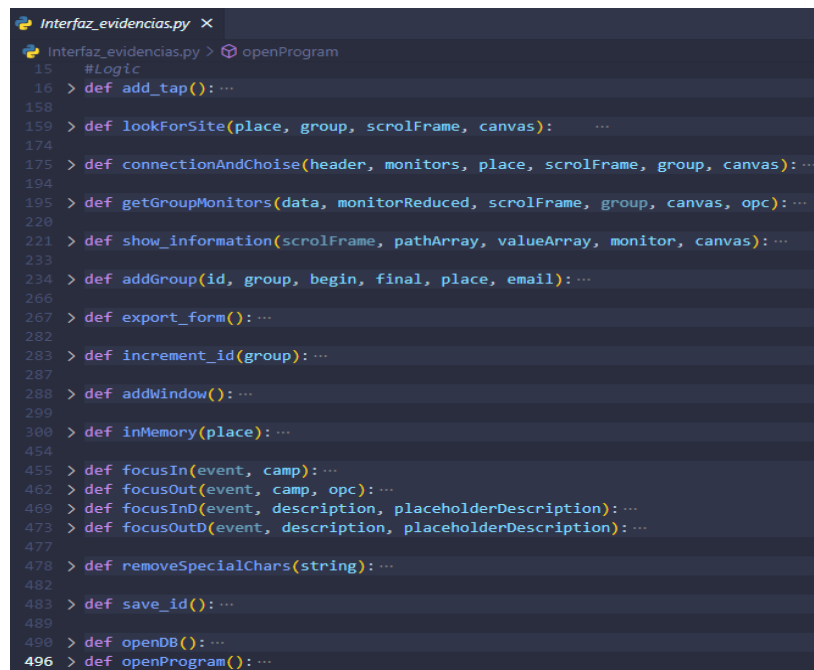
Ilustración 8. Script login. Autoría propia.

Conforme se aprecia en la ilustración 8, el script o guion lógico junto con su interfaz gráfica se componen de cinco métodos o funciones. El método "validar_ingreso" se utiliza para llevar a cabo la verificación de los campos de entrada que contendrán los datos proporcionados por el operador. Este método se subdivide en "validar_correo" y "validar_campos". El primero de ellos tiene la responsabilidad de confirmar la congruencia del formato del correo electrónico, mientras que el segundo verifica que ningún campo permanezca sin información. El flujo de ejecución continúa con el método "guardar_informacion", que se ocupa de guardar las credenciales de acceso y de proporcionarlas a los otros scripts o guiones lógicos, lo que les permite conectarse al entorno de monitoreo de SiteScope de manera efectiva. Finalmente, se ejecuta el método "ejecutar_bat", el cual activa el sistema encargado de programar la captura de evidencias.

3.2.1.2. Interfaz_evidencias.py

El presente guion operativo aloja la lógica esencial para la activación de la interfaz gráfica, la selección de hosts y la inserción de los parámetros requeridos con el

propósito de programar la captura de evidencias, culminando con la generación y exportación de un archivo en formato CSV que recoge los datos correspondientes quedando almacenado en un directorio cuyo nombre es el ID de la programación, almacenado en el directorio “base_datos”. Este guion operativo en cuestión consta de un conjunto compuesto por dieciocho métodos, los cuales exponen minuciosamente el funcionamiento inherente y engendran los desenlaces anticipados de la operación.



```
Interfaz_evidencias.py X
Interfaz_evidencias.py > openProgram
15 #Logic
16 > def add_tap(): ...
158
159 > def lookForSite(place, group, scrolFrame, canvas): ...
174
175 > def connectionAndChoise(header, monitors, place, scrolFrame, group, canvas): ...
194
195 > def getGroupMonitors(data, monitorReduced, scrolFrame, group, canvas, opc): ...
220
221 > def show_information(scrolFrame, pathArray, valueArray, monitor, canvas): ...
233
234 > def addGroup(id, group, begin, final, place, email): ...
266
267 > def export_form(): ...
282
283 > def increment_id(group): ...
287
288 > def addWindow(): ...
299
300 > def inMemory(place): ...
454
455 > def focusIn(event, camp): ...
462 > def focusOut(event, camp, opc): ...
469 > def focusInD(event, description, placeholderDescription): ...
473 > def focusOutD(event, description, placeholderDescription): ...
477
478 > def removeSpecialChars(string): ...
482
483 > def save_id(): ...
489
490 > def openDB(): ...
496 > def openProgram(): ...
```

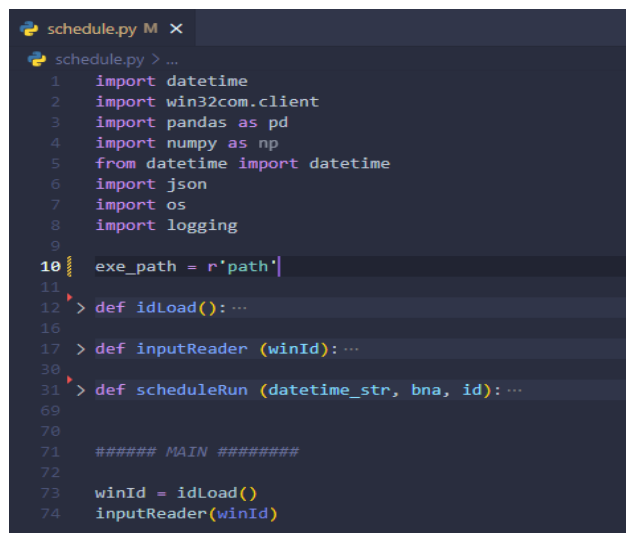
Ilustración 9. Script Interfaz_evidencias.py. Autoría propia.

Como lo describe la ilustración 9, los métodos en consideración asumen el papel cardinal en el desarrollo subyacente a la interfaz gráfica. En su esencia, estas funciones se dedican a la generación de la interfaz gráfica, al establecimiento de conexión con el

entorno de producción SiteScope, a la identificación precisa de los monitores que serán elegidos para captura de evidencias y exportar archivo csv con datos de programación.

3.2.1.3. Schedule.py

Este guion operativo asume la responsabilidad de orquestar la planificación de la captura de evidencias a través del empleo del programador de tareas de Windows. Su estructura lógica se compone de tres métodos que se encargan de interpretar el contenido del archivo CSV almacenado, de configurar el mencionado programador de tareas en segundo plano y de este modo, basándose en los horarios proporcionados a través de la interfaz, se ejecuta la captura de evidencias (consulte Ilustración 10 para más detalles).



```
schedule.py M X
schedule.py > ...
1 import datetime
2 import win32com.client
3 import pandas as pd
4 import numpy as np
5 from datetime import datetime
6 import json
7 import os
8 import logging
9
10 exe_path = r'path|'
11
12 > def idLoad(): ...
16
17 > def inputReader (winId): ...
30
31 > def scheduleRun (datetime_str, bna, id): ...
69
70
71 ##### MAIN #####
72
73 winId = idLoad()
74 inputReader(winId)
```

Ilustración 10. Schedule.py. Autoría propia.

La ilustración 10 esclarece métodos vitales para programar la captura de evidencias en el programador de tareas de Windows. Estos métodos aseguran la correcta ejecución del guion operativo que captura evidencias en el entorno SiteScope,

configurando con precisión el programador de tareas en el horario establecido. El módulo denominado idLoad asume la responsabilidad de la lectura del identificador asociado a la programación de captura de evidencias. A continuación, el módulo denominado inputReader lleva a cabo la adquisición de la totalidad de la información concerniente a dicha programación. Por último, el módulo scheduleRun se encarga de orquestar la programación destinada a activar el script Sitescope_rest.py que concuerda con el identificador previamente mencionado.

3.2.1.4. Sitescope_rest.py

El guion operativo en consideración, está diseñado con el propósito de capturar evidencias en el contexto del entorno SiteScope. Dicho guion se compone de ocho métodos que desempeñan la función de extraer datos de host y monitoreo para la captura de evidencias desde el archivo CSV, y de exportar información concerniente al estado y descripciones de monitores y grupos en formato Excel como evidencias, el cual quedará almacenado de manera automática en el directorio nombrado con su identificador, ubicado dentro de “base_datos” (véase 3.3. Resultados para más detalles).

```

1 > def openRegister(): ...
2
3 register = openRegister()
4 registeredUser = register["user"]
5 registeredPassword = register["password"]
6 user = registeredUser
7 password = registeredPassword
8
9 > def inputReader(id_input): ...
10
11 > def mainProcess (data): ...
12
13 > def getMonitorStatus(header, monitor) : ...
14
15 > def getGroupStatus(header, monitor) : ...
16
17 > def getGroupMonitors(header, element): ...
18
19 > def collectInformation(data): ...
20
21 > def addToExcel(input_id): ...
22
23 > def autoReport(): ...
24
25
26 #MAIN
27 inputReader(sys.argv[1])
28 mainProcess(monitors)
29 addToExcel(sys.argv[1])
30 autoReport()

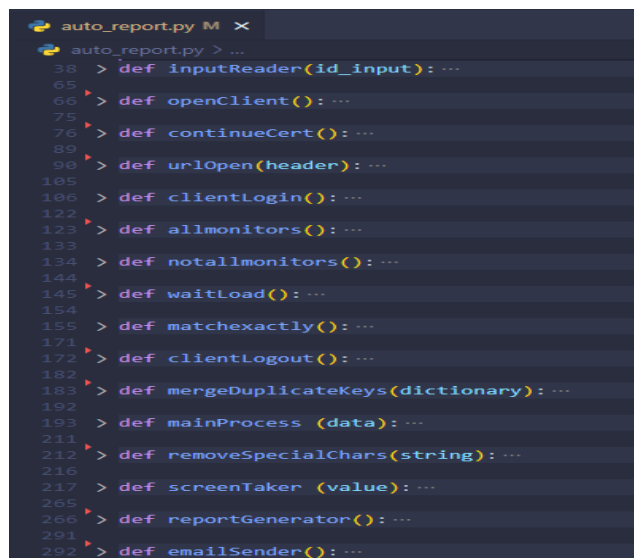
```

Ilustración 11. Sitescope_rest.py. Autoría propia.

La ilustración 11, brinda una representación visual de la estructura lógica elaborada con el propósito de efectuar la adquisición de evidencias. openRegister se encarga de la conexión con el entorno de monitoreo SiteScope. InputReader lee el archivo csv, para posteriormente en el mainProcess, tener acceso a los monitores y grupos las cuales serán dispuestas en la ubicación específica designada como la carpeta correspondiente al ambiente de automatización. La adquisición de estas evidencias se lleva a cabo a través de la interfaz de programación de aplicaciones (API) de acceso público SiteScope. Esta API se construye mediante métodos que tienen un prefijo "get" y se utiliza en consonancia con los selectores de consulta configurados en el entorno Micro Focus SiteScope.

3.2.1.5. Auto_report.py

El guion operativo en cuestión asume la responsabilidad de llevar a cabo las capturas de pantalla del entorno de monitoreo SiteScope, siendo estas posteriormente integradas en un archivo en formato PDF. Este archivo resultante se almacena de manera automática en la carpeta designada bajo el nombre "db_evidencias_capturadas".



```
auto_report.py M X
auto_report.py > ...
38 > def inputReader(id_input): ...
65
66 > def openClient(): ...
75
76 > def continueCert(): ...
89
90 > def urlOpen(header): ...
105
106 > def clientLogin(): ...
122
123 > def allmonitors(): ...
133
134 > def notallmonitors(): ...
144
145 > def waitLoad(): ...
154
155 > def matchexactly(): ...
171
172 > def clientLogout(): ...
182
183 > def mergeDuplicateKeys(dictionary): ...
192
193 > def mainProcess (data): ...
211
212 > def removeSpecialChars(string): ...
216
217 > def screenTaker (value): ...
265
266 > def reportGenerator(): ...
291
292 > def emailSender(): ...
```

Ilustración 12. Auto_report.py. Autoría propia.

La representación visual contenida en la Ilustración 12 exhibe los métodos que asumen la responsabilidad de llevar a cabo el proceso de captura de pantalla. Este procedimiento, aunque automatizado, sigue los mismos pasos que llevaría a cabo un administrador del cliente. Específicamente, la automatización efectúa capturas de pantalla mediante interacciones de clics. Para materializar esta operación, se ha implementado una estructura de directorios denominada

"Assets" en la carpeta raíz. El guion operativo contrasta las imágenes archivadas en dicha ubicación con las imágenes localizadas en el escritorio de Windows. En el caso de un parecido entre las imágenes, el guion procede a ejecutar las directrices previamente configuradas. Las capturas son temporalmente alojadas en la carpeta "screens," ubicada igualmente en el entorno de la automatización, y se eliminan conforme se genera el archivo PDF finalizado que alberga las evidencias [2].

3.2.1.6. Send_email.py

Este guion lógico tiene la responsabilidad de facilitar el envío de las evidencias capturadas al correo corporativo designado. Esta acción se lleva a cabo a través de la interacción entre Python y la biblioteca pyautogui. Utilizando esta biblioteca, el guion simula la actividad del operador, abriendo la aplicación de mensajería y gestionando el envío de todos los archivos desde la programación de captura de evidencias.

```

1 import pyautogui
2 import os
3 import json
4
5 general_path = os.path.dirname(os.path.abspath(__file__))
6 json_id_name = "identification.json"
7 json_id_path = os.path.join(general_path, json_id_name)
8 identification = ""
9 json_register_name = "registro.json"
10 json_register_path = os.path.join(general_path, json_register_name)
11 register = ""
12
13
14 > def openGmail(): ...
22
23 > def identificationReader(): ...
36
37 > def writeCredentials(): ...
70
71 > def writeBody(): ...
89
90 > def attachEvidence(): ...
111
12 > def sendEmail(): ...
23
24 # MAIN
25 openGmail()
26 identificationReader()
27 writeCredentials()
28 writeBody()
29 attachEvidence()
30 #sendEmail()

```

Ilustración 13. Script send_email.

Como se puede apreciar en la ilustración 13, el proceso de envío de las evidencias por correo electrónico involucra seis métodos o funciones. El primer método, denominado "openGmail", se encarga de establecer la conexión con el cliente de correo electrónico y acceder a la bandeja de entrada. El segundo método, "identificationReader", se encarga de identificar las evidencias correctas a enviar, basándose en los identificadores de las programaciones incluidos en los nombres de los directorios dentro del directorio "base_datos". El tercer método, "writeCredentials", gestiona la introducción del correo destinatario y el asunto del email. El cuarto método, "writeBody", se encarga de completar el contenido del correo electrónico. El quinto método, "attachEvidence", se dedica a adjuntar todos los archivos que deben ser enviados. Finalmente, el sexto método, "sendEmail", presiona el botón de envío del correo electrónico.

3.2.2. Frontend interfaz gráfica

A continuación, se presenta el diseño de la interfaz gráfica desarrollado para la programación de captura de evidencias. La elaboración del diseño consideró especialmente las herramientas disponibles en la librería TKinter de Python.

GESTOR CAPTURA DE EVIDENCIAS SITESCOPE

Host de producción: HOST

Nombre relacionado: nombre,nombre,nombre...

Monitor Grupo

Hora y fecha de inicio: dd/mm/aa-HH:mm:ss

Hora y fecha final: dd/mm/aa-HH:mm:ss

Correo Corporativo: Email de notificación

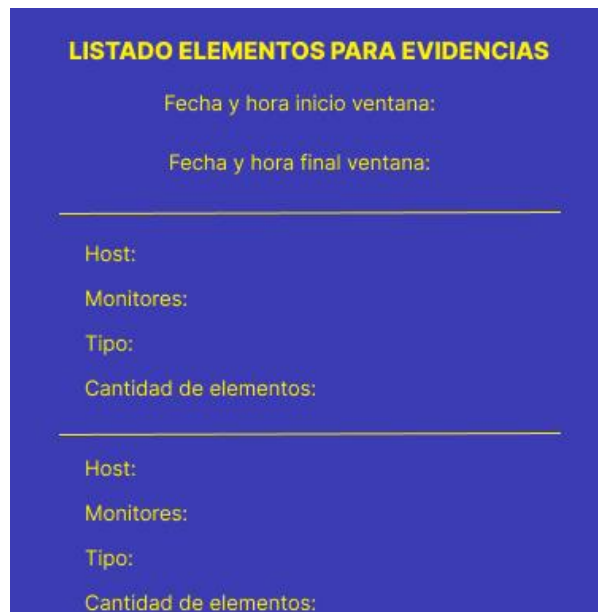
Descripción:
Descripción sobre programación de captura de evidencias

Tipo	Nombre	Path
------	--------	------

Ilustración 14. Interfaz gráfica. Autoría propia.

La Ilustración 14 representa el diseño de la interfaz gráfica destinada a configurar la captura de evidencias. La usabilidad de esta interfaz se basa en la interacción del cliente con un formulario, que le permite especificar el nombre de los monitores o grupos de monitores deseados, separados por comas. Tras la búsqueda, el usuario puede visualizar los resultados y programar la ejecución del script para la captura de evidencias (vea sección 3.3. Resultados para mayor información).

Adicionalmente, es necesario incluir la fecha/hora de inicio y fin, una descripción, y el correo corporativo para recibir notificaciones al término de la ventana. Estos elementos garantizan una adecuada coordinación y comunicación interna durante el proceso de mantenimiento.



The image shows a blue window titled "LISTADO ELEMENTOS PARA EVIDENCIAS" in yellow text. It contains two sections, each separated by a horizontal line. The first section has labels for "Fecha y hora inicio ventana:", "Fecha y hora final ventana:", "Host:", "Monitores:", "Tipo:", and "Cantidad de elementos:". The second section has labels for "Host:", "Monitores:", "Tipo:", and "Cantidad de elementos:".

Ilustración 15. Diseño resumen evidencias. Autoría propia.

Después de realizar la búsqueda, al seleccionar el botón "Agregar a Evidencias", se recopila la información ingresada en los campos de la interfaz, la cual se resume en una nueva ventana, como se muestra en la ilustración 15. Esta ventana incluye el nombre del host en el que se realizará la búsqueda, los monitores o grupos específicos y la cantidad de elementos que serán sometidos al proceso de toma de evidencia. Este diseño permite agregar más hosts y proporciona una visión clara y organizada de la información ingresada (véase Ilustración 14. **Interfaz gráfica. Autoría propia.** para

más detalles). Al finalizar, se exporta el archivo csv que se encarga de la ejecución de los scripts para tomar las evidencias. El archivo csv estará almacenado en la carpeta “db_evidencias_programadas” contenida en el directorio principal que contiene el ambiente de la solución, con el nombre “programacion_ventanasIDventana” (consulte Ilustración 16 para más detalles).

3.2.3. Backend solución consumo API

Utilizando la API pública de SiteScope, se establece la conexión con los diversos hosts del cliente para obtener información de los monitores seleccionados a través de la interfaz gráfica. Esta API se usa dentro de la lógica de este desarrollo, de la siguiente manera:

```
api_url = 'https://' + host + ':port/SiteScope/api/monitors/'
```

[Código 1. Autoría propia]

La búsqueda en SiteScope permite recopilar información vital del monitoreo, como el estado de los monitores, nombres, descripciones y otros detalles relevantes para el proceso de toma de evidencias. Esta obtención sistemática de datos a través de la API garantiza un flujo eficiente y preciso de la información necesaria para el correcto funcionamiento del sistema.

Una vez que se disponen de los datos de los monitores, se procede a generar un archivo Excel que contendrá la información detallada, presentada de manera tabulada. El Excel se estructurará en dos hojas de cálculo denominadas "evidencia PRE" y "evidencia POST", en función de la fecha y hora de inicio y finalización de la ventana de mantenimiento respectiva (véase sección 3.3.2. Archivo Excel para mayor información).

3.2.4. Notificación por email

Una vez completado el proceso de captura de evidencias, se generará una notificación por correo electrónico dirigida a la cuenta corporativa proporcionada durante la programación de la toma de evidencias. Esta notificación será enviada de manera inmediata al finalizar la ventana de mantenimiento y obtener la evidencia en formato Excel y las dos capturas de pantalla (pre y post). La primera de estas alternativas, actualmente en funcionamiento, opera considerando el proveedor de servicios de correo electrónico, específicamente Gmail. A través del uso de comparaciones basadas en imágenes y haciendo uso, una vez más, de la biblioteca "pyautogui", se procede a la composición del correo electrónico, incluyendo los archivos adjuntos pertinentes, y posteriormente se procede al envío al destinatario (véase sección 3.2.1.6. Send_email.py para más detalles). Sin embargo, es importante destacar que este método presenta numerosas limitaciones debido a la necesidad de configuraciones específicas para lograr la coincidencia precisa de las imágenes.

Por otro lado, la segunda alternativa, que se encuentra en desarrollo en colaboración con el cliente, implica la utilización de un servicio de correo electrónico proporcionado por plataformas como AWS o Google Cloud, que permite el envío de correos electrónicos de manera más efectiva, al no depender de coincidencias visuales, sino de una configuración directa y controlada desde el propio servicio.

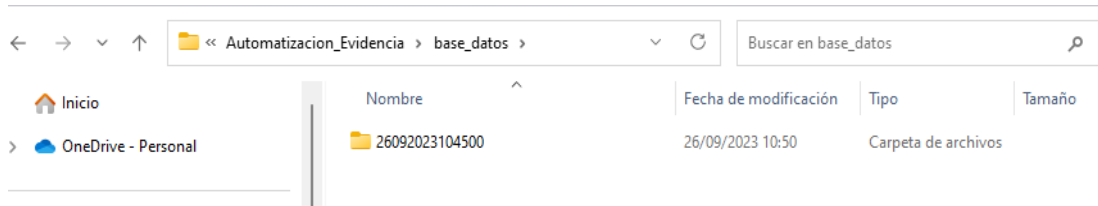
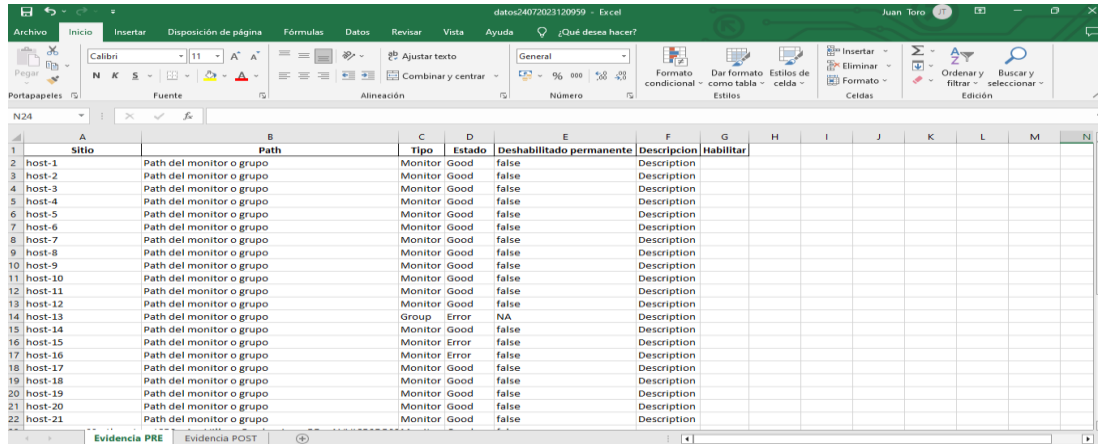


Ilustración 17. Archivo csv dentro del directorio db_evidencias_programadas. Autoría propia.

La ilustración 17, presenta el directorio que alberga los archivos CSV. Dicho directorio desempeñará la función de base de datos para las programaciones de capturas, con un índice de búsqueda único basado en el ID de cada programación.

3.3.2. Archivo Excel



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	Sitio	Path	Tipo	Estado	Deshabilitado permanente	Descripcion	Habilitar							
2	host-1	Path del monitor o grupo	Monitor	Good	false	Description								
3	host-2	Path del monitor o grupo	Monitor	Good	false	Description								
4	host-3	Path del monitor o grupo	Monitor	Good	false	Description								
5	host-4	Path del monitor o grupo	Monitor	Good	false	Description								
6	host-5	Path del monitor o grupo	Monitor	Good	false	Description								
7	host-6	Path del monitor o grupo	Monitor	Good	false	Description								
8	host-7	Path del monitor o grupo	Monitor	Good	false	Description								
9	host-8	Path del monitor o grupo	Monitor	Good	false	Description								
10	host-9	Path del monitor o grupo	Monitor	Good	false	Description								
11	host-10	Path del monitor o grupo	Monitor	Good	false	Description								
12	host-11	Path del monitor o grupo	Monitor	Good	false	Description								
13	host-12	Path del monitor o grupo	Monitor	Good	false	Description								
14	host-13	Path del monitor o grupo	Group	Error	NA	Description								
15	host-14	Path del monitor o grupo	Monitor	Good	false	Description								
16	host-15	Path del monitor o grupo	Monitor	Error	false	Description								
17	host-16	Path del monitor o grupo	Monitor	Error	false	Description								
18	host-17	Path del monitor o grupo	Monitor	Good	false	Description								
19	host-18	Path del monitor o grupo	Monitor	Good	false	Description								
20	host-19	Path del monitor o grupo	Monitor	Good	false	Description								
21	host-20	Path del monitor o grupo	Monitor	Good	false	Description								
22	host-21	Path del monitor o grupo	Monitor	Good	false	Description								

Ilustración 18. Excel evidencia PRE y POST. Autoría propia.

Como se puede apreciar en la ilustración 18, el gestor de captura de evidencias se encarga de recopilar información esencial proveniente de los monitores para verificar el correcto funcionamiento del entorno de monitoreo. Antes de iniciar la ventana de mantenimiento, se procede a capturar las primeras evidencias, las cuales son almacenadas en la hoja de cálculo denominada "Evidencias PRE". De manera similar, al concluir la ventana de mantenimiento, se efectúa una nueva captura de evidencias, las cuales se archivan en la hoja de cálculo designada como "Evidencias POST". Esta disposición garantiza que todos los datos relevantes del proceso de monitoreo se encuentren debidamente registrados y disponibles para su análisis y evaluación posterior. El archivo Excel estará almacenado en su directorio, dentro del directorio "base_datos" contenida en el directorio principal que contiene el ambiente de la

solución, con el nombre “datos<IDventana>”. La siguiente ilustración demuestra lo anterior:

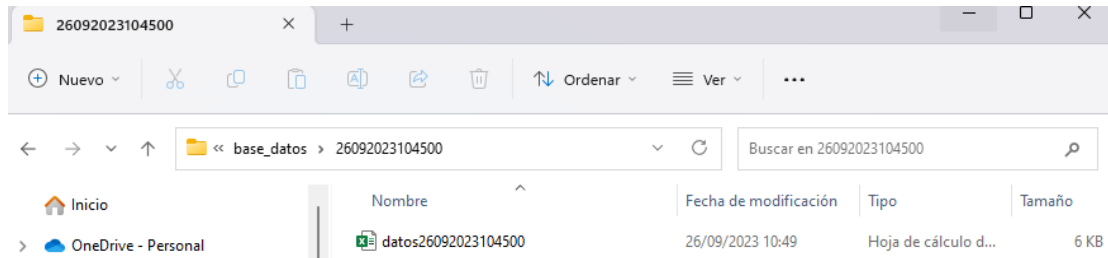


Ilustración 19. Evidencias guardadas en directorio. Autoría propia.

Como se puede observar en la ilustración 19 el archivo Excel se almacena automáticamente en el directorio con su ID dentro del directorio “base_datos”. Este se almacena junto con el archivo CSV y los screenshot que se detallarán a continuación.

3.3.3. Capturas de pantalla PDF

Posterior a la recolección de evidencias en formato Excel, se procede con la captura de pantallas que complementará el reporte en formato PDF, resultando en un proceso integral de toma de evidencias. Para llevar a cabo esta tarea, el script diseñado debe interactuar con el entorno Windows, permitiéndole acceder automáticamente a cada host mediante el cliente local de Microfocus SiteScope. El proceso de captura de pantallas se ejecutará, almacenando las imágenes obtenidas en la carpeta "screens" ubicada en el directorio principal, y generando un archivo PDF denominado "report_<IDventana>", que será almacenado en su directorio, dentro del directorio "base_datos".

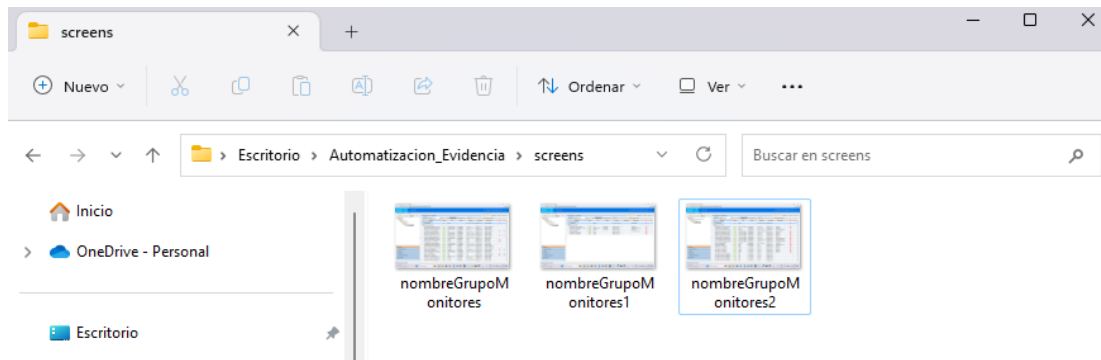


Ilustración 20. Capturas de pantalla en el directorio screens. Autoría propia.

La ilustración 20 presenta el directorio que alberga las capturas de pantalla. Dicho directorio desempeñará la función de contener temporalmente las imágenes. Una vez las imágenes estén en el archivo pdf, éstas se eliminarán del directorio para no entrar en conflicto con las demás evidencias.

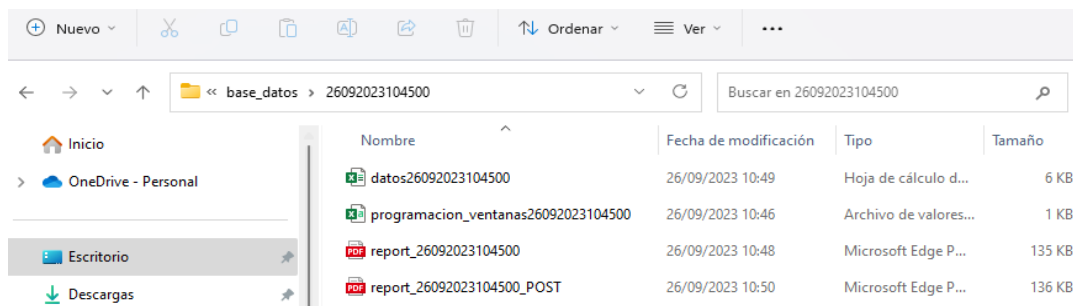


Ilustración 21. Reporte PDF en directorio. Autoría propia.

La ilustración 21 presenta el directorio que alberga las evidencias en formato Excel y pdf. Dicho directorio desempeñará la función de base de datos para las capturas exitosas, con un índice de búsqueda único basado en el ID de cada programación.

El personal responsable cuenta ahora con la capacidad de programar múltiples capturas de evidencias, lo que les permite obtener en tiempo real información actualizada sobre el funcionamiento del monitoreo directamente en su correo electrónico. Esta mejora se traduce en un ahorro de tiempo y esfuerzo, lo que contribuye a una mayor calidad en el desempeño de sus labores (véase Ilustración 21).

En último término, en relación a la interfaz, se llevaron a cabo alteraciones no previamente contempladas en el diseño frontal de la interfaz. El propósito subyacente consistió en concebir una presentación visual más amigable y acorde con la identidad corporativa de BPS.

3.3.4. Interfaz gráfica versión 1



The image shows a web browser window with the title "Gestor captura de evidencias". The main content area has a teal header with the word "LOGIN" in white. Below the header, there are three input fields with labels in teal: "Correo Electrónico:", "Usuario ATH:", and "Contraseña:". Each label is followed by a white rectangular input box. At the bottom of the form is a teal button with the word "Ingresar" in white.

Ilustración 22. Login aplicación. Autoría propia.

La ilustración 22, exhibe la pantalla de inicio de sesión, en la cual se requiere ingresar el correo electrónico del operador que utilizará la herramienta de gestión de evidencias, así como las credenciales necesarias para la conexión de la interfaz con SiteScope MicroFocus. Esta constituye la primera vista de la solución al iniciar el proceso.



Ilustración 23. Interfaz gráfica resultante.

La ilustración 23, representa la interfaz definitiva desarrollada para la gestión de la captura de evidencias. Esta interfaz satisface los requisitos previamente mencionados y genera el archivo CSV fundamental para el adecuado funcionamiento de la automatización. Es relevante destacar que las modificaciones de diseño efectuadas no estaban previamente planificadas y se llevaron a cabo conforme a la identidad de la empresa.

A continuación, se presentan las siguientes gráficas que ilustran las mejoras en términos de productividad, eficiencia operativa y calidad de los datos obtenidos:

3.3.5. Productividad

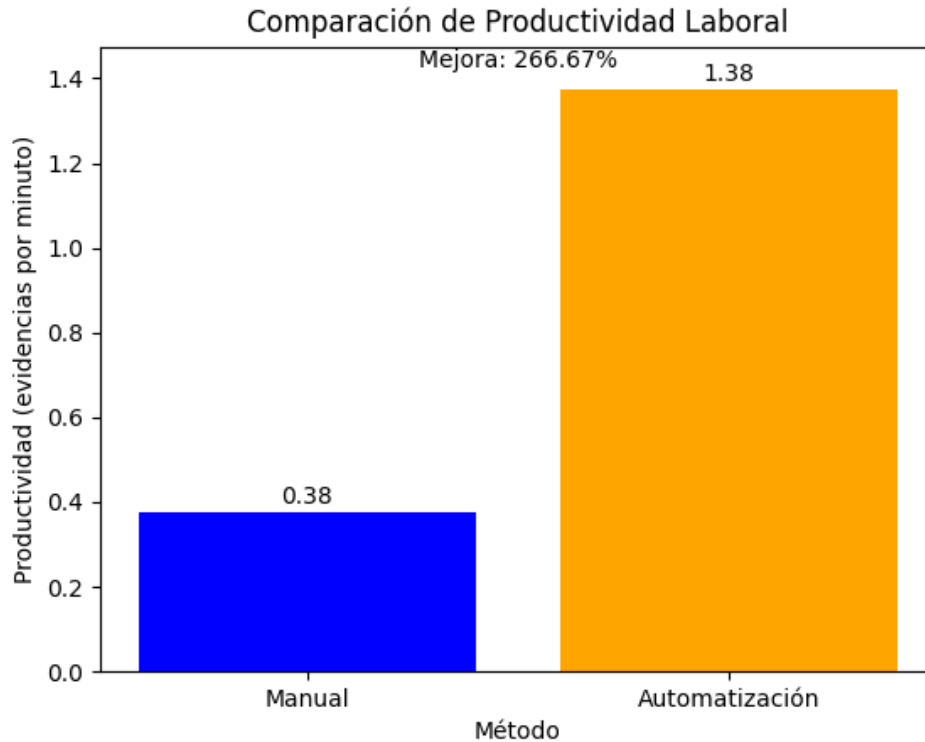


Ilustración 24. Comparación de aumento en productividad. Autoría propia.

Para obtener los resultados de la ilustración 24, se utiliza la fórmula de productividad laboral “*Productividad Laboral = (Output o Producción) / (Input o Recursos Utilizados)*” que se basa en el concepto económico de relación entre la producción obtenida y los recursos utilizados para obtenerla. El objetivo es medir cuánta producción o valor se genera a partir de una determinada cantidad de recursos [16]. El gráfico ilustra el incremento de productividad en la toma de evidencias. Se evidencia un aumento del 266.67% en el enfoque automatizado, en contraste con el procedimiento manual de recolección de evidencias. Dicha productividad se obtuvo al calcular el total de evidencias recolectadas de forma manual y automática, en un

periodo de 8 minutos. Una vez culminado el tiempo, se registran 3 procesos de evidencias exitosos manuales versus 11 procesos de evidencias exitosos automáticos [2].

3.3.6. Eficiencia operativa

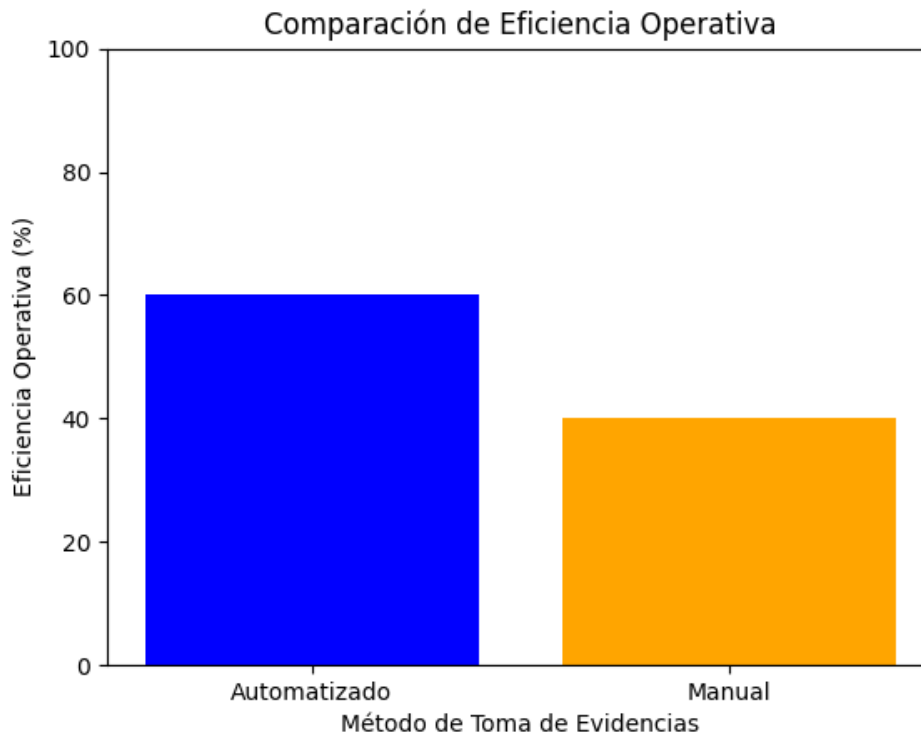


Ilustración 25. Comparación eficiencia operativa. Autoría propia.

La gráfica anterior representa el incremento porcentual de la eficiencia operativa. Dicho incremento asciende a un 20%, lo que conlleva a una significativa optimización del tiempo promedio empleado en la toma de evidencias. La eficiencia operativa se evaluó mediante el cálculo del porcentaje para cada método, dividiendo el número de evidencias tomadas de forma automatizada o manual entre el total de evidencias y multiplicando el resultado por 100 para obtener el porcentaje correspondiente [16].

3.3.7. Mejora de datos

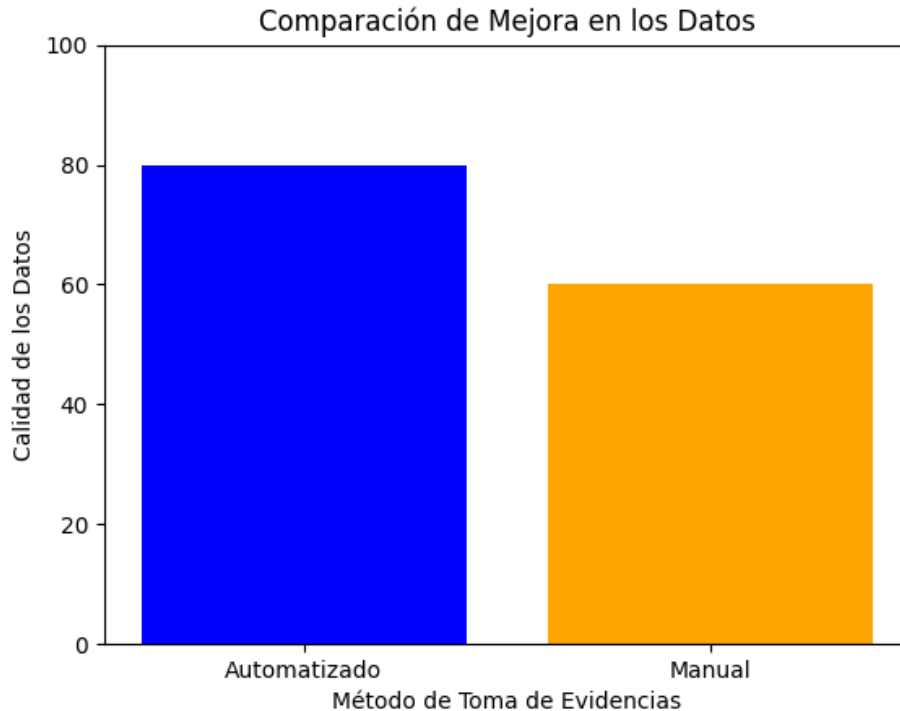


Ilustración 26. Comparación mejora de colecta de datos. Autoría propia.

La gráfica anterior representa la comparación de la mejora en los datos obtenidos entre la toma de evidencias automatizada y la toma manual. Se puede observar un aumento significativo en la cantidad y calidad de datos obtenidos mediante el proceso automatizado, representando un 20% más de datos relevantes en comparación con el proceso manual. Esta diferencia sustancial en la obtención de datos resalta la eficacia y precisión del enfoque automatizado, lo que conlleva a una toma de decisiones más informada y fundamentada en la gestión de los monitores. Se obtuvieron los resultados en base al principio de completitud del análisis de datos, resumido en la siguiente fórmula: “*Compleitud = (Datos Presentes) / (Datos Esperados)*” [16] [2].

Finalmente, los datos recopilados y el análisis comparativo con métodos previos de captura de evidencias, manifiestan claramente los beneficios de la interfaz implementada. Se ha observado una disminución en la carga de trabajo, una mayor eficiencia operativa y una mejora en la calidad de los datos recopilados.

PARTE IV

Conclusiones

Capítulo 4

Conclusiones y recomendaciones

La presente sección de conclusiones representa el cierre de un proyecto significativo y trascendental en el ámbito del monitoreo. A lo largo del desarrollo de esta iniciativa, se ha buscado ofrecer una solución integral para la automatización de la toma de evidencias, con el objetivo de agilizar y mejorar los procesos de recolección y análisis de información. A través de la implementación de una interfaz gráfica intuitiva y funcional, se ha asegurado la eficiencia y efectividad en la programación y seguimiento de las capturas de evidencias, proporcionando una herramienta valiosa para el personal encargado. En esta sección, se presentarán los resultados obtenidos, los logros alcanzados y las perspectivas futuras, enfocadas hacia una mejora continua y el aporte a la excelencia en el ámbito del monitoreo.

4.1. Conocimientos adquiridos y aportes como pasante

Durante el período de pasantía, el estudiante realizó un valioso aporte a los proyectos de monitoreo mediante sus sólidos conocimientos en Cloud, redes y programación. Su participación fue fundamental en la concepción y desarrollo de proyectos de monitoreo que arrojaron resultados sobresalientes para la empresa. Asimismo, contribuyó significativamente a elevar la calidad del servicio ofrecido al cliente mediante la implementación de soluciones automatizadas (consulte Capítulo 4 para más detalles).

En términos generales, el estudiante alcanzó con éxito el objetivo de aprender y destacarse en el entorno empresarial. Definió un camino que culminó con el proyecto de automatización de la toma de evidencias, consolidando así un período de pasantía productivo y enriquecedor.

A través del desarrollo de la automatización de evidencias mediante el uso del lenguaje de programación Python, se logró un enriquecimiento significativo de los conocimientos en múltiples áreas. La implementación de esta solución impulsó la adquisición de nuevos saberes en el ámbito del Frontend al emplear la biblioteca Tkinter para la creación de interfaces gráficas interactivas y visualmente atractivas. Asimismo, se fomentó un mayor entendimiento en el campo del Backend al explorar la integración de APIs, lo que permitió la recopilación de datos y la interacción con

sistemas externos. Además, la interacción directa con el entorno Windows para generar resultados precisos y esperados contribuyó a un conocimiento más profundo de las interacciones entre tecnologías y sistemas, enriqueciendo aún más la experiencia y competencia técnica.

4.2. Conclusiones

Tras la implementación del software toma de evidencias, se desprenden las siguientes conclusiones de relevancia:

1. La concepción e implementación de scripts en el entorno Python para la selección, programación y captura automatizada de evidencias provenientes de los monitores en SiteScope ha resultado en un enfoque altamente eficaz y eficiente. La automatización de estas operaciones ha demostrado mejorar la gestión del proceso de adquisición de datos, al tiempo que establece un meticuloso y coherente registro de las evidencias planificadas. Esta integración sistemática no solo agiliza la toma de evidencias, sino que también facilita un análisis más profundo y riguroso, lo que promueve una base sólida para la toma de decisiones respaldadas por datos. (véase sección 4.2. Conclusiones para más detalles).
2. La creación de una interfaz gráfica en Python, diseñada con enfoque en la gestión integral de los scripts de programación y captura de evidencias, representa un logro trascendente en la optimización de la eficiencia operativa. Su implementación, intuitiva y altamente funcional, alineada con los requerimientos del cliente, constituye un paso significativo hacia la

facilitación del proceso de administración y monitoreo, posibilitando una interacción fluida y empoderada con el sistema.

3. La materialización exitosa de la funcionalidad que hace uso de la API pública de SiteScope, para la adquisición combinada de evidencias textuales en formato Excel y capturas visuales en formato PDF, durante el monitoreo en entornos SiteScope, revela un avance significativo en la automatización de procesos y la mejora de la eficiencia en la gestión de mantenimientos (revise sección 3.3. Resultados para más detalles). La integración fluida y sincronizada de estas operaciones provee un mecanismo coherente y sistemático para la obtención y documentación de información valiosa, que, al ser enviada de manera automatizada al correo corporativo al concluir la ventana de mantenimiento, otorga una capa adicional de agilidad y coherencia en la administración de la infraestructura tecnológica [2].

4.3. Recomendaciones

Como recomendaciones para futuras mejoras, se sugiere continuar optimizando la interfaz con nuevas funcionalidades y características que puedan adaptarse a distintos escenarios y necesidades específicas de los usuarios.

El alcance del proyecto, se ha enfocado únicamente en la automatización de la captura de evidencias, aunque se destaca que la API de SiteScope ofrece una amplia gama de funcionalidades adicionales. A modo de sugerencia, se propone la incorporación de una interfaz adicional que permita un control más completo de las capturas de evidencias, brindando una visualización ordenada de los eventos correspondientes. Asimismo, se podría considerar la capacidad de habilitar o deshabilitar monitores y/o grupos de monitores, con el propósito de lograr una interfaz integral para la gestión de las ventanas de mantenimiento.

Bibliografía

- [1] C. B. S. Paul Hershey, «Systems of Systems approach for monitoring and response across net-centric enterprise systems,» IEEE International Systems Conference, San Diego, CA, USA, 2010.
- [2] BPS Consultores, «Documento interno,» BPS Consultores, Bogotá, 2023.
- [3] Y. Y. F.-Y. W. Rui Qin, «Optimizing the revenue for ad exchanges in header bidding advertising markets,» IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017.
- [4] IEEE, «2755-2017 - IEEE Guide for Terms and Concepts in Intelligent Process Automation,» IEEE Std 2755, USA, 2017.
- [5] P. B. a. R. G. Bhavani, «Investigation for reducing energy consumption for a university campus in Dubai using automation,» IEEE Recent Advances in Intelligent Computational Systems (RAICS), Trivandrum, India, 2015.
- [6] J. H. a. S. C. X. Chang, «Study on Integrated logistics Network Model and Network Design for Waste Electrical and Electronic Equipment,» IEEE International Conference on Service Operations and Logistics, and Informatics, Shanghai, China, 2006.
- [7] D. L. a. V. Y. Lisanti, «The impact of internet technology on creating new positions in Indonesia study case: Companies in Jakarta,» International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 2015.
- [8] T. -e. C. a. Y. -j. W. Z. -s. Ren, «Precision irrigation decision-making support by using an improved genetic algorithm,» World Automation Congress, Kobe, Japan, 2010.
- [9] CEPAL. ILPES, «cepal.org,» 10 01 2002. [En línea]. Available: <https://www.cepal.org/es/publicaciones/7275-competencia-conflictos-regulatorios-la-industria-telecomunicaciones-america>. [Último acceso: 28 09 2023].
- [10] Micro Focus, «Operations Bridge Manage,» docs.microfocus, 01 01 2023. [En línea]. Available: https://docs.microfocus.com/doc/Operations_Bridge_Manager/2023.05/Home. [Último acceso: 28 09 2023].
- [11] Microfocus, «Documentación SiteScope Microfocus,» SiteScope Microfocus, 01 01 2023. [En línea]. Available: https://docs.microfocus.com/doc/SiteScope/11.40/Get_Started/Introduction. [Último acceso: 28 09 2023].

- [12] Python, «Documentación de Python,» Python.org, 01 08 2023. [En línea]. Available: <https://docs.python.org/es/3.11/>. [Último acceso: 12 09 2023].
- [13] Microsoft, «Documentación Microsoft Office Excel,» Excel Microsoft, 01 01 2023. [En línea]. Available: <https://support.microsoft.com/en-us/excel>. [Último acceso: 12 09 2023].
- [14] Universidad de los Andes, «uniandes.edu.co,» planeacion uniandes, 01 01 2016. [En línea]. Available: https://planeacion.uniandes.edu.co/images/Arquitectura-Institucional/INS-45-1-01-01_Modelamiento_de_Procesos_con_BPMN1.pdf. [Último acceso: 14 09 2023].
- [15] D. C. Shaw, «Intelligent Infrastructure Asset Condition Monitoring,» IET International Conference On Railway Condition Monitoring, Birmingham, UK, 2006.
- [16] X. I. E. & C. M. Rotllan-Puig, «LPDynR: A new tool to calculate the land productivity dynamics indicator,» Ecological Indicators, ISSN 1470-160X, 2021.
- [17] V. B. G. S. Iulia Stamatescu, «Distributed Monitoring of Smart Buildings with Cloud Backend Infrastructure,» 5th International Conference on Control, Decision and Information Technologies (CoDIT), Thessaloniki, Greece, 2018 .
- [18] IEEE, «IEE Colloquium on 'Condition Monitoring and Fault Tolerance',» IEE Colloquium on Condition Monitoring and Fault Tolerance, USA, 1990.
- [19] C. Sayago Abreo, « Plataforma que integra los monitoreos del área de seguridad de la información y ciberseguridad de una compañía vigilada por la Superintendencia Financiera de Colombia,» Universidad Santo Tomás, Bogotá, Co, 2019.
- [20] J. Arias Estupiñán, «Diseño e implementación backend de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y medianas empresas.,» Universidad Santo Tomás, Bogotá, Co, 2022.

Anexos

Esta sección proporciona los anexos incorporados que actúan como complementos y respaldo de los capítulos mencionados previamente en el documento.