Universidad ORT Uruguay
Facultad de Ingeniería

Optimización de la gestión de cambios en infraestructura de Tecnología de la Información: un enfoque basado en la automatización de mapeo de dependencias.

Entregado como requisito para la obtención del título de Máster en Gestión de Sistemas de Información.

Sofía Barreiro – 307317

Evelyn González – 169243

Tutor: Gerardo Matturro

2025

Declaración de autoría

Nosotras, Sofía Barreiro y Evelyn González, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizamos el proyecto de tesis del Máster en gestión de sistemas de información;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotras;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Evelyn González

24 de marzo de 2025

Sofía Barreiro

24 de marzo de 2025

Agradecimientos

Agradecemos a nuestras familias, compañeros y tutor por todo el apoyo recibido.

Abstract

En el contexto de la empresa de estudio, la gestión de cambios en infraestructura enfrenta problemas significativos. A pesar de los esfuerzos realizados para asegurar una adecuada comunicación sobre un cambio programado, no hay visibilidad de las dependencias entre los sistemas, lo que dificulta la correcta evaluación del impacto y aumenta los riesgos operacionales.

Este trabajo de investigación tiene como objetivo evaluar la efectividad de una solución automatizada que permita visualizar las dependencias entre los sistemas que involucran servidores Windows y optimizar la comunicación de los cambios.

La metodología elegida para llevar a cabo este trabajo es investigación-acción. Esta metodología tiene un enfoque iterativo que facilita la identificación y resolución de problemas mediante un ciclo continuo de planificación, implementación, análisis y documentación. Este enfoque permite desarrollar un plan específico para abordar los problemas identificados, evaluar los resultados obtenidos y generar soluciones prácticas y adaptadas al contexto de la empresa de estudio.

Como solución a los problemas identificados, se propone el diseño e implementación de un algoritmo para la generación automática de árboles de dependencias y mejoras para la comunicación de un cambio programado.

En conclusión, la implementación de una solución automatizada y alineada con las mejores prácticas de ITIL contribuye a la mejora continua de los servicios de infraestructura que brinda el área de tecnologías de la información (TI) de la empresa de estudio, asegurando una gestión de cambios más eficiente.

Palabras clave

Tecnologías de la información, mapa de dependencias, monitoreo de infraestructura, gestión de cambios, automatización.

Indice

1. Introducción

En un mundo cada vez más interconectado y dependiente de las Tecnologías de la Información (TI), las organizaciones deben gestionar de manera eficiente su infraestructura tecnológica para garantizar la continuidad operativa y minimizar riesgos operativos.

En el contexto de la empresa de estudio, la gestión de cambios en infraestructura de TI enfrenta problemas significativos. A pesar de los esfuerzos realizados para asegurar una comunicación adecuada acerca de los cambios programados, no hay visibilidad sobre las dependencias entre los sistemas. Esta carencia incrementa la probabilidad de que se pasen por alto componentes que no fueron identificados como parte de una solución y que, ante un cambio, podrían verse afectados.

El objetivo principal de esta investigación es evaluar la efectividad de una solución automatizada para la visualización de las dependencias de los sistemas que están alojados en servidores con sistema operativo Windows. Esta solución permitirá optimizar la evaluación previa del impacto de un cambio y, por lo tanto, mejorar la planificación, ejecución y comunicación de estos, reduciendo así los riesgos existentes.

La metodología elegida en este estudio es investigación-acción, la cual sigue un enfoque iterativo que facilita la identificación de problemas, la implementación de soluciones y la evaluación continua de los resultados. Este enfoque permitirá desarrollar un plan específico para resolver las deficiencias observadas en la gestión de cambios, con soluciones adaptadas al contexto particular de la empresa de estudio. A través de ciclos continuos de planificación, implementación, análisis y documentación, se evaluará la efectividad de la solución propuesta y se propondrán ajustes según los resultados obtenidos.

2. Marco teórico

2.1. Términos, definiciones y conceptos relacionados.

TI (Tecnología de la información): La Tecnología de la Información (TI) abarca el uso de software, computadoras y otros dispositivos electrónicos para procesar, almacenar, transmitir y recuperar información [1].

Infraestructura de TI: La infraestructura de TI es el conjunto de componentes físicos y virtuales que permiten el funcionamiento de los sistemas tecnológicos en una organización. Algunos de estos componentes son: *switches*, *routers*, servidores, base de datos y aplicaciones. La infraestructura de TI puede estar en centros de datos de la empresa, en la nube o en entornos híbridos [2].

Monitoreo de infraestructura: El monitoreo de infraestructura implica supervisar, analizar y gestionar el rendimiento, la disponibilidad y el estado de los componentes tecnológicos de una empresa, con el objetivo de prevenir pérdidas y proteger su reputación frente a posibles fallas [3].

Gestión de cambios: La gestión de cambios en infraestructura de TI es una práctica diseñada para minimizar interrupciones en los servicios de TI al realizar modificaciones, adiciones o eliminaciones en cualquiera de sus componentes [4].

ITIL: ITIL® (*Information Technology Infrastructure Library*®) es el marco de trabajo más conocido diseñado para planificar, entregar y administrar el ciclo de vida general de TI dentro de una organización. Proporciona una orientación eficaz para llevar a cabo la transformación de la organización. [5]

Gestión de cambios en el contexto de ITIL: La gestión de cambios en el contexto de ITIL, es un proceso fundamental para garantizar que los cambios en los servicios de TI se realicen de manera controlada, alineada con los objetivos estratégicos de la organización, y minimizando el impacto en la operación de los servicios. Este proceso está diseñado para gestionar todas las modificaciones en los sistemas informáticos, desde la solicitud de cambio hasta su implementación y cierre, asegurando que los cambios sean beneficiosos sin generar interrupciones significativas en los servicios. [6]

El propósito de la gestión de cambios según ITIL es controlar el ciclo de vida de todos los cambios, permitiendo que estos se lleven a cabo con el menor impacto posible en los servicios y en la operación del negocio. Este proceso incluye una serie de pasos que garantizan que los cambios se registren, evalúen, autoricen, prioricen, planifiquen, prueben, implementen y documenten de manera controlada. En esencia, busca que cualquier modificación realizada en el entorno de TI sea perfectamente identificada, evaluada y aprobada para evitar riesgos innecesarios**.**

Los cambios pueden originarse por diversas razones, como las peticiones de los clientes, avances tecnológicos, o incluso oportunidades estratégicas dentro de la organización. Cada cambio debe alinearse con los objetivos del negocio, maximizando el valor para el cliente, reduciendo al mínimo las incidencias, y asegurando que las tecnologías implementadas estén alineadas con las necesidades y metas organizacionales.

Uno de los riesgos más comunes en la gestión de cambios es la falta de documentación adecuada. Si bien los cambios pueden ser comunicados, puede suceder que, al implementarlos, ciertos aspectos del sistema dejen de funcionar correctamente. Este tipo de fallos ocurre principalmente cuando no se tiene documentado adecuadamente qué depende de qué en un sistema, o cuando la documentación no se actualiza después de realizar los cambios. Además, la falta de revisión o de un proceso de validación efectivo puede dar lugar a fallos no previstos, lo que afecta la operatividad de los servicios.

La comunicación es un aspecto clave en la gestión de cambios, especialmente cuando estos afectan o pueden afectar a varios grupos y sistemas dentro de la organización. Es esencial que, antes de ejecutar cualquier cambio, se asegure que todos los involucrados estén al tanto de la modificación que se va a realizar, que hayan evaluado sus posibles impactos y que hayan tomado las medidas necesarias para mitigar cualquier posible efecto negativo. De esta manera, se minimiza la posibilidad de interrupciones en el servicio, y se asegura la correcta coordinación de las acciones entre los distintos grupos involucrados.

En el marco de la gestión de cambios, la creación y mantenimiento de diagramas de dependencias entre los diferentes sistemas y servicios es crucial. Estos diagramas permiten identificar las relaciones entre los componentes del sistema y evaluar

rápidamente el impacto de los cambios en los sistemas afectados. El objetivo es tener siempre una herramienta actualizada que refleje todas las dependencias y que facilite a los equipos de TI la revisión de todos los sistemas afectados por un cambio sin necesidad de que cada área revise individualmente sus sistemas. De esta manera, se optimiza el tiempo y se reduce el riesgo de que se pase por alto alguna dependencia crítica. [7][8]

La gestión de cambios implica una serie de pasos detallados para asegurar que las modificaciones en el sistema se realicen de forma estructurada y controlada. Los pasos clave incluyen [9]:

- Solicitud de cambio: Se registra y autoriza una solicitud de cambio.
- Evaluación y valoración del cambio: Se evalúa el impacto del cambio, así como sus riesgos y beneficios, para determinar su viabilidad.
- Autorización y planificación: Una vez evaluado, el cambio debe ser autorizado y planificado con una estrategia clara para su implementación.
- Implementación y cierre: El cambio se implementa de acuerdo con lo planificado, y una vez completado, se revisa y cierra el proceso, documentando cualquier lección aprendida.

Existen diferentes tipos de cambios en función de su urgencia y nivel de riesgo [10]:

- Cambio estándar: Son cambios preautorizados que se consideran rutinarios, como la instalación de parches o actualizaciones menores.
- Cambio normal: Cambios que siguen un proceso estándar de evaluación y autorización.
- Cambio de emergencia: Cambios de alta prioridad que deben realizarse inmediatamente debido a problemas críticos, como una vulnerabilidad de seguridad.

Para asegurar que un cambio no afecte negativamente el servicio, es necesario contar con una evaluación detallada del impacto, lo cual permite balancear los riesgos y beneficios

del cambio. Sin una evaluación adecuada, el cambio podría no generar los beneficios esperados e incluso podría tener efectos adversos.

Las "7 R's" de ITIL ayudan a guiar este proceso, asegurando que todas las variables relevantes del cambio sean consideradas [11]:

- *Raised* (quién solicita el cambio),
- *Reason* (por qué se realiza),
- *Return* (qué se espera del cambio),
- *Risk* (cuáles son los riesgos),
- *Resource* (qué recursos se necesitan),
- *Responsible* (quién es el responsable), y
- *Relationship* (relación con otros cambios).

Automatización en la gestión de cambios: La automatización en la gestión de cambios implica el uso de herramientas y procesos automatizados para gestionar y aplicar cambios en la infraestructura de TI, reduciendo el riesgo de errores humanos y mejorando la eficiencia. La automatización permite que las tareas repetitivas y de bajo valor agregado, se realicen de manera más eficiente. Esto permite a los equipos de TI que se enfoquen en tareas más estratégicas y de mayor valor, mejorando la productividad general [12].

Asignación de dependencias: La asignación de dependencias se define como el proceso de identificar, comprender y visualizar las relaciones que existen entre aplicaciones, sistemas y procesos dentro de las operaciones de TI de una organización. Un mapeo adecuado de dependencias ayuda a las organizaciones a descubrir vulnerabilidades, optimizar procesos y comprender el impacto que puede tener un problema en un componente sobre el resto del sistema [13].

3. Problema de investigación

Actualmente, el proceso de cambios en infraestructura de TI de la empresa de estudio carece de una herramienta integral que permita mapear de manera centralizada las dependencias entre los distintos sistemas y servicios involucrados. Esto causa que los responsables de los sistemas no tengan una visión clara sobre cómo un cambio afectará a otros servicios o sistemas, aumentando el riesgo de interrupciones y deterioro en la calidad del servicio.

El problema se agrava cuando los cambios afectan a múltiples grupos o sistemas, ya que la falta de comunicación y la falta de un proceso claro de evaluación y coordinación entre las partes involucradas puede resultar en impactos negativos en la operación de los servicios. Aunque se avisan los cambios, no siempre se evalúan ni se toman las acciones preventivas necesarias para mitigar los riesgos asociados.

Este escenario subraya la necesidad urgente de implementar una solución efectiva que permita una mejor gestión de los cambios, asegurando que todas las dependencias estén documentadas, actualizadas y disponibles en una herramienta centralizada de fácil acceso. La investigación que se presenta tiene como objetivo identificar las deficiencias existentes en el proceso de gestión de cambios en servidores Windows, definir los requisitos necesarios para mejorar dicha gestión y proponer una solución que optimice la comunicación y ejecución de los cambios. De esta manera, se busca reducir los riesgos asociados, maximizar el valor del cliente y garantizar la alineación de las tecnologías con los objetivos del negocio, contribuyendo a la mejora continua de los servicios TI en la empresa de estudio.

4. Objetivo general y específicos

4.1. Objetivo general

El objetivo general de esta investigación es analizar el proceso de cambios actual sobre los sistemas que están alojados en servidores con sistema operativo Windows y evaluar la efectividad de una solución automatizada para la visualización de sus dependencias.

Esta solución permitirá optimizar la evaluación previa del impacto de un cambio y, por lo tanto, mejorar la planificación, ejecución y comunicación de estos.

4.2. Objetivos específicos

Se definen tres objetivos específicos:

a) Identificar y analizar las deficiencias actuales en la gestión de cambios en servidores Windows en términos de visibilidad y coordinación de dependencias.
b) Desarrollar y definir los requisitos funcionales y técnicos para una solución automatizada que integre y visualice las dependencias entre servidores, aplicaciones y bases de datos.
c) Formular un plan de acción para la implementación futura de la solución basado en la retroalimentación de los actores clave.

4.3. Preguntas de investigación

Se definen dos preguntas de investigación que guiarán este trabajo hacia los objetivos planteados:

Pregunta de investigación 1 (PI1): ¿Cuáles son las principales deficiencias en la gestión de cambios actuales en servidores Windows que afectan la visibilidad y coordinación de dependencias?

Pregunta de investigación 2 (PI2): ¿Qué requisitos funcionales y técnicos debe cumplir una solución automatizada para mejorar la visibilidad de dependencias y optimizar la gestión de cambios?

5. Identificación de actores e informantes clave

Los actores identificados son clave para identificar y comprender los problemas y necesidades de la organización. Se identifican tres actores clave: el líder de plataforma técnica, el líder del equipo encargado de la administración de los servidores y el líder del centro de atención a usuarios.

El líder de plataforma técnica de infraestructura de sistemas supervisa los equipos responsables de la administración del dominio, bases de datos y monitoreo. Tiene un conocimiento detallado del sistema de monitoreo actual, por lo tanto, gracias a su experiencia, puede orientarnos sobre cómo se implementó la herramienta de monitoreo actual y cómo se utiliza actualmente para dar aviso de un cambio programado.

El líder del equipo encargado de la administración de los servidores tiene un rol esencial en la infraestructura tecnológica de la organización. Supervisa la creación, mantenimiento y disponibilidad de los servidores que alojan la mayoría de las aplicaciones y herramientas utilizadas por los empleados de la empresa. Su experiencia le permite anticipar posibles fallos y tomar decisiones rápidas para minimizar el impacto en la operación diaria, facilitando la continuidad de los procesos de la organización. Su equipo trabaja en estrecha colaboración con otros equipos de infraestructura y con el equipo del centro de atención a usuarios, para asegurar que los sistemas, el soporte y las herramientas colaborativas funcionen de manera eficiente y segura.

Por otro lado, el líder del centro de atención a usuarios se encarga de ofrecer soporte y soluciones a los empleados que enfrentan problemas tecnológicos, tanto de *hardware* como de *software*, proporcionando asistencia a través de diversos canales de comunicación. Su experiencia puede brindarnos detalles acerca de las consecuencias de una maniobra (cualquier actividad realizada sobre uno o varios elementos de la infraestructura que puede generar una indisponibilidad) o de un cambio que no fue comunicado correctamente.

6. Diseño metodológico

6.1. Metodología elegida y justificación

La metodología elegida para llevar a cabo este trabajo es investigación-acción [14]. Este método es un enfoque iterativo que se organiza en ciclos y requiere de una fuerte colaboración entre los investigadores y actores clave. Estos últimos son las personas que conocen el contexto del problema a resolver, y pueden brindar retroalimentación de valor en cada iteración.

Cada ciclo se compone por cinco etapas:

1. Diagnóstico: Identificación y comprensión del problema y el contexto organizacional, a menudo utilizando entrevistas a los actores clave, *focus groups* y análisis de datos.

2. Planificación de la acción: Desarrollo de un plan concreto para abordar el problema identificado.

3. Toma de acción: Implementación del plan, ya sean cambios directos en las operaciones o recomendaciones para su adopción presentadas a los actores clave.

4. Evaluación: Análisis de los resultados obtenidos evaluando si se logró resolver el problema o si es necesario rediseñar el plan.

5. Aprendizaje: Documentación de los hallazgos y generación de conocimiento práctico y teórico que beneficie tanto a la organización como a la comunidad de investigación.

Se eligió la metodología de investigación-acción porque el estudio se lleva a cabo en una empresa específica, donde es posible colaborar con actores clave. Su conocimiento y experiencia son esenciales tanto para comprender el problema con mayor profundidad como para validar el plan de acción y aportar retroalimentación valiosa. Además, el enfoque iterativo de esta metodología permite mejorar continuamente el plan, ajustándolo según la opinión de los

involucrados y la nueva información que se obtiene. De este modo, la solución propuesta no solo resulta viable en teoría, sino que también se adapta a las necesidades reales de la organización.

6.2.Plan de iteraciones

En este trabajo se llevarán a cabo dos ciclos de la metodología investigación-acción con el objetivo de abordar de manera iterativa y colaborativa el problema identificado sobre la gestión de cambios en servidores y el mapeo de dependencias.

En cada iteración se define y mejora un plan de acción para la futura implementación de la solución propuesta considerando la retroalimentación obtenida. Es importante destacar que la implementación en sí misma está fuera del alcance de este trabajo.

En el primer ciclo de investigación-acción, se realizará un diagnóstico a través de entrevistas estructuradas con los actores e informantes clave. Además, se investigarán herramientas disponibles en el mercado que puedan contribuir a resolver el problema, así como las herramientas actualmente utilizadas por la empresa. Este diagnóstico permitirá identificar las necesidades, limitaciones y dependencias específicas de la organización en relación con la gestión de cambios.

Posteriormente, con base en la información recopilada, se desarrollará un plan de acción preliminar que incluirá propuestas para optimizar la documentación de dependencias, mejorar la comunicación entre equipos y evaluar herramientas potenciales para la centralización de información. Este plan será presentado a los actores e informantes clave para su revisión y retroalimentación. La información obtenida durante esta etapa se documentará y analizará cuidadosamente para identificar las áreas de mejora.

El segundo ciclo se enfocará en perfeccionar y validar el plan de acción inicial, integrando las observaciones y sugerencias obtenidas en el primer ciclo. Se iniciará con una revisión detallada del plan de acción para incorporar ajustes que aborden las limitaciones o inquietudes identificadas. También se considerará la inclusión de nuevas herramientas o procesos según las recomendaciones de los actores clave.

7. Primer ciclo de investigación-acción

El objetivo de esta primera etapa es diagnosticar el estado actual de la infraestructura, identificar deficiencias y desarrollar un plan de acción preliminar para abordar las necesidades de gestión de cambios y mapeo de dependencias

7.1. Diagnóstico

7.1.1 Entrevistas a informantes clave

Con el objetivo de obtener información relevante para comprender a fondo las necesidades y problemas de la empresa, se realizaron entrevistas a los actores clave involucrados.

En primer lugar, se planificaron las preguntas de las entrevistas según el área de trabajo de cada actor clave. Para los actores con roles más técnicos (el líder de la plataforma técnica y el líder del equipo encargado de la administración de los servidores) se realizó una misma versión con el objetivo de comprender el proceso actual de cambios en los servidores y cómo la falta de una correcta gestión de dependencias afecta su trabajo diariamente. Por otro lado, la entrevista al líder del centro de atención a usuarios se estructuró con un foco diferente. En este caso, se escribieron preguntas enfocadas en evaluar la efectividad del sistema de notificación de maniobras, comprender el impacto operativo de estos eventos y conocer los procedimientos y la carga de trabajo asociada a incidentes imprevistos.

Una vez planificadas las entrevistas, se enviaron por correo junto con una introducción sobre el contexto de esta investigación, y se obtuvieron las respuestas requeridas detalladas en los ANEXOS 1, 2 y 3.

Luego de leer y analizar las respuestas obtenidas, surgieron nuevas preguntas y áreas de interés en las que se quería profundizar más, por lo que se optó por realizar una segunda ronda de entrevistas. En esta oportunidad, se utilizó el formato de reunión virtual, fomentando un ambiente abierto y colaborativo para que los informantes pudieran expresar sus opiniones y reflexiones sin

restricciones. Esta segunda ronda fue realizada únicamente a los actores con perfil más técnico ya que el objetivo era comprender en detalle cómo funciona la herramienta de monitoreo interna actual, definir los requerimientos funcionales que debía presentar la solución al problema de la gestión de dependencias y evaluar posibles alternativas. Las respuestas obtenidas se detallan en los ANEXOS 4 y 5.

Durante cada entrevista virtual, tomamos notas detalladas para asegurarnos de no perder información importante. Este enfoque nos permitió obtener una visión integral y precisa de las necesidades clave dentro de la empresa, lo cual fue fundamental para el desarrollo de un plan de acción que se ajuste al contexto y solucione los puntos de dolor de los actores. Además, la colaboración activa y el intercambio de ideas sobre posibles soluciones en tiempo real fue muy enriquecedor para ambas partes.

7.1.2 Desarrollo de las preguntas de investigación basados en las entrevistas.

A partir de las entrevistas realizadas, abordamos nuestras dos primeras preguntas de investigación relacionadas a las deficiencias en la gestión de cambios en servidores Windows, centrándonos especialmente en la visibilidad y coordinación de dependencias.

Problemas identificados (PI1)

- Ausencia de un mapeo confiable de dependencias entre los elementos de la infraestructura: No existe un registro claro y confiable de las relaciones entre los objetos, lo que dificulta una gestión eficiente.
- Falta de automatización en el árbol de dependencias: Actualmente, el árbol de dependencias se define de forma manual, lo que lo convierte en un proceso engorroso y poco frecuente, resultando rápidamente obsoleto.
- Información dispersa y no estandarizada: La información sobre las relaciones entre los objetos no está normalizada y se encuentra distribuida en varias fuentes y equipos, lo que complica su acceso y análisis.

- Falta de visibilidad del impacto de los cambios: Cuando se realiza un cambio en un objeto, no se tiene claridad sobre qué otros objetos pueden verse afectados.
- Falta de automatización en la comunicación de maniobras: Las notificaciones se envían manualmente vía mail, lo que genera demoras y posibles errores en el proceso de cambios.

Requisitos funcionales y técnicos (PI2)

- Automatización y visualización de dependencias: La solución debe integrar y mostrar las relaciones entre servidores, aplicaciones y bases de datos, facilitando la gestión de cambios y su impacto.
- Estandarización de monitoreo: Es necesario establecer un proceso que normalice y consolide los objetos involucrados en el monitoreo, garantizando coherencia y precisión en la información.
- Creación de un mapa de dependencias confiable: Aunque no sea completamente preciso, debe representar fielmente la infraestructura actual, proporcionando visibilidad y control.
- Integración sin impacto significativo: La solución debe incorporarse sin alterar drásticamente la forma en que se utiliza la herramienta actual, evitando problemas en la interfaz y minimizando la necesidad de capacitación adicional.
- Continuidad con la herramienta de monitoreo existente: Dado que forma parte de la cultura organizacional y está profundamente integrada en los procesos diarios, no se considera viable reemplazarla por una herramienta externa.

7.1.3 Relevamiento de la herramienta de monitoreo actual

La herramienta de monitoreo utilizada en la organización ha sido desarrollada y mantenida internamente durante más de 20 años, lo que refleja su sólida trayectoria dentro de la infraestructura tecnológica de la empresa. A lo largo de este tiempo, ha evolucionado y se ha adaptado a los avances tecnológicos, manteniendo un monitoreo confiable de toda la infraestructura. Durante este proceso, se han identificado diversos desafíos, los cuales fueron discutidos en la

entrevista con el líder del área de infraestructura. Hoy en día, la herramienta forma parte de la cultura organizacional de la empresa.

Los elementos principales de la herramienta son los objetos, los cuales representan los distintos componentes de la infraestructura que necesitan ser monitoreados. Cada objeto dentro del sistema de monitoreo tiene asignados los siguientes atributos: nombre, tipo, entorno, estado (afectado, disponible, maniobra, desactivado) y equipo responsable.

Los tipos de objetos definen qué tipo de componentes de la infraestructura de TI es monitoreado. Algunos de los tipos definidos son:

- **SER**: monitorea la disponibilidad de servidores.
- **DRV**: monitorea el espacio en los discos.
- **WEB**: monitorea la disponibilidad de sitios web.
- **SRV**: monitorea la ejecución de servicios de Windows.
- **CER**: monitorea el vencimiento de certificados.
- **OCO**: monitorea tareas programadas en un servidor.
- **BD**: monitorea la instancia SQL.

Los objetos se configuran para que avisen y/o alerten mediante parámetros definidos por el equipo responsable. Los avisos son recibidos mediante correo electrónico. En el caso de las alertas, además de correoo electrónico, se recibe una llamada telefónica por el equipo de operaciones.

Tanto en caso de alerta como de aviso, el estado del objeto es “AFECTADO”.

Los objetos SER, DRV, WEB, SRV, CER, OCO y BD están creados para que alerten. Los objetos DRV y CER además de alertar, pueden avisar.

Todos los objetos que pertenecen a una misma solución generalmente tienen el mismo “entorno” cuyo nombre representa a la solución a la que pertenecen. Sin embargo, existe una excepción con algunos servidores de base de datos. Estos servidores (que son objetos de tipo “SER”) contienen bases de datos de diferentes

soluciones y se definen en un entorno “BD”, independientemente de las soluciones a la que pertenezca.

La nomenclatura para los nombres de los objetos sigue el siguiente formato:

- **Servidor (SER)**: [NombreServidor]_(aviso|alarma)
- **Discos (DRV)**: [NombreServidor]*[C|NombreDisco]*(aviso|alarma)
- **Sitios web (WEB)**: [NombreServidor]_(aviso|alarma)_URL
- **Servicios Windows (SRV)**: [NombreServidor]_[NombreServicio], por ejemplo, DNS o algún otro programa.
- **Certificados (CER)**: [NombreServidor]_[NombreCertificado]
- **Tareas (OCO)**: [NombreServidor]_[NombreTarea]
- **Instancia SQL (BD)**: [NombreServidor]_bd

El sistema requiere la instalación de un agente que consume un *web service* en cada servidor donde se definen los objetos a monitorear, y se ejecuta a través de un servicio de Windows.

7.1.4 Árbol de dependencias.

Un árbol de dependencias en un sistema de monitoreo es una herramienta crucial para la gestión eficiente de la infraestructura tecnológica, ya que proporciona una visión clara y estructurada de cómo los distintos componentes están interconectados, mejorando la capacidad de respuesta ante incidentes y optimizando la planificación de mantenimiento y cambios.

El sistema de monitoreo actual dispone de una funcionalidad para la creación manual de un árbol de dependencias donde se pueden relacionar los diferentes componentes de las soluciones tecnológicas definidas, como servidores, servicios, aplicaciones, bases de datos, redes, y otros recursos críticos.

En primer lugar, se deben crear dos nuevos objetos sin acciones: un objeto sistema (de tipo SIS) y uno o varios objetos agrupadores (de tipo AGR). El objeto sistema hace de “padre” del árbol. El o los objetos agrupados, permiten brindar la representación de un balanceador o simplemente para agrupar dos objetos que tienen alta disponibilidad o, en el caso de base de datos, tengan un *cluster* definido. Estos objetos solo brindan una relación lógica y permiten ordenar y jerarquizar los objetos.

El árbol representa las dependencias entre los distintos objetos que ya están previamente definidos. Por ejemplo, un objeto de base de datos puede depender de la disponibilidad del servidor de base de datos. Las dependencias pueden ser unidireccionales (un componente depende de otro) o bidireccionales.

Permite a los equipos de monitoreo entender rápidamente las implicaciones de un fallo o cambio en cualquier parte de la infraestructura. Si un servicio falla, el árbol de dependencia ayuda a identificar qué otros componentes se verán afectados, lo que facilita una respuesta más rápida y precisa.

El árbol de dependencia suele presentarse en una interfaz visual, donde los nodos representan los componentes monitoreados y las conexiones entre ellos indican las dependencias.

Los cambios en el estado de un nodo (por ejemplo, un servicio que falla) se reflejan visualmente en el árbol con un color, proporcionando a los operadores una visión clara y accesible del estado de la infraestructura: gris indica disponibilidad, rojo indica indisponibilidad y azul muestra una maniobra en curso.

En la Figura 1 se puede visualizar un ejemplo de un mapa de dependencias de la herramienta de monitoreo interna.

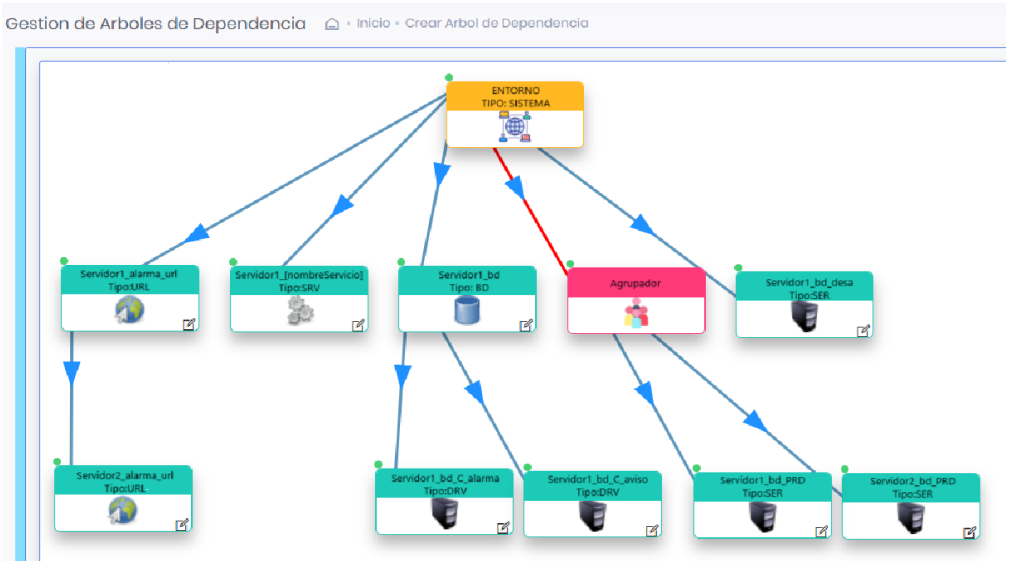

Figura 1. Diagrama del árbol de dependencias correspondiente a la herramienta de monitoreo actual.

7.1.5 Relevamiento del proceso de cambios

Existen diferentes procedimientos a seguir dependiendo del tipo de cambio que se necesite realizar. Estos cambios pueden ser:

- Cambio de urgencia: el cambio de urgencia es un cambio que se debe introducir tan pronto como sea posible debido a un incidente crítico en la solución o debido a una mejora crítica. Se considera crítica una situación que impida o pueda impedir la capacidad de brindar el servicio al cliente final.
- Cambio de emergencia: el cambio de emergencia es un cambio que se debe introducir tan pronto como sea posible debido a que la solución se encuentra inoperativa o en riesgo de quedar inoperativa.

Ante este tipo de cambio se realizan los siguientes pasos:

- Se informa inmediatamente de la urgencia a través de un canal de Teams a los equipos de: Operaciones, Infraestructura y jefes de equipos de desarrollo. Se comunica el problema y su impacto sin burocracia ni procesos formales previos.
- Se prioriza la solución rápida del problema y una vez resuelto se comunica que el sistema está nuevamente operativo a través de Teams.
- Se elabora un informe con la solución final y los cambios realizados. Se realiza una reunión de lecciones aprendidas para analizar: ¿Qué sucedió? ¿Por qué no se detectó antes?
  Con base en este análisis, se evalúa si es necesario tomar medidas como:

  - Implementar nuevas alarmas o monitoreos.
  - Mejorar la planificación y prevención de incidentes.
  - Ajustar procesos para evitar futuros problemas similares.

- Cambio estándar: el cambio estándar es un cambio pre-autorizado por el Comité de Gerentes, de bajo riesgo, que no genera indisponibilidad total.
- Cambio planificado: el cambio planificado es cambio que requiere de planificación y coordinación.

  Ante este tipo de cambio se realizan los siguientes pasos:

  - Se coordina con el equipo o los equipos involucrados y define el procedimiento de maniobra, fecha y hora de ejecución.
  - Se crea una maniobra en la herramienta de monitoreo actual, indicando: el nombre de la maniobra, descripción breve de la maniobra, persona responsable, la fecha y hora a realizarse y los objetos involucrados para la misma (se deben buscar y seleccionar dichos objetos).
  - Se envía mail al equipo de operaciones con una breve descripción de la maniobra, indicando los datos más relevantes, el link de la maniobra creada en la herramienta de monitoreo y el plan de la maniobra adjunto con las tareas y equipos responsables.

- El equipo de operaciones envía el mail a toda la unidad de TI.

7.1.6 Investigación de herramientas de monitoreo

En el mercado existen diversas herramientas de monitoreo de infraestructura que abordan el problema del mapeo de dependencias desde distintas perspectivas. Estas herramientas permiten realizar un análisis comparativo de las estrategias empleadas y sirven como referencia para la solución automatizada propuesta en este estudio.

Dado el amplio número de opciones disponibles, se presenta un análisis de tres herramientas destacadas, seleccionadas con base en la encuesta de Gartner sobre las mejores soluciones de monitoreo de infraestructura [15]. Las herramientas seleccionadas son: Zabbix, Naggios XI y SolarWinds. Para cada una de ellas se proporciona una breve descripción sobre cómo abordan la problemática del mapeo de dependencias y cómo implementan su visualización.

Zabbix

Zabbix es una plataforma de monitoreo de código abierto diseñada para supervisar redes, servidores, aplicaciones y servicios en infraestructuras de TI de cualquier escala [16]. Ofrece capacidades de recopilación de datos en tiempo real, alertas y visualización para garantizar el rendimiento y la disponibilidad de los sistemas.

Actualmente en Zabbix se pueden crear mapas de dependencias [17] pero es un proceso manual. Los propios usuarios tienen que manualmente crear el mapa indicando su configuración inicial, ingresar todos los elementos pertenecientes a la red (indicando tipo, nombre, etiquetas, etc) y por último crear los *links* entre los elementos, como muestran la figura 2 y 3.

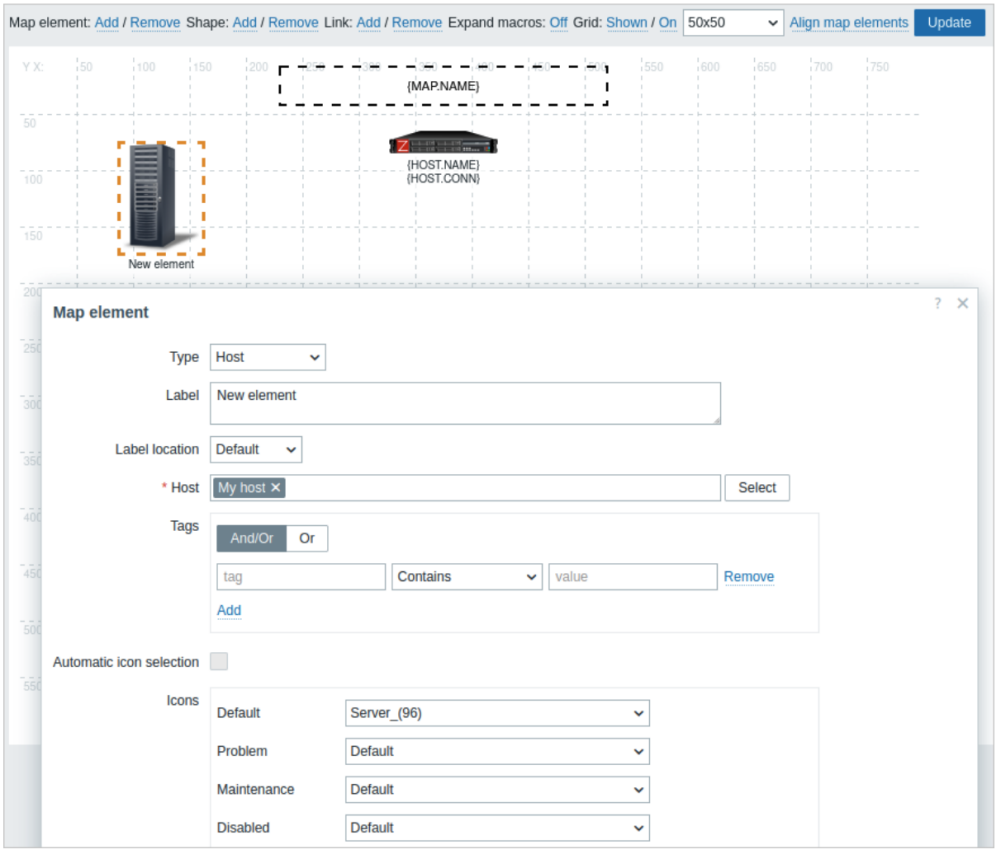

Figura 2. Ejemplo de creación de un elemento de tipo *Host* en Zabbix.

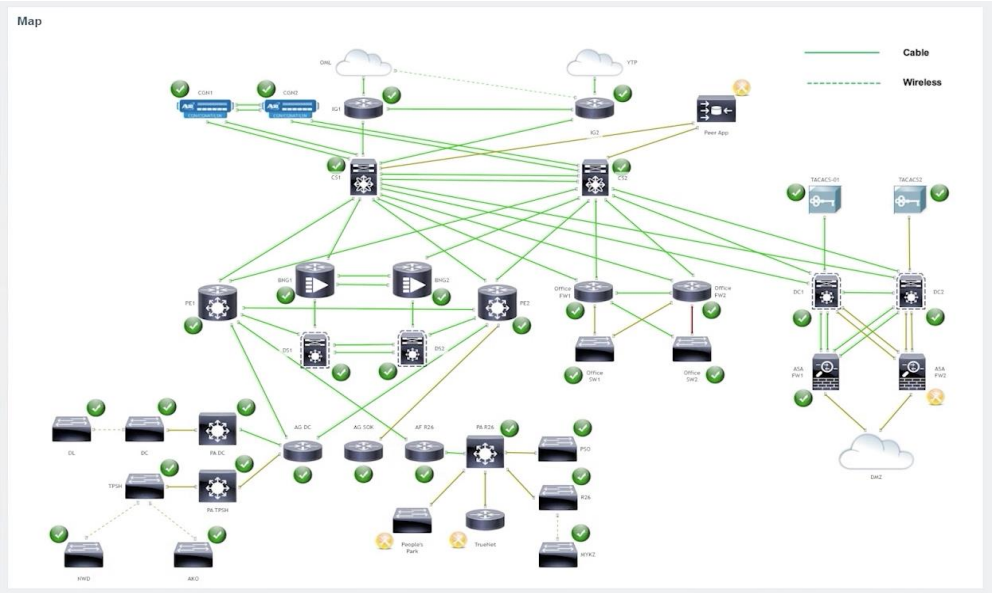

Figura 3. Ejemplo de un árbol de dependencia en Zabbix.

Afortunadamente, la herramienta proporciona una API que puede ser consumida para crear cada elemento del mapa, lo que habilita a desarrolladores a crear algoritmos para la creación del diagrama de forma automática, aunque dicha funcionalidad aún no esté soportada oficialmente por Zabbix. Un ejemplo es el repositorio AutoMapper creado por la consultora SOMONE [18], que contiene un algoritmo que puede crear el mapa en Zabbix automáticamente. Este algoritmo se alimenta de 3 conjuntos de datos: la configuración inicial del mapa, los hosts definidos en Zabbix y ciertas etiquetas extras definidas en cada host.

Las etiquetas que utiliza son las siguientes:

- am.host.type => tipo del host
- am.host.label => nombre del host
- am.link.connect_to => nombre del host al que está conectado
- am.link.label => nombre del link
- am.link.color => color del link
- am.link.draw_type => tipo de línea usado (continua, punteada, etc).

Si cada host posee estas etiquetas definidas, el algoritmo correrá sin inconvenientes y se creará el diagrama sin necesidad de crear elementos gráficos manualmente.

Nagios XI

Nagios XI es una plataforma de monitoreo empresarial de pago, desarrollada a partir del proyecto de código abierto Nagios Core [19]. Su objetivo es facilitar la integración de la herramienta en entornos corporativos, superando las limitaciones de Nagios Core, que requiere un alto nivel de conocimiento técnico y dominio de la línea de comandos para su uso efectivo. Para abordar este desafío, Nagios XI incorpora funcionalidades clave como

asistentes de configuración intuitivos, capacidades avanzadas para entornos multiinquilino y una interfaz gráfica personalizable [20], ofreciendo así una solución más accesible y eficiente.

Nagios XI incluye la visualización de un mapa de dependencias que muestra la relación entre los elementos de la red y se genera automáticamente a partir de relaciones “padre” definidas en la configuración de los objetos [21], como se visualiza en la Figura 4.

Cada host en la red puede tener asignado un host padre, lo que establece una jerarquía entre los dispositivos monitoreados. Esta relación permite graficar las conexiones lógicas entre los elementos de la infraestructura donde las líneas reflejan su estructura jerárquica.

Además, el mapa de dependencias también muestra el estado de cada host (si está activo o caído), lo que facilita la identificación de problemas y ayuda a determinar rápidamente su posible origen.

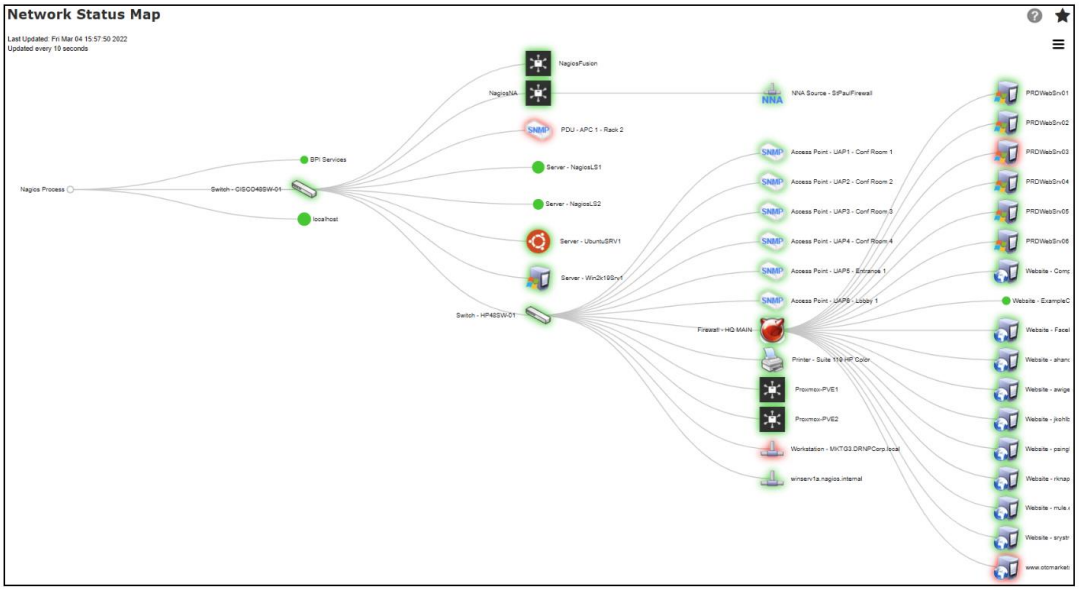

Figura 4. Mapa de dependencias en Nagios XI

Solarwinds

Solarwinds es una empresa que ofrece servicios *self-hosted* o SaaS de observabilidad y monitoreo, entre otras funcionalidades [22]. Ofrece una visibilidad unificada y completa para aplicaciones nativas en la nube, locales e híbridas, tanto personalizadas como comerciales, con el objetivo de garantizar niveles óptimos de servicio y la satisfacción de los usuarios en los servicios clave del negocio.

SolarWinds también proporciona visualizaciones gráficas de las relaciones entre las entidades monitoreadas, conocidas como "mapas inteligentes" [23]. Estos mapas, al igual que en Nagios XI, se generan automáticamente, ofreciendo una representación clara y dinámica de la infraestructura monitoreada, como se visualiza en la Figura 5. Para poder ver las relaciones entre cada elemento de la infraestructura en el gráfico se deben crear dependencias manualmente [24]. Para ello en primer lugar se selecciona el nodo “padre”, luego los “hijos” y por último un nombre que identifique esa dependencia. Una vez definidas todas las dependencias, se podrán observar todos los elementos jerarquizados en el mapa inteligente. A partir de este mapa, se podrá a su vez crear otro mapa customizado, agregando cambios o configuraciones de forma manual [25].

A diferencia de Nagios XI, SolarWinds ofrece la posibilidad de enriquecer el mapa inteligente al incluir datos sobre la latencia y la pérdida de paquetes en cada enlace entre los elementos monitoreados. Esta funcionalidad, denominada *Connection Quality Polling*, utiliza la técnica de *polling* para recopilar y mostrar esta información [26].

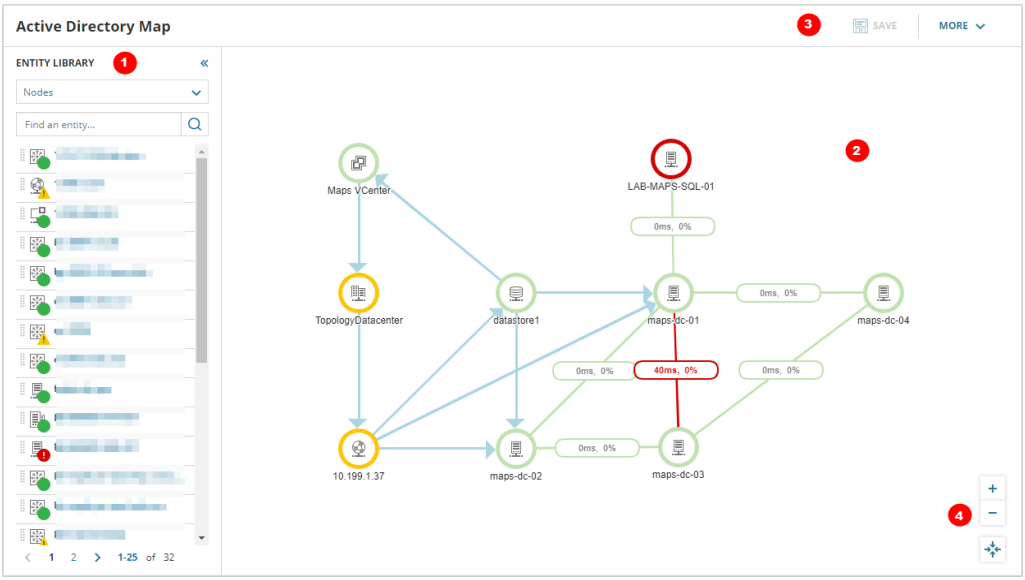

Figura 5. Mapa de dependencias en SolarWinds

7.2.Plan de acción

En esta sección se diseña un plan de acción que será presentado a los actores clave para obtener retroalimentación.

7.2.1 Propuesta de solución

A partir de los comentarios recibidos en la primera entrevista con los actores y la investigación de herramientas de monitoreo externas, se elabora el plan teniendo en cuenta los siguientes aspectos:

- La generación del árbol de dependencias debe ser automática.
- No se desea agregar nuevos campos durante la creación de un objeto en la herramienta de monitoreo actual.
- Cada objeto ya cuenta con los campos definidos: entorno, tipo, nombre, estado y equipo responsable.

- En la solución de estudio se administran únicamente objetos con los siguientes tipos:

    - SER: Monitorea la disponibilidad de servidores.

    - DRV: Monitorea el espacio en los discos.
    - WEB: Monitorea la disponibilidad de sitios web.
    - SRV: Monitorea la ejecución de servicios de Windows.
    - CER: Monitorea el vencimiento de certificados.
    - OCO: Monitorea tareas programadas en un servidor.
    - BD: Monitorea la instancia SQL.
- El sistema de nombres actual de los objetos tiene una estructura definida.
    - Servidor (SER): [NombreServidor]_(aviso|alarma)
    - Discos (DRV): [NombreServidor]*[C|NombreDisco]*(aviso|alarma)
    - Sitios web (WEB): [NombreServidor]_(aviso|alarma)_URL
    - Servicios Windows (SRV): [NombreServidor]_[NombreServicio], por ejemplo, DNS o algún otro programa.
    - Certificados (CER): [NombreServidor]_[NombreCertificado]
    - Tareas (OCO): [NombreServidor]_[NombreTarea]
    - Instancia SQL (BD): [NombreServidor]_bd
- No se busca cambiar el sistema de monitoreo, sino continuar utilizando la herramienta interna.
- Las herramientas externas investigadas utilizan etiquetas o relaciones "padre/hijo" para definir dependencias o jerarquías.
- Actualmente, las notificaciones sobre maniobras planificadas se realizan de forma manual.

Se propone:

1) **Planificación de una nueva funcionalidad dentro de la herramienta interna que, a partir de los campos entorno, tipo y nombre de los objetos monitoreados, infiera automáticamente las relaciones entre ellos.** Luego, a través de una API, se generarán los objetos gráficos correspondientes sin necesidad de intervención manual.

   Para la planificación de esta funcionalidad se deben tener en cuenta los siguientes aspectos:

   - Agrupación por entorno: Se creará un mapa específico para cada entorno definido.
   - Estructura del mapa:
     - Cada mapa incluirá un objeto raíz denominado "SISTEMA", que representará el entorno.
     - Si es necesario, se generarán objetos agrupadores para organizar servidores con nombres similares o relacionados. Por ejemplo, si existen servidores ser1, ser2, serDB1, serDB2, se crearán dos agrupadores: uno para los "ser" y otro para los "serDB".
     - Si un servidor no tiene otros con nombres similares, no se generará un agrupador y dependerá directamente del objeto "SISTEMA".
   - Relaciones entre objetos:
     - Los objetos de tipo "DRV", "WEB", "CER" y "SVC" dependerán de un objeto "SER".
     - La relación se establecerá utilizando la información del nombre del objeto que indica a qué servidor está asociado.

- Relaciones con BDD:
    - El equipo encargado de controlar las bases de datos, debe crear sus objetos con la siguiente nomenclatura:
        - [NombreServidorDB]_bd para el objeto de tipo BD que controla la disponibilidad de la instancia de SQL.
        - [NombreServidorDB]_[NombreDisco]_[alarma|aviso] para los objetos de tipo DRV
    - Todos los objetos que contengan en su nombre [NombreServidorDB] estarán relacionados con el objeto de tipo SER con ese nombre.
    - La mayoría de los servidores de base de datos son compartidos por varias soluciones y/o aplicaciones.
- Objetos huérfanos:
    - Aquellos objetos cuyo nombre no coincida parcialmente con el de un servidor dentro del mismo entorno, se mostrarán como elementos aislados en el mapa.

Este enfoque permitirá una generación automática y estructurada de mapas de dependencias, optimizando la visualización y gestión de los objetos en la infraestructura.

Con el fin de llevar la teoría a la práctica, se presenta un diagrama (Figura 6) de cómo se vería el árbol de dependencias si se ejecutara un algoritmo automatizado con las características planteadas, a partir de objetos reales que existen actualmente en la herramienta de monitoreo.

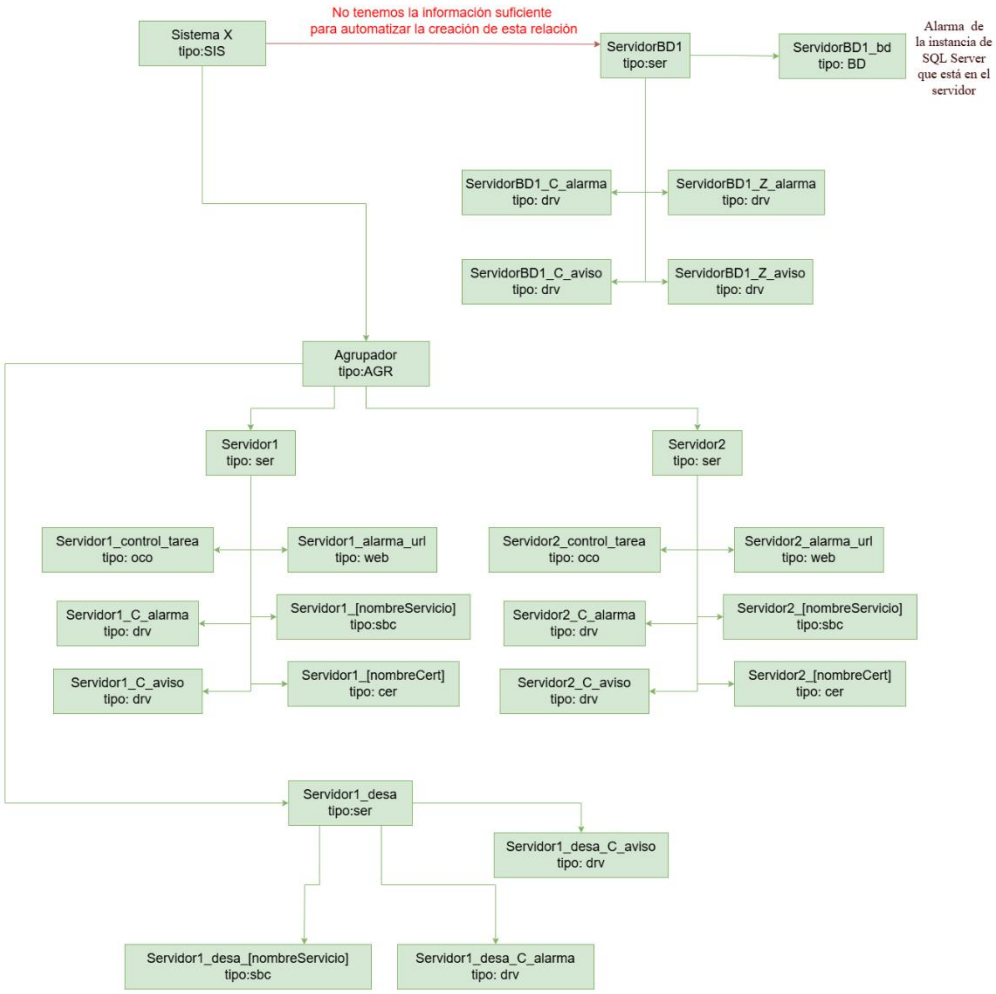

Figura 6. Diagrama del árbol de dependencias correspondiente al ciclo 1

2) Automatización del envío de notificaciones de maniobras

Actualmente, cuando se crea una maniobra, el equipo responsable debe notificar manualmente al equipo de operaciones mediante un correo electrónico. Se propone automatizar este proceso, de modo que, al registrar una maniobra en la herramienta interna de monitoreo, se genere y envíe automáticamente un correo con todos los detalles relevantes. Esto garantizará una comunicación más ágil y reducirá el riesgo de omisiones o demoras en la notificación, además de un cambio en la automatización del proceso de aviso de maniobras planificadas.

La implementación de estos dos cambios permitirá agilizar el proceso de gestión de cambios al:

- Automatizar las notificaciones de maniobras, eliminando la necesidad de envíos manuales y reduciendo demoras en la comunicación.

- Identificar de manera proactiva los objetos afectados ante una maniobra o indisponibilidad, permitiendo detectar servidores, bases de datos y aplicaciones impactadas por cambios en la infraestructura y mejorando la capacidad de respuesta ante incidentes.

- Comprender las relaciones de dependencia entre elementos, facilitando la evaluación de riesgos y la adopción de medidas preventivas para minimizar fallos.

- Brindar mayor visibilidad sobre la disponibilidad de los objetos a usuarios no técnicos, como el equipo de CAU y operaciones, quienes podrán consultar el mapa de dependencias para verificar la restauración de sistemas sin depender del equipo técnico responsable.

A su vez, la automatización del mapa de dependencias permitirá obtener beneficios adicionales a futuro, tales como:

- Identificación de servidores, bases de datos o aplicaciones “huérfanos”, sin relaciones con otros elementos, optimizando su documentación, uso y mantenimiento.

- Facilitación de simulaciones de impacto ante cambios en la infraestructura, como actualizaciones de sistemas operativos o migraciones de bases de datos, ayudando a prever y mitigar riesgos.

- Evaluación del impacto de la obsolescencia de un servidor o servicio, mediante la generación de un "*score* de obsolescencia" que indique el nivel de riesgo asociado a cada componente.

- Definición y gestión de maniobras directamente en la plataforma de monitoreo, mejorando el seguimiento y la coordinación de actividades.

7.3.Toma de acción

En esta etapa del ciclo de investigación-acción se realiza una intervención indirecta en donde se presenta el plan de acción al líder de plataforma técnica y al jefe del equipo que administra servidores y servicios colaborativos, con el fin de obtener *feedback* relevante para comenzar un nuevo ciclo.

En primer lugar, se les envió un correo con el documento del plan de acción (ver ANEXO 6) y luego se realizaron las entrevistas siguiendo el documento.

En las entrevistas recibimos sugerencias y recomendaciones para mejorar la solución propuesta (ver ANEXOS 7 y 8).

7.3.1 *Feedback* del líder de plataforma técnica.

Sobre el mapa de dependencias:

La propuesta planteada ha recibido comentarios positivos y aspectos a mejorar, destacándose varios puntos clave:

- Dinamismo de tipos de objetos: Si en el futuro surgen nuevos tipos de objetos, hay que modificar el algoritmo para entender cómo este nuevo tipo se relaciona con los otros tipos. Es importante tener en cuenta este aspecto para asegurar la flexibilidad del sistema a futuros cambios. Sugiere definir reglas que describan las relaciones entre los tipos de objeto, de manera que, si surge un nuevo tipo, simplemente se deban agregar nuevas reglas y no modificar el algoritmo en sí.
- Agrupadores por ambiente: Se sugiere crear agrupadores específicos para cada ambiente (desarrollo, test, producción).
- Ampliación de información:
    - Una propuesta interesante es la creación de un mecanismo en el que cualquier usuario pueda agregar más información al árbol sin modificar el algoritmo o los objetos en la herramienta. De esta forma se crea de manera colectiva una gran fuente de datos que enriquecerá el potencial del mapa de dependencias.

    - Esto solucionaría el problema de los servidores de base de datos, que actualmente tienen el entorno “BD”, y no se pueden asociar automáticamente al entorno de una solución. Esta información que no se puede inferir automáticamente, puede ser agregada a la nueva herramienta por un usuario más adelante.
    - Con este mecanismo se podría visualizar y consultar de forma rápida información necesaria antes de un cambio, antes de una compra, o por obsolescencia.
    - Algunos ejemplos serían:
        - Asociar la versión de sistema operativo a cada objeto servidor para obtener los servidores afectados ante el fin de soporte de una versión de sistema operativo.
        - Asociar la compra a cada solución tecnológica, para obtener los objetos afectados antes de que termine el contrato.
- Independencia de la herramienta de monitoreo interna:
    - Para el mapa de dependencia, el líder de la plataforma propone desarrollar una aplicación independiente a la herramienta de monitoreo. De esta manera, no se modificaría nada de la herramienta de monitoreo actual.
    - Esta nueva herramienta consumiría los datos de la herramienta de monitoreo actual (y también datos externos de usuarios) de una forma normalizada, y a través de un algoritmo y reglas definidas, se genera el árbol.
    - En el caso que la herramienta interna de monitoreo detecte alguna actualización de estado (por ej: una indisponibilidad) en alguno de sus objetos, se le enviará un aviso a la herramienta del mapa de dependencias, para que esta vuelva a renderizar el árbol.
    - En esta aplicación, los árboles de dependencia no serán estáticos, sino que se pueden generar cada vez que se necesite.
    - Propone utilizar una base de datos de grafos para la generación del árbol de dependencias. Por ej: Neo4j (https://neo4j.com/).

    - Sugiere que la interfaz gráfica de la nueva herramienta sea accesible directamente desde la herramienta actual. De esta manera, se garantizaría una integración fluida para los usuarios, permitiéndoles interactuar con la nueva funcionalidad sin necesidad de salir del entorno en el que ya trabajan.
- Persistencia y comparación de versiones: Es fundamental garantizar que los datos y registros se conserven como *snapshots* a lo largo del tiempo para poder realizar comparaciones entre versiones anteriores y actuales de la infraestructura.

Sobre la automatización de la notificación de maniobras:

A la propuesta del envío automático de un correo con todos los detalles relevantes, propone los siguientes puntos:

- En vez de enviar automáticamente el correo al equipo de operaciones para dar aviso de maniobras planificadas, generar un borrador del correo con la información de la maniobra a planificar para poder previsualizarlo y modificarlo antes del envío y enviarlo a los equipos afectados y/o interesados del conocimiento de la maniobra.
- Establecer un sistema de suscripción a objetos podría ser beneficioso para notificar automáticamente cualquier cambio o maniobra que impacte a esos objetos, mejorando la visibilidad y la capacidad de respuesta ante incidentes.

7.3.2 *Feedback* del jefe del equipo que administra servidores

Sobre el mapa de dependencias:

Para mejorar la comprensión del mapa de dependencias, se podría considerar integrar filtros avanzados y opciones de búsqueda. Esto permitiría a los usuarios buscar objetos por nombre, tipo, relación o entorno, optimizando la navegación en mapas de gran tamaño.

La propuesta menciona notificaciones, pero sería beneficioso expandir para que se integren con sistemas de gestión de incidentes o plataformas de monitoreo adicionales, como herramientas de alertas, para facilitar la respuesta inmediata ante un cambio o fallo en la infraestructura.

Sería provechoso desarrollar una funcionalidad de simulación que permita realizar análisis predictivos para evaluar el impacto de posibles cambios en la infraestructura (por ejemplo, actualizaciones de sistemas operativos, migraciones de bases de datos o cambios en la configuración de redes). Esto permitiría prever y mitigar problemas antes de que ocurran.

Además de la validación de relaciones, la herramienta debería ser capaz de identificar inconsistencias o posibles errores en la estructura de dependencias (por ejemplo, objetos huérfanos mal conectados o dependencias cíclicas) y notificar a los administradores para que puedan corregirlas antes de que causen problemas mayores.

Sobre la automatización de la notificación de maniobras:

Con respecto a la automatización de maniobras, se podría enviar un correo automáticamente con el aviso de maniobra planificada y los objetos involucrados a los equipos técnicos previamente definidos.

Además, se podría añadir un sistema que permita realizar un seguimiento de las notificaciones de maniobras, incluyendo un historial accesible para los usuarios. Esto no solo permitiría saber qué se notificó y cuándo, sino también verificar si se tomaron medidas correctivas en función de las notificaciones anteriores.

7.4.Evaluación

La retroalimentación recibida fue muy completa y de gran valor, aportando muchos puntos clave para mejorar. Sin embargo, si se intentara incorporar todas las recomendaciones en la próxima iteración del plan, el alcance sería demasiado amplio. Por eso, se decidió priorizar algunos aspectos, centrados en los siguientes objetivos para el próximo ciclo: crear una solución independiente, minimalista y flexible para

la visualización del mapa de dependencias, y mejorar el proceso de notificación de maniobras automatizando los avisos, asegurando que se pueda modificar su contenido cuando sea necesario.

Con respecto al mapa de dependencias, se priorizan las siguientes recomendaciones para aplicar en la siguiente iteración:

1. Independencia de la herramienta de generación del mapa de dependencias, ya que permite una integración más limpia y provee la libertad de utilizar nuevas herramientas tecnológicas para la implementación del algoritmo sin necesidad de modificar la herramienta de monitoreo actual. A su vez podría asignarse un equipo independiente para el comienzo de su implementación.

2. Dinamismo de tipos de objetos, porque ofrece la posibilidad de extender el algoritmo sin modificarlo al crear nuevos tipos o cambiar los existentes. El algoritmo se ejecutaría a partir de reglas de relación definidas en un archivo de configuración, separado de su implementación.

3. Agrupadores por ambiente. Optimiza la visualización segmentando automáticamente por ambiente (desarrollo, producción, etc), utilizando la nomenclatura existente de los servidores.

Con respecto a la automatización de notificación de maniobras se consideran las siguientes sugerencias a abordar en la siguiente iteración:

1. Generación de un borrador de correo previo al envío de la notificación de maniobra. Mejora la automatización sin perder control sobre el contenido de los correos que en múltiples ocasiones se debe agregar información extra y debe ser revisado.

2. Envío de correo automático de maniobras a los equipos técnicos. Mejora la comunicación y reduce el riesgo de omisiones en la gestión de cambios. Además, su implementación es de bajo esfuerzo ya que la herramienta de monitoreo actual ya contiene toda la información que será incluida en el correo.

El resto de las recomendaciones que no fueron mencionadas serán consideradas para próximos ciclos o trabajos futuros, ya que no se alinean con los objetivos del próximo ciclo.

7.5.Aprendizajes

Para concluir este primer ciclo, se presentan los aprendizajes obtenidos en diversas áreas clave. En particular, se reflexiona sobre el desarrollo del proceso de investigación-acción, el contexto en el que se llevó a cabo y la importancia de la participación activa de los involucrados. Además, se expone cómo los resultados de este estudio serán compartidos con la comunidad científica y otros interesados en la temática.

7.5.1 Reflexión sobre el proceso Investigación-acción

El uso de la metodología de investigación-acción en este estudio ha aportado estructura y claridad al proceso, proporcionando una guía ordenada para el desarrollo del trabajo. Sus ciclos iterativos han permitido profundizar progresivamente en la problemática, lo que ha facilitado la identificación de nuevos aspectos y limitaciones a medida que avanza la investigación. A su vez, esta metodología ha fomentado un intercambio constante con los líderes clave, quienes han aportado nuevas ideas entre cada fase del estudio, enriqueciendo así la solución propuesta.

Sin embargo, la metodología investigación-acción demanda un tiempo considerable, ya que cada ciclo requiere reflexión, ajuste y validación de los hallazgos antes de avanzar. Además, es un enfoque exigente, tanto en términos de planificación como de ejecución, pues requiere un compromiso continuo de todos los involucrados para garantizar su efectividad. A pesar de estas dificultades, su enfoque progresivo y participativo la convierte en una estrategia valiosa para abordar problemas complejos y mejorar soluciones de manera iterativa.

7.5.2 Análisis sobre el contexto

Durante la ejecución del primer ciclo de investigación-acción se identificaron condiciones del contexto que han influido en el desarrollo del estudio, tanto facilitando como limitando la implementación de cambios. A continuación, se identifican y analizan dichas condiciones indicando el aprendizaje asociado.

Condiciones organizacionales.

La empresa de estudio, la cual cuenta con años de trayectoria, valora profundamente su cultura y sus formas de trabajo establecidas. Esto hace que cualquier cambio en herramientas o procesos utilizados diariamente sea recibido con cautela, ya que la prioridad es minimizar la fricción para los trabajadores. La adopción de nuevas soluciones suele enfrentarse a resistencia, no por falta de interés en la innovación, sino porque el riesgo de afectar la operativa diaria pesa más que los beneficios percibidos a largo plazo. Esto se vio reflejado en una de las entrevistas en donde el líder de la plataforma afirmó que considerar otra herramienta de monitoreo no era una opción y que no quería modificar el proceso de creación de los objetos ya que era parte de la cultura.

Además, dentro de esta organización, el análisis detallado de procesos suele quedar relegado a un segundo plano frente a tareas urgentes. Aunque invertir tiempo en comprender y mejorar los flujos de trabajo actuales puede traer beneficios significativos en el mediano y largo plazo, en la práctica, este tipo de iniciativas son consideradas de baja prioridad. Por ejemplo, actualmente al crear una maniobra el responsable debe enviar un aviso de forma manual, aunque se podría automatizar sin grandes esfuerzos. Sin embargo, debido a la gran cantidad de tareas urgentes, hasta el momento no se había analizado el proceso completo e identificado esa posibilidad de mejora.

Es clave entender que la transformación digital no se trata solo de automatizar procesos o introducir tecnología, sino de optimizar todo el flujo de trabajo, aprovechando los recursos existentes y nuevos de manera estratégica. Un enfoque

integral permite lograr mayor eficiencia y maximizar los beneficios sin generar disrupciones innecesarias en la operativa actual.

Factores externos

Más allá del contexto interno, también hubo factores externos que influyeron en el desarrollo de la investigación. Aspectos como la disponibilidad de los colaboradores, especialmente debido a tiempos de licencia o cargas laborales, impactaron en la planificación y coordinación de entrevistas.

7.5.3 Valoración de la participación activa

La participación activa de los actores ha sido un elemento clave en este estudio, permitiendo una comprensión más profunda del contexto y una interacción más dinámica.

Durante el estudio, se experimentó con dos formas de comunicación: el intercambio escrito, a través de correos electrónicos, y la interacción directa mediante reuniones virtuales. Si bien la comunicación escrita permitió recopilar información de manera estructurada y asincrónica, notamos que tenía limitaciones, como respuestas más breves, falta de claridad en algunos puntos y menor oportunidad para el debate.

Por otro lado, las reuniones virtuales resultaron significativamente más beneficiosas. Estas permitieron una comprensión más profunda del contexto, la posibilidad de aclarar dudas en el momento y un intercambio más dinámico de ideas. Además, la interacción en tiempo real favoreció una mayor participación y compromiso por parte de los colaboradores, enriqueciendo el análisis y permitiendo descubrir aspectos que podrían haber pasado desapercibidos en una comunicación escrita.

7.5.4 Transferencia de conocimiento

Este documento es el resultado de un proyecto final de maestría en Gestión de Sistemas de Información y forma parte de un esfuerzo por contribuir al conocimiento en esta área. Su contenido estará disponible para la comunidad, permitiendo que otros profesionales e investigadores puedan beneficiarse de los hallazgos, reflexiones y aprendizajes aquí presentados. La difusión de este trabajo busca no solo documentar la experiencia, sino también servir como referencia para futuras investigaciones y aplicaciones en contextos similares.

8. Segundo ciclo de investigación-acción

El objetivo del segundo ciclo de investigación-acción se centra en refinar, ajustar o mejorar las intervenciones realizadas en el primer ciclo, basándose en los hallazgos y aprendizajes obtenidos hasta ese momento. A medida que se avanza en el proceso, el objetivo también puede ampliarse para abordar problemas adicionales o más profundos que hayan emergido durante la implementación del primer ciclo.

8.1. Diagnóstico

En el ciclo anterior los actores clave proporcionaron recomendaciones sobre diversos aspectos de mejora. Tras una evaluación detallada, se definieron cuáles de estas sugerencias serán consideradas o implementadas en esta nueva etapa.

Como resultado, el plan de acción será modificado para incluir las siguientes mejoras:

Sobre el mapa de dependencias:

- Independizar la herramienta de generación del mapa.
- Permitir dinamismo en los tipos de objetos.
- Implementar agrupadores por ambiente.

Sobre la automatización de la notificación de maniobras:

- Generar un borrador de correo antes del envío.
- Enviar notificaciones automáticas a los equipos técnicos.

Para llevar a cabo estas mejoras, es necesario realizar investigaciones específicas que orienten su implementación. En particular, se explorará el uso de bases de datos basadas en grafos para optimizar la representación del mapa de dependencias. También se investigará una forma de extraer las reglas de relación entre los objetos para mejorar la flexibilidad del algoritmo. Por último, se estudiarán estrategias para la diferenciación por ambientes, permitiendo una segmentación más eficiente.

8.1.1. Investigación de base de datos orientadas a grafos

Las bases de datos orientadas a grafos [27] son un tipo de sistema de almacenamiento que modela la información en forma de nodos y relaciones (aristas), en lugar de tablas y filas como en las bases de datos relacionales. Estas bases de datos están diseñadas para manejar eficientemente datos con múltiples conexiones, permitiendo representar información de manera más intuitiva y flexible. Un ejemplo de una base de datos de grafos es Neo4j [28].

En la tabla 1 se muestran las principales diferencias entre bases de datos de grafos y bases de datos relacionales.

Característica	**Base de datos relacional (PostgreSQL)**	**Base de datos de grafos (Neo4j)**
Estructura de datos	Tablas con filas y columnas	Nodos y relaciones
Forma de consultas	SQL (con JOINs) entre tablas	Cypher (recorrido de grafos)
Velocidad en consultas	Disminuye con consultas complejas con muchas relaciones	Se mantiene eficiente al recorrer solo los datos relevantes
Visualización de los datos	Tablas	Redes de nodos interconectados

Tabla 1. Diferencias entre bases de datos de grafos y bases de datos relacionales

Cuando se trabaja con relaciones complejas, las bases de datos relacionales pueden volverse ineficientes, ya que las consultas requieren múltiples operaciones de JOIN. En cambio, Neo4j permite realizar consultas recorriendo directamente

las conexiones entre nodos, lo que mejora el rendimiento. En particular Neo4j utiliza Cypher como lenguaje de consulta que es similar a SQL en muchos sentidos, excepto que SQL se refiere a elementos almacenados en una tabla, mientras que Cypher se refiere a elementos almacenados en un grafo.

Otro de los beneficios clave que tienen las bases de datos de grafos es la capacidad de visualizar los datos de manera gráfica, lo que facilita la interpretación de relaciones complejas. En bases de datos relacionales, los resultados se presentan en formato de tabla, sin una representación clara de las conexiones. En cambio, en bases de datos de grafos como Neo4j, las consultas pueden visualizarse en forma de redes de nodos interconectados, facilitando su análisis.

En resumen, Neo4j es una excelente opción para un mapa de dependencias debido a su capacidad para modelar relaciones complejas de manera natural, sus consultas eficientes y sus herramientas de visualización.

8.1.2. Motor de reglas

Un motor de reglas es un sistema de software que gestiona y aplica reglas para automatizar decisiones y comportamientos [29]. Incluye módulos para definir, gestionar, ejecutar reglas y procesar resultados, permitiendo la implementación de lógica empresarial compleja.

Un ejemplo de motor de reglas es Drools [30], un motor de código abierto basado en el lenguaje Java y proporcionado por JBoss. Este motor permite manejar la lógica de negocio de manera flexible y dinámica, sin necesidad de modificar el código fuente cada vez que cambian las reglas. Esto es crucial en sistemas complejos donde las condiciones y relaciones pueden evolucionar con el tiempo. Su uso se alinea con el Principio de Abierto/Cerrado de SOLID [31], ya que el código base permanece cerrado a modificaciones pero abierto a extensiones mediante la adición de nuevas reglas sin alterar la estructura existente. Esto mejora la mantenibilidad, facilita la escalabilidad y reduce el riesgo de introducir errores al modificar el sistema.

Por ejemplo, consideremos la clase InfraElement, que representa elementos de infraestructura con los atributos *type* (tipo de elemento) y *name* (nombre del elemento).

A continuación, se muestra una regla en Drools que establece que un objeto de tipo "DRV" debe depender de uno de tipo "SER" cuando ambos pertenecen al mismo entorno y el nombre del "DRV" contiene el nombre del "SER":

```
rule "Un objeto de tipo DRV debe depender de uno de tipo SER si tienen el mismo entorno y el nombre del DRV contiene el nombre del SER"

when

    $drv: InfraElement(type == "DRV")

    $ser: InfraElement(type == "SER", environment == $drv.environment, $drv.name contains $ser.name)

then

    DependencyService.createDependency($ser, $drv);

end
```

Este tipo de reglas permite automatizar la identificación de dependencias, asegurando una correcta estructuración y gestión de los elementos de infraestructura.

8.1.3. Diferenciación por ambientes

Agrupar los elementos según los ambientes (desarrollo, producción, etc.) ofrece diversos beneficios clave, como una mejor organización, una gestión más eficiente y una optimización en la coordinación de los cambios en los servidores. Afortunadamente, los servidores incluidos en este estudio ya contienen información sobre el ambiente al que pertenecen en sus nombres.

Los nombres de los servidores siguen la siguiente estructura: “[prefijo que indica si es físico o virtual][tecnología o aplicación instalada][ambiente][número diferenciador]”, lo que facilita la extracción del ambiente al que pertenecen para su correcta clasificación.

8.2.Plan de acción

En esta sección se revisa y actualiza la propuesta de solución presentada en el ciclo anterior priorizando las sugerencias planteadas por los actores clave con mayor impacto en esta etapa.

8.2.1.Propuesta de solución

Considerando los siguientes aspectos:

- La generación del árbol de dependencias debe ser completamente automática e independiente del sistema de monitoreo actual.
- Otras herramientas utilizan etiquetas o relaciones "padre/hijo" para definir dependencias o jerarquías.
- No se desea agregar nuevos campos durante la creación de un objeto en la herramienta interna.
- Cada objeto ya cuenta con los campos definidos: entorno, tipo, nombre, estado y equipo responsable.
- En la solución de estudio se administran únicamente objetos con los siguientes tipos:
    - SER: Monitorea la disponibilidad de servidores.
    - DRV: Monitorea el espacio en los discos.
    - WEB: Monitorea la disponibilidad de sitios web.
    - SRV: Monitorea la ejecución de servicios de Windows.
    - CER: Monitorea el vencimiento de certificados.
    - BD: Monitorea la instancia SQL.

- Si se crea un nuevo tipo de objeto en el futuro, el algoritmo de generación del árbol de dependencias no debería verse afectado. El algoritmo debe estar abierto a la extensión y cerrado a la modificación.

- El sistema de nombres actual de los objetos tiene una estructura definida.

    - Servidor (SER): [NombreServidor]_(aviso|alarma)

    - Discos (DRV): [NombreServidor]*[C|NombreDisco]*(aviso|alarma)

    - Sitios web (WEB): [NombreServidor]_(aviso|alarma)_URL

    - Servicios Windows (SRV): [NombreServidor]_[NombreServicio], por ejemplo, DNS o algún otro programa.

    - Certificados (CER): [NombreServidor]_[NombreCertificado]

    - Tareas (OCO): [NombreServidor]_[NombreTarea]

    - Instancia SQL (BD): [NombreServidor]_bd

- El [NombreServidor] tiene la estructura [prefijo para diferenciar si es físico o virtual][tecnología o aplicación instalada][ambiente][número diferenciador], lo que permite la diferenciación por ambientes.

- Actualmente, las notificaciones sobre maniobras planificadas se realizan de forma manual.

Se propone:

1. **Implementación de un nuevo algoritmo de generación de árboles de dependencias a partir de datos normalizados, independiente a la herramienta de monitoreo actual. A partir de la información de los objetos obtenida en esta herramienta, los datos se normalizan y el algoritmo infiere automáticamente las relaciones entre ellos. A pesar que el algoritmo es independiente a la herramienta interna de**

monitoreo, el árbol se podrá visualizar en ella para no generar fricción en la usabilidad habitual de la plataforma.

- Agrupación por entorno: Se creará un mapa específico para cada entorno definido. Si existe más de un ambiente (desarrollo, producción, etc) para un mismo entorno, se creará un mapa para cada uno de los ambientes por separado.

- Estructura del mapa:

    - Cada mapa incluirá un nodo raíz de tipo "SISTEMA", que representará el entorno.

    - Si es necesario, se generarán nodos de tipo “AGRUPADOR” para organizar servidores con nombres similares o relacionados. Por ejemplo, si existen servidores ser1, ser2, serDB1, serDB2, se crearán dos agrupadores: uno para los "ser" y otro para los "serDB".

- Relaciones entre objetos:

    - Se utilizará la base de datos de grafos Neo4j para representar las relaciones entre los objetos.

    - Se creará una estructura de reglas de dependencia que definen cómo se relacionan dos tipos de objetos. Por cada tipo de objeto se deberá definir una regla que indique las condiciones que debe tener otro objeto para ser su padre. Por ejemplo, existirá una regla que indique que un objeto tipo “DRV” tendrá como padre a un objeto de tipo “SER” cuyo nombre coincida con el comienzo del suyo. De esta manera si se crea un nuevo tipo de objeto en el futuro, sólo se deberá definir una nueva regla, y el algoritmo podrá relacionarlo correctamente con los otros objetos. Por defecto se crearán las siguientes reglas:

        - Los objetos de tipo “SISTEMA” no tienen un nodo padre.

        - Los objetos de tipo “SER” pueden tener como padre al objeto “SISTEMA” o a un “AGRUPADOR”, dependiendo si existen más objetos de tipo “SER” con nombre similar (por ejemplo: [nombreServidor]1, [nombreServidor]2, [nombreServidor]3).

        - Todos los objetos de tipo “AGRUPADOR” tendrán como padre al objeto de tipo “SISTEMA”.

        - Todos los objetos de los tipos "DRV", "WEB", "CER", "SVC", “OCO” y “DB” tendrán como padre al objeto de tipo "SER" cuyo nombre coincida con el comienzo del suyo.

    - Para los servidores de base de datos (objetos de tipo “SER” con un hijo de tipo “BD”) no existirá una regla definida por defecto, ya que un servidor de base de datos puede estar relacionado con muchas soluciones diferentes, por lo que no tendrá un padre único. Además, esta información no se puede inferir automáticamente por el momento ya que su entorno siempre es “BD” y su nombre no tiene una estructura que nos brinde la información suficiente. Es por esto que surge la necesidad de poder crear relaciones directas entre objetos. Para esto se define una regla de excepción, indicando que el objeto X tiene de padre al objeto Y.

- Nomenclatura de objetos BD:

    - El equipo encargado de controlar las bases de datos, debe crear sus objetos con la siguiente nomenclatura:

        - [NombreServidorDB]_bd para el objeto de tipo BD que controla la disponibilidad de la instancia de SQL.
        - [NombreServidorDB]_[NombreDisco]_[alarma|aviso] para los objetos de tipo DRV
    - Todos los objetos que contengan en su nombre [NombreServidorDB] estarán relacionados con el objeto de tipo SER con ese nombre.

- Objetos huérfanos:
    - Aquellos objetos cuyo nombre no coincida parcialmente con el de un servidor dentro del mismo entorno, se mostrarán como elementos aislados en el mapa.

Este enfoque permitirá una generación automática y estructurada de mapas de dependencias, optimizando la visualización y gestión de los objetos en la infraestructura.

Con el fin de llevar la teoría a la práctica, se presenta un diagrama (Figura 7) de cómo se vería el árbol de dependencias si se ejecutara un algoritmo automatizado con las características planteadas, a partir de objetos reales que existen actualmente en la herramienta de monitoreo.

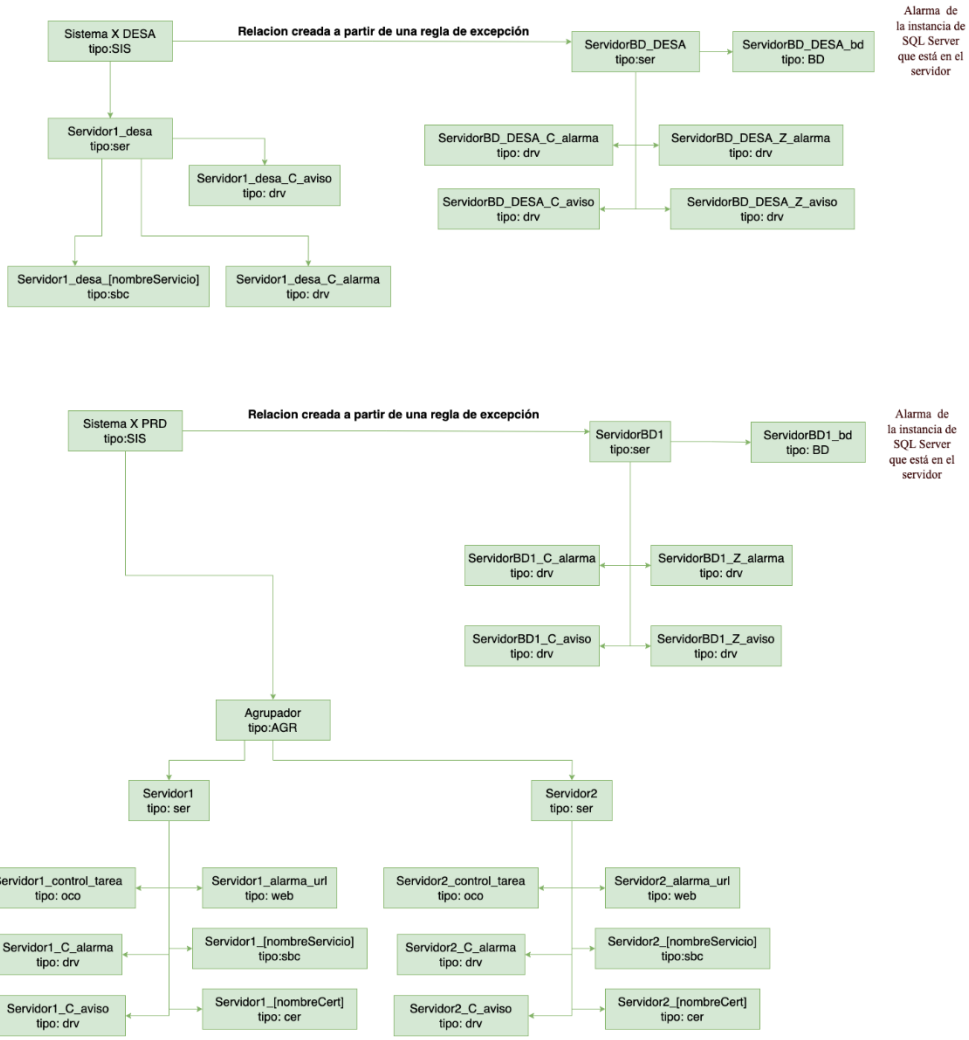

Figura 7. Diagrama del árbol de dependencias correspondiente al ciclo 2

2. Mejora en el proceso de maniobras planificadas

Actualmente, cuando se crea una maniobra, el equipo responsable debe notificar manualmente al equipo de operaciones mediante un correo electrónico. Se propone:

- Generar automáticamente un borrador del correo de aviso de maniobra con la siguiente información:
  - *Raised* (quién solicita el cambio),
  - *Reason* (por qué se realiza),
  - *Return* (qué se espera del cambio),

        - *Risk* (cuáles son los riesgos),
        - *Resource* (qué recursos se necesitan),
        - *Responsible* (quién es el responsable), y
        - *Relationship* (relación con otros cambios).

      De esta manera se podrá revisar o modificar su contenido y luego se envía automáticamente desde la plataforma.

    - Envío de un correo automático a los equipos técnicos involucrados en la maniobra con la información de todos los objetos que pueden verse afectados.

La implementación de estos dos cambios permitirá agilizar el proceso de gestión de cambios al:

- Automatizar las notificaciones de maniobras, eliminando la necesidad de envíos manuales y reduciendo demoras en la comunicación.

- Identificar de manera proactiva los objetos afectados ante una maniobra o indisponibilidad, permitiendo detectar servidores, bases de datos y aplicaciones impactadas por cambios en la infraestructura y mejorando la capacidad de respuesta ante incidentes.

- Comprender las relaciones de dependencia entre elementos, facilitando la evaluación de riesgos y la adopción de medidas preventivas para minimizar fallos.

- Brindar mayor visibilidad sobre la disponibilidad de los objetos a usuarios no técnicos, como el equipo de CAU y operaciones, quienes podrán consultar el mapa de dependencias para verificar la restauración de sistemas sin depender del equipo técnico responsable.

8.3.Toma de acción

En esta etapa del ciclo investigación-acción, se presentó el plan de acción en formato archivo de texto a los actores clave para que pudieran analizarlo de manera asincrónica (ver ANEXO 9). Al finalizar, cada uno de los actores nos envió su retroalimentación por correo (ver ANEXOS 10 y 11). Para esta segunda iteración consideramos que no era necesario involucrar al líder del centro de atención a usuarios ya que nuestra solución propuesta se basa principalmente en la herramienta actual de monitoreo y esta persona no está familiarizada con el desarrollo dicha solución. Sin embargo, al finalizar el proceso de retroalimentación con los líderes técnicos, se le presentará un resumen simplificado de la solución propuesta y cómo impactaría sus labores diarias en caso de ser implementada en un futuro.

8.3.1.Feedback del líder de la plataforma técnica

Sus observaciones fueron constructivas y aportaron claridad en varios aspectos sin implicar cambios significativos en el plan de acción original. A continuación, se resumen los principales puntos de su retroalimentación:

- Resaltó ideas con las que está de acuerdo y dejó comentarios positivos, destacando los puntos fuertes del enfoque propuesto. No realizó ninguna crítica que implique un cambio significativo en el plan de acción original.
- Ejemplificó algunos aspectos clave para aportar mayor claridad sobre su posible implementación.
- Sugirió la incorporación de un nodo “RAÍZ” que agrupe todos los nodos “SISTEMA”. Esto permitiría estructurar un único árbol en lugar de uno por cada entorno o sistema, alineando mejor el modelo con la teoría de grafos.
- Redactó y completó conceptos que estaban incompletos, como “objetos huérfanos” y “maniobra”, asegurando mayor precisión en la definición de términos clave.
    - Objetos huérfanos: “Son aquellos nodos sin predecesores ni sucesores, o sea no tienen aristas entrantes ni salientes. Identificarlos permitirá

detectar fallos o desviaciones del estándar definido para la generación automática del árbol de dependencias, facilitando la corrección de la nomenclatura. Es decir, su propietario podrá ajustar el nombre correspondiente, lo que permitirá iterar nuevamente el proceso y obtener una nueva versión del árbol con información más completa y precisa"

- Maniobra: "Se define como maniobra cualquier actividad, ya sea proactiva o correctiva, realizada sobre uno o varios objetos y que pueda generar una indisponibilidad temporal (maniobra disruptiva) o el riesgo de que esto ocurra (maniobra no disruptiva). Por ello, al trabajar con el árbol de dependencias, si nos enfocamos en los objetos involucrados, es altamente factible inferir qué sistemas y equipos se verán afectados mediante algoritmos basados en teoría de grafos."

- Priorizó las conclusiones en base a los aspectos que considera más importantes, resaltando los hallazgos más críticos para la efectividad del estudio. En particular, los dos beneficios que él considera más importantes son:

  - Identificar de manera proactiva los objetos afectados ante una maniobra o indisponibilidad, permitiendo detectar servidores, bases de datos y aplicaciones impactadas por cambios en la infraestructura y mejorando la capacidad de respuesta ante incidentes.

  - Comprender las relaciones de dependencia entre elementos, facilitando la evaluación de riesgos y la adopción de medidas preventivas para minimizar fallos.

8.3.2 Feedback del jefe del equipo que administra servidores

Sus comentarios fueron positivos y constructivos, aportando ideas clave para mejorar la gestión y adopción del sistema. A continuación, se resumen los principales puntos de su retroalimentación:

- Resaltó aspectos positivos del estudio, destacando los puntos fuertes y el valor que aporta la solución propuesta.

- Planteó la importancia de definir un plan para garantizar que la información relacionada con el sistema se mantenga actualizada y centralizada, asegurando así su confiabilidad y utilidad a largo plazo.

- Sugirió incluir una fase de capacitación sobre la herramienta, con el objetivo de que los usuarios estén familiarizados con el nuevo sistema antes de su implementación completa, facilitando así su adopción y uso eficiente.

8.4.Evaluación

8.4.1 Evaluación de recomendaciones obtenidas en las entrevistas

Tras analizar las recomendaciones recibidas, se decidió enfocar los esfuerzos en dos mejoras clave: la incorporación del nodo "RAÍZ" y la definición más precisa de los conceptos de "objeto huérfano" y "maniobra".

La incorporación de un nodo "RAÍZ" que agrupe todos los nodos "SISTEMA" permitirá estructurar un único árbol, lo que simplificará la implementación del algoritmo y alineará mejor el modelo con la teoría de grafos.

A su vez, incluir definiciones claras para “objeto huérfano” y “maniobra” dentro del plan de acción ayudará a evitar ambigüedades, mejorar la comunicación y ejecución, y garantizar mayor consistencia en el proceso.

Si bien toda la retroalimentación es valiosa, se han priorizado las recomendaciones que generan mayor impacto en esta fase del proyecto sin ampliar demasiado el alcance, dado que este es el último ciclo de trabajo. Las demás quedarán como oportunidades de mejora para el futuro.

8.4.2 Evaluación de la conformidad de los actores clave

Los actores involucrados expresaron conformidad con la solución propuesta y el plan de acción para su implementación. Se evidenció un mayor

entusiasmo y compromiso en sus devoluciones, con recomendaciones más detalladas y constructivas.

Durante el primer ciclo de revisión, el feedback fue más crítico y recomendaron cambios significativos para el plan de acción definido. Sin embargo, en el segundo ciclo, las observaciones se centraron principalmente en ajustes menores, reflejando una mayor alineación con la propuesta final.

En general, los actores clave consideran que el plan de acción presentado es una base sólida para optimizar el proceso de gestión de cambios. Reconocen claramente sus beneficios y están entusiasmados con su posible implementación futura.

8.5. Aprendizajes

Al concluir el segundo ciclo, se identificaron varios aprendizajes.

En primer lugar, fue muy positivo recibir sugerencias sobre herramientas que los propios actores ya han investigado. Al ser expertos en sus áreas, conocen mejor las limitaciones y preferencias tecnológicas, por lo que escuchar sus aportes resulta clave. Un ejemplo de esto fue la recomendación de Neo4j por parte del líder de la plataforma técnica.

Si bien las reuniones presenciales o por Zoom son valiosas, la disponibilidad de los actores clave no siempre lo permite. Incorporar opciones asincrónicas facilitó la recopilación de feedback sin depender de la coordinación de horarios.

La metodología investigación-acción permitió una colaboración continua con los actores clave, lo que ayudó a comprender mejor el problema y definir una solución más acotada, maximizando sus beneficios y definiendo una base para futuras mejoras.

Incluir varios actores desde el inicio fue útil para obtener una visión amplia del problema. Sin embargo, a medida que se avanzó en la definición de la solución, se identificó que no todos los actores necesitaban seguir participando activamente,

especialmente aquellos cuya área no estaba directamente relacionada con la implementación final.

9. Análisis y discusión de resultados

En esta sección se presenta el plan de acción final de este proyecto, diseñado con un enfoque centrado en el problema identificado y optimizado a partir de las recomendaciones brindadas por los actores clave. Este plan integra las mejoras y ajustes sugeridos durante el proceso de revisión, asegurando una solución alineada con las necesidades del equipo y los objetivos del proyecto.

Para definir el plan de acción consideramos los siguientes aspectos:

- La generación del árbol de dependencias debe ser completamente automática e independiente del sistema de monitoreo actual.
- Otras herramientas utilizan etiquetas o relaciones "padre/hijo" para definir dependencias o jerarquías.
- No se desea agregar nuevos campos durante la creación de un objeto en la herramienta interna.
- Cada objeto ya cuenta con los campos definidos: entorno, tipo, nombre, estado y equipo responsable.
- En la solución de estudio se administran únicamente objetos con los siguientes tipos:
    - SER: Monitorea la disponibilidad de servidores.
    - DRV: Monitorea el espacio en los discos.
    - WEB: Monitorea la disponibilidad de sitios web.
    - SRV: Monitorea la ejecución de servicios de Windows.
    - CER: Monitorea el vencimiento de certificados.
    - BD: Monitorea la instancia SQL.
- Si se crea un nuevo tipo de objeto en el futuro, el algoritmo de generación del árbol de dependencias no debería verse afectado. El algoritmo debe estar abierto a la extensión y cerrado a la modificación.
- El sistema de nombres actual de los objetos tiene una estructura definida.
    - Servidor (SER): [NombreServidor]_(aviso|alarma)
    - Discos(DRV): [NombreServidor]*[C|NombreDisco]*(aviso|alarma)
    - Sitios web (WEB): [NombreServidor]_(aviso|alarma)_URL

        - Servicios Windows (SRV): [NombreServidor]_[NombreServicio], por ejemplo, DNS o algún otro programa.
        - Certificados (CER): [NombreServidor]_[NombreCertificado]
        - Tareas (OCO): [NombreServidor]_[NombreTarea]
        - Instancia SQL (BD): [NombreServidor]_bd
- El [NombreServidor] tiene la estructura [prefijo para diferenciar si es físico o virtual][tecnología o aplicación instalada][ambiente][número diferenciador], lo que permite la diferenciación por ambientes.
- Actualmente, las notificaciones sobre maniobras planificadas se realizan de forma manual.

9.1. Plan de acción

Proponemos el siguiente plan de acción teniendo en cuenta los objetivos específicos planteados y las recomendaciones recibidas por los diferentes actores elegidos:

1. **Implementación de un nuevo algoritmo de generación de árboles de dependencias a partir de datos normalizados, independiente a la herramienta de monitoreo actual**.

   A partir de la información de los objetos obtenida en esta herramienta, los datos se normalizan y el algoritmo infiere automáticamente las relaciones entre ellos. A pesar que el algoritmo es independiente a la herramienta interna de monitoreo, el árbol se podrá visualizar en ella para no generar fricción en la usabilidad habitual de la plataforma.

   - Estructura del mapa:
     - La raíz del árbol será un nodo de tipo “ÁRBOL”, que tendrá como hijos a todos los otros objetos.
     - Por cada entorno diferente, existirá un nodo de tipo “SISTEMA”, que será hijo del objeto “ÁRBOL”. Por ejemplo, si se cuenta con objetos de los entornos EN1, EN2 y EN3, existirán 3 nodos “SISTEMA” como hijos de “ÁRBOL”. Si existe más de un

ambiente (desarrollo, producción, etc) para un mismo entorno, se creará un nodo “SISTEMA” para cada uno de los ambientes por separado.

- Si es necesario, se generarán nodos de tipo “AGRUPADOR” para organizar servidores con nombres similares o relacionados. Por ejemplo, si existen servidores ser1, ser2, serDB1, serDB2, se crearán dos agrupadores: uno para los "ser" y otro para los "serDB".

- Relaciones entre objetos:
  - Se utilizará la base de datos de grafos Neo4j para representar las relaciones entre los objetos.
  - Se creará una estructura de reglas de dependencia que definen cómo se relacionan dos tipos de objetos. Por cada tipo de objeto se deberá definir una regla que indique las condiciones que debe tener otro objeto para ser su padre. Por ejemplo, existirá una regla que indique que un objeto tipo “DRV” tendrá como padre a un objeto de tipo “SER” cuyo nombre coincida con el comienzo del suyo. De esta manera si se crea un nuevo tipo de objeto en el futuro, sólo se deberá definir una nueva regla, y el algoritmo podrá relacionarlo correctamente con los otros objetos. Por defecto se crearán las siguientes reglas:
    - Los objetos de tipo “ARBOL” no tienen un nodo padre.
    - Los objetos de tipo “SISTEMA” tienen a “ARBOL” como padre.
    - Los objetos de tipo “SER” pueden tener como padre al objeto “SISTEMA” o a un “AGRUPADOR”, dependiendo si existen más objetos de tipo “SER” con nombre similar (por ejemplo: [nombreServidor]1, [nombreServidor]2, [nombreServidor]3).
    - Todos los objetos de tipo “AGRUPADOR” tendrán como padre al objeto de tipo “SISTEMA”.

        - Todos los objetos de los tipos "DRV", "WEB", "CER", "SVC", "OCO" y "DB" tendrán como padre al objeto de tipo "SER" cuyo nombre coincida con el comienzo del suyo.
    - Para los servidores de base de datos (objetos de tipo "SER" con un hijo de tipo "BD") no existirá una regla definida por defecto, ya que un servidor de base de datos puede estar relacionado con muchas soluciones diferentes, por lo que no tendrá un padre único. Además, esta información no se puede inferir automáticamente por el momento ya que su entorno siempre es "BD" y su nombre no tiene una estructura que nos brinde la información suficiente. Es por esto que surge la necesidad de poder crear relaciones directas entre objetos. Para esto se define una regla de excepción, indicando que el objeto X tiene de padre al objeto Y.
- Nomenclatura de objetos BD:
    - El equipo encargado de controlar las bases de datos, debe crear sus objetos con la siguiente nomenclatura:
        - [NombreServidorDB]_bd para el objeto de tipo BD que controla la disponibilidad de la instancia de SQL.
        - [NombreServidorDB]_[NombreDisco]_[alarma|aviso] para los objetos de tipo DRV
    - Todos los objetos que contengan en su nombre [NombreServidorDB] estarán relacionados con el objeto de tipo SER con ese nombre.
- Objetos huérfanos
    - Aquellos objetos cuyo nombre no coincida parcialmente con el de un servidor dentro del mismo entorno, se mostrarán como elementos aislados en el mapa.

      Los objetos huérfanos son aquellos nodos sin predecesores ni sucesores, o sea no tienen aristas entrantes ni salientes. Identificarlos permitirá detectar fallos o desviaciones del estándar definido para la generación automática del árbol de dependencias,

facilitando la corrección de la nomenclatura. Es decir, su propietario podrá ajustar el nombre correspondiente, lo que permitirá iterar nuevamente el proceso y obtener una nueva versión del árbol con información más completa y precisa.

Este enfoque permitirá una generación automática y estructurada de mapas de dependencias, optimizando la visualización y gestión de los objetos en la infraestructura.

Con el fin de llevar la teoría a la práctica, se presenta un diagrama (Figura 8) de cómo se vería el árbol de dependencias si se ejecutara un algoritmo automatizado con las características planteadas, a partir de objetos reales que existen actualmente en la herramienta de monitoreo.

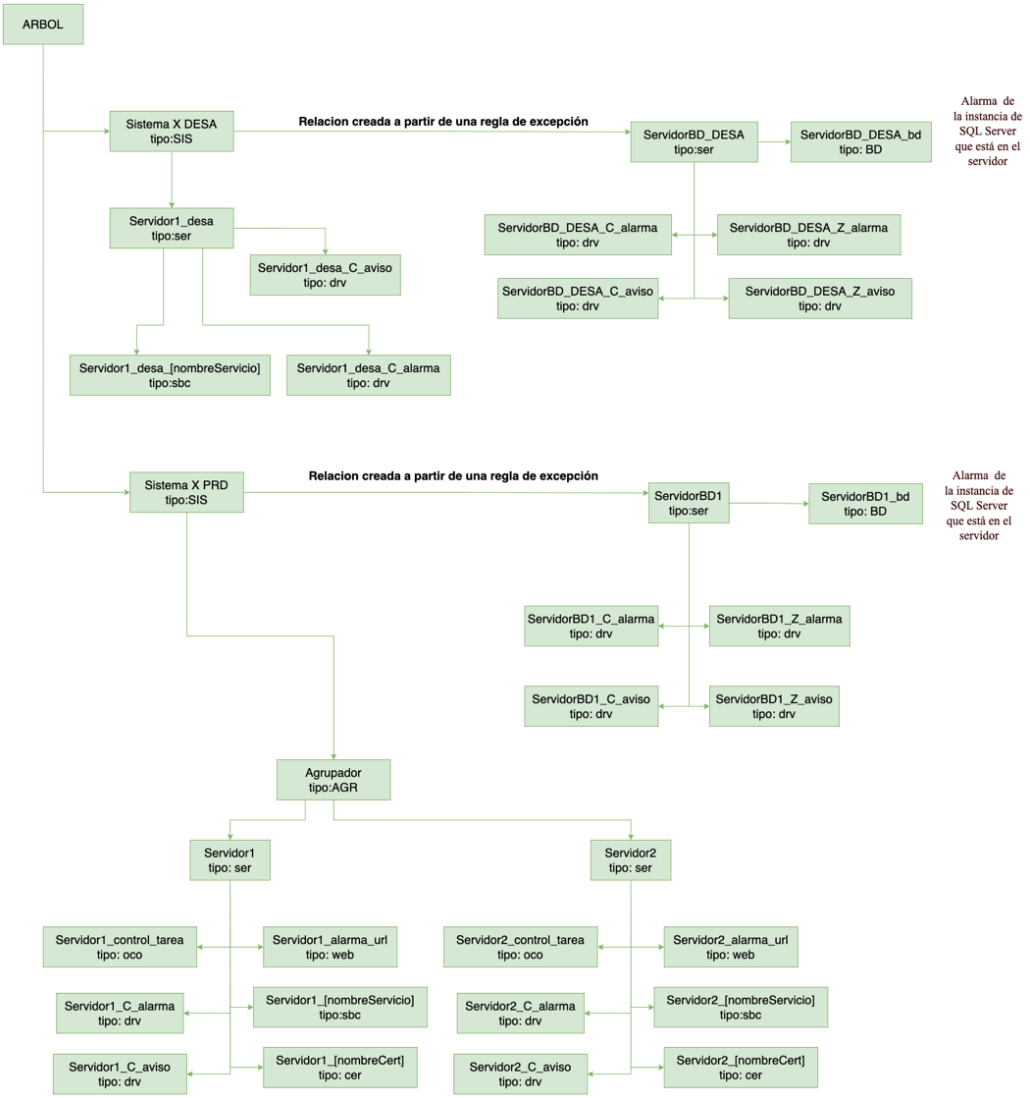

Figura 8. Diagrama del árbol de dependencias correspondiente al plan de acción final

2. Mejora en el proceso de maniobras planificadas

Se define como maniobra cualquier actividad, ya sea proactiva o correctiva, realizada sobre uno o varios objetos y que pueda generar una indisponibilidad temporal (maniobra disruptiva) o el riesgo de que esto ocurra (maniobra no disruptiva).

Actualmente, cuando se crea una maniobra, el equipo responsable debe notificar manualmente al equipo de operaciones mediante un correo electrónico. Como parte del plan de acción se propone:

- Generar automáticamente un borrador del correo de aviso de maniobra con la siguiente información:
    - *Raised* (quién solicita el cambio),
    - *Reason* (por qué se realiza),
    - *Return* (qué se espera del cambio),
    - *Risk* (cuáles son los riesgos),
    - *Resource* (qué recursos se necesitan),
    - *Responsible* (quién es el responsable), y
    - *Relationship* (relación con otros cambios).

  De esta manera se podrá revisar o modificar su contenido y luego se envía automáticamente desde la plataforma.

- Envío de un correo automático a los equipos técnicos involucrados en la maniobra con la información de todos los objetos que pueden verse afectados.

10. Conclusiones

En el contexto actual de la gestión de infraestructuras de TI, la optimización de los procesos de gestión de cambios es crucial para asegurar la continuidad y eficiencia de los servicios. La falta de documentación actualizada sobre las dependencias entre sistemas, así como la escasa coordinación entre los diferentes actores involucrados, representa un riesgo significativo que puede llevar a fallos inesperados en los servidores y, por ende, a interrupciones en los servicios críticos.

La implementación de los dos cambios propuestos: la automatización del mapeo de dependencias y la mejora en las notificaciones de maniobras, permitirá agilizar significativamente el proceso de gestión de cambios. Identificar de manera proactiva los objetos afectados ante una maniobra o una indisponibilidad permitirá detectar rápidamente servidores, bases de datos y aplicaciones impactadas por cambios en la infraestructura, mejorando así la capacidad de respuesta ante incidentes.

Además, comprender las relaciones de dependencia entre los distintos elementos facilitará la evaluación de riesgos y la adopción de medidas preventivas para minimizar fallos y evitar consecuencias negativas en otros sistemas. La automatización de las notificaciones de maniobras eliminará la necesidad de envíos manuales, lo que reducirá demoras en la comunicación y agilizará la implementación de cambios.

La implementación de los cambios propuestos cumple con los principios fundamentales de ITIL en lo que respecta a la gestión de cambios y la mejora continua de los servicios. ITIL enfatiza la necesidad de controlar los cambios en la infraestructura de TI para minimizar los riesgos de interrupciones en los servicios, optimizar el uso de los recursos y mejorar la calidad general de los servicios entregados.

La automatización del mapeo de dependencias y la identificación proactiva de los objetos afectados se alinean con las mejores prácticas de ITIL para la gestión de cambios. ITIL promueve una evaluación cuidadosa de los riesgos asociados con los cambios y la planificación de medidas preventivas. Al comprender las relaciones entre los diferentes elementos de la infraestructura, se facilita esta evaluación de riesgos, garantizando que las acciones correctivas o preventivas puedan implementarse con antelación.

La gestión de incidentes también se ve reforzada, ya que la capacidad de identificar rápidamente los objetos afectados por una maniobra o indisponibilidad mejora la respuesta ante incidentes, lo cual es otro pilar fundamental dentro del marco ITIL. Además, la automatización de las notificaciones de maniobras y la eliminación de procesos manuales se ajustan a la filosofía de ITIL de optimizar los procesos operativos y mejorar la eficiencia.

El plan de acción propuesto no solo responde a las necesidades inmediatas del equipo, sino que también sienta las bases para una infraestructura de TI más robusta y resiliente. Mejorando la comunicación y coordinación entre equipos, se ofrecerá una plataforma que permita gestionar los cambios de manera proactiva, con un impacto mínimo en la operación. Esta solución, además de optimizar los procesos actuales, contribuirá a la mejora continua de los servicios TI en la organización y fortalecerá la alineación de la infraestructura con los objetivos del negocio.

Por último, la visibilidad mejorada sobre la disponibilidad de los objetos a usuarios no técnicos, como el equipo de CAU y operaciones, también está alineada con las recomendaciones de ITIL de fomentar la colaboración entre diferentes equipos y mejorar la comunicación en la gestión de servicios. ITIL subraya la importancia de mantener a todas las partes interesadas informadas durante el ciclo de vida de los servicios, lo cual se logra con el sistema propuesto.

En resumen, la implementación no solo resuelve los problemas específicos detectados en la gestión de cambios, sino que también refuerza los principios y prácticas de ITIL, garantizando que la infraestructura de TI esté alineada con las mejores prácticas internacionales para la gestión de servicios y el ciclo de vida de los cambios.

11. Amenazas a la validez

Las amenazas a la validez pueden clasificarse en dos categorías: validez interna y validez externa. La validez interna hace referencia a la certeza de que los resultados obtenidos son producto directo de la variable estudiada y no de influencias externas o factores no controlados. Por otro lado, la validez externa hace referencia a la posibilidad de generalizar los hallazgos a otros entornos, organizaciones o momentos.

11.1. Amenazas a la validez internas

La validez interna del estudio se garantizó mediante entrevistas con personas con experiencia en la gestión de cambios en servidores dentro de la empresa de estudio. Esto aseguró que la información recopilada fuera precisa y relevante para el contexto organizacional. Además, la propuesta de solución fue validada con los mismos expertos, lo que refuerza la alineación del sistema con las necesidades reales de la empresa.

Sin embargo, podrían existir factores que afecten la validez interna, como la posible ausencia de otras perspectivas dentro de la organización, lo que podría haber revelado riesgos adicionales. Aun así, la metodología utilizada permitió obtener resultados confiables para la empresa, y futuras investigaciones podrían ampliar el alcance del análisis incorporando más actores y fuentes de datos cuantitativos.

11.2. Amenazas a la validez externas

En el caso de este proyecto, la solución fue diseñada específicamente para la infraestructura y procesos internos de la empresa de estudio, por lo que su aplicabilidad fuera de este entorno no fue un objetivo central de la investigación.

Sin embargo, futuras investigaciones podrían explorar adaptaciones del sistema para otros entornos y evaluar su aplicabilidad en diferentes organizaciones con características similares.

12. Trabajos futuros

A continuación, se resumen las recomendaciones que quedarán fuera del alcance de este trabajo, para desarrollos futuros, en función del análisis de las sugerencias obtenidas durante las entrevistas.

Sobre el mapa de dependencias:

- Posibilidad de añadir más información por parte de los usuarios.

  Esta funcionalidad permitiría mejorar significativamente la calidad y precisión de los datos, enriqueciendo la base de conocimientos y facilitando consultas más completas y confiables en el futuro. Al permitir que los usuarios contribuyan con información adicional, el árbol de datos se volvería más robusto, detallado y representativo de la realidad operativa de la infraestructura.

- Persistencia y comparación de versiones

  Esta funcionalidad permitirá almacenar un historial de cambios en la infraestructura, facilitando el seguimiento de modificaciones a lo largo del tiempo y mejorando la trazabilidad de las decisiones tomadas. Con un historial de versiones, los equipos podrían comparar configuraciones pasadas con el estado actual, identificar qué cambios han generado mejoras o problemas y, si es necesario, revertir configuraciones a versiones anteriores de manera segura.

- Visualización avanzada (filtros y búsqueda)

  La posibilidad de aplicar filtros o realizar búsquedas avanzadas proporcionaría información valiosa para la planificación de maniobras, optimizando la toma de decisiones y minimizando riesgos.

- Simulaciones de impacto en la infraestructura

Esta funcionalidad permitirá evaluar de manera anticipada cómo afectarían ciertos cambios en la infraestructura antes de que sean aplicados, reduciendo riesgos y generando maniobras más efectivas.

- Notificación de inconsistencias en el mapa de dependencias

  Esta funcionalidad permitirá identificar y alertar automáticamente sobre inconsistencias en el mapa de dependencias, ayudando a mejorar la precisión y confiabilidad de la infraestructura. Las inconsistencias pueden incluir objetos huérfanos (elementos sin relaciones definidas), dependencias circulares o referencias a sistemas inexistentes, lo que podría generar errores en la planificación y ejecución de cambios.

- Suscripción y notificación automática de cambios en los objetos

  Esta funcionalidad permitirá que los usuarios se suscriban a notificaciones automáticas sobre cualquier cambio realizado en los objetos dentro del mapa, tales como entornos, servidores o bases de datos. De esta manera, los equipos estarán siempre al tanto de las modificaciones relevantes durante las maniobras planificadas.

Sobre el proceso de gestión de cambios:

- Historial de maniobras

  Esta funcionalidad permitirá crear un registro detallado y completo de todas las maniobras realizadas en la infraestructura, lo que sería de gran valor para auditorías y revisiones posteriores. El historial incluiría información sobre qué cambios fueron realizados, quién los llevó a cabo, cuándo se realizaron, qué sistemas o servicios fueron afectados y cuáles fueron los resultados de cada intervención.

- Capacitación de la herramienta

La capacitación adecuada garantiza que los usuarios finales puedan sacar el máximo provecho de la herramienta, comprendiendo sus funcionalidades y cómo utilizarlas de manera eficiente.

13. Referencias bibliográficas

[1] Ayuntamiento de Alfafar. (2014). Glosario de tecnologías de la información y la comunicación (TIC). [Enlace]. Disponible en: https://www.alfafar.es/wp-content/uploads/2014/11/Glosario_TIC_Alfafar.pdf. [Accedido: 11-dic-2024].

[2] *IBM, "What Is IT Infrastructure?", IBM.com.* [Enlace]. Disponible en*: https://www.ibm.com/topics/infrastructure*. [Accedido: 11-dic-2024].

[3] IBM, " *What is IT change management?", IBM.com*. [Enlace]. Disponible en: https://www.ibm.com/mx-es/topics/infrastructure-monitoring. [Accedido: 11-dic-2024].

[4] *Atlassian, "Change Management", Atlassian.* [Enlace]. Disponible en: *https://www.atlassian.com/itsm/change-management*. [Accedido: 11-dic-2024].

[5] ITIL Training Experts, "ITIL Certification in Uruguay," *ITIL Training Experts*. [Enlace]. Disponible en: ITIL® Training Certification. [Accedido: 11-dic-2024].

[6] *AXELOS, "ITIL Service Management", AXELOS.* [Enlace]. Disponible en: https://www.axelos.com/certifications/itil-service-management. [Accedido: 11-dic-2024].

[7] M. Peña Casanova y C. Anías Calderón, "Sistema para ejecutar políticas sobre infraestructuras de Tecnologías de la Información," *Ingeniare. Revista chilena de ingeniería*, vol. 27, no. 3, 2019, pp. 479-494. [Enlace]. Disponible en: https://research-ebsco-com.proxy.timbo.org.uy/linkprocessor/plink?id=152dfc00-5c81-3c5b-b14e-28de758e1984. [Accedido: 11-dic-2024].

[8] L. F. Quintero Gómez y H. Peña Villamil, "Modelo basado en ITIL para la Gestión de los Servicios de TI en la Cooperativa de Caficultores de Manizales," *Scientia et Technica*, vol. 22, no. 4, diciembre 2017, pp. 133-142, Universidad Tecnológica de Pereira, ISSN 0122-1701. [Enlace]. Disponible en: https://research-ebsco-com.proxy.timbo.org.uy/linkprocessor/plink?id=263ae40d-8b45-3a9c-b055-cdbea3f53fb8. [Accedido: 11-dic-2024].

[9] NinjaOne, "Gestión de cambios ITIL: ¿Qué es?", *NinjaOne*. [Enlace]. Disponible en: https://www.ninjaone.com/es/it-hub/endpoint-management/gestion-de-cambios-de-itil-que-es/. [Accedido: 11-dic-2024].

[10] ManageEngine, "Tipos de cambios según ITIL," *ManageEngine*. [Enlace]. Disponible en: https://www.manageengine.com/latam/service-desk/gestion-de-cambios-ti/tipos-de-cambios-segun-itil.html. [Accedido: 11-dic-2024].

[11] *Change Management Insight, "7 R's of Change Management," Change Management Insight*. [Enlace]. Disponible en:

https://changemanagementinsight.com/7-rs-of-change-management/. [Accedido: 11-dic-2024].

[12] *Skills in Information Technology*, "*How can you automate ITIL processes and reduce human intervention?*", *LinkedIn*. [Enlace]. Disponible en: https://es.linkedin.com/advice/0/how-can-you-automate-itil-processes-reduce-ikk8e?lang=es&lang=es#:~:text=La%20automatizaci%C3%B3n%20ITIL%20es%20el,con%20una%20m%C3%ADnima%20intervenci%C3%B3n%20humana. [Accedido: 11-dic-2024].

[13] *IBM, "What Is Dependency Mapping?", IBM.com*, 8 enero 2024. [Enlace]. Disponible en: *https://www.ibm.com/topics/dependency-mapping*. [Accedido: 11-dic-2024].

[14] M. Staron, *Action Research in Software Engineering: Theory and Applications*, Springer, 2020.

[15] Gartner, "Infrastructure Monitoring Tools," *Gartner.com*, 2025. [Enlace]. Disponible en: https://www.gartner.com/reviews/market/infrastructure-monitoring-tools?sort=ratingsCount_desc. [Accedido: 10-ene-2025].

[16] Zabbix SIA, "Zabbix - The Enterprise-Class Open Source Network Monitoring Solution," *Zabbix.com*, 23-jul-2018. [Enlace]. Disponible en: https://www.zabbix.com/. [Accedido: 10-ene-2025].

[17] Zabbix SIA, "Network maps," *Zabbix.com*, 2025. [Enlace]. Disponible en: https://www.zabbix.com/documentation/current/en/manual/config/visualization/maps. [Accedido: 10-ene-2025].

[18] SomoneIT, "zabbix-AutoMapper," *GitHub*. [Enlace]. Disponible en: https://github.com/SomoneIT/zabbix-AutoMapper. [Accedido: 10-ene-2025].

[19] Nagios, "Nagios XI," *Nagios*, 4-feb-2025. [Enlace]. Disponible en: https://www.nagios.com/products/nagios-xi/.[Accedido: 10-ene-2025].

[20] Nagios, "Nagios Core vs Nagios XI," *Nagios*, 15-mar-2024. [Enlace]. Disponible en: https://www.nagios.com/article/nagios-core-vs-nagios-xi/. [Accedido: 10-ene-2025].

[21] Nagios, "Nagios XI - Network Status Map Customization," *Nagios.com*, 2024. [Enlace]. Disponible en: https://support.nagios.com/kb/article/nagios-xi-network-status-map-customization-726.html. [Accedido: 10-ene-2025].

[22] SolarWinds, "Administración de la capacidad de observación, las bases de datos y los servicios de TI," *SolarWinds.com*, 2024. [Enlace]. Disponible en: https://www.solarwinds.com/es. [Accedido: 10-ene-2025].

[23] SolarWinds, "Dynamic Intelligence Maps," *SolarWinds.com*, 2025. [Enlace]. Disponible en: https://www.solarwinds.com/hybrid-cloud-observability/use-cases/intelligent-mapping. [Accedido: 10-ene-2025].

[24] SolarWinds, "Create a dependency between network objects," *SolarWinds.com*, 2016. [Enlace]. Disponible en: https://documentation.solarwinds.com/en/success_center/orionplatform/content/core-creating-a-new-dependency-sw1316.htm. [Accedido: 10-ene-2025].

[25] SolarWinds, "Intelligent Maps for SolarWinds Platform products," *SolarWinds.com*, 2016. [Enlace]. Disponible en: https://documentation.solarwinds.com/en/success_center/orionplatform/content/core-orion-maps-intro.htm. [Accedido: 10-ene-2025].

[26] SolarWinds, "Understand connections on Intelligent Maps in the SolarWinds Platform," *SolarWinds.com*, 2016. [Enlace]. Disponible en: https://documentation.solarwinds.com/en/success_center/orionplatform/content/core-orion-maps-connections.htm. [Accedido: 10-ene-2025].

[27] R. R. Reyes, "Comparación entre Bases de Datos Orientadas a Grafo y Relacional basado en el rendimiento," *Serie Científica de La Universidad de Las Ciencias Informáticas*, vol. 13, no. 5, pp. 174–195, 2020.

[28] Neo4j, "Neo4j Graph Database & Analytics – The Leader in Graph Databases," *Graph Database & Analytics*, 2024. [Enlace]. Disponible en: https://doi.org/10423062553/neo4j-social-share-21. [Accedido: 20-feb-2025].

[29] S. Wang, L. Mei, Y. Tao, and C. Tong, "Design and realization of distributed Rule Engine for scene linkage of Internet of Things," in *2024 6th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI)*, 2024, pp. 396–401. [Enlace]. Disponible en: https://doi-org.proxy.timbo.org.uy/10.1109/IoTAAI62601.2024.10692405. [Accedido: 20-feb-2025].

[30] Drools, "*Drools rule engine: Drools Documentation," Drools.org*, 2025. [Enlace]. Disponible en: https://docs.drools.org/8.44.0.Final/drools-docs/drools/rule-engine/index.html. [Accedido: 20-feb-2025].

[31] R. C. Martin, *Agile Software Development, Principles, Patterns, and Practices: Pearson New International Edition*, Pearson, 2014.

ANEXO 1: Entrevista inicial al líder de plataforma técnica

Somos estudiantes del máster en gestión de sistemas de información y estamos realizando nuestro proyecto final.

El tema que elegimos investigar es la gestión de cambios en infraestructura de TI en el contexto de esta organización, ya que cuando se realizan los mismos y no se coordina ni comunica correctamente, afectan a múltiples grupos o sistemas y no tenemos visibilidad de esto.

Para intentar buscar una solución a este problema, pensamos en crear un mapa de dependencia entre servidores, base de datos, aplicaciones y usuarios que pueden estar afectados cuando un sistema no está disponible por un cambio a realizar.

Para entender mejor este problema y cómo afecta a sus colaboradores estamos realizando esta entrevista. Desde ya muchas gracias por su colaboración.

1. **¿Cómo es el procedimiento ante un cambio en TI (ej. actualización de sistema)?**

   Existe un procedimiento de CALIDAD (COORDINACIÓN DE CAMBIOS A PRODUCCIÓN) que básicamente tiene como objetivo:

   - Definir las actividades a ejecutar para realizar un cambio en producción y evitar o minimizar los efectos negativos a los clientes de TIC partiendo de una solicitud de cambio en producción.

   Como consecuencia se intenta:

   - evitar el re trabajo,
   - mejorar la coordinación entre los diferentes equipos que participan
   - tener un punto de control del seguimiento de la actividad
   - contar con un calendario general de todos los cambios
   - asignar eficientemente los recursos

- mantener informados a todos los interesados durante el cambio

Resumiendo, el procedimiento define el tipo de cambio y cómo debe ser tratado (coordinación, avisos, documentación, urgencia, etc).

Estos cambios pueden ser:

- **Cambio de Emergencia:** cambio que se debe introducir tan pronto como sea posible debido a que la Solución se encuentra inoperativa o en riesgo de quedar inoperativa.
- **Cambio de Urgencia:** cambio que se debe introducir tan pronto como sea posible debido a un incidente crítico en la solución o debido a una mejora crítica solicitada por el cliente que no fue planificada. Se considera crítico una situación que impida la capacidad de brindar el servicio al cliente final.
- **Cambio Estándar:** cambio pre-autorizado por el Comité de Gerentes, de bajo riesgo, que no genera indisponibilidad total.
- **Cambio Planificado:** cambio que requiere de planificación y coordinación.

2. **Ante un cambio, ¿tienen identificados todos los sistemas, base de datos, aplicaciones y usuarios que se verán afectados?**

   De forma automática no.

   Como puedo diferenciar tres tipos de cambios…

   A. Elementos de infraestructura de base (ej, almacenamiento, switches, servidores físicos centrales);
   B. Elementos más de plataforma de base (ej, sistema operativo, versiones de base de datos)
   C. Elementos que componen ya una aplicación en concreto (ej, una nueva versión de una app X)

Para el caso (A) no hay nada automático. En general el equipo encargado tiene sus herramientas particulares para la cual pueden hacer consultas y detectar o informar elementos que son afectados a otros equipos de más alto nivel. Si ya estamos hablando de infra de cpd como racks, ahí ya la información está muy dispersa o es difícil de conseguir.

En lo que respecto al punto (B), tampoco hay algo automático pero ahí sí se está más cerca de lograrlo si se puede inferir desde el sistema de monitoreo. Lo que pasa en la actualidad es que si bien hay alarmas, quizás puedan faltar más nivel de detalle que permita obtener esa info de forma más automatizada.

Finalmente, el punto (C), involucra generalmente menos objetos y el equipo que administra la app seguramente tenga la documentación pertinente.

Resumiendo, en la actualidad se está trabajando en una visión UNIFICADA y AUTOMATIZADA donde cada equipo aporte su información y luego se consolide en la herramienta actual de monitoreo, sumado a que se puedan inferir reglas (relaciones) automáticas a través de las alarmas definidas.

El potencial o idea central es que hoy todo está informatizado en algún repositorio y que interconectando y estandarizando las consultas para recopilarlas y dejarlas disponible de forma centralizada, puede ser un objetivo alcanzable en el mediano plazo.

Herramientas como Ansible, base de datos orientadas a grafos y algoritmos, sumado a las alarmas que monitorean los servicios, puede ser el camino factible para seguir

3. **¿Cómo se realiza la coordinación antes de una maniobra?**

Una vez que se detecta la necesidad de la realización de un cambio, el dueño de la solución planifica el cambio con las diferentes actividades para luego coordinar a través del equipo de Programación de la Operación la fecha y horas más todas las notificaciones pertinentes.

Se tendría que definir la maniobra en la herramienta de monitoreo interna con el objetivo de dar visibilidad y seguimiento a la misma.

Los cambios en elementos que involucran PRODUCCIÓN siempre se tendrían que coordinar por este medio… si involucra elementos NO PRODUCTIVO (ej DESARROLLO) queda por fuera de este procedimiento, lo que quizás sea una oportunidad de mejora estandarizar este tema.

4. **Ante un cambio, ¿notifican a CAU?**

   Los avisos se comunican en general a través de una casilla genérica y en particular, si el cambio impacta fuertemente a los usuarios finales (ya sea por la importancia del sistema a modificar o son productos transversales como el correo corporativo), se hace una profundización más exhaustiva con el CAU, explicando el impacto o posibles llamados que pueden recibir.

5. **Nuestra idea de mejora involucra la creación de un mapa de dependencias entre servidores, base de datos, aplicaciones y usuarios para identificar y notificar de manera sencilla la indisponibilidad de los sistemas y sus afectados. ¿Qué opinas al respecto? ¿Qué sistema utilizas de los que ya tenemos operativos?**

   Respecto a tener un sistema que tenga una visión unificada es algo super valioso para la calidad del servicio que se brinda ya que se puede utilizar para varios casos de uso:

   a. Detectar los elementos involucrados ante un cambio.

      Ejemplo, tengo que modificar la versión del sistema operativo de un servidor X y saber en dónde y a quién impactan.

   b. Detectar los elementos involucrados ante un incidente.

      Ejemplo, se cae una BD X y rápidamente saber los elementos afectados y a qué usuarios y sistemas afecta.

c. Simular situaciones del estilo ¿qué pasaría si tengo que actualizar tal o cual objeto?

d. Detectar objetos “CALIENTES”, o sea aquellos que tienen mayor impacto o relevancia en los sistemas, para luego trabajar sobre ellos, por ej: armando un buen plan de contingencia, especializando y formando a los técnicos, priorizando compras de soporte y mantenimiento, etc.

El desafío está en que cada equipo tiene parte de la historia y la fuente es diversa y todo tendría que terminar en “algo” estandarizado que permita una visión global unificada, oportuna y actualizada.

Actualmente, contamos con alarmas definidas en la herramienta de monitoreo interna que permiten monitorear elementos críticos de las diferentes soluciones tecnológicas. Es una herramienta que tiene el potencial que está inmersa en la cultura de TIC. El desafío en el que estamos actualmente es poder llegar desde las alarmas (algo obligatorio para todo sistema productivo) a grafos de dependencia (inferidos automáticamente).

ANEXO 2: Entrevista inicial al líder del equipo encargado de la administración de los servidores

Somos estudiantes del máster en gestión de sistemas de información y estamos realizando nuestro proyecto final.

El tema que elegimos investigar es la gestión de cambios en infraestructura de TI en el contexto de esta organización, ya que cuando se realizan los mismos y no se coordina ni comunica correctamente, afectan a múltiples grupos o sistemas y no tenemos visibilidad de esto.

Para intentar buscar una solución a este problema, pensamos en crear un mapa de dependencia entre servidores, base de datos, aplicaciones y usuarios que pueden estar afectados cuando un sistema no está disponible por un cambio a realizar.

Para entender mejor este problema y cómo afecta a sus colaboradores estamos realizando esta entrevista. Desde ya muchas gracias por su colaboración.

1. **¿Cómo es el procedimiento ante un cambio en TI (ej. actualización de sistema)?**

   Existen diferentes tipos de cambios o maniobras: en casos de urgencia y emergencia, donde se prioriza que la solución vuelva a quedar operativa en el tiempo indicado (dependiendo del TMO); en casos de un cambio estándar, el cual no genera indisponibilidad total y cual esta preautorizado por el Comité de Gerentes y el cambio planificado, el cual requiere de planificación y coordinación.

2. **Ante un cambio, ¿tienen identificados todos los sistemas, base de datos, aplicaciones y usuarios que se verán afectados?**

   No. ojalá tuviéramos una base de conocimiento transversal que pudiéramos consultar.

   Los documentos, si bien están actualizados, están separados por equipos. No existe un sistema donde se centralice toda la información.

3. **¿Cómo se realiza la coordinación antes de una maniobra?**

   En una maniobra planificada, el procedimiento sería el siguiente:

   a. Se coordina la maniobra con los equipos involucrados (día, hora, técnicos asignados, tareas a realizar, marcha atrás).
   b. Se crea una "maniobra" en el sistema de monitoreo para que los objetos afectados no alarmen durante el cambio.
   c. Se envía un mail al equipo de operaciones para que sean los facilitadores de la misma y den aviso del cambio a toda la unidad de TI.
   d. El equipo de operaciones envía el mail con el aviso.
   e. Se realiza la maniobra el día y fecha estipulada.

4. **Ante un cambio, ¿notifican a CAU?**

   Si. Cuando los operadores envían el mail a toda la unidad de TI.

   Además, en caso de maniobras urgentes tenemos canales de Teams para informar el problema y la resolución.

5. **Nuestra idea de mejora involucra la creación de un mapa de dependencias entre servidores, base de datos, aplicaciones y usuarios para identificar y notificar de manera sencilla la indisponibilidad de los sistemas y sus afectados. ¿Qué opinas al respecto? ¿Qué sistema utilizas de los que ya tenemos operativos?**

   Me parece muy buena idea. Se podría automatizar parte de la solución actual de dependencias de la herramienta de monitoreo, que actualmente es manual.

ANEXO 3: Entrevista inicial al líder del centro de atención a usuarios

Somos estudiantes del máster en gestión de sistemas de información y estamos realizando nuestro proyecto final.

El tema que elegimos investigar es la gestión de cambios en infraestructura de TI en el contexto de esta organización, ya que cuando se realizan los mismos y no se coordina ni comunica correctamente, afectan a múltiples grupos o sistemas y no tenemos visibilidad de esto.

Para intentar buscar una solución a este problema, pensamos en crear un mapa de dependencia entre servidores, base de datos, aplicaciones y usuarios que pueden estar afectados cuando un sistema no está disponible por un cambio a realizar.

Para entender mejor este problema y cómo afecta a sus colaboradores estamos realizando esta entrevista. Desde ya muchas gracias por su colaboración.

1. **¿Estas de acuerdo con la forma de notificación de maniobras (via mail)?**

   SÍ, estoy de acuerdo. Actualmente el aviso se hace con anticipación y al momento de iniciar la maniobra.

   También me parece útil coordinar con nuestro equipo cuando la maniobra puede afectar a servicios importantes que pueden generar grandes repercusiones a nivel de los usuarios.

2. **Si tuvieras que describir un día en el que ocurrirá una maniobra, ¿cómo sería?**

   El día de la maniobra se vuelve a difundir el aviso entre el equipo para que estén todos atentos a posibles reclamos por parte de los usuarios y cómo responder ante consultas esperadas.

3. **¿Cuál es el promedio de llamadas por día relacionadas a maniobras? ¿Qué porcentaje representan en relación al total de llamadas? Si no tienen el dato puede ser un estimativo o percepción.**

   Para poder responder esta pregunta hay que considerar varios factores, como si la maniobra es disruptiva o no, si es exitosa y a qué porcentaje de usuarios afecta dentro de la organización de no resultarlo.

   En el caso de una maniobra que sea transparente al usuario, el porcentaje de llamadas es despreciable.

   En el caso de que una maniobra falle y sea un sistema crítico en la organización, se alarman nuestros indicadores de monitoreo y el porcentaje final depende del tiempo de resolución o vuelta atrás de la misma.

   Por ejemplo, la última maniobra que presentó una falla y estaba relacionada a un sistema crítico, representó el 30% de las llamadas del día, en este caso la respuesta a vuelta atrás fue rápida, 15 minutos aproximadamente.

4. **¿Cuál es el procedimiento en el caso de que ocurra una baja imprevista de un sistema?**

   En el caso de una baja imprevista de un sistema, se genera una solicitud al equipo correspondiente, se da aviso al jefe del equipo y se publica un aviso en un canal de Teams para que todos los equipos que puedan estar involucrados estén al tanto.

   Además, en caso de que el sistema sea crítico para poder brindar nuestro servicio, por ejemplo, caída de alguno de nuestros canales de comunicación, se sigue un protocolo de contingencia que tenemos definido.

5. **Nuestra idea de mejora involucra la creación de un mapa de dependencias entre servidores, base de datos, aplicaciones y usuarios para identificar y notificar de manera sencilla la indisponibilidad de los sistemas y sus afectados. ¿Qué opinas al respecto?**

Me parece muy buena idea y sería beneficioso para nuestro equipo, ya que al estar notificados los afectados, se podría disminuir la cantidad de llamadas que recibimos en caso de falla.

ANEXO 4: Notas de la segunda entrevista al líder de la plataforma técnica, en formato de reunión virtual.

Introducción:

Luego de analizar las respuestas brindadas en la primera entrevista y para entender mejor este problema y cómo afecta a sus colaboradores estamos realizando esta segunda entrevista. Desde ya muchas gracias por su colaboración.

Preguntas:

1. **¿Cómo tienes identificado el procedimiento ante un cambio en TI? Es decir, ¿qué tipo de comunicaciones se tienen que realizar y cuándo?**

   Cambio de emergencia (incidente), se le avisa a: comité de gerentes, operaciones, personas que van a trabajar en ello. Luego de solucionado, se hace informe con la solución final y cambios realizados. Se busca que se resuelva el problema cuanto antes, sin toda la burocracia, planificación, informar por correo, etc.

   Existe canal de teams, por allí notifican cosas urgentes. Rapidamente se informa a operaciones e infraestructura, y jefes de equipos de desarrollo. Se informa sobre el problema, lo que impacta, y luego de resolver se avisa que todo está ok.

   Luego, reunion de lecciones aprendidas (que fue lo que paso? era evitable el problema? porque no se detecto previamente?), y luego informe mas formal. Se pueden desencadenar acciones -> poner mas alarmas, mas planificacion, etc.

   En otros cambios menos urgentes, hay etapas previas, planificacion de cada paso (plan de ejecucion), identificacion de involucrados y planificacion de actividades. No se usa teams por lo gral, se usa correo. Cuando finaliza la maniobra, se difunde a traves de operaciones.

   Cosas que hicieron últimamente

   - Inversión para incorporar equipos que tengan alta redundancia. equipos con varias fuentes. Cambio de equipamiento en caliente sin tener que bajar

el servicio. Esto minimiza impactos en cambios. Ahorra tiempo de planificación, dadas de baja de los servicios, etc.

- Desacoplar soluciones de hardware a través de la virtualización. De esa forma se permite migrar un objeto obsoleto de hardware sin bajarla. Esto permitió maniobras pre-aprobadas.

2. ¿Qué requisitos funcionales debería tener un mapa de dependencias?

En la entrevista anterior nos habías mencionado:

a. Detectar los elementos involucrados ante un cambio.
b. Detectar los elementos involucrados ante un incidente.
c. Simular situaciones del estilo ¿qué pasaría si tengo que actualizar tal o cual objeto?
d. Detectar objetos "CALIENTES", o sea aquellos que tienen mayor impacto o relevancia en los sistemas.
e. El desafío en el que estamos actualmente es poder llegar desde las alarmas (algo obligatorio para todo sistema productivo) a grafos de dependencia (inferidos automáticamente)

¿Agregarías alguno más?

- Que el árbol sea confiable y representativo

- Si vamos de abajo hacia arriba (infra hasta soluciones), hay que ver cómo recolectar info sobre cómo se relacionan los elementos de hardware (infraestructura)). Son fuentes heterogeneas, no normalizadas, diferentes fuentes (equipos y elementos).

  - Objetos tienen jerarquia. Un BDD, aplicacion no tiene sentido si un servidor no lo aloja. Una solucion no tiene sentido si no esta asociado a una BDD y un servidor. A traves de la definicion de objetos, poder inferir reglas, que armen el arbol solo. Quizás no es el mejor arbol, pero capaz que aporta mas que no tener nada.

- Toda la parte de bajo nivel, desde servidores para abajo, se podria automatizar/saber. Ya mas arriba (bdd, aplicaciones, etc), se podria inferir por ej si hay un servidor SAP, y una BDD se llama SAP, inferir que esta dentro de ese servidor. A menos que se recopile info de otros equipos y sea una.

- En un maniobra, saber a que equipo involucro. Voy a afectar a algun otro elemento? Es critico? Que salte alguna lista de servicios involucrados o de personas a los que le puedo avisar.

- Tiene que ser automático y dinámico. No puede ser manual. Pero hoy en dia la info de hardware, servidores, etc, esta distribuida en varias fuentes. Hay que preguntarle a varias personas para saber donde estan alojados los servidores, etc etc. Fuentes diversas en distintos formatos. Hay que normalizar para que puedan relacionarse. Saber a que proveedor pertenece un servidor ayuda por si necesita mantenimiento o actualización.

- Estaria bueno simular obsolescencia de servicios/servidores, para saber como afectará a otros elementos.

- Aplicar un *score* de obsolescencia, cuanto mas alto el score, mas riesgo de obsolescencia. se le daría valor a los nodos, sumar hasta la raíz, y ver cuales son los sistemas más críticos.

- Ver objetos huérfanos.

- No agregar más pasos al creador de cada objeto. Dejarlo opcional. Si quieren agregar más info con un buen nombre, que lo hagan.

Todos los objetos tienen: Entorno tipo y nombre, por ej todos los sistemas SAP tienen un mismo entorno -> SAP. Podrian estar en la misma rama. tipo servidor, tipo drb (disco), hay reglas para inferir. Jugar con etiquetas/nomenclaturas, etc. Si estandarizamos cómo escribir los objetos, mejora las inferencias. Equipo responsable también puede ayudar para inferir cosas.

El principal objetivo es mantener sistemas activos, con poco tiempo de downtime, mantener a los usuarios informados, y mejorar la atencion a usuario en general. Detectar servicios “calientes” para mejorarlos, y minimizar riesgos.

3. **¿Con qué funcionalidades cuenta la herramienta interna de monitoreo actualmente con respecto a los mapas de dependencias? ¿Se usa actualmente? ¿Con qué objetivo? En caso de visualizar indisponibilidad en este mapa, realiza alguna acción o podría disparar alguna acción? ¿Cómo crees que podría mejorar?**

   Hoy hay objetos, y se depende de los equipos para que armen ese mapa.

   El objeto propaga hacia arriba el error, y alerta sistema en el que forma parte. Si se cae servidor de xchange, impacta arbolito de xchange cambiando su color. En el caso que haya una maniobra, pone el color en celestito.

   Falta el arbolito. Problema: nadie va a armar el arbolito manual. Son muchos objetos interrelacionados. La info esta dispersa en varios equipos.

   Otros programas: infieren dependencias en función de logs de servidores, con quien tiene comunicaciones abiertas y en qué puerto? No es la idea que la herramienta interna de monitoreo haga esto, para eso contratarian otras soluciones. Hay que darle mucha visibilidad, problema de privacidad.

4. **¿Conocés la historia de la herramienta interna de monitoreo? Han probado otras herramientas o se han basado en otras herramientas para ampliar sus funcionalidades?**

   Se desarrolló hace 22/23 años. En esa época capaz no estaban tan desarrolladas las herramientas, y no había una visión integradora de una única herramienta. La herramienta interna ha logrado:

   - Seguir manteniendo idea base: escucho, me entero de lo más importante y aviso de una forma a un operador que involucre a la guardia.

- No tiene agente, o un servicio muy bajo/ muy poco invasivo. Casi no precisa actualización, en java, minimiza ser invasivo en los servicios.

- Está en la cultura de la empresa. Todos piensan en funcion de la cultura. Ya todos saben limitaciones y funcionalidades de la herramienta actual y comparten ese lenguaje.

- No tiene costo adicional. Solucion integral, hecha en casa. Desarrollo a medida.

- Oracle es el mejor que puede monitorear/administrar cosas de Oracle, lo mismo con Microsoft, etc. Cuando se compra Oracle, ya te viene la herramienta de monitoreo de Oracle. La herramienta interna resuelve una parte del problema. Las herramientas de Oracle, etc, informa a la herramienta interna, hace el objeto alarma, etc etc. Se administra la cosa fina con la mejor herramienta especifica para cada caso y lo importante va a la herramienta interna, para que despues entren en el estandar de monitoreo de cualquier infraestructura.

- Hay mucho *knowhow* metido en la herramienta.

- Antes se usaba Nagios XL, pero se dejo de usar.

5. **¿Están dispuestos a reemplazar a la herramienta de monitoreo que utilizan actualmente? Hemos averiguado, que tanto Nagios como Zabbix son herramientas de monitoreo de infraestructura que tienen versiones gratuitas, con opciones de pago para obtener.**

   Por el momento no es una opción. La cultura que se creó sobre la herramienta actual es muy fuerte.

Muchas gracias por tu tiempo!

ANEXO 5: Notas de la segunda entrevista al líder del equipo que administra los servidores.

Introducción:

Luego de analizar las respuestas brindadas en la primera entrevista y para entender mejor este problema y cómo afecta a sus colaboradores estamos realizando esta segunda entrevista. Desde ya muchas gracias por su colaboración.

Preguntas:

1. **¿Cómo tienes identificado el procedimiento ante un cambio en TI? Es decir, ¿qué tipo de comunicaciones se tienen que realizar y cuándo?**

   Actualmente existen diferentes procesos a seguir dependiendo del tipo de cambio que sea.

   Estos cambios pueden ser:

   - Cambio de Emergencia: cambio que se debe introducir tan pronto como sea posible debido a que la solución se encuentra inoperativa o en riesgo de quedar inoperativa.

     Ante este tipo de cambios se realizan los siguientes pasos:

     1. Notificación inmediata. Se informa de la urgencia a través de un canal de Teams, notificando a Operaciones, Infraestructura y jefes de equipos de desarrollo. Se comunica el problema y su impacto sin burocracia ni procesos formales previos.

     2. Se prioriza la solución rápida del problema y una vez resuelto se comunica que el sistema está nuevamente operativo a través de Teams.

     3. Documentación y análisis. Más adelante se elabora un informe con la solución final y los cambios realizados. Se realiza una reunión de lecciones aprendidas para analizar: ¿Qué sucedió? ¿Por qué no se

detectó antes?

Con base en el análisis, se pueden tomar medidas como:

a. Implementar nuevas alarmas o monitoreos.

b. Mejorar la planificación y prevención de incidentes.

c. Ajustar procesos para evitar futuros problemas similares.

- Cambio de Urgencia: cambio que se debe introducir tan pronto como sea posible debido a un incidente crítico en la solución o debido a una mejora crítica solicitada por el cliente que no fue planificada. Se considera crítico una situación que impida la capacidad de brindar el servicio al cliente final.
- Cambio Estándar: cambio pre-autorizado por el Comité de Gerentes, de bajo riesgo, que no genera indisponibilidad total.
- Cambio Planificado: cambio que requiere de planificación y coordinación.

Ante estos tipos de cambio se deben realizar los siguientes pasos:

1. Coordinar con el equipo o los equipos involucrados y definir procedimiento de maniobra, fecha y hora de ejecución.

2. Crear una maniobra en la herramienta de monitoreo actual, indicando: el nombre de la maniobra, descripción breve de la maniobra, persona responsable, la fecha y hora a realizarse y los objetos involucrados para la misma (se deben buscar y seleccionar dichos objetos).

3. Enviar mail al equipo de operaciones con una breve descripción de la maniobra, indicando los datos más relevantes, el link de la maniobra creada en la herramienta de monitoreo y el plan de la maniobra adjunto con las tareas y equipos responsables.

4. El equipo de operaciones envía el mail a toda la unidad de TI.

2. **¿Qué requisitos funcionales debería tener un mapa de dependencias?**
3. **¿Con qué funcionalidades cuenta la herramienta de monitoreo interna actualmente con respecto a los mapas de dependencias? ¿Se usa actualmente? ¿Con qué objetivo? En caso de visualizar indisponibilidad en este mapa, realiza alguna acción o podría disparar alguna acción? ¿Cómo crees que podría mejorar?**
4. **¿Conocés la historia de la herramienta interna de monitoreo? Han probado otras herramientas o se han basado en otras herramientas para ampliar sus funcionalidades?**

La herramienta de monitoreo actual la administra otro equipo. Respondo las preguntas 2,3 y 4 a continuación de lo que estuve investigando.

La herramienta de monitoreo actualmente utilizada en la organización ha sido desarrollada y mantenida internamente durante más de 20 años, lo que refleja su sólida trayectoria dentro de la infraestructura tecnológica de la empresa. A lo largo de este tiempo, ha evolucionado y se ha adaptado a los avances tecnológicos, manteniendo un monitoreo confiable de toda la infraestructura. Durante este proceso, se han identificado diversos desafíos, los cuales fueron discutidos en la entrevista con el sponsor del área de infraestructura. Hoy en día, la herramienta forma parte integral de la cultura organizacional de la empresa.

Los elementos principales de la herramienta son los objetos, los cuales representan los distintos componentes de la infraestructura que necesitan ser monitoreados. Cada objeto dentro del sistema de monitoreo tiene asignados los siguientes atributos clave:

- Nombre
- Tipo
- Entorno
- Estado

- Equipo responsable

Algunos de los tipos de objetos definidos en el sistema incluyen:

- **SER**: Monitorea la disponibilidad de servidores.
- **DRV**: Monitorea el espacio en los discos.
- **WEB**: Monitorea la disponibilidad de sitios web.
- **SRV**: Monitorea la ejecución de servicios de Windows.
- **CER**: Monitorea el vencimiento de certificados.
- **OCO**: Monitorea tareas programadas en un servidor.
- **BD**: Monitorea la instancia SQL.

Los objetos DRV y CER, pueden ser de aviso, el cual notifica mediante un correo al equipo responsable que alcanzó el límite de espacio definido o que faltan x días para expirar, o de alerta, el cual notifica mediante un correo y una llamada por parte del equipo de operaciones al equipo responsable para que sea atendido. En ambos casos, el estado del objeto es “AFECTADO”.

Los objetos SER, WEB, SRV, OCO y BD son de alerta, los cuales notifican mediante un correo y una llamada por parte del equipo de operaciones al equipo responsable cuando el estado del objeto es “AFECTADO”.

Todos los objetos que pertenecen a una misma solución generalmente tienen el mismo “entorno” cuyo nombre representa a la solución a la que pertenecen. Sin embargo, existe una excepción con algunos servidores de base de datos. Estos servidores (que son objetos de tipo “SER”) contienen bases de datos de diferentes soluciones y se definen en un entorno “BD”, independientemente de las soluciones a la que pertenezca.

La nomenclatura para los nombres de los objetos sigue el siguiente formato:

- **Servidor (SER)**: [NombreServidor]_(aviso|alarma)

- **Discos (DRV)**: [NombreServidor]*[C|NombreDisco]*(aviso|alarma)
- **Sitios web (WEB)**: [NombreServidor]_(aviso|alarma)_URL
- **Servicios Windows (SRV)**: [NombreServidor]_[NombreServicio], por ejemplo, DNS o algún otro programa.
- **Certificados (CER)**: [NombreServidor]_[NombreCertificado]
- **Tareas (OCO)**: [NombreServidor]_[NombreTarea]
- **Instancia SQL (BD)**: [NombreServidor]_bd

El sistema requiere la instalación de un agente que consume un web service en cada servidor donde se definen los objetos a monitorear, y se ejecuta a través de un servicio de Windows.

Árbol de dependencia.

Un árbol de dependencia en un sistema de monitoreo es una herramienta crucial para la gestión eficiente de la infraestructura tecnológica, ya que proporciona una visión clara y estructurada de cómo los distintos componentes están interconectados, mejorando la capacidad de respuesta ante incidentes y optimizando la planificación de mantenimiento y cambios.

Características del árbol de dependencia del sistema actual:

El sistema de monitoreo actual, dispone de árbol de dependencias a través de una representación gráfica se puede visualizar las relaciones entre los diferentes componentes de las soluciones tecnológicas definidas, como servidores, servicios, aplicaciones, bases de datos, redes, y otros recursos críticos.

Cada árbol asociado a un sistema se debe crear de forma manual. En primer lugar, se deben crear dos nuevos objetos sin acciones: un objeto sistema (de tipo SIS) y uno o varios objetos agrupadores (de tipo AGR). El objeto sistema hace de "padre" del árbol. El o los objetos agrupados, permiten brindar la representación de un balanceador o simplemente para agrupar dos objetos que tienen alta

disponibilidad o, en el caso de base de datos, tengan un cluster definido. Estos objetos solo brindan una relación lógica y permiten ordenar y jerarquizar los objetos.

El árbol representa las dependencias entre los distintos objetos que ya están previamente definidos. Por ejemplo, un objeto de base de datos puede depender de la disponibilidad del servidor de base de datos. Las dependencias pueden ser unidireccionales (un componente depende de otro) o bidireccionales.

Permite a los equipos de monitoreo entender rápidamente las implicaciones de un fallo o cambio en cualquier parte de la infraestructura. Si un servicio falla, el árbol de dependencia ayuda a identificar qué otros componentes se verán afectados, lo que facilita una respuesta más rápida y precisa.

El árbol de dependencia suele presentarse en una interfaz visual, donde los nodos representan los componentes monitoreados y las conexiones entre ellos indican las dependencias.

Los cambios en el estado de un nodo (por ejemplo, un servicio que falla) se reflejan visualmente en el árbol con un color, proporcionando a los operadores una visión clara y accesible del estado de la infraestructura:

- Gris indica disponibilidad
- Rojo indica indisponibilidad.
- Azul muestra una maniobra en curso.

5. **¿Están dispuestos a reemplazar a la herramienta de monitoreo actual? Hemos averiguado, que tanto Nagios como Zabbix son herramientas de monitoreo de infraestructura que tienen versiones gratuitas, con opciones de pago para obtener.**

   Por lo que me comentaron no están dispuestos a reemplazar la herramienta actual, ya que forma parte de la cultura de la empresa y no tiene costo adicional de mantenimiento ni consultoría.

ANEXO 6: Plan de acción correspondiente al ciclo 1 presentado a los actores clave para su retroalimentación.

Problema

El objetivo de este estudio es analizar las deficiencias en la gestión de cambios en servidores Windows dentro de la organización, identificando los factores que generan demoras en el proceso, fallos inesperados y afectaciones en la continuidad del servicio.

A partir de las entrevistas realizadas y los relevamientos realizados sobre el proceso de cambios actual y la herramienta de monitoreo interna, se identificaron los siguientes problemas:

1. Ausencia de un mapeo confiable de dependencias entre los elementos de la infraestructura: No existe un registro claro y confiable de las relaciones entre los objetos, lo que dificulta una gestión eficiente.

2. Falta de automatización en el árbol de dependencias: Actualmente, el árbol de dependencias se define de forma manual, lo que lo convierte en un proceso engorroso y poco frecuente, resultando rápidamente obsoleto.

3. Información dispersa y no estandarizada: La información sobre las relaciones entre los objetos no está normalizada y se encuentra distribuida en varias fuentes y equipos, lo que complica su acceso y análisis.

4. Falta de visibilidad del impacto de los cambios: Cuando se realiza un cambio en un objeto, no se tiene claridad sobre qué otros objetos pueden verse afectados.

5. Falta de automatización en la comunicación de maniobras: Las notificaciones se envían manualmente vía mail, lo que genera demoras y posibles errores en el proceso de cambios.

A partir de las entrevistas realizadas, se han identificado los siguientes requisitos funcionales y técnicos para la solución propuesta:

1. Automatización y visualización de dependencias: La solución debe integrar y mostrar las relaciones entre servidores, aplicaciones y bases de datos, facilitando la gestión de cambios y su impacto.

2. Estandarización de monitoreo: Es necesario establecer un proceso que normalice y consolide los objetos involucrados en el monitoreo, garantizando coherencia y precisión en la información.

3. Creación de un mapa de dependencias confiable: Aunque no sea completamente preciso, debe representar fielmente la infraestructura actual, proporcionando visibilidad y control.

4. Integración sin impacto significativo: La solución debe incorporarse sin alterar drásticamente la forma en que se utiliza la herramienta actual, evitando problemas en la interfaz y minimizando la necesidad de capacitación adicional.

5. Continuidad con la herramienta de monitoreo existente: Dado que forma parte de la cultura organizacional y está profundamente integrada en los procesos diarios, no se considera viable reemplazarla por una herramienta externa.

Propuesta de solución

Considerando los siguientes aspectos:

- La solución debe ser completamente automática.

- Otras herramientas utilizan etiquetas o relaciones "padre/hijo" para definir dependencias o jerarquías.

- No se desea agregar nuevos campos durante la creación de un objeto en la herramienta interna.

- Cada objeto ya cuenta con los campos definidos: entorno, tipo, nombre, estado y equipo responsable.

- En la solución de estudio se administran únicamente objetos con los siguientes tipos:

    - SER: Monitorea la disponibilidad de servidores.
    - DRV: Monitorea el espacio en los discos.
    - WEB: Monitorea la disponibilidad de sitios web.
    - SRV: Monitorea la ejecución de servicios de Windows.
    - CER: Monitorea el vencimiento de certificados.
    - BD: Monitorea la instancia SQL.
- El sistema de nombres actual de los objetos tiene una estructura definida.
    - Servidor: [NombreServidor]_(aviso|alarma)
    - Discos: [NombreServidor]*[C|NombreDisco]*(aviso|alarma)
    - Sitios web: [NombreServidor]_(aviso|alarma)_URL
    - Servicios Windows: [NombreServidor]_[NombreServicio], por ejemplo, DNS o algún otro programa.
    - Certificados: [NombreServidor]_[NombreCertificado]
    - Base de datos: [NombreServidor]_bd
- No se busca cambiar el sistema de monitoreo, sino continuar utilizando la herramienta interna.
- Actualmente, las notificaciones sobre maniobras planificadas se realizan de forma manual.

Se propone:

1. Planificación de una nueva funcionalidad dentro de la herramienta interna que, a partir de los campos entorno, tipo y nombre de los objetos monitoreados, infiera

automáticamente las relaciones entre ellos. Luego, a través de una API, se generarán los objetos gráficos correspondientes sin necesidad de intervención manual.

Funcionamiento**:**

1. Agrupación por entorno: Se creará un mapa específico para cada entorno definido.
2. Estructura del mapa:
    - Cada mapa incluirá un objeto raíz denominado "SISTEMA", que representará el entorno.
    - Si es necesario, se generarán objetos agrupadores para organizar servidores con nombres similares o relacionados. Por ejemplo, si existen servidores ser1, ser2, serDB1, serDB2, se crearán dos agrupadores: uno para los "ser" y otro para los "serDB".
    - Si un servidor no tiene otros con nombres similares, no se generará un agrupador y dependerá directamente del objeto "SISTEMA".
3. Relaciones entre objetos:
    - Los objetos de tipo "DRV", "WEB", "CER" y "SVC" dependerán de un objeto "SER".
    - La relación se establecerá utilizando la información del nombre del objeto que indica a qué servidor está asociado.
4. Relaciones con BDD:
    - El equipo encargado de controlar las bases de datos, debe crear sus objetos con la siguiente nomenclatura:
        1. [NombreServidorDB]_bd para el objeto de tipo BD que controla la disponibilidad de la instancia de SQL.
        2. [NombreServidorDB]_[NombreDisco]_[alarma|aviso] para los objetos de tipo DRV

- Todos los objetos que contengan en su nombre [NombreServidorDB] estarán relacionados con el objeto de tipo SER con ese nombre.

- La mayoría de los servidores de base de datos son compartidos por varias soluciones y/o aplicaciones.

5. Objetos huérfanos:

    - Aquellos objetos cuyo nombre no coincida parcialmente con el de un servidor dentro del mismo entorno, se mostrarán como elementos aislados en el mapa.

Este enfoque permitirá una generación automática y estructurada de mapas de dependencias, optimizando la visualización y gestión de los objetos en la infraestructura.

2. Automatización del envío de notificaciones de maniobras

Actualmente, cuando se crea una maniobra, el equipo responsable debe notificar manualmente al equipo de operaciones mediante un correo electrónico. Se propone automatizar este proceso, de modo que, al registrar una maniobra en la herramienta interna de monitoreo, se genere y envíe automáticamente un correo con todos los detalles relevantes. Esto garantizará una comunicación más ágil y reducirá el riesgo de omisiones o demoras en la notificación, además de un cambio en la automatización del proceso de aviso de maniobras planificadas.

Conclusión

La implementación de estos dos cambios permitirá agilizar el proceso de gestión de cambios al:

- Automatizar las notificaciones de maniobras, eliminando la necesidad de envíos manuales y reduciendo demoras en la comunicación.

- Identificar de manera proactiva los objetos afectados ante una maniobra o indisponibilidad, permitiendo detectar servidores, bases de datos y aplicaciones

impactadas por cambios en la infraestructura y mejorando la capacidad de respuesta ante incidentes.

- Comprender las relaciones de dependencia entre elementos, facilitando la evaluación de riesgos y la adopción de medidas preventivas para minimizar fallos.

- Brindar mayor visibilidad sobre la disponibilidad de los objetos a usuarios no técnicos, como el equipo de CAU y operaciones, quienes podrán consultar el mapa de dependencias para verificar la restauración de sistemas sin depender del equipo técnico responsable.

A su vez, la automatización del mapa de dependencias permitirá obtener beneficios adicionales a futuro, tales como:

- Identificación de servidores, bases de datos o aplicaciones “huérfanos”, sin relaciones con otros elementos, optimizando su documentación, uso y mantenimiento.

- Facilitación de simulaciones de impacto ante cambios en la infraestructura, como actualizaciones de sistemas operativos o migraciones de bases de datos, ayudando a prever y mitigar riesgos.

- Evaluación del impacto de la obsolescencia de un servidor o servicio, mediante la generación de un "score de obsolescencia" que indique el nivel de riesgo asociado a cada componente.

- Definición y gestión de maniobras directamente en la plataforma de monitoreo, mejorando el seguimiento y la coordinación de actividades.

ANEXO 7: Resumen de la entrevista realizada al líder de la plataforma técnica en formato virtual sobre el plan de acción del ciclo 1

1. Feedback sobre el mapa de dependencias:

- Dinamismo de tipos de objetos: Si en el futuro surgen nuevos tipos de objetos, hay que modificar el algoritmo para entender cómo este nuevo tipo se relaciona con los otros tipos. Es importante tener en cuenta este aspecto para asegurar la flexibilidad del sistema a futuros cambios. Sugiere definir reglas que describan las relaciones entre los tipos de objeto, de manera que si surge un nuevo tipo, simplemente se deban agregar nuevas reglas y no modificar el algoritmo en sí.

- Agrupadores por ambiente: Se sugiere crear agrupadores específicos para cada ambiente (desarrollo, test, producción). Actualmente los servidores tienen información sobre el ambiente en su nombre, por lo que se puede considerar para agrupar. Por ej, un nombre de un servidor de desarrollo tiene la estructura: [prefijo para diferenciar si es físico o virtual][tecnología o aplicación instalada][ambiente][número diferenciador].

- Ampliación de información:

  Una propuesta interesante es la creación de un mecanismo en el que cualquier usuario pueda agregar más información al árbol sin modificar el algoritmo o los objetos en la herramienta. De esta forma se crea de manera colectiva una gran fuente de datos que enriquecerá el potencial del mapa de dependencias.

  Esto solucionaría el problema de los servidores de base de datos, que actualmente tienen el entorno “BD”, y no se pueden asociar automáticamente al entorno de una solución. Esta información que no se puede inferir automáticamente, puede ser agregada a la nueva herramienta por un usuario más adelante.

  Con este mecanismo se podría visualizar y consultar de forma rápida información necesaria antes de un cambio, antes de una compra, o por obsolescencia.

Algunos ejemplos serían:

- Asociar la versión de sistema operativo a cada objeto servidor para obtener los servidores afectados ante el fin de soporte de una versión de sistema operativo.

- Asociar la compra a cada solución tecnológica, para obtener los objetos afectados antes de que termine el contrato.

- Independencia de la herramienta de monitoreo interna:

  Para el mapa de dependencia, propone desarrollar una aplicación independiente a la herramienta de monitoreo. De esta manera, no se modifica nada de la herramienta de monitoreo actual.

  Esta nueva herramienta consumiría los datos de la herramienta de monitoreo actual (y también datos externos de usuarios) de una forma normalizada, y a través de un algoritmo y reglas definidas se genera el árbol.

  En el caso que la herramienta interna de monitoreo detecte alguna actualización de estado (por ej: una indisponibilidad) en alguno de sus objetos, se le enviará un aviso a la herramienta del mapa de dependencias, para que esta vuelva a renderizar el árbol.

  En esta aplicación, los árboles de dependencia no serán estáticos, sino que se puede ejecutar cada vez que se necesite.

  Propone utilizar una base de datos de grafos para la generación del árbol de dependencias. Por ej: Neo4j (https://neo4j.com/).

  Sugiere que la interfaz gráfica de la nueva herramienta sea accesible directamente desde la herramienta actual. De esta manera, se garantizaría una integración fluida para los usuarios, permitiéndoles interactuar con la nueva funcionalidad sin necesidad de salir del entorno en el que ya trabajan.

- Persistencia y comparación de versiones: Es fundamental garantizar que los datos y registros se conserven como “snapshots” a lo largo del tiempo para poder realizar comparaciones entre versiones anteriores y actuales de la infraestructura.

2. Feedback sobre la automatización de la notificación de maniobras:

A la propuesta del envío automático de un correo con todos los detalles relevantes, propone los siguientes puntos:

- En vez de enviar automáticamente el correo al equipo de operaciones para dar aviso de maniobras planificadas, generar un borrador del correo con la información de la maniobra a planificar para poder previsualizarlo y modificarlo antes del envío y enviarlo a los equipos afectados y/o interesados del conocimiento de la maniobra.

- Establecer un sistema de suscripción a objetos podría ser beneficioso para notificar automáticamente cualquier cambio o maniobra que impacte a esos objetos, mejorando la visibilidad y la capacidad de respuesta ante incidentes.

ANEXO 8: Resumen de la entrevista realizada al líder del equipo encargado de la administración de los servidores en formato virtual sobre el plan de acción del ciclo 1

1. Feedback sobre el mapa de dependencias:

Para mejorar la comprensión del mapa de dependencias, se podría considerar integrar filtros avanzados y opciones de búsqueda. Esto permitiría a los usuarios buscar objetos por nombre, tipo, relación o entorno, optimizando la navegación en mapas de gran tamaño.

La propuesta menciona notificaciones, pero sería beneficioso expandir para que se integren con sistemas de gestión de incidentes o plataformas de monitoreo adicionales, como herramientas de alertas, para facilitar la respuesta inmediata ante un cambio o fallo en la infraestructura.

Sería provechoso desarrollar una funcionalidad de simulación que permita realizar análisis predictivos para evaluar el impacto de posibles cambios en la infraestructura (por ejemplo, actualizaciones de sistemas operativos, migraciones de bases de datos o cambios en la configuración de redes). Esto permitiría prever y mitigar problemas antes de que ocurran.

Además de la validación de relaciones, la herramienta debería ser capaz de identificar inconsistencias o posibles errores en la estructura de dependencias (por ejemplo, objetos huérfanos mal conectados o dependencias cíclicas) y notificar a los administradores para que puedan corregirlas antes de que causen problemas mayores.

2. Feedback sobre la automatización de la notificación de maniobras:

Con respecto a la automatización de maniobras, se podría enviar un correo automáticamente con el aviso de maniobra planificada y los objetos involucrados a los equipos técnicos previamente definidos.

Además, se podría añadir un sistema que permita realizar un seguimiento de las notificaciones de maniobras, incluyendo un historial accesible para los usuarios. Esto no solo permitiría saber qué se notificó y cuándo, sino también verificar si se tomaron medidas correctivas en función de las notificaciones anteriores.

Como mejora a futuro, se podría implementar una funcionalidad que permita que, cuando un objeto cambie de estado (por ejemplo, sea actualizado, reparado, descontinuado o reemplazado), se envíen notificaciones automáticas no solo a los equipos operativos, sino también a los usuarios que podrían verse impactados por el cambio (desarrolladores, técnicos, equipos de QA, etc.). Esto ayudaría a mantener a todas las partes interesadas informadas en tiempo real.

ANEXO 9: Plan de acción correspondiente al ciclo 2 presentado a los actores clave para su retroalimentación.

Problema

El objetivo de este estudio es analizar las deficiencias en la gestión de cambios en servidores Windows dentro de la organización, identificando los factores que generan demoras en el proceso, fallos inesperados y afectaciones en la continuidad del servicio.

A partir de las entrevistas realizadas y los relevamientos realizados sobre el proceso de cambios actual y la herramienta de monitoreo interna, se identificaron los siguientes problemas:

1. Ausencia de un mapeo confiable de dependencias entre los elementos de la infraestructura: No existe un registro claro y confiable de las relaciones entre los objetos, lo que dificulta una gestión eficiente.

2. Falta de automatización en el árbol de dependencias: Actualmente, el árbol de dependencias se define de forma manual, lo que lo convierte en un proceso engorroso y poco frecuente, resultando rápidamente obsoleto.

3. Información dispersa y no estandarizada: La información sobre las relaciones entre los objetos no está normalizada y se encuentra distribuida en varias fuentes y equipos, lo que complica su acceso y análisis.

4. Falta de visibilidad del impacto de los cambios: Cuando se realiza un cambio en un objeto, no se tiene claridad sobre qué otros objetos pueden verse afectados.

5. Falta de automatización en la comunicación de maniobras: Las notificaciones se envían manualmente vía mail, lo que genera demoras y posibles errores en el proceso de cambios.

A partir de las entrevistas realizadas, se han identificado los siguientes requisitos funcionales y técnicos para la solución propuesta:

1. Automatización y visualización de dependencias: La solución debe integrar y mostrar las relaciones entre servidores, aplicaciones y bases de datos, facilitando la gestión de cambios y su impacto.

2. Estandarización de monitoreo: Es necesario establecer un proceso que normalice y consolide los objetos involucrados en el monitoreo, garantizando coherencia y precisión en la información.

3. Creación de un mapa de dependencias confiable: Aunque no sea completamente preciso, debe representar fielmente la infraestructura actual, proporcionando visibilidad y control.

4. Integración sin impacto significativo: La solución debe incorporarse sin alterar drásticamente la forma en que se utiliza la herramienta actual, evitando problemas en la interfaz y minimizando la necesidad de capacitación adicional.

5. Continuidad con la herramienta de monitoreo existente: Dado que forma parte de la cultura organizacional y está profundamente integrada en los procesos diarios, no se considera viable reemplazarla por una herramienta externa (como Zabbix, Nagios, etc).

Propuesta de solución

Considerando los siguientes aspectos:

- La generación del árbol de dependencias debe ser completamente automática **e** independiente del sistema de monitoreo actual.

- Otras herramientas utilizan etiquetas o relaciones "padre/hijo" para definir dependencias o jerarquías.

- No se desea agregar nuevos campos durante la creación de un objeto en la herramienta interna.

- Cada objeto ya cuenta con los campos definidos: entorno, tipo, nombre, estado y equipo responsable.

- En la solución de estudio se administran únicamente objetos con los siguientes tipos:
    - SER: Monitorea la disponibilidad de servidores.
    - DRV: Monitorea el espacio en los discos.
    - WEB: Monitorea la disponibilidad de sitios web.
    - SRV: Monitorea la ejecución de servicios de Windows.
    - CER: Monitorea el vencimiento de certificados.
    - BD: Monitorea la instancia SQL.
- **Si se crea un nuevo tipo de objeto en el futuro, el algoritmo de generación del árbol de dependencias no debería verse afectado. El algoritmo debe estar abierto a la extensión y cerrado a la modificación.**
- El sistema de nombres actual de los objetos tiene una estructura definida.
    - Servidor (SER): [NombreServidor]_(aviso|alarma)
    - Discos (DRV): [NombreServidor]*[C|NombreDisco]*(aviso|alarma)
    - Sitios web (WEB): [NombreServidor]_(aviso|alarma)_URL
    - Servicios Windows (SRV): [NombreServidor]_[NombreServicio], por ejemplo, DNS o algún otro programa.
    - Certificados (CER): [NombreServidor]_[NombreCertificado]
    - Tareas (OCO): [NombreServidor]_[NombreTarea]
    - Instancia SQL (BD): [NombreServidor]_bd
- El [NombreServidor] tiene la estructura [prefijo para diferenciar si es físico o virtual][tecnología o aplicación instalada][ambiente][número diferenciador], lo que permite la diferenciación por ambientes.

- Actualmente, las notificaciones sobre maniobras planificadas se realizan de forma manual.

Se propone:

1. Implementación de un nuevo algoritmo de generación de árboles de dependencias a partir de datos normalizados, independiente a la herramienta de monitoreo actual. A partir de la información de los objetos obtenida en esta herramienta, los datos se normalizan y el algoritmo infiere automáticamente las relaciones entre ellos. A pesar que el algoritmo es independiente a la herramienta interna de monitoreo, el árbol se podrá visualizar en ella para no generar fricción en la usabilidad habitual de la plataforma.

Funcionamiento**:**

- Agrupación por entorno: Se creará un mapa específico para cada entorno definido.
- Estructura del mapa:
    - Cada mapa incluirá un nodo raíz de tipo "SISTEMA", que representará el entorno.
    - Si es necesario, se generarán nodos de tipo “AGRUPADOR” para organizar servidores con nombres similares o relacionados. Por ejemplo, si existen servidores ser1, ser2, serDB1, serDB2, se crearán dos agrupadores: uno para los "ser" y otro para los "serDB".
- Relaciones entre objetos:
    - Se utilizará la base de datos de grafos Neo4j para representar las relaciones entre los objetos.
    - Se creará una estructura de reglas de dependencia que definen cómo se relacionan dos tipos de objetos. Por cada tipo de objeto se deberá definir una regla que indique las condiciones que debe tener otro objeto para ser su padre. Por ejemplo, existirá una regla que indique que un objeto tipo

"DRV" tendrá como padre a un objeto de tipo "SER" cuyo nombre coincida con el comienzo del suyo. De esta manera si se crea un nuevo tipo de objeto en el futuro, sólo se deberá definir una nueva regla, y el algoritmo podrá relacionarlo correctamente con los otros objetos. Por defecto se crearán las siguientes reglas:

- Los objetos de tipo "SISTEMA" no tienen un nodo padre.
- Los objetos de tipo "SER" pueden tener como padre al objeto "SISTEMA" o a un "AGRUPADOR", dependiendo si existen más objetos de tipo "SER" con nombre similar (por ejemplo: [nombreServidor]1, [nombreServidor]2, [nombreServidor]3).
- Todos los objetos de tipo "AGRUPADOR" tendrán como padre al objeto de tipo "SISTEMA".
- Todos los objetos de los tipos "DRV", "WEB", "CER", "SVC", "OCO" y "DB" tendrán como padre al objeto de tipo "SER" cuyo nombre coincida con el comienzo del suyo.

Para los servidores de base de datos (objetos de tipo "SER" con un hijo de tipo "BD") no existirá una regla definida por defecto, ya que un servidor de base de datos puede estar relacionado con muchas soluciones diferentes, por lo que no tendrá un padre único. Además, esta información no se puede inferir automáticamente por el momento ya que su entorno siempre es "BD" y su nombre no tiene una estructura que nos brinde la información suficiente. Es por esto que surge la necesidad de poder crear relaciones directas entre objetos. Para esto se define un nuevo tipo reservado llamado "RELACION_DIRECTA" en el que cualquier usuario podrá definir simplemente que el objeto X tiene de padre al objeto Y.

- Nomenclatura de objetos BD:
  - El equipo encargado de controlar las bases de datos, debe crear sus objetos con la siguiente nomenclatura:

        - [NombreServidorDB]_bd para el objeto de tipo BD que controla la disponibilidad de la instancia de SQL.

        - [NombreServidorDB]_[NombreDisco]_[alarma|aviso] para los objetos de tipo DRV

    - Todos los objetos que contengan en su nombre [NombreServidorDB] estarán relacionados con el objeto de tipo SER con ese nombre.

- Objetos huérfanos:

    - Aquellos objetos cuyo nombre no coincida parcialmente con el de un servidor dentro del mismo entorno, se mostrarán como elementos aislados en el mapa.

Este enfoque permitirá una generación automática y estructurada de mapas de dependencias, optimizando la visualización y gestión de los objetos en la infraestructura.

2. Mejora en el proceso de maniobras planificadas

Actualmente, cuando se crea una maniobra, el equipo responsable debe notificar manualmente al equipo de operaciones mediante un correo electrónico. Se propone:

1. Generar automáticamente un borrador del correo de aviso de maniobra con la información necesaria para TI. De esta manera se podrá revisar o modificar su contenido y luego se envía automáticamente desde la plataforma.

2. Envío de un correo automático a los equipos técnicos involucrados en la maniobra con la información de todos los objetos que pueden verse afectados.

Conclusión

La implementación de estos dos cambios permitirá agilizar el proceso de gestión de cambios al:

- Automatizar las notificaciones de maniobras, eliminando la necesidad de envíos manuales y reduciendo demoras en la comunicación.

- Identificar de manera proactiva los objetos afectados ante una maniobra o indisponibilidad, permitiendo detectar servidores, bases de datos y aplicaciones impactadas por cambios en la infraestructura y mejorando la capacidad de respuesta ante incidentes.

- Comprender las relaciones de dependencia entre elementos, facilitando la evaluación de riesgos y la adopción de medidas preventivas para minimizar fallos.

- Brindar mayor visibilidad sobre la disponibilidad de los objetos a usuarios no técnicos, como el equipo de CAU y operaciones, quienes podrán consultar el mapa de dependencias para verificar la restauración de sistemas sin depender del equipo técnico responsable.

ANEXO 10: Resumen de la entrevista realizada al líder de la plataforma técnica sobre el plan de acción del ciclo 2.

1. Feedback sobre el mapa de dependencias:

Quizás, para que nodo queden todos los sistemas sueltos… sin padre… el grafo puede tener una RAIZ que se llame "ARBOL" y que sus hijos sean los sistemas… así queda más alineado a la teoría de grafos como definición de árbol.

Estaría bueno pulir algunas definiciones.

- Objetos huerfanos: Formalizando, son aquellos nodos sin predecesores ni sucesores, o sea no tienen aristas entrantes ni salientes. Identificarlos permitirá detectar fallos o desviaciones del estándar definido para la generación automática del árbol de dependencias, facilitando la corrección de la nomenclatura. Es decir, su propietario podrá ajustar el nombre correspondiente, lo que permitirá iterar nuevamente el proceso y obtener una nueva versión del árbol con información más completa y precisa.
- Maniobra: Se define como maniobra cualquier actividad, ya sea proactiva o correctiva, realizada sobre uno o varios objetos y que pueda generar una indisponibilidad temporal (maniobra disruptiva) o el riesgo de que esto ocurra (maniobra no disruptiva). Por ello, al trabajar con el árbol de dependencias, si nos enfocamos en los objetos involucrados, es altamente factible inferir qué sistemas y equipos se verán afectados mediante algoritmos basados en teoría de grafos.

Con respecto a las conclusiones, las reordernaría segun importancia.

La implementación de estos dos cambios permitirá agilizar el proceso de gestión de cambios al:

- Identificar de manera proactiva los objetos afectados ante una maniobra o indisponibilidad, permitiendo detectar servidores, bases de datos y aplicaciones impactadas por cambios en la infraestructura y mejorando la capacidad de respuesta ante incidentes.

- Comprender las relaciones de dependencia entre elementos, facilitando la evaluación de riesgos y la adopción de medidas preventivas para minimizar fallos.
- Automatizar las notificaciones de maniobras, eliminando la necesidad de envíos manuales y reduciendo demoras en la comunicación.
- Brindar mayor visibilidad sobre la disponibilidad de los objetos a usuarios no técnicos, como el equipo de CAU y operaciones, quienes podrán consultar el mapa de dependencias para verificar la restauración de sistemas sin depender del equipo técnico responsable.

ANEXO 11: Resumen de la entrevista realizada al líder del equipo encargado de la administración de los servidores sobre el plan de acción del ciclo 2.

El plan de acción está muy detallado y claro.

¿Pensaron en crear un proceso para garantizar que la información relacionada se mantenga actualizada y centralizada?

Como aspecto a implementar a futuro, estaría bueno incluir un plan de pruebas detallado para la integración y un enfoque de capacitación para garantizar que los usuarios estén familiarizados con el nuevo sistema antes de la implementación completa.