**UNIVERSIDAD HISPANOAMERICANA**

**ESCUELA DE INFORMÁTICA**

**BACHILLERATO**

# PROPUESTA DE MEJORA AL PROCESO AUTOMATICO DE PARCHEO PARA SERVIDORES RED HAT EN EL DEPARTAMENTO DE EITS DE LA EMPRESA EXPERIAN COSTA RICA PARA EL 2024

**Sustentante:**

**Manuel Salazar González**

**Tutor:**

**Estrellita Jenkins Miranda**

**Agosto, 2024**

## Tabla de Contenidos

# Índice de Tablas

## Índice de Ilustraciones

## Declaración Jurada

### DECLARACIÓN JURADA

Yo Manuel de Jesús Salazar González, mayor de edad, portador de la cédula de identidad número 206830213 egresado de la carrera de Ingeniería Informática de la Universidad Hispanoamericana, hago constar por medio de éste acto y debidamente apercibido y entendido de las penas y consecuencias con las que se castiga en el Código Penal el delito de perjurio, ante quienes se constituyen en el Tribunal Examinador de mi trabajo de tesis para optar por el título de Bachiller, juro solemnemente que mi trabajo de investigación titulado: Propuesta de mejora al proceso automático de parcheo para servidores Red Hat en el departamento de EITS de la empresa Experian Costa Rica para el 2024, es una obra original que ha respetado todo lo preceptuado por las Leyes Penales, así como la Ley de Derecho de Autor y Derecho Conexos número 6683 del 14 de octubre de 1982 y sus reformas, publicada en la Gaceta número 226 del 25 de noviembre de 1982; incluyendo el numeral 70 de dicha ley que advierte; articulo 70. Es permitido citar a un autor, transcribiendo los pasajes pertinentes siempre que éstos no sean tantos y seguidos, que puedan considerarse como una producción simulada y sustancial, que redunde en perjuicio del autor de la obra original. Asimismo, quedo advertido que la Universidad se reserva el derecho de protocolizar este documento ante Notario Público. en fe de lo anterior, firmo en la ciudad de San José, a los 27 días del mes de junio del año dos mil veinticuatro.

Firma del estudiante

Cédula 206830213

**Carta de Aprobación de Tutor**

# CARTA DEL TUTOR

San José, 8 de agosto del 2024

**Kattia Huertas**
**Directora**
**Ingeniería Informática**
**Universidad Hispanoamericana**
**Sede Llorente**

Estimada señora:

El estudiante Manuel De Jesús Salazar González **cédula** de identidad número 206830213, me ha presentado, para efectos de revisión y aprobación, el trabajo de investigación denominado **"PROPUESTA DE MEJORA AL PROCESO AUTOMATICO DE PARCHEO PARA SERVIDORES RED HAT EN EL DEPARTAMENTO DE EITS DE LA EMPRESA EXPERIAN COSTA RICA PARA EL 2024 ",** el cual ha elaborado para optar por el grado académico de Bachiller en Ingeniería Informática.

En mi calidad de tutor, he verificado que se han hecho las correcciones indicadas durante el proceso de tutoría y he evaluado los aspectos relativos a la elaboración del problema, objetivos, justificación; antecedentes, marco teórico, marco metodológico, tabulación, análisis de datos; conclusiones y recomendaciones.

De los resultados obtenidos por el postulante, se obtiene la siguiente calificación:

| | | |
|---|---|---|
| a) Original del tema | 10% | 10% |
| b) Cumplimiento de entrega de avances | 20% | 20% |
| c) Coherencia entre los objetivos, los instrumentos aplicados y los resultados de la investigación | 30% | 30% |
| d) Relevancia de las conclusiones y recomendaciones | 20% | 20% |
| e) Calidad, detalle del marco teórico | 20% | 20% |
| **TOTAL** | | **100%** |

En virtud de la calificación obtenida, se avala el traslado al proceso de lectura.

Atentamente,

**ESTRELLITA JENKINS MIRANDA**

**Cédula 16250359**

Estrellita Jenkins Miranda

Firmado digitalmente por Estrellita Jenkins Miranda
Fecha: 2024.08.08 22:33:39 -06'00'

## Carta de Aprobación del Lector

### CARTA DE LECTOR

**San José, 10 de Diciembre de 2024.**

**Universidad Hispanoamericana**
**Sede Llorente**
**Carrera**

**Estimada señora**

El estudiante **MANUEL SALAZAR GONZALEZ**, cédula de identidad **206830213**, me ha presentado para efectos de revisión y aprobación, el trabajo de investigación denominado **"PROPUESTA DE MEJORA AL PROCESO AUTOMATICO DE PARCHEO PARA SERVIDORES RED HAT EN EL DEPARTAMENTO DE EITS DE LA EMPRESA EXPERIAN COSTA RICA PARA EL 2024"**, el cual ha elaborado para obtener su grado de **BACHILLERATO**.

He revisado y he hecho las observaciones relativas al contenido analizado, particularmente lo relativo a la coherencia entre el marco teórico y análisis de datos, la consistencia de los datos recopilados y la coherencia entre éstos y las conclusiones; asimismo, la aplicabilidad y originalidad de las recomendaciones, en términos de aporte de la investigación. He verificado que se han hecho las modificaciones correspondientes a las observaciones indicadas.

Por consiguiente, este trabajo cuenta con mi aval para ser presentado en la defensa pública.

Atte.

MARCO VINICIO SOTO MONGE (FIRMA)

Firmado digitalmente por MARCO VINICIO SOTO MONGE (FIRMA)
Fecha: 2024.12.10 20:49:41 -06'00'

**Marco Vinicio Soto Monge**
**110360428**
**CARNET: 4720**

## Carta de Autorización de Publicación

**UNIVERSIDAD HISPANOAMERICANA**
**CENTRO DE INFORMACION TECNOLOGICO (CENIT)**
**CARTA DE AUTORIZACIÓN DE LOS AUTORES PARA LA CONSULTA, LA REPRODUCCION PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DE LOS TRABAJOS FINALES DE GRADUACION**

San José, 15 de enero de 2025

Señores:
Universidad Hispanoamericana
Centro de Información Tecnológico (CENIT)

Estimados Señores:

El suscrito (a) Manuel de Jesús Salazar González con número de identificación 206830213 autor (a) del trabajo de graduación titulado PROPUESTA DE MEJORA AL PROCESO AUTOMATICO DE PARCHEO PARA SERVIDORES RED HAT EN EL DEPARTAMENTO DE EITS DE LA EMPRESA EXPERIAN COSTA RICA PARA EL 2024 presentado y aprobado en el año 2025 como requisito para optar por el título de Bachillerato; (SI) autorizo al Centro de Información Tecnológico (CENIT) para que con fines académicos, muestre a la comunidad universitaria la producción intelectual contenida en este documento.

De conformidad con lo establecido en la Ley sobre Derechos de Autor y Derechos Conexos N° 6683, Asamblea Legislativa de la República de Costa Rica.

Cordialmente,

206830213

Firma y Documento de Identidad

## Dedicatoria

A mi maravillosa esposa, quien ha sido mi apoyo mas grande en la realización, seguimiento y finalización de este proceso y de mis estudios. Quien no ha dudado en arme las fuerzas que necesito para finalmente completar esta etapa en mi desarrollo académico,

A mis padres y a mi familia que siempre están apoyándome, y a quienes no les puedo agradecer lo suficiente por todo lo que han hecho por mí.

A mi Dios quien siempre nos bendice a mi persona y a mi familia.

## Agradecimiento

A mi Dios padre, a mi familia y a mi maravillosa esposa quienes les agradezco con mi vida por todo lo que he logrado en mi vida.

## Resumen

En la actualidad, la automatización del parcheo en servidores se ha vuelto crucial para mantener la seguridad y el rendimiento de los sistemas operativos, especialmente en entornos empresariales donde los servidores manejan datos sensibles y requieren una alta disponibilidad. El parcheo automatizado permite la actualización oportuna de software, minimizando riesgos de vulnerabilidades y mejorando la resiliencia del sistema ante amenazas de seguridad.

En este trabajo se propone una mejora al proceso de parcheo automático en servidores Linux Red Hat, abordando las brechas del sistema actual a través de un análisis exhaustivo de su infraestructura y la evaluación de plataformas alternativas que puedan optimizar el ciclo de parcheo. El estudio inicia con una revisión detallada de la plataforma de parcheo actual y continúa con el análisis de opciones que podrían ofrecer mayores beneficios en términos de seguridad, eficiencia y control.

Al inicio, abordaremos detalles de la empresa en la que se realizara este proyecto de investigación, y se darán a conocer la problemática y el problema que será el centro de este trabajo.

Seguidamente, se detallarán los conceptos generales, así como tecnologías o plataformas que están, o estarán involucradas en el proceso que permitan llegar a una conclusión que satisfactoria. Esto en base a las necesidades o brechas que se identificaran del proceso realizado actualmente.

Finalmente se presentarán distintas propuestas con el objetivo de compararlas y seleccionar la que presenta los mayores beneficios a la empresa para resolver el problema anteriormente expuesto.

# Capítulo I: Problema Del Proyecto.

## 1.1 Antecedentes y Justificación del Proyecto.

El presente proyecto tiene como objetivo principal implementar un proceso automatizado de parcheo para servidores Linux Red Hat en Experian Costa Rica. Esta iniciativa surge como respuesta a la necesidad de fortalecer la seguridad informática de la empresa, optimizar el trabajo del equipo de Entreprise Information Technology Services, al que nos referiremos de ahora en adelante como EITS, y mitigar las vulnerabilidades presentes en los sistemas operativos.

A continuación, se detallarán los antecedentes del proyecto, la problemática que se busca abordar, los objetivos específicos a alcanzar, el alcance y las limitaciones del proyecto y el cronograma de actividades.

### *1.1.1 Antecedentes del contexto de la empresa.*

**Experian: Una empresa global con trayectoria**

Experian, fundada en Nottingham, Inglaterra en 1980, tiene una larga trayectoria de más de 40 años en la industria de la información y servicios financieros. Con presencia en más de 30 países, Experian ayuda a empresas y consumidores a prosperar a través de su estrategia centrada en:

- Facilitar el acceso al crédito y préstamos.
- Empoderar a los consumidores a mejorar su salud financiera.
- Combatir el fraude y proteger la identidad.
- Tomar decisiones inteligentes basadas en datos.
- Fortalecer las relaciones con los clientes.

**Objetivos:** De acuerdo con Experian **(2024)**, los objetivos de la empresa son:

- Mejorar la salud financiera para todos. Alcanzar a 100 millones de personas a través de productos de innovación para el 2025.
- Diversidad. Incrementar la proporción de mujeres en el comité ejecutivo en un 30%, como lideres senior en un 40%, en los niveles medios en un 42% y el total de la fuerza laboral en un 47% para el 2024.
- Conservación ambiental. Convertir todas las operaciones propias en carbono neutral para el 2030.

**Misión:** Impulsar la inclusión financiera y facilitar el acceso justo a créditos costeables para los consumidores. Para lograrlo, colocamos a nuestros clientes al frente de nuestra estrategia de negocio. Exploramos de manera continua nuevas formas de usar nuestros datos y recursos para mejorar el bienestar financiero. **(Experian, 2024)**

**Valores:** Con base en Experian **(2024)** los valores de la empresa son:

1. Trabajo con integridad. La confianza no se da, se gana; por esto Experian está comprometido a trabajar con integridad y operar responsablemente en cada área de su negocio.
2. Inversión en la comunidad. Experian está comprometido a invertida su tiempo, recursos y asociaciones en crear un mejor futuro para la comunidad.
3. Desbloquear el poder de la información. La información tiene la capacidad de transformar la vida para mejor, por lo que Experian implementa nuevas tecnologías e innovaciones para hacerlo realidad.

4. Valorar a su gente. Experian está comprometido en apoyar y reclutar talento diverso para innovar el negocio, fortalecer las comunidades y empoderar a los clientes.

**Negocio al que se dedica**: Experian es el líder mundial en proveer información, servicios analíticos y financieros a organizaciones y clientes para asistirles en manejar el riesgo y recompensa de las decisiones comerciales y financieras.

Según Experian (2024) los 4 principales grupos de negocio son:

1. Servicios crediticios. Provee asistencia con la evaluación de los riesgos o recompensas asociados a proveer crédito a clientes o negocios, lo que permite tomar decisiones informadas haciendo el proceso más rápido y fácil para las partes involucradas.
2. Análisis de decisiones. Aumenta la velocidad y calidad de la toma de decisiones mediante habilidades analíticas y especialistas en productos de software.
3. Servicios de mercadeo. Experian Marketing Suite permite a los negocios mejorar su poder adquisitivo, la lealtad de los clientes y el retorno de inversión.
4. Servicios al consumidor. Provee acceso en línea y seguro a los consumidores a su historial crediticio, dándoles las herramientas necesarias para administrar y mejorar su estado financiero.

En Costa Rica, Experian es el líder en el reporte crediticio de consumidores y empresas. Cuenta con más de 1300 colaboradores, brinda a diferentes mercados servicios que incluyen servicio al cliente, finanzas, análisis de decisiones, servicios de información empresarial, calidad de datos, tecnologías de la información, entre otros.

**Historia de la Organización**: Tal y como se menciona en CINDE **(2023)** Experian Costa Rica inicio operaciones en el 2008 atraída por el talento costarricense, estabilidad política y cercanía a sus mayores clientes en Norteamérica, estableciendo uno de sus 4 centros de operaciones globales que adicionalmente brindan soporte a América Latina y Europa.

En los últimos años las exportaciones costarricenses de servicios basados en conocimiento alcanzaron $6 412.8 millones, representando un crecimiento en el sector de más del 12%.

Además del impacto económico con el que Experian ha colaborado en este crecimiento, también cabe destacar el presentar a Costa Rica como el líder de manera amplia de la exportación de servicios en Latinoamérica. Este éxito está basado en la alta especialización de talento humano, las habilidades digitales y el dominio del inglés.

### 1.1.2 Justificación del proyecto.

Experian Costa Rica alberga un centro de entrega global en el que consolida la mayoría de las unidades de negocio de la empresa, manteniéndose como líder de la región y aliado estratégico de las organizaciones. Esto por el personal altamente capacitado en el país y las políticas de la empresa para administrar su negocio, basada en la confianza que proyecta y provee a sus clientes.

La cultura profesional en Experian se enfoca en la confianza de los clientes y socios como uno de sus mayores activos al ser una empresa de datos, tal y como lo establece Experian (2023), la empresa maneja la información de 1.3 billones de personas y 166 millones de negocios globalmente, por lo que es esencial mantener la confianza de sus colaboradores, socios y clientes. Asegurase de obtener, almacenar y administrar los datos de manera responsable y segura es fundamental para su éxito.

El departamento de ETIS, impulsado por esta cultura profesional, tiene la responsabilidad de asegurar que los servidores que forman parte de la infraestructura que obtiene, almacena y administra la información propia de la empresa y sus clientes, se encuentren al día con las actualizaciones de seguridad en los sistemas operativos y últimas versiones de los agentes utilizados para diferentes propósitos.

Los ataques a activos informáticos pueden provocar la pérdida de disponibilidad, confidencialidad o integridad de la información; lo que suele implicar graves consecuencias para las empresas y a menudo se ocasionan daños irreparables. (Montesino, Baluja, & Porvén, 2013, p. 1) Así que se identifica la importancia y múltiples factores que resaltan la necesidad de mantener los servidores actualizados y solventar las vulnerabilidades de la evolución del software que puedan comprometer la información.

Para hacer frente a este desafío, el grupo de EITS busca una propuesta para la aplicación programada de parches de seguridad y actualizaciones de agentes a los servidores Linux Red Hat de su infraestructura. *Las actualizaciones para aplicaciones y sistemas operativos dentro de una organización es uno de los principales requisitos para cumplir con los estándares de seguridad, los principales ataques a organizaciones se dan principalmente cuando se tiene sistemas operativos obsoletos, software no actualizado ya que generalmente tienden a tener bugs o agujeros de seguridad que permite a un atacante aprovecharse de esta vulnerabilidad.* (Meza & Zambrano, 2018, p. 2)

Según  Zaidman (2017) , es fundamental dirigir los parches o actualizaciones de seguridad hacia la minimización del tiempo de exposición a amenazas, con el objetivo de reducir el riesgo y fortalecer la seguridad de la infraestructura. operativo. Esto resalta la importancia de realizar un proceso constante de parcheo en los servidores, implementando actualizaciones diseñadas para enfrentar las amenazas identificadas, para prevenir o mitigar su impacto en el entorno operativo.

Implementar una solución automatizada para la aplicación de los parches y actualizaciones de seguridad más recientes permitiría al equipo de EITS gestionar y prevenir de manera más eficiente la seguridad de los servidores Red Hat. Esto resultaría en una disminución proporcional de las vulnerabilidades detectadas en los servidores que requieran intervención manual por parte del equipo de EITS, lo que a su vez aumentaría la eficiencia y productividad del equipo de trabajo y reduciría el riesgo de ataques informáticos, aportando valor a la organización. Según Montesino, Baluja & Porvén (2013), una estrategia efectiva para simplificar y fortalecer la administración de la seguridad informática en un entorno dinámico y lleno de amenazas en constante evolución es la automatización de los controles de seguridad.

## 1.2 Definición del Problema.

### *1.2.1 Problemática*.

El proyecto se realizará en Experian Costa Rica para presentar una propuesta para el parcheo automático de paquetes del sistema operativo y agentes requeridos por el equipo de soporte según los estándares de seguridad, automatización y mejores prácticas, definidos y reconocidos por las normas e instituciones a nivel global dentro de la administración de negocios y proyectos. Esto para mitigar las vulnerabilidades de los servidores manteniéndolos al día con sus actualizaciones y de manera transparente para los clientes.

En la actualidad el equipo de EITS realiza la actualización de los paquetes de Red Hat mediante una plataforma que se encuentra en el final de su ciclo de vida dentro de la empresa y cuyo soporte esta por finalizar. La metodología actual tiene incumplimientos en el área de seguridad y control de acceso a la calendarización de los servers, y una forma no muy eficiente de reportar y detallar los problemas del parcheo. Estas carencias recargan al equipo de EITS con trabajo redundante o que no es parte del soporte del equipo, generando incidentes por vulnerabilidades y un incremento en número de incidentes o casos que no necesariamente son

soportados por el equipo. La documentación de las limitaciones de esta plataforma y los problemas causados por estas han permitido definir una lista de requerimientos que debe resolver la nueva propuesta.

Estos requerimientos tienen el fin de cumplir con el objetivo del equipo de EITS de reducir la carga de trabajo doble o mal asignado, que no es parte del soporte proporcionado y a su vez documentar de manera apropiada los casos o incidentes que si deben ser resueltos por el equipo de EITS para facilitar su resolución, detección de patrones, identificación de problemas mayores , identificación de áreas de mejora dentro del proceso y aplicación de una tecnología más reciente y fácil de aprender, implementar y administrar.

La organización tiene problemas para definir la solución o propuesta que cumpla con los requerimientos documentados, entre los que se pueden citar el control de acceso a la calendarización de servidores, documentación de los errores del proceso, correcta asignación de incidentes según su naturaleza, separación de deberes, roles y responsabilidades, escalabilidad de la plataforma y ruta de escalación para reportar problemas de la plataforma. El incumplimiento de estos requisitos por parte de la plataforma actual resulta en el incremento de vulnerabilidades detectadas en los servidores, generando trabajo adicional para resolverlas manualmente, que a su vez requiere coordinación con los clientes para obtener ventanas de mantenimiento adicionales para parchear los servidores de producción, lo que conlleva a atrasos de semanas o incluso meses para resolver incidentes parcheo o solicitudes de parcheo de vulnerabilidades específicas.

Adicionalmente la falta de documentación y manuales de operación limitan la capacidad del equipo de EITS de comprender, solucionar o mejorar los procesos en la plataforma actual para resolver los posibles incidentes de manera preventiva.

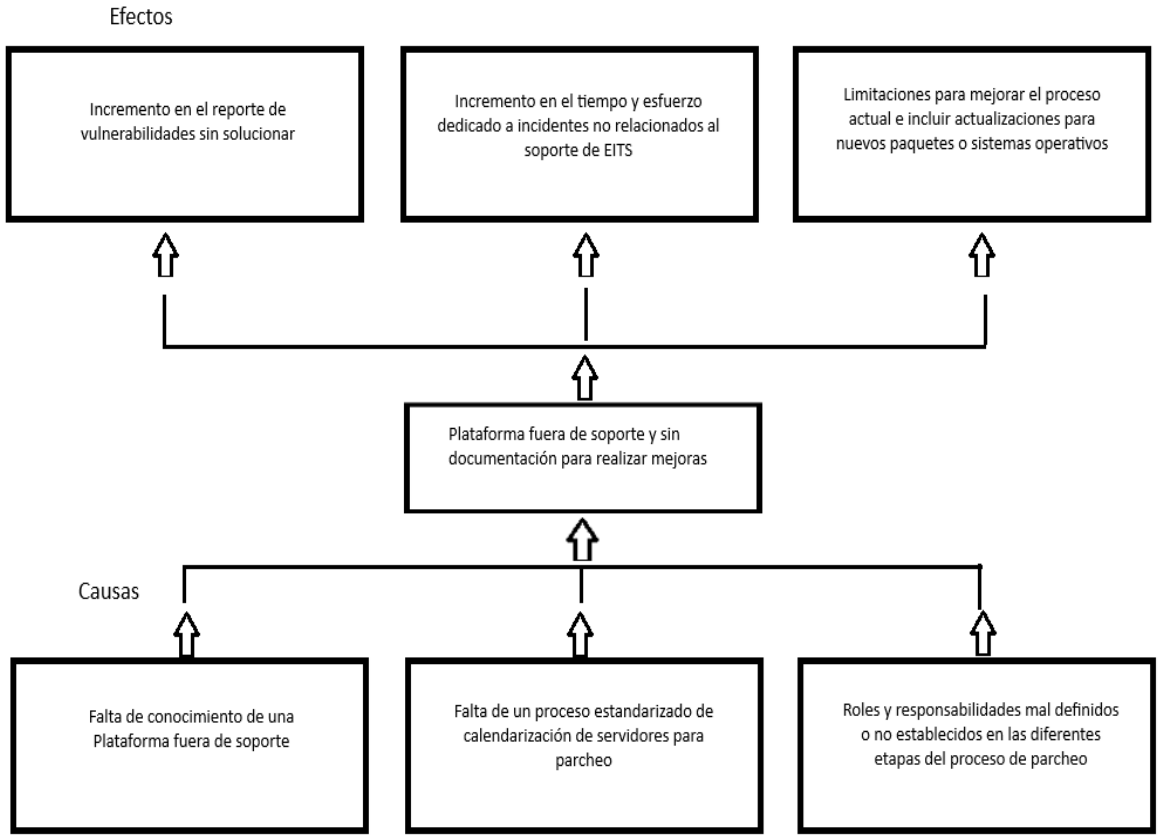


*Ilustración 1 Diagrama Causa-Efecto. Fuente: Elaboración propia*

### *1.2.2 Problema General.*

¿Como se puede reducir el número de vulnerabilidades causadas por el parcheo pendiente o fallido de servidores, evitando disminuir la carga de trabajo del grupo de EITS y sentando las bases para futuras mejoras y desarrollo?

### *1.2.3 Problemas Específicos.*

- ¿Como identificar las posibles mejoras de la plataforma actual del proceso de parcheo?
- ¿Como determinar la separación de deberes, roles y responsabilidades durante el proceso de parcheo?

- ¿Como se administrará y controlará el acceso a la calendarización del parcheo de servidores?
- ¿Como se ejecutará y administrará la solución planteada desde la plataforma de Ansible?

## 1.3 Objetivos del Proyecto.

### *1.3.1 Objetivo General.*

Presentar una propuesta de mejora al parcheo automático de servidores Linux Red Hat, a través del análisis de la plataforma de parcheo actual, así como el estudio y análisis de múltiples plataformas que puedan reemplazarla, para la resolución de las brechas y carencias identificadas en el proceso actual

### *1.3.2 Objetivos Específicos.*

- Analizar los diferentes procesos del ciclo parcheo, mediante la observación y estudio de la plataforma de parcheo actual para determinar los requerimientos mínimos y áreas de mejora a solventar.
- Identificar los roles y responsabilidades en cada una de las etapas del ciclo de parcheo, mediante entrevistas con los técnicos e ingenieros del departamento de EITS, para definir la correcta separación de deberes.
- Definir el control de acceso a la calendarización del parcheo de servidores, mediante el análisis de las capacidades de la plataforma ServiceNow, para implementar una capa de seguridad al proceso de parcheo.
- Elaborar un plan de integración de los objetivos anteriores, a través de una solución integral para el parcheo automático de servidores Red Hat.

## 1.4 Alcances y Limitaciones.

### *1.4.1 Alcances.*

- Realizar un diagnóstico y un entendimiento de los procesos del ciclo de parcheo actual para trasladarlos a una nueva plataforma que permita la mejora de los mismos.
- Hacer un diseño optimo de como deberían realizarse los diferentes procesos que comprenden el ciclo de parcheo de los servidores Linux Red Hat.
- Realizar un estudio de mercado de las herramientas que pueden ser posibles soluciones como mejora a la plataforma actual.
- Proponer la herramienta idónea que cumpla con los requerimientos del proceso de ciclo de parcheo e implemente las mejoras identificadas al proceso actual.

### *1.4.2 Limitaciones.*

- El proyecto será sometido a estudio una vez completado y presentado para determinar la viabilidad como solución deseada.
- El proyecto se centrará en servidores Linux Red Hat en la región de Norte América ya que son el grupo principal de soporte del departamento de EITS y serían los elegidos para utilizar esta solución si se decide implementar y antes de expandirla a otros sistemas operativos.

## 1.5 Cronograma de Actividades.

A continuación, se detalla el cronograma de actividades del proyecto, incluyendo las fases, etapas y capítulos, así como las actividades a realizar en cada uno.

| Fase #1: Proponer una mejora al proceso automático de parcheo de servidores Red Hat | | | | | |
|---|---|---|---|---|---|
| | Responsable | Estado | Inicio | Fin | Progreso |
| **Fase #2: Capítulo 1 Problema del proyecto.** | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| Antecedentes y justificación del proyecto. | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| Definición del problema. | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| Objetivos del proyecto. | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| Alcances y limitaciones. | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| Cronograma de actividades. | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| Definición de entregables. | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| Entrega de la documentación. | Manuel Salazar G. | Completo | 20/11/2023 | 10/12/2023 | 100% |
| | | | | | |
| **Fase #3: Capítulo 2 Marco teórico.** | Manuel Salazar G. | Completo | 18/02/2024 | 18/03/2024 | 100% |
| Conceptos teóricos y técnicos. | Manuel Salazar G. | Completo | 18/02/2024 | 18/03/2024 | 100% |
| Entrega de la documentación. | Manuel Salazar G. | Completo | 18/03/2024 | 18/03/2024 | 100% |
| | | | | | |
| **Fase #4: Capítulo 3 Marco metodológico.** | Manuel Salazar G. | Completo | 23/03/2024 | 24/03/2024 | 100% |
| Tipo y enfoque de la investigación. | Manuel Salazar G. | Completo | 23/03/2024 | 24/03/2024 | 100% |
| Fuentes y sujetos de información. | Manuel Salazar G. | Completo | 23/03/2024 | 24/03/2024 | 100% |
| Técnicas y herramientas de recolección de datos. | Manuel Salazar G. | Completo | 23/03/2024 | 24/03/2024 | 100% |
| Variables de investigación. | Manuel Salazar G. | Completo | 23/03/2024 | 24/03/2024 | 100% |
| Diseño de la investigación | Manuel Salazar G. | Completo | 23/03/2024 | 24/03/2024 | 100% |
| Matriz de coherencia | Manuel Salazar G. | Completo | 23/03/2024 | 24/03/2024 | 100% |
| | | | | | |
| **Fase #5: Capítulo 4 Diagnostico de la situación actual.** | Manuel Salazar G. | Completo | 05/04/2024 | 20/04/2024 | 100% |
| Diagnóstico administrativo u operativo | Manuel Salazar G. | Completo | 05/04/2024 | 20/04/2024 | 100% |
| Diagnóstico técnico | Manuel Salazar G. | Completo | 05/04/2024 | 20/04/2024 | 100% |
| Diagnóstico de percepción | Manuel Salazar G. | Completo | 05/04/2024 | 20/04/2024 | 100% |
| Brechas o conclusiones del diagnóstico | Manuel Salazar G. | Completo | 05/04/2024 | 20/04/2024 | 100% |
| | | | | | |
| **Fase #6: Capítulo 5 Propuesta del proyecto.** | Manuel Salazar G. | Completo | 13/05/2024 | 23/06/2024 | 100% |
| Requerimientos | Manuel Salazar G. | Completo | 13/05/2024 | 23/06/2024 | 100% |
| Diseño de la propuesta | Manuel Salazar G. | Completo | 13/05/2024 | 23/06/2024 | 100% |
| Prototipo de la propuesta | Manuel Salazar G. | Completo | 13/05/2024 | 23/06/2024 | 100% |
| | | | | | |
| **Fase #7: Capítulo 6 Conclusiones y recomendaciones.** | Manuel Salazar G. | Completo | 23/06/2024 | 26/06/2024 | 0% |
| | | | | | |

*Ilustración 2 Cronograma de actividades. Fuente: Elaboración propia basado en Ramírez, K. (2021)*

# Capítulo II: Marco Teórico.

La investigación y propuesta se desarrollan en la informática empresarial, donde las nuevas tecnologías y métodos de automatización son pilares importantes del desarrollo de nuevos procesos y de la estandarización de los ya predefinidos y que son parte de las mejores prácticas.

La investigación seleccionada se basa en la búsqueda de un impacto positivo en el grupo operaciones de un departamento que tiene a su cargo la seguridad del activo más importante de la empresa Experian, como lo es la información.

Con base en lo anterior, este proyecto tiene el objetivo de automatizar el proceso de parcheo de seguridad de los servidores Linux y los procesos adyacentes de calendarización del parcheo, reporte y asignación de incidentes, y habilitar la escalabilidad tanto de la infraestructura como de nuevas características para el proceso.

En este capítulo se muestran los conceptos claves dentro del desarrollo del proyecto.

## 2.1 Conceptos Generales

En esta sección se presentan los conceptos generales que proporcionan el conocimiento básico y fundamental para la comprensión y entendimiento de los elementos más avanzados de la investigación.

### *2.1.1 Actualización de Seguridad o Parche de Seguridad*

Se comprende por actualizaciones de seguridad, los paquetes o programas que los desarrolladores de software crean y distribuyen con el objetivo de mejorar o reparar ciertas características de sus programas o sistemas operativos y que puedan representar una debilidad o posible punto de explotación por parte de criminales informáticos o hackers.

Mejorando claridad y fluidez considerar: Según FasterCapital (2023), desde la perspectiva de los usuarios, las actualizaciones regulares proporcionan una capa adicional de

protección, garantizando la seguridad de la información personal y profesional. Desde el punto de vista de los desarrolladores, estas actualizaciones no solo refuerzan la seguridad del software, sino que también mejoran su funcionalidad, la experiencia del usuario y su compatibilidad con otros sistemas. Es esencial que los desarrolladores se mantengan actualizados con los últimos parches para prevenir posibles vulnerabilidades que podrían resultar en consecuencias legales y daños a la reputación, tanto a nivel individual como empresarial.


*Ilustración 3 Importancia de las actualizaciones. Fuente: FasterCapital (2023)*

### *2.1.2 Vulnerabilidades y Exploits*

Según Avast (2024), las vulnerabilidades representan las debilidades en la seguridad que surgen en los sistemas informáticos debido a errores, fallos o la evolución constante de la tecnología, lo que puede propiciar su aparición. Estas vulnerabilidades, como se describe, son fallas en el software que generan puntos débiles en la seguridad, permitiendo a los

ciberdelincuentes acceder a los sistemas informáticos. Por otro lado, un exploit se define como un programa diseñado específicamente para aprovechar estas vulnerabilidades. Es responsabilidad de los desarrolladores corregir estas vulnerabilidades para prevenir la explotación de estas.

Según López (2023), para protegerse contra las vulnerabilidades informáticas, es esencial establecer políticas de seguridad sólidas, mantener actualizado tanto el software como el hardware, y estar al tanto de las últimas amenazas en el ámbito informático. Las suscripciones a servicios de vulnerabilidades ofrecen información actualizada sobre posibles riesgos, permitiendo a las organizaciones tomar medidas preventivas para mitigar los riesgos antes de que se conviertan en problemas graves. Esto resalta la importancia de aplicar de manera continua parches a los sistemas informáticos mediante suscripciones de vulnerabilidades y un ciclo de actualización que implemente las correcciones identificadas por dichas suscripciones.


*Ilustración 4 Amenaza vs Vulnerabilidad. Fuente: INCIBE (2017)*

### *2.1.3 Ciclo de Parcheo*

El proceso de parcheo implica la identificación e implementación de actualizaciones en el sistema operativo, aplicaciones y paquetes para mantener los dispositivos seguros y actualizados. Según Ballejos (2023), este ciclo consta de diversas etapas, que abarcan desde la identificación de activos y software base, la creación de políticas de parcheo, la supervisión y prueba de los sistemas de parcheo, hasta el despliegue, verificación y documentación de los parches.

Por otro lado, INCIBE (2018) destaca algunas buenas prácticas en la gestión de parches, como establecer controles compensatorios para situaciones donde los parches no estén disponibles de inmediato, clasificar los parches según su importancia y realizar una exhaustiva verificación antes de implementarlos en entornos de producción. En contraste, Korzeniowski (2016) enfatiza la importancia de establecer una jerarquía en la aplicación de parches en servidores y decidir qué partes del proceso de parcheo automatizar, lo que contribuye a reducir la carga de trabajo de mantenimiento. Estas prácticas resaltan la aplicación, mejora y automatización del proceso de parcheo como fundamentales en la administración de la infraestructura digital empresarial.

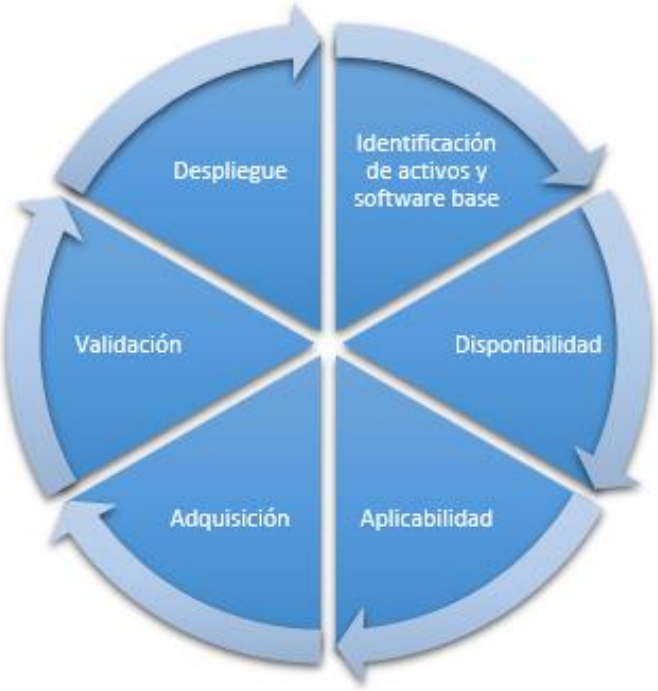


*Ilustración 5 Fases de la gestión de parches. Fuente: (INCIBE, 2018)*

### *2.1.4 Automatización de procesos informáticos*

La automatización de procesos informáticos hace uso de recursos tecnológicos para simplificar la ejecución o realización de tareas. Red Hat indica que la automatización de la TI, también denominada automatización de la infraestructura, consiste en utilizar sistemas de software para crear instrucciones y procesos repetibles que reemplacen o reduzcan la interacción humana con los sistemas de TI. El software de automatización funciona dentro de los límites de esas instrucciones, herramientas y marcos, para realizar las tareas con muy poca intervención humana, o sin ella. (Red Hat, 2023)

De acuerdo también con Red Hat (2023) la implementación de la automatización de procesos de TI presenta las siguientes ventajas:

1. La automatización de la TI no es una solución universal, pero al adoptar un enfoque integral, puede liberar al personal de tareas manuales y repetitivas.
2. La implementación de la automatización en la empresa puede aumentar la productividad de los equipos al reducir errores y optimizar procesos.
3. La automatización fomenta la mejora en la colaboración entre los miembros del equipo al simplificar tareas rutinarias y permitir un enfoque en actividades más estratégicas.
4. Al eliminar las tareas tediosas y repetitivas, los empleados pueden dedicar más tiempo a actividades de mayor importancia y complejidad, lo que beneficia tanto a la empresa como a su personal.

### *2.1.5 Plataforma TrueSight BladeLogic de BMC*

TrueSight BladeLogic, ahora conocido como TrueSight Automation, es una suite de herramientas desarrollada por BMC Software (2024) que la describe como una plataforma diseñada para gestionar y optimizar la infraestructura de TI. Esta plataforma permite a las organizaciones automatizar procesos críticos relacionados con servidores físicos, virtuales y en la nube.

Entre las principales características de esta plataforma, BMC Software (2024) y Cybermark (2024) mencionan las siguientes:

- Automatización de Servidores: TrueSight Automation permite la provisión, configuración, parcheo y mantenimiento de servidores, mejorando la eficiencia operativa y reduciendo errores humanos

- Gestión de Vulnerabilidades: La integración con TrueSight Vulnerability Management permite a los usuarios identificar y remediar vulnerabilidades de seguridad, priorizando acciones basadas en el impacto empresarial
- Cumplimiento Normativo: Ofrece políticas preconfiguradas que ayudan a las organizaciones a cumplir con normativas como CIS, HIPAA y PCI, facilitando auditorías continuas y asegurando que los sistemas estén siempre en conformidad
- Interfaz Unificada: La plataforma proporciona una interfaz única para ejecutar comandos y scripts en múltiples plataformas, lo que simplifica la gestión de tareas administrativas
- Informes y Auditoría: TrueSight incluye capacidades avanzadas de informes que permiten a los administradores realizar auditorías efectivas sobre el estado de la infraestructura y las configuraciones aplicadas

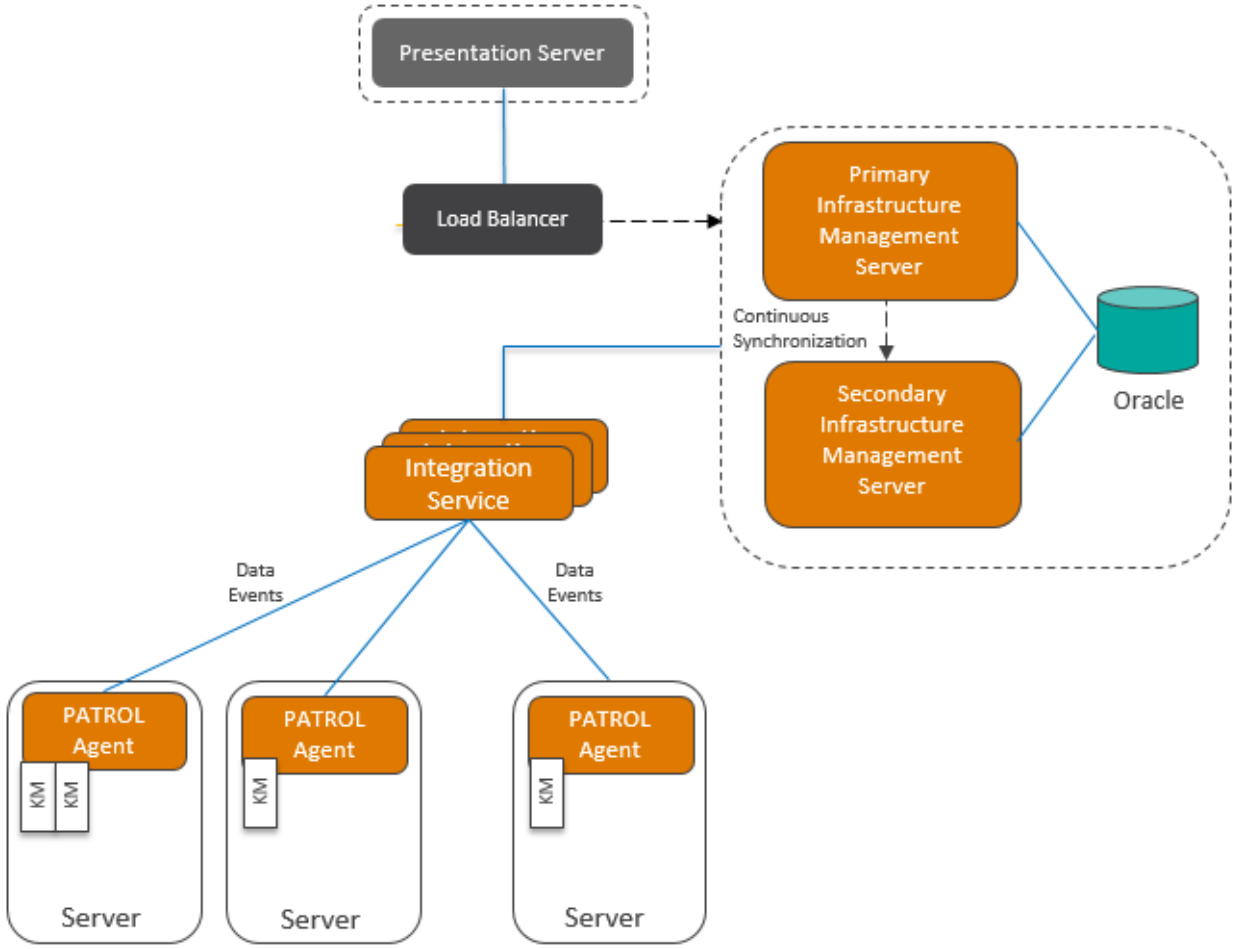


*Ilustración 6 Ejemplo de infraestructura de la plataforma TrueSight BladeLogic. Fuente: BMC Software*

### *2.1.6 Plataforma Ansible de Red Hat*

Ansible Automation Platform consiste en una herramienta desarrollada por Red Hat para la automatización y administración de procesos, tareas y configuraciones de sistemas informáticos. Los desarrolladores definen la plataforma como, un motor open source que automatiza procesos informáticos como preparación de infraestructura, gestión de configuración, implementación de aplicaciones y organización de sistemas. Se utiliza para instalar software, automatizar tareas, mejorar seguridad, aplicar parches, organizar flujos de trabajo. Red Hat Ansible Automation Platform ofrece soporte para el ciclo de vida empresarial y

funciones específicas para estandarizar, implementar y ampliar la automatización. (Red Hat, 2022)

Majnaly (2023) en lista las siguientes características principales de Ansible

1. Utiliza archivos de configuración llamados Playbooks escritos en YAML, compuestos por Plays y Tareas ejecutadas por Módulos Ansible
2. No requiere agentes para funcionar en los sistemas gestionados, lo que lo diferencia de otras herramientas como Puppet
3. Permite la automatización de tareas repetitivas, el parcheo de servidores Linux utilizando módulos como YUM, y se integra con diversas herramientas como Docker, Kubernetes, Microsoft Azure, entre otras.

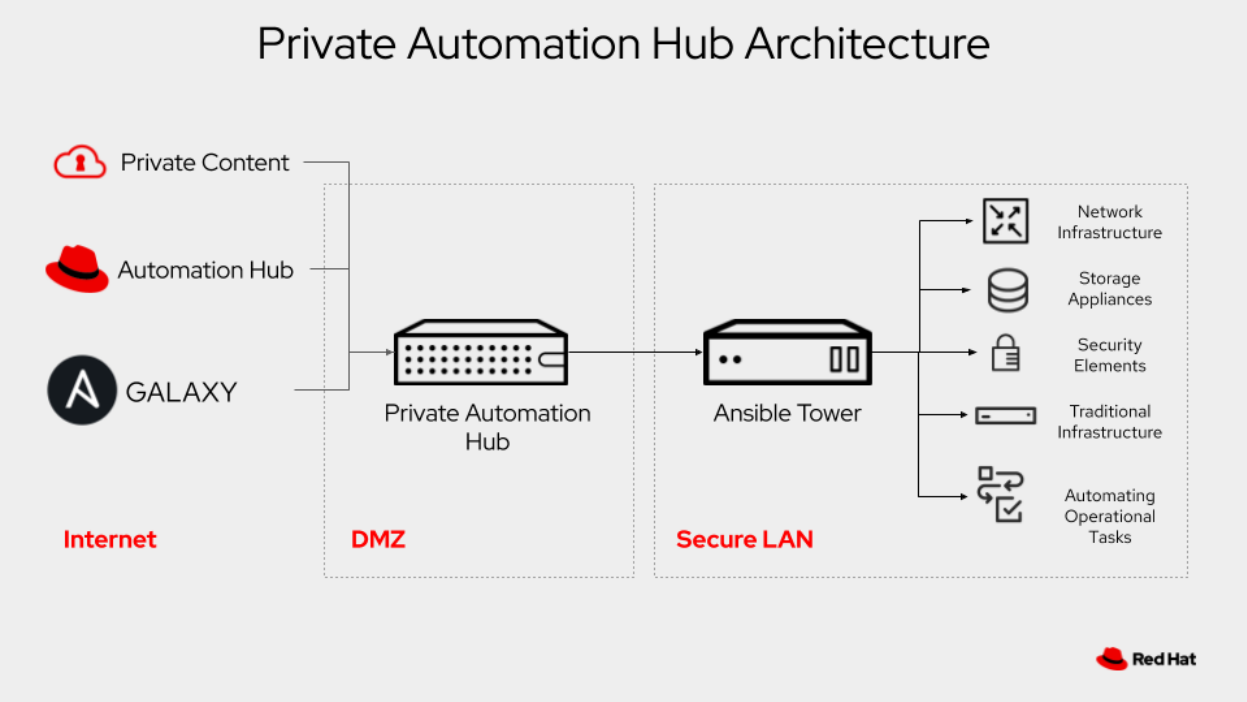


*Ilustración 7 Arquitectura de un Hub Privado de Automatización. Fuente Cavanaugh (2020)*

### *2.1.7 Plataforma Tanium*

Tanium es una plataforma de gestión de endpoints desarrollada por la empresa de Tanium Inc. Tanium es una plataforma de gestión de endpoints que permite a las organizaciones obtener visibilidad y control sobre sus activos de TI en tiempo real. Diseñada para abordar los desafíos de seguridad y gestión en entornos complejos, Tanium se destaca por su capacidad para proporcionar información instantánea sobre el estado de los dispositivos conectados a la red (Tanium Inc., 2024).

Basado en Tanium (2024) y Microsoft (2024) la plataforma presenta características tales como:

- Visibilidad en Tiempo Real: Tanium utiliza una arquitectura de comunicación patentada que permite a las organizaciones obtener información en tiempo real sobre todos sus endpoints, sin importar su ubicación o la cantidad de dispositivos. Esto proporciona una visión global y actualizada de toda la infraestructura.

- Gestión Unificada de Endpoints: La plataforma permite administrar tareas comunes de TI como parches, inventarios de software y hardware, configuración, cumplimiento normativo y auditorías de seguridad de manera centralizada, lo que mejora la eficiencia y reduce los riesgos.

- Detección y Respuesta a Amenazas (EDR): Tanium ofrece capacidades de detección de amenazas, investigación y respuesta en tiempo real. Esto ayuda a identificar rápidamente posibles brechas de seguridad y mitigar los incidentes antes de que puedan causar un daño significativo.

- Gestión de Parcheo: Permite implementar parches de seguridad de manera proactiva y garantizar que los dispositivos estén al día, lo cual es fundamental para proteger la infraestructura contra vulnerabilidades.
- Automatización y Remediación: La plataforma puede automatizar tareas repetitivas o críticas para la seguridad y la gestión de TI, mejorando la eficiencia operativa y reduciendo el tiempo de respuesta ante problemas.
- Cumplimiento Normativo y Seguridad: Tanium ayuda a las organizaciones a garantizar el cumplimiento de normativas y políticas internas, proporcionando herramientas para verificar configuraciones, identificar riesgos y resolver vulnerabilidades de manera ágil.

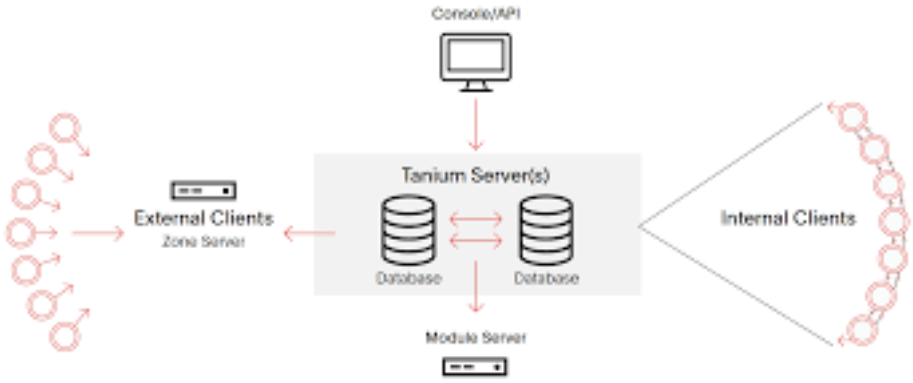


*Ilustración 8 Ejemple infraestructura Tanium. Fuente: Tanium.*

### *2.1.8 Control de Revisión*

El control de revisiones es una práctica aplicada a la elaboración de documentos y al desarrollo de software, y es una de las mejores prácticas definidas para rastrear y registrar cambios realizados, habilitando el trabajo colaborativo entre equipos o compañeros.

De acuerdo con Atlassian (2024) el control de versiones permite gestionar los cambios en el código a lo largo del tiempo. Registran todas las modificaciones en un tipo especial de

base de datos, permitiendo a los desarrolladores retroceder para resolver errores y minimizar interrupciones. También protege el código de desastres y errores humanos, garantizando su integridad. Facilita la colaboración entre desarrolladores al realizar un seguimiento de los cambios individuales, evitando conflictos en el trabajo concurrente y problemas como cambios incompatibles o la falta de visibilidad sobre las versiones disponibles para los usuarios.

Atlassian (2024) también menciona las siguientes ventajas de la utilización de un sistema de control de versiones:

1. Historial completo de cambios a largo plazo en todos los archivos, incluyendo creación, eliminación y modificaciones de contenido. Importante para analizar errores, solucionar problemas en versiones anteriores y mantener un registro detallado con autor, fecha y notas explicativas.
2. Creación de ramas y fusiones para trabajar en flujos independientes de cambios, permitiendo la verificación de conflictos y la integración posterior.
3. Trazabilidad de cada cambio en el software, conexión con herramientas de gestión de proyectos como Jira y anotaciones descriptivas para comprender el propósito y objetivo de cada modificación. Facilita el análisis de la causa raíz, la comprensión del código y la alineación con el diseño a largo plazo del sistema.

### *2.1.9 BitBucket*

BitBucket es un software diseñado para el desarrollo de código, que permite realizar el control de revisiones. Fue desarrollado por Atlassian (2024) que lo define como una plataforma de alojamiento de repositorios de código basada en la nube que facilita la colaboración en equipos de desarrollo, ofreciendo características como integración con Jira, revisiones de código eficientes y la capacidad de crear ramas desde Jira Software.

La diferencia entre BitBucket y otra herramienta bastante conocida para el desarrollo de software como lo es GitHub es que BitBucket si proporciona soporte para la creación y desarrollo de repositorios privados, mientras que todo el contenido creado en GitHub es accesible a toda la comunidad.

Otras ventajas que ofrece BitBucket son enlistadas por HostGator (2023)

1. Rendimiento escalable. La plataforma es responsiva, lo que permite que su capacidad de rendimiento no se vea afectada incluso con una alta demanda sobre el servidor.
2. Alta disponibilidad. Si algún elemento del clúster principal tiene problemas, el resto podrá atender todas las solicitudes sin impacto sobre los clientes.
3. Espejo inteligente. Permite configurar copias alojadas en repositorios remotos, lo que reduce el tiempo que lleva fusionar el trabajo de los colaboradores.
4. Autocorrección de archivos. Detecta y recupera los errores sin interrumpir las operaciones.
5. Seguridad. El código creado está seguro en la nube o servidor local, y se pueden acceder a través de controles de acceso como permisos de IP o verificación de 2 pasos.

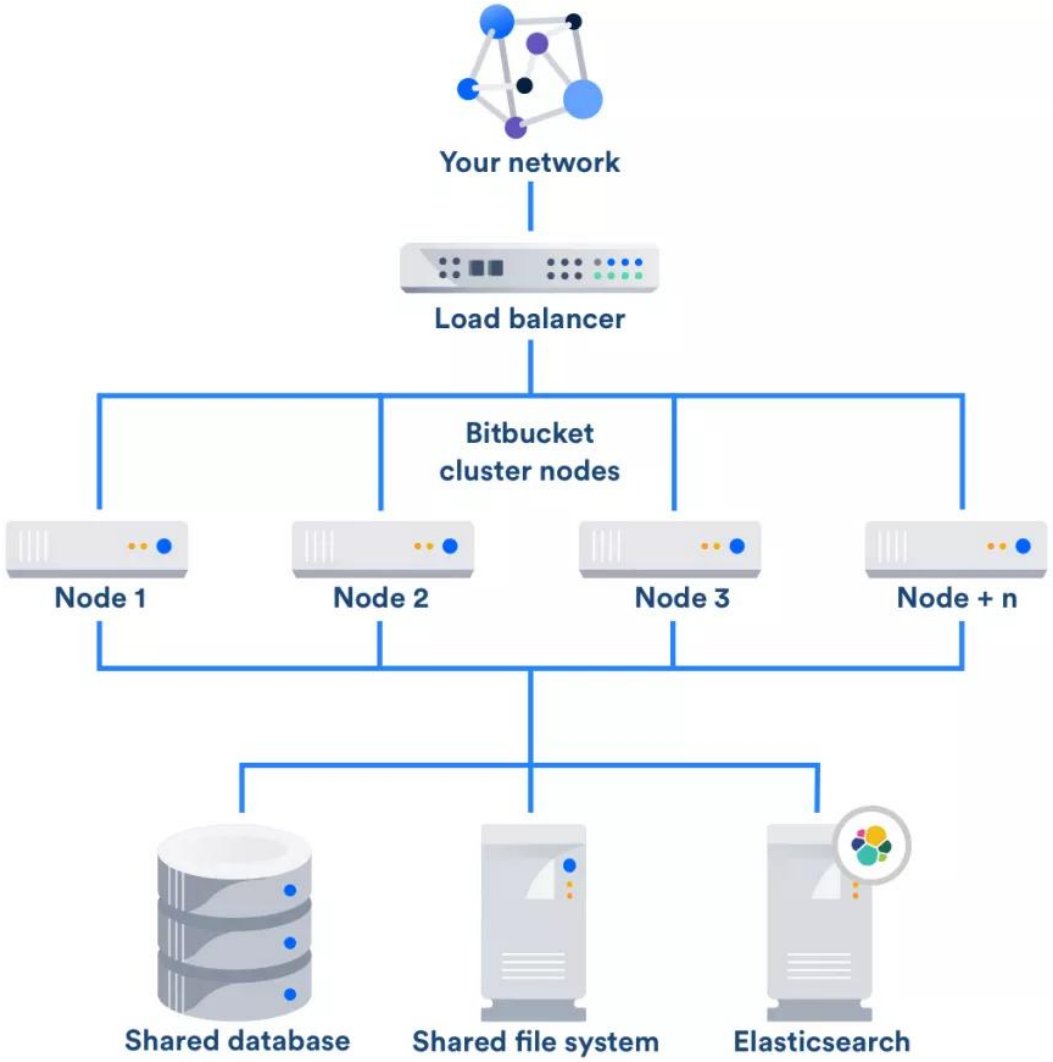


*Ilustración 9 Infraestructura BitBucket. Fuente Atlassian (2024)*

### *2.1.10 ITSM*

Por sus siglas en ingles ITSM significa gestión de servicios de TI (IT Service Management) y ServiceNow (2024) lo define como un enfoque estratégico que abarca los procesos y tecnologías utilizados para planificar, implementar y brindar servicios de tecnología de la información, así como ofrecer soporte a los mismos.

ServiceNow (2024) menciona que la gestión de servicios de TI (ITSM) es crucial en la actualidad, ya que la tecnología de la información abarca todas las áreas de una organización. ITSM coordina eficazmente los servicios de TI para garantizar valor real al cliente, al tiempo

que mejora la eficacia empresarial y aumenta la productividad de los empleados; y en el área de TI, más específicamente, los beneficios que ofrece ITSM son:

1. Productividad mejorada: Objetivos alineados respaldados por servicios confiables para realizar más tareas con menos problemas.
2. Mayor satisfacción del usuario: Enfoque en las necesidades del usuario al prestar TI como servicio.
3. Mejor escalado de procesos: Procesos más eficaces permiten gestionar más desarrollo de TI sin reducir la calidad.
4. Detección de incidentes y respuesta más rápidas: Visibilidad mejorada de TI para identificar y responder a incidentes antes de que se conviertan en problemas.

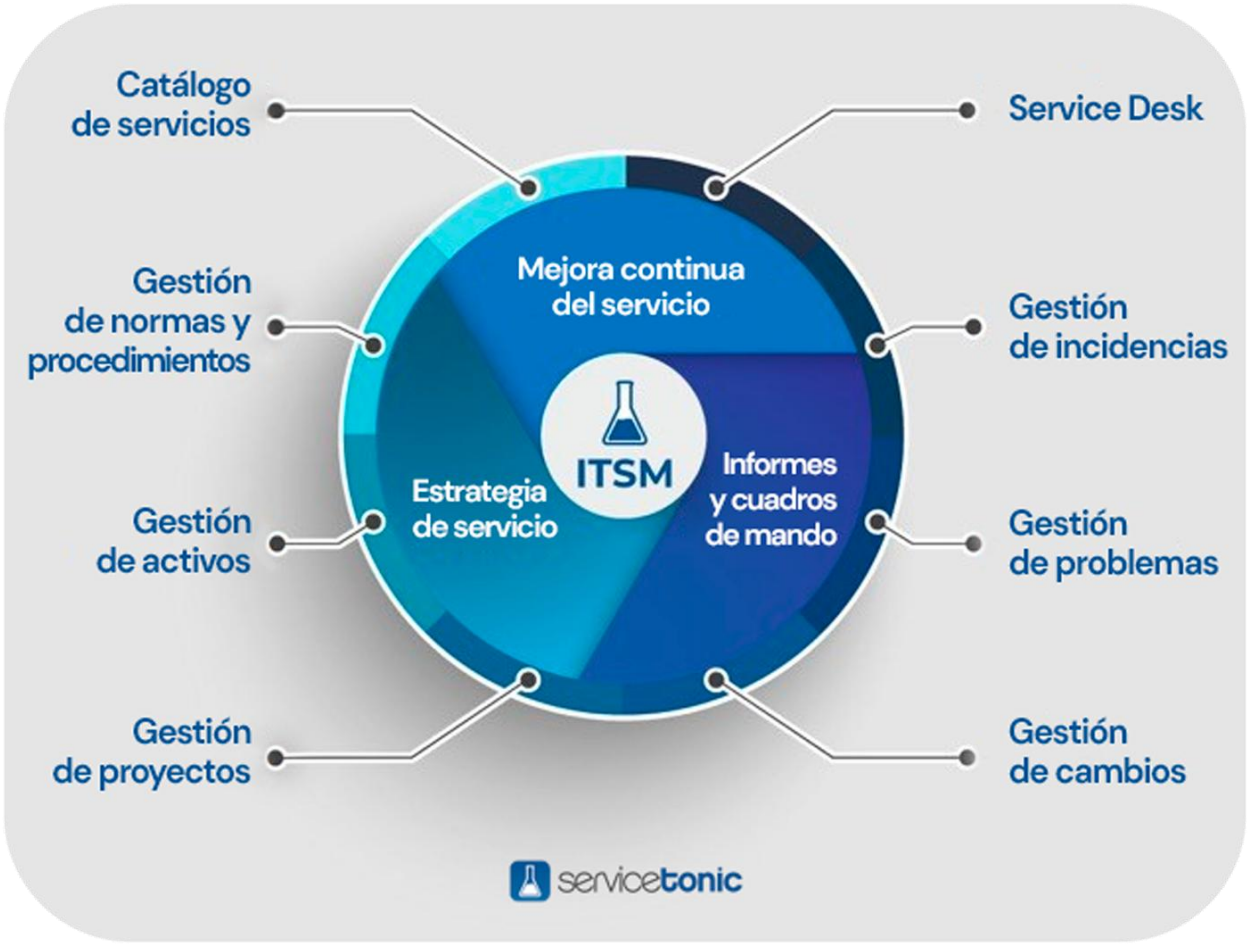


*Ilustración 10 Gestión de ITSM. Fuente ServiceTonic (2024)*

### *2.1.11 ServiceNow*

ServiceNow es la plataforma desarrollada por la empresa del mismo nombre, fundada en 2004 por Fred Luddy, ex-CTO de Peregrine Systems. Actualmente es el líder mundial en ITSM o gestión de servicios de IT. ServiceNow (2024) nos indica que la plataforma ofrece potentes funciones de gestión de servicios de TI para optimizar procesos, crear experiencias fluidas y fomentar la innovación. Además, Snow Software proporciona soluciones que se integran con ServiceNow para mejorar la toma de decisiones de TI, optimizar el gasto, rastrear el uso y mitigar riesgos. La integración de Snow con ServiceNow ayuda a automatizar

procesos, mejorar la precisión de los datos en la CMDB (Base de Datos de Gestión de Configuración) y aumentar la eficiencia operativa general.

De acuerdo con ServiceNow (2024) los beneficios de utilizar esta plataforma son:

1. Ofrece experiencias simplificadas que unifican datos de toda la organización en una interfaz intuitiva.
2. La automatización con intención de ServiceNow libera a los empleados y mejora los resultados empresariales.
3. Proporciona agilidad organizacional con una plataforma inteligente que permite adaptarse rápidamente y fomenta la innovación.

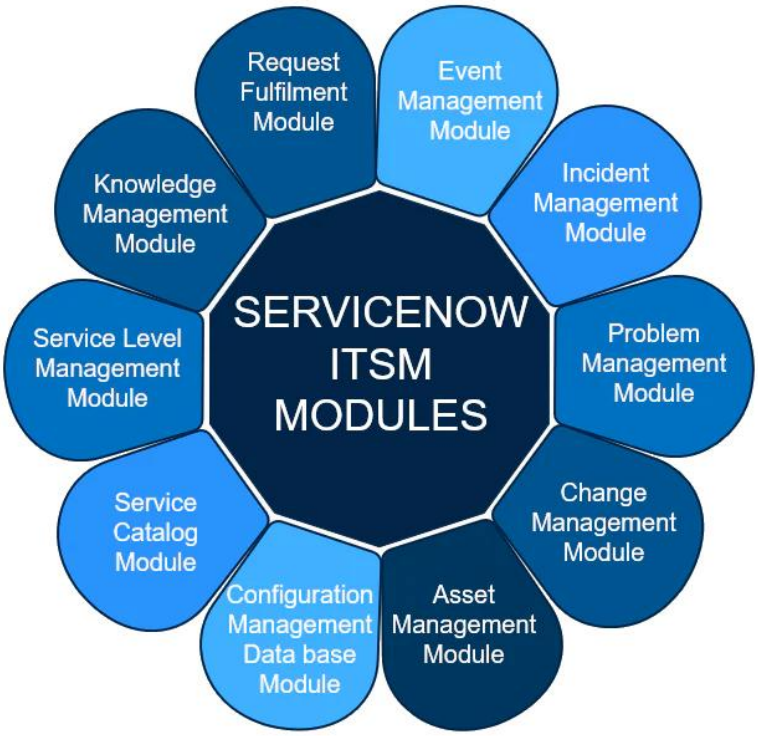


*Ilustración 11 Módulos de ITSM en ServiceNow. Fuente ITSM Docs*

## 2.2 Análisis

En esta sección se presentan conceptos relacionados a la etapa de análisis. Estos son aquellos conceptos relacionados análisis de información, eventos o manipulación de información que se plantearan en la investigación.

### *2.2.1 Análisis de información*

La adecuada definición de requerimientos conlleva a la recopilación de información precisa y necesaria para ser analizada. Ortega (2024) menciona que el análisis de la información es un proceso fundamental que implica decodificar datos contenidos en documentos específicos para obtener información valiosa y confiable que respalde la toma de decisiones en organizaciones.

Por su parte, El Centro Europeo de Postgrado CEUPE (2021) indica que el análisis se basa en métodos estructurados que incluyen la recolección, transformación, limpieza y modelado de datos para obtener información detallada y útil. Además, el análisis de la información puede involucrar tanto datos cuantitativos, que se refieren a números y estadísticas, como datos cualitativos, que se centran en comprender significados y contextos.

### *2.2.2 Modelado Basado en Eventos*

El modelo basado en eventos consiste en un enfoque arquitectónico centrado en la generación, detección, consumo y reacción a eventos significativos en un sistema. Este modelo permite la flexibilidad y agilidad al permitir que los componentes del sistema operen de forma independiente y respondan a eventos sin una dependencia directa entre sí. En este contexto, los eventos se definen como sucesos o cambios relevantes en un sistema o su entorno externo, los cuales pueden desencadenar acciones, procesos o comunicación entre los diferentes componentes de un sistema de software.

Red Hat (2019) describe la arquitectura basada en eventos como un modelo y una arquitectura de software diseñada para registrar, comunicar y procesar eventos entre servicios desacoplados, permitiendo mantener la asincronía de los sistemas y compartir información de manera eficiente.

AWS (2023) destaca que la arquitectura basada en eventos promueve un acoplamiento flexible entre los componentes de un sistema, lo que conduce a una mayor agilidad y escalabilidad. Además, menciona que, con esta arquitectura, los desarrolladores ya no necesitan escribir código personalizado para sondear, filtrar y enrutar eventos, lo que aumenta la agilidad del desarrollo.

La importancia del modelado basado en eventos es lo esencial que se ha vuelto para la evaluación económica y la toma de decisiones en diversos campos, ya que permite simular eventos significativos, analizar escenarios complejos y evaluar alternativas de manera detallada. De acuerdo con AWS (2023) los beneficios del modelado basado en objetos son:

- Ofrece ventajas significativas en términos de escalabilidad, agilidad, extensibilidad, reducción de complejidad, auditoría y costos. Al desacoplar servicios, se logra que los componentes puedan escalar y fallar de forma independiente, aumentando la resiliencia de una aplicación.
- Facilita el diseño de sistemas casi en tiempo real, alejándose del procesamiento basado en lotes. Los eventos se generan al cambiar el estado de una aplicación, lo que permite escalar verticalmente tanto los eventos como la capa que los procesa.
- Es altamente extensible, permitiendo a otros equipos ampliar características sin afectar los microservicios existentes. Al publicar eventos, las nuevas funciones pueden integrarse fácilmente como consumidores sin modificar la solución actual.

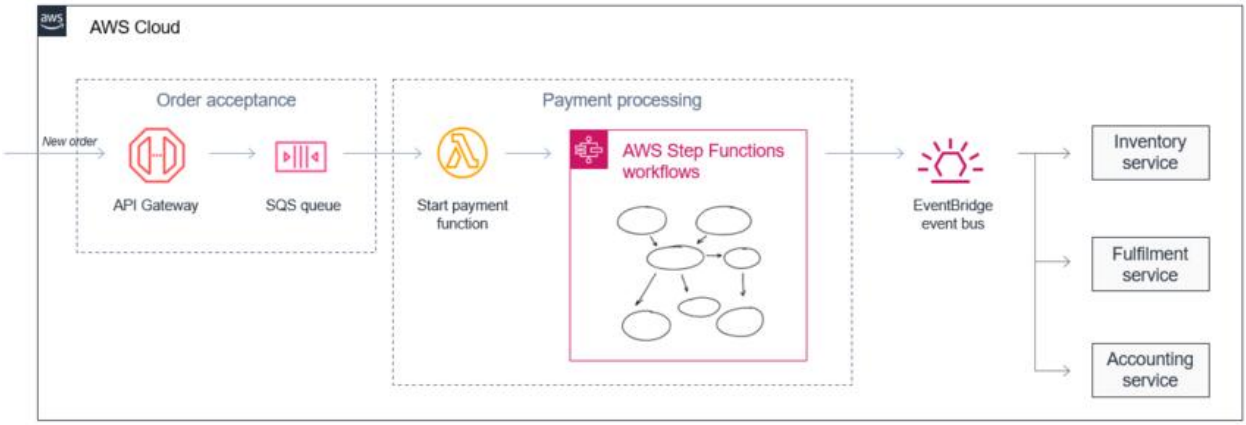


*Ilustración 12 Ejemplo de un modelo basado en eventos para un sistema de aceptación de pedidos. Fuente: AWS (2023)*

### *2.2.3 Diseño de Base de Datos*

El proceso de diseño de una base de datos es esencial e implica una serie de pasos cruciales para asegurar su eficacia y funcionalidad.

El diseño de una base de datos consiste en una secuencia de pasos que facilitan la creación, implementación y mantenimiento de los sistemas de gestión de datos empresariales. Según Naeem (2023), el objetivo principal del diseño de una base de datos es generar modelos físicos y lógicos que definan la estructura del sistema de base de datos propuesto.

Naeem (2023) también destaca que un buen diseño de base de datos garantiza coherencia, elimina la redundancia de datos y mejora el rendimiento. Esta metodología de diseño ahorra tiempo durante el desarrollo, evita la duplicación de datos mediante tablas de valores únicos y claves, optimiza el rendimiento con consultas sencillas y facilita el mantenimiento. Por el contrario, un diseño deficiente puede ocasionar problemas con mínimas interrupciones.

Microsoft (2024) indica en su blog que el diseño de una base de datos se rige por principios clave.

- Evitar la duplicación de información es fundamental para prevenir errores y optimizar el espacio.
- La corrección e integridad de los datos son esenciales para informes precisos y decisiones acertadas.
- Un buen diseño de base de datos divide la información en tablas temáticas, facilita la unión de datos, asegura la precisión e integridad, y se adapta a las necesidades de informes y procesamiento de datos.

Además de los principios, el diseño de base de datos cuenta con 3 etapas que se identifican como Diseño Conceptual, Diseño Lógico y Diseño Físico (Ilustración 13). De acuerdo con Reyes (2019) En la etapa del Diseño Conceptual, se obtiene una estructura de la información independiente de la tecnología a utilizar. En el Diseño Lógico, se consideran aspectos más físicos y se transforma la estructura obtenida en el diseño conceptual. Por último, en el Diseño Físico, se busca una mayor eficiencia y se completan aspectos de implementación física dependiendo del SGBD (Sistema de Gestión de Bases de Datos) utilizado.

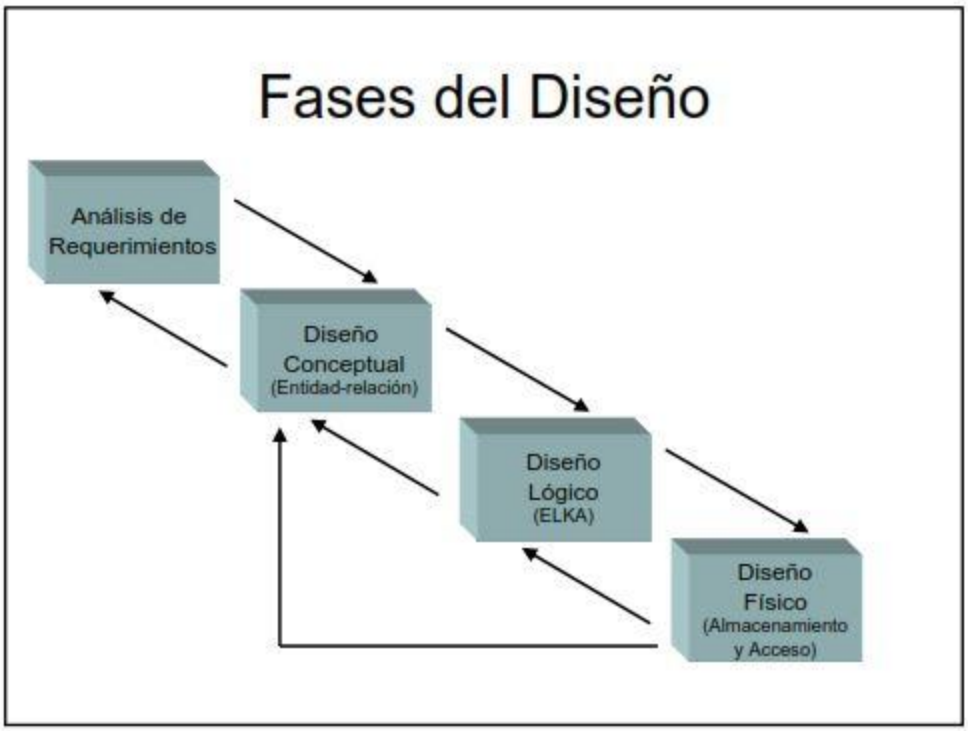


*Ilustración 13 Fases de diseño de una Base de Datos. Fuente Reyes (2019)*

### *2.2.4 Lenguaje de programación*

Un lenguaje de programación es un conjunto de reglas y símbolos que permiten a un programador escribir instrucciones que una computadora puede entender y ejecutar. Pérez y Merino (2023) lo definen como una estructura con reglas sintácticas y semánticas que permite impartir instrucciones a un programa de computadora. En términos teóricos, un lenguaje de programación es un sistema de comunicación con una estructura específica, contenido y uso, utilizado para escribir el código fuente de un software.

Según sus objetivos y herramientas, los lenguajes de programación se pueden clasificar en 3 tipos (Ilustración 14). Según RockContent (2018) estos son:

- Lenguaje máquina: Es el lenguaje más primitivo y se basa en la numeración binaria, compuesta únicamente por 0 y 1. Las máquinas o computadoras utilizan directamente este tipo de lenguaje.
- Lenguajes de programación de bajo nivel: Estos lenguajes son un poco más fáciles de interpretar que el lenguaje máquina, pero su interpretación puede variar según la máquina o computadora que se esté programando. Se caracterizan por estar más cerca del hardware y ser específicos para una arquitectura concreta.
- Lenguajes de programación de alto nivel: Son lenguajes más cercanos al lenguaje humano y ofrecen una mayor abstracción del hardware subyacente. Estos lenguajes son más fáciles de entender y escribir para los programadores, ya que se centran en la resolución de problemas sin tener que preocuparse por detalles específicos del hardware.

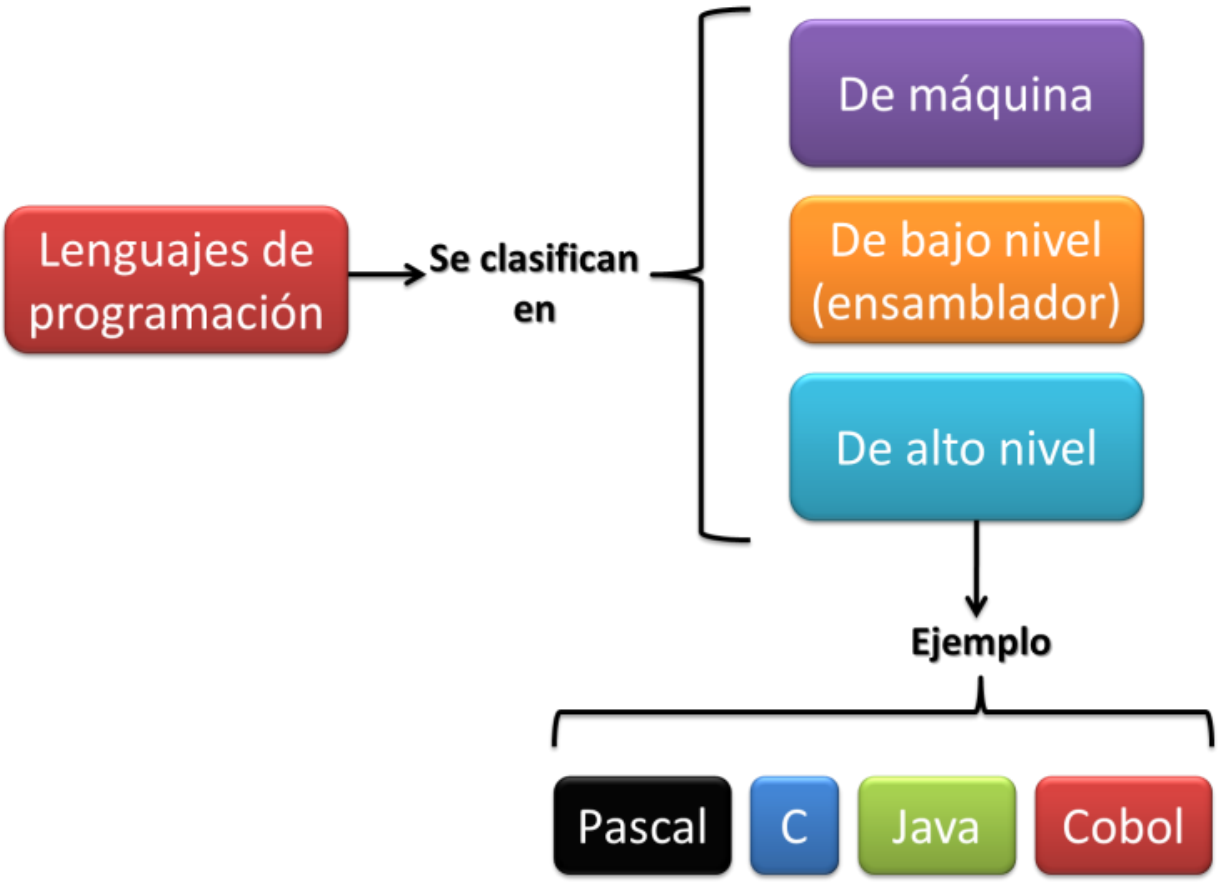


*Ilustración 14 Tipos de Lenguaje de Programación. Fuente RockContent (2018)*

### *2.2.5 Shell Scripts*

Para comprender que son los Shell scripts es necesario conocer lo básico sobre el Shell de Unix/Linux de antemano. Según TechTarget (2024) el shell es la interfaz de línea de comandos (CLI) del sistema operativo y el intérprete de un conjunto de comandos utilizados para comunicarse con el sistema. Lo que significa que puede recuperar, procesar y almacenar información en un ordenador o servidor.

También TechTarget (2024) indica que un script de shell de Unix es un archivo de texto que contiene una secuencia de comandos diseñados para ser ejecutados en un sistema operativo basado en Unix, como Linux. Estos scripts combinan una serie de comandos que, de otra manera, tendrían que ser escritos uno a uno en el teclado, en un solo script.

Las ventajas de utilizar shell scripts son diversas y se centran en la automatización de tareas, la eficiencia en la ejecución de comandos y la simplificación del trabajo en sistemas Unix/Linux. Gustavo B. (2023) hace referencia a las siguientes:

- Automatización de tareas: Los shell scripts permiten automatizar tareas rutinarias, como copias de seguridad de archivos, actualizaciones del sistema, limpieza de directorios y más.
- Eficiencia y ahorro de tiempo: Al escribir secuencias de comandos en un script, se agiliza el flujo de trabajo al eliminar la necesidad de ejecutar manualmente múltiples comandos uno por uno.
- Facilidad de uso: Los shell scripts simplifican la ejecución de comandos complejos al compilarlos en un único archivo ejecutable, lo que facilita su uso repetido sin tener que recordar cada comando.
- Estructura y organización: Los comandos en un script se pueden estructurar en una secuencia lógica para garantizar que se ejecuten en el orden correcto cada vez que se ejecute el script.
- Transparencia y legibilidad: Al estar escritos en un archivo de texto legible, los shell scripts permiten a otros usuarios revisar y comprender fácilmente su contenido, lo que facilita la colaboración y el mantenimiento.
- Portabilidad: Los scripts pueden transferirse a diferentes distribuciones de Linux y seguir funcionando siempre que los comandos del shell estén disponibles en el sistema operativo específico, lo que brinda flexibilidad y compatibilidad.

La Ilustración 15 presenta un ejemplo básico de un Shell script que imprime el mensaje ‘Hello World’ en el computador.

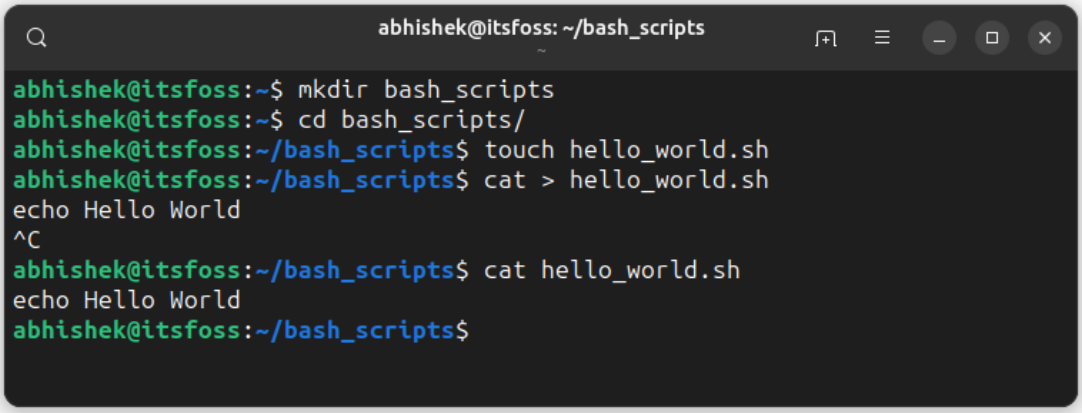


*Ilustración 15 Ejemplo básico de shell script. Fuente Prakash (2024)*

### *2.2.6 Serialización de datos*

La definición básica de serialización es el proceso de convertir el estado de un objeto en un formato que se pueda almacenar o transportar. Para detallar aún más, Ruelas (2017) explica que es un proceso que convierte objetos o estructuras de datos en memoria a flujos de bytes, permitiendo su almacenamiento en dispositivos, bases de datos o su transmisión a través de redes. Estos flujos de bytes son independientes de la arquitectura de los equipos y pueden ser reconstruidos posteriormente en su forma original, incluso en programas escritos en diferentes lenguajes mediante el proceso inverso de deserialización (Ilustración 16). El objetivo principal de este proceso es facilitar la persistencia y el intercambio de datos entre sistemas heterogéneos, creando representaciones portables de los objetos.

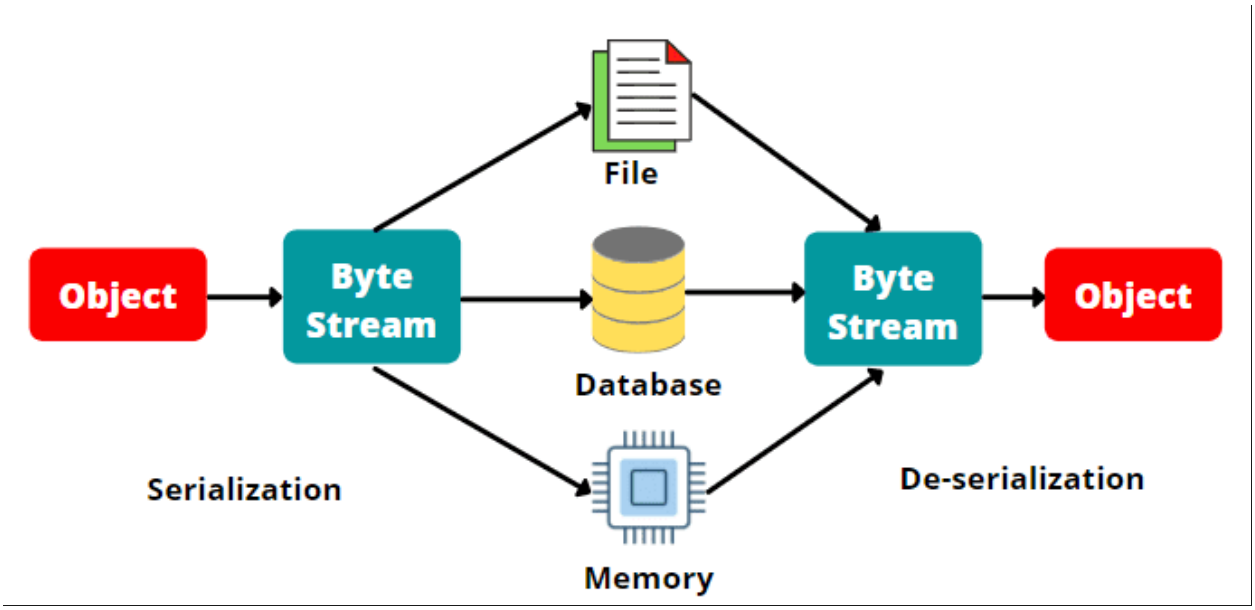


*Ilustración 16 Serialización vs Deserialización. Fuente: ScienceTech Easy (2021)*

## 2.3 Diseño de propuesta

En esta sección se presentan conceptos relacionados a la etapa de diseño de la propuesta.

### *2.3.1 YAML*

El lenguaje YAML, que significa "YAML Ain't Markup Language" (YAML no es un lenguaje de marcado), es un lenguaje de formato de serialización de datos que facilita la legibilidad y la capacidad de escritura del usuario y se encarga de almacenar archivos de configuración y se puede usar junto con todos los lenguajes de programación.

Basado en Red Hat (2023) YAML se utiliza ampliamente en el diseño de archivos de configuración y en la representación de datos de forma estructurada. La principal ventaja de YAML radica en su legibilidad y facilidad de escritura, lo que lo hace ideal para archivos de configuración que deben ser comprensibles para los humanos. Este lenguaje no permite tabulaciones literales y utiliza la sangría al estilo Python para definir la estructura de los datos.

Es compatible con varios tipos de datos como cadenas, números, booleanos, listas y diccionarios, lo que lo convierte en una herramienta versátil para la serialización de datos en diferentes entornos de programación. Gracias a estas características YAML es especialmente útil en entornos donde la claridad y la facilidad de lectura son prioritarias, como en el desarrollo de aplicaciones y sistemas de automatización.

La ilustración 17 muestra un ejemplo comparativo entre los lenguajes XML, JSON and YAML que demuestra la facilidad con la que se puede leer, escribir y comprender YAML comparado a los otros lenguajes.

| XML | JSON | YAML |
|---|---|---|
| <Servers><br><Server><br><name>Server1</name><br><owner>John</owner><br><created>123456</created><br><status>active</status><br></Server><br></Servers> | {<br>Servers: [<br>{<br>name: Server1,<br>owner: John,<br>created: 123456,<br>status: active<br>}<br>]<br>} | Servers:<br>- name: Server1<br>owner: John<br>created: 123456<br>status: active |

*Ilustración 17 Comparación XML, JSON y YML. Fuente: Deshpande (2019)*

### *2.3.2 Integración de Tecnologías*

La investigación explora la posibilidad y capacidad e integrar diferentes plataformas y tecnologías para resolver las necesidades de los usuarios y facilitar la automatización de procesos en sus diferentes etapas. De acuerdo con FasterCapital (2024) la integración de tecnologías se refiere a la fusión fluida y eficiente de diferentes tecnologías, sistemas o procesos para que funcionen juntos sin problemas. Este proceso es fundamental para impulsar asociaciones estratégicas exitosas, como se evidencia en colaboraciones entre empresas como Uber y Spotify, Nike y Apple, y Salesforce y Google Cloud. Al integrar tecnologías de manera perfecta, se logra mejorar la colaboración entre personas, equipos y organizaciones, lo

que a su vez desbloquea beneficios como una mejor comunicación, colaboración, eficiencia y productividad.

Por su parte Efiempresa (2024) menciona los siguientes beneficios de la integración tecnológica

- Seguridad de la información: Protege los datos contra intrusiones, aunque requiere medidas de seguridad informática para prevenir ataques.
- Suministro de información en tiempo real: Facilita la interacción instantánea en actividades como transacciones bancarias y comunicación en línea.
- Administración eficiente del tiempo: Reduce el tiempo necesario para tareas, mejorando la eficiencia y productividad.
- Optimización de procesos: Aumenta la velocidad de respuesta y la calidad de los procesos, buscando altos niveles de eficiencia.
- Adaptación a los cambios: Se ajusta a las evoluciones del entorno y las necesidades de usuarios y empresas, estandarizando procesos y mejorando la calidad de forma continua.


*Ilustración 18 Beneficios de integración de tecnologías. Fuente: FasterCapital (2024)*

Sin embargo, junto con estos beneficios, la integración de tecnologías también presenta desafíos y desventajas. Según Telefónica (2023) algunas de las posibles desventajas incluyen distracciones y falta de atención, especialmente en entornos educativos, donde el acceso ilimitado a recursos en línea puede desviar la atención de los estudiantes. Además, el uso excesivo e inadecuado de la tecnología puede llevar a una relación compulsiva con las mismas, lo que a su vez puede tener efectos adversos en la salud, la vida social y académica de los individuos.

# Capítulo III: Marco Metodológico.

Este capítulo se refiere al diseño metodológico o de investigación del proyecto. Al ser una modalidad de aplicación, las técnicas y métodos descritos en esta sección buscan alcanzar una meta materializable, no un resultado teórico o planteamiento de hipótesis.

## 3.1 Tipos y Enfoque de la Investigación

En esta sección se exponen el tipo de la investigación que permitirá clasificarla basado en su aplicación, objeto de estudio o medición. Además, se presenta el enfoque de la investigación basado en el resultado del proyecto, considerado en función de sus variables, su racionalidad y objetividad de los resultados.

### *3.1.1 Tipo de investigación.*

La presente investigación pretende el desarrollo de una propuesta que permita solucionar las necesidades del departamento de EITS en Experian CR y mejorar el proceso actualmente implementado para el parcheo de servidores Red Hat. Basado en lo anterior, el tipo de investigación que se ejercerá será Investigación Aplicada, y las personas que fungirán como fuentes primarias de información serán los ingenieros miembros del departamento de EITS, expertos en diferentes áreas y etapas del proceso de parcheo.

En el libro: Cómo elaborar y asesorar una investigación de tesis, el autor nos presenta el siguiente concepto de investigación de campo:

Son las investigaciones en las que la recopilación de información se realiza enmarcada por el ambiente especifico en el que se presenta el fenómeno de estudio. En la realización de estas tesis se utiliza un método exclusivo de investigación y se diseñan ciertas herramientas para recabar información que solo se aplican en el medio en el que actúa el fenómeno de estudio. (Muñoz , 1998, p. 18)

### *3.1.2 Enfoque de investigación.*

Una vez definido el tipo de investigación, se debe establecer el enfoque de esta, que puede ser cuantitativo, cualitativo y presentar una mezcla de ambos. Pero antes definiremos los conceptos de cada uno.

**Enfoque cuantitativo.** Representa un conjunto de procesos organizado de manera secuencial para comprobar ciertas suposiciones. Cada fase precede a la siguiente y no podemos eludir pasos, el orden es riguroso, aunque desde luego, podemos redefinir alguna etapa. Parte de una idea que se delimita y, una vez acotada, se generan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o perspectiva teórica. De las preguntas se derivan hipótesis y determinan y definen variables; se traza un plan para probar las primeras (diseño, que es como “el mapa de la ruta”); se seleccionan casos o unidades para medir en estas las variables en un contexto específico (lugar y tiempo); se analizan y vinculan las mediciones obtenidas (utilizando métodos estadísticos), y se extrae una serie de conclusiones respecto de la o las hipótesis. **(Sampieri & Mendoza, 2018, p. 5)**

**Enfoque cualitativo.** En estas investigaciones, se estudian fenómenos de manera sistemática. Sin embargo, en lugar de comenzar con una teoría y luego “voltear” al mundo empírico para confirmar si esta es apoyada por los datos y resultados, el investigador comienza el proceso examinando los hechos en sí y revisado los estudios previos, ambas acciones de manera simultánea, a fin de generar una teoría que sea consistente con lo que está observando que ocurre. De igual forma, se plantea un problema de investigación, pero normalmente no es tan específico como en la indagación cuantitativa. Va enfocándose paulatinamente. La ruta se va descubriendo o construyendo de acuerdo con el contexto y los eventos que ocurren conforme se desarrolla el estudio. Las investigaciones cualitativas suelen producir preguntas antes, durante o después de la recolección y análisis de los datos. La acción indagatoria se mueve de manera dinámica entre los hechos y su interpretación, y resulta

un proceso más bien “circular” en el que la secuencia no siempre es la misma, puede variar en cada estudio. **(Sampieri & Mendoza, 2018, p. 8)**

La investigación para presentar una propuesta de parcheo automático de paquetes del sistema operativo en Experian Costa Rica sería aplicada. Este tipo de investigación se caracteriza por buscar soluciones prácticas a problemas específicos y aplicar los conocimientos adquiridos en la práctica.

En cuanto al enfoque de la investigación, se podría considerar un enfoque mixto, ya que se requiere tanto la recopilación de datos cualitativos (como la identificación de problemas en la metodología actual y la definición de requerimientos) como cuantitativos (como la medición del impacto de las vulnerabilidades y atrasos en la resolución de incidentes). La combinación de ambos enfoques permitiría obtener una visión integral y fundamentada para desarrollar la propuesta de parcheo automático de manera efectiva y eficiente.

## 3.2 Fuentes y Sujetos de Información.

Las fuentes y sujetos de información son elementos fundamentales en la construcción de argumentos sólidos y bien fundamentados en cualquier trabajo académico. Las fuentes de información desempeñan un papel vital al proporcionar la base necesaria para respaldar las afirmaciones y conclusiones presentadas. Por otro lado, los sujetos de información representan los temas, conceptos o áreas de estudio que son minuciosamente explorados y analizados en la tesis, contribuyendo así a la solidez y relevancia del trabajo investigativo.

### *3.2.1 Fuentes de información.*

Representan la base sobre la cual se construye el conocimiento y se sustentan los argumentos presentados. La selección cuidadosa y crítica de fuentes confiables y relevantes es fundamental para garantizar la solidez y validez de la investigación, así como para enriquecer

el trabajo académico con aportes significativos. Las fuentes de información se clasifican de la siguiente manera.

**Fuentes Primarias*.*** En su libro Metodología de la Investigación Científica Eufemia Reyes, define una fuente de información primaria como:

También llamada fuente documental, es la que comprende el material de primera mano referente al objeto de estudio. Puede ser fuente primaria un trabajo creado por algún testigo o protagonista de un evento histórico en el que estos son descritos, pero también pueden incluirse objetos físicos, artículos periodísticos, cartas o diarios personales. Estas fuentes son los documentos que registran o corroboran el conocimiento inmediato de la investigación. Incluyen libros, revistas, informes técnicos y tesis. **(Reyes E. , 2022)**

La presente investigación cuenta que las siguientes fuentes primarias:

- Observación del proceso actual.
- Entrevistas con los ingenieros de Red Hat, del grupo de EITS y clientes.
- Análisis de datos recopilados.
- Documentación del departamento de EITS.

**Fuentes Secundarias*.*** Eufemia Reyes también describe qué son las fuentes secundarias en su libro Metodología Investigación Científica de esta manara:

Son textos basados en fuentes primarias, implican generalización, análisis, síntesis, interpretación o evaluación. Este renglón incluye las enciclopedias los anuarios, manuales, almanaques, las bibliografías y los índices, entre otros. Los datos que integran las fuentes secundarias se basan en documentos primarios. **(Reyes E. , 2022)**

La presente investigación hará uso de las siguientes fuentes secundarias:

- Artículos publicados en internet
- Libros sobre los temas principales
- Páginas web sobre información relacionada

### *3.2.2 Sujetos de Información.*

Mata (2021) menciona que el concepto de sujeto de información se refiere a las personas u objetos de estudio en investigaciones científicas, siendo la población de interés para el estudio. En este contexto, los sujetos de estudio pueden ser individuos o grupos cuyas características, opiniones, experiencias, entre otros aspectos, son relevantes para investigaciones cuantitativas o cualitativas.

La selección cuidadosa de fuentes confiables y la delimitación precisa de los sujetos de información son aspectos fundamentales para garantizar la coherencia, relevancia y originalidad de la investigación presentada en la tesis. En la Tabla 1 se presentan los sujetos de información pertinentes a esta investigación.

*Tabla 1 Sujetos de información consultados.*

| ***Puesto*** | ***Profesión*** | ***Experiencia*** | ***Relación con el tema*** |
|---|---|---|---|
| Gerente departamento EITS | Ingeniero en Sistemas | 11 años | Es quien supervisa a todo el equipo de EITS y se encarga del manejo y distribución de la carga de trabajo del equipo. |
| Equipo de Ingenieros de soporte Linux | Ingenieros en Sistemas | 6 a 8 años | Es quien provee el soporte a los servidores Linux manejados por el departamento de EITS. Incluyendo el proceso de parcheo. |
| Líder técnico del departamento de automatización | Diseñador de software | 6 años | Encargado del desarrollo y soporte de la plataforma de ansible para la automatización de procesos dentro de Experian. |
| Ingeniero de soporte de la aplicación ServiceNow | Analista de datos | 6 años | Encargado del soporte y mejora de las bases de datos de activos registrados de Experian en ServiceNow |

*Fuente: Elaboración propia*

### 3.3 Técnicas y Herramientas de Recolección de Datos.

En esta sección se detallan las técnicas y herramientas utilizadas en la investigación para la recopilación de datos e información pertinentes al tema a desarrollar.

Lifeder (2021) menciona que las técnicas de recolección de datos son herramientas utilizadas para recopilar información de manera organizada y con un propósito específico en campos como la investigación científica, empresarial, estadística y marketing. Estas técnicas se dividen en cualitativas, cuantitativas y mixtas, cada una con enfoques distintos para obtener datos numéricos precisos o información detallada sobre contextos y fenómenos sociales. Es crucial seleccionar las técnicas adecuadas según los objetivos de la investigación para obtener la información apropiada.

En el caso de la presente investigación, al ser de enfoque mixto, recolectara información cuantitativa y cualitativa a la vez.

### *3.3.1 Técnica Delphi*

La técnica o método Delphi es una técnica de consulta iterativa que utiliza la opinión de un grupo de expertos para llegar a un consenso sobre un tema específico. De acuerdo con Torres (2024) su origen y nombre se remonta al oráculo de Delfos en la antigua Grecia, y se utiliza para recopilar opiniones de expertos, planificar estrategias, tomar decisiones en situaciones inciertas, evaluar riesgos y predecir tendencias futuras. El Método Delphi ha sido aplicado con éxito en pronósticos científicos, tecnológicos, de negocios y políticas públicas, demostrando una alta precisión en comparación con otros métodos de pronóstico. Además, se ha utilizado en investigaciones de diversas áreas, como la salud, la educación y la gestión, permitiendo a expertos anónimos contribuir con sus opiniones de manera estructurada y controlada.

Para poder aplicar dichas técnicas se usaron las siguientes herramientas:

**Entrevista.** Lifeder **(2021)** lo define como una conversación planificada donde el investigador plantea preguntas o temas a uno o varios individuos para obtener información específica. Puede ser presencial, telefónica o virtual, siendo importante la interacción personal

en algunos casos para captar la comunicación no verbal. Se clasifica en estructurada (preguntas definidas previamente), semiestructurada (con guía de preguntas, pero flexibilidad para nuevas) e informal (sin preguntas definidas, temas introducidos espontáneamente). Esta herramienta se aplica al inicio del proceso de investigación para recopilar información necesaria para iniciar, e implica coordinar reuniones con los sujetos de información pertinentes.

**Observación.** Lifeder **(2021)** indica que la observación es una técnica que implica observar el fenómeno a analizar para obtener información cualitativa o cuantitativa. En investigación cualitativa, permite analizar relaciones y comportamientos, mientras que en la cuantitativa se usa para seguimiento de fenómenos o funcionamiento de máquinas. Enfoques cualitativos requieren categorizar las observaciones para un análisis ordenado, relacionándolas con información de otras técnicas para mayor validez. Esta herramienta se aplica durante las fases iniciales de la investigación para la compresión del proceso y situación actual, así como para determinar los avances en el diseño de la propuesta para la solución.

## 3.4 Variables de Investigación.

A continuación, se presentan las variables obtenidas de la investigación

*Tabla 2 Variables de investigación*

| Objetivo especifico | Variable asociada | Descripción |
|---|---|---|
| Analizar los diferentes procesos del ciclo parcheo, mediante la observación y estudio de la plataforma de parcheo actual para determinar los requerimientos mínimos y áreas de mejora a solventar. | Estructura del código | Analizar la organización del código en la plataforma actual. |
| | Lenguajes y tecnologías utilizadas | Investigar los lenguajes de programación y tecnologías empleadas en los scripts y código de la plataforma. |
| Identificar los roles y responsabilidades en cada una de las etapas del ciclo de parcheo, mediante entrevistas con los técnicos e ingenieros del departamento de EITS, para definir la correcta separación de deberes. | Etapas del proceso de parcheo | Describir y documentar cada etapa del proceso de parcheo actual. |
| | Roles y responsabilidades | Investigar quiénes son los responsables de cada etapa del proceso de parcheo y cuáles son sus funciones específicas. |
| | Flujo de trabajo | Analizar cómo se gestionan las transiciones entre las diferentes etapas del proceso de parcheo. |
| Definir el control de acceso a la calendarización del parcheo de servidores, mediante el análisis de las capacidades de la plataforma ServiceNow, para implementar una capa de seguridad al proceso de parcheo. | Frecuencia de parcheo | Determinar la frecuencia con la que se realizarán los parcheos de |
| | Procedimientos de emergencia | Identificar los procedimientos a seguir si es necesario realizar un parcheo urgente. |
| Elaborar un plan de integración de los objetivos anteriores, a través de una solución integral para el parcheo automático de servidores Red Hat. | Automatización de tareas | Determinar qué tareas del proceso de parcheo pueden automatizarse utilizando Ansible y cómo se implementarán. |

*Fuente: Elaboración propia*

## *3.5* Diseño de la Investigación.

En esta sección se presenta cómo se desarrolla el proyecto de investigación, incluyendo las diferentes etapas secuenciales, las técnicas que se realizarán y su función.

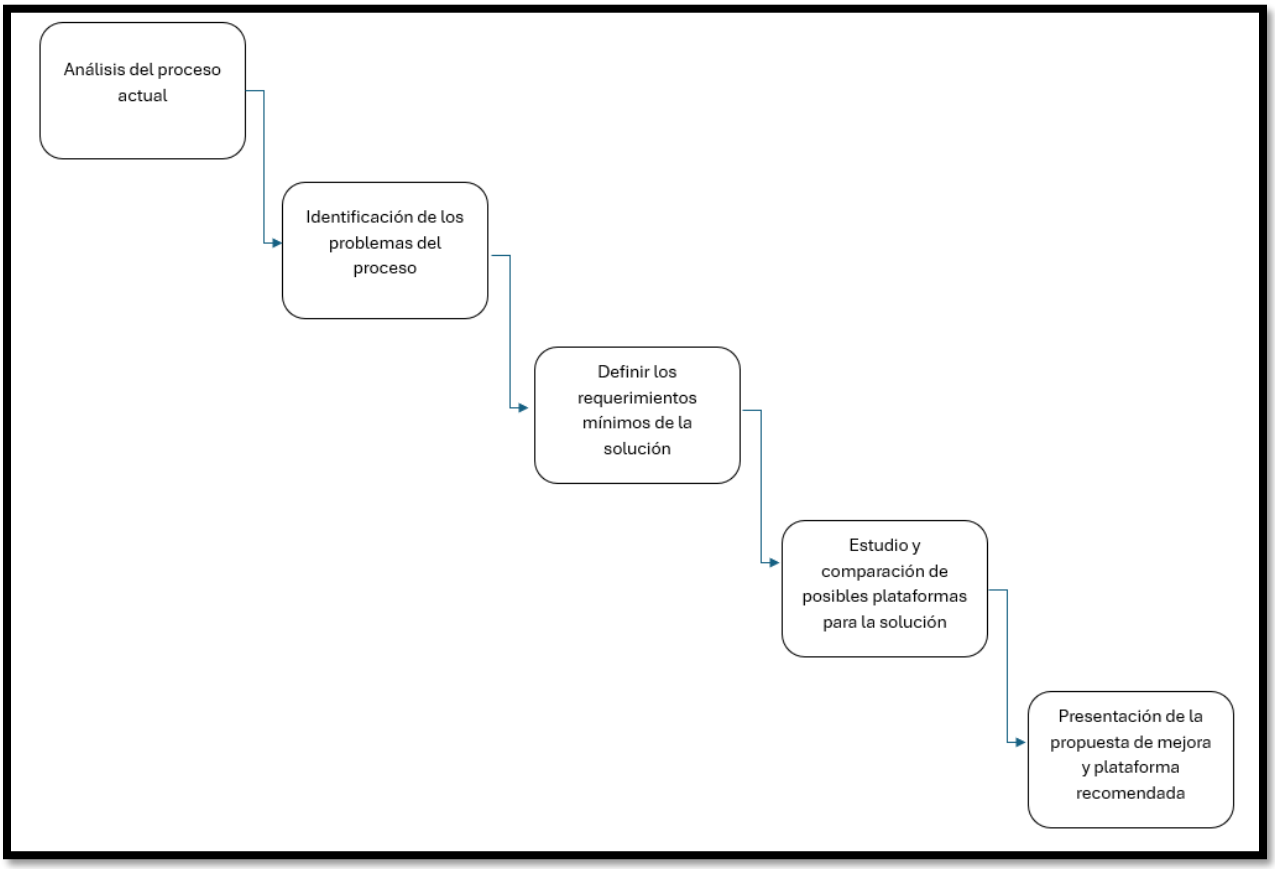


*Ilustración 19 Diseño de la Investigación. Fuente: Elaboración propia.*

### *3.5.1 Etapa 1. Análisis del proceso actual.*

En la Etapa 1, se recopilan datos relacionados con los procesos existentes y los requisitos clave a considerar en la automatización del parcheo de servidores. Para lograr esto, se emplean técnicas de entrevista con expertos involucrados en las diversas fases del proceso, junto con la aplicación de la técnica de observación del sistema de parcheo actual.

### *3.5.2 Etapa 2. Identificación de los problemas del proceso.*

En la Etapa 2, se evalúa la información recopilada en la fase previa para comprender la situación actual del departamento y del proceso identificando los problemas, carencias posibles áreas de mejora del mismo. A través de la técnica de observación y el análisis de los datos y requisitos definidos, se obtiene una visión detallada de las brechas existentes.

### *3.5.3 Etapa 3. Estudio y comparación de posibles plataformas para la solución.*

En esta etapa se analizan las opciones disponibles en el mercado que cumplan con los requisitos definidos, incluyendo un estudio económico, técnico y de capacidad de integración con aplicaciones y plataformas ya utilizadas por la empresa Experian.

### *3.5.4 Etapa 4. Presentación de la propuesta de mejora y la plataforma recomendada.*

En esta etapa se realiza la presentación de la propuesta a los sujetos de información involucrados en el proceso, y mediante una reiteración de las técnicas de entrevista y observación se presenta la plataforma que se recomienda para el desarrollo e implementación de la propuesta.

## 3.6 Matriz de Coherencia.

En esta sección se presenta una matriz de coherencia la cual brinda una relación entre objetivo, entregable, instrumentos y temas de marco teórico, entre otros aspectos.

*Tabla 3 Matriz de Coherencia.*

| Objetivo | Entregable | Fase / Etapa | Técnica de recolección de información | Instrumentos | Temas relacionados |
|---|---|---|---|---|---|
| *Analizar los diferentes procesos del ciclo parcheo, mediante la observación y estudio de la plataforma de parcheo actual para entender y comprender las distintas etapas, funciones, integraciones y resultados del proceso* | *Realizar un diagnóstico y un entendimiento de los procesos del ciclo de parcheo actual para trasladarlos a una nueva plataforma que permita la mejora de los mismos* | *Análisis del proceso actual* | *Entrevista*<br><br>*Observación* | *Documentación existente y acceso al proceso actual* | *Actualización de seguridad*<br><br>*Vulnerabilidades y Exploits*.<br><br>*Ciclo de parcheo* |
| *Identificar los problemas en cada una de las etapas del ciclo de parcheo, mediante entrevistas con los técnicos e ingenieros del departamento de EITS, para definir las áreas de mejora que la propuesta debe abordar* | *Hacer un diseño optimo de como deberían realizarse los diferentes procesos que comprenden el ciclo de parcheo de los servidores Linux Red Hat* | *Identificación de los problemas del proceso* | *Observación* | *Software de diagramas de flujo.*<br><br>*Casos de uso.* | *Ciclo de parcheo*<br><br>*Automatización procesos informáticos*<br><br>*Análisis de información* |
| *Analizar diferentes opciones para solventar los problemas identificados en los procesos, mediante un estudio de mercado de distintas plataformas disponibles y que cumplan con los requisitos mínimos, para seleccionar la herramienta idónea de implementar* | *Realizar un estudio de mercado de las herramientas que pueden ser posibles soluciones como mejora a la plataforma actual* | *Estudio y comparación de posibles plataformas para la solución.* | *Observación* | *Documentación existente* | *ITSM*<br><br>*ServiceNow*<br><br>*Diagramas de caso de uso*<br><br>*Modelado basado en eventos*<br><br>*Diseño de base de datos* |
| *Elaborar una propuesta que presente las mejoras definidas al proceso y la* | *Proponer la herramienta idónea que cumpla con los requerimientos del proceso de ciclo* | *Presentación de la Propuesta* | *Entrevista*<br><br>*Observación* | *Software de diagramas de flujo.*<br><br>*Casos de uso.* | *Automatización procesos informáticos*<br><br>*Shell scripts* |

| | | | | | |
|---|---|---|---|---|---|
| *plataforma recomendad para realizarlas, mediante el diseño de una solución integral que incluya los objetivos anteriores, para asegurar que cubra los requerimientos definidos* | *de parcheo e implemente las mejoras identificadas al proceso actual* | | | *BitBucket*<br><br>*ServiceNow Ansible Automation Platform* | *Serialización de datos*<br><br>*YAML*<br><br>*Integración de tecnologías* |

*Fuente: Elaboración propia.*

# Capítulo IV: Diagnóstico de la Situación Actual.

En la empresa Experian, el departamento de EITS tiene a cargo la administración del ciclo de parcheo de los servidores Red Hat soportados por la organización. Pese a la implementación de un proceso automatizado para este proceso, este tiene ciertas falencias e incumplimientos de los requerimientos actuales de los clientes, cuya infraestructura ha crecido y cambiado en los últimos años. Por tanto, esta solución fuera de soporte no puede solventar las necesidades del negocio.

Por otro lado, los problemas identificados por la misma plataforma durante el proceso, esperados hasta cierto nivel, están sobrecargando de trabajo al equipo de soporte por la naturaleza del proceso, el cómo presenta y asigna los problemas, y porque actualmente no hay ingenieros o técnicos que conozcan a esta antigua plataforma para mejorar o desarrollar nuevas características que resuelvan los problemas repetitivos y cuya causa se identifica, o desarrollar nuevas características que soporten actualizaciones a software adicional; además de que no hay contrato actual con el proveedor de la plataforma.

## 4.1 Diagnóstico Administrativo u Operativo.

A continuación, se explican los distintos procesos que conforman el ciclo de parcheo para los servidores de Linux Red Hat en conjunto con la plataforma actual de automatización llamada TrueSight Blade Logic .

### *4.1.1 Agregar un servidor al ciclo de parcheo.*

Para habilitar un servidor dentro del ciclo de parcheo, se crea un registro del servidor agregándolo una base de datos local dedicada a este proceso. La base de datos se encuentra en un cluster de servidores MySQL los cuales son utilizados para otras bases de datos administradas por otros departamentos y unidades de negocio. En el caso del proceso de parcheo, el diagrama entidad-relación de la base de datos es demasiado grande y complejo por

lo que es difícil de agregar en este proyecto, además de que contiene relaciones con otras tablas y bases de datos que por confidencialidad no pueden incluirse. Sin embargo, en la ilustración 20 se muestra un extracto del diagrama con la información que es relevante al proceso de parcheo.

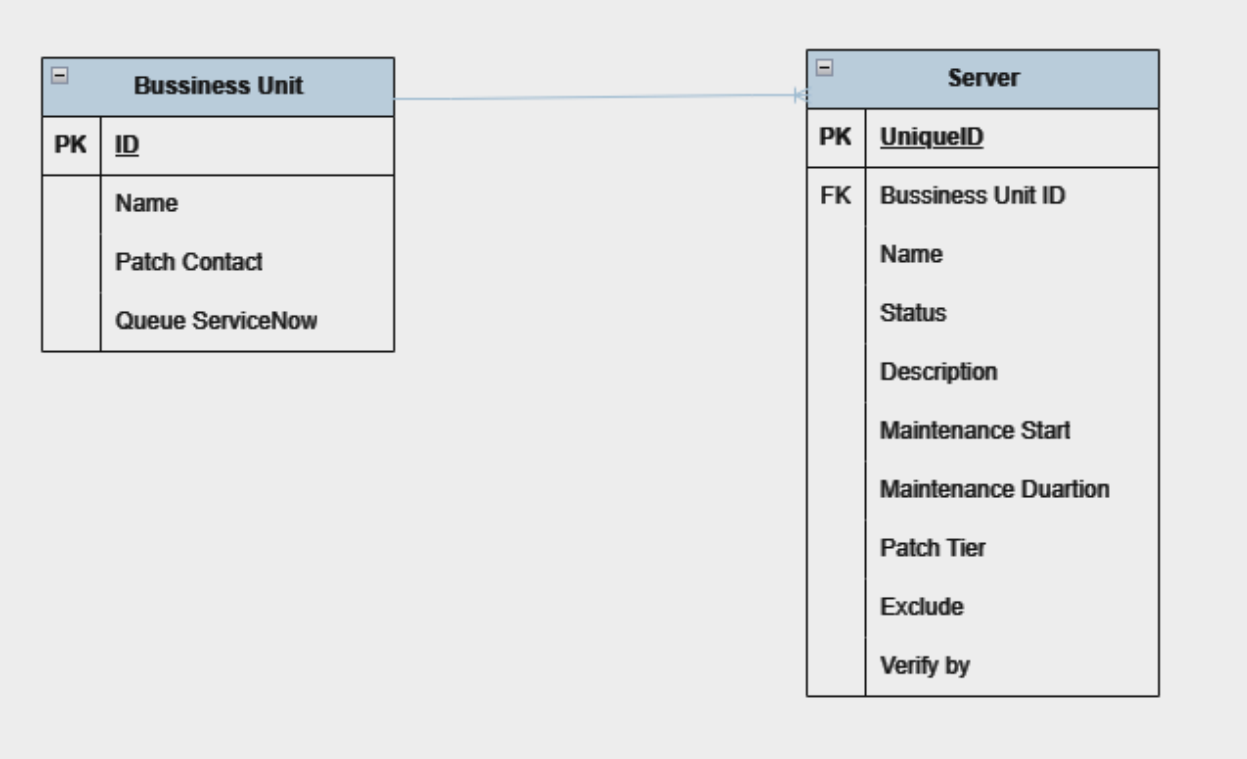


*Ilustración 20 Base de datos de parcheo actual. Fuente: Elaboración propia.*

De los ingenieros del departamento de EITS, solo uno tiene acceso a esta base de datos para agregar, modificar o eliminar registros de manera permanente mediante consultas de SQL. Debido a esta configuración, se presentan 2 escenarios:

- Para agregar nuevos servidores creados en el ambiente, los ingenieros a cargo de la configuración de dichos servidores dependen del ingeniero con accesos a la base de datos para agregarlos durante el proceso de creación y configuración.

- Para agregar servidores ya existentes en el ambiente, los clientes o dueños de dichos servidores deben solicitarlo mediante un formulario en ServiceNow (ver ilustración 21) que es asignado de manera aleatoria a los ingenieros de soporte de EITS, y estos a su vez deben coordinar con el Ingeniero que tiene acceso a la base de datos.

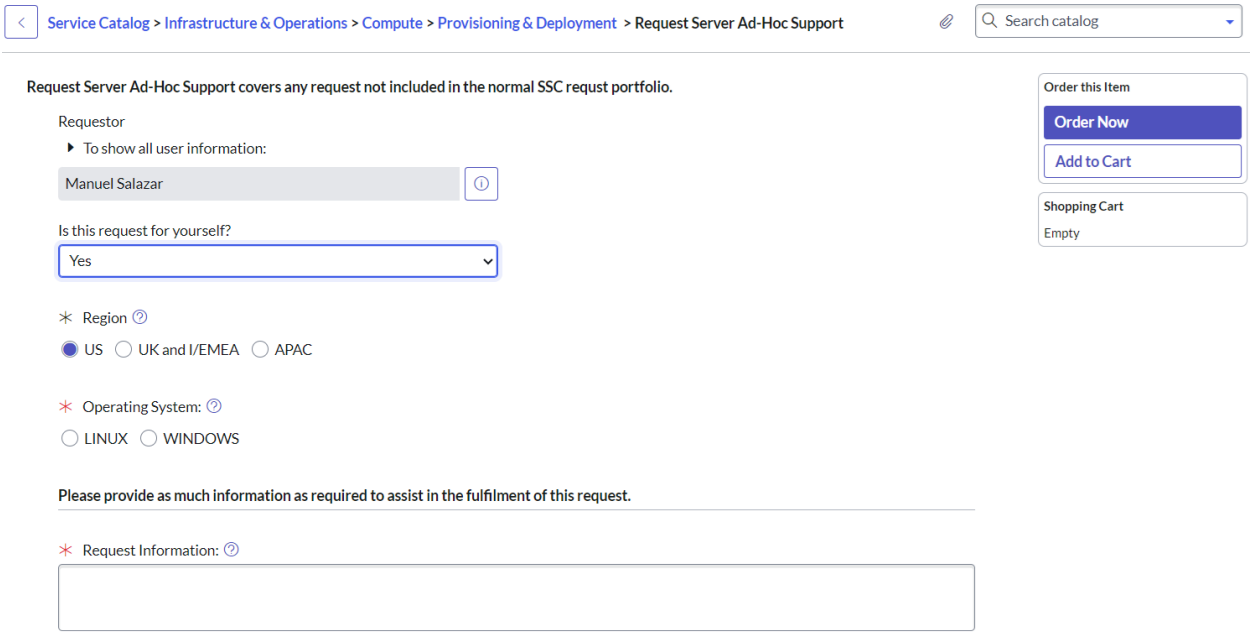


*Ilustración 21 Formulario de ServiceNow. Fuente: Experian.*

Las situaciones descritas anteriormente generan una sobrecarga de trabajo, así como dependencia sobre el ingeniero que tiene el único acceso administrador a la base de datos, dejando limitados a los clientes sobre control sobre sus propios servidores.

Esta dependencia puede causar retrasos en habilitar nuevos servidores cuyo proceso total, en lugar de tardar minutos, puede tomar horas o días dependiendo de la carga de trabajo y disponibilidad del ingeniero para agregar servidores nuevos a la base de datos de parcheo.

Actualmente no existe documentación para los ingenieros o los clientes sobre esta etapa del proceso. Los detalles se consideran como de conocimiento general por los clientes e

ingenieros y simplemente se transfieren a los nuevos integrantes de los equipos cuando se considere necesario.

### *4.1.2 Calendarización del servidor.*

Una vez agregado a la base de datos, el servidor es visible en un portal ASP Web diseñado solo para mostrar los registros y detalles de base de datos relacionados a la calendarización en el proceso de parcheo, tal y como se muestra en la ilustración 22.

| Server Name | Status | Description | Maintenance Start | Maintenance Duration | Patch Tier | Exclude | Verified By | |
|---|---|---|---|---|---|---|---|---|
| | Test / Dev | | 2021-03-11 01:00 AM | 150 min | Tier 10 | | | Update |
| | Production | | 2021-03-19 10:00 PM | 60 min | Tier 5 | | | Update |
| | Production | | 2021-03-31 09:00 PM | 480 min | Tier 20 | | | Update |
| | Production | | 2021-03-20 12:00 AM | 120 min | Tier 10 | | | Update |
| | Staging | | 2021-03-14 03:30 AM | 120 min | Tier 18 | | | Update |
| | Production | | 2021-04-03 06:00 AM | 120 min | Tier 9 | | | Update |
| | Staging | | 2021-03-20 08:30 AM | 480 min | Tier 20 | | | Update |
| | Production | | 2021-03-12 06:30 AM | 480 min | Tier 5 | | | Update |
| | Test / Dev | | 2021-03-13 04:00 AM | 480 min | Tier 10 | | | Update |

*Ilustración 22 Registros de la base de datos local en formato XML. Fuente: Experian*

A partir de este portal se pueden crear ‘vistas’ al editar el URL de la página y colocar en un parámetro en específico el ID de la Unidad de Negocio que se desea consultar, basados en los registros guardados en la base de datos (ver ilustración 20). De esta forma se mostrarán todos los servidores relacionados a ese ID. Esto permite crear 2 tipos de vistas:

1. Vista Administrador: Orientada a todos los ingenieros del departamento EITS. Muestra todos los registros de la base de datos y permite realizar cambios como cambiar la fecha de parcheo, cambiar la ventana de mantenimiento, cambiar la categoría de parcheo y excluir el servidor del parcheo de manera temporal, es decir, en el próximo ciclo de parcheo cualquier cambio será revertido.

2. Vista Cliente: Orientada a los clientes. Muestra los registros de los servidores de los que los clientes son dueños, basado en el URL editado de la página con el ID de su unidad de negocio. Permite realizar cambios como cambiar la fecha de

parcheo, cambiar la ventana de mantenimiento, cambiar la categoría de parcheo y excluir el servidor del parcheo de manera temporal, es decir, en el próximo ciclo de parcheo cualquier cambio será revertido.

El departamento de EITS cuenta con un grupo de 3 script hosts con sistema operativo Linux, los cuales tienen el objetivo de ejecutar ciertos scripts de manera automatizada. Uno de estos scripts envía a todas las Unidades de Negocio una notificación al inicio de cada ciclo de parcheo, mediante el correo asociado a cada una en la base de datos (Patch Contact en ilustración 20), con el link de la vista para que puedan acceder, validar y cambiar detalles en la calendarización de los servidores según lo vean necesario.

En la ilustración 23 se muestra en detalle las interacciones entre los elementos mencionados en esta etapa

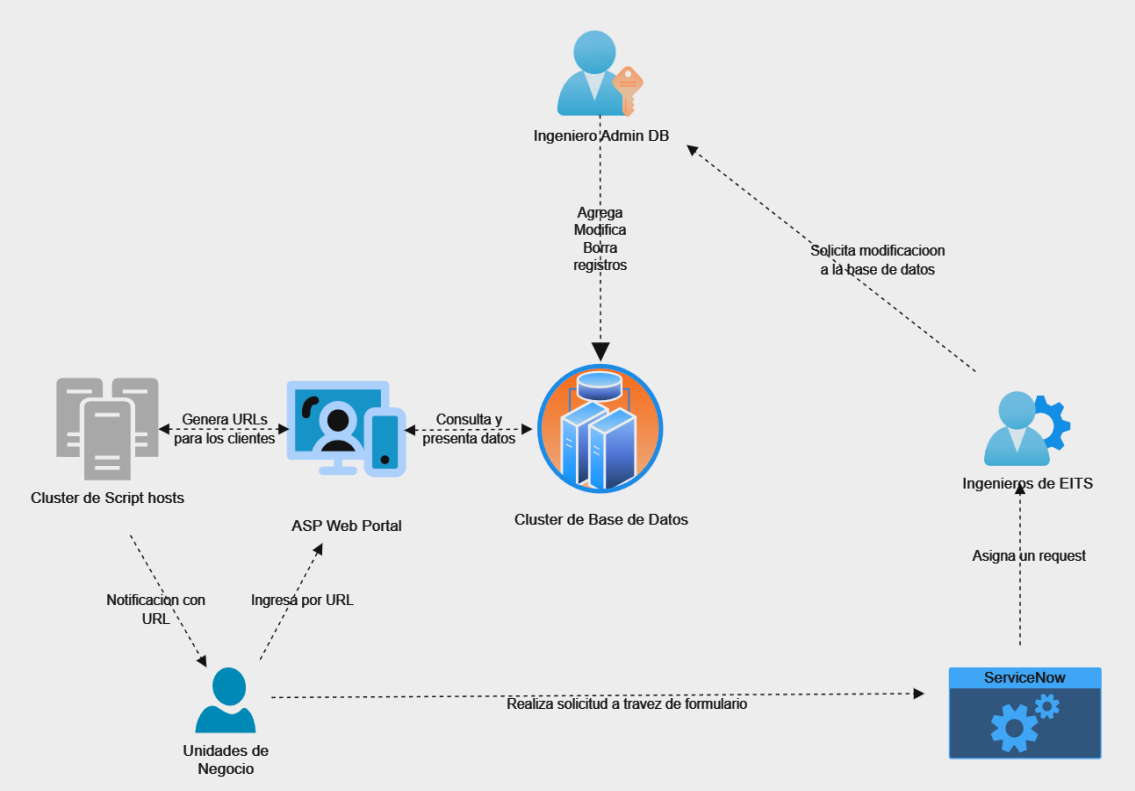


*Ilustración 23 Interacción con la base de datos. Fuente: Elaboración propia.*

Estos links generados, incluyendo el de administrador, no tienen ningún tipo de control de acceso, lo cual genera un gran riesgo de seguridad e integridad del proceso ya que, en la eventualidad de un error o filtro de información, si un miembro de una unidad de negocios accede a una vista que no es la de su propio grupo, puede de manera intencional o no intencional modificar las fechas de parcheo de servidores críticos de manera que pueda causar interrupción de servicios importantes a clientes internos y externos.

Lo mismo sucede con la vista de Administrador, ya que los cambios no estarían limitados una única unidad de negocio, y podrían generarse interrupciones de servicio en los servidores de múltiples unidades de servicio al mismo tiempo.

### *4.1.3 Creación de la tarea de parcheo.*

Con el servidor creado en la base de datos y su fecha de parcheo definida en el registro, la siguiente etapa corresponde a la plataforma de automatización TrueSight BladeLogic en la cual se han desarrollado tareas automatizadas para los siguientes procesos:

- Identificación de los servidores por parchear en las próximas 24 horas.
- Creación y ejecución de la tarea de pre-parcheo.
- Creación y ejecución de la tarea de parcheo.
- Validación de errores y creación de incidentes.

En esta sección nos enfocaremos en las tareas de identificación de servidores y creación de tareas de pre-parcheo. El proceso consiste en escanear el registro de la base de datos de parcheo e identificar todo servidor cuya fecha de parcheo este definido entre la fecha y hora en la que se escanea el registro de la base de datos, y las próximas 24 horas. Esta tarea está programada a ejecutarse cada hora sin excepción y consiste en un script desarrollado en

lenguaje NSH que realiza consultas MySQL a la base de datos. La tarea tiene configurados los controles para omitir cualquier servidor que no cumpla el criterio anteriormente mencionado.

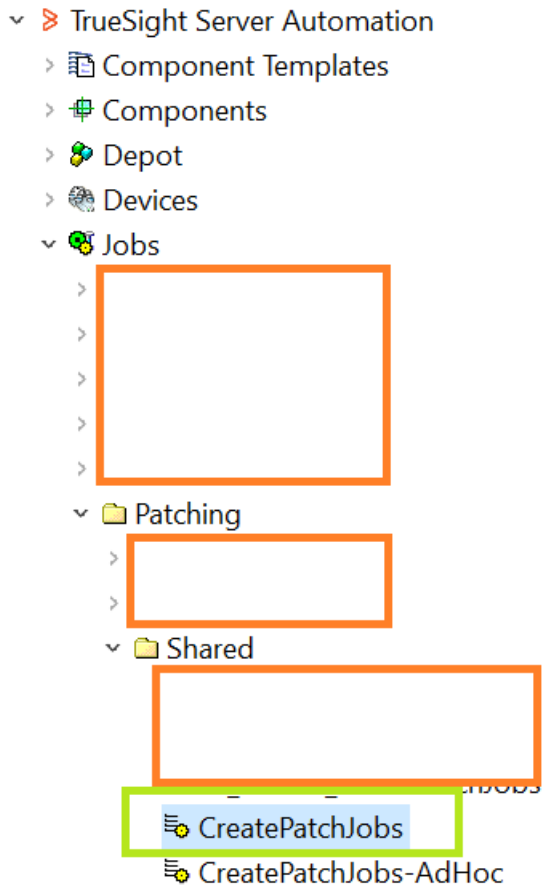


*Ilustración 24 Tarea de escaneo y creación de tareas programadas. Fuente: Experian.*

Cuando la tarea programada a definido la lista de servidores, procede a crear un 'schedule' o ejecución programada de las tareas de pre-parcheo y parcheo, tomando en consideración lo siguiente:

- La tarea de pre-parcheo se calendariza 4 horas antes de la fecha de parcheo definida para el servidor en la base de datos.
- La tarea de parcheo se calendariza a la hora y fecha definida para el servidor en la base de datos.

- Se crea una tarea de pre-parcheo y una de parcheo por cada servidor en la lista definida

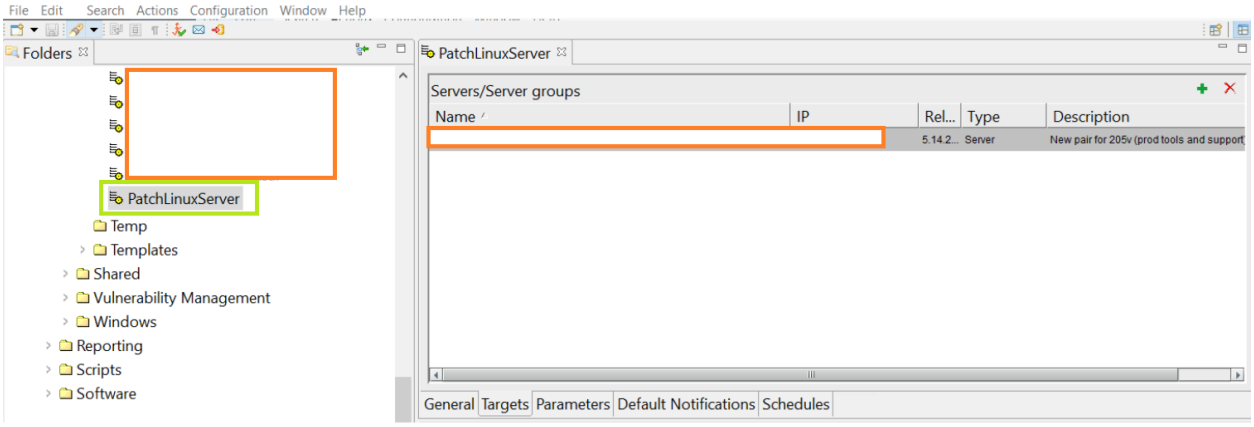


*Ilustración 25 Ejemplo tarea de parcheo programada para un servidor. Fuente: Experian.*

Una vez que se han creado estas tareas programadas para un servidor, el proceso de calendarización se considera completado y cualquier cambio en la fecha de parcheo, ya sea por parte de los clientes o ingenieros del departamento de EITS en la base de datos no tendrán ningún efecto.

Esta situación, obliga a los clientes asegurarse de que sus servidores estén listos para parchear 24hrs antes de la fecha establecida. De lo contrario, si se requiere cambiar o excluir un servidor que ya fue agregado al archivo de calendarización, debido a un imprevisto, cambio o problema de último momento, los clientes deberán crear una solicitud en el formulario de ServiceNow (ver Ilustración 21) para los ingenieros del departamento de EITS, ya que estos deberán ingresar a la plataforma de BladeLogic, buscar en la tarea programada de pre-parcheo o parcheo el registro relacionado al servidor que se debe cambiar o eliminar, y realizar los cambios requeridos (ver Ilustración 25).

Al ser un proceso completamente manual es susceptible a error humano, y por su complejidad ha causado problemas mayores en el pasado cuando se han requerido estos

cambios de última hora. En un ambiente optimizado, el proceso debería estar 100% bajo el control de los clientes sin ninguna intervención adicional requerida al cambiar la fecha de parcheo del servidor.

### *4.1.4 Creación y ejecución de la tarea de parcheo.*

Con la tarea de parcheo creada y calendarizada, la plataforma de TrueSight BladeLogic iniciara el parcheo en la hora y fecha indicados. El proceso de parcheo consta de 3 etapas llamadas a lo interno de la empresa y departamento de EITS como PrePatch, Patch y PostPatch. En cada una de las etapas se guarda la información o logs del proceso y su resultado, ya sea exitoso o fallido, en archivos locales de cada servidor parcheado (ver ilustración 26).

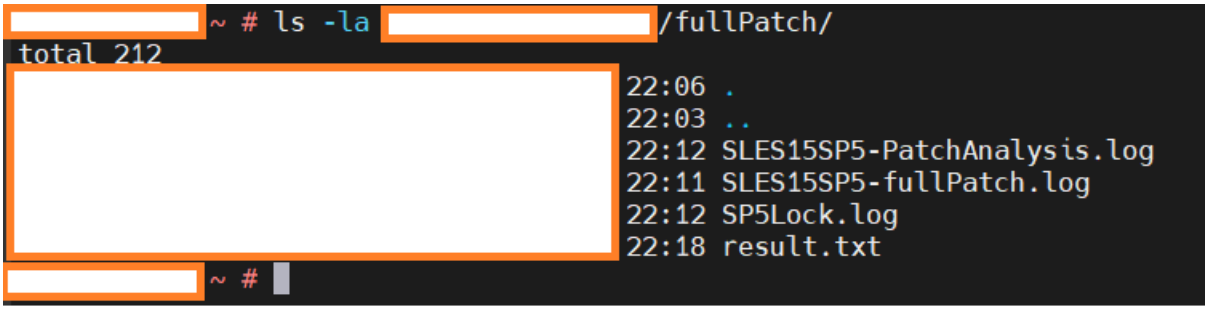


*Ilustración 26 Logs de información. Fuente: Experian*

Sin embargo, en las etapas PrePatch y PostPatch también hay scripts y procesos definidos por los clientes dueños de los servidores, los cuales son desarrollados y soportados por ellos mismos, y sobre los cuales el equipo de EITS no tiene ninguna responsabilidad.


*Ilustración 27 Ejemplo pre-Patch y post-Patch scripts. Fuente: Experian*

Durante la ejecución de las tareas automatizadas de parcheo, el código de parcheo contiene validaciones necesarias para la detección de errores en la ejecución de las mismas. Al detectar un fallo el proceso guarda información respectiva a la causa del error y un mensaje relacionado en los logs que almacena localmente en el servidor sobre el que ejecuta el parcheo (ver ilustración 28) como guía para que el ingeniero del departamento de EITS realice la investigación del problema una vez le sea asignado.

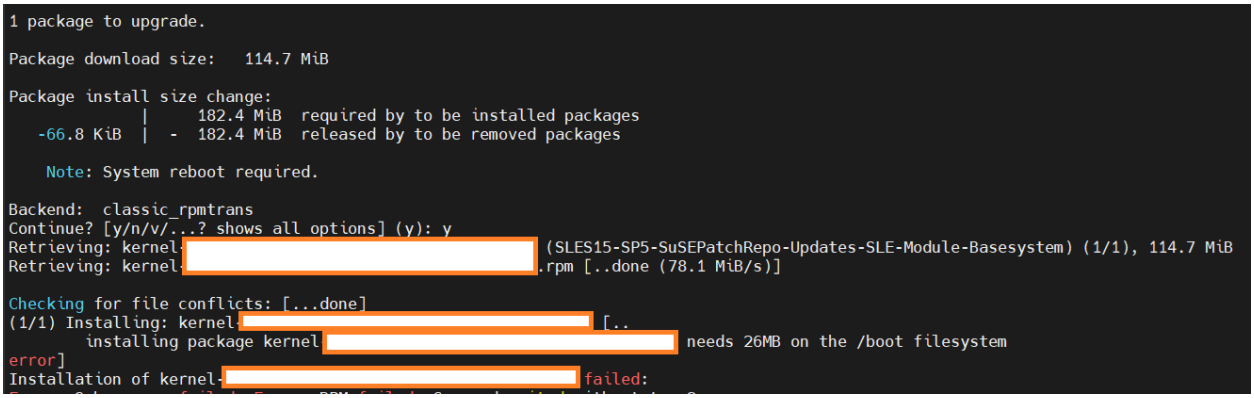


*Ilustración 28 Ejemplo de error de parcheo. Fuente: Experian.*

### *4.1.5 Generación de incidentes o tiquetes de soporte.*

Dentro del código de las tareas automatizadas de parcheo se han programado las validaciones necesarias para que al final de la ejecución, la tarea crea dentro de una ubicación en el cluster de script hosts anteriormente mencionados, un archivo con el nombre del servidor y con un 0 si el proceso completo sin problemas, o un 1 cuando un error ocurre. En el cluster de script hosts se encuentra una PowerShell script que se ejecuta cada hora y que escanea esta ubicación para enviar una notificación de éxito o fallo a las unidades de negocio, respecto al parcheo de sus servidores haciendo consulta a la base de datos y relacionando la información.

A la misma vez esta tarea se encarga de crear un incidente mediante una integración con la plataforma de ServiceNow, utilizada por la empresa Experian globalmente para el

manejo de tickets de soporte. Esto lo hace gracias a la una llamada del API de ServiceNow dentro del código de programación del script  y lo asigna automáticamente al equipo de EITS ya que así es como se encuentra definido (ver ilustración 29). Debido a esto, todo incidente durante el proceso de parcheo es asignado al departamento de EITS


*Ilustración 29 Llamada al API de ServiceNow. Fuente: Experian*

Esta configuración en la llamada del API de ServiceNow ocasiona que, aunque el problema esté relacionado a los scripts no soportados por el departamento de EITS y que deben ser corregidos por los dueños de los servidores, sean de igual forma asignados a los ingenieros de EITS quienes deben estar revisando y reasignando los tiquetes a quienes corresponda. Esta situación también puede escalar aún más cuando no hay registro o contacto de los dueños de los servidores por lo que los incidentes no se pueden reasignar y los problemas quedan sin resolver por uno o varios meses o ciclos de parcheo.

### *4.1.6 Manejo y resolución de incidentes o tiquetes de soporte.*

Como se mencionó en apartados anteriores, los incidentes creados durante el proceso de parcheo son asignados al departamento de EITS automáticamente y estos pueden ser

listados en la plataforma de ServiceNow, en el queue creado para este departamento, tal y como se muestra en la ilustración 30.

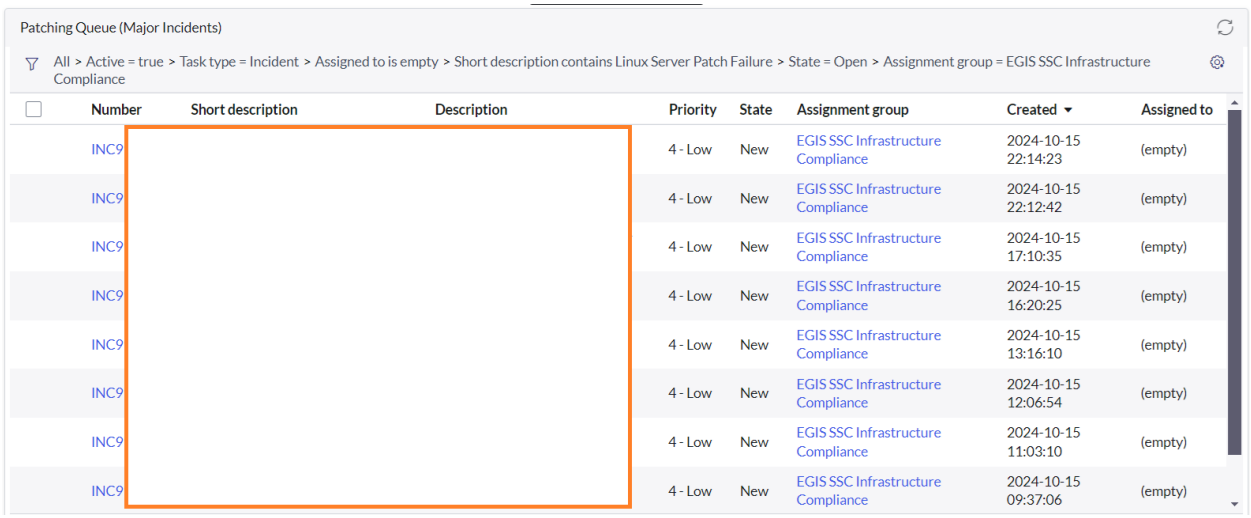


*Ilustración 30 Queue de Incidentes del departamento de EITS. Fuente: Experian*

Cada uno de los tiquetes cuenta con información relacionada a los registros de los servidores en la plataforma de ServiceNow más otros detalles que no son relevantes al proceso. Para efectos del proyecto nos enfocaremos en:

- Numero de Incidente
- Nombre del servidor
- Ambiente
- Grupo al que se asigna
- Ingeniero al que se asigna
- Descripción Corta
- Descripción

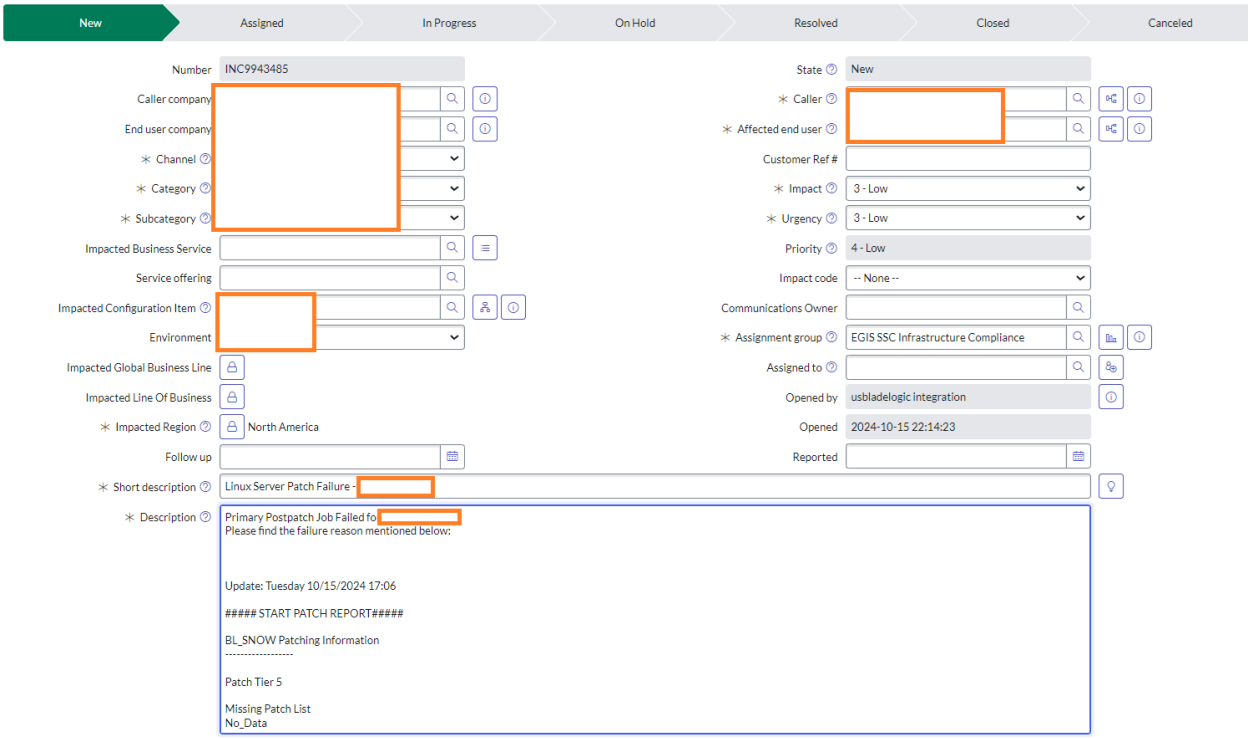


*Ilustración 31 Ejemplo de Incidente de parcheo. Fuente: Experian.*

En la ilustración 31 se evidencia que el detalle del error o problema encontrado durante el proceso de parcheo no se agrega al tiquete. Esto debido a que la información relacionada a los problemas o éxito del proceso de parcheo se almacena localmente en cada servidor (ver ilustración 26).

Los incidentes, por defecto, no son asignados a los técnicos y este proceso es realizado manualmente por el líder técnico del grupo de ingenieros de soporte de servidores del departamento de EITS. Esta tarea requiere en ocasiones de varias horas al día por parte del ingeniero ya que no consta de solo asignar los tiquetes, debe analizarlos, determinar cuáles son parte del soporte del departamento de EITS y cuales son de otro departamento o de los mismos clientes (en el caso de los pre-patch y post-patch scripts). Debe también identificar patrones como identificar múltiples tiquetes relacionados al mismo servidor, en cuyo caso deberán ser asignados a un mismo ingeniero para determinar si hay un problema mayor sobre el cual enfocarse.

A la hora de trabajar en los incidentes, para realizar validaciones o para intentar solucionar un problema de parcheo, los ingenieros necesitan ingresar a cada servidor para analizarlo y debido a la limitada información del tiquete, tanto el ingeniero que asigna tiquetes como el ingeniero que los trabaja, no tienen un conocimiento de antemano de la complejidad del problema o de cuánto tiempo tomara resolverlo.

Cada ingeniero trabaja 8 horas al día, 48 horas semanales, las cuales debe dedicar al soporte de servidores Linux que incluye resolver otros problemas adicionales a los incidentes creados por parcheo, también a documentación, entrenamientos y demás tareas relacionadas a su rol. Basados en esta información, en experiencias comentadas por los mismos ingenieros en las entrevistas y en los detalles mencionados del ciclo de parcheo podemos crear el siguiente escenario:

Si a un ingeniero le son asignados 10 tiquetes de problemas de parcheo de distintos servidores, deberá ingresar a cada uno a los 10 servidores para iniciar su investigación y definir si debe ser solucionado por el departamento de EITS o por los dueños de los servidores, reasignarlo o iniciar el proceso para resolverlo, y/o escalar el problema al siguiente nivel si se requiere. Ante esta situación, el ingeniero puede invertir de 30min a 1hr en un solo incidente, que sumado a los 10 nuevos que tiene asignados, e incluidos los que pueda arrastrar de semanas o días anteriores, puede causar que dedique 8 o más horas al día y por consiguiente puede tomar entre el 80 y 90 % del tiempo del ingeniero en su día o semana en solo incidentes de parcheo, dependiendo de la cantidad de tiquetes por revisar y sin tomar en cuenta otro tipo de incidentes, solicitudes o proyectos que pude estar trabajando.

### *4.1.7 Realizar cambios o mejoras al código de parcheo.*

El código de parcheo se encuentra programado y configurado directamente en la plataforma de TrueSight BladeLogic, y consta de una serie de tareas programadas en lenguaje de programación NSH.

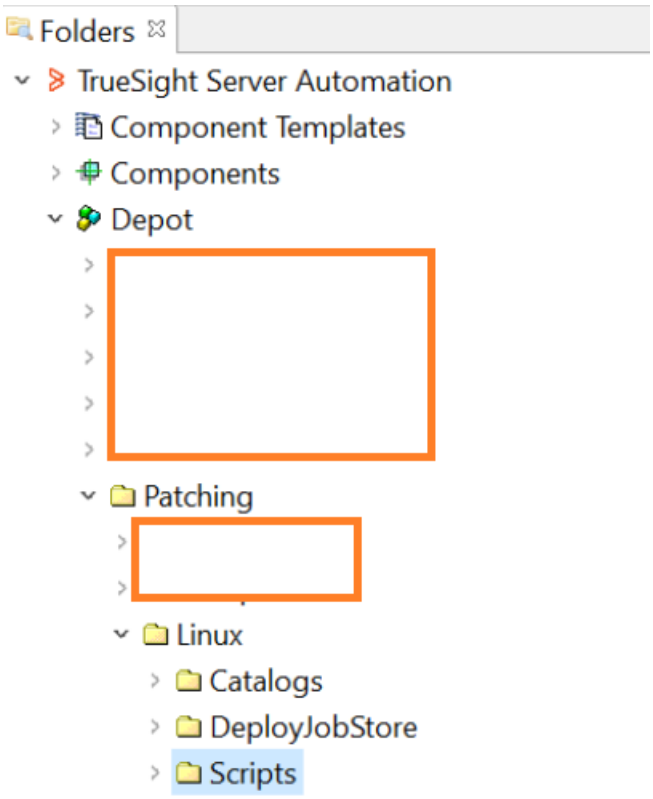


*Ilustración 32 Ubicación código de parcheo en BladeLogic. Fuente: Experian*

Como se muestra en la ilustración 32, el código está separado en archivos contenidos dentro de una estructura de directorios que sigue las mimas reglas y funciones que la estructura de directorios del sistema Operativo Windows. Por lo tanto, no existen características o funcionalidades relacionadas a los repositorios de código como colaboración de diferentes desarrolladores de manera simultánea, control de revisiones, control de publicaciones, validaciones o prueba de funcionalidades antes de ser publicadas.

El conocimiento general dentro del departamento de EITS relacionado a este código, la forma de crear nuevas características o modificar las existentes también se encuentra muy limitado o es casi inexistente ya que no hay documentación tanto a lo interno del departamento, como proporcionada por el proveedor. Los últimos ingenieros que tenían este conocimiento no forman parte del departamento o han iniciado labores en otras empresas, por lo que la

generación de ingenieros actual ha quedado limitada en su capacidad para realizar cambios o resolver problemas relacionados al código de parcheo.

Al combinarse estas situaciones, se han causado múltiples errores e incrementos inesperados en la cantidad de incidentes de parcheo en el pasado, no relacionados a problemas en los servidores, si no al proceso, debido a errores en el desarrollo o cambio del código. Esto ha llevado a mantener una estructura y código de parcheo sin mayor modificación o actualización por casi de 3 años.

Las pocas características que se han agregado, debido al nivel de complejidad han tardado meses en realizarse y al no haber controles o mejores prácticas en el desarrollo de las mismas, no existe documentación o incluso comentarios en el código de exactamente como se crearon o que funciones realizan. Lo que conlleva a gran inversión de esfuerzo y tiempo en entender estos procesos.

## 4.2 Diagnóstico Técnico

A continuación, se detalla a profundidad las infraestructuras que interactúan en el proceso de parcheo actualmente presente para los servidores Red Hat en la empresa Experian.

### *4.2.1 Infraestructura de servidores Red Hat Soportados.*

La empresa Experian cuenta con múltiples unidades de negocio a nivel local y global, las cuales cuentan con recursos y servidores en distintos datacenters, ya sea de forma física o en servidores de ESXi VMWare. Debido a la naturaleza de cada unidad de negocio, nivel de confidencialidad o por su ubicación geográfica, el soporte de dichos servidores estará asignado a diferentes grupos o departamentos. En nuestro caso nos enfocaremos en servidores con sistema operativo Linux Red Hat 7.9, Linux Red Hat 8.X y Linux Red Hat 9.X que pertenecen a las Unidades de Negocio soportadas por el departamento de EITS en Norte América, los cuales

están distribuidos en 2 datacenters, para un total de 1657 servidores según el último registro: Sin embargo, debido al constante crecimiento de la infraestructura, utilizaremos un aproximado de 2000 servidores para los cálculos y estimaciones de la propuesta. Todos estos servidores cuentan con muy distintas configuraciones de CPU, RAM y almacenamiento dependiendo de las necesidades de cada unidad de negocio.

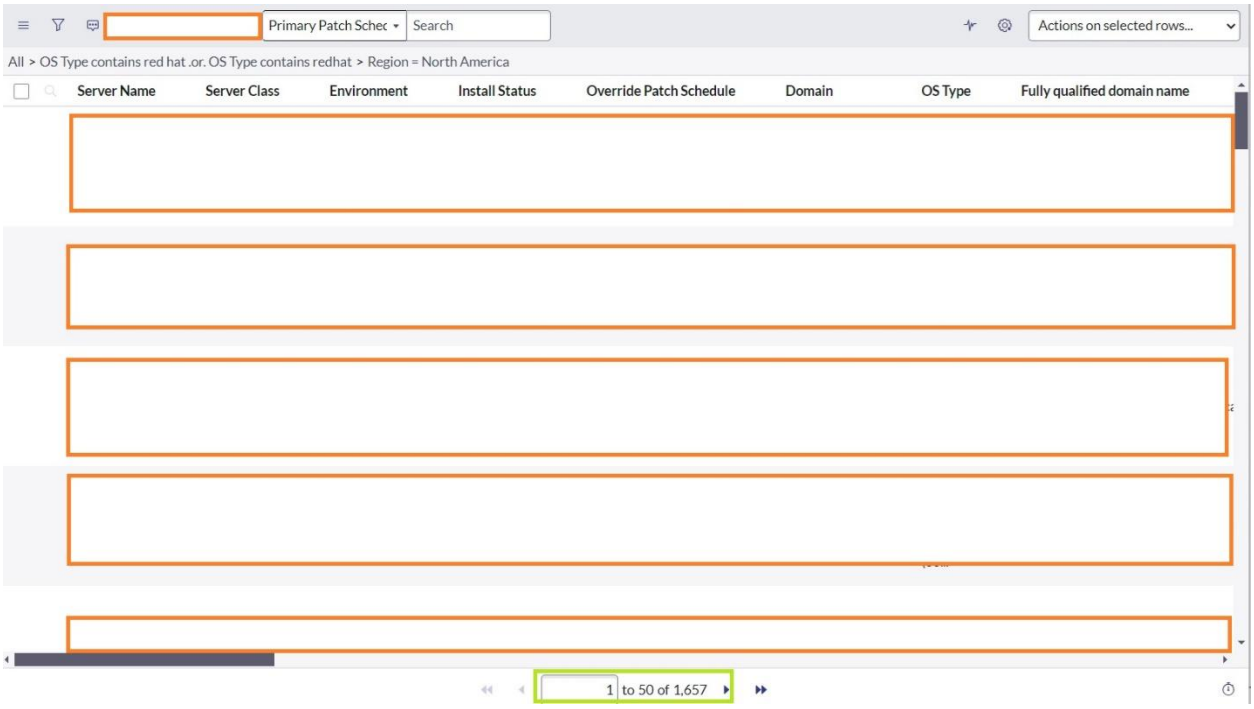


*Ilustración 33 Cantidad de servidores soportados por EITS. Fuente: Experian.*

Como se puede detallar en la ilustración 34, ambos datacenters están intercomunicados a través de la red privada de Experian y por defecto no tienen acceso a internet a menos que sea a través de un proxy server que necesita ser configurado directamente en los servidores. A su vez, los servidores son accesibles desde las plataformas de TrueSight BladeLogic, ServiceNow y Red Hat Satellite, ya que estas se encuentran habilitadas en los mismos datacenters y las reglas de firewall están configuradas para permitir la comunicación entre estas plataformas y los servers en ambos datacenters ya sea por conexiones directas de SSH o a través de agentes como es el caso de ServiceNow y BladeLogic.

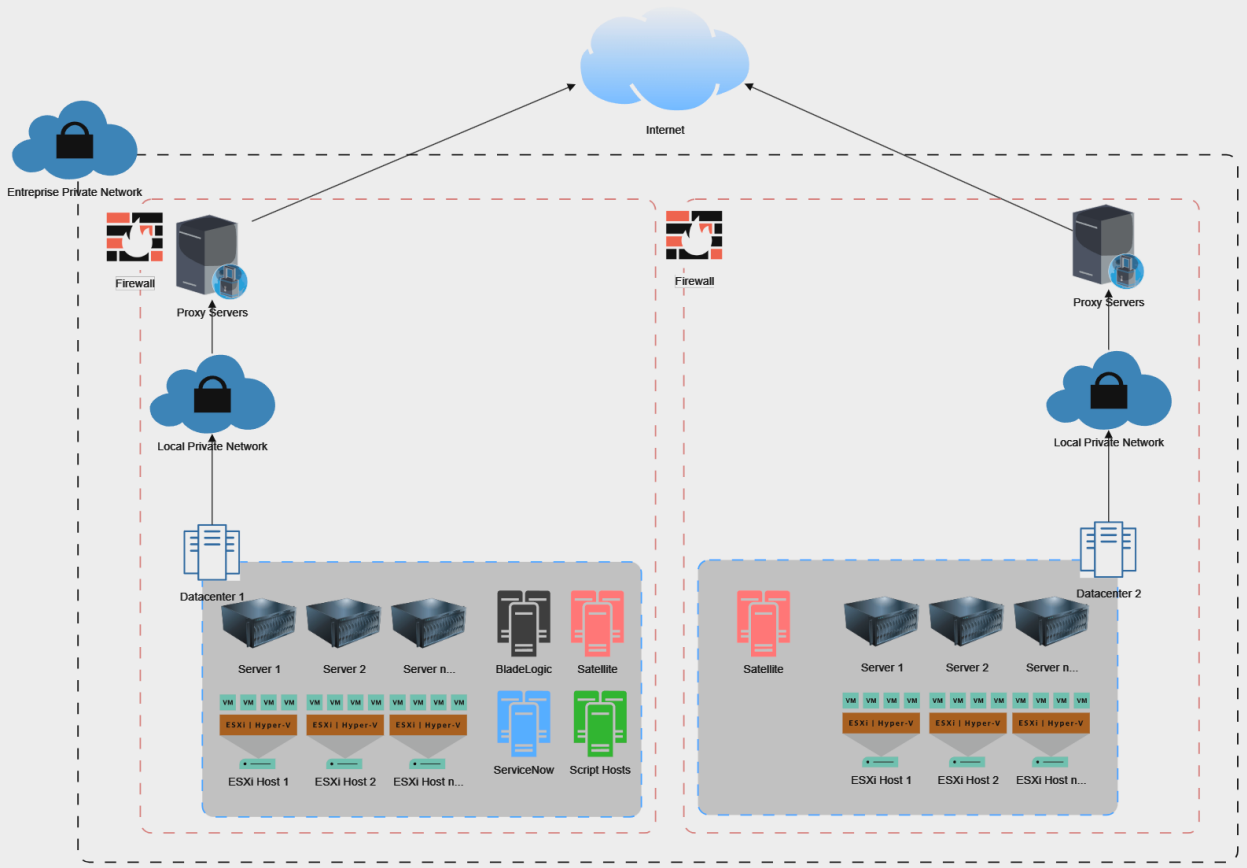


*Ilustración 34 Infraestructura de Servidores soportados por EITS. Fuente: Experian.*

### *4.2.2 Infraestructura de la Plataforma TrueSight BladeLogic*

La plataforma de automatización utilizada actualmente para el proceso de parcheo es TrueSight BladeLogic, desarrollada por BMC Software para la administración de servidores, redes y dispositivos de almacenamiento, permitiendo a la organización el automatizar tareas críticas de administración.

En la empresa Experian, BladeLogic se encuentra configurada en una combinación de servidores Admin configurados en cluster, y servidores de ejecución configurados con un load balancer.

Respecto a los servidores Admin, consiste en 2 servidores virtuales almacenados en un VMWare ESXi host diferente cada uno y configurados en cluster a nivel de aplicación para

redundancia. Cada uno utiliza sistema operativo Windows Server 2016 con las siguientes especificaciones básicas:

- CPU: 4
- RAM: 32GB
- Almacenamiento: 4TB y 100GB

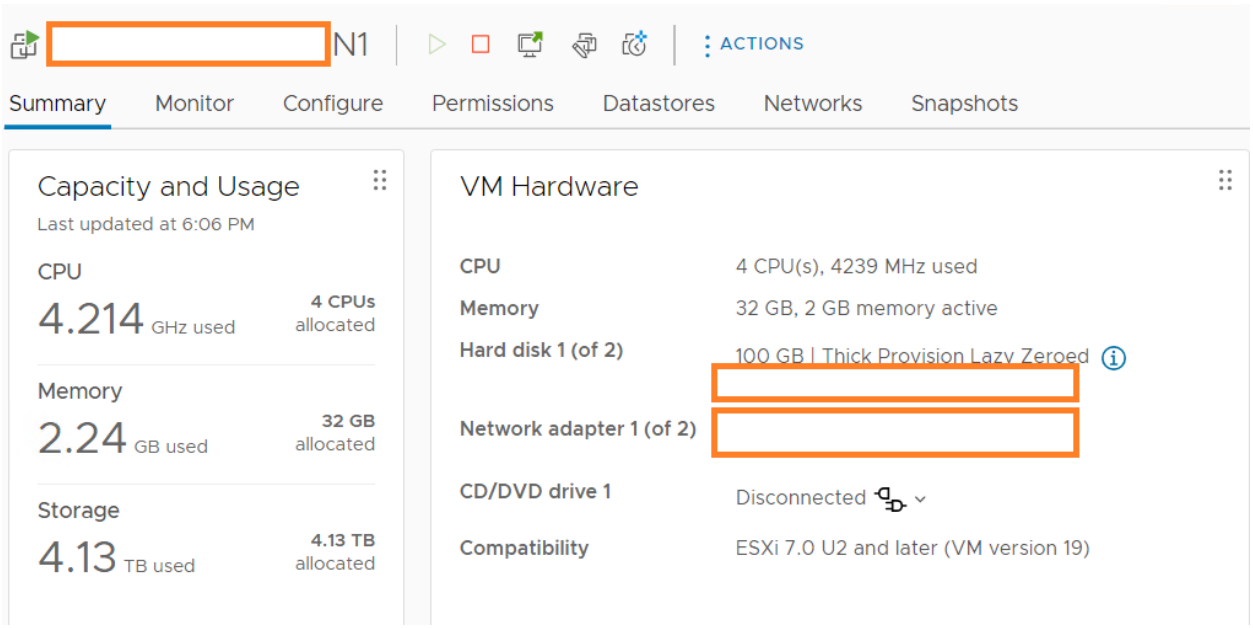


*Ilustración 35 Servidor Admin de BladeLogic. Fuente: Experian.*

En cuanto a los servidores de ejecución, estos consisten en un grupo de 6 servidores virtuales almacenados en 3 VMWare ESXi hosts diferentes con un load balancer de aplicación para redundancia. Cada uno utiliza sistema operativo Windows Server 2016 con las siguientes especificaciones básicas:

- CPU: 12
- RAM: 32GB
- Almacenamiento: 100GB y 80GB

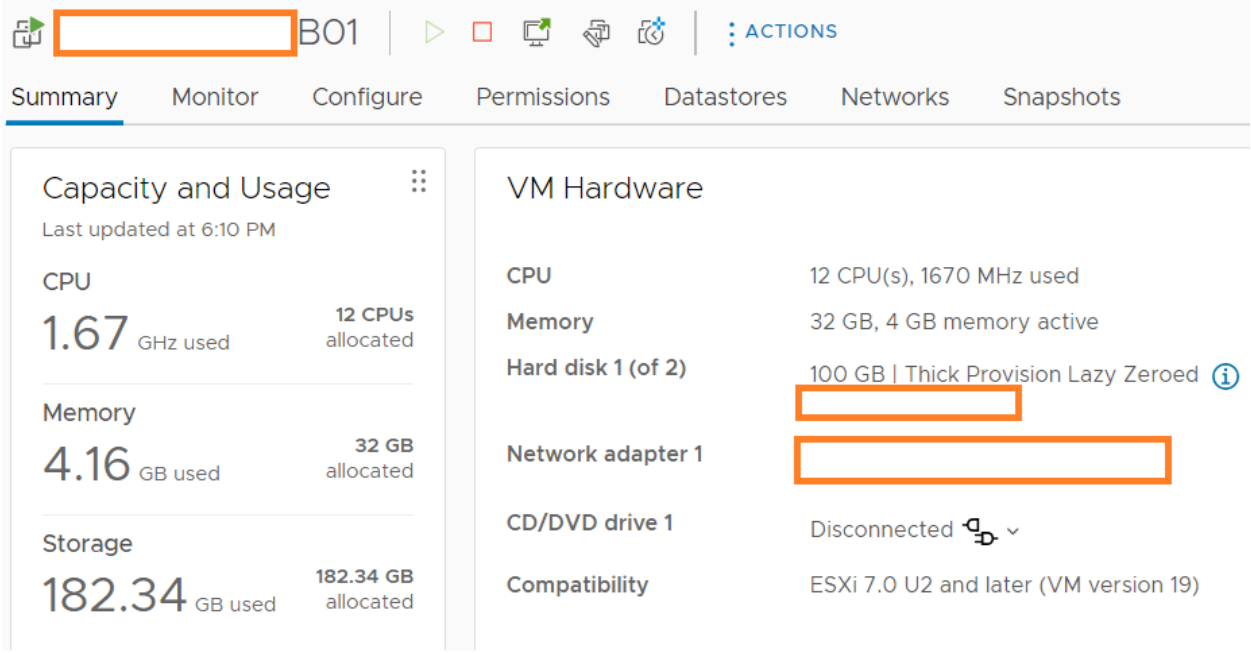


*Ilustración 36 Servidor de ejecución de BladeLogic. Fuente: Experian .*

El código de parcheo y todas las tareas programadas se encuentran alojadas en los servidores Admin. Al ejecutarse cualquiera de estas tareas, la plataforma asigna el proceso a uno de los servidores de ejecución a través del load balancer. A nivel interno de los servidores de ejecución la tarea es tomada por el host con menor carga de trabajo. Estos servidores son los que interactúan con la base de datos de parcheo en MySQL para las consultas durante el proceso de parcheo y los que ejecutan las tareas de parcheo sobre los servidores soportados por el departamento de EITS.

La comunicación entre los servidores de ejecución y todos los servidores Red Hat soportados, se realiza mediante un agente que es previamente instalado en los servidores en el momento de su configuración, ya sea como servidores virtuales o físicos. En caso de ser necesario de reinstalar el agente, o si un servidor lo pierde, los ingenieros de EITS tiene acceso a los instaladores en un share local desde el que pueden instalarlo en el servidor requerido para reestablecer la comunicación con la plataforma.

Los ingenieros de EITS tienen acceso de administrador a cualquiera de estos servidores de manera directa a través de conexiones remotas (RDP), y también son dueños de los registros de los servidores en la plataforma de ServiceNow, lo que les permite realizar cambios a dichos registros.

En la ilustración 37 se muestra a detalle la infraestructura de esta plataforma y una vista general de sus interacciones con los ingenieros u otras plataformas.

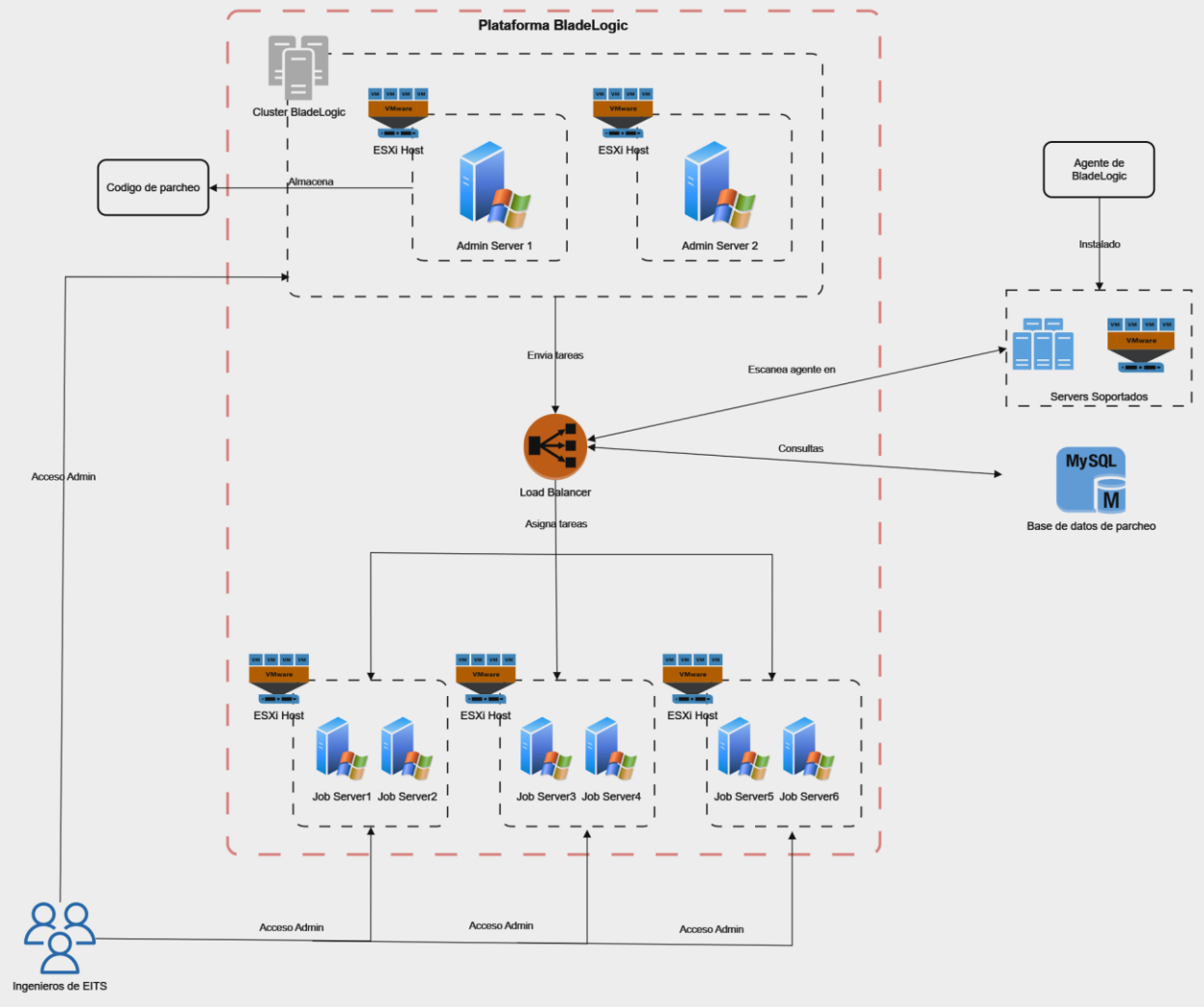


*Ilustración 37 Infraestructura TrueSight BladeLogic. Fuente: Elaboración propia.*

### *4.2.3 Infraestructura de la Plataforma ServiceNow*

ServiceNow es una plataforma en la nube que brinda integración con soluciones para gestionar servicios empresariales y operaciones de TI. Su principal objetivo es optimizar y automatizar procesos como gestión de incidentes, solicitud de servicios, administración de recursos o activos y más.

En Experian, la plataforma es utilizada por diversos departamentos y no es exclusiva del departamento de EITS. Esta implementada en un ambiente hibrido que incluye servidores físicos en múltiples regiones llamados Mid Servers, y un ambiente de Cloud dedicado, implementado dentro de la red de ServiceNow, al cual Experian tiene acceso a través de un web portal para las consultas y diversos usos que se le da a la plataforma.

Específicamente en la región de Norte América, se encuentran configurados 15 Mid servers dentro de la red de Experian, los cuales operan con Windows Server 2022 y las siguientes especificaciones básicas:

- CPU: 8
- RAM: 32G
- Almacenamiento: 4TB

Estos Mid Servers tienen el propósito de escanear el ambiente por servidores que tengan el agente de ServiceNow instalado para registrarlos en la base de datos de la plataforma llamada CMDB.

Esta infraestructura es soportada por otro equipo totalmente ajeno a EITS, y por motivos de confidencialidad no se pueden mostrar más detalles relacionado a la configuración de los servidores o diagramas de infraestructura. Sin embargo, en la ilustración 38 se presenta un

bosquejo básico de como se ha configurado la plataforma, de acuerdo con el arquitecto líder del equipo de soporte de ServiceNow en Experian.

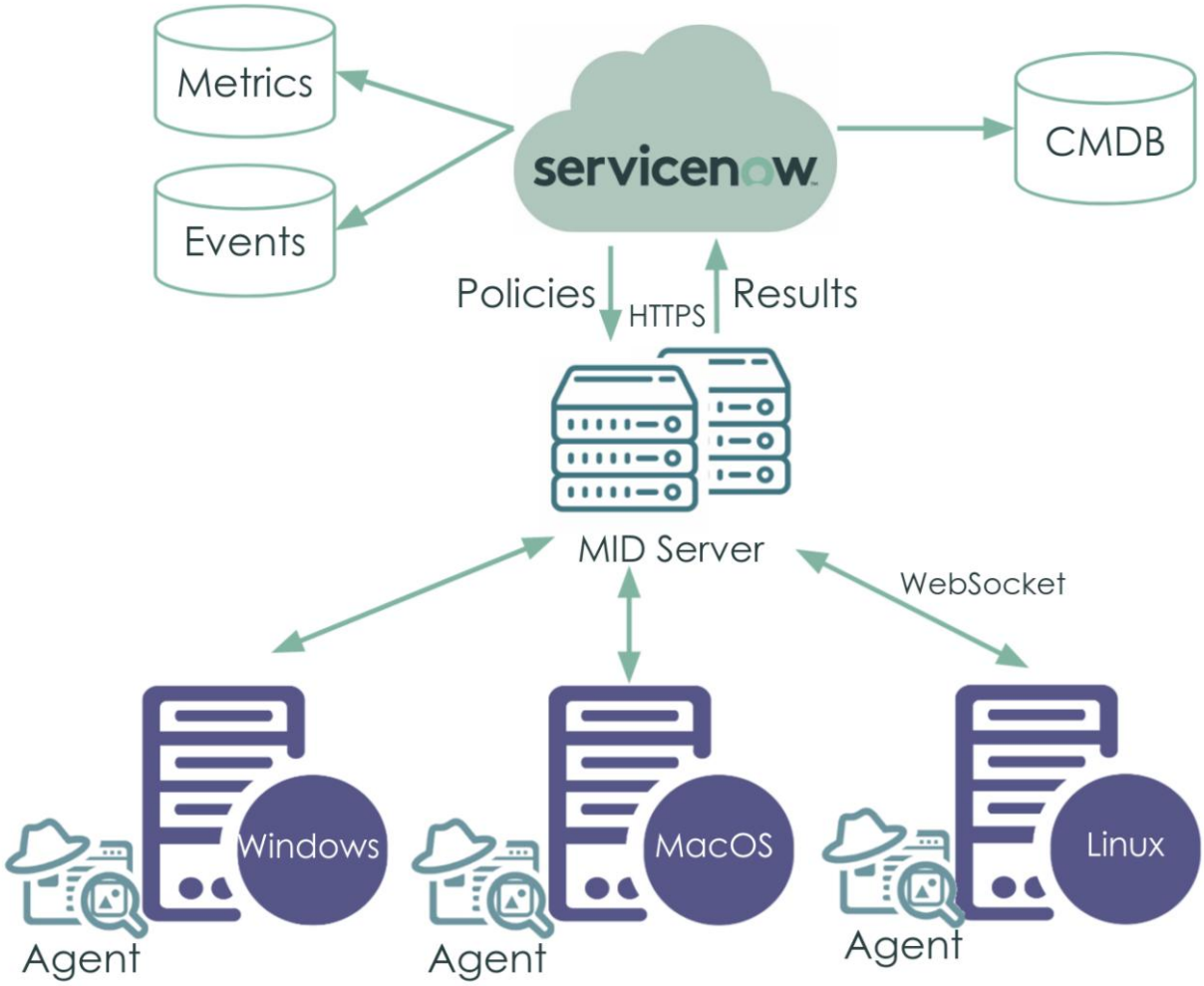


*Ilustración 38 Ejemplo de infraestructura de ServiceNow. Fuente: ServiceNow.*

### *4.2.4 Infraestructura de Red Hat Satellite*

La plataforma de Red Hat Satellite es utilizada para obtener las ultimas actualizaciones de los paquetes de sistema operativo disponibles por parte de Red Hat de manera automatizada, esto a través de la sincronización de los repositorios locales con los repositorios globales en el Content Delivery Network (CDN) de Red Hat.

Para esto la plataforma ha sido configurada en servidores virtuales almacenados en hosts VMWare ESXi de la siguiente manera

- En Datacenter 1:
    - El servidor Satellite principal y único con conexión a Internet. Se conecta al CDN de Red Hat para obtener las actualizaciones de los paquetes de manera automática.
    - Se configuran 2 servidores Capsule que sirven como bastiones del servidor principal. Replican el contenido de los paquetes y se configura en un load balancer. Los servidores de los clientes se registran a estos Capsule para obtener las actualizaciones.
- En Datacenter 2:
    - Se configuran 2 servidores Capsule que sirven como bastiones del servidor principal. Replican el contenido de los paquetes y se configura en un load balancer. Los servidores de los clientes se registran a estos Capsule para obtener las actualizaciones.

En la ilustración 39 se muestra en detalle la estructura de la plataforma. En el proceso de parcheo es de vital importancia para el contenido de las actualizaciones por lo que es necesario mencionar y detallar, sin embargo, su configuración y estructura se mantendrá sin modificaciones durante el diseño e implementación de este proyecto.

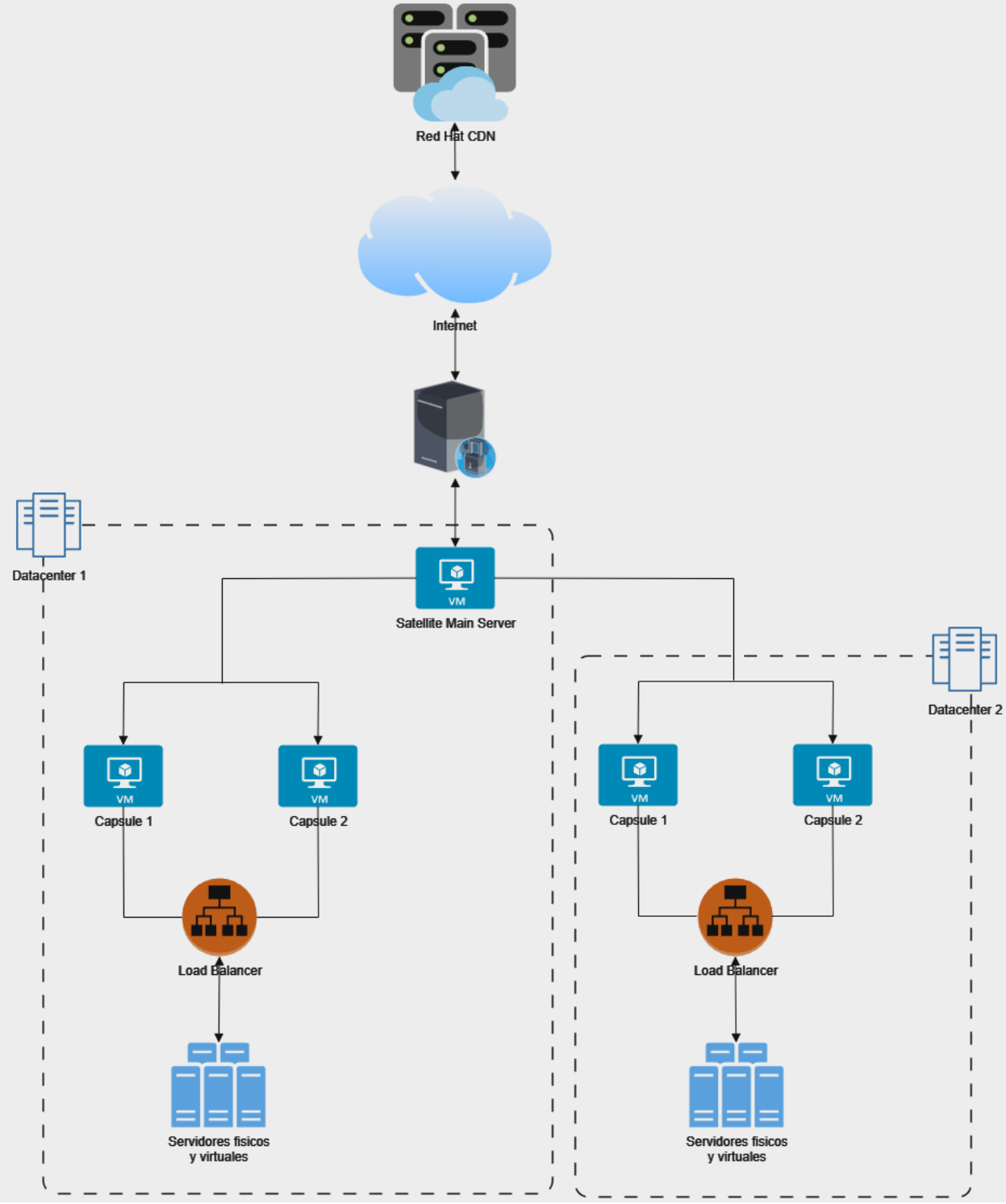


*Ilustración 39 Infraestructura Red Hat Satellite. Fuente: Elaboración propia*

### *4.2.5 Infraestructura de Cluster de Script Hosts*

La organización a la que pertenece el departamento de EITS tiene a su disposición un cluster conformado por 3 servidores virtuales almacenados en VMWare ESXi hosts diferentes, que operan con sistema operativo Linux Red Hat y las siguientes especificaciones básicas

- CPU: 4
- RAM: 8GB
- Almacenamiento: 128GB

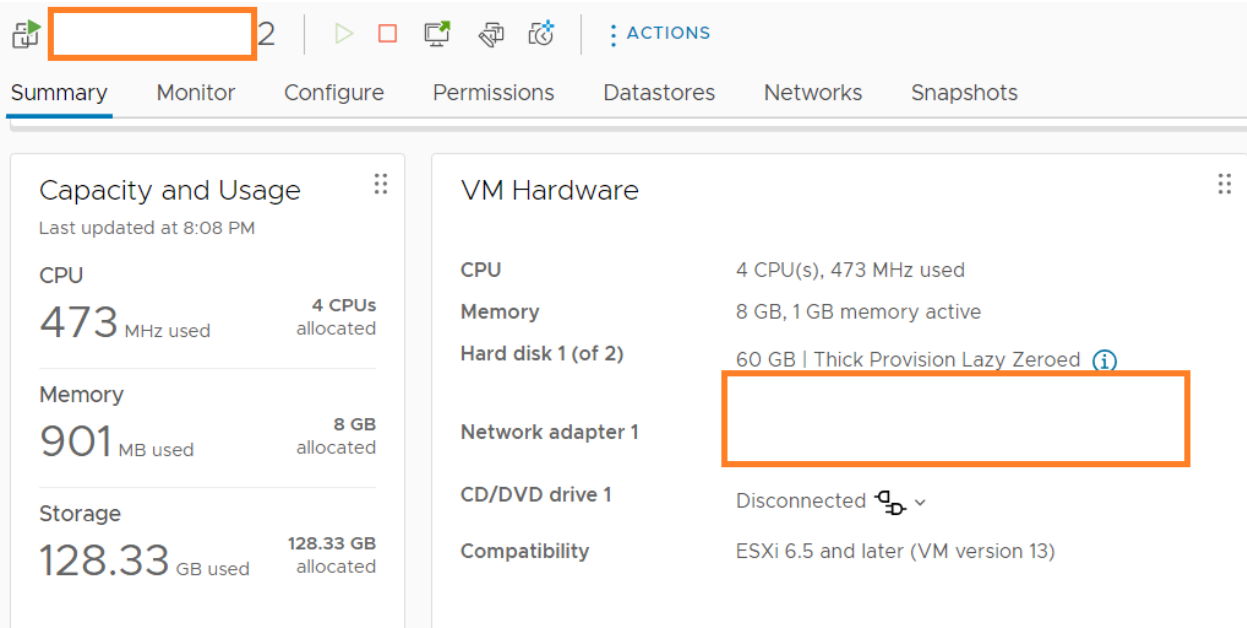


*Ilustración 40 Ejemplo de servidor del cluster de Script Hosts. Fuente: Experian.*

El propósito de estos servidores es la ejecución de scripts BASH para tareas relevantes en servidores Linux, y scripts de PowerShell para tareas relevantes en servidores Windows. La ejecución de dichos scripts se realiza a través de cronjobs calendarizados en los servers. Las tareas y scripts varían en cuanto a su propósito y soporte, pero las que son relevantes al departamento de EITS y el proceso de parcheo son

1. Script de generación de incidentes de soporte.

2. Script de envío de notificaciones a las unidades de negocio.

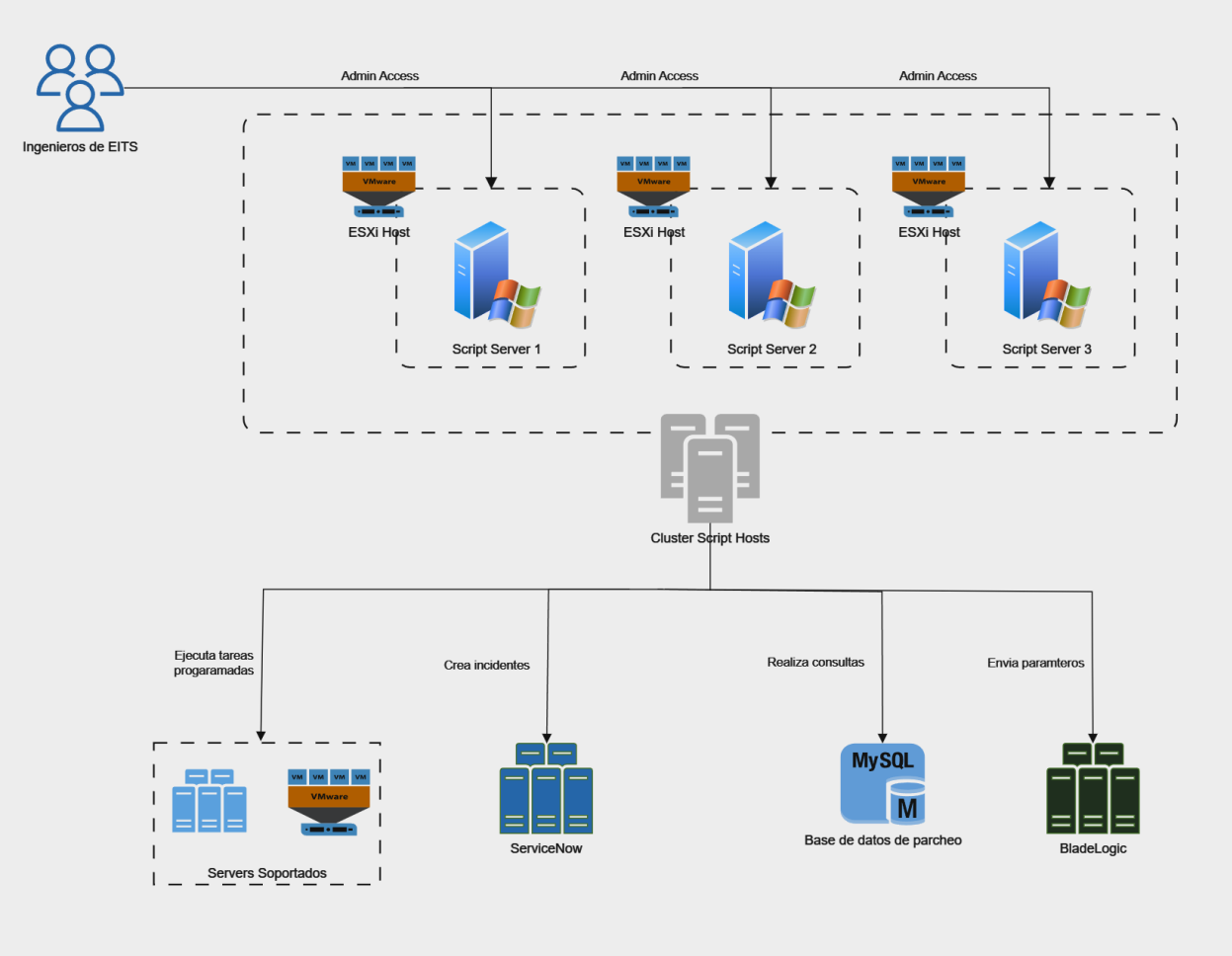


*Ilustración 41 Interacciones del cluster de Script Hosts. Fuente: Elaboración propia.*

## 4.3 Diagnóstico de percepción

Según las entrevistas y reuniones con técnicos e ingenieros del departamento de EITS, la situación que afecta al equipo es cómo el proceso actual sobrecarga al equipo por las limitaciones de la plataforma actual. La mayoría de los pasos o etapas del ciclo de parcheo requieren, en cierto nivel, la participación manual o presencial del equipo de soporte para hacer cambios o solicitudes que incluso los dueños de los servers podrían realizar.

Los clientes y dueños de los servidores soportados por el departamento de EITS manifiestan que la situación del proceso actual de parcheo podría mejorarse para permitirles mayor control y acceso a la administración del parcheo de sus sistemas, así como a la información del estado y resultado del parcheo. Esto les permitiría trabajar sobre los problemas de aquellas etapas que se encargan y reducir la cantidad de incidentes por esos problemas.

La gerencia del departamento de EITS hace énfasis en la necesidad de la implementación de una plataforma más actual y que cuente con respaldo y soporte por parte del proveedor. Esto permitiría no solo la aplicación de mejores técnicas en el ciclo de parcheo, sino que también habilitaría entrenamientos y documentación a los técnicos e ingenieros, para el desarrollo de mejoras, nuevas características al sistema, y la adaptación o escalabilidad del proceso para soportar los otros sistemas operativos que administra el departamento.

## *4.*4 Brechas o conclusiones del diagnóstico

*Tabla 4 Brechas encontradas*

| Situación actual | Brecha encontrada | Situación deseada |
|---|---|---|
| El proceso de parcheo de servidores inicia agregando el servidor a una base de datos local básica. Los técnicos y clientes acceden a través de una página web con links específicos. | Los usuarios de diferentes grupos pueden acceder y modificar información de otros grupos, representando un riesgo de seguridad. | Implementar controles de acceso más estrictos para que solo los usuarios autorizados puedan realizar cambios en sus propios servidores. |
| La plataforma TrueSight Blade Logic realiza escaneos constantes y genera archivos con detalles para evitar consultas a la base de datos completa. | Los cambios en el parcheo se deben solicitar 24 horas antes; si no, los técnicos deben modificar manualmente la programación, susceptible a errores humanos. | Automatizar el proceso de modificaciones de parcheo, permitiendo a los clientes hacer cambios sin intervención manual del equipo de soporte. |

| | | |
|---|---|---|
| La información de los procesos de parcheo se almacena localmente en cada servidor. Los ingenieros deben ingresar a cada servidor individualmente para revisar y resolver incidentes. | El acceso manual a múltiples servidores para resolver problemas es ineficiente y consume hasta el 90% del tiempo de los ingenieros. | Centralizar el almacenamiento de logs y resultados del parcheo para acceso más eficiente y rápido por parte de los ingenieros. |
| La plataforma TrueSight Blade Logic está configurada con 6 servidores de ejecución en load balancer, pero no tiene soporte ni actualizaciones del proveedor. | Ausencia de soporte y actualizaciones del proveedor y uso de un lenguaje de programación (NSH) con conocimiento limitado en la empresa. | Migrar a una plataforma de parcheo actual con soporte del proveedor y mayor disponibilidad de documentación y capacitación. |
| La comunicación entre TrueSight Blade Logic y los servidores es a través de un agente instalado en cada servidor. | Si el agente tiene problemas o no está instalado, la plataforma no puede realizar procesos de parcheo. | Mejorar la instalación y monitoreo del agente en todos los servidores, asegurando su correcto funcionamiento y conexión. |
| La plataforma ServiceNow genera y asigna tiquetes de soporte, pero la reasignación manual consume tiempo y es ineficiente. | Los incidentes relacionados con scripts no soportados deben ser reasignados manualmente, lo cual consume tiempo y puede dejar problemas sin resolver durante mucho tiempo. | Automatizar la reasignación de tiquetes a los responsables correctos y asegurar que todos los servidores tengan información de contacto actualizada. |
| El equipo de EITS está sobrecargado debido a las limitaciones de la plataforma actual y la necesidad de intervención manual en muchas etapas del proceso de parcheo. | Sobrecarga del equipo de trabajo y participación manual excesiva en el proceso de parcheo. | Implementar una plataforma moderna que automatice más etapas del proceso de parcheo, reduciendo la intervención manual y la carga de trabajo del equipo de soporte. |
| Los clientes desean mayor control y acceso a la administración del parcheo de sus sistemas y a la | Falta de control y acceso adecuado para los clientes sobre la administración y resultados del parcheo. | Proveer a los clientes herramientas y acceso necesarios para que puedan gestionar el parcheo de sus |

| | | |
|---|---|---|
| información del estado y resultado del parcheo. | | sistemas y consultar el estado y resultados de manera autónoma. |
| La gerencia del departamento de EITS enfatiza la necesidad de una plataforma con respaldo y soporte del proveedor, lo cual permitiría entrenamientos y documentación para técnicos e ingenieros. | Necesidad de una plataforma soportada por el proveedor para aplicar mejores técnicas, entrenamientos y documentación, y para la escalabilidad y adaptación del proceso. | Implementar una plataforma de parcheo con respaldo y soporte del proveedor, incluyendo entrenamientos y documentación adecuada para los técnicos e ingenieros, y que permita la escalabilidad y adaptación del proceso para otros sistemas operativos. |

*Fuente: Elaboración propia.*

# Capítulo V: Propuesta de Proyecto.

En este capítulo se presenta las propuestas planteadas para solventar las brechas identificadas en el capítulo anterior. Los detalles de la propuesta han sido alineados con los objetivos específicos definidos al inicio de este documento.

Se detallarán la estructura, fundamentos y elementos críticos que darán base a esta propuesta iniciando por el aspecto operativo, que hace referencia a la mejora en los procesos del ciclo de parcheo, el aspecto técnico, que detalla el análisis de las plataformas propuestas para la solución, y finalmente la presentación de la propuesta de mejora al proceso completo.

## 5.1 Aspecto Operativo.

A continuación, se explican las mejoras propuestas a los distintos procesos que conforman el ciclo de parcheo para los servidores Linux Red Hat soportados por el departamento EITS en la empresa Experian.

### *5.1.1 Administración de los registros de parcheo de los servidores.*

La empresa Experian hace uso a nivel global de la plataforma ServiceNow que, entre otros servicios, permite la administración y registro de todos los servidores que son parte de la infraestructura a través de la base de datos almacenada en la nube de ServiceNow llamada Configuration Management Data Base o CMDB por sus siglas en inglés (ver ilustración38). Adicionalmente, en esta base de datos se registra información de otros aspectos que son de relevancia para los registros de los servidores como, Unidades de Negocio, grupos de soporte, usuarios, ambientes, incidentes, solicitudes de cambio, solicitudes de soporte, entre otros.

La plataforma ofrece la funcionalidad de crear y relacionar nuevas tablas en la base de datos, editar las que ya están creadas, y realizar cierto nivel de automatización o validaciones según se requieran. Con base en estas funcionalidades, se propone dejar de utilizar la base de

datos de parcheo en el cluster de MySQL a cambio de utilizar la base de datos de CMDB para el manejo del registro de parcheo de los servidores tomando en cuenta los siguientes puntos:

- Agregar un nuevo campo o elemento a la tabla existente en CMDB para el registro de servidores, llamado “Sistema de Parcheo” con las siguientes características

*Tabla 5 Campo Sistema de Parcheo.*

| Registro Servidores | | | | | | |
|---|---|---|---|---|---|---|
| **Campo** | **Descripción** | **Tipo** | **Editable** | **Valores** | **Valor Defecto** | **Permite Nulos** |
| Sistema de Parcheo | Representa si el servidor será parcheado manualmente o por el proceso automático. | Lista | Si | - Automático<br>- Manual | Nulo | Si |

*Fuente: Elaboración propia.*

- Crear una nueva tabla en la base de datos CMDB llamada “Tabla de Parcheo” con los siguientes campos

*Tabla 6 Tabla de Parcheo.*

| Tabla de Parcheo | | | | | |
|---|---|---|---|---|---|
| **Campo** | **Descripción** | **Tipo** | **Editable** | **Valor Defecto** | **Permite Nulos** |
| ID | Identificación del registro generado de forma automática y aleatoria | Numérico Aleatorio | No | N/A | No |
| Nombre de Servidor | Nombre del Servidor para registrar | Texto | No | N/A | No |
| Ambiente | Ambiente al que pertenece el servidor | Texto | No | N/A | No |
| Unidad de Negocio | Unidad de Negocio a la que pertenece el servidor | Texto | No | N/A | No |
| Contacto Unidad de Negocio | Persona contacto principal de la Unidad de Negocio | Texto | No | N/A | No |
| Queue Unidad de Negocio | Queue o cola de ServiceNow a la cual asignar tiquetes | Texto | No | N/A | No |
| Equipo de Soporte | Equipo de IT que brinda soporte de parcheo al servidor. | Texto | No | N/A | No |
| Correo Equipo Soporte | Contacto principal al equipo de soporte | Texto | No | N/A | No |
| Queue Equipo Soporte | Queue o cola de ServiceNow a la cual asignar tiquetes | Texto | No | N/A | No |
| Inicio Ciclo Parcheo | Fecha en la que inicia el ciclo de parcheo. En Experian por defecto esta fecha corresponde al segundo martes de cada mes | Fecha | Si | Segundo Martes de cada mes | No |
| Fin Ciclo Parcheo | Fecha en la que finaliza el ciclo de parcheo. En Experian la fecha final corresponde a los 30 días siguientes al inicio del ciclo de parcheo | Fecha | Si | 30 días después de Inicio Ciclo Parcheo | No |

| | | | | | |
|---|---|---|---|---|---|
| Fecha parcheo primaria | Fecha en la que se realiza el atento primario de parcheo | Fecha | Si | 5 días después de la fecha actual | No |
| Fecha parcheo secundaria | Fecha en la que se realiza un intento secundario de parcheo | Fecha | Si | N/A | Si |
| Fecha parcheo adicional | Fecha en la que se desea hacer un intento adicional de parcheo | Texto | Si | N/A | Si |
| Excluir | Determina si el servidor debe excluirse del ciclo de parcheo | Booleano | Si | False | No |
| Ciclos de exclusión | Cantidad de ciclos de parcheo en los cuales excluir el servidor | Numérico | Si | 0 | No |
| Detalle parcheo | Detalle del resultado de parcheo | Texto | Si | N/A | Si |
| Etapa parcheo | Etapa de parcheo en la que se encuentra el servidor | Texto | Si | PreParcheo | No |
| Estado parcheo | Estado del proceso de parcheo | Texto | Si | Listo | No |
| Ventana mantenimiento | Horas durante las que el servidor estará en mantenimiento por parcheo | Numérico | Si | 4 | No |

*Fuente: Elaboración propia.*

- Los valores en los campos no editables de la Tabla 6 se obtendrán de las tablas ya existentes en CMDB, a través de las relaciones creadas entre dichas tablas y la “Tabla de Parcheo”.
- Habilitar en ServiceNow la automatización para agregar un registro nuevo a “Tabla de Parcheo” siempre que al registro del servidor en CMDB se le modifique el campo “Sistema de Parcheo” a “Automático”.
- Habilitar en ServiceNow la automatización para eliminar un registro de “Tabla de Parcheo” siempre que al registro del servidor en CMDB se le modifique el campo “Sistema de Parcheo” de “Automático” a “Manual ” o a un valor nulo/vacío.

- Habilitar en la vista del registro de servidor (ver ilustración 42) en ServiceNow, el campo de “Sistema de Parcheo” para que pueda ser modificado manualmente.

- Habilitar en la vista del registro de servidor en ServiceNow la opción para acceder a la vista del registro del mismo servidor en la tabla de parcheo para visualizar los elementos no editables, así como visualizar y editar los elementos editables del registro.

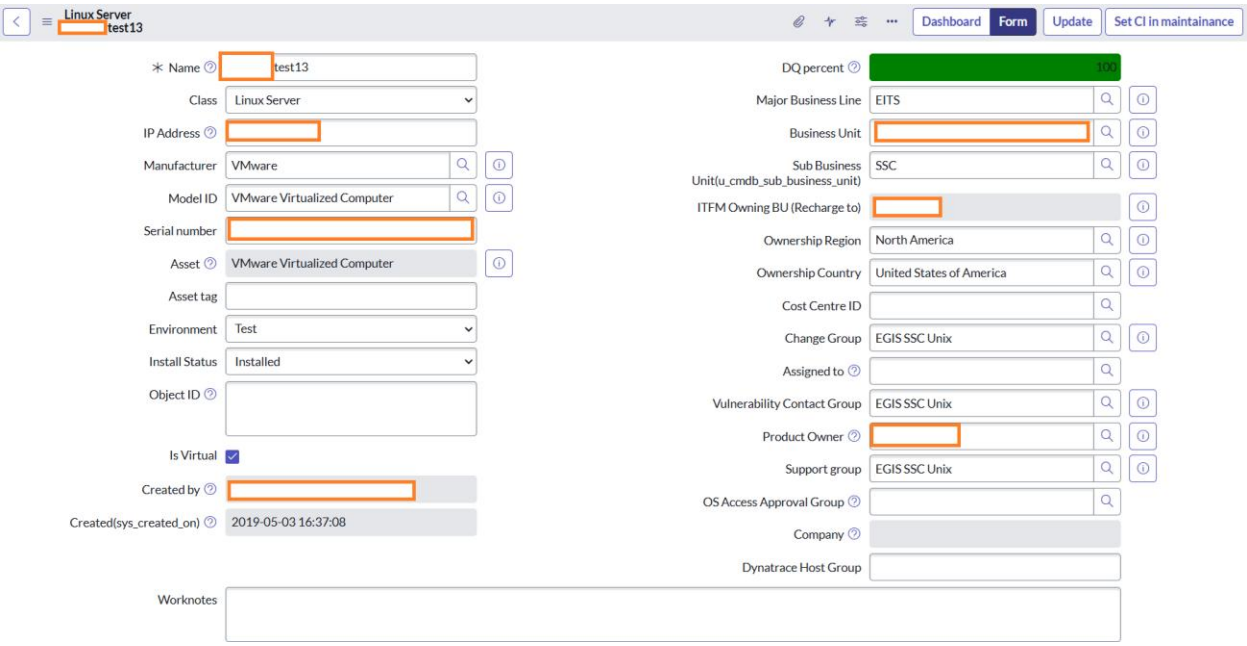


*Ilustración 42 Registro de Servidor en CMDB. Fuente: Experian.*

En la ilustración 43 se pueden visualizar como serían las relaciones entre la propuesta Tabla de Parcheo y las tablas de las que obtiene la información requerida según los puntos anteriormente descritos para la administración de los registros de parcheo. De la misma forma que el diagrama entidad-relación anterior, esta es una representación de las tablas y campos de interés de la propuesta.

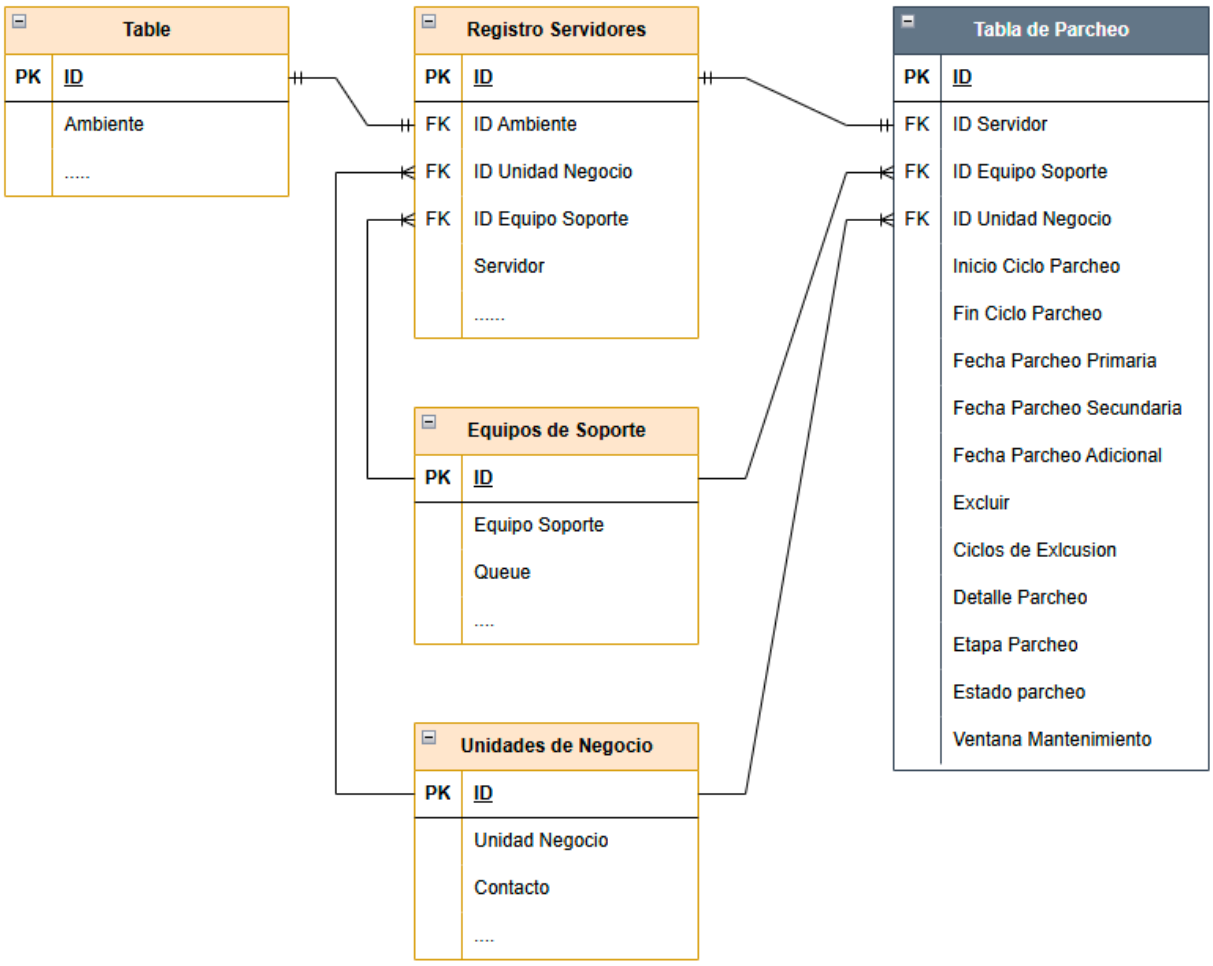


*Ilustración 43 Diagrama entidad-relación de la propuesta. Fuente: Elaboración propia.*

Las relaciones entre las tablas descritas, y la habilitación de los campos y vistas requeridas para esta propuesta, permitirían mejorar los siguientes procesos del ciclo de parcheo:

**Control de acceso.** La relación con las tablas de Unidad de Negocio y Equipos de Soporte, permite que el acceso al registro del servidor en Tabla de Parcheo pueda restringirse mediante validaciones y condiciones en ServiceNow, asegurando que solo los miembros de los respectivos equipos de Unidad de Negocio (clientes dueños del servidor) y de Equipo de Soporte (por defecto será el departamento de EITS) puedan realizar modificaciones, evitando que personas no relacionadas al equipo de ingenieros de EITS o al negocio realicen cambios

en las fechas o estado de parcheo. La ilustración 44 muestra un ejemplo de los campos que determinarían el grupo con acceso al servidor en el tab Server Ownership.

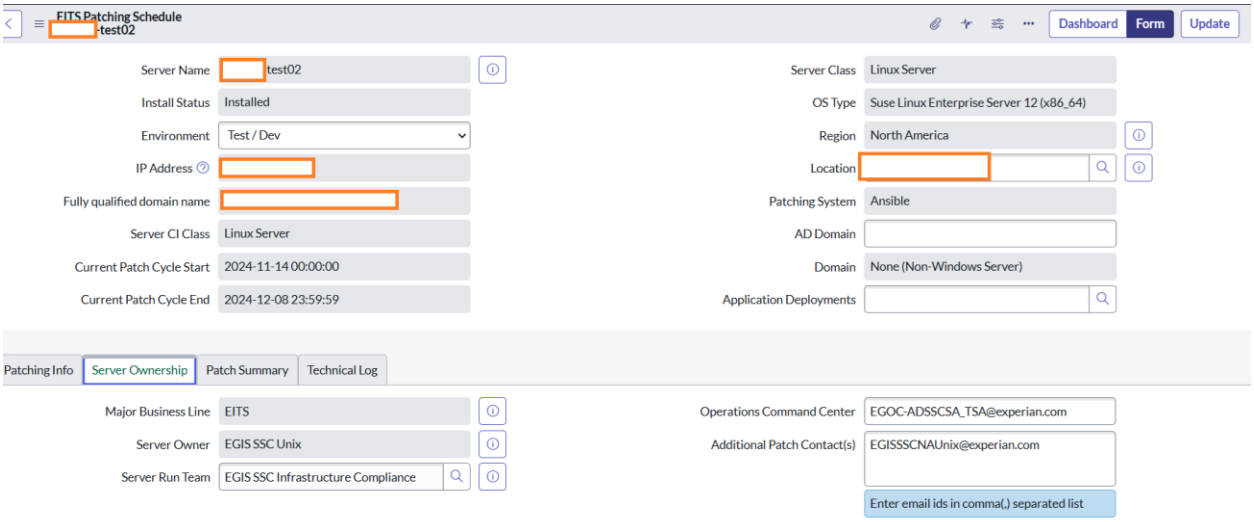


*Ilustración 44 Ejemplo de Ownership del servidor. Fuente: Experian.*

**Agregar un servidor al ciclo de parcheo**. Con el acceso controlado a través de la plataforma de ServiceNow, se puede delegar la responsabilidad de agregar o remover servidores existentes en el ciclo de parcheo a los clientes dueños de los servidores. Esto se puede realizar de manera efectiva mediante la creación de la documentación y entrenamientos que indiquen el paso a paso para los clientes en como modificar el registro de sus servidores. Por otro lado, para nuevos servidores, la responsabilidad estará asignada a los ingenieros del departamento de EITS que, como parte del proceso de creación de servidores, ya realizan modificaciones al Registro de Servidores en ServiceNow, por lo que sería cuestión de actualizar el proceso para asegurar que el campo “Sistema de Parcheo” este configurado por defecto en “Automático”.

**Calendarización de un servidor**. Al igual que el proceso para agregar servidores al ciclo de parcheo, la responsabilidad del proceso de calendarización puede ser delegada a los clientes dueños de los servidores, que a su vez les daría control total sobre la administración de los tiempos de parcheo de los servidores que manejan sus aplicaciones. Para servidores

existentes, con el acceso a la vista de los registros de la Tabla de Parcheo de sus servidores pueden calendarizar los intentos de parcheo como consideren necesario, así como excluirlos cuando es requerido. En el caso de nuevos servidores, cuando estos son solicitados los clientes pueden brindar los detalles de las fechas deseadas de parcheo, para que sean registrados de esta forma por los ingenieros de EITS; o pueden también modificarlos una vez los ingenieros de EITS han confirmado que están configurados y listos en la infraestructura y el registro en Tabla de Parcheo preparado para modificación. Al remover la dependencia de un único ingeniero que puede agregar registros, el proceso para completar la entrega de nuevos servidores calendarizados en el ciclo de parcheo se reduce de horas o días a minutos. ServiceNow implementara validaciones en los campos de fechas de parcheo para que estas no puedan ser definidas en el pasado, o en fechas fuera del ciclo de parcheo actual. En la ilustración 45 se muestra un ejemplo de la sección habilitada para calendarizar el servidor en el tab Pathing Info.

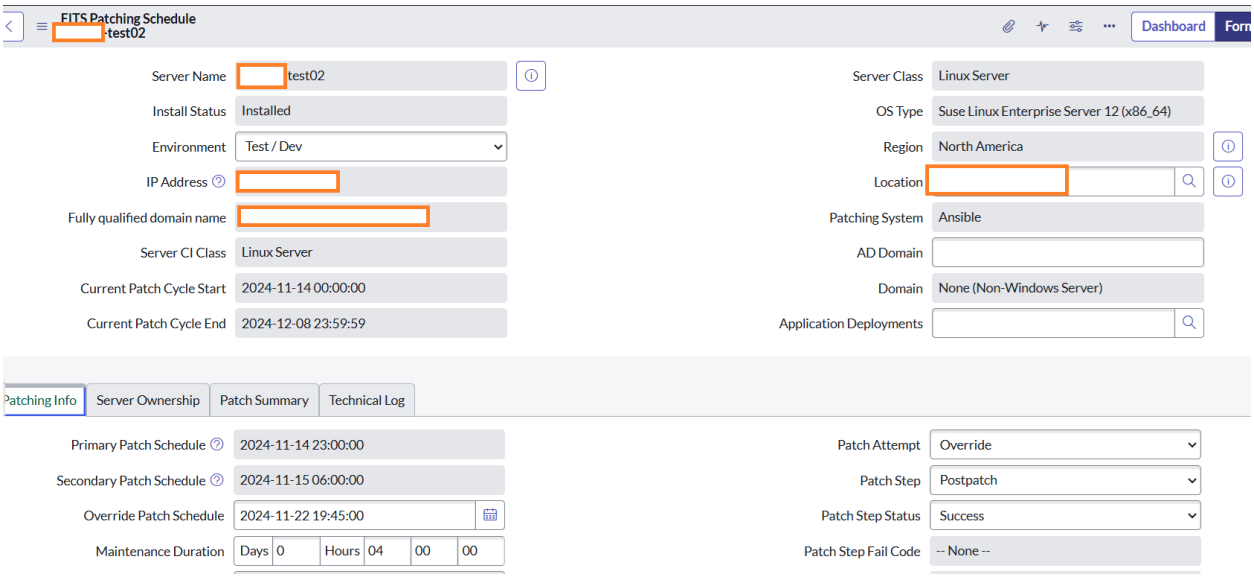


*Ilustración 45 Ejemplo de calendarización del servidor. Fuente: Experian.*

**Creación y ejecución de las tareas de parcheo**. Basado en el proceso actual, la creación y ejecución de las tareas de pre-parcheo y parcheo se ejecutarían en una plataforma de automatización, cuyos detalles son explicados a profundidad en la sección 5.2 Aspecto

Técnico, y que deberá ser capaz de tener integración con la plataforma ServiceNow para realizar consultas y cambios a la Tabla de Parcheo durante el proceso. La propuesta de utilizar CMDB e integrarla a las tareas programadas de parcheo permitirá la creación de las tareas de pre-parcheo y parcheo basadas en las fechas definidas en cada uno de los registros de la Tabla de Parcheo. Al hacer uso de estos valores, la tarea de identificación de servidores y creación de tareas programadas puede ajustarse para que se ejecute cada 15 minutos en lugar de cada hora, y que identifique los servidores que están por parchearse en las próximas 4 horas, momento en que inicia el pre-parcheo; de esta forma los usuarios pueden realizar cambios en la fecha de parcheo hasta horas antes de la fecha y hora de inicio, en contraste de las 24 horas antes que requiere el proceso actual.

La propuesta también permite un seguimiento más detallado del estado del proceso de un servidor, al incluir funcionalidades en el código de parcheo que modifique los valores en los campos:

- Detalle parcheo: Remplazaría los logs de parcheo almacenados localmente en los servidores durante el proceso. De esta forma, en la vista de los registros de la Tabla de Parcheo, los usuarios serían capaces de ver los detalles del proceso durante y al final de la ejecución. Estos mismos detalles se incluirían en el mensaje de parcheo exitoso, o la información de error que a su vez seria parte del incidente de soporte.

- Etapa parcheo: Representaría la etapa de parcheo en la que se encuentra el servidor en un momento determinado durante la ejecución del proceso. Los valores que almacena representan las etapas del proceso anteriormente mencionadas Pre-Patch, Patch y Post-Patch. La plataforma de automatización debería ser capaz, a través del código de parcheo, de actualizar el valor de este campo dependiendo de en qué etapa del proceso se encuentre.

- Estado parcheo: Representaría el estado de la etapa de parcheo en la que se encuentra el servidor en un momento determinado durante la ejecución del proceso. Los valores que almacena serian Listo, Ejecución, Exitoso y Fallido. La plataforma de automatización debería ser capaz, a través del código de parcheo, de actualizar el valor de este campo dependiendo de los detalles en la Tabla 7.
- En la ilustración 45 se pueden ver ejemplificados las etapas y estados de parcheo para un servidor.

*Tabla 7 Relación entre Etapa Parcheo y Estado Parcheo.*

| Relación entre Etapa Parcheo y Estado Parcheo | | | |
|---|---|---|---|
| **Servidor** | **Etapa Parcheo** | **Estado Parcheo** | **Descripción** |
| Servidor01 | Pre-Patch | Listo | La etapa Pre-Patch está lista para ejecución en la hora y fecha definidos |
| | | Ejecución | La etapa Pre-patch está en ejecución |
| | | Exitoso | La etapa Pre-Patch finalizo exitosamente |
| | | Fallido | La etapa Pre-Patch finalizo con un problema |
| | Patch | Listo | La etapa Patch está lista para ejecución en la hora y fecha definidos |
| | | Ejecución | La etapa Patch está en ejecución |
| | | Exitoso | La etapa Patch finalizo exitosamente |
| | | Fallido | La etapa Patch finalizo con un problema |
| | Post-Patch | Listo | La etapa Post-Patch está lista para ejecución en la hora y fecha definidos |
| | | Ejecución | La etapa Post-Patch está en ejecución |
| | | Exitoso | La etapa Post-Patch finalizo exitosamente |
| | | Fallido | La etapa Post-Patch finalizo con un problema |

*Fuente: Elaboración propia.*

### *5.1.2 Administración de los incidentes de soporte.*

Los incidentes de soporte son administrados por la plataforma de ServiceNow a nivel Global para la empresa Experian, abarcando el proceso desde su creación hasta la resolución de los mismos. El proceso de parcheo actual esta complementado con la generación de incidentes de manera automática, sin embargo, no está hecho de manera eficiente. El API de la plataforma de ServiceNow permite no solamente la creación de incidentes, sino también la personalización del contenido, asignación, correlación con otros elementos o registros dentro de la base de datos de CMDB y demás funciones.

De acuerdo con estas opciones que brinda la automatización de ServiceNow, se propone mejorar los procesos de administración de incidentes de soporte, identificación de patrones en los problemas de parcheo y tiempos de resolución mediante los siguientes puntos:

- Agregar un nuevo campo o elemento a la tabla existente en CMDB para el registro de incidentes, llamado “ID Parcheo” con las siguientes características

*Tabla 8 Campo ID Parcheo en Registro de Incidentes.*

| Registro Incidentes | | | | | |
|---|---|---|---|---|---|
| **Campo** | **Descripción** | **Tipo** | **Editable** | **Valor Defecto** | **Permite Nulos** |
| ID Parcheo | Relaciona la tabla de incidentes con Tabla de Parcheo mediante el ID de cada registro. | Numérico | No | Automático | Si |

*Fuente: Elaboración propia.*

- Crear en ServiceNow la automatización que permita, por cada incidente de parcheo generado, almacenar en el nuevo campo el respectivo ID del registro en la Tabla de parcheo. Esta relación permitirá crear una vista en ServiceNow que muestre el histórico de todos los Incidentes de parcheo generados para un mismo servidor registrado en la tabla de parcheo. El histórico de incidentes puede proporcionar información acerca de problemas recurrentes o patrones similar para identificar problemas mayores con el servidor o el proceso. En la ilustración 44 se muestra un ejemplo desarrollado en ServiceNow de cómo se mostraría el historial de incidentes para un solo servidor.

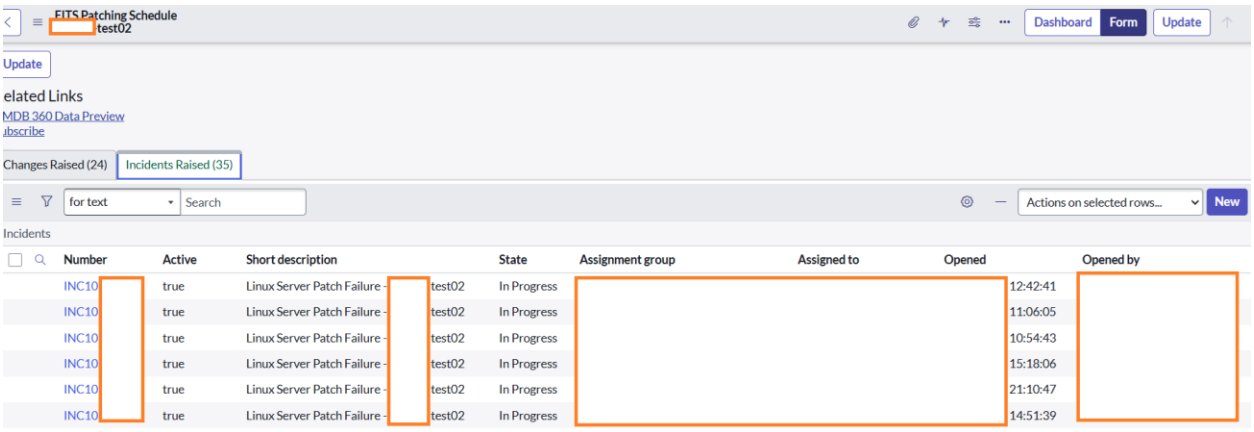


*Ilustración 46 Ejemplo de vista de Incidentes. Fuente: Experian.*

- Optimizar el código de parcheo para que además de guardar la información de los resultados del proceso de parcheo en “Tabla de Parcheo”, les de formato y almacene también en el campo de “Descripción” en la tabla de Registro de Incidentes, cuando se cree un incidente de parcheo.

- Optimizar el código de parcheo para que identifique la etapa y tarea en la que se genere el incidente. Si el problema surge durante la ejecución de los PrePatch o PostPatch scripts soportados por las Unidades de Negocio, al generar el tiquete, en el campo de “Grupo Asignado” debe registrarse el valor respectivo al queue de la Unidad de Soporte del servidor de acuerdo con la relación con Tabla de Parcheo. Si el problema surge durante cualquier otra etapa del proceso, al generar el tiquete, en el campo de “Grupo Asignado” debe registrarse el valor respectivo al queue del equipo de soporte del servidor (departamento de EITS) de acuerdo con la relación con Tabla de Parcheo.

- En la ilustración 47 se muestra un ejemplo de la descripción creada para un Incidente generado del proceso de parcheo.

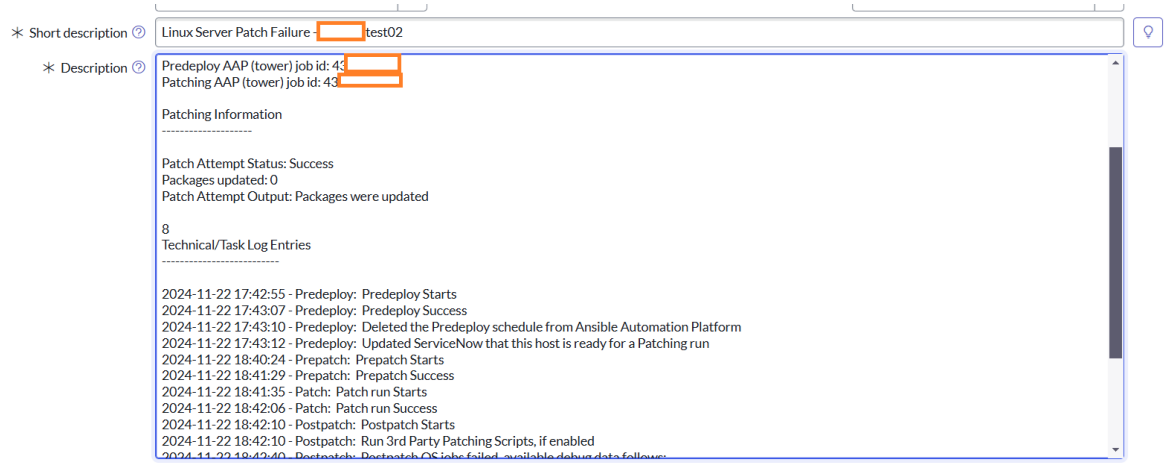


*Ilustración 47 Ejemplo de la descripción de un incidente de parcheo. Fuente: Experian.*

### *5.1.3 Administración del código de parcheo.*

En el caso específico del código de parcheo, se ha mencionado en apartados anteriores que es almacenado, administrado y ejecutado por la plataforma de BladeLogic. Sin embargo, esta no es la única opción con la que cuenta la empresa Experian para la administración de código de automatización.

Actualmente la Experian cuenta con el producto licenciado de Atlassian BitBucket, descrito en detalle en el Capítulo 2 apartado 2.1.7 BitBucket, que entre muchas de sus capacidades ofrece control de revisión durante el desarrollo de código, permitiendo así no solo la colaboración simultanea de los ingenieros del departamento de EITS con el rol de desarrolladores, sino también la aplicación de las Mejores Prácticas de desarrollo de software tales como:

- Estrategia de ramas y bifurcación (Branch Strategy)
- Revisión de Código.
- Publicaciones calendarizadas.

- Notificaciones al realizar Pull Requests o Publicaciones al código principal.
- Control de versiones.

Con base a estas características, se propone utilizar BitBucket como la plataforma para la administración del código de parcheo y utilizar una estrategia de ramas que permita crear múltiples ramas de acuerdo con el propósito del trabajo que se realizara en cada una utilizando las siguientes definiciones

1. Main: Sera la rama principal, donde se almacenará y publicará el código principal de parcheo para los ambientes de producción. A partir de esta rama se crearán solo las ramas de Hotfix y Dev.
2. Dev: Sera la rama dedicada al ambiente de desarrollo o pruebas y almacenara y publicara el código de parcheo para los ambientes de que no son producción. A partir de esta rama se crearán solo las ramas de Feature.
3. Feature: Serán las ramas que se crearán a partir de Dev para el desarrollo de mejoras al código, nuevas características o cambios de estructura. Se unirán al código en Dev solamente a través de pull requests que generarán notificaciones, serán validadas y aprobadas por el proceso de Revisión de código.
4. Hotfix: Serán las ramas que se crearán a partir de Main para el desarrollo de arreglos a problemas inmediatos del código. Se unirán al código en Main y Dev solamente a través de pull requests que generarán notificaciones, serán validadas y aprobadas por el proceso de revisión de código.
5. Release: Serán las ramas que se crearán a partir de Dev unir el código a Main después de que se hayan integrado ramas de Feature suficientes para hacerlas públicas. Se unirán al código en Main solamente a través de pull requests que

generarán notificaciones, serán validadas y aprobadas por el proceso de revisión de código.

Con estas recomendaciones podemos ver la estrategia de ramas representada en la ilustración 48.

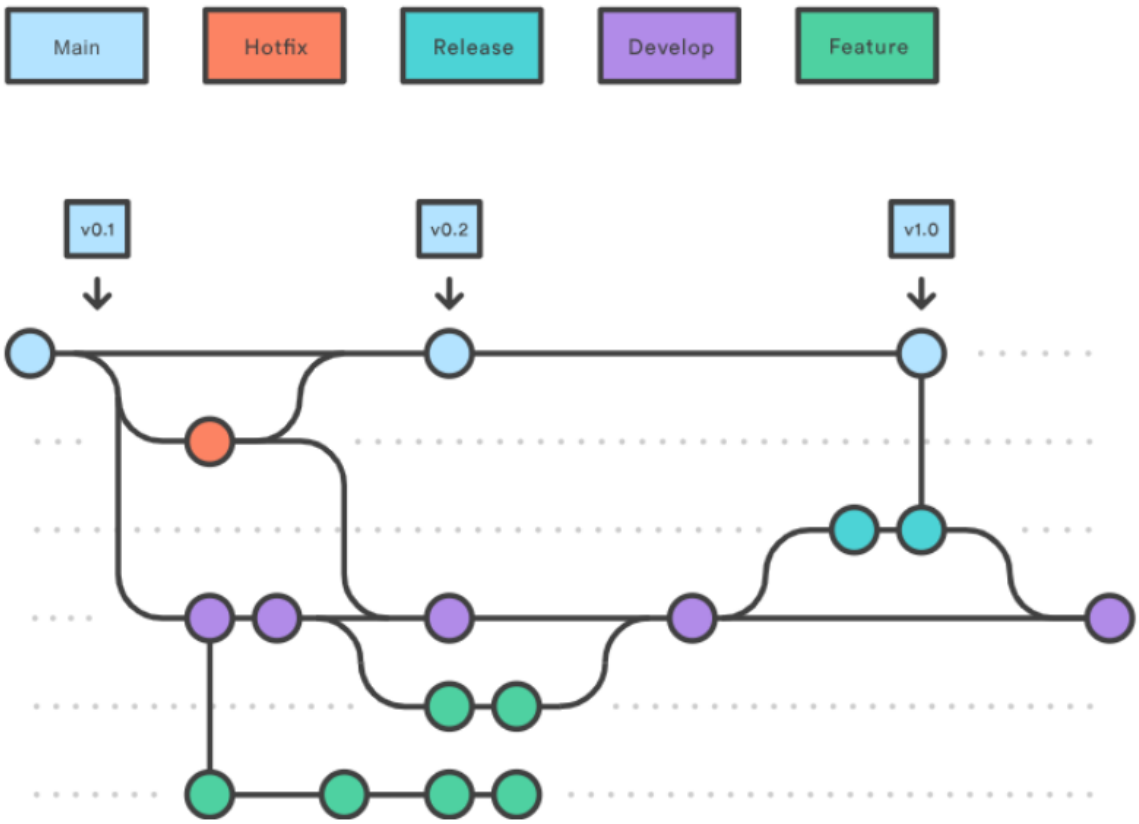


*Ilustración 48 Estrategia de Ramas. Fuente: Experian.*

En la ilustración 49 se muestra la interacción de las características antes mencionadas, que ofrece BitBucket al realizar pull requests para integrar el código a las ramas de Main o Dev.

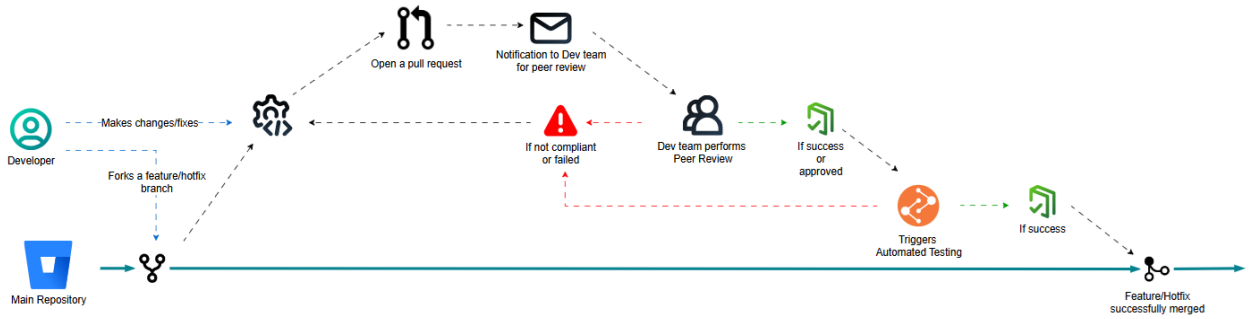


*Ilustración 49 Flujo de Pull Requests. Fuente: Elaboración Propia.*

Adicionalmente, se recomienda BitBucket porque ofrece integración con las plataformas de automatización que se plantean en la sección 5.2 Aspecto Técnico, y con otros productos de Atlassian utilizados por Experian como Jira para la administración del trabajo asignado a los ingenieros, y Confluence, utilizado para todo tipo de documentación.

El código necesitara ser interpretado del lenguaje NSH y trasladado a un lenguaje más versátil y accesible, capaz de ser utilizado por las plataformas de automatización, como lo es el Lenguaje YAML, sobre el cual la empresa Experian cuenta con cursos online para los ingenieros y existe documentación sobre el mismo creado por otros departamentos, lo cual reducirían en gran medida la curva de aprendizaje.

En la ilustración 50 se muestra un ejemplo de repositorio de BitBucket creado como ejemplo solamente para efectos de este documento.

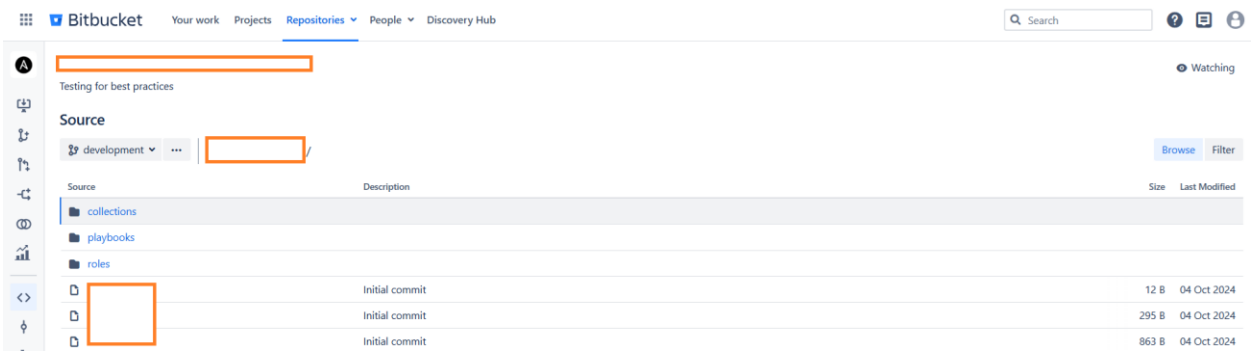


*Ilustración 50 Ejemplo Repositorio de BitBucket. Fuente: Experian.*

## 5.2 Aspecto Técnico.

A continuación, se detalla a profundidad las plataformas analizadas como posibles recomendaciones para la implementación de la propuesta de mejora en el proceso de parcheo actualmente presente para los servidores Red Hat en la empresa Experian.

### *5.2.1 Plataforma de Automatización TrueSight BladeLogic.*

#### Descripción.

La plataforma BladeLogic es la solución que actualmente utiliza Experian para el proceso de parcheo automático de servidores. El continuar con esta solución no requeriría de implementación de una nueva infraestructura debido a que la existente se puede utilizar para el mismo propósito. Pero, si se requerirá realizar una actualización de todo el software de la plataforma, renovar licencias para los servidores de la aplicación, así como para todos los servidores soportados. La plataforma requiere de un agente que este instalado y en ejecución en todos los servidores y estaría pendiente de actualización en toda la infraestructura.

Al mantener la plataforma actual, el código de parcheo se mantendría en la misma ubicación y utilizando el mismo lenguaje de programación. Este requeriría una inversión adicional de tiempo, recursos económicos y humanos por parte de Experian para las capacitaciones necesarias que se requieran por parte del proveedor para la compresión del lenguaje NSH. Además de que este sistema cuenta con las limitaciones para la implementación de un proceso correcto de desarrollo de código y mejores prácticas a través del control de revisiones.

La solución mantendría la misma relación con la plataforma de ServiceNow y la base de datos MySQL para el registro y calendarización de los servidores. A pesar de que esto podría mejorarse y migrarse todo a ServiceNow, este proceso es dependiente del resultado de las capacitaciones en NSH para el equipo de EITS pues requerirá de cambios en el código para

poder implementar dichos cambios. Por este motivo dicha mejora sería una recomendación para el futuro pues no es posible de implementar inmediatamente. De la misma forma la administración y resolución de incidentes de parcheo no presentara modificación alguna, lo cual dejara algunas de las brechas actuales sin resolver, dependiendo del éxito de capacitación y entendimiento por parte de los ingenieros para realizar mejoras al proceso en la plataforma de BladeLogic.

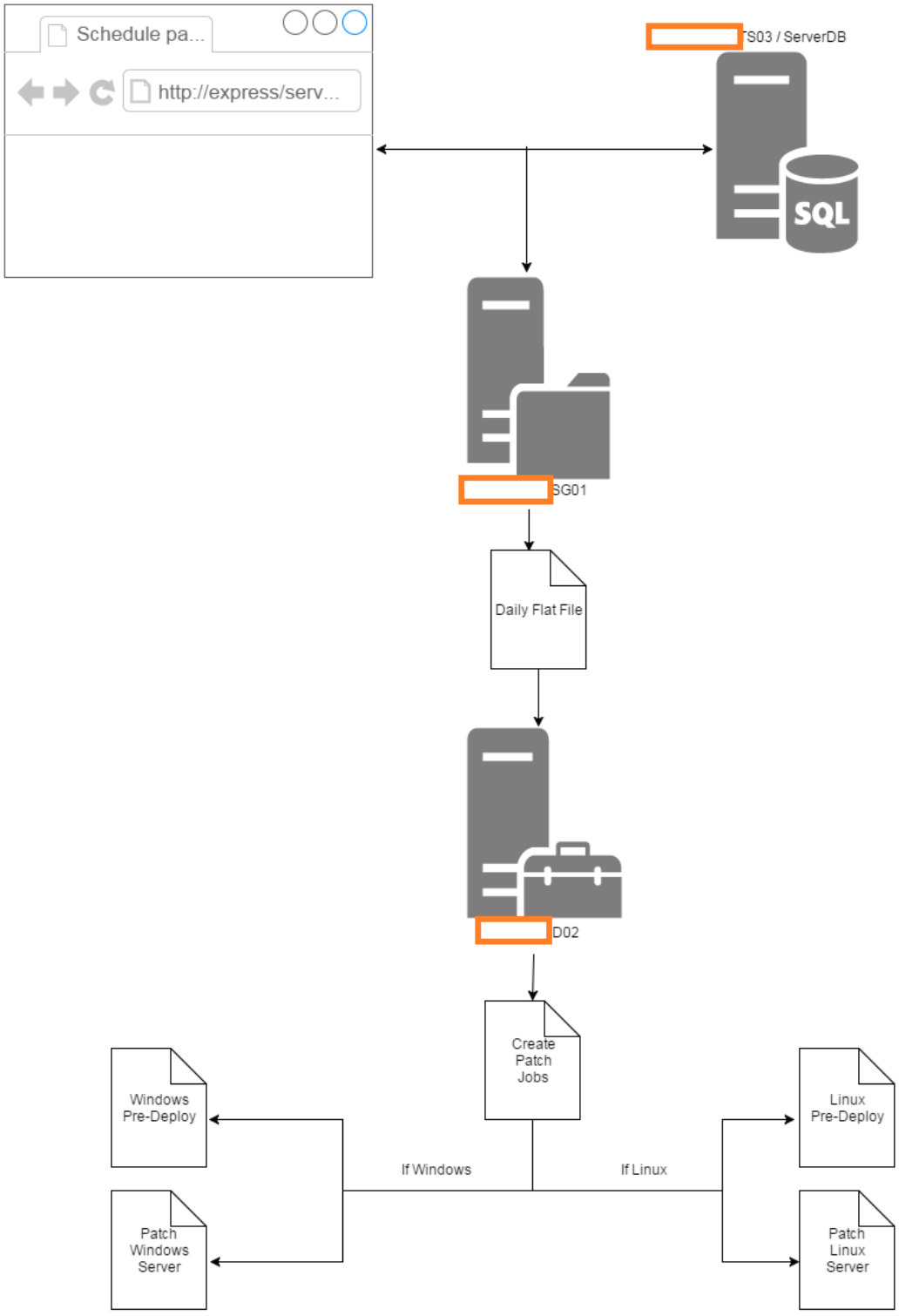


*Ilustración 51 Propuesta Plataforma BladeLogic. Fuente: Experian.*

**Costos de Licencias y Software.**

Al continuar con la plataforma BladeLogic, Experian mantendrá en un gasto mensual relacionado a las licencias de los servidores y el soporte limitado que ofrece el proveedor. De acuerdo con el departamento de EITS, las licencias tienen un costo de $80 por servidor al año, es decir $6.67 aproximadamente al mes, lo que significa que, para los 2000 servidores objetivo de este proyecto, el costo mensual es de $13 333.34 aproximadamente.

**Costos de Mantenimiento de Infraestructura.**

Los servidores dedicados a la plataforma son servidores virtuales de VMWare (detallados en el capítulo 4) cuyo costo de mantenimiento y licencia deberá mantenerse al decidir continuar con la plataforma. Sin embargo, esto reduciría al mínimo el trabajo adicional que se requeriría para habilitar una nueva plataforma.

Basados en sus características, y los acuerdos por parte de Experian con el proveedor VMWare y la información proporcionada por el departamento de EITS, los 2 servidores administrativos tendrían un costo mensual por mantenimiento y operación de $200 cada uno, para un total de $400 dólares mensuales. Mientras que para los 6 servidores de ejecución sería un costo mensual de $300 cada uno, para un total de $1800 dólares mensuales. Esto representa un gasto total mensual de $2200 para el mantenimiento de la infraestructura de la plataforma de BladeLogic en servidores virtuales de VMWare.

**Costos de Capacitación y Recurso Humano.**

Adicional a las licencias, se debe tomar en cuenta el costo del recurso humano, en este caso representado por 25 contratistas expertos en TI, que trabajan en las brechas que la solución actual presenta y definidas en el Capítulo IV. El salario promedio para estos Ingenieros de TI es cercano a los $3000 mensuales, lo que representa un costo de $75 000 al mes. Para este grupo de ingenieros se requerirá una capacitación apropiada por parte del BCM Software

para el aprendizaje y entendimiento del lenguaje NSH, que para la empresa Experian podría tener un costo aproximado de $2400, para un total de $60000.

En la Tabla 09 se presenta en detalle el costo mensual total que Experian asume con la solución actual de parcheo de los servidores Linux Red Hat y el agregado del costo único por capacitación que sería el único costo adicional.

*Tabla 9 Costo de la solución actual de parcheo.*

| Elemento | Costo Único | Costo Mensual Recurrente |
|---|---|---|
| Licencias BladeLogic | - | $13 333.34 |
| Mantenimiento | - | $2 200 |
| Capacitación | $60 000 | - |
| Recurso Humano | - | $75 000 |
| **Costo Total** | **$60 000** | **$90 533.34** |

*Fuente: Elaboración propia.*

### *5.2.2 Plataforma de Automatización Ansible.*

**Descripción.**

La implementación de la plataforma de automatización Ansible requerirá de la adquisición de licencias tanto del producto en si, como para los servidores que serán registrados en la plataforma; ya que, a diferencia de BladeLogic y Tanium, Ansible funciona a través de registro de servidores y conexión por SSH por lo que no tiene dependencia de un agente en ejecución en los servidores. Por lo que necesita de la correcta configuración de redes y autenticación de dominio para tener acceso a todos los servidores requeridos.

Una vez obtenidas las licencias, se necesita la creación de un ambiente para su administración, en este caso, en AWS Cloud que la empresa Experian posee en el que deberán de habilitarse al menos 32 instancias EC2, 1 base de datos RDS, 1 load balancer y un 1 VPC

para la conexión interna de la infraestructura. Cada uno de estos elementos tendrán un costo mensual para su mantenimiento.

Con la implementación de la plataforma Ansible, se habilita la posibilidad de integrar repositorios de Bitbucket, lo que permitiría la integración de un proceso de mejora y desarrollo de código que siga las mejores prácticas del mercado basadas principalmente en el control de revisiones, tal y como fue detallado en el apartado 5.1.3 de este capítulo. A su vez, la implementación de este proceso de desarrollo de código permitirá implementar las mejoras requeridas para una optimizada administración, asignación y resolución de los tiquetes de soporte.

Incluido como parte de gasto de adquisición de las licencias y el software, se incluyen horas de capacitación y soporte para el departamento de EITS que permitirá la comprensión y entendimiento del lenguaje YAML a ser utilizado en el código de parcheo para esta solución, y serán impartidos durante el proceso de desarrollo e implementación de esta plataforma. El desarrollo de esta nueva versión del código de parcheo permite la integración entre las aplicaciones ServiceNow, Ansible y Satellite no solo para el proceso principal de parcheo, sino también para implementar las mejoras deseadas en el reporte de estado del proceso de parcheo, identificación de problemas, registro de resultados en la base de datos o en los incidentes generados, así como la correcta asignación y resolución de estos.

En la ilustración 52 se presenta la forma en la que interactuaran la plataforma Ansible con las otras tecnologías disponibles en Experian para esta solución.

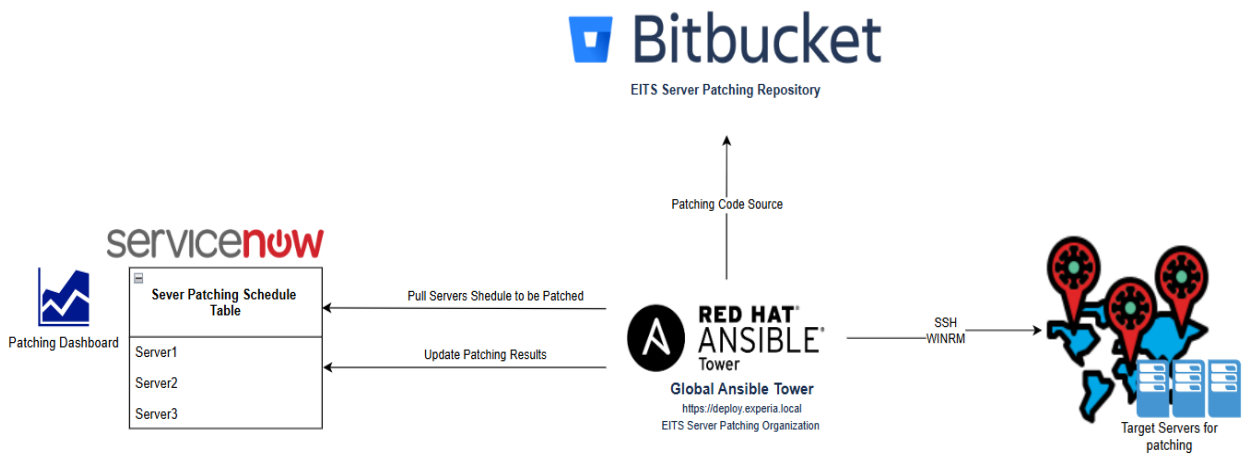


*Ilustración 52 Propuesta de Infraestructura Plataforma Ansible. Fuente: Experian.*

Por otro lado, en la imagen 53, se presenta el detalle a nivel de infraestructura de la propuesta.

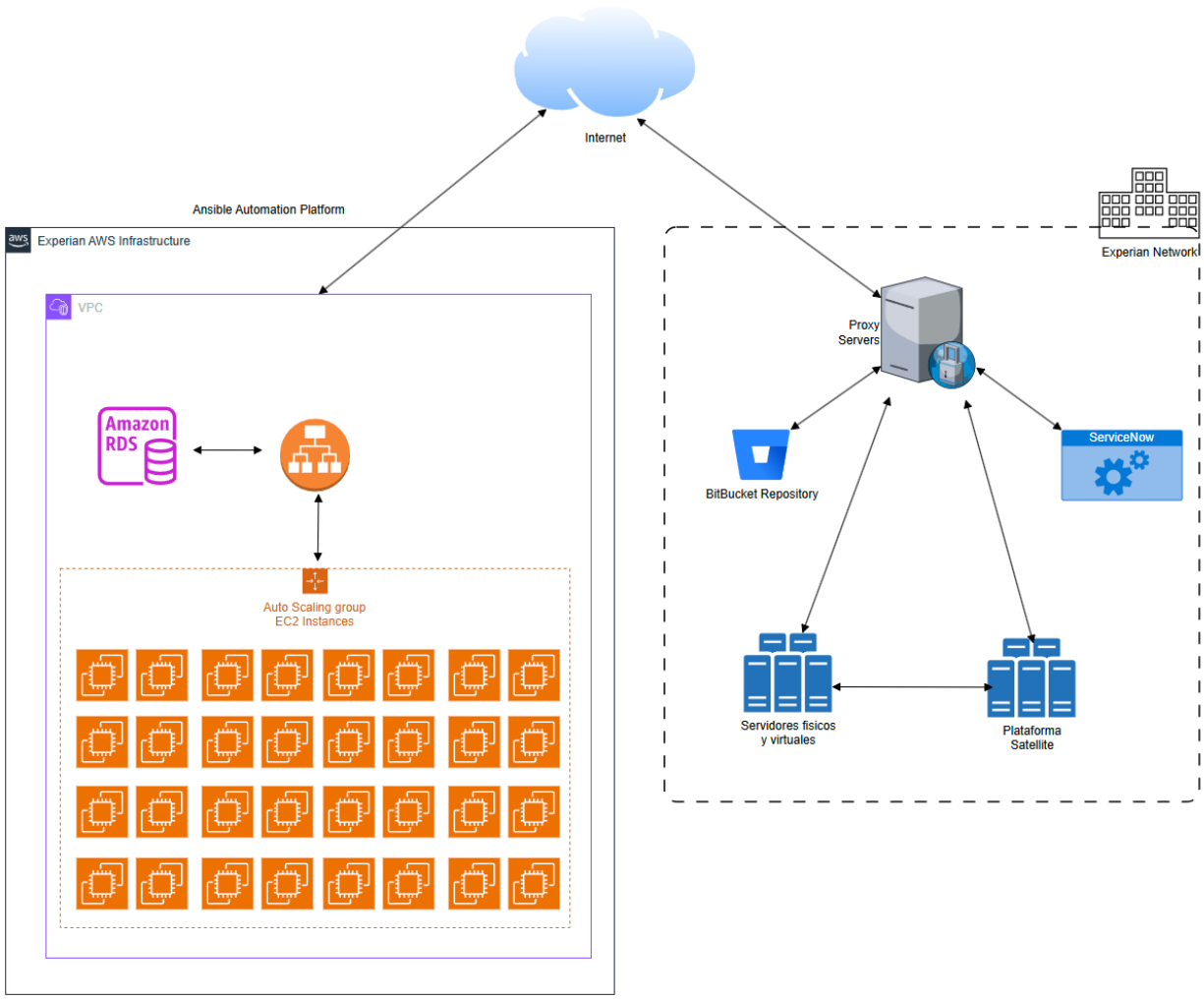


*Ilustración 53 Detalles de la infraestructura de la propuesta Ansible. Fuente: Elaboración propia.*

### Costos de Licencias y Software.

La plataforma de Ansible Automation requiere de 2 tipos de inversiones económicas para su desarrollo, siendo estas un gasto único para el desarrollo y configuración de la solución con asistencia de los ingenieros de Red Hat, y un gasto mensual que incluye el pago de las licencias requeridas para cada uno de los servidores que serán parcheados por medio de la plataforma, y el soporte técnico de la plataforma, cuyo valor puede variar dependiendo del tipo de soporte que se desee.

De acuerdo con Red Hat (2024) se ofrecen 2 tipos de soporte para la plataforma de Ansible Automation:

- Estándar**:** Provee soporte de la plataforma de 8:00am a 5:00pm, incluye actualizaciones y mantenimiento, y brinda acceso total a los recursos integrados de Ansible para diseñar, gestionar y ajustar elementos en toda la empresa.
- Premium**:** Provee soporte de la plataforma 24 horas al día, los 7 días de la semana, incluye actualizaciones y mantenimiento, y brinda acceso total a los recursos integrados de Ansible para diseñar, gestionar y ajustar elementos en toda la empresa.

Las opciones de precio basadas en el tipo de soporte pueden variar de una empresa a otra, dependiendo de las negociaciones de contrato y otros factores. En el caso de Experian el desarrollo e implementación de la solución Ansible Automation Platform tendría un costo único aproximado de $100 000. Adicionalmente se desea obtener las licencias y suscripción Premium, las cuales tendrían un costo aproximado de $10 000 anual por cada 100 licencias o servidores en la plataforma.

Tomando en cuenta que la solución de Ansible Automation Platform para el parcheo de servidores Linux Red Hat de Experian esta inicialmente planeada para administrar 2000 servidores, el costo mensual, adicional al pago único de la implementación, seria de aproximadamente $16 666.67.

**Costos de Desarrollo y Mantenimiento de Infraestructura.**

La solución de Ansible Automation Platform puede ser desarrollada para ambientes de Cloud, lo que significa que su infraestructura puede administrarse a través de soluciones en la nube como Amazon AWS, Microsoft Azure entre otros. En el caso de Experian, la empresa

tiene ambientes empresariales en Amazon AWS, por lo que se decidió plantear la solución como se recomienda.

De acuerdo con los requerimientos de infraestructura de Ansible para administrar el parcheo de aproximadamente 2000 servidores, se elabora la Tabla 10 Infraestructura AWS para Ansible Automation Platform, que muestra cuales elementos serían necesarios para una adecuada implementación:

*Tabla 10 Infraestructura AWS para Ansible Automation Platform.*

| Servicio AWS | Cantidad | Tipo | Detalle |
|---|---|---|---|
| Amazon EC2 | 32 | m5.large | Sistema Operativo Linux<br>8GB RAM, 2 vCPU<br>Instancias Spot Dedicadas |
| Amazon RDS for MySQL | 1 | db.m5.large | Almacenamiento 30GB<br>Utilización bajo demanda<br>Multi-zone habilitado (redundancia) |
| Elastic Load Balancer | 1 | Aplicación | Un load balancer de aplicación. |
| Amazon Virtual Private Cloud (VCP) | 1 | - | Ingreso de datos: 10TB al mes<br>Egreso de datos: 10TB al mes |

*Fuente: Elaboración propia.*

En base a estos elementos se puede utilizar la solución en línea AWS Pricing Claculator (2024) que provee AWS, para un cálculo aproximado del costo de la infraestructura anteriormente mencionada. En la Tabla 11 se muestra el detalle de los precios aproximados de cada uno de los elementos de la infraestructura, basado en la información de la calculadora de AWS cuyos detalles están agregados en el Anexo 7.2.

*Tabla 11 Cálculo aproximado del costo de la infraestructura en AWS.*

| Servicio AWS | Cantidad | Precio unitario mensual | Precio mensual subtotal |
|---|---|---|---|
| Amazon EC2 | 32 | $120,83 | $3866.55 |
| Amazon RDS for MySQL | 1 | $278.64 | $278.64 |
| Elastic Load Balancer | 1 | $86.51 | $86.51 |
| Amazon Virtual Private Cloud (VCP) | 1 | $1024.00 | $1024.00 |
| | | **Precio Mensual Total** | **$5255.52** |

*Fuente: Elaboración propia.*

**Costos de Capacitación y Recurso Humano.**

La implementación de una nueva solución requeriría de la capacitación de los ingenieros del departamento de EITS, y la misma se encuentra incluida como parte del gasto único de $100 000 para la implementación y configuración de la plataforma de Ansible Automation. Hay que considerar que, tras implementarse los ingenieros de EITS, deberán seguir trabajando en los incidentes que la nueva solución reportará como parte del proceso estándar de parcheo y relacionado con configuraciones, recursos o mejoras necesarias del ambiente.

Se estima que la cantidad de incidentes que se reportan al mes se reducirán de un 85% a un 90% una vez implementada la nueva propuesta, en comparación a lo que se genera actualmente. Para el 15 % de incidentes restantes, el departamento de EITS plantea que pueden abordarse con recursos dedicados, cubiertos por 12 ingenieros de TI con un salario mensual de unos $3 000. Esto generaría un costo adicional a tomar en cuenta de $36 000 mensuales.

*Tabla 12 Costo total de la implementación la plataforma Ansible.*

| Elemento | Costo Único | Costo Mensual Recurrente |
|---|---|---|
| Implementación de Ansible Automation Platform | $100 000 | |
| Licencias Ansible Automation Platform | - | $16 666.67 |
| Infraestructura | - | $5 255.52 |
| Recurso Humano | - | $36 000 |
| **Costo Total** | **$100 000** | **$57 922,19** |

*Fuente: Elaboración propia.*

### *5.2.3 Plataforma de Automatización Tanium.*

**Descripción.**

La empresa Experian cuenta actualmente con la implementación de la plataforma de Tanium, la cual es utilizada para procesos de monitoreo de servidores y aplicaciones, así como para la recaudación de información de los distintos sistemas que monitorea.

En el caso del proceso de automatización de parcheo, Tanium requiere de la adquisición de un módulo especifico de parcheo, su implementación, instalación y configuración para la plataforma anteriormente mencionada, esto proceso genera un gasto adicional representado en una licencia nueva que Experian debe costear de manera mensual, por cada servidor parcheado en la plataforma, esto adicional a la licencia que también se debe pagar solo por el servicio básico.

Además del licenciamiento, debe configurarse y expandirse el ambiente de servidores diseñado en máquinas virtuales de VMWare con la siguiente configuración:

- 2 servidores, para el manejo de los módulos y los cuales deberán estar configurados en redundancia en el datacenter 1.

- 8 servidores, para funcionar como administradores y habilitar la comunicación con los servidores soportados en cada datacenter, para un total de 16 servidores.
- 2 servidores de zona que se utilizan para interconectar los datacenters, así como para permitir la conexión a la red externa de Tanium.
- Cluster de 2 base de datos SQL redundantes en cada datacenter para el almacenamiento de la información obtenida a través del monitore que ofrece la aplicación.
- La infraestructura requiere de un servidor consola para permitir el acceso de los administradores, en este caso lo ingenieros del departamento de EITS, pero al ya existir un sistema de consola, este solo requiere de proveer los permisos y accesos necesarios a los ingenieros.

Con respecto a las 2 plataformas anteriormente presentadas, Tanium presenta 2 características importantes:

1. Es similar a la plataforma BladeLogic, en el hecho de que requiere de la instalación de un agente en todos los servidores que soporta, monitorea o parchea para habilitar la comunicación Peer to Peer. Lo que se convierte en una dependencia para esta propuesta
2. A diferencia de Ansible, Tanium no es un orquestador de automatización. Esto quiere decir que Tanium permite la automatización y simplificación de tareas, es decir se pueden automatizar los comandos o scripts de parcheo. Pero toda la lógica de registro de servidores, calendarización, creación de templates, determinación de cuándo y cómo se ejecutan dichas tareas, identificación de errores, formateo de logs, creación de incidentes y demás, deberán ser todas

transferidas a la plataforma de ServiceNow. Esto requerirá de un trabajo adicional en colaboración con el equipo de soporte de ServiceNow y tomará mayor tiempo en el desarrollo e implementación de la propuesta.

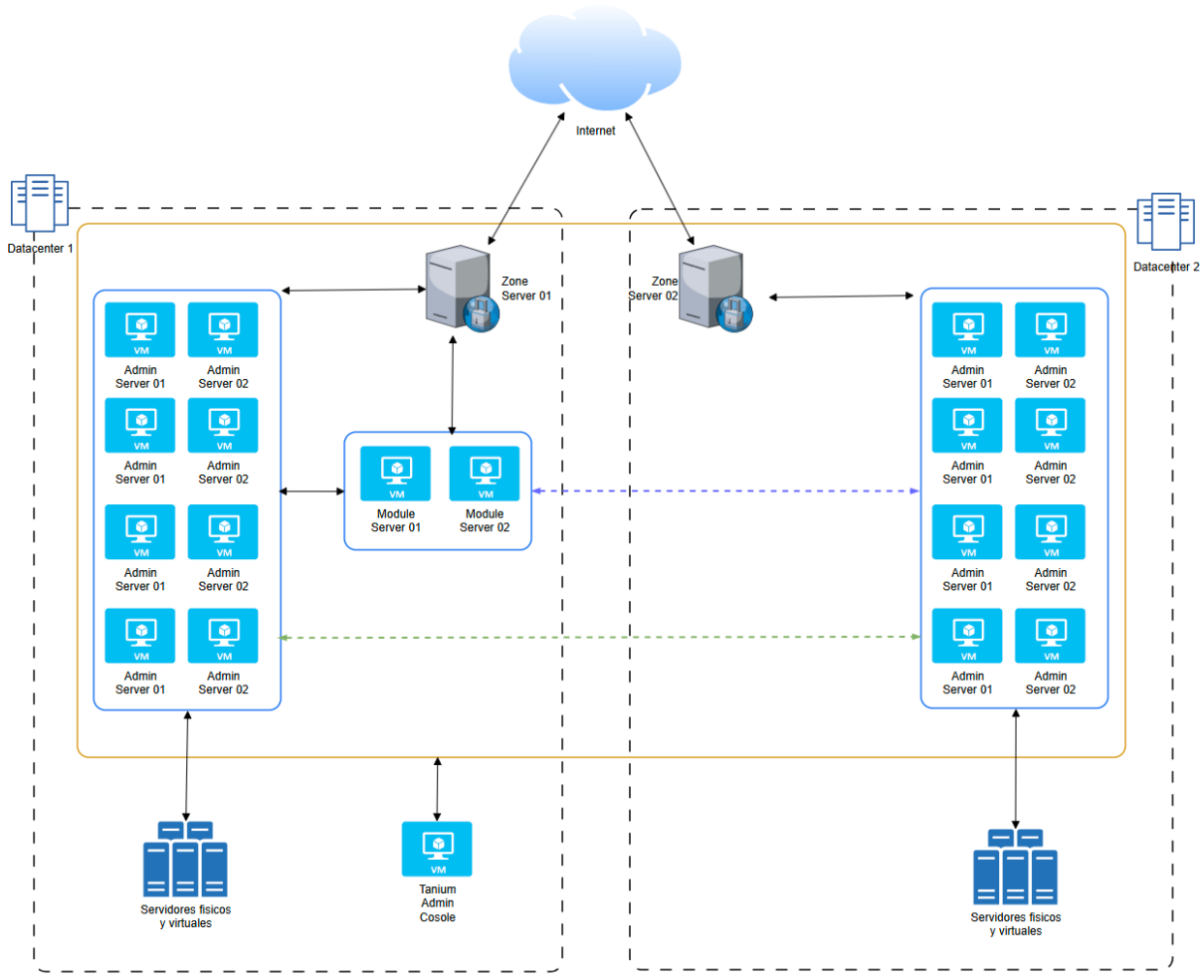


*Ilustración 54 Propuesta de Infraestructura Plataforma Tanium. Fuente: Elaboración propia .*

**Costos de Licencias y Software.**

A diferencia de la plataforma de Ansible, para la plataforma Tanium Experian solo necesita hacer las inversiones para obtener las licencias de los servidores que serán soportados por la plataforma. Sin embargo, como se mencionó anteriormente, la licencia básica

de Tanium y la de parcheo son costos separados y que deben cubrirse por cada servidor a ser parcheado.

Para Experian, el costo aproximado por la licencia básica es de $8 por servidor para un total de $16 000, mientras que la licencia para el módulo de parcheo es de $10 por servidor para un total de $20 000. Esto representa una inversión mensual de $36 000 para los 2000 servers a ser soportados por la plataforma.

**Costos de Desarrollo y Mantenimiento de Infraestructura.**

La plataforma de Tanium se encuentra desarrollada en ambiente de VMWare ESXi, por lo que los nuevos servidores que se requieren serán configurados en este mismo ambiente. En el caso de Experian, la empresa tiene ambientes empresariales de ESXi en ambos de sus datacenters para la región de Norte América.

De acuerdo con los requerimientos de infraestructura de Tanium para administrar el parcheo de aproximadamente 2000 servidores, se elabora la Tabla 13 Configuración VMWare ESXi de Tanium, que muestra cuales elementos serían necesarios para una adecuada implementación:

*Tabla 13 Configuración VMWare ESXi de Tanium.*

| Tipo Servidor | Cantidad | Detalle |
|---|---|---|
| Servidor de Módulos | 2 | Sistema Operativo Windows<br>16GB RAM, 8 vCPU<br>500G de almacenamiento |
| Servidor Administrador | 16 | Sistema Operativo Windows<br>32GB RAM, 12 vCPU<br>500G de almacenamiento |
| Servidor de Zona | 2 | Sistema Operativo Windows<br>16GB RAM, 8 vCPU<br>500G de almacenamiento |
| Servidor Base de Datos | 4 | Sistema Operativo Windows<br>32GB RAM, 12 vCPU<br>2000G de almacenamiento |

*Fuente: Elaboración propia.*

En base a estos elementos, y a los acuerdos entre Experian y VMWare, se presentan en la Tabla 14 el detalle de los precios aproximados mensuales, respectivos al mantenimiento de cada uno de los elementos de la infraestructura.

*Tabla 14 Cálculo aproximado del costo de la infraestructura en VMWare ESXi.*

| Servidor | Cantidad | Precio unitario mensual | Precio mensual subtotal |
|---|---|---|---|
| Servidor de Módulos | 2 | $199 | $398 |
| Servidor Administrador | 16 | $2 466.5 | $4 933 |
| Servidor de Zona | 2 | $199 | $398 |
| Servidor Base de Datos | 4 | $644 | $1 288 |
| | | **Precio Mensual Total** | **$7 017** |

*Fuente: Elaboración propia.*

**Costos de Capacitación y Recurso Humano.**

La implementación la plataforma de Tanium requeriría de la capacitación de los ingenieros del departamento de EITS, el cual se realizará durante el proceso de implementación e impartido por empleados propios de Experian expertos en la administración de la plataforma de Tanium, por lo que no tendría un gasto adicional por este rubro. Sin embargo, el departamento de EITS deberá invertir en 2 ingenieros contratistas para los cambios necesarios en la automatización por parte del equipo de soporte de ServiceNow, para realizar los cambios necesarios, ya que como se mencionó anteriormente, ServiceNow pasaría a ser el orquestador de las etapas y tareas de parcheo. El salario aproximado de los ingenieros contratistas es aproximadamente de $2 500, lo que representaría un total de $25 000 para 2 ingenieros por un periodo de 5 meses, que se estima se tardaría en implementar la solución.

Similar a la implementación de la plataforma de Ansible, se estima que la cantidad de incidentes que se reportan al mes se reducirán de un 85% a un 90% una vez implementada la

nueva propuesta, en comparación a lo que se genera actualmente. El 15 % de incidentes restantes, serian abordados con recursos dedicados, cubiertos por 12 ingenieros de TI con un salario mensual de unos $3 000. Esto generaría un costo adicional a tomar en cuenta de $36 000 mensuales.

*Tabla 15 Costo total de la implementación la plataforma Tanium.*

| Elemento | Costo Único | Costo Mensual Recurrente |
|---|---|---|
| Licencias Básica y de Parcheo | - | $36 000 |
| Infraestructura | - | $7 017 |
| Recurso Humano | $25 000 | $36 000 |
| **Costo Total** | **$25 000** | **$79 017** |

*Fuente: Elaboración propia.*

### 5.3 Análisis Económico.

A continuación, se presenta la Tabla 16 que muestra un resumen de las implicaciones económicas de la implementación de cada una de las plataformas analizadas para esta propuesta.

*Tabla 16 Resumen Económico de las Plataformas de Automatización.*

| Plataforma | Costo Único | Costo Mensual Recurrente |
|---|---|---|
| TrueSight BladeLogic | $60 000 | $90 533,34 |
| Ansible | $100 000 | $57 922,19 |
| Tanium | $25 000 | $79 017 |

*Fuente: Elaboración propia.*

Con la información presentada anteriormente se puede realizar una proyección anual para cada plataforma que permita representar la comparación de los costos de implementación de la solución propuesta, antes durante y después de ser completada.

**Proyección para TrueSight BladeLogic.** En la Ilustración 55 se presentan los detalles de dicha proyección.

| Año | Ene. | Feb. | Mar. | Abr. | May. | Jun. | Jul. | Ago. | Sep. | Oct. | Nov. | Dic. | Costo Anual |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2023 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | **$1 086 400,08** |
| 2024 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | **$1 146 400,08** |
| | | | | | | | $60 000 | Capacitacion BladeLogic | | | | | |
| 2025 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | **$1 086 400,08** |

*Ilustración 55 Proyección Anual BladeLogic. Fuente: Elaboración propia.*

Basados en los detalles de la proyección se puede determinar que:

- En el año 2023 Experian tuvo un gasto anual de $1 086 400,08 en el soporte de la solución actual de parcheo a través de Blade Logic.
- En el año 2024, de mantenerse la solución y realizar una inversión para capacitación del departamento de EITS, Experian tendría un gasto adicional en Julio para el pago de dicho entrenamiento, finalizando con un costo anual de $1 146 400.08 incurriendo en un aumento de $ 60 000.
- La capacitación tendría un plazo de 2 meses durante el 2024 finalizando en septiembre del mismo año.
- El año 2025 Experian mantendría el mismo costo anual en comparación al costo del año 2023 pero sin mayor beneficio económico u operacional inmediato.

**Proyección para Ansible Automation**. En la Ilustración 56 se presentan los detalles de dicha proyección.

| Año | Ene. | Feb. | Mar. | Abr. | May. | Jun. | Jul. | Ago. | Sep. | Oct. | Nov. | Dic. | Costo Anual |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2023 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | **$1 086 400,08** |
| 2024 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $57 922,19 | **$1 153 788,93** |
| | | | | | | | $100 000 | Implementacion Ansible | | | | | |
| 2025 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | $57 922,19 | **$695 066,28** |

*Ilustración 56 Proyección Anual Ansible Automation. Fuente: Elaboración propia.*

Basados en los detalles de la proyección se puede determinar que:

- En el año 2023 Experian tuvo un gasto anual de $1 086 400.08 en el soporte de la solución actual de parcheo a través de Blade Logic.
- En el año 2024, de implementarse la solución propuesta para el parcheo de los servidores mediane Ansible Automation Platform, Experian tendría un gasto adicional en Julio para el pago de la implementación y configuración de la solución, finalizando con un costo anual de $1 153 788.93 incurriendo en un aumento de $67 388.85 .
- La implementación de la nueva solución tendría un plazo de 4 meses durante el 2024 finalizando en noviembre del mismo año.
- El año 2024 tendría 1 de los 12 meses con el costo ajustado a la nueva solución, la cual representa una disminución del costo mensual en $ 32 611.15, permitiendo iniciar el retorno de la inversión antes del final de año.
- El año 2025 Experian tendría un gasto anual de $ 695 066.28 en el soporte de la nueva solución de parcheo, lo cual representa un ahorro de $ 391 333.8 anual en comparación al costo del año 2023.

El ahorro mensual de $ 25 377.81, permitiría a Experian recuperar su inversión inicial de $100 000 para la implementación de la solución para finales de marzo del 2025.

**Proyección para Tanium**. En la Ilustración 57 se presentan los detalles de dicha proyección.

| Año | Ene. | Feb. | Mar. | Abr. | May. | Jun. | Jul. | Ago. | Sep. | Oct. | Nov. | Dic. | Costo Anual |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2023 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | **$1 086 400,08** |
| 2024 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | $90 533,34 | **$1 111 400,08** |
| | | | | | | | $25 000 | Implementacion Tanium | | | | | |
| 2025 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | $79 017,00 | **$984 204,00** |

*Ilustración 57 Proyección Anual Tanium. Fuente: Elaboración propia*

Basados en los detalles de la proyección se puede determinar que:

- En el año 2023 Experian tuvo un gasto anual de $1 086 400.08 en el soporte de la solución actual de parcheo a través de Blade Logic.
- En el año 2024, de implementarse la solución propuesta para el parcheo de los servidores mediane Tanium, Experian tendría un gasto adicional en Julio para el pago de los recursos adicionales de ServiceNow, finalizando con un costo anual de $1 111 400.08 incurriendo en un aumento de $25 000 .
- La implementación de la nueva solución tendría un plazo de 5 meses durante el 2024 finalizando en diciembre del mismo año.
- Los cambios económicos con la nueva plataforma serian visibles hasta el año 2025.
- El año 2025 Experian tendría un gasto anual de $ 984 204.00 en el soporte de la nueva solución de parcheo, lo cual representa un ahorro de $ 102 196.08 anual en comparación al costo del año 2023.

El ahorro mensual de $ 8 516.34, permitiría a Experian recuperar su inversión inicial de $25 000 para la implementación de la solución para mitad de abril del 2025

## 5.4 Análisis de Viabilidad.

A continuación, se detalla la Tabla 17 que presenta una comparativa entre las descritas plataformas que pueden utilizarse como posible solución a la mejora del proceso de parcheo de servidores Red Hat en la empresa Experian.

*Tabla 17 Comparativa entre posibles soluciones.*

| | BladeLogic | Ansible | Tanium |
|---|---|---|---|
| No requiere implementación de nueva infraestructura | ☑ | ☒ | ☒ |
| No depende de agentes en los servidores soportados | ☒ | ☑ | ☒ |
| Presenta mayor beneficio económico | ☒ | ☑ | ☒ |
| Presenta menor complejidad de implementación | ☑ | ☒ | ☒ |
| Presenta mejora en la administración de los registros de parcheo | ☒ | ☑ | ☑ |
| Presenta mejora en la administración y asignación de incidentes | ☒ | ☑ | ☑ |
| Presenta mejora en la administración y desarrollo de código | ☒ | ☑ | ☑ |
| Presenta mejora en la calidad de soporte y calidad de vida del departamento EITS | ☒ | ☑ | ☑ |
| Brinda facilidad de escalar y desarrollo de mejoras en el futuro | ☒ | ☑ | ☑ |
| **Total Positivos** | 2 | 7 | 5 |

*Fuente: Elaboración propia.*

En base con la información de la comparativa anterior, se puede concluir que la plataforma recomendada para la propuesta de mejora del proceso de parcheo automático de servidores Red Hat para la empresa Experian seria Ansible Automation Platform.

## 5.5 Análisis de Riesgo de la Propuesta Seleccionada.

A continuación, se presenta la Tabla 18 que detalla el análisis de riesgo de la propuesta Ansible Automation Platform.

*Tabla 18 Análisis de Riesgo.*

| Riesgo | Detalle | Mitigación |
|---|---|---|
| Parcheo inesperado de servidores. | La posibilidad de que distintas personas del mismo equipo puedan actualizar las fechas de parcheo, puede causar que el proceso se ejecute en fechas u horas no deseadas si no hay coordinación por parte de los equipos. | Se envían notificaciones a los dueños de los servidores cada vez que se modifican o actualizan las fechas de parche de sus sistemas. O sea que se hagan manualmente durante el parcheo, o al inicio de cada ciclo cuando se actualizan automáticamente a las nuevas fechas. |
| Interrupciones de sistema de los clientes. | Al actualizarse todos los paquetes en el servidor, el mismo debe reiniciarse al final del proceso para aplicar los cambios, lo cual genera la interrupción momentánea del servicio. Si no se planea este mantenimiento las aplicaciones podrían quedar inaccesibles después del parcheo. | Los dueños de los servidores están a cargo de crear los scripts de Pre-patch y Post-Patch los cuales deben detener y apagar los servicios o aplicaciones en los servidores al iniciar el parcheo; y luego reiniciarlos al finalizar y completar el reinicio de los servidores para asegurarse que las aplicaciones estén disponibles.<br><br>De igual manera se implementa el sistema de control de cambios y prueba a través de SNOW lo que permite validar los parches d seguridad en ambientes de prueba antes de que impacten sistemas de producción. |

| | | |
|---|---|---|
| Interrupciones de sistema de la plataforma de parcheo. | Los sistemas informáticos están siempre propensos a sufrir interrupciones de sistema, y la infraestructura de AWS que aloja la plataforma de Ansible no es la excepción | El costo de las licencias tanto de Ansible como de AWS incluyen el soporte Premium que brinda sistemas redundantes en cada elemento disponible, así como ayuda inmediata las 24 horas del día los 7 días de la semana, proporcionando un 99.99% de resiliencia en los sistemas. |
| Utilización no autorizada o inapropiada de credenciales. | Ansible requiere de las credenciales de administrador para ingresar a los sistemas y realizar tareas de parcheo y automatización. Usuarios con acceso a la plataforma podrían tener acceso a los credenciales y contraseñas. | Habilitar una credencial de administrador, centralizada en la plataforma de CyberArk, lo cual permite la rotación de la contraseña cada 24 horas y puede ser integrada con Ansible para mantener la misma actualizada durante el proceso de Patching. De esta manera se controla el acceso a la misma mediante las bóvedas de seguridad de CyberArk. |
| Familiarización con el proceso. | La implementación de una nueva solución puede llevar a la confusión de tanto los clientes como de los equipos de soporte, que a su vez puede generar interrupciones de sistema a causa de una errónea interpretación del proceso. | Comunicación constante y documentación para los equipos de soporte y clientes, que puedan informar sobre los cambios esperados en el ambiente, procedimientos y cómo pueden afectar sus sistemas y aplicaciones. |

*Fuente: Elaboración propia.*

### 5.6 Cronograma de Actividades de Implementación de la Propuesta Seleccionada.

El periodo de implementación de la solución de parcheo a través de Ansible dentro del ambiente de Experian está estimado en un tiempo de 16 semanas, incluyendo desde el diseño de la infraestructura hasta la migración de los servidores del ambiente de Producción a la nueva solución, que constaría de la última etapa del proyecto.

A continuación, se detalla la propuesta de actividades en estas 16 semanas.

*Tabla 19 Cronograma de actividades de implementación.*

| Semana | Actividad | Detalles | Equipos Participantes |
|---|---|---|---|
| 1 | Creación de la infraestructura requerida.<br><br>Instalación Ansible Automation Platform. | Crear tablas requeridas en ServiceNow.<br>Crear infraestructura AWS.<br>Cread cuentas y credenciales de parcheo en CyberArk.<br>Instalar y configurar Ansible Automation Platform.<br>Solicitar apertura de firewall requerido. | Red Hat.<br>Equipo de soporte de Ansible de Experian<br>Equipo de seguridad de Experian.<br>Equipo de redes de Experian.<br>Equipo de soporte de AWS para Experian. |
| 2 | | | |
| 3 | Integración de tecnologías y desarrollo de la solución. | Desarrollo del código de parcheo en BitBucket e integrar a Ansible.<br>Integrar credenciales de CyberArk a Ansible.<br>Integrar base de datos de ServiceNow a Ansible.<br>Creación de templates o procesos requeridos en Ansible para la ejecución de las tareas. | Red Hat.<br><br>Equipo de soporte de Ansible de Experian. |
| 4 | | | |
| 5 | | | |
| 6 | Pruebas de rendimiento, validaciones y mejoras. | Realizar pruebas de flujo y carga de trabajo de la solución diseñada en un ambiente controlado.<br>Documentación de áreas de mejora y problemas encontrados<br>Desarrollo de soluciones a problemas encontrados en la validación. | Red Hat.<br><br>Equipo de soporte de Ansible de Experian.<br><br>Equipo de soporte de Servidores Linux (EITS) |
| 7 | | | |
| 8 | Documentación y capacitación. | Elaboración de la documentación requerida para el continuo soporte de la solución, procesos de escalación y políticas definidas del proceso.<br>Capacitación a los equipos y clientes en las funcionalidades y políticas de la nueva solución.<br>Capacitación a los equipos de soporte en el mantenimiento y mejora de la solución. | Red Hat.<br><br>Equipo de soporte de Ansible de Experian.<br><br>Equipo de soporte de Servidores Linux (EITS)<br><br>Clientes |
| 9 | Migración, prueba y monitoreo de los servidores de ambiente de prueba | Migrar los servidores de los ambientes que no son de producción a la nueva plataforma de parcheo.<br>Realizar el proceso en grupos. | Equipo de soporte de Servidores Linux (EITS)<br><br>Clientes |
| 10 | | | |
| 11 | | | |
| 12 | | | |
| 13 | Migración, prueba y monitore de los servidores de ambiente de producción | Migrar los servidores de los ambientes que son de producción a la nueva plataforma de parcheo.<br>Realizar el proceso en grupos | Equipo de soporte de Servidores Linux (EITS)<br><br>Clientes |
| 14 | | | |
| 15 | | | |

*Fuente: Elaboración propia.*

# Capítulo VI: Conclusiones y Recomendaciones.

## 6.1 Conclusión general

La propuesta de mejora de automatización del proceso de parcheo para servidores Red Hat en Experian se presenta como una solución innovadora y eficiente para gestionar y proteger el entorno informático de la empresa. A partir del análisis de la situación actual, se identificaron las brechas y deficiencias existentes, las cuales se abordan mediante la implementación de herramientas modernas y metodologías automatizadas que optimizan el proceso de parcheo.

**Se concluye que la propuesta de mejora del proceso de parcheo no solo mejora la eficiencia operativa, sino que también reduce significativamente el tiempo y esfuerzo necesarios para mantener el ambiente actualizado y protegido de vulnerabilidades.** Esta solución integral responde a las necesidades actuales en el área de tecnologías de la información, garantizando la seguridad y confiabilidad de la infraestructura de Experian.

### *6.1.1 Conclusión 1.*

Se concluye que existen grandes deficiencias relacionadas en el proceso actual de parcheo respecto a la administración del registro de servidores, que comprende los subprocesos de agregar servidores al sistema de parcheo actual, calendarización de los servidores y creación de las tareas programadas de parcheo. Estas deficiencias incluyen el control de acceso de los registros, dependencia de ciertos individuos en el departamento de EITS, dependencia de otras infraestructuras de servidores y scripts para administrar notificaciones, roles no bien definidos o aplicados, creación de las tares de parcheo y las limitaciones que existen para poder realizar cambios a las fechas de parcheo de un servidor, y generan problemas de seguridad y de separación de responsabilidades que evidencian un proceso ineficiente e inseguro.

### *6.1.2 Conclusión 2.*

Se concluye que la forma en que se realiza la administración de los incidentes de soporte por parte del proceso y plataforma actuales de parcheo es ineficiente y uno de las factores que incurren en el aumento de la carga de trabajo del equipo de soporte del departamento de EITS, debido a las deficiencias identificadas en la forma en que se almacenan los logs de información, los detalles que se agregan o muestran en los incidentes creados y la incapacidad de la plataforma actual de definir los equipos de soporte apropiados para cada tipo de incidente.

### *6.1.3 Conclusión 3.*

Se concluye que la administración del código de parcheo requiere de mejoras relacionadas no solo a la lógica y facilidad de entendimiento del proceso, sino también a la documentación del mismo, así como a la implementación de mejores prácticas en el desarrollo de software, las cuales son inexistentes en el ambiente actual, agregando un mayor nivel de complejidad y dificultad a la hora de realizar arreglos, cambios o mejoras.

### *6.1.4 Conclusión 4.*

Se concluye que en relación con todo el proceso de parcheo en la plataforma de automatización actual, existe deficiencias relacionadas a la documentación del flujo y etapas del proceso, la cual es muy limitada y en ciertas etapas o procesos inexistente, lo que aumenta en gran medida la dificultad y complejidad que tiene el departamento de EITS para el desarrollo de cambios o mejoras que puedan mitigar la cantidad de incidentes generados mensualmente por este proceso.

### *6.1.5 Conclusión 5.*

Se concluye que la plataforma actual de TrueSight BladeLogic no cuenta con las características y funcionalidades que permitan mejorar en poca o gran medida el proceso automático de parcheo para los servidores Linux Red Hat, y que a largo plazo la

implementación de una plataforma más actualizada y diseñada con mejores capacidades para este proceso, así como de posibilidad de integraciones con otras tecnologías, brinda un mayor beneficio a la empresa Experian así como al departamento de EITS en los ámbitos operativos, técnicos y económicos.

## 6.2 Recomendaciones

### *6.2.1 Recomendación 1.*

Se recomienda la migración de la base de datos de un cluster de servidores MySQL dedicados, a la base de datos CMDB que Experian ya utiliza y es provista por la plataforma ServiceNow. Esta migración permitiría utilizar los procesos de automatización y generación de formularios, vistas, registros y dashboards que provee la plataforma y permitiría agregar controles de acceso a los registros por usuarios o grupos de usuarios, calendarización más eficiente y visualmente amigable con los usuarios, soporte de cambios de último momento cuando se requieran y proveer información cuando no sean permitidos. También se habilitaría la capacidad de mostrar la información del estado del servidor en cada una de las etapas del proceso de parcheo.

### *6.2.2 Recomendación 2.*

Se recomienda que, una vez realizada la migración de la base de datos de parcheo en MySQL a CMDB de ServiceNow, y al ser la misma plataforma que administra los incidentes de soporte, se realice la integración entre la tabla de parcheo y las tablas relacionadas para administración de los incidentes con el objetivo de optimizar la información que se brinda en cada uno de los incidentes que se puedan crear durante el proceso de parcheo y poder crear relaciones e historial entre los servidores y sus incidentes. De esta forma se habilita la capacidad al equipo de soporte de EITS de identificar posibles patrones o problemas mayores

dentro del proceso; así como de determinar con prontitud la complejidad de cada incidente y el tiempo requerido para resolverlos.

### *6.2.3 Recomendación 3.*

Se recomienda la migración del código de parcheo de la plataforma BladeLogic y del lenguaje de programación NSH, a la plataforma BitBucket y al lenguaje de programación YAML, con el objetivo de aprovechar las características que brinda esta plataforma, ya utilizada por Experian en otros proyectos y departamentos, para la optimización en el desarrollo de código, implementación de mejores prácticas en el mercado, simplificar la escalabilidad del proceso y capacitar a los miembros del departamento de EITS para implementar cambios y mejoras futuras.

### *6.2.4 Recomendación 4.*

Se recomienda priorizar el documentar, en la plataforma interna de Experian para la creación de artículos de información y soporte, los detalles relacionados a los requerimientos, diseño, infraestructura, procesos de implementación y soporte de la solución y plataforma que se desee implementar; así como definir un proceso de documentación estándar de artículos de conocimiento para los procesos que surjan relacionados a la resolución de incidentes. Esto con el objetivo de mejorar el conocimiento general del equipo de soporte y los clientes respecto al proceso de parcheo, y reducir en gran medida la curva de aprendizaje para miembros nuevos de los equipos involucrados respecto a este tema.

### *6.2.5 Recomendación 5.*

Se recomienda la implementación de Ansible Automation Platform para la orquestación y automatización del proceso de parcheo automático de los servidores Linux Red Hat para la empresa Experian, con el objetivo de integrarla a las recomendaciones y plataformas propuestas, debido a las capacidades y características que presenta. Estas permitirán

rediseñar el proceso para que sea más escalable con respecto al crecimiento de la infraestructura soportada, incrementar la cantidad y calidad de la documentación del proceso y habilitar las estructuras que permitan futuras mejoras de desarrollo y adaptación en el proceso.

La implementación de estas recomendaciones permitirá a Experian optimizar el proceso de parcheo automático de servidores Red Hat, mejorar la seguridad y confiabilidad de su infraestructura informática, y liberar tiempo y recursos para enfocarse en iniciativas estratégicas más importantes. El éxito de este proyecto dependerá de la colaboración efectiva entre todas las partes interesadas, el compromiso de la gerencia y la asignación adecuada de recursos.

# Capítulo VII: Anexos.

## 7.1 Carta de aceptación de la empresa.

05 de diciembre de 2023

**Señor**
**Julián Córdoba Sanabria**
**Coordinador de Investigación**
**Universidad Hispanoamericana**

Estimado Señor:

Reciba un cordial saludo. Por este medio, me permito manifestar el interés de la empresa Experian en el proyecto denominado *"Propuesta de un proceso automático de parcheo para servidores Red Hat a través de Ansible"* presentado por el estudiante Manuel Salazar Gonzalez, cédula de identidad 206830213 para ser realizado en el equipo EITS Unix Operations, este proyecto es de suma importancia para nuestra organización ya que con él se busca solventar la necesidad de reemplazar el proceso actual que es realizado por una plataforma que está por cumplir su ciclo de vida, brindando herramientas a clientes para que puedan administrar de manera efectiva la calendarización de sus sistemas, y a los equipos de soporte una forma eficiente de reportar y resolver incidentes, adicionalmente Steven Monestel, IT Infrastructure Manager participará en la defensa del estudiante.

Quedo atento a cualquier consulta o detalle adicional.

Atentamente,

experian.
07 DIC. 2023
Recibido por:

**Steven Monestel**

**IT Infrastructure Manager**

**Experian**

## 7.2 Propuesta económica de la infraestructura de AWS.



Póngase en contacto con su representante de AWS: Comuníquese con el departamento de ventas

Exportar fecha: **7/23/2024** Idioma: **Español**

URL estimada: **https://calculator.aws/#/estimate?nc2=pr&id=ca0af21ba8af1d5164140679bf0072c654067e68**

### Resumen de la estimación

| Costo inicial | Costo mensual | Costo total de 12 months |
|---|---|---|
| **0,00 USD** | **5255,52 USD** | **63.066,24 USD** Incluye el costo inicial |

### Estimación detallada

| Nombre | Grupo | Región | Costo inicial | Costo mensual |
|---|---|---|---|---|
| **Amazon EC2** | No se ha aplicado ningún grupo | US East (Ohio) | 0,00 USD | 3866,55 USD |

**Estado**: -
**Descripción**:
**Resumen de la configuración**: Tenencia (Instancias dedicadas), Sistema operativo (Linux), Carga de trabajo (Consistent, Número de instancias: 32), Instancia EC2 por adelantado (m5.large), Pricing strategy (Spot Discount: -1), Habilitar la monitorización (desactivado), DT Entrada: Not selected (0 TB al mes), DT Salida: Not selected (0 TB al mes), DT Intra-región: (0 TB al mes)

| Nombre | Grupo | Región | Costo inicial | Costo mensual |
|---|---|---|---|---|
| **Amazon RDS for MySQL** | No se ha aplicado ningún grupo | US East (Ohio) | 0,00 USD | 278,46 USD |

**Estado**: -
**Descripción**:
**Resumen de la configuración**: Cantidad de almacenamiento (30 GB), Almacenamiento para cada instancia RDS (SSD de uso general (gp2)), Nodos (1), Tipo de instancia (db.m5.large), Utilización (solo bajo demanda) (100 %Utilized/Month), Opción de implementación (Multi-AZ), Modelo de precios (OnDemand)

| | | | | |
|---|---|---|---|---|
| **Elastic Load Balancing** | No se ha aplicado ningún grupo | US East (Ohio) | 0,00 USD | 86,51 USD |

**Estado**: -
**Descripción**:
**Resumen de la configuración**: Número de balanceadores de carga de aplicaciones (1)

| | | | | |
|---|---|---|---|---|
| **Amazon Virtual Private Cloud (VPC)** | No se ha aplicado ningún grupo | US East (Ohio) | 0,00 USD | 1024,00 USD |

**Estado**: -
**Descripción**:
**Resumen de la configuración**: DT Entrada: Internet (10 TB al mes), DT Salida: Este de EE. UU. (Norte de Virginia) (100 TB al mes), DT Intra-región: (0 TB al mes), Costo por transferencia de datos (1024)

Reconocimiento

La Calculadora de precios de AWS proporciona únicamente una estimación de sus tarifas de AWS y no incluye los impuestos que puedan aplicarse. El valor real de sus tarifas depende de una serie de factores, entre los que se incluye su uso real de AWS. Obtener más información

## Referencias Bibliográficas.


Atlassian. (2024). *Breve presentación de BitBucket*. Obtenido de www.atlassian.com: https://www.atlassian.com/es/software/bitbucket/guides/getting-started/overview#a-brief-overview-of-bitbucket

Atlassian. (2024). *Failover for Bitbucket Data Center*. Obtenido de confluence.atlassian.com: https://confluence.atlassian.com/enterprise/failover-for-bitbucket-data-center-687022231.html

Atlassian. (2024). *Qué es el control de versiones*. Obtenido de www.atlassian.com: https://www.atlassian.com/es/git/tutorials/what-is-version-control

Avast. (2024). *Exploits: todo lo que debe saber*. Obtenido de Exploits: todo lo que debe saber: https://www.avast.com/es-es/c-exploits

AWS. (2023). *¿Qué es la arquitectura basada en eventos (EDA)?* Obtenido de aws.amazon.com: https://aws.amazon.com/es/what-is/eda/

AWS Princing Calculator. (2024). *AWS Princing Calculator*. Obtenido de AWS Princing Calculator: https://calculator.aws/#/?nc2=pr

Ballejos, L. (2023). *Explicación del ciclo de vida de la gestión de parches*. Obtenido de www.ninjaone.com: https://www.ninjaone.com/es/blog/ciclo-de-vida-de-la-gestion-de-parches-explicacion/

BMC Software. (2024). *BladeLogic Server and Network Automation*. Obtenido de www.bmc.com: https://www.bmc.com/it-solutions/brands/bladelogic.html

Cavanaugh, S. (2020). *Control your content with private Automation Hub*. Obtenido de www.redhat.com: https://www.redhat.com/en/blog/control-your-content-with-private-automation-hub

CEUPE. (2021). *¿Qué es el análisis de la información?* Obtenido de www.ceupe.com.ar: https://ceupe.com.ar/blog/que-es-el-analisis-de-la-informacion/

CINDE. (2023). *Experian fortalece su huella con 250 nuevos empleos en Costa Rica*. Obtenido de cinde.org: https://www.cinde.org/es/noticias/experian-fortalece-su-huella-con-250-nuevos-empleos-en-costa-rica

Cybermark. (2024). *TrueSight Server Automation*. Obtenido de cybermark.net: https://www.cybermak.net/partner/truesight-server-automation/

Deshpande, R. (2019). *YAML basics in Kubernetes*. Obtenido de developer.ibm.com: https://developer.ibm.com/tutorials/yaml-basics-and-usage-in-kubernetes/

Efiempresa. (2024). *Integración Tecnológica, Sus Características y Beneficios*. Obtenido de efiempresa.com: https://efiempresa.com/es/blog/integracion-tecnologica-beneficios/

Experian. (2023). *Treating data with respect*. Obtenido de www.experianplc.com: https://www.experianplc.com/responsibility/treating-data-with-respect

Experian. (2024). *¿Quiénes somos?* Obtenido de www.experian.es: https://www.experian.es/sobre-nosotros

Experian. (2024). *Experian*. Obtenido de www.experianplc.com: https://www.experianplc.com/

FasterCapital. (2023). *Importancia De Las Actualizaciones De Software Y Parches De Seguridad Regulares*. Obtenido de fastercapital.com: https://fastercapital.com/es/tema/importancia-de-las-actualizaciones-de-software-y-parches-de-seguridad-regulares.html/1

FasterCapital. (2024). *Integracion tecnologica Integracion perfecta Aprovechamiento de la tecnologia en asociaciones estrategicas*. Obtenido de fastercapital.com: https://fastercapital.com/es/contenido/Integracion-tecnologica--Integracion-perfecta--Aprovechamiento-de-la-tecnologia-en-asociaciones-estrategicas.html

Fonseca, L. (2021). *10 ejemplos de diagramas de casos de uso (y cómo crearlos)*. Obtenido de es.venngage.com: https://es.venngage.com/blog/diagramas-de-casos-de-uso/

Gustavo, B. (2023). *Bash Script: qué es, cómo escribir uno y ejemplos*. Obtenido de www.hostinger.es: https://www.hostinger.es/tutoriales/bash-script-linux

HostGator. (2023). *Bitbucket: descubre qué es, para qué sirve y cómo se usa*. Obtenido de www.hostgator.mx: https://www.hostgator.mx/blog/como-funciona-bitbucket/

INCIBE . (2017). *Amenaza vs Vulnerabilidad, ¿sabes en qué se diferencian?* Obtenido de www.incibe.es: https://www.incibe.es/empresas/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian

INCIBE. (2018). *Gestión de parches en sistemas de control*. Obtenido de www.incibe.es: https://www.incibe.es/incibe-cert/blog/gestion-parches-sistemas-control

Korzeniowski, P. (2016). *Parche los servidores correctamente sin perder tiempo*. Obtenido de www.computerweekly.com: https://www.computerweekly.com/es/consejo/Parche-los-servidores-correctamente-sin-perder-tiempo

Lifeder. (2021). *7 Técnicas e Instrumentos para la Recolección de Datos.* Obtenido de www.lifeder.com: https://www.lifeder.com/tecnicas-instrumentos-recoleccion-datos/

López, J. (2023). *Asociación clave para Detectar Vulnerabilidades: Fuentes Oficiales*. Obtenido de www.qualoom.es: https://www.qualoom.es/blog/suscripciones-para-la-deteccion-de-vulnerabilidades-de-seguridad/

Manjaly, S. (2023). *Qué es Ansible: la herramienta DevOps para automatizar tareas de IT*. Obtenido de blog.invgate.com: https://blog.invgate.com/es/ansible

Mata, L. (2021). *Los sujetos de estudio*. Obtenido de investigaliacr.com: https://investigaliacr.com/investigacion/los-sujetos-de-estudio/

Meza, M., & Zambrano, J. (2018). *IMPLEMENTACIÓN DE UN PROCESO Y POLÍTICAS PARA LA GESTIÓN DE ACTUALIZACIONES DE SOFTWARE Y PARCHES DE SEGURIDAD DE PRODUCTOS MICROSOFT EN UNA INSTITUCIÓN SIN FINES DE LUCRO.* (E. S. Litoral, Ed.) Obtenido de www.dspace.espol.edu.ec: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.dspace.espol.edu.ec/bitstream/123456789/45959/1/D-106537%20Meza-Zambrano.pdf

Microsoft. (2024). *Conceptos básicos del diseño de una base de datos*. Obtenido de support.microsoft.com: https://support.microsoft.com/es-es/topic/conceptos-b%C3%A1sicos-del-dise%C3%B1o-de-una-base-de-datos-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5#

Microsoft. (2024). *Tanium*. Obtenido de learn.microsoft.com: https://learn.microsoft.com/es-es/copilot/security/plugin-tanium

Montesino, R., Baluja, W., & Porvén, J. (2013). *Gestión automatizada e integrada de controles de seguridad informática* . (V. RIELAC, Ed.) Obtenido de scielo.sld.cu: http://scielo.sld.cu/pdf/eac/v34n1/eac04113.pdf

Muñoz , C. (1998). *Cómo elaborar y asesorar una investigación de tesis.* Editorial Prentice-Hall.

Naeem, T. (2023). *Diseño de bases de datos: aprenda a diseñar una buena base de datos*. Obtenido de www.astera.com: https://www.astera.com/es/type/blog/all-you-need-to-know-about-database-design/

NorthWare. (2022). *Requerimientos en el desarrollo de software y aplicaciones*. Obtenido de www.northware.mx: https://www.northware.mx/blog/requerimientos-en-el-desarrollo-de-software-y-aplicaciones/

Ortega, C. (2024). *Análisis de la información: Qué es, etapas, tipos y ejemplos*. Obtenido de www.questionpro.com: https://www.questionpro.com/blog/es/analisis-de-la-informacion/

Pérez, J., & Merino, M. (2023). *Lenguaje de Programación*. Obtenido de definicion.de: https://definicion.de/lenguaje-de-programacion/

Prakash, A. (2024). *Bash Basics Series #1: Create and Run Your First Bash Shell Script*. Obtenido de itsfoss.com: https://itsfoss.com/create-bash-script/

Ramirez, K. (2021). *Repositorio Institucional Sapiencia.* Obtenido de Repositorio Institucional Sapiencia: http://13.87.204.143/xmlui/handle/123456789/6663

Red Hat. (2019). *¿Qué es la arquitectura basada en eventos?* Obtenido de www.redhat.com: https://www.redhat.com/es/topics/integration/what-is-event-driven-architecture

Red Hat. (2022). *Conceptos básicos de Ansible*. Obtenido de www.redhat.com: https://www.redhat.com/es/topics/automation/learning-ansible-tutorial

Red Hat. (2023). *El concepto de automatización*. Obtenido de www.redhat.com: https://www.redhat.com/es/topics/automation/whats-it-automation

Red Hat. (2023). *¿Qué es YAML?* Obtenido de https://www.redhat.com/: https://www.redhat.com/es/topics/automation/what-is-yaml

Red Hat. (2024). *Red Hat Ansible Automation Platform*. Obtenido de Red Hat Ansible Automation Platform: https://www.redhat.com/es/technologies/management/ansible/pricing

Reyes, A. (2019). *Etapas en el diseño de Base de Datos*. Obtenido de www.slideshare.net: https://www.slideshare.net/anielkar/etapas-en-el-diseo-de-base-de-datos

Reyes, E. (2022). *Metodología de la Investigación Científica.* Publishing Inc.

RockContent. (2018). *Conoce los tipos de lenguaje de programación más usados en la actualidad*. Obtenido de rockcontent.com: https://rockcontent.com/es/blog/tipos-de-lenguaje-de-programacion/

Ruelas, U. (2017). *¿Qué es la serialización o marshalling?* Obtenido de codingornot.com: https://codingornot.com/que-es-la-serializacion-o-marshalling

Sampieri, R., & Mendoza, C. (2018). *Metodología de la Investigación. Las Rutas Cuantitativa, Cualitativa y Mixta.* McGraw-Hill Interamericana.

ScienceTech Easy. (2021). *Serialization in Java | Deserialization, Example*. Obtenido de www.scientecheasy.com: https://www.scientecheasy.com/2021/07/serialization-in-java.html/

ServiceNow. (2024). *¿Qué es ITSM?* Obtenido de www.servicenow.com: https://www.servicenow.com/latam/products/itsm/what-is-itsm.html

ServiceNow. (2024). *¿Qué es ServiceNow?* Obtenido de www.servicenow.com: https://www.servicenow.com/latam/what-is-servicenow.html

ServiceTonic. (2024). *¿Qué es ITSM?* Obtenido de www.servicetonic.com: https://www.servicetonic.com/es/service-desk/que-es-itsm/

Tanium Inc. (2024). *Autonomous Endpoint Management*. Obtenido de tanium.com: https://www.tanium.com

TechTarget. (2024). *Definition Shell Script*. Obtenido de www.techtarget.com: https://www.techtarget.com/searchdatacenter/definition/shell-script

Telefónica. (2023). *7 ventajas y desventajas de las TIC en la educación*. Obtenido de www.telefonica.com: https://www.telefonica.com/es/sala-comunicacion/blog/ventajas-desventajas-tic-educacion/

Torres, D. (2024). *Qué es el método Delphi, para qué sirve y ejemplos.* Obtenido de blog.hubspot.es: https://blog.hubspot.es/sales/metodo-delphi

Valdez, V. (2015). *Hablando del desarrollo en México, podemos inferir que un gran número de los proyectos de desarrollo de sistemas fracasan por no realizar una adecuada definición, especificación, y administración de los requerimientos.* Obtenido de www.northware.mx: https://www.northware.mx/blog/tecnicas-efectivas-para-la-toma-de-requerimientos/

Zaidman, E. (2017). *Seguridad informática¿Vulnerabilidades técnicas o errores humanos?* Obtenido de revistas.unlp.edu.ar: https://revistas.unlp.edu.ar/econo/article/view/3638/3438