



PLAN DE MANTENIMIENTO EVOLUTIVO

PLAN DE MANTENIMIENTO EVOLUTIVO

MINISTERIO DE MINAS Y ENERGÍA

REPÚBLICA DE COLOMBIA

ESTRATEGIA DE TRANSFORMACIÓN DIGITAL

Pág.

INTRODUCCIÓN.....	3.
1. CLASIFICACIÓN DE SISTEMAS DE INFORMACIÓN POR PRIORIDAD.....	4.
2. JUSTIFICACIÓN TÉCNICA DE SISTEMAS MONOLÍTICOS.....	6.
3. CLASIFICACIÓN MONOLITO VS PRIORIDAD 1.....	9.
4. PLAN DE ACCIÓN Y COMPROMISOS CONSOLIDADOS.....	13.
5. <u>PLAN TECNICO DE SOLUCION POR APLICACIÓN (PRIORIDAD1)</u>	<u>14.</u>
6. <u>AVANCES EN LA GESTIÓN (ARQUITECTURA DE SIS-INF Y PMO).....</u>	<u>16.</u>
CONCLUSIONES.....	17.

ÍNDICE DE TABLAS

<u>Tabla 1: Control de cambios</u>	18
<u>Tabla 2: Ruta de aprobación</u>	18

INTRODUCCION

Este plan técnico define la hoja de ruta para la modernización de los sistemas de información de misión crítica (Prioridad 1) del Ministerio de Minas y Energía que actualmente operan bajo arquitecturas monolíticas. El objetivo central es dar cierre a la **Brecha 29**, garantizando la escalabilidad, resiliencia y seguridad del ecosistema tecnológico.

Se establece la creación de un plan de mantenimiento evolutivo a partir del catálogo de sistemas de información, previamente se creó el documento de “aplicaciones priorizadas” en donde la clasificación responde a la siguiente lógica de priorización:

- **Prioridad 1:** Sistemas core que soportan la resiliencia institucional y la soberanía de datos (operativos, administrativos, misionales críticos).
- **Prioridad 2:** Sistemas habilitadores de la interoperabilidad y la relación directa con el ciudadano (portales, gestión documental y servicios de información).
- **Prioridad 3:** Soluciones de innovación y tecnologías emergentes, enfocadas en movilidad y agilidad.

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

1. CLASIFICACIÓN DE SISTEMAS DE INFORMACIÓN POR PRIORIDAD.

Se genero una matriz en Excel identificando cada sistema de información por la clasificación anteriormente mencionada, dando como resultado la matriz : “clasificación sistemas información por prioridad.

TABLA 1. CLASIFICACION PRIORIDAD

Sigla	Descripción	Prioridad
SIGAME	Indicadores de gestión, metas y riesgos.	1
GLP	Registro subsidios GLP.	1
SUNA	Seguimiento a proyectos Fondo de Fomento.	1
SDG	Declaración de Producción Gas Natural.	1
SISEG	Fondo de Solidaridad (Energía/Gas).	1
SITH	Transporte de Hidrocarburos.	1
NEON	Recursos físicos y contratación.	1
AVANZAME	Gestión Integral Proyectos Minero-Energético.	1
SIVEEIC	Vigilancia estratégica e inteligencia.	1
MINA CONOCIMIENTO	Repositorio manuales y procedimientos.	1
NORMATIVAME	Consulta de normatividad interna.	1
GEOSERVER	Servidor datos geoespaciales.	1
ENERGÍA CAMBIO	Tarifas de energía.	1
SARA	Administración Talento Humano.	1
AUTOGEST	Autoservicio Talento Humano.	1
ENCUESTA CULTURA	Medición cultura organizacional.	1
OBSERVATORIO AMBIENTAL	Monitoreo asuntos ambientales.	1
MONITOREO RED	Infraestructura tecnológica (OpManager).	1

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

Sigla	Descripción	Prioridad
CARGA ME	Reporte estaciones de carga eléctrica.	2
SICOM	Comercialización/distribución combustibles.	2
ARGO SGDA	Gestión documental y archivos.	2
RAIS	Inventario fuentes de radiación.	2
PINES	Seguimiento permisos ambientales.	2
SI.MINERO	Automatización trámites industria minera.	2
MICROSITIO PIGCCME	Portal información cambio climático.	2
GEOPORTAL	Gestión/visualización información geográfica.	2
DIIE	Dictámenes de instalaciones eléctricas.	2
GEOVISOR IDE-ME	Gestión datos geoespaciales.	2
GEONETWORK	Catálogo recursos geoespaciales.	2
VUT	Ventanilla Única de Trámites.	2
SIGASA	Gestión conflictos socioambientales.	2
EITI	Transparencia industrias extractivas.	2
LINEA ETICA	Buzón integridad y transparencia.	2
INTRANET	Información interna funcionarios.	2
PW	Portal Web (Sede Electrónica).	2
MOODLE	Aula Virtual cursos.	2
KOHA	Gestión biblioteca.	2
REDCOME	Gestión conocimiento colaborativa.	2
YACO	Plataforma telefonía e integración.	2
SIGDI	Guías Digitales.	2

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

Sigla	Descripción	Prioridad
APLICATIVO ASISTENCIAS	Gestión asistencia reuniones.	3
AVANZAME PLUS MOBILE	Reporte avance proyectos en sitio.	3
AVANZAME APP MOBILE	Consulta ciudadana proyectos.	3
CARNET VIRTUAL	Consulta carnet digital.	3
TAIGA	Gestión proyectos TI (ágiles).	3
INDY	Asistente Virtual.	3

Apuntando paralelamente a cerrar la brecha 23 (Crear un DRP más robusto para el aplicativo. Cambiar a una arquitectura de microservicios para evitar la arquitectura monolito y tener un esquema de alta disponibilidad; Implementar la autenticación con el Directorio Activo de la entidad para gestionar correctamente el ciclo de vida de los usuarios.)

A partir del análisis exhaustivo del “DOCUMENTO ANALISIS DE APLICACIONES MONOLITICAS” y el archivo “CATÁLOGO O DIRECTORIO DE SIS-INF”, se han identificado todos los sistemas de información clasificados como monolíticos en el ecosistema del Ministerio de Minas y Energía.

2. JUSTIFICACIÓN TÉCNICA DE SISTEMAS MONOLÍTICOS.

A partir del análisis exhaustivo del “DOCUMENTO ANALISIS DE APLICACIONES MONOLITICAS” y el archivo “CATÁLOGO O DIRECTORIO DE SIS-INF”, se han identificado todos los sistemas de información clasificados como monolíticos en el ecosistema del Ministerio de Minas y Energía.

A continuación, se presenta la matriz detallada con el nombre del sistema y la justificación técnica de su clasificación:

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

TABLA 2. JUSTIFICACION MONOLITO.

SISTEMA DE INFORMACIÓN	DESCRIPCIÓN / RAZÓN DE CLASIFICACIÓN COMO MONOLITO
SIGAME (Proceso de Planeación Estratégica)	Utiliza Frontend Web Forms C# y Backend C# (.Net Framework 4.8) desplegado en IIS. Existe un acoplamiento rígido entre la lógica de presentación y servidor, operando sobre un sistema operativo heredado (Windows Server 2012) con una base de datos centralizada (SQL Server 2019) donde coexisten múltiples módulos en un solo artefacto.
SISTEMA SUBSIDIOS GLP EN CILINDROS	El backend funciona como un proceso único e indivisible escrito en Python 3.6 (servidor Gunicorn), gestionando todas las peticiones con una base de datos unificada (Postgres 9.4). Cualquier actualización requiere detener y reiniciar todo el servidor.
SISEG (Subsidios Hidrocarburos y Energía)	Desarrollado en Java 8 y framework clásico Struts 2.0 / Spring. Toda su lógica y librerías de reportes se comprimen y despliegan en un único archivo "JAVA WAR" sobre servidores Wildfly/JBoss con BD PostgreSQL.
SITH (Transporte de Hidrocarburos)	Posee una arquitectura heredada (<i>legacy</i>) construida en Java 7 y framework Struts 2.0. Al igual que SISEG, todo el aplicativo se empaqueta en un único archivo "JAVA WAR".
ECOSISTEMA DE PORTALES CMS (Portal Web/Sede Electrónica, Intranet, EITI Colombia, Línea Ética)	Aunque usan lenguajes modernos (Python/Django con Wagtail), agrupan en un solo repositorio y proceso todas las funciones (gestión de contenidos, chatbots, usuarios, PQRSD). Tienen dependencia total de un servidor único Nginx y una BD Postgres centralizada.
NEON (Gestión de Recursos Físicos y Contratación)	Arquitectura monolítica con dependencia de componentes <i>legacy</i> ("Java antiguo"). Opera sobre versiones obsoletas (Windows Server 2008 R2 y SQL Server 2008) y centraliza todo el ciclo contractual en un solo bloque, funcionando como un "silo cerrado".
AVANZAME	Clasificado como un "monolito modular" en transición. Combina Java 8 EE + JSF en arquitectura MVC acoplada. Más de 15 módulos dependen de una sola base de datos (Postgres 12), generando que la caída de un componente pueda comprometer todo el sistema.

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

SISTEMA DE INFORMACIÓN	DESCRIPCIÓN / RAZÓN DE CLASIFICACIÓN COMO MONOLITO
MINA DEL CONOCIMIENTO	Desarrollo interno indivisible estructurado como un monolito en Python y base de datos PostgreSQL 13.7.
AULA VIRTUAL (MOODLE)	Es estructuralmente una aplicación monolítica ("Monolito Proyecto PHP") donde el núcleo (<i>core</i>), los plugins adicionales y la base de datos operan acoplados en el mismo entorno de ejecución.
APLICATIVO DE ASISTENCIAS DIGITALES	Proyecto Monolito basado en el framework Laravel (PHP 7) y MySQL. Se despliega como una unidad única de software sin conexión modular al directorio activo.
FONDO DE BECAS (VIEJO)	Arquitectura Monolito que acopla estrechamente el sitio web, la capa de acceso y la capa de desarrollo. Programado en Visual Basic .NET sobre infraestructura obsoleta.
ENCUESTA DE CULTURA	Es un caso de "monolito contenedorizado". A pesar de estar montado en Docker, todos los componentes (JavaScript, CSS, y backend Python/Django) siguen operando como una sola pieza unificada.
NORMATIVAME	Su arquitectura base es un monolito en PHP con BD MySQL centralizada. Al tener su código fuente encriptado, es una caja cerrada (bloque binario inalterable) que imposibilita cualquier desacoplamiento o mejora.
RAIS	Arquitectura en donde el frontend (React) y el backend acoplado (C# ASP.NET Core API) residen físicamente en el mismo servidor de Windows Server 2022 ("monolito todo en el mismo servidor").
SIMINERO	Plataforma histórica altamente acoplada y basada en tecnologías <i>legacy</i> como Java j2sdk y base de datos Oracle 11G.
SIGASA (Gestión de Asuntos Ambientales)	Sistema gestionado como una unidad de software cerrada y acoplada, construido bajo el framework Symfony 3.3 y MySQL.
ARGO SGDA	Monolito desarrollado en PHP 8 (basado en el software Orfeo). Centraliza toda la gestión documental institucional en un solo bloque ejecutable sobre Apache Tomcat.
MICROSITIO PIGCCME	Sistema desarrollado <i>In House</i> con arquitectura monolítica en PHP, donde el sitio web, la capa de acceso a datos y la capa de desarrollo están fuertemente acopladas

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

En conclusión estos sistemas se clasifican como monolitos porque comparten un "despliegue único" (ej. empaquetados WAR), escalabilidad vertical rígida, dependencia de infraestructura heredada y una carencia de APIs granulares (microservicios) que limita severamente la interoperabilidad entre ellos.

Basados en las dos matrices anteriores se crea una nueva matriz sobre la cual nos proponemos iniciar y proponer un plan de mantenimiento.

3. CLASIFICACIÓN MONOLITO VS PRIORIDAD 1.

Basados en las dos matrices anteriores se crea una nueva matriz sobre la cual nos apoyaremos para iniciar y proponer un plan de mantenimiento.

TABLA 3. CLASIFICACION MONOLITO VRS PRIORIDAD 1.

Sistema de Información	Descripción / Razón de clasificación como Monolito	Categoría de Prioridad
SIGAME (Proceso de Planeación Estratégica)	Utiliza Frontend Web Forms C# y Backend C# (.Net Framework 4.8) desplegado en IIS. Existe un acoplamiento rígido entre la lógica de presentación y servidor, operando sobre un sistema operativo heredado con una BD centralizada donde coexisten múltiples módulos en un solo artefacto.	Prioridad 1



PLAN DE MANTENIMIENTO EVOLUTIVO

SISTEMA SUBSIDIOS GLP EN CILINDROS

El backend funciona como un proceso único e indivisible escrito en Python 3.6, gestionando todas las peticiones con una base de datos unificada (Postgres 9.4). Cualquier actualización requiere detener y reiniciar todo el servidor.

Prioridad 1

SISEG (Subsidios Hidrocarburos y Energía)

Desarrollado en Java 8 y framework clásico Struts 2.0. Toda su lógica y librerías de reportes se comprimen y despliegan en un único archivo "JAVA WAR" sobre servidores Wildfly/JBoss.

Prioridad 1

SITH (Transporte de Hidrocarburos)

Posee una arquitectura heredada (*legacy*) construida en Java 7 y framework Struts 2.0. Al igual que SISEG, todo el aplicativo se empaqueta en un único archivo "JAVA WAR".

Prioridad 1

NEON (Gestión de Recursos Físicos y Contratación)

Arquitectura monolítica con dependencia de componentes *legacy* ("Java antiguo"). Opera sobre versiones obsoletas y centraliza todo el ciclo contractual en un solo bloque, funcionando como un "silo cerrado".

Prioridad 1

AVANZAME

Clasificado como un "monolito modular" en transición. Combina Java 8 EE + JSF en arquitectura MVC acoplada. Más de 15 módulos dependen de una sola base de datos (Postgres 12).

Prioridad 1

MINA DEL CONOCIMIENTO

Desarrollo interno indivisible estructurado como un monolito en Python y base de datos PostgreSQL 13.7.

Prioridad 1



PLAN DE MANTENIMIENTO EVOLUTIVO

NORMATIVAME	Su arquitectura base es un monolito en PHP con BD MySQL centralizada. Al tener su código fuente encriptado, es una caja cerrada (bloque binario inalterable) que imposibilita cualquier desacoplamiento.	Prioridad 1
ECOSISTEMA DE PORTALES CMS (Portal Web, Intranet, EITI Colombia, Línea Ética)	Aunque usan lenguajes modernos (Python/Django con Wagtail), agrupan en un solo repositorio y proceso todas las funciones (gestión de contenidos, chatbots, usuarios). Tienen dependencia total de un servidor único Nginx y una BD centralizada.	Prioridad 2 (Aplica para PW, Intranet, EITI y Línea Ética)
AULA VIRTUAL (MOODLE)	Es estructuralmente una aplicación monolítica ("Monolito Proyecto PHP") donde el núcleo (<i>core</i>), los plugins adicionales y la base de datos operan acoplados en el mismo entorno de ejecución.	Prioridad 2
RAIS	Arquitectura en donde el frontend (React) y el backend acoplado (C# ASP.NET Core API) residen físicamente en el mismo servidor de Windows Server 2022 ("monolito todo en el mismo servidor").	Prioridad 2
SIMINERO	Plataforma histórica altamente acoplada y basada en tecnologías <i>legacy</i> como Java j2sdk y base de datos Oracle 11G.	Prioridad 2
SIGASA (Gestión de Asuntos Ambientales)	Sistema gestionado como una unidad de software cerrada y acoplada, construido bajo el framework Symfony 3.3 y MySQL.	Prioridad 2



PLAN DE MANTENIMIENTO EVOLUTIVO

ARGO SGDA	Monolito desarrollado en PHP 8 (basado en el software Orfeo). Centraliza toda la gestión documental institucional en un solo bloque ejecutable sobre Apache Tomcat.	Prioridad 2
MICROSITIO PIGCCME	Sistema desarrollado <i>In House</i> con arquitectura monolítica en PHP, donde el sitio web, la capa de acceso a datos y la capa de desarrollo están fuertemente acopladas.	Prioridad 2
APLICATIVO DE ASISTENCIAS DIGITALES	Proyecto Monolito basado en el framework Laravel (PHP 7) y MySQL. Se despliega como una unidad única de software sin conexión modular al directorio activo.	Prioridad 3
FONDO DE BECAS (VIEJO)	Arquitectura Monolito que acopla estrechamente el sitio web, la capa de acceso y la capa de desarrollo. Programado en Visual Basic .NET sobre infraestructura obsoleta.	<i>No especificada / N/A</i> (No figura en el listado de priorización)
ENCUESTA DE CULTURA	Es un caso de "monolito contenedorizado". A pesar de estar montado en Docker, todos los componentes siguen operando como una sola pieza unificada.	<i>No especificada / N/A</i> (No figura en el listado de priorización)

4. ESTRATEGIA PARA EL CIERRE DE LA BRECHA 23.

Para lograr el cambio desde aplicaciones monolíticas hacia un esquema de alta disponibilidad y microservicios con autenticación centralizada, la estrategia arquitectónica se basa en tres pilares:

- **Transición a Microservicios y Contenedores:** Se realizará una descomposición funcional de los monolitos mediante el análisis de contextos delimitados (*Bounded Contexts*) para establecer las fronteras técnicas de cada nuevo microservicio. Las aplicaciones se empaquetarán y desplegarán utilizando orquestadores de contenedores (como Docker/Kubernetes) para permitir escalabilidad independiente y despliegues sin tiempo de inactividad.
- **Esquema de Alta Disponibilidad (HA) y Cloud:** La modernización exige la migración de cargas de trabajo a la nube (Brecha B1) y la configuración de clústeres de bases de datos (PostgreSQL y SQL Server) en esquemas Activo-Activo o Activo-Pasivo (Brecha B41). Esto debe integrarse con un Plan de Recuperación ante Desastres (DRP) sectorial desplegado en un Centro de Datos Alternativo.
- **Autenticación Centralizada (Directorio Activo):** Se debe abandonar la autenticación local (propia de cada monolito). Todo inicio de sesión debe redirigirse al proveedor de identidad (IdP) del Ministerio (Directorio Activo / Azure AD) utilizando protocolos de autenticación modernos como OIDC (OpenID Connect) u OAuth2, implementando herramientas de Gestión de Identidades y Accesos (IAM/IGA) para automatizar el ciclo de vida de las cuentas de usuario de manera transversal.

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

5. PLAN TECNICO DE SOLUCION POR APLICACIÓN (PRIORIDAD 1).

De acuerdo con el "Documento Análisis de Aplicaciones Monolíticas" y la clasificación de prioridades, las siguientes aplicaciones Tipo 1 tienen este plan de intervención:

NEON (Gestión de Recursos Físicos y Contratación)

- **Diagnóstico:** Monolito en "Java antiguo", acoplado, que opera como un silo de datos sin integración documental, sobre Windows Server y SQL Server 2008.
- **Solución Técnica:** Refactorización progresiva del backend. El diseño de software exige generar **consultas optimizadas en el backend** para que el despliegue administrativo se sincronice en tiempo de ejecución, reduciendo la carga de CPU de los clústeres. Además, se deben exponer APIs REST para inyectar automáticamente los contratos al gestor documental ARGO (Brecha B22).

AVÁNZAME (Gestión Integral de Proyectos)

- **Diagnóstico:** "Monolito modular" en Java 8 EE + JSF, donde más de 15 módulos dependen de una sola base de datos (Postgres 12), generando que una falla comprometa la sesión de todos los usuarios.
- **Solución Técnica:** Desacoplamiento de la base de datos única y extracción de los módulos funcionales hacia microservicios. A nivel evolutivo, se absorberá la gestión de proyectos mineros (desplazando el uso de Excel) y se generarán integraciones automatizadas de interoperabilidad (Brecha B34 y B25).

	<h2 style="text-align: center;">PLAN DE MANTENIMIENTO EVOLUTIVO</h2>
---	--

SIGAME (Proceso de Planeación Estratégica)

- **Diagnóstico:** Acoplamiento rígido con Frontend Web Forms C# y Backend .Net Framework 4.8 centralizado en IIS y base de datos SQL Server 2019.
- **Solución Técnica:** Reingeniería hacia un framework moderno (.NET Core/6+), separando el frontend del backend (API REST) para su contenedorización. Evolutivamente, debe integrarse con el nuevo sistema ERP institucional para sincronizar metas e indicadores financieros.

SISEG y SITH (Sistemas de Subsidios y Transporte de Hidrocarburos)

- **Diagnóstico:** Monolitos clásicos basados en Java 7/8 y el framework *legacy* Struts 2.0, empaquetados en un único archivo WAR.
- **Solución Técnica:** Refactorización tecnológica para dividirlos en servicios más granulares. Para el SITH, el plan evolutivo (Brecha B37) incluye la **integración con sensores de Internet de las Cosas (IoT)** en los oleoductos para la recolección de información automática, eliminando los reportes manuales.

SISTEMA SUBSIDIOS GLP EN CILINDROS y MINA DEL CONOCIMIENTO

- **Diagnóstico:** Monolitos indivisibles desarrollados en Python (Gunicorn) y PostgreSQL.
- **Solución Técnica:** Contenedorización de las aplicaciones, separación de la capa de acceso a datos y modernización a través del patrón API Gateway para estandarizar su comunicación interna.

NORMATIVAME (y sistemas legados SUNA / SDG)

- **Diagnóstico:** Bloques binarios inalterables (código fuente encriptado o sin acceso a fuentes), imposibles de desacoplar o actualizar.
- **Solución Técnica (Marchitamiento / Sunset):** Estas aplicaciones **NO** se migrarán a microservicios. La estrategia es el **marchitamiento técnico**, finalizando sus proyectos

	<h2>PLAN DE MANTENIMIENTO EVOLUTIVO</h2>
---	--

actuales mientras se extrae la información histórica. Serán sustituidas por nuevas plataformas propias de código abierto. En el caso de Normativame, se reemplazará por una solución apalancada en **Inteligencia Artificial** (Brecha B38) para la búsqueda avanzada de leyes y decretos

6. AVANCES EN LA GESTION (ARQUITECTURA DE SISTEMAS DE INFORMACION Y PMO).

Según el informe *“Trazabilidad del Portafolio de Iniciativas y Gestión de Brechas”*, el avance en la gobernanza y ejecución de estos planes demuestra una alta alineación estratégica:

Avances de la PMO (Project Management Office):

- Se delimitó una **gobernanza clara de responsabilidades**. El PMO Camilo Moreno lidera la validación global de los proyectos de modernización (Arquitectura Cloud y microservicios), mientras que la PMO Diana Erika Ramírez lidera directamente el seguimiento a las fábricas de software de los sistemas **NEON y Avánzame**.
- **Gestión Ágil en Avánzame**: El proyecto se está gestionando activamente mediante épicas e historias de usuario en la herramienta **Taiga**. Esta práctica asegura una inserción atómica de los requerimientos del MRAE directamente en el pipeline DevSecOps de las fábricas de software.
- **Unificación de Trazabilidad**: Se acordó unificar la "plantilla del mes" de la PMO con los formatos del equipo de Arquitectura para el reporte de proyectos, extrayendo hitos y bloqueos sin generar reprocesos administrativos.

Avances de los Arquitectos de Sistemas de Información:

- El equipo ha trascendido el diagnóstico inicial para asumir un **rol de gobernanza activa**. Esto ha garantizado que decisiones como el marchitamiento de Normativame/SUNA o la migración de NEON cuenten con una justificación técnica sólida y no se ejecuten como "silos".

- **Actualización iterativa:** Se determinó que los artefactos de arquitectura (como los modelos C4 y diagramas de despliegue) se actualizarán continuamente reflejando las iteraciones ágiles sobre el diseño original a medida que los proyectos avancen en la fábrica de software, evitando la degradación de la documentación.
- **Gestión de Vulnerabilidades Centralizada:** En conjunto con los líderes de seguridad, las vulnerabilidades detectadas en estos sistemas de información se están centralizando y gestionando a través de la herramienta institucional **Aranda**, notificando a los jefes directos y haciendo trazabilidad a la remediación.

	<p>PLAN DE MANTENIMIENTO EVOLUTIVO</p>
---	---

CONCLUSIONES

En conclusión, estos sistemas se clasifican como monolitos porque comparten un "despliegue único" (ej. empaquetados WAR), escalabilidad vertical rígida, dependencia de infraestructura heredada y una carencia de APIs granulares (microservicios) que limita severamente la interoperabilidad entre ellos.

El equipo ha trascendido el diagnóstico inicial para asumir un rol de gobernanza activa. Esto ha garantizado que decisiones como el marchitamiento de Normativame/SUNA o la migración de NEON cuenten con una justificación técnica sólida y no se ejecuten como "silos". Se determinó que los artefactos de arquitectura se actualizarán continuamente reflejando las iteraciones ágiles sobre el diseño original a medida que los proyectos avancen en la fábrica de software, evitando la degradación de la documentación. En conjunto con los líderes de seguridad, las vulnerabilidades detectadas en estos sistemas de información se están centralizando y gestionando a través de la herramienta institucional Aranda, notificando a los jefes directos y haciendo trazabilidad a la remediación

	PLAN DE MANTENIMIENTO EVOLUTIVO
---	--

Tabla 1: Control de cambios

CONTROL DE CAMBIOS		
Versión	Descripción	Fecha
1.0	PLAN DE MANTENIMIENTO EVOLUTIVO	14/04/2026

Tabla 2: Ruta de Aprobación

RUTA DE APROBACIÓN		
ELABORÓ O ACTUALIZÓ: HARVEY GORDILLO SAAVEDRA JEISSON FABIAN PEREZ RODRIGUEZ	REVISIÓN METODOLOGICA: JEISSON FABIAN PEREZ RODRIGUEZ	APROBÓ: GUILLERMO ALBERTO VALLEJO MESA