

Migración del Sistema de Información para la Administración del Talento Humano de la Policía Nacional: una revisión sistemática

Migration of the Human Talent Management Information System of the National Police: A Systematic Review

Migração do Sistema de Informação para Gestão do Talento Humano da Polícia Nacional: uma revisão sistemática

Nelson Javier Peñaranda Lizcano^{a*} | Duperly Graciela Ducuara Ramírez^b
| Diego Mauricio Delgado Villota^c | Yeison Alexis Murillo Pineda^d

^a<https://orcid.org/0000-0002-1755-5980> Policía Nacional de Colombia, Bogotá D. C., Colombia

^b<https://orcid.org/0000-0002-9746-034X> Policía Nacional de Colombia, Bogotá D. C., Colombia

^c<https://orcid.org/0000-0002-2894-667X> Policía Nacional de Colombia, Bogotá D. C., Colombia

^d<https://orcid.org/0000-0001-7485-9131> Policía Nacional de Colombia, Bogotá D. C., Colombia

- Fecha de recepción: 2021-02-26
- Fecha concepto de evaluación: 2021-04-25
- Fecha de aprobación: 2021-07-26
<https://doi.org/10.22335/rlct.v13i3.1422>

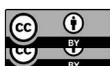
Para citar este artículo/To reference this article/Para citar este artigo: Peñaranda, N.J., Ducuara, D.G., Delgado, D.M., & Murillo, Y.A. (2021). Migración del Sistema de Información para la Administración del Talento Humano de la Policía Nacional: una revisión sistemática. *Revista Logos Ciencia & Tecnología*, 13(3), 112-127. <https://doi.org/10.22335/rlct.v13i3.1422>

RESUMEN

La Policía Nacional de Colombia tiene a su disposición sistemas de información que coadyuvan eficientemente al cumplimiento de su misionalidad, entre ellos se encuentra el Sistema para la Administración del Talento Humano (SIATH). Este sistema tiene 20 años de funcionamiento, fue desarrollado en lenguaje *Oracle Forms and Reports*, considerado actualmente como obsoleto y restrictivo, ya que depende de la ejecución de un *plugin* de *Java*, desplegado únicamente a través del entorno web de Internet Explorer, que, en la actualidad, no cuenta con soporte de Microsoft, ocasionando una amenaza en la disponibilidad de la información de la institución. Aplicando el protocolo de revisión sistemática y metaanálisis (PRISMA), se examinó en diferentes bases de datos bibliográficas, mediante ecuaciones de búsqueda y categorías de análisis, con criterios de inclusión y exclusión, artículos relacionados con la migración y reimplementación de aplicaciones. Como resultado se obtuvo que, la reimplementación del SIATH es la opción de migración más recomendada, pues conserva la arquitectura funcional y los objetos almacenados en la base de datos, utilizando la plataforma de desarrollo .NET, acorde a las capacidades institucionales, y permite innovar en la gestión del talento humano, garantizando la disponibilidad de la información y facilitando la interacción del usuario final.

Palabras clave: Diseño de sistemas, método, migración, recursos humanos, tecnología.

* Autor de correspondencia. Correo electrónico: nelson.penaranda@correo.policia.gov.co



ABSTRACT

The National Police of Colombia has at service information systems that helps efficiently to the fulfilment of its mission, among them we found the Management System for Human Resources (SIATH). This system has 20 years of functioning, it was developed in Oracle Forms and Reports language, and currently, it is considered obsolete and restrictive, because it depends on the execution of a plugin from Java, that it's deploy only by web site of Internet Explorer, which currently doesn't have Microsoft support, causing a menace in the availability of the information. So that, applying the revision protocol and metanalysis (PRISMA), it was checked in different data bases, by means of research equations and analysis categories with standard of inclusion and exclusion, related articles with the migration and implementation of applications. As a result, it was obtained that the reimplementation of SIATH is the option of the migration recommended, maintaining the functional architecture and the objects kept in the data base, using the develop platform .NET according to the institutional capacities, allowing to innovate in the process of human resources, guarantying the availability of information and facilitating the interaction of the final user.

Keywords: Systems design, method, migration, human resources, technology.

RESUMO

A Polícia Nacional da Colômbia dispõe de sistemas de informação que contribuem de forma eficiente para o cumprimento de sua missão, entre eles está o Sistema de Administração do Talento Humano (SIATH). Este sistema está em operação há 20 anos, foi desenvolvido na linguagem Oracle Forms e Reports, atualmente considerada obsoleta e restritiva pois depende da execução de um plugin Java, implantado apenas através do navegador web Internet Explorer, que, atualmente não conta com suporte da Microsoft, o que ameaça a disponibilidade das informações da instituição. Aplicando o protocolo de revisão sistemática e metanálise (PRISMA), os artigos relacionados à migração e reimplantação de aplicativos foram examinados em diferentes bases de dados bibliográficos, utilizando equações de busca e categorias de análise, com critérios de inclusão e exclusão. Como resultado, obteve-se que a reimplantação do SIATH é a opção de migração mais recomendada, pois preserva a arquitetura funcional e os objetos armazenados no banco de dados, utilizando a plataforma de desenvolvimento .NET, de acordo com as capacidades institucionais, e permite inovação na gestão do talento humano, garantindo a disponibilidade de informações e facilitando a interação do usuário final.

Palavras-chave: Desenho de sistemas, método, migração, recursos humanos, tecnologia.

En la década de los setenta, con el desarrollo de la tecnología de sistemas fueron creados una serie de programas que tenían como finalidad la gestión de los datos de las empresas; al comienzo se necesitó un potencial alto de conocimiento y grandes equipos para poder interactuar con estas aplicaciones, pero con el tiempo, los mismos desarrolladores han mejorado estas técnicas de creación, a fin de simplificar y popularizar la utilización de este tipo de software (Monnier, 2016).

Los sistemas de información desempeñan un papel importante en la sociedad y en el entorno en que habitamos, son herramientas poderosas y alternas de apoyo a los gerentes para la toma de decisiones. De hecho, los sistemas de información ayudan a la alta dirección a mo-

nitorear el rendimiento y funcionamiento de la empresa, identificando si se están presentando problemas de carácter estratégico, para tener oportunidades de mejora, con el propósito de crecer acertadamente, de esta manera pueden ofrecer mejores servicios a sus clientes y pueden posicionarse en un mundo tan competitivo (Hamidian & Ospino, 2015).

La Policía Nacional de Colombia en su estructura orgánica cuenta con la Oficina de Telemática (OFITE), encargada a nivel nacional de direccionar y gestionar todo lo relacionado con las tecnologías de la información; uno de sus grupos de trabajo es el de Administración de Recursos Tecnológicos, que es responsable de custodiar, dar soporte y mantenimiento a todos los sistemas de informa-

ción que se encuentran asignados a esta dependencia, entre ellos el Sistema para la Administración del Talento Humano (SIATH) desde el cual se gestiona toda la carga administrativa de la Policía y sus bases de datos constituyen un insumo para otras aplicaciones, dado que en estas se almacenan datos personales y laborales de todos los empleados activos y retirados de la institución, al igual que la de sus familiares (Colombia, 2015).

Teniendo en cuenta lo anterior, esta investigación realiza una revisión sistemática de la migración de los sistemas de información, para conocer las metodologías y arquitecturas de desarrollo utilizadas por diferentes autores, con el objetivo de obtener el discernimiento necesario para proponer una metodología de desarrollo que se ajuste mejor a la institución, considerando las nuevas tecnologías y las capacidades del capital humano con el que cuenta la Policía.

El SIATH fue desarrollado por OFITE en el año 2002 en lenguaje *Oracle Forms and Reports*; este sistema de información es implementado vía web únicamente a través del navegador de Internet Explorer, debido a que necesita la ejecución de un *plugin* de Java. De igual manera, es importante mencionar que desde el 12 de enero de 2016 Microsoft no proporciona actualizaciones de seguridad ni soporte técnico a este navegador y, en cualquier momento, puede ser inhabilitado, causando la indisponibilidad del sistema y paralizando administrativamente a la institución (Pang et al., 2016).

Por consiguiente, se realizó la revisión de trabajos de investigación y artículos científicos en revistas indexadas en la plataforma Scimago, en los cuales se evidencian, mediante diferentes estudios, cuáles son las mejores prácticas para el proceso de rediseño de un sistema de información; algunas de estas prácticas serán tomadas como referente para adquirir el conocimiento y tener la capacidad de tomar decisiones sobre la migración o reimplementación de este sistema heredado. Además, se proponen metodologías y plataformas de desarrollo que son necesarias para entender el código o lenguaje de programación heredado y obtener los cimientos requeridos, para escoger la nueva herramienta tecnológica en que se va a rediseñar el SIATH de la Policía Nacional de Colombia. Por último, se describe el método de recolección y validación de datos utilizado en la investigación, seguido de los resultados y de las conclusiones.

Esta investigación hace una revisión sistemática de literatura científica mediante tres categorías de análisis, pre-

viamente seleccionadas y que se relacionan directamente con el título del artículo, así: metodologías de desarrollo, planes de gestión y migración de sistemas (Gonzalo Muñoz et al., 2017).

Los sistemas de información son desarrollados e implementados para mejorar el trabajo realizado en las empresas, optimizando tiempos y recursos, así como también, minimizando los errores humanos que, con frecuencia, se pueden cometer. Las grandes empresas y entidades del Estado que llevan mucho tiempo en el mercado pueden tener dentro de su infraestructura software heredado (Medina et al., 2019).

El software heredado es un sistema con tecnología antigua o desactualizada que, por su robustez, complejidad u otras variables críticas, sigue vigente dentro de una organización; igualmente, estos sistemas no cuentan con soporte técnico ni mantenimiento, pero tampoco se pueden reemplazar fácilmente (Torres et al., 2018).

Según Yarza et al. (2021) el periodo de obsolescencia de los sistemas compuestos se está reduciendo y paralelamente se crea la necesidad de migrar e integrar los sistemas heredados. Con los avances de la tecnología, se propone utilizar dos traductores binarios adaptables a la máquina, uno de ellos con características dinámicas y el otro totalmente estático, para ser usado en la propiedad de tiempo real. Igualmente, se debe hacer la recompilación de código, con la premisa de tenerlo disponible y con un nivel suficientemente alto que lo categorice como independiente o al menos casi independiente de la plataforma en que se ejecuta.

Las cuatro áreas de investigación propuestas son: la traducción binaria, que forma una subárea de investigación por parte de las soluciones adaptables por máquina; el software heredado en tiempo real; la aplicación de la sincronización con soluciones de retargetable como un subgrupo; y la validación de los tiempos en la migración (Salazar et al., 2014).

Es necesario definir el modelo de sistema heredado en tiempo real, que ejecuta un conjunto de tareas periódicas alineadas con la política de programación estática predefinida; las siguientes secciones hacen referencia a los principales elementos de modelado del sistema heredado (Hamman et al., 2018), así:

1. Modelo de aplicación: define la relación del primer software heredado, compuesto por un conjunto de ta-

reas periódicas representadas a través de una tupla, donde las variables a intervenir son el periodo de la tarea por un límite superior del tiempo en ejecución (Guamán et al., 2018).

2. Modelo de ejecución: radica en un hiperperiodo que determina el tiempo después del cual el patrón de ejecución de la tarea se pueda repetir, y esta compuesto a su vez por un conjunto de marcos (Flore & Marx Gómez, 2020).
3. Aplicación de ejemplo: es un código de bloque secuencial que inicia leyendo los datos de entrada en su estado interno y termina calculando los resultados y actualizando sus estados. El comportamiento de la aplicación consiste en una secuencia lógica de alternancia de bits, alternando un bit específico en una variable de salida (Senn, 2017).

Un lenguaje de especificación de tiempo MULTIC es aplicable a los sistemas heredados, pues se basa en patrones de especificaciones y declaraciones parecidas al lenguaje natural, a través de trazas de tiempo que cumplen un patrón. *MULTIC Time Specification Language (MTSL)*, siendo un lenguaje de especificación de tiempo definido, asume que los nuevos sistemas se construirán a partir de componentes, que interactúan con otros elementos por medio de un conjunto de puertos, vinculados con conectores (Oliver et al., 2018).

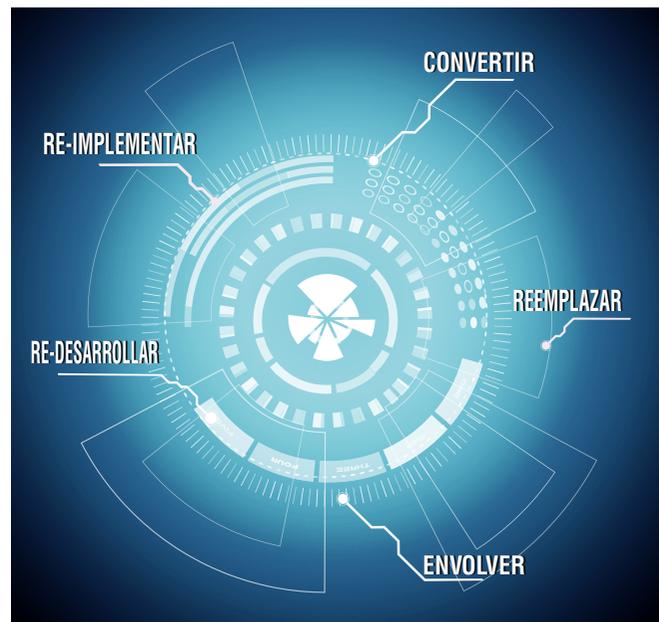
Dando continuidad al proceso de revisión documental, Li et al. (2019) presentaron una nueva metodología para el diseño de sistemas en el procesamiento de señales, usando interfaces para la programación de aplicaciones ligeras y de flujo de datos en diferentes capas de abstracción, con un conjunto integrado de herramientas y bibliotecas desarrolladas para cerrar cualquier brecha generada en el diseño.

En la figura 1 se observan las opciones de migración que se puede escoger en el momento de tomar la decisión respecto a la migración de un software heredado.

La metodología propuesta se enfoca en la implementación de un co-diseño de software con hardware con el mapeo de aplicaciones optimizado, para facilitar el desarrollo y la aplicación de los módulos a crear; para esto, se proponen plataformas y lenguajes de programación heterogéneos con una metodología de diseño y un conjunto integrado de herramientas y bibliotecas que se desarrollan para conseguir los objetivos, aplicando la

metodología STMC o STMCM, para las cuales es usual llevar el nombre de las diferentes instituciones que las desarrollan, por ejemplo: Sassari, Tampere, Marylan o Cagliari. Estas permiten la experimentación con diferentes niveles de abstracción durante el proceso del diseño, para no delimitar el talento humano vinculado al proceso, y permiten de manera productiva la iteración de modelos de flujos de datos, plataformas de destino heterogéneas e integración con otros lenguajes y arquitecturas de desarrollo.

Figura 1
Opciones de migración



La Arquitectura Orientada a Servicios (SOA) es el nexo que une las metas de negocio con el sistema de software; su misión es otorgar flexibilidad y automatización de los procesos tecnológicos con reducción de los costos de operación (Castro-León, 2020). En este artículo, se presenta un modelo de decisión para reingeniería que combina factores estratégicos y técnicos con análisis de costo-beneficio, en decisiones de integración versus migración; la arquitectura SOA proporciona un marco conceptual, basado en estándares para sistemas empresariales flexibles y adaptables. Esta noción es relativa a esta investigación, para evaluar las alternativas estratégicas en función de validar el costo beneficio del rediseño propuesto (Hustad & Olsen, 2021).

Sin embargo, existe un modelo de soporte de decisiones así: 1) análisis estratégico, 2) análisis de arquitectura y desarrollo de soluciones en especificaciones detalladas

del sistema y 3) análisis de costo-beneficio tangible de proyectos de reingeniería (Peralta-Ascue & Merma-Aroni, 2017).

Asimismo, Canfora et al. (2008) señalan como una buena opción la modernización de los sistemas heredados a plataformas con arquitectura orientada a servicios, y argumentan que el problema de rediseñar la capa de usuario original del sistema en los requerimientos se soluciona a través de un contenedor, el cual puede interactuar con el sistema en nombre del usuario.

En el proceso de migración propuesto, la técnica de envoltura presentada se puede manejar con el objetivo de reutilizar las funcionalidades de los usuarios, que ya estaban implementadas en el software heredado, se mencionan cuatro fases: seleccionar los servicios candidatos, hacer el involucramiento de los casos de uso seleccionados, implementación y validación del caso de uso envuelto (Salazar et al., 2014).

Continuando con la revisión sistemática, Sneed y Verhoef (2019) proponen la reimplementación para la migración de un sistema heredado, junto a la conversión, el empaquetado y la remodelación, estando estos términos y condiciones alineados a la finalidad de la migración del SIATH, para lo cual es necesario revisar la arquitectura técnica y reemplazar el código. Pero la arquitectura funcional y los casos de uso deben permanecer como están, ya que la conversión total del sistema no es factible y la remodelación puede ser demasiado costosa o puede representar un riesgo para la gestión de los procesos, teniendo en cuenta la complejidad de los objetos almacenados y la dinámica institucional.

El proceso de reimplementación propuesto comienza con la medición del código fuente actual y finaliza con la integración del componente reimplementado; en el mundo de las TI de hoy, el mercado ofrece varios tipos de lenguaje para elegir, por ejemplo: UML, BPML, HTML, XML, y lenguajes específicos de dominio (Trubiani et al., 2018).

Si el lenguaje de destino está orientado a objetos, los ingenieros deben utilizar diagramas UML, es decir, diagramas de clases, diagramas de secuencia y diagramas de transición de estado, que pueden aplicarse a cualquier organización que tenga en funcionamiento una aplicación o sistema de información con software heredado, para el cual es obligatoria la reimplementación a una arquitectura actual, con la cual pueda innovar y explotar nuevos resultados (Torres et al., 2018).

Los Lenguajes de Descripción de Arquitectura (ADL) actúan como una entidad consistente en cuatro "Cs", que por su abreviatura hace alusión a los componentes, conectores, configuraciones y restricciones; los ADL, en su creación, se basaron en los lenguajes de interconexión de módulos (MIL) de los años 70 (Woods & Bashroush, 2015).

Estos lenguajes se focalizan en la estructura de alto nivel de la aplicación antes que en los detalles de implementación de los módulos concretos, se describe la experiencia obtenida a partir de la creación de una descripción arquitectónica grande para un sistema de información complicado, muy similar al sistema SIATH de la Policía Nacional de Colombia que se quiere reimplementar. Los lenguajes ADL aplican un enfoque simple, de poca ceremonia y altamente prescriptivo, para minimizar la posibilidad de que los equipos produzcan inconsistencias y la necesidad de ser reelaborados posteriormente (Fernández García, 2011). Se propone el ADL como alternativa de modelado de un sistema de información a gran escala, utilizando el uso de una notación de descripción de arquitectura simple y específica de dominio en un contexto industrial (Harris & Harris, 2022).

Siguiendo la línea de métodos para la migración de software Kargar et al. (2019) encontraron que un modo de comprender un código de software es a través de la modularización. Existen sistemas de información complejos, robustos e importantes para la organización; estas son características que encajan en los conceptos de software heredado. Al ser sistemas robustos que necesitan comprender y entender un código para su migración, es importante optar por técnicas que conlleven a una correcta lectura y comprensión del código. Es en este punto que interviene la modularización, técnica que permite extraer la estructura del sistema, y con la que se lo puede clasificar por funcionalidades y métodos, logrando la comprensión de subsistemas que cooperan entre sí. Además, implica la conversión del sistema en módulos, para primero comprender el funcionamiento y luego migrar la aplicación por pasos, de acuerdo a las funciones del mismo (Vidal Silva et al., 2018).

Si bien la técnica de la modularización parece una herramienta práctica e intuitiva, el autor plantea que esta técnica solo funciona cuando el programa está desarrollado en un único lenguaje de programación, esto dificulta la aplicación para los sistemas de información actuales y para la propuesta específica de migración del SIATH, ya que la arquitectura de desarrollo está orientada a las

aplicaciones web que se caracterizan por estar escritas o compuestas de varios lenguajes de programación como JavaScript, HTML; para las funciones y métodos se utiliza Java, C++, C-Sharp, entre otros, y las bases de datos que actualmente utiliza la Policía Nacional de Colombia son de Oracle (Raffino, 2018).

Ante la dificultad de que la modularización solo pueda aplicarse a software escrito en un único lenguaje, se ha propuesto un método que se compone de dos grafos: grafo de dependencia semántica y grafo de similitud nominal, técnicas que se pueden aplicar independientemente de la sintaxis de la aplicación (Yandrapally et al., 2015).

La migración de sistemas de información sea o no heredados tiene un grado de complejidad, su motivación requiere de una necesidad justificada, que soporte asumir los riesgos y gastos que la migración conlleva. Generalmente, cuando se habla de migración de sistemas de información, se hace referencia a la migración del código fuente a otra tecnología, arquitectura o *framework*, así como la migración de los datos; sin embargo, la sola migración de datos puede ser compleja y requiere por sí sola un tratamiento diferencial. Dentro de la propuesta para la migración del SIATH no se ha contemplado la migración de datos, puesto que estos están bien apoyados en una tecnología con amplio soporte y desarrollo.

Pérez (2017) encontró una propuesta metodológica para la migración de datos que apunta de manera general a los hitos para tener en cuenta en dicho proceso. Dentro de la revisión de documentos y trabajos realizada por el autor, este propone que todo plan de migración debe: primero, definir la metodología de migración y el plan de trabajo; segundo, establecer métricas para supervisar la calidad del dato en la migración de bases de datos o la calidad del código para la migración de software; tercero, plantear un escenario y entorno de pruebas; cuarto, precisar la fuente y responsabilidad de los datos o del código según corresponda; quinto, prever posibles contingencias, por ello un plan de recuperación necesita de copias de respaldo completas; sexto, para la migración de datos es indispensable considerar ingreso y migración manual de datos. Por lo tanto, la construcción de un plan de migración debe recoger todas las recomendaciones posibles y prever todos los escenarios que se puedan desplegar. Aunque la metodología del autor parece estática y vertical, tiene componentes importantes que se deben tener en cuenta para el diseño de la propuesta en la migración del SIATH (Tsuji et al., 2021).

Adicionalmente, para este trabajo se consideró de gran importancia buscar artículos de investigación de las fuerzas militares de otros países, entre las que se encuentra la propuesta de Stacey Raza (2015), quien aplica una metodología moderna con procesos y procedimientos optimizados, para dar un mejor aprovechamiento de los recursos materiales, financieros y humanos a la problemática existente en la Armada del Ecuador, mediante la implementación de un SAP para la gestión del talento humano. La solución a esta problemática se hace mediante el modelo tradicional de migración de datos, que consiste en extraer, transformar y cargar los datos del sistema. Asimismo, se deben desarrollar actividades relacionadas con el sistema integrado de gestión institucional con una dinámica propia, planeando los objetivos administrativos y operativos, las tareas y metas a corto y largo plazo; así como la ejecución de los recursos, que incluyen los ajustes del plan operativo (García Cifuentes, 2016).

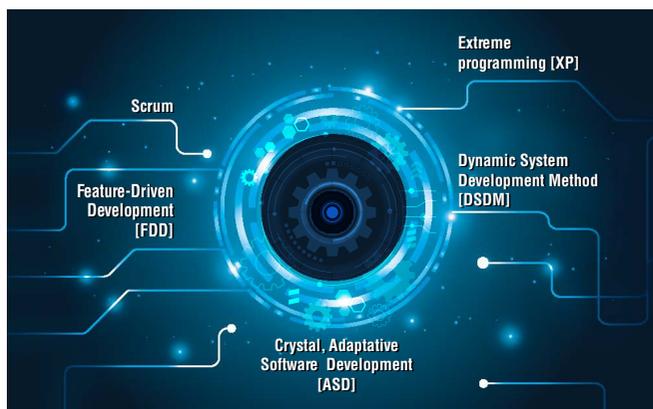
Igualmente, en las Fuerzas Armadas de Malasia (MAF), Arifin y Tajudeen (2020) estudiaron el impacto de los sistemas de información de recursos humanos en el entorno militar. El uso de las tecnologías de la información y las comunicaciones es usual en el sector público de Malasia, y han sido implementadas por el Plan Estratégico de TIC, como canal de comunicación, y para cumplir con la gestión administrativa. Para lograr el objetivo del estudio realizado, aplicaron un enfoque cualitativo que abarcó dos DGF y entrevistas personales, como herramientas idóneas para explorar los sentimientos personales y el comportamiento tomado por parte del equipo humano que conforma la MAF. Se tuvieron en cuenta las variables de calidad del sistema, de la información, del servicio, la infraestructura de TIC, la seguridad, el soporte del comandante y la formación para medir el impacto en el uso de estos sistemas, y los beneficios personales conseguidos fueron clasificados en tres categorías: desempeño laboral, productividad y toma de decisiones (Balalaie et al., 2018).

La usabilidad en los sistemas de información es una propiedad que las empresas deben controlar a través de su personal de tecnología; por esta razón, se programan planes de revisión para identificar qué tanto del software está en funcionamiento, y más al detalle, si todos los formularios y reportes son utilizados. Así, Chávez (2018) describió cuál es la realidad actual de las empresas y qué deben hacer para mejorar su nivel de productividad gracias a las aplicaciones; además, valoró el esfuerzo realizado por estas para aplicar técnicas y métodos en la ingeniería de usabilidad al proceso de desarrollo.

Por otra parte, se hace necesario la reimplementación del SIATH, puesto que, con el estudio realizado para levantar los requerimientos, se observó que la interfaz no es amigable y clara con el usuario. En ocasiones el sistema arroja errores que no son comprensibles y, en lugar de ayudar al usuario a identificar el inconveniente, hace más tediosa su labor, lo cual redundará en un gasto de tiempo adicional; por estar desarrollado en un lenguaje de código heredado, la escalabilidad del sistema para la creación de nuevos formularios es compleja y demorada, en este sentido, es muy poco dinámico para llevar a cabo los procesos que maneja la institución (Yu et al., 2021).

Las empresas que se deciden por una transformación digital de fondo necesariamente deben incluir, implementar y desarrollar metodologías ágiles con el propósito de hacer entrega de servicios y productos con calidad, costos reducidos, tiempos mínimos y optimización de recursos. En la ciudad de Cali, Colombia, Navarro et al. (2013) compararon de manera objetiva las diferencias fundamentales de las metodologías tradicionales y las ágiles con el propósito de disminuir el trámite que implica la aplicación de las metodologías tradicionales en los proyectos de mediana y pequeña escala. La figura 2 presenta algunas de las metodologías ágiles de desarrollo más utilizadas en la actualidad.

Figura 2
Metodologías ágiles de desarrollo



Esta investigación fue importante porque permitió encontrar una metodología que se adapte al proceso de desarrollo de sistemas de radio-software, empleando plataformas y software abierto, compararlas y decidiendo por la metodología más adecuada en este tipo de proyectos (Gechman, 2019).

La selección de una metodología simple da como resultado la posibilidad de elegir alguna de las siguientes en

su orden de jerarquía: Scrum, Extreme Programming (XP), Dynamic System Development, Method (DSDM), Crystal, Adaptive Software Development (ASD) y Feature-Driven Development (FDD), que pueden ser integradas en cualquier empresa que tenga contemplado utilizar metodologías ágiles para el desarrollo de aplicaciones (Nuottila et al., 2016).

Con relación a este concepto Ribeiro y Domingues (2018) manifiestan que las metodologías de desarrollo de software han ido creciendo y sufriendo una maduración durante los últimos años con un resultado favorable en el sector privado, por el contrario, en el sector público esa evolución no ha sido tan clara. Sin embargo, para este estudio se obtuvo una buena aceptación por parte del personal de la institución, a pesar de que se identificaron algunas resistencias y desacuerdos con respecto a esta metodología ágil.

Por otro lado, Zumba (2018) presenta una revisión bibliográfica que demuestra cómo ha sido el proceso evolutivo de las metodologías y modelos de software dividiéndolo en dos grupos: las clásicas o iniciales y las metodologías ágiles. Las ágiles surgieron por la necesidad y demanda de los mercados con la expansión de los sistemas de información y el impulso que dio internet; los proyectos de software requerían ser muy dinámicos, de desarrollo acelerado y con la interacción recurrente del usuario dentro del proceso de construcción, este concepto se puede aplicar mejor para el sector privado, donde la satisfacción del cliente y la obtención de recursos son los principales ejes de motivación (Borja López, 2013). Por el contrario, las metodologías clásicas se caracterizan por tener un enfoque en secuencias, definido y rígido. Es importante mencionar que la Policía Nacional de Colombia cuenta con un procedimiento estandarizado para el desarrollo de software que, una vez comparado con las metodologías de desarrollo existentes, se asemeja al modelo en cascada (Montero Molina et al., 2018).

Dando alcance al objetivo de este artículo, se quiere dar a conocer la necesidad que existe de reimplementar o modernizar el código heredado que tiene una empresa, en relación con las cajas blancas, que son un tipo de pruebas que se le hacen al software sobre las funciones internas de los módulos. Garcés et al. (2018) señalan que las necesidades de modernizar el software heredado en las organizaciones implican grandes retos y proponen un enfoque de transformación de caja blanca para modificar la arquitectura de la aplicación, sin que esta pierda valor; el modelo propuesto facilita la confi-

guración de la arquitectura antes de realizar la transformación real de la aplicación, además, presenta un enfoque en técnicas de ingeniería basado en modelos (MDE), y fue aplicado a un plan piloto de desarrollo de la empresa colombiana Asesoftware. Como conclusión de esa investigación, se presenta un proceso genérico de caja blanca para reimplementar software heredado en tecnologías modernas, en la empresa Asesoftware, que cuenta con muchos años de experiencia en la migración manual de aplicaciones desarrolladas en lenguaje Oracle Forms y que son actualizadas en Java y .Net; el enfoque se puede adecuar para tener en cuenta la escalabilidad y viabilidad del proceso, ya que el metamodelo agnóstico de la tecnología sirve como pivote que facilita la transformación.

La búsqueda a profundidad de textos científicos, relacionados con el tema de investigación muestra, según Trías et al. (2015), que los Sistemas de Gestión de Contenidos (CMS) son plataformas robustas que sirven para almacenar gran contenido digital, gestionado desde las aplicaciones web. Además, proponen el uso de transformaciones automatizadas para generar el rediseño de sistemas de información a partir de los ya conocidos sistemas heredados, siguiendo una herradura de proceso que tiene como base tres procesos básicos: análisis de un sistema existente, transformación lógica y desarrollo de un nuevo sistema. El primer proceso se ubica en el extremo izquierdo de la herradura; el segundo, atraviesa la parte superior y el tercer proceso baja por el extremo derecho de la herradura.

La investigación presentada se centra en la etapa de ingeniería inversa del método propuesto, para migrar una aplicación web tradicional a una web basada en CMS; como primera parte, es preciso utilizar un lenguaje de modelado que permita definir los modelos específicos de la plataforma, que representan la sintaxis del código fuente en PHP, en la segunda parte, se define e implementa un conjunto de transformaciones M2M automatizadas, que permiten la generación de modelos independientes de plataforma.

Al hacer la comparación de metodologías de desarrollo Molina y Pedreira (2020) mencionan el auge que tiene el desarrollo de aplicaciones web, como consecuencia del avance de la tecnología; además de la dependencia de las empresas y de las personas del internet en un mundo globalizado. La aparición de nuevas metodologías de desarrollo se presenta, en parte, por las inquietudes de quienes interactúan con código y desarrollan

aplicaciones. En este artículo se destaca la iteración que se tiene con las metodologías ágiles, debido a su facilidad para solucionar los errores y advertencias generadas durante el proceso, el cambio en los requerimientos por parte de los clientes, los inconvenientes de seguridad y ciberseguridad, así como las dificultades en el diseño e implementación del sistema (Maida & Pacienza, 2015).

Algunas de las metodologías estudiadas son: hipermedia, orientada a objetos; metodología de gestión de relaciones (RMM); metodología de diseño de hipermedia; lenguaje de modelado web (WebML); modelo de diseño de hipermedia orientado a objetos (OOHDM); ingeniería web basada en Uml (UWE) y Hera.

Gholami et al. (2017) plantean los desafíos en la migración de sistemas heredados a la nube; ellos establecen diferentes puntos de vista y enfoques que deben ser tenidos en cuenta para la elaboración de un plan de migración de software. Adicionalmente, señalan algunos de los múltiples beneficios que tiene la arquitectura de servicios en la nube, tales como: minimizan los costos de infraestructura y mantenimiento, cuentan con servicio especializado y soporte, presentan un uso eficiente de los recursos, entre otros.

No solamente basta mencionar estas ventajas de la arquitectura en la nube, un mecanismo importante para que las empresas y organizaciones den el paso hacia la migración consiste en que las mismas empresas realicen sus propias investigaciones, descubran los desafíos y retos, evalúen los riesgos, para tener claridad a la hora de tomar decisiones (De Fuentes-Martínez & Hernández-López, 2020).

Asimismo, se deben mencionar las diferentes definiciones que algunos autores le han atribuido al software heredado, pues el hecho de que algunos sistemas de información estén desarrollados para entornos web y estén alojados en servidores dedicados con arquitectura basada en la nube, no implica que sea un software heredado, una de las principales características para determinar el uso del software que soporte y aplique la tecnología para la nube es la elasticidad (Abdellatif et al., 2021).

Los sistemas heredados, por lo general, son antiguos, y fueron creados y diseñados antes de tener el concepto de arquitectura en la nube; la elasticidad de estos sistemas está definida y limitada por los recursos proporcio-

dados por el servidor; por el contrario, la arquitectura de la nube cuenta con recursos de tendencia ilimitada, que se ajustan en escala vertical, de acuerdo a la demanda del servicio. El factor elasticidad es el más complejo de implementar en la nueva arquitectura, a la hora de migrar información (Punnoose & De, 2021).

La migración de sistemas de información se debe realizar de acuerdo a las capas del software; en general, cada sistema de información se compone de una capa de usuario, una capa de lógica del negocio y una capa de enlace de datos; de acuerdo a estas capas, existen diferentes opciones de migración a la nube que se denominan variantes de migración y se clasifican en cinco grupos. La primera variante consiste en migrar la capa de negocio, pues esta es compatible con los servicios de nube *IaaS Infrastructure as a Service*, este modelo sugiere mantener los datos en la organización. La segunda opción propone realizar reingeniería a través de un servicio en la nube *SaaS, Software as a Service*, para la capa de aplicación. En la tercera opción, se sugiere implementar la base de datos en un proveedor de almacenamiento a través del modelo *IaaS*, manteniendo la capa de lógica del negocio en la organización y los datos en la nube. En la cuarta opción se realiza un trabajo más complejo, pues implica no solo migrar el esquema de la DB, sino también los datos en un proveedor de almacenamiento y, finalmente, se propone migrar en un servicio *IaaS* mediante un esquema de prestación de servicios tanto para la capa de aplicación como para la capa de enlace de datos (Silva-López et al., 2015).

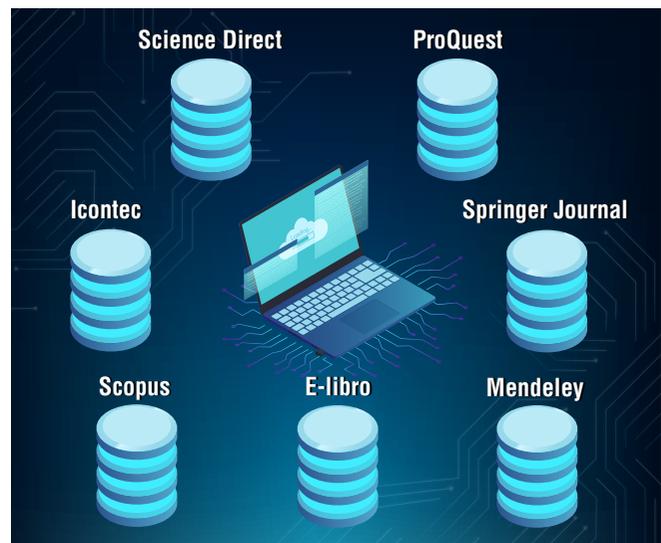
Por último, Berón et al. (2020) mencionan que, tanto el mantenimiento como la migración de sistemas de información son procesos cruciales en las organizaciones, dado el impacto que hoy en día tienen los sistemas y los datos en el *Core* del negocio, aunque muchas veces las motivaciones para migrar los sistemas de información heredados pasan por aspectos técnicos de arquitectura y obsolescencia tecnológica, los cuales son válidos. Poco se encuentra sobre la migración de un sistema heredado por causa de la calidad y la seguridad, ya que estos aspectos se pueden ajustar dentro de la arquitectura actual sin tener que desplegar procesos de migración.

partir de las siguientes categorías de análisis: planes de gestión para el rediseño de un sistema, migración de sistemas y metodologías de desarrollo, para esto, se utilizaron operadores booleanos que optimizaron las consultas realizadas en las bases de datos seleccionadas (Moher et al., 2009).

El análisis de los trabajos se realizó mediante la búsqueda sistemática de literatura en las bases de datos seleccionadas, con palabras clave, ecuaciones de búsqueda en inglés y en revistas de investigación reconocidas por la comunidad científica, con el propósito de obtener información sobre estudios de alto nivel, nuevos conceptos que permitan abrir la brecha del conocimiento actual y a su vez, observar otras opciones para el rediseño de un sistema de información con código heredado.

En la figura 3, se exteriorizan las bases de datos que se consultaron en la búsqueda de información relevante para la investigación.

Figura 3
Bases de datos

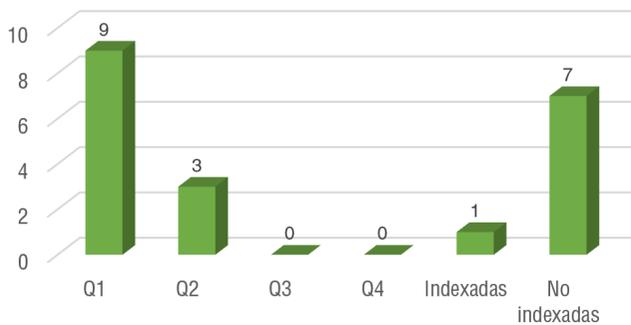


Asimismo, se realizó una búsqueda a profundidad en revistas de investigación, de las cuales se extrajeron artículos relacionados con migración, rediseño y reimplementación de sistemas heredados; en estos documentos se exponen arquitecturas y metodologías de desarrollo que deben ser estudiadas a profundidad, para tener la capacidad de escoger y proponer la más acertada para la investigación.

■ Método

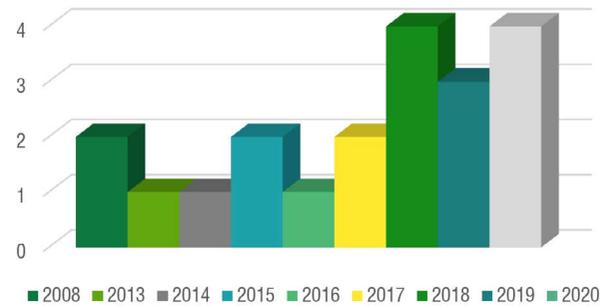
Se efectuó una revisión exploratoria sistemática, siguiendo el procedimiento indicado por el modelo PRISMA, a

Figura 4
Categoría de las revistas



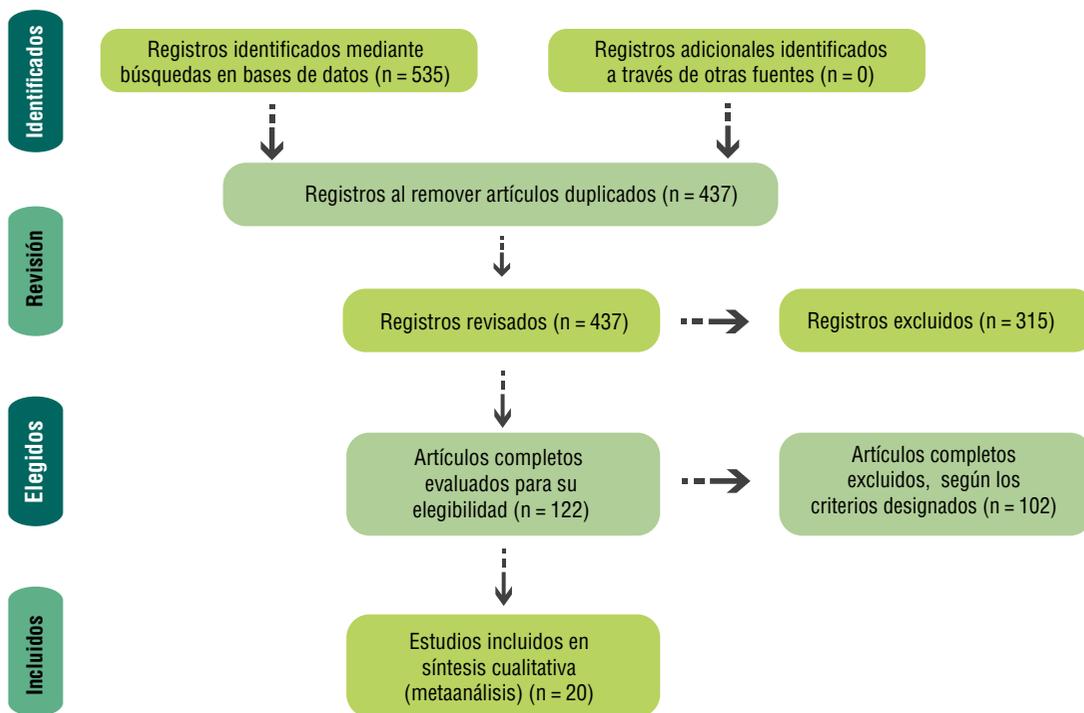
En las figuras 4 y 5 se especifica la categoría de las revistas consultadas, según su respectivo cuartil en la plataforma de Scimago, así como los años y la cantidad de publicaciones consultadas para artículos y tesis.

Figura 5
Año de publicación



En la figura 6 se describe el flujo de información, donde se muestran las fases en que se hizo la revisión sistemática, filtrando los documentos encontrados por los criterios de inclusión y exclusión, basados en la declaración PRISMA.

Figura 6
Diagrama de flujo PRISMA



Nota. Moher et al., 2009

- **Fase 1:** los documentos identificados se seleccionaron por medio de los métodos de búsqueda relacionados anteriormente, se aplicaron ecuaciones de búsqueda junto con las categorías de análisis escogidas y se eligieron los artículos que aplican a la investigación propuesta. Después, se creó una matriz en la herramienta Excel con los siguientes atributos: (a) nombre del

artículo, (b) autores, (c) tipo de documento, (d) resumen, (e) metodología, (f) categoría, (g) cuartil de la revista, (h) año de publicación, (i) país, (j) variable 1 arquitectura, (k) variable 2 área de aplicación.

- **Fase 2:** se seleccionaron los estudios que describen aspectos generales de rediseño, reimplementación o

migración de sistemas, con impacto favorable o desfavorable en la migración y, preferiblemente, de artículos que estuviesen en revistas indexadas.

- **Fase 3:** los documentos elegidos fueron evaluados con base en el tema de investigación, se filtraron los estudios de reimplementación, rediseño y migración de sistemas de información, arquitecturas de desarrollo y metodologías de implementación.
- **Fase 4:** se seleccionaron 20 artículos para hacer el análisis, según el problema planteado para esta investigación y así generar como resultado una propuesta argumentativa con la mejor opción para la migración del SIATH. Lo anterior, con el propósito de mitigar el riesgo de disponibilidad de la información para la Policía Nacional de Colombia; los datos obtenidos hacen parte de la revisión realizada a los documentos citados, y cumpliendo estrictamente con las fases nombradas anteriormente (Hernández et al., 1991).

Resultados

La Policía Nacional de Colombia identificó gracias a la Oficina de Control Interno, mediante una auditoría, la amenaza que existe en la disponibilidad de información gestionada desde el SIATH y con esta investigación se fundamentó la reimplementación del sistema. La revisión sistemática planteada para este análisis, analizó en profundidad 20 documentos de investigación (9 de ellos de Holanda), que aclararon dudas con respecto al tema de la migración o reimplementación de sistemas heredados. Asimismo, el estudio señala que las metodologías ágiles de desarrollo también son muy acogidas en la actualidad, ya que presentan un procedimiento más flexible para los desarrolladores, en el momento que necesiten hacer cambios a profundidad o ajustes al requerimiento inicial.

En el área de aplicación propuesta como segunda variable, se puede percibir que los autores, cuando proponen la reimplementación o migración de un sistema heredado, que es bastante complejo y robusto, recomiendan ir trabajando paralelamente la migración con el sistema en producción; igualmente dejan claro, que en el sector privado se hacen estos procesos con más frecuencia; pero a través de otro estudio, las personas también están dispuestas a aceptar la migración de su software heredado.

En la tabla 1 se describen algunas características de los documentos revisados, que se utilizaron en la matriz de

investigación para realizar el análisis sistemático de la literatura.

Tabla 1
Características descriptivas de los estudios de investigación

Autores	País de publicación	Idioma	Palabras clave
Yarza et al. (2021)	Holanda	Inglés	<i>Legacy software; Real-time systems; Retargeting; Time contract.</i>
Li et al. (2019)	Holanda	Inglés	<i>Dataflow; Deep learning; Hardware/software co-design; Low power techniques; Model-based design; Signal processing systems.</i>
Umar y Zordan (2009)	Holanda	Inglés	<i>Application integration; Application migration; Cost-benefit analysis; Enterprise architectures; Enterprise service bus (ESB); SOA patterns.</i>
Sneed y Verhoef (2019)	Holanda	Inglés	<i>Code refactoring; Code rewriting; Data renaming; Reverse engineering; Software migration; Software re-implementation.</i>
Woods y Bashroushb (2015)	Holanda	Inglés	<i>Architecture description language; Industrial experience report; Software architecture discovery.</i>
Stacey Raza (2015)	Ecuador	Español	
Chávez (2018)	Perú	Español	
Navarro et al. (2013)	Colombia	Español	<i>Adaptive software development; Agile methodologies; Crystal; Dynamic system development method; Feature-Driven Development; Scrum; XP.</i>
Garcés et al. (2018)	Holanda	Inglés	<i>.Net; Configuration; Industrial case study; Java; Model-driven engineering (MDE); Oracle forms; Quality attributes.</i>
Trías et al. (2015)	Holanda	Inglés	<i>Architecture-Driven Modernization; Content Management System; Model-driven Engineering; Reverse Engineering; Software Migration; Web application.</i>
Ribeiro y Domingues (2018)	Holanda	Inglés	<i>Agile Methodologies; Change Management; Project Management; Public Sector.</i>

Autores	País de publicación	Idioma	Palabras clave
Arifin y Tajudeen (2020)	Taiwán	Inglés	<i>HRMIS; Impact; Information systems; Malaysia; Military.</i>
Molina y Pedreira (2020)	Holanda	Inglés	<i>Development methodologies Web applications Comparison Agile methodologies</i>
Fahmideh et al. (2017)	Reino Unido	Inglés	<i>Cloud computing; Cloud migration; Cloud migration process; Legacy systems.</i>
Kargar et al. (2019)	Reino Unido	Inglés	<i>Clustering; Modularization; Multi-language software systems; Software comprehension; Software evolution; Source code.</i>
Canfora et al. (2008)	Holanda	Inglés	<i>Legacy system migration; Legacy system wrapping; Service Oriented Architecture; Web Service.</i>
Pérez (2017)	Ecuador	Español	
Berón et al. (2020)	Argentina	Español	Migración de sistemas de software; nube computacional.
Zumba (2018)	Ecuador	Español	Metodologías de desarrollo de software; metodologías ágiles; modelos
Nuottila et al. (2016)	Portugal	Inglés	<i>Adoption of agile methods; Agile barriers; Agile challenges; Agile methods; Agile project management; Software project management.</i>

Autores	Enfoque	Diseño/ Alcance	Estudio	Categorías de análisis
Stacey Raza (2015)	Cualitativo	Experimental/ exploratorio	Transversal	Metodologías de desarrollo
Chávez (2018)	Cualitativo	No experimental/ descriptivo	Transversal	
Navarro et al. (2013)	Cualitativo	Experimental/ descriptivo	Transversal	
Garcés et al. (2018)	Cualitativo	Experimental/ descriptivo	Transversal	
Trías et al. (2015)	Cualitativo	Experimental/ descriptivo	Transversal	
Ribeiro y Domingues (2018)	Cualitativo	No experimental/ descriptivo	Transversal	
Arifin y Tajudeen (2020)	Cualitativo	No experimental/ descriptivo	Transversal	
Molina y Pedreira (2020)	Mixto	Experimental/ exploratorio	Transversal	
Fahmideh et al. (2017)	Cualitativo	Experimental/ descriptivo	Transversal	
Kargar et al. (2019)	Cualitativo	Experimental/ descriptivo	Transversal	
Canfora et al. (2008)	Cualitativo	Experimental/ descriptivo	Transversal	Migración de Sistemas
Pérez (2017)	Cualitativo	Experimental/ descriptivo	Transversal	
Berón et al. (2020)	Cualitativo	No experimental/ descriptivo	Transversal	
Zumba (2018)	Cualitativo	No experimental/ descriptivo	Transversal	
Nuottila et al. (2016)	Cualitativo	No experimental/ descriptivo	Transversal	

Para referir el enfoque y diseño del documento analizado, se elaboró la tabla 2, en la cual se relacionaron directamente las categorías de análisis seleccionadas en la investigación.

Tabla 2
Características del enfoque y diseño del estudio de investigación

Autores	Enfoque	Diseño/ Alcance	Estudio	Categorías de análisis
Yarza et al. (2021)	Cualitativo	Experimental/ descriptivo	Transversal	Planes de gestión para el rediseño de un sistema
Li et al. (2019)	Cualitativo	Experimental/ descriptivo	Transversal	
Umar y Zordan (2009)	Cualitativo	No experimental/ exploratorio	Transversal	
Sneed y Verhoef (2019)	Cualitativo	Experimental/ descriptivo	Transversal	
Woods y Bashroushb (2015)	Cualitativo	Experimental/ descriptivo	Transversal	

Los resultados muestran una inclinación por las metodologías ágiles de desarrollo debido a la optimización de recursos materiales y talento humano; se pueden mencionar las siguientes metodologías como opcionales al momento de migrar o reimplementar un sistema heredado: Scrum, Extreme Programming [XP], Dynamic System Development Method [DSDM], Crystal, Adaptative Software Development [ASD], Feature-Driven Development [FDD]. Al tener clara la metodología, se debe escoger el proceso a realizar, que puede ser: convertir, reimplementar, redesarrollar, envolver, reemplazar o migrar; todo depende del estudio previo realizado por la empresa al software y a las necesidades presentadas.

Por último, es necesario reimplementar el SIATH, evitando al máximo la materialización de la amenaza en torno

a la disponibilidad de la información. Se recomienda la utilización de una metodología moderna, a través de una arquitectura de desarrollo actualizada, amigable al usuario y funcional, que garantice la gestión efectiva de la información en este sistema de información.

■ Discusión

La presente revisión sistemática tenía como objeto conocer las metodologías y arquitecturas de desarrollo utilizadas por diferentes autores de artículos de investigación, a fin de emplear las más apropiadas para realizar la migración de información del SIATH de la Policía Nacional. De acuerdo con la búsqueda realizada en las diferentes bases de datos de artículos científicos y el análisis de los mismos, los resultados señalaron que, preferiblemente, debe emplearse una metodología de desarrollo moderna, con procesos y procedimientos optimizados. De aquí la importancia de extraer, transformar y cargar los datos al sistema, para planear dinámicamente la ejecución de los objetivos administrativos y operativos de la institución, contribuyendo a brindar una solución a la problemática existente en la Policía Nacional de Colombia, para la gestión del talento humano a través del SIATH.

De acuerdo a los aspectos descriptivos, se observó que, en su gran mayoría el enfoque utilizado en la literatura científica analizada es el experimental/descriptivo, se dio mediante estudios cualitativos (diecinueve) y uno mixto; estos clasificados en tres categorías de análisis (planes de gestión para el rediseño de un sistema, metodologías de desarrollo y migración de sistemas). Estas características metodológicas fueron primordiales para la recolección de información que permitiera seleccionar el proceso de migración más factible para el SIATH. En este sentido, la migración de un sistema de información se hace necesaria teniendo clara la metodología de desarrollo viable para aplicar, debido a la existencia de metodologías tradicionales, modernas y ágiles; estas últimas, son utilizadas con mayor frecuencia, en virtud del corto tiempo para obtener buenos resultados; pero también es conveniente puntualizar que, para la reimplementación del sistema SIATH, no es viable una metodología ágil de desarrollo debido a la robustez y complejidad de esta aplicación.

En general, el análisis realizado a la literatura científica indica que tanto el mantenimiento como la migración de sistemas de información son procedimientos trascendentales en las organizaciones, dado el efecto que tienen los

sistemas y los datos en el *Core* de la organización (Berón et al., 2020). Por ello, en un sistema con más de veinte años, desarrollado en un lenguaje con limitaciones para su ejecución como el SIATH, es vital contar con referentes donde los autores puedan dar fe de los resultados obtenidos y de las prácticas de desarrollo llevadas a cabo, para evitar traumatismos en la migración de la aplicación, y permitiendo un mejor aprovechamiento de los recursos humanos, materiales y financieros con los que cuenta la Policía Nacional de Colombia.

■ Conclusiones

De los artículos revisados, se tomó como referente el escrito por Yarza et al. (2021), el cual describe la migración de software heredado basado en entornos de ejecución y aplicación a través de modelos; este método aplica para la reimplementación del SIATH, para lo cual es necesario realizar un consolidado de los formularios que existen actualmente, teniendo en cuenta las limitaciones que se puedan presentar por la cantidad de procedimientos y funciones que existen en el código de la aplicación y la interacción que se tiene con otros sistemas. Es necesario establecer un modelo que permita revisar la usabilidad de la aplicación y hacer una depuración en la capa de presentación del sistema.

La opción de migración, resultado de la investigación, es la reimplementación; es necesario revisar la arquitectura técnica, rediseñando los casos de uso y reemplazando el código fuente, conservando la arquitectura funcional y los objetos almacenados en la base de datos que son invocados desde la aplicación. Se recomienda el patrón arquitectónico en capas para la reimplementación del sistema a futuro, utilizando el modelo vista-controlador, separando los datos de la lógica del negocio y finalizando con el uso de la plataforma de desarrollo de software .NET, como la más recomendada según la investigación, para la reimplementación del SIATH.

Por lo tanto, se establece que, al ser reimplementado el SIATH en una arquitectura de desarrollo moderna, las capas de aplicación y presentación permitirán la interacción entre el usuario y el software a través de un entorno visual sencillo y claro, simplificando y optimizando el cumplimiento de las tareas asignadas al usuario funcional.

Durante el desarrollo de la reimplementación se pueden presentar limitaciones para la puesta en producción del sistema debido a su robustez y a la cantidad de nove-

dades administrativas que tiene la Policía Nacional de Colombia, que son gestionadas desde esta aplicación. No obstante, se puede optar por priorizar algunas novedades y por sacar a producción el software, reimplementándolo en paralelo con el sistema actual. A futuro, teniendo como base la presente investigación, es factible proponer la reimplementación de todos los sistemas de información desarrollados en software heredado, pertenecientes a esta institución, a fin de obtener aplicaciones amigables con el usuario, que permitan la escalabilidad y la mejora de su seguridad, para garantizar la disponibilidad, integridad y confidencialidad de la información.

■ Agradecimientos

A la Policía Nacional de Colombia, en particular la Oficina de Telemática, por proporcionar las herramientas y los espacios para el desarrollo de la presente investigación; asimismo, a la Maestría en Telemática de la Universidad Cooperativa de Colombia, por fomentar la investigación y el desarrollo en ciencia y tecnología, brindando sus conocimientos y valiosos comentarios en la construcción del manuscrito.

Por último, esta investigación fue sufragada con recursos propios. No contó con fuentes de financiación por parte de ninguna institución.

Conflicto de intereses: Los autores manifiestan no tener ningún conflicto de intereses.

■ Referencias

- Abdellatif, M., Shatnawi, A., Mili, H., Moha, N., Bousaidi, G. El, Hecht, G., Privat, J., & Guéhéneuc, Y. G. (2021). A taxonomy of service identification approaches for legacy software systems modernization. *Journal of Systems and Software*, 173, 110868. <https://doi.org/10.1016/j.jss.2020.110868>
- Arifin, M. A., & Tajudeen, F. P. (2020). Impact of human resources information systems in the military environment. *Asia Pacific Management Review*, 25(August 2013), 198–206. <https://doi.org/10.1016/j.apmr.2020.02.001>
- Balalaie, A., Heydarnoori, A., Jamshidi, P., Tamburri, D. A., & Lynn, T. (2018). Microservices migration patterns. *Software - Practice and Experience*. <https://doi.org/10.1002/spe.2608>
- Berón, M., Pérez, N., Riesco, D., Montejano, G., Pereira, M. J., Nováis, P., & Henriques, P. (2020). *Ingeniería de Software: Estrategias de Desarrollo, Mantenimiento y Migración de Sistemas en la Nube*. 504–508.
- Borja López, Y. (2013). Metodología Ágil de Desarrollo de Software – XP. *Espe Mevast*.
- Canfora, G., Fasolino, A. R., Frattolillo, G., & Tramontana, P. (2008). A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures. *Journal of Systems and Software*, 81(4), 463–480. <https://doi.org/10.1016/j.jss.2007.06.006>
- Castro-León, M. (2020). Arquitectura orientada a servicios, un enfoque basado en proyectos. *Enseñanza y Aprendizaje de Ingeniería de Computadores*. <https://doi.org/10.30827/digibug.32208>
- Chávez Sánchez, V. M. (2018). Rediseño e implementación de la capa de presentación de una aplicación software de ventas y facturación para garantizar mayor usabilidad. *Universidad Privada Del Norte*.
- Colombia, P. N. de. (2015). *Resolución 00090 DE 2015. Estructura OFITE (2).pdf*.
- De Fuentes Martínez, A., & Hernández López, M. S. (2020). Implementación del modelo de plataforma como servicio para el desarrollo de un sistema de producción colectiva de conocimiento en el ecosistema digital de Google. In *Prácticas docentes en los nuevos escenarios tecnológicos de aprendizaje*.
- Fernandez Garcia, J. (2011). Arquitectura en el Universo Online. *@tic. Revista d'innovació Educativa*. <https://doi.org/10.7203/attic.6.336>
- Flore, A., & Marx Gómez, J. (2020). Development and comparison of migration paths for smart grids using two case studies. *Heliyon*, 6(9), e04913. <https://doi.org/10.1016/j.heliyon.2020.e04913>
- Garcés, K., Casallas, R., Álvarez, C., Sandoval, E., Salamanca, A., Viera, F., Melo, F., & Soto, J. M. (2018). White-box modernization of legacy applications: The oracle forms case study. *Computer Standards and Interfaces*, 57(December 2016), 110–122. <https://doi.org/10.1016/j.csi.2017.10.004>
- García Cifuentes, J. P. (2016). Design Thinking para Innovar, desde la Interdisciplinariedad e Interculturalidad, en Ingeniería. *Encuentro Internacional de Educación En Ingeniería ACOFI*.
- Gechman, M. (2019). Software Development Methodologies. In *Project Management of Large Software-Intensive Systems*. <https://doi.org/10.1201/9780429027932-4>

- Gholami, M. F., Daneshgar, F., Beydoun, G., & Rabhi, F. (2017). Challenges in migrating legacy software systems to the cloud — an empirical study. In *Information Systems* Vol. 67, 100–113. <https://doi.org/10.1016/j.is.2017.03.008>
- Gonzalo Muñoz, V., Sobrino Callejo, M. R., Benítez Sastre, L., & Coronado Marín, A. (2017). Revisión sistemática sobre competencias en desarrollo sostenible en educación superior. *Revista Iberoamericana de Educación*. <https://doi.org/10.35362/rie730289>
- Guamán, D., Yaguachi, Lady, Cueva C, S., Jaramillo H, D., & Soto, F. (2018). Evaluación del rendimiento en el proceso de migración de una aplicación monolítica a microservicios. *CISTI 2018 - 13th Iberian Conference on Information Systems and Technologies*.
- Hamidian F, B. F., & Ospino S, G. R. (2015). ¿Por qué los sistemas de información son esenciales? *Anuario*, 38(2011), 161–183. Retrieved from <http://servicio.bc.uc.edu.ve/derecho/revista/idc38/art07.pdf>
- Hamman, J. J., Nijssen, B., Bohn, T. J., Gergel, D. R., & Mao, Y. (2018). The variable infiltration capacity model version 5 (VIC-5): Infrastructure improvements for new applications and reproducibility. *Geoscientific Model Development*. <https://doi.org/10.5194/gmd-11-3481-2018>
- Harris, S. L., & Harris, D. (2022). Hardware Description Languages. In S. L. Harris & D. Harris (Eds.), *Digital Design and Computer Architecture*. 170–235. <https://doi.org/10.1016/B978-0-12-820064-3.00004-0>
- Hernandez Sampieri, R., Fernandez Collado, C., & Baptista Lucio, M. del P. (1991). Definición del tipo de investigación a realizar: básicamente exploratoria, descriptiva, correlacional o explicativa. In *Metodología de la investigación*. https://www.uv.mx/personal/cbustamante/files/2011/06/Metodologia-de-la-Investigaci%C3%83%C2%B3n_Sampieri.pdf
- Hustad, E., & Olsen, D. H. (2021). Creating a sustainable digital infrastructure: The role of service-oriented architecture. *Procedia Computer Science*, 181, 597–604. <https://doi.org/10.1016/j.procs.2021.01.210>
- Kargar, M., Isazadeh, A., & Izadkhah, H. (2019). Multi-programming language software systems modularization. *Computers and Electrical Engineering*, 80. <https://doi.org/10.1016/j.compeleceng.2019.106500>
- Li, L., Sau, C., Fanni, T., Li, J., Viitanen, T., Christophe, F., Palumbo, F., Raffo, L., Huttunen, H., Takala, J., & Bhattacharyya, S. S. (2019). An integrated hardware/software design methodology for signal processing systems. *Journal of Systems Architecture*, 93(April 2018), 1–19. <https://doi.org/10.1016/j.sysarc.2018.12.010>
- Maida, E., & Pacienza, J. (2015). Metodologías de desarrollo de software. *Biblioteca Digital de La Universidad Católica Argentina*.
- Medina, J., Pineda, E., & Téllez, F. R. (2019). Requerimientos de Software: Prototipado, software heredado y análisis de documentos. *Revista Científica Ingeniería y Desarrollo*, 37, 25–36. Retrieved from <http://rcientificas.uninorte.edu.co/index.php/ingenieria/article/view/11452>
- Moher D, Liberati A, Tetzlaff J, A. D. (2009). PRISMA 2009 Flow Diagram. In *The PRISMA statement*.
- Molina, J., & Pedreira, N. (2020). Comparison of development methodologies in web applications. *Information and Software Technology*, 45(11), 699. <https://doi.org/10.1016/j.infsof.2019.106238>
- Monnier, M. (2016). Capítulo primero. In *Los Sistemas de Información Geográfica (SIG)*. 259–291. <https://doi.org/10.4000/books.ifea.585>
- Montero Molina, B., Ceballos Vite, H., & Dávila Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiral Revista Multidisciplinaria de Investigación*, 2(17), 114–121. Retrieved from https://www.researchgate.net/publication/327537074_Metodologias_agiles_frente_a_las_tradicionales_en_el_proceso_de_desarrollo_de_software
- Navarro, A., Fernández, J., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software A review of agile methodologies for software development. *Universidad Icesi*.
- Nuottila, J., Aaltonen, K., & Kujala, J. (2016). Challenges of adopting agile methods in a public organization. *International Journal of Information Systems and Project Management*, 4(3), 65–85. <https://doi.org/10.12821/ijispm040304>
- Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., & Goodfellow, I. J. (2018). Realistic evaluation of deep semi-supervised learning algorithms. *Advances in Neural Information Processing Systems*, 2018-Decem, 3235–3246. Neural information processing systems foundation.
- Pang, P. C. I., Chang, S., Verspoor, K., & Pearce, J. (2016). Designing health websites based on users' web-ba-

- sed information-seeking behaviors: A mixed-method observational study. *Journal of Medical Internet Research*, 18(6), 1–11. <https://doi.org/10.2196/jmir.5661>
- Peralta-Ascue, M., & Merma-Aroni, J. L. (2017). Integración de procesos de negocio aplicando la arquitectura orientada a servicios (SOA). *Interfases*. <https://doi.org/10.26439/interfases2017.n10.1771>
- Pérez, W. (2017). Propuesta de una metodología para el proceso de Migración de Datos en entornos empresariales. *Departamento de Ciencias de La Computación. Universidad de Las Fuerzas Armadas ESPE.*, 9.
- Punnoose, R., & De, S. K. (2021). Phase-wise migration of multiple legacy applications—A graph-theoretic approach. *Information and Software Technology*, 137, 106606. <https://doi.org/10.1016/j.infsof.2021.106606>
- Raffino, M. E. (2018). Lenguaje de Programación: Concepto, Tipos y Ejemplos. Retrieved from Concepto.de website: <https://es.scribd.com/document/437380386/Lenguaje-de-Programacion>
- Ribeiro, A., & Domingues, L. (2018). Acceptance of an agile methodology in the public sector. *Procedia Computer Science*, 138, 621–629. <https://doi.org/10.1016/j.procs.2018.10.083>
- Salazar, A. S., Gonnet, S., & Leone, H. (2014). Migración de Sistemas Heredados a Cloud Computing. *XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIO)-XV Simposio Argentino de Ingeniería de Software*.
- Senn, J. A. (2017). Análisis y diseño de sistemas de información. In M. H. Interamericana (Ed.), *BMC Public Health* (2nd ed.). México.
- Silva-López, R. B., Méndez-Gurrola, I. I., & Herrera Alcántara, O. (2015). Metamodelo de aprendizaje estratégico (MAE): Arquitectura de la capa de infraestructura, solución basada en la Cloud Computing. *Research in Computing Science*. <https://doi.org/10.13053/racs-93-1-15>
- Sneed, H., & Verhoef, C. (2019). Re-implementing a legacy system. *Journal of Systems and Software*, 155, 162–184. <https://doi.org/10.1016/j.jss.2019.05.012>
- Stacey Raza, P. (2015). “Estrategia para la migración y conversión de datos de los sistemas de software independientes de la Armada del Ecuador a S.A.P.” *Bio-mass Chem Eng*, 49(23–6), 1–15.
- Trias, F., De Castro, V., Lopez-Sanz, M., & Marcos, E. (2015). Migrating traditional web applications to CMS-based web applications. *Electronic Notes in Theoretical Computer Science*, 314, 23–44. <https://doi.org/10.1016/j.entcs.2015.05.003>
- Torres Silva, D., Ortiz Galván, J. D., Andrade Gómez, H., & Rivera López, R. (2018). Una Propuesta de Metodología para la Migración de Sistemas Heredados A proposal for a legacy systems migration methodology. *Programación Matemática y Software*, 10(November), 40–49. Retrieved from https://www.researchgate.net/publication/328792735_Una_Propuesta_de_Metodologia_para_la_Migracion_de_Sistemas_Heredados
- Trubiani, C., Bran, A., van Hoorn, A., Avritzer, A., & Knoche, H. (2018). Exploiting load testing and profiling for Performance Antipattern Detection. *Information and Software Technology*, 95, 329–345. <https://doi.org/10.1016/j.infsof.2017.11.016>
- Tsuji, S., Cristia, A., & Dupoux, E. (2021). SCALa: A blueprint for computational models of language acquisition in social context. *Cognition*, 104779. <https://doi.org/10.1016/j.cognition.2021.104779>
- Vidal Silva, C. L., Bustamante, M. A., Lapo, M. del C., & Núñez, M. de los Á. (2018). En la búsqueda de soluciones MapReduce modulares para el trabajo con BigData: Hadoop orientado a aspectos. *Información Tecnológica*, 29(2), 133–140. <https://doi.org/10.4067/s0718-07642018000200133>
- Woods, E., & Bashroush, R. (2015). Modelling large-scale information systems using ADLs - An industrial experience report. *Journal of Systems and Software*, 99, 97–108. <https://doi.org/10.1016/j.jss.2014.09.018>
- Yandrapally, R., Sridhara, G., & Sinha, S. (2015). Automated modularization of GUI test cases. *Proceedings - International Conference on Software Engineering*, 1, 44–54. <https://doi.org/10.1109/ICSE.2015.27>
- Yarza, I., Azkarate-askatsua, M., Onaindia, P., Grüttner, K., Ittershagen, P., & Nebel, W. (2021). Legacy software migration based on timing contract aware real-time execution environments. *Journal of Systems and Software*, 172, 110849. <https://doi.org/10.1016/j.jss.2020.110849>
- Yu, C., Li, Q., Liu, K., Chen, Y., & Wei, H. (2021). Industrial Design and Development Software System Architecture Based on Model-Based Systems Engineering and Cloud Computing. *Annual Reviews in Control*, 51, 401–423. <https://doi.org/10.1016/j.arcontrol.2021.04.011>
- Zumba, J. P. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, 3(10), 20–33. <https://doi.org/10.33890/innova.v3.n10.2018.651>