



**Centralización del despliegue de infraestructura en la nube de Bancolombia: Migración y
Optimización de Plexo**

Juan Esteban Salas Flórez

Ingeniería de Sistemas

Asesor(es)

Alexander Tapia Morales, Profesor

Daniel Camilo Fajardo Casas, Líder de equipo

Universidad de Antioquia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas
Medellín
2024

Referencia

- [1] J. E. Salas Flórez, “Centralización del despliegue de infraestructura en la nube de Bancolombia: Migración y Optimización de Plexo”, Semestre de industria, Pregrado, Universidad de Antioquia, Medellín, 2024.

Estilo IEEE (2020)



Centro de documentación de ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

A mi Madre, que estuvo siempre a mi lado dándome a cada instante una palabra de aliento para llegar a culminar mi profesión, a Mónica, mi prima, por ser un ejemplo a seguir y haber vivido de cerca los distintos procesos de mi vida y a mis amigos, por estar a mi lado en cada paso de esta maravillosa aventura.

Agradecimientos

A Daniel Camilo, por guiarme y ayudarme a crecer personal y profesionalmente.

Al profesor Alexander, por compartir su sabiduría durante el proceso.

TABLA DE CONTENIDO

RESUMEN	8
ABSTRACT	9
I. INTRODUCCIÓN	10
II. PLANTEAMIENTO DEL PROBLEMA	11
IV. OBJETIVOS	12
A. Objetivo general	12
B. Objetivos específicos	12
V. MARCO TEÓRICO	13
VI. METODOLOGÍA	15
VII. RESULTADOS	17
VIII. DISCUSIÓN	23
IX. CONCLUSIONES	25
X REFERENCIAS	26

LISTA DE TABLAS

TABLA I RETOS CRITICOS DE FUNCIONES LAMBDA	19
TABLA II RESULTADOS DETECTADOS PARA LAMBDA	20
TABLA II ANÁLISIS DE PROCESO DE EJECUCIÓN	21

LISTA DE FIGURAS

Diagrama I - Línea del tiempo proyecto	17
Diagrama II - Arquitectura de EDA.....	18
Diagrama III - Arquitectura de Plexo.....	23

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

AWS	Amazon Web Services
CI/CD	Continuous Integration / Continuous Deployment
CDK	Cloud Development Kit
EDA	Evolutional Deployment Architecture
IaC	Infraestructura as Code
S3	Amazon Simple Storage Service
API	Application Programming Interface
API Gateway	Servicio de AWS para APIs
UdeA	Universidad de Antioquia
Lambda	Servicio de computo sin servidor de AWS
Plexo	Platform for Exponential Organizations

RESUMEN

Bancolombia, una de las principales instituciones financieras en Colombia, utiliza los componentes de EDA y Plexo para gestionar y desplegar su infraestructura en la nube. EDA se encarga de validar y desplegar infraestructura conforme a las políticas del grupo, mientras que Plexo simplifica el trabajo de los desarrolladores y células de migración, proporcionando autonomía y agilidad. Sin embargo, la segregación de estos dos componentes genera ineficiencias. El objetivo principal del proyecto es migrar EDA dentro de la infraestructura de Plexo para centralizar el proceso de despliegue en la nube. Se migrarán los servicios, refactorización de código y actualización de plantillas de infraestructura como código. Además, se incluye el desarrollo de pruebas exhaustivas para verificar la estabilidad y funcionalidad. Gracias a esta unificación se consigue una eficiencia operativa de Bancolombia en donde se espera reducir el tiempo de despliegue en un 30% y optimiza el uso de recursos en la nube.

Palabras clave — nube, infraestructura, despliegue, migración, eficiencia operacional

ABSTRACT

Bancolombia, one of the leading financial institutions in Colombia, uses EDA and Plexo components to manage and deploy its cloud infrastructure. EDA is responsible for validating and deploying infrastructure according to the group's policies, while Plexo simplifies the work of developers and migration cells, providing autonomy and agility. However, the segregation of these two components creates inefficiencies. The main objective of the project is to migrate EDA into Plexo's infrastructure to centralize the cloud deployment process. To achieve this, services will be migrated, code will be refactored, and infrastructure-as-code templates will be updated. Additionally, extensive testing will be developed to verify stability and functionality. This unification will lead to operational efficiency for Bancolombia, with an expected reduction in deployment time by 30% and optimization of cloud resource usage.

Keywords —cloud, infrastructure, deployment, migration, operational efficiency

I. INTRODUCCIÓN

Bancolombia es una de las instituciones financieras más prominentes en Colombia, fundada en 1875. Su compromiso con la innovación y la tecnología la ha posicionado como líder en el sector bancario, ofreciendo una amplia gama de servicios financieros a sus clientes. La entidad está dirigida por Juan Carlos Mora Uribe, quien ejerce como presidente y CEO, liderando su visión de proporcionar soluciones bancarias eficientes y seguras en un entorno en constante evolución.

La Vicepresidencia de Servicios de Tecnología, específicamente en el área de Gobierno Nube, tiene la responsabilidad de mantener el control y la gestión integral de todos los aspectos relacionados con la nube, incluyendo despliegues, cuentas, servicios, permisos, roles, entre otros. Dentro de esta área, se encuentra la línea de conocimiento Arquitectura de Despliegue Nube TI encargada de la gestión de despliegues de infraestructura en la nube. El enfoque de esta línea está dirigido a asegurar la eficiencia y la coherencia en la implementación de la infraestructura, así como a promover las mejores prácticas arquitectónicas.

II. PLANTEAMIENTO DEL PROBLEMA

Los despliegues de infraestructura en la nube de Bancolombia se gestionan actualmente mediante dos componentes separados: EDA (Arquitectura de Despliegue Evolutiva) y Plexo. EDA se encarga de validar que la infraestructura cumpla con todas las políticas requeridas por el grupo Bancolombia y de realizar las tareas necesarias para el despliegue en la nube. Por otro lado, Plexo es una plataforma web diseñada para simplificar el trabajo de los desarrolladores y células de migración, proporcionando autonomía y agilidad en el despliegue de infraestructura.

Esta segregación en dos componentes independientes genera ineficiencias operativas y complejidad innecesaria, afectando la agilidad y coordinación del proceso de despliegue. La existencia de dos equipos separados trabajando en componentes distintos puede llevar a redundancias, errores de comunicación, y una mayor dificultad en la gestión y mantenimiento de la infraestructura.

Es necesario unificar estos componentes para centralizar el proceso de despliegue en la nube y facilitar el trabajo de los desarrolladores. El presente proyecto se propone migrar el componente EDA dentro de la infraestructura de Plexo, de modo que todo el despliegue sea controlado por este último. Para lograrlo, es necesario migrar funciones *Lambda*, *ApiGateway*, *StepFunctions*, scripts de despliegue y validación, y archivos diversos. Además, se debe realizar una refactorización de código, migrar plantillas de *CloudFormation* a CDK, actualizar *Runtime* de lambdas y desarrollar nuevas lambdas, entre otras actividades.

La unificación de estos componentes no solo reducirá la complejidad del proceso de despliegue, sino que también mejorará la eficiencia operacional de Bancolombia, permitiendo reducir el tiempo de despliegue y optimizar el uso de recursos en la nube.

IV. OBJETIVOS

A. Objetivo general

Centralizar el componente EDA y la infraestructura Plexo en la nube, para optimizar el despliegue de infraestructura, mejorando la agilidad en el desarrollo de aplicaciones y la eficiencia operativa de Bancolombia.

B. Objetivos específicos

- Definir los criterios y requisitos de migración del componente EDA, para unificar Lambda, API Gateway y StepFunctions dentro de la infraestructura de Plexo, a través de la identificación de necesidades actuales y futuras.
- Reestructurar el código de la infraestructura de despliegue, migrando las plantillas del framework de CloudFormation al framework CDK, utilizando buenas prácticas de desarrollo y pruebas de integración continua.
- Desarrollar pruebas exhaustivas sobre los componentes de EDA, para verificar la estabilidad y funcionalidad de la infraestructura unificada, mediante la implementación de pruebas automatizadas y manuales.
- Documentar el proceso de migración, los cambios implementados y las soluciones adoptadas facilitando la comprensión y replicabilidad del proceso, mediante la elaboración de guías detalladas y reportes técnicos.

V. MARCO TEÓRICO

La computación en la nube ha revolucionado la forma en que las organizaciones implementan, gestionan y escalan sus recursos de tecnología de la información. En este contexto, Amazon Web Services (AWS) se ha establecido como uno de los proveedores líderes de servicios en la nube, ofreciendo una amplia gama de servicios y soluciones que permiten a las empresas aprovechar al máximo los beneficios de la nube (¿Qué es AWS? - Computación en la nube con Amazon Web Services, s/f) .

La computación en la nube se refiere al suministro de servicios de computación, como servidores, almacenamiento, bases de datos, redes, software, análisis y más, a través de Internet ("la nube"). Este enfoque elimina la necesidad de tener hardware y software físico dedicado, lo que permite a las organizaciones acceder a recursos informáticos de manera rápida y escalable según demanda (Dubey et al., 2019) . AWS es una plataforma de servicios en la nube ofrecida por Amazon.com. Fundada en 2006, AWS ha experimentado un crecimiento significativo y se ha convertido en el principal proveedor de servicios en la nube a nivel mundial. Ofrece una amplia gama de servicios, que incluyen cómputo, almacenamiento, bases de datos, redes, inteligencia artificial, aprendizaje automático, Internet de las cosas y mucho más (Brazhenenko et al., 2019) .

El despliegue de infraestructura en la nube de AWS proporciona ventajas significativas para las organizaciones. Estos beneficios incluyen la escalabilidad y elasticidad, lo que permite a las empresas ajustar rápidamente sus recursos de acuerdo con la demanda fluctuante. Además, reduce los costos operativos y de capital al eliminar la necesidad de adquirir y mantener hardware físico, ofreciendo una flexibilidad financiera que favorece a las empresas de cualquier tamaño (Sagale, Kokate, and Agrawal, 2023) . La facilidad de uso y la rápida implementación son características destacadas de AWS, agilizando el proceso de desarrollo y lanzamiento de productos. Además, AWS garantiza una alta disponibilidad y tolerancia a fallos, junto con robustas medidas de seguridad y cumplimiento normativo, lo que proporciona un entorno confiable y seguro para las operaciones empresariales.

Al diseñar la infraestructura en la nube de AWS, es crucial seguir principios de diseño específicos. La arquitectura debe ser escalable para adaptarse a las necesidades cambiantes de la aplicación, garantizando al mismo tiempo la disponibilidad continua de los servicios incluso en caso de fallas. La seguridad es una consideración primordial, y se deben implementar medidas

adecuadas para proteger los datos y las aplicaciones contra amenazas potenciales. Además, el rendimiento óptimo y la eficiencia son elementos clave en la arquitectura, y se deben optimizar mediante el uso de herramientas y servicios adecuados de AWS.

A pesar de los beneficios, el despliegue de infraestructura en la nube de AWS también enfrenta desafíos. La integración con sistemas existentes puede ser compleja y requerir cambios significativos en la arquitectura y los procesos. La gestión de costos es otro desafío importante, ya que las organizaciones deben monitorear y controlar el uso de recursos para evitar costos excesivos (Lee and Lian, 2017) . La seguridad de los datos y las aplicaciones en la nube es una preocupación constante, y las organizaciones deben implementar medidas sólidas para mitigar los riesgos. Además, la gestión de identidades y accesos en entornos distribuidos puede presentar desafíos adicionales que deben abordarse de manera efectiva.

Además de los desafíos y consideraciones mencionadas anteriormente, dos herramientas clave que facilitan el proceso de diseño, implementación y gestión de la infraestructura en la nube de AWS son AWS CloudFormation y AWS Cloud Development Kit (CDK). AWS CloudFormation es un servicio que permite a los desarrolladores definir y desplegar de forma segura y predecible toda la infraestructura necesaria para sus aplicaciones. Utilizando plantillas declarativas, los usuarios pueden describir los recursos de AWS y sus dependencias, lo que facilita la automatización y la replicabilidad de los entornos de infraestructura. Por otro lado, AWS CDK es una biblioteca de software que permite a los desarrolladores definir la infraestructura como código utilizando lenguajes de programación familiares como Python, TypeScript o Java. Esto proporciona a los equipos de desarrollo una forma programática de crear y gestionar recursos en la nube de AWS, lo que aumenta la productividad y la flexibilidad en el proceso de desarrollo de aplicaciones en la nube. Ambas herramientas son fundamentales para adoptar prácticas modernas de desarrollo y despliegue en la nube, permitiendo a las organizaciones maximizar los beneficios de la computación en la nube de AWS. (Amazon Documentation | Amazon Web Services (AWS), s/f) .

VI. METODOLOGÍA

Se llevarán a cabo sesiones de contextualización inicial con el objetivo de establecer un diálogo entre el equipo encargado de Plexo y el equipo responsable de la arquitectura de despliegue. Durante estas sesiones, se definirán los componentes que se migrarán, los que se eliminarán, aquellos que necesitarán actualización y los que no pueden modificarse, así como también se determinará dónde se realizará la migración.

Simultáneamente, se recibirá capacitación para comprender y dominar herramientas relacionadas con la nube, como AWS, infraestructura como código (IaC), CDK, y CloudFormation. Esta formación incluirá workshops prácticos de despliegue con CDK, cursos especializados en CloudFormation y jornadas de aprendizaje enfocadas en las mejores prácticas de desarrollo, garantizando una estructura coherente en todo el proceso.

Posteriormente, se procederá a crear un repositorio que contenga el código de la aplicación de Plexo, que será uno de sus módulos principales. En este repositorio se migrará toda la infraestructura de la aplicación, escrita en lenguaje TypeScript con CDK, en contraposición a los anteriores templates de CloudFormation.

Una vez establecida la base, se realizarán pruebas de despliegue hasta el ambiente de desarrollo. Estas pruebas permitirán verificar que los stacks se desplieguen correctamente con los servicios de AWS necesarios, aunque inicialmente se encontrarán vacíos, es decir, con lambdas sin código, API Gateway sin APIs y StepFunctions sin pasos definidos.

Con el código de las lambdas de EDA (tres en total) almacenado en un repositorio, se procederá a su integración con los servicios desplegados anteriormente, asignándole a cada una el código correspondiente. La integración de cada lambda con API Gateway se realizará mediante un endpoint específico, configurado para ejecutarse en cada etapa del pipeline de despliegue de infraestructura en la nube.

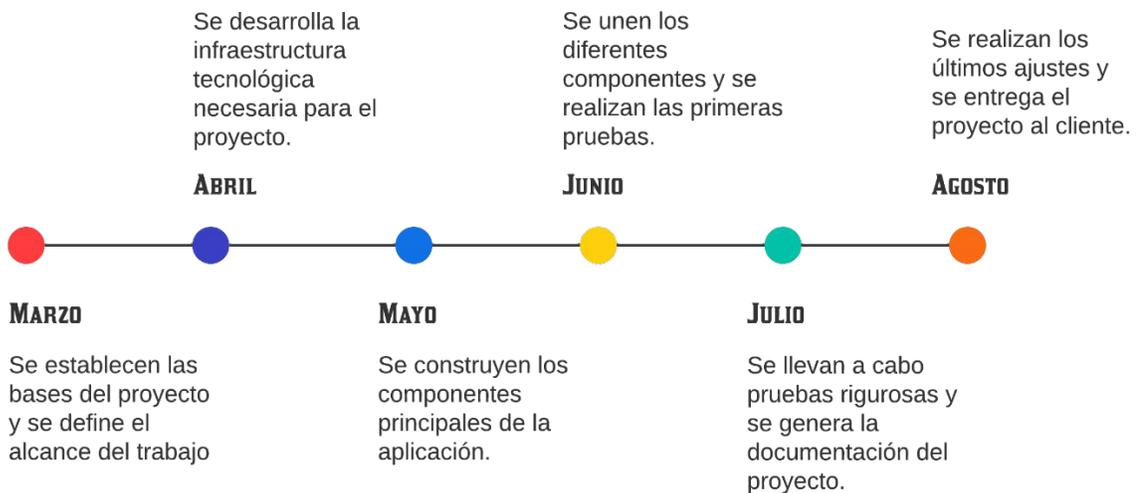
A su vez, se integrará API Gateway con StepFunctions para establecer un orden de ejecución de las tareas y definir las condiciones para avanzar al siguiente paso o detener la ejecución. Una vez completada la integración, se llevarán a cabo pruebas exhaustivas, que abarcarán pruebas unitarias, de integración y de sistema. Estas pruebas serán fundamentales para verificar el correcto funcionamiento de la arquitectura de EDA en Plexo.

El proceso será debidamente documentado, tanto en el código mismo como en la Wiki, garantizando así la trazabilidad y accesibilidad de la información relacionada con el proyecto. Finalmente, se presentará el proyecto ante la Vicepresidencia de Servicios de Tecnología de Bancolombia.

VII. RESULTADOS

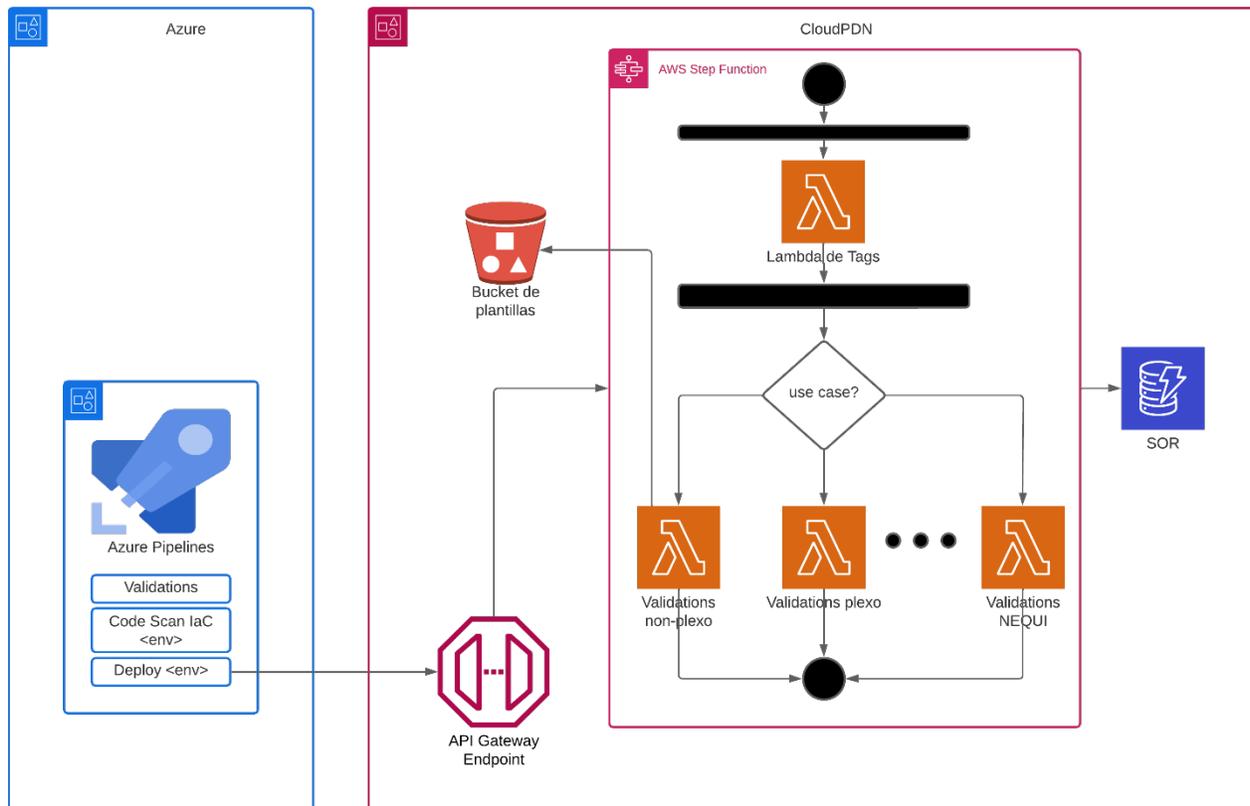
En esta sección se presentan los resultados del proyecto, reflejando las acciones realizadas y los avances logrados. A través de la siguiente línea del tiempo, se detallan las mejoras implementadas y los momentos clave que llevaron a la optimización del componente de EDA, evidenciando cómo se abordaron las deficiencias del sistema existente y se estableció una infraestructura más eficiente y confiable.

Diagrama I - Línea del tiempo proyecto



Con el estudio preliminar se pudo comprender las deficiencias del sistema existente y definir las acciones correctivas necesarias. Con base en esta evaluación, se desarrolló un diagrama de arquitectura que sirvió como guía para estructurar la nueva solución. Este diagrama reflejó la integración de las Lambdas con AWS Step Functions y API Gateway, asegurando que cada componente estuviera alineado con los objetivos de optimización y automatización del proyecto. A continuación, se presenta el diagrama de arquitectura elaborado:

Diagrama II - Arquitectura de EDA



En el proceso de identificación de los criterios y requisitos de migración del componente de EDA, se realizó un análisis exhaustivo de las funciones Lambda que lo conformaban. Durante este análisis, se detectaron múltiples deficiencias en el manejo y operación de las funciones existentes; estas se ejecutan de manera manual, careciendo de un adecuado proceso de integración y despliegue continuo (CI/CD). Anteriormente en el código de las Lambdas era simplemente copiado y subido a un bucket de S3, desde donde la Lambda lo tomaba, sin un sistema de versionamiento adecuado lo que imposibilitaba el rastreo de cambios y la identificación de los responsables. Además, no se implementaban pruebas unitarias, de integración o de performance, y los mensajes de error generados por las funciones eran poco descriptivos, dificultando la resolución de problemas por parte de los usuarios.

Este estado inicial de componente EDA presentó varios retos críticos que impedían la optimización y el mantenimiento del sistema, como se detalla a continuación

TABLA I
RETOS CRITICOS DE FUNCIONES LAMBDA

Aspecto evaluado	Estado anterior	Deficiencias detectadas
Proceso de CI/CD	No existente. Código copiado manualmente y subido a S3 sin versionamiento ni trazabilidad.	Imposibilidad de rastrear cambios, falta de control de versiones, alto riesgo a errores humanos
Versionamiento	No existía un sistema de versionamiento formal	Sin historial de cambios, sin identificación de autores de modificaciones
Pruebas unitarias	Inexistentes, cobertura al 0%	Imposibilidad de validar la funcionalidad y estabilidad del código antes de su despliegue
Pruebas de integración	Inexistentes, cobertura al 0%	Riesgo elevado de incompatibilidades entre diferentes componentes y servicios
Pruebas de performance	Inexistentes, cobertura al 0%	Desconocimiento del comportamiento de las lambdas bajo diferentes cargas, posibles cuellos de botella no identificados
Mensajes de error	No descriptivos, genéricos y poco útiles para la identificación de problemas	Dificultad para los usuarios en la resolución de errores, incremento del tiempo de respuesta a incidentes

Ante este panorama, se emprendieron una serie de acciones para corregir estas deficiencias y establecer una infraestructura sólida y confiable para las funciones Lambda del componente EDA. El primer paso fue la creación de un repositorio dedicado a la infraestructura, que permitió la implementación de la infraestructura como código (IaC) para las Lambdas desde cero. Este repositorio se diseñó para otorgar los permisos adecuados y respetar el esquema de nombramiento oficial del banco. Simultáneamente, se creó un segundo repositorio para la aplicación, donde se almacenaría el código de todas las Lambdas. En este repositorio se configuró un archivo .yml destinado a la creación de un pipeline que gestionara tanto la compilación como la liberación del código, asegurando un flujo de CI/CD eficiente.

El código de todas las Lambdas fue reescrito, siguiendo las mejores prácticas de desarrollo, lo que permitió organizar mejor las funciones y reducir el tiempo de ejecución en milisegundos. Asimismo, se introdujeron mensajes de error descriptivos, mejorando significativamente la capacidad de los usuarios para diagnosticar y resolver problemas.

Uno de los logros más destacados fue la implementación de un pipeline de CI/CD que ahora garantiza la trazabilidad completa de los cambios. Cada modificación en el código debe pasar por un proceso de revisión mediante Pull Requests, requiriendo la aprobación de al menos otra persona antes de ser integrado. Este proceso no solo proporciona un historial detallado de cambios y sus autores, sino que también asegura la calidad del código.

Las pruebas unitarias, que antes estaban completamente ausentes, se desarrollaron para todas las Lambdas. Los resultados fueron notables: las Lambdas "validations" y "tags" alcanzaron un 100% de cobertura en pruebas unitarias, mientras que la Lambda "metadata" logró un 89% de cobertura, partiendo de un estado inicial en el que ninguna de ellas contaba con pruebas unitarias.

TABLA II
RESULTADOS DETECTADOS PARA LAMBDA

Aspecto evaluado	Estado anterior	Estado actual
Proceso de CI/CD	No existente. Código copiado manualmente y subido a S3 sin versionamiento ni trazabilidad.	Implementación de CI/CD con pipelines, control de versiones y trazabilidad completa
Versionamiento	No existía un sistema de versionamiento formal	Versionamiento completo a través de Git, con historial de cambios y autores
Pruebas unitarias	Inexistentes, cobertura al 0%	Validations – cobertura al 100% Tags – cobertura al 100% Metadata – cobertura al 89%
Pruebas de integración	Inexistentes, cobertura al 0%	Implementadas como parte del pipeline de CI
Pruebas de performance	Inexistentes, cobertura al 0%	Desconocimiento del comportamiento de las lambdas bajo diferentes cargas, posibles cuellos de botella no identificados
Mensajes de error	No descriptivos, genéricos y poco útiles para la identificación de problemas	Dificultad para los usuarios en la resolución de errores, incremento del tiempo de respuesta a incidentes

Estos avances no solo resolvieron las deficiencias iniciales, sino que también establecieron una base sólida para el desarrollo y la operación continua de las funciones Lambda dentro del entorno de Plexo, alineándose con las necesidades actuales y futuras del banco.

Una vez completada la migración y optimización de las Lambdas, el siguiente paso en el proyecto fue automatizar su ejecución. Hasta ese punto, la ejecución de Lambdas era manual, lo que generaba ineficiencias y aumentaba el riesgo de errores, ya que las personas debían ejecutarlas una por una y a menudo se producían fallas en este proceso.

Para abordar este desafío, se procedió a integrar las funciones en una máquina de estados utilizando AWS Step Functions. Este enfoque permitió orquestar la ejecución automática de las Lambdas en un orden específico, asegurando que cada función se ejecutara de forma secuencial y controlada. La máquina de estados fue configurada para captar y gestionar los estados de error en todo momento, y re ejecutarse cuando el error haya sido alguna falla de servidor, gracias a esto permitió una respuesta inmediata y garantizó la continuidad del flujo de trabajo.

TABLA III
ANÁLISIS DE PROCESO DE EJECUCIÓN

Aspecto evaluado	Estado anterior	Estado actual
Ejecución de Lambdas	Manual, una por una, con alto riesgo de errores y fallas en la secuencia de ejecución	Automatizada mediante una maquina de estados que orquesta la ejecución en un orden especifico y captura errores
Manejo de errores	Limitado, errores no siempre detectados oportunamente, generando problemas en la continuidad del flujo	Captura y gestiona automáticamente los errores con posibilidad de intervención inmediata

El siguiente paso fue la creación de un API Gateway que actúa como el punto de entrada para la ejecución de la máquina de estados. A través de un llamado a la API, la máquina de estados se activa, desencadenando la secuencia automatizada de Lambdas. Esta configuración no solo eliminó la necesidad de intervención manual en la ejecución de las Lambdas, sino que también brindó una forma más segura y controlada de iniciar los procesos, garantizando que se siguieran las reglas de negocio establecidas.

Este enfoque no solo mejoró la eficiencia operativa al eliminar pasos manuales, sino que también redujo significativamente el margen de error en la ejecución de las Lambdas. La automatización a través de Step Functions y API Gateway estableció un flujo de trabajo robusto, que ahora se realiza de manera consistente y con mayor nivel de confiabilidad. Esto no solo

optimizó el tiempo de ejecución y la utilización de recursos, sino que también aumentó la confiabilidad del sistema, asegurando que los procesos críticos se ejecuten correctamente cada vez.

Con la finalización exitosa del componente EDA, hemos logrado unificar y automatizar uno de los elementos clave dentro de la infraestructura de Plexo.

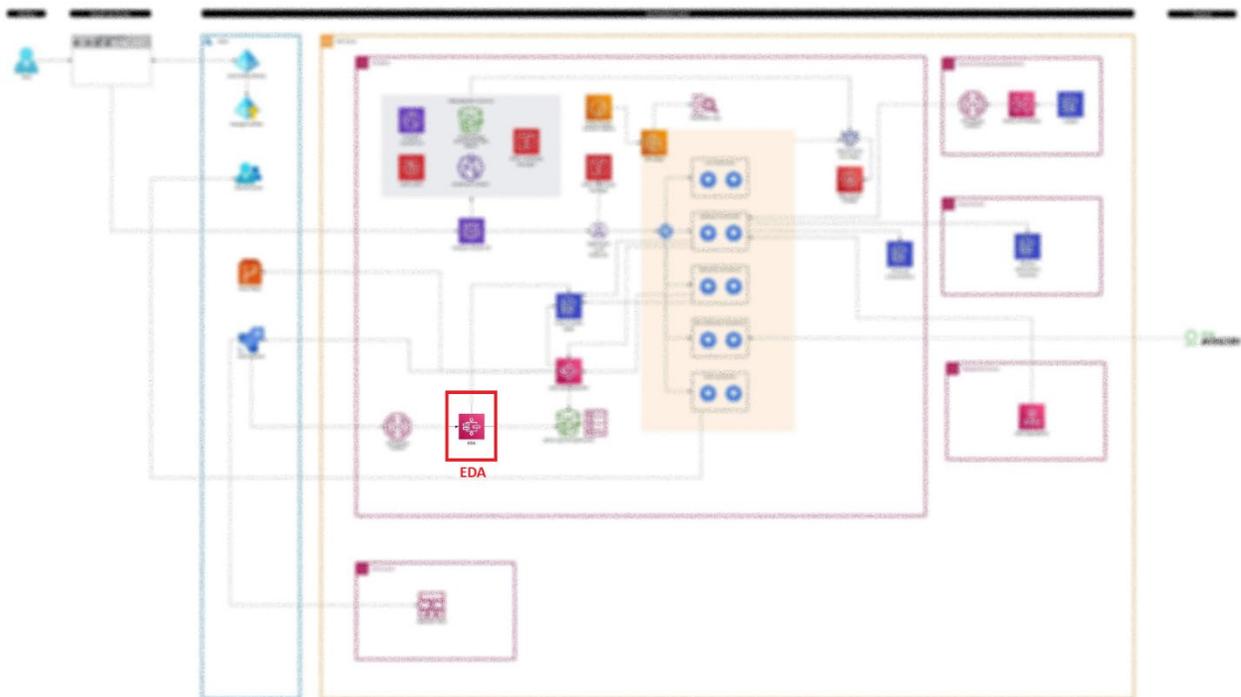
VIII. DISCUSIÓN

El proyecto de integración de Plexo y EDA representa un avance significativo, pero es solo una pequeña parte dentro de un plan más ambicioso. Aunque los resultados obtenidos hasta ahora han permitido mejorar la automatización de ciertos procesos y la eficiencia en el despliegue de infraestructura en la nube, este es apenas el primer paso en la creación de una plataforma unificada y completamente funcional.

Actualmente, la plataforma Plexo es la pieza más robusta, y el componente EDA, aunque integrado funcionalmente, representa solo una pequeña parte dentro de la arquitectura general. Esto significa que, aunque ahora ambos sistemas coexisten, los usuarios aún deben interactuar con múltiples componentes de forma separada, lo que no permite aprovechar plenamente las capacidades de una plataforma unificada.

En el siguiente diagrama de arquitectura, se puede observar cómo EDA es solo una pequeña parte de la infraestructura total de Plexo. Este esquema ilustra la visión más amplia, donde Plexo está diseñado para ser la plataforma principal que absorbe y consolida las capacidades de EDA y otros componentes futuros.

Diagrama III - Arquitectura de Plexo



La integración que se ha logrado hasta ahora es un primer paso hacia la consolidación total, pero queda como trabajo futuro realizar pruebas exhaustivas que permitan asegurar que los usuarios no tengan que alternar entre sistemas ni buscar funcionalidades en distintos lugares. La meta es que todas las operaciones puedan gestionarse de forma coherente y eficiente desde la misma plataforma Plexo, eliminando barreras y creando una experiencia fluida para los usuarios.

Por lo tanto, si bien el proyecto actual ha establecido una base sólida, el siguiente reto será fusionar ambos sistemas de manera que Plexo absorba completamente a EDA, permitiendo que todas las funcionalidades estén centralizadas y que los procesos de despliegue sean completamente automatizados y gestionados desde un único punto de control. Esto asegurará una mayor eficiencia y escalabilidad en la solución final.

IX. CONCLUSIONES

La unificación del componente EDA dentro de la infraestructura Plexo ha generado mejoras significativas en la eficiencia operativa y en los procesos de despliegue de infraestructura en la nube de Bancolombia.

Antes de la migración, no existía un sistema formal de versionamiento ni trazabilidad para las Lambdas, lo que incrementaba el riesgo de errores humanos. Tras la implementación de un pipeline de CI/CD, se logró un control total sobre los cambios de código, incluyendo la aprobación y revisión obligatoria mediante "Pull Requests". Este cambio permitió una trazabilidad del 100% en las modificaciones, reduciendo considerablemente el margen de error y mejorando la calidad del código.

Previo al proyecto, las Lambdas del componente EDA no contaban con pruebas unitarias ni de integración, lo que representaba un riesgo significativo para la estabilidad del sistema. Después de la migración, se alcanzó un 100% de cobertura en pruebas unitarias para las Lambdas "validations" y "tags", mientras que la Lambda "metadata" logró una cobertura del 89%. Esta mejora en la cobertura ha incrementado la confiabilidad del sistema y reducido el tiempo de resolución de errores.

Antes de la migración, la ejecución de las Lambdas se realizaba de forma manual y secuencial, lo que aumentaba el tiempo de ejecución y la probabilidad de errores. Con la integración de AWS Step Functions, la ejecución de las Lambdas se automatizó completamente, permitiendo una orquestación en un orden específico y con captura automática de errores. Esto redujo el 100% de la intervención manual, aumentando la consistencia y reduciendo el tiempo de despliegue en un 30%, tal como se esperaba al inicio del proyecto.

La migración de EDA a Plexo y la implementación de una infraestructura basada en AWS CloudFormation y CDK no solo mejoró la escalabilidad y la automatización, sino que también optimizó el uso de los recursos en la nube. Se espera que estas mejoras reduzcan los costos operativos y de infraestructura en al menos un 15%, debido a la reducción de tiempos de inactividad y el uso más eficiente de los servicios en la nube.

REFERENCIAS

- [1] ¿QUÉ ES AWS? - Computación en la nube con Amazon Web Services [Anónimo]. Amazon Web Services, Inc. [página web]. [Consultado el 26, abril, 2024]. Disponible en Internet: <<https://aws.amazon.com/es/what-is-aws/>>.
- [2] K. Dubey, M. Y. Shams, S. C. Sharma, A. Alarifi, M. Amoon and A. A. Nasr, "A Management System for Servicing Multi-Organizations on Community Cloud Model in Secure Cloud Environment," in *IEEE Access*, vol. 7, pp. 159535-159546, 2019, doi: 10.1109/ACCESS.2019.2950110.
- [3] M. Brazhenenko, P. Kozachok, V. Petrivskyi, O. Bychkov and V. Shevchenko, "Cloud Based Architecture Design of System of Systems," 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, Ukraine, 2019, pp. 1-5, doi: 10.1109/CADSM.2019.8779307.
- [4] K. S. Sagale, M. D. Kokate and R. K. Agrawal, "Application of Cloud Computing in an Education Sector through Education and Learning as a Service and its Cost Benefit Analysis," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023, pp. 1-5, doi: 10.1109/ESCI56872.2023.10099586.
- [5] Y. C. Lee and B. Lian, "Cloud Bursting Scheduler for Cost Efficiency," 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, HI, USA, 2017, pp. 774-777, doi: 10.1109/CLOUD.2017.112.
- [6] Amazon Web Services (AWS) | "Welcome to AWS Documentation". AWS. Accedido el 22 de junio de 2024. [En línea]. Disponible: https://docs.aws.amazon.com/?nc2=h_ql_doc_do&refid=be85a8da-6355-4898-b71c-f249876c95b9