



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Implementación de arquitectura serverless en AWS
para edición de imágenes para campañas de ventas**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

José David SÁNCHEZ TELLO

ASESOR

Mg. John Ledgard TRUJILLO TREJO

Lima, Perú

2026



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Sánchez, J. (2026). *Implementación de arquitectura serverless en AWS para edición de imágenes para campañas de ventas*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	José David Sánchez Tello
Tipo de documento de identidad	DNI
Número de documento de identidad	77177052
URL de ORCID	https://orcid.org/0000-0002-5987-4920
Datos de asesor	
Nombres y apellidos	John Ledgard Trujillo Trejo
Tipo de documento de identidad	DNI
Número de documento de identidad	06187585
URL de ORCID	https://orcid.org/0000-0002-0563-4809
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Víctor Hugo Bustamante Olivera
Tipo de documento	DNI
Número de documento de identidad	25655590
Miembro del jurado 1	
Nombres y apellidos	Cesar Alberto Molina Neyra
Tipo de documento	DNI
Número de documento de identidad	40553679
Datos de investigación	
Línea de investigación	No aplica

Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima
	Distrito: Cercado de Lima Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2025
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 14:00 horas del día 31 de marzo del año 2026, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Bustamante Olivera Víctor Hugo (Presidente), Mg. Molina Neyra Cesar Alberto (Miembro) y el Mg. Trujillo Trejo John Ledgard (Miembro Asesor), usando la plataforma Meet (<http://meet.google.com/wcc-jfha-tms>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“IMPLEMENTACIÓN DE ARQUITECTURA SERVERLESS EN AWS PARA EDICIÓN DE IMÁGENES PARA CAMPAÑAS DE VENTAS”**, por el Bachiller **Sánchez Tello José David**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiendo efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **18 (DIECIOCHO)**.

A continuación el Presidente de Jurados el Mg. Bustamante Olivera Víctor Hugo, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 14:51 horas, se levantó la sesión.


Presidente

Mg. Bustamante Olivera Víctor Hugo



Miembro

Mg. Molina Neyra Cesar Alberto



Miembro Asesor

Mg. Trujillo Trejo John Ledgard



CERTIFICADO DE SIMILITUD

Yo, John Ledgard Trujillo Trejo en mi condición de asesor acreditado mediante Resolución Decanal N° 001268-2025-D-FISI/UNMSM y OFICIO N° 0034-PTTSP-FISI-2025 del Trabajo de Suficiencia Profesional, cuyo título es Implementación de Arquitectura serverless en AWS para Edición de Imágenes para Campañas de Ventas, presentado por el bachiller Sánchez Tello José David para optar el Título Profesional de Ingeniero de Sistemas, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 10% de similitud, nivel PERMITIDO para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del **TITULO PROFESIONAL DE INGENIERO DE SISTEMAS**.

Lima, 4 de junio de 2026

Firma del Asesor:

Nombres y Apellidos del asesor: John Ledgard Trujillo Trejo

DNI: 06187585

DEDICATORIA

Quiero dedicar el presente trabajo principalmente a mis padres, les dedico cómo un agradecimiento sincero sobre todo el esfuerzo que hicieron para brindarme una educación de calidad y asegurar que lograra mis metas.

A Hernán, mi padre, que siempre motivó a estudiar la carrera que yo quisiera y que con su esfuerzo me ha sabido llevar siempre por el camino correcto.

A Rocío, mi madre, por su amor y apoyo incondicional que siempre me inspiró a cada día ser mejor persona y profesional.

Y, por último, pero no menos importante, a mi prometida Kelly que siempre ha creído en mí y me ha apoyado en todo momento.

AGRADECIMIENTO

Agradezco a mi asesor por su guía en la elaboración de presente trabajo.

A los docentes de la Facultad de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos por sus enseñanzas que me convirtieron en un profesional de éxito.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**Implementación de Arquitectura serverless en AWS para Edición de Imágenes
para Campañas de Ventas**

Autor: Sánchez Tello, José David

Asesor: Trujillo Trejo, John Ledgard

Título: Trabajo de Suficiencia Profesional para optar el Título Profesional de Ingeniero de Sistema

Fecha: Diciembre de 2025

RESUMEN

El presente trabajo describe la implementación de una arquitectura serverless en la nube de AWS para edición de imagen de una forma automatizada para Campañas de Ventas para una empresa que produce y vende artículos de belleza, anteriormente el equipo debía contratar un equipo de edición que se encargaban de realizar estos diseños de forma manual, lo cual demoraba mucho tiempo en el diseño y limitaba al equipo generando un cuello de botella, dificultando así empezar las campañas promocionales además de tener un costo asociado por el equipo de edición.

Al ser una empresa de artículos de belleza durante el transcurso del año se realizan distintas campañas promocionales, por lo que es completamente necesario realizar una actualización masiva de los catálogos de productos en bases a estas campañas, en especial en las ofertas y la edición del catálogo según las campañas, este proceso lento y costoso representa un riesgo considerable como errores en edición u ofertas, que podrían afectar a la empresa tanto económicamente como en su reputación.

Cómo solución de implemento la automatización de este proceso a través de una arquitectura serverless (sin servidor) utilizando los servicios de AWS que incluye en guardar las imágenes en un servicio de AWS para almacenar objetos llamado Bucket s3 y que funciones de AWS llamadas Lambdas que realicen este procedimiento en base a métricas mandadas por el equipo en formato de JavaScript Object Notation (JSON) a través del servicio de AWS llamado Api Gateway (que exporta una API), para guardarlo en una carpeta diferente del mismo Bucket s3 que el equipo tendría acceso

para usarlos en las campañas publicitarias de la empresa, el proceso se gestiona por el servicio de AWS llamado Step Function.

En los logros obtenido, se logró reducir los tiempos de la edición de estas imágenes, lo que podría ser un proceso que podría demorarle al equipo 3 días, se redujo a 2 horas y sin intervención de ningún editor durante la ejecución.

Palabras claves: Arquitectura serverless, AWS, campañas de ventas, catálogos, JSON, Lambda, Bucket S3, Cloud Computing, Step Function.

MAJOR NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF SYSTEMS AND INFORMATICS ENGINEERING
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING

Implementation of Serverless Architecture in AWS for Image Editing for Sales Campaigns

Author: Sánchez Tello, José David

Advisor: Trujilo Trejo, John Ledgard

Title: Professional Sufficiency Work to opt for the Professional Title of Systems Engineer

Date: December 2025

ABSTRACT

This paper describes the implementation of a serverless architecture in the AWS cloud for automated image editing for sales campaigns for a company that produces and sells beauty products. Previously, the team had to hire an editing team to do these designs manually, which took a long time and limited the team, creating a bottleneck and making it difficult to start promotional campaigns, in addition to the associated cost of the editing team.

As a beauty company, multiple promotional campaigns are carried out throughout the year, making it absolutely necessary to perform mass updates of the product catalogs based on these campaigns, especially in terms of offers and catalog editing according to the campaigns. This slow and costly process represents a considerable risk, such as editing or offer errors, which could affect the company both financially and in terms of its reputation.

As a solution, the automation was implemented through a serverless architecture using AWS services, the solution includes storing images in an AWS object storage services known as Amazon S3, and using Lambdas to perform this procedure based on metrics sent by the team in JSON format, to save it in a different folder in the same S3 bucket that the team would have access to for use in the company's advertising campaigns, this process is managed by AWS service called Step Function.

Regarding the achievements results, the time required to edit these images was reduced from a process that could take the team three days to just two hours, without the intervention of any editor during execution.

Keywords: Serverless architecture, AWS, sales campaigns, catalogs, JSON, Lambda, Bucket s3, Cloud computing, Step functions.

TABLA DE CONTENIDO

Contenido

INTRODUCCIÓN.....	1
CAPÍTULO I – TRAYECTORIA PROFESIONAL.....	3
CAPÍTULO II – CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA.....	10
2.1. EMPRESA – ACTIVIDAD QUE REALIZA	10
2.2. VISIÓN	10
2.3. MISIÓN.....	11
2.4. ORGANIZACIÓN DE LA EMPRESA.....	11
2.5. ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS	11
2.6. EXPERIENCIA PROFESIONAL REALIZADAS EN LA ORGANIZACIÓN	12
CAPÍTULO III – ACTIVIDADES DESARROLLADAS.....	14
3.1. Situación Problemática.....	14
3.1.1. Problema general:.....	14
3.1.2. Problemas específicos:.....	15
3.2. Solución.....	15
3.2.1. Objetivos.....	15
3.2.2. Alcance.....	15
3.2.2.1. Alcance funcional	15
3.2.2.2. Alcance técnico	16
3.2.2.3. Alcance Organizacional	16
3.2.3. Etapas y metodología.....	17
3.2.4. Fundamentos utilizados.....	21
3.2.5. Implementación de las áreas, procesos, sistemas y buenas practicas.....	25
Capítulo IV Reflexión crítica de la experiencia	40
Capitulo V conclusiones y recomendaciones.....	42
5.1. Conclusiones	42
5.2. Recomendaciones	43
5.3. Fuentes de información.....	44
5.4. Glosario.....	47

ÍNDICE DE TABLAS

Tabla 1	Experiencias Laborales	3
Tabla 2	Educación Académica	6
Tabla 3	Cursos Realizados	6
Tabla 4	certificaciones del autor	7
Tabla 5	Publicaciones realizadas	8
Tabla 6	Idiomas estudiados por el autor	8
Tabla 7	Stack de tecnologías del autor	8
Tabla 8	Plan de Trabajo	17
Tabla 9	Diferencia entre Testing Manual y automático	24
Tabla 10	Roles del proyecto	25
Tabla 11	Detalle de JSON de Escalas	32
Tabla 12	Detalle de JSON de Campaña	33
Tabla 13	costos por mes del personal a cargo del proyecto	37
Tabla 14	Calculo de gastos en infraestructura	38
Tabla 15	Tiempo de Trabajo en edición	38
Tabla 16	Evaluación de Costo/Tiempo	39
Tabla 17	Detalle de VAN y TIR	39

ÍNDICE DE FIGURAS

Figura 1	Organigrama general de la empresa	11
Figura 2	Área que pertenece el autor	11
Figura 3	Casos de Usos	18
Figura 4	Arquitectura de la solución	19
Figura 5	Integrando aplicaciones durante la modernización	22
Figura 6	Interfaz web de gestión de ejecuciones de ediciones	27
Figura 7	Interfaz web - logs	28
Figura 8	Imagen Simple 1	28
Figura 9	Imagen Simple 2	29
Figura 10	Imagen Compuesta	29
Figura 11	Imagen Compuesta 2	30
Figura 12	Archivo de escalas	32
Figura 13	Archivo de Campañas	33
Figura 14	Git Actions	35
Figura 15	New Relic	35
Figura 16	Interfaz de PagerDuty	36

INTRODUCCIÓN

El presente Trabajo de Suficiencia Profesional (TSP) tiene el fin de describir la experiencia que se obtuvo en el desarrollo e implementación de una solución tecnológica aplicada a una empresa del sector de belleza. La organización enfrenta la necesidad constante de generar y renovar material gráfico para las campañas de ventas, lo cual resulta ser un desafío por el tiempo limitado y los costos que representa la edición manual de imágenes. Bajo ese contexto, se plantea una propuesta orientada a superar dichos obstáculos mediante un sistema que permita automatizar parte del proceso, que no necesite ser gestionada y que garantice que los insumos publicitarios estén listos y con la calidad requerida.

La elaboración de este trabajo responde directamente a una problemática visible en muchas compañías que trabajan con campañas comerciales recurrentes. En fechas como Navidad, Día de la Madre, San Valentín o promociones especiales, la presión sobre los equipos de marketing y diseño es considerable, pues deben producir catálogos y anuncios en plazos muy cortos. Cuando la edición depende exclusivamente de un grupo especializado, se generan retrasos y cuellos de botella que afectan el inicio planeado de las campañas y, en consecuencia, el cumplimiento de los objetivos comerciales. Este trabajo busca dar respuesta a esa necesidad, ofreciendo una solución que agilice la edición de imágenes y reduzca la dependencia de procesos manuales.

La propuesta se centra, por tanto, en resolver de manera práctica los cuellos de botella que dificultan la productividad y el aprovechamiento de oportunidades de mercado. Para ello, se implementó un flujo de automatización que permite procesar imágenes en base a especificaciones predefinidas, cargadas en un formato estructurado y fáciles de gestionar por el equipo. Con este enfoque, el proyecto satisface la necesidad de la empresa de contar con material gráfico actualizado en cada campaña, asegurando tanto la calidad de las ediciones como su disponibilidad en plazos reducidos, aun cuando la frecuencia de las promociones sea elevada.

El desarrollo del trabajo implicó aplicar conocimientos adquiridos a lo largo de la carrera de Ingeniería de Sistemas, en particular en áreas vinculadas a la arquitectura de soluciones, automatización de procesos y trabajo colaborativo con equipos multidisciplinarios. De igual modo, se requirió coordinar de manera estrecha con los responsables de marketing y diseño para adaptar la solución a sus requerimientos específicos. Esta experiencia evidencia cómo la integración entre la tecnología y las necesidades de negocio puede aportar resultados concretos, optimizando el

tiempo de edición de imágenes y reduciendo significativamente los costos operativos.

Se pone en práctica el perfil profesional del ingeniero de sistemas mediante el diseño, desarrollo y gestión de sistemas de información, así como el soporte técnico especializado para el uso eficiente de los recursos de hardware y software y la administración de infraestructuras tecnológicas y redes de datos. Según la Universidad Nacional Mayor de San Marcos (s. f.), el egresado de Ingeniería de Sistemas cuenta con las competencias necesarias para supervisar y administrar sistemas y redes de información, contribuyendo a la operación eficiente de las organizaciones. Asimismo, se demuestra la importancia de proponer soluciones innovadoras en un entorno empresarial que demanda rapidez, flexibilidad y capacidad de adaptación frente a los cambios constantes del mercado. En este sentido, el proyecto no solo resuelve una necesidad inmediata de la organización, sino que también constituye un ejemplo de cómo aplicar la ingeniería de sistemas para generar ventajas competitivas sostenibles.

Finalmente, este trabajo de suficiencia profesional se presenta como una muestra del nivel de madurez alcanzado en la formación académica y profesional, y como evidencia de que los conocimientos teóricos, cuando se aplican a casos reales, pueden generar beneficios palpables en el desempeño organizacional. La solución desarrollada representa una herramienta clave para la empresa del sector belleza, pues le permite optimizar recursos, responder de manera más ágil a sus clientes y consolidar su posición en un mercado altamente dinámico.

Además, el presente trabajo de Suficiencia Profesional es una guía para que otros equipos puedan mejorar sus tiempos de ediciones automatizándolo mediante servicios serverless de AWS, se compone de manera estructurada 5 capítulos, que se detallan a continuación:

- En el CAPITULO I, se detalla la experiencia profesional del autor, donde se demuestra la experiencia lograda.
- En el CAPITULO II, se describe el contexto en donde se llevó el presente trabajo, detallando datos de la organización, así como las funciones desempeñadas por el autor en su rol dentro de la misma.
- En el CAPITULO III, se desarrolla la descripción del problema y su solución, detallando los objetivos, alcance, metodología e implementación de dicha solución, y finalmente la evaluación de la misma.
- En el CAPITULO IV, se expone la reflexión crítica del autor sobre la experiencia descrita.
- En el CAPITULO V, se detalla las conclusiones y recomendaciones del presente trabajo.

CAPÍTULO I – TRAYECTORIA PROFESIONAL

El autor se desempeña como arquitecto Cloud en AWS, con experiencia en el diseño y desarrollo de soluciones en la nube.

Actualmente desarrolla sus labores en Applying como Cloud Arquitecto de Soluciones Cloud de aplicaciones, encargado de monitorear, optimizar procesos.

Anteriormente el autor trabajó para empresas de diferentes rubros e industrias. Se detallan en la Tabla 1:

Tabla 1
Experiencias Laborales

Empresa/Periodo	Descripción de actividades realizadas
Applying Consulting SAC (Abril 2025 – Actualidad)	Cargo: Arquitecto de Soluciones Cloud Actividades: - Funciones de SRE de aplicaciones, asegurando la estabilidad de múltiples aplicaciones críticas en entornos de producción. - Diseño e implementación de arquitecturas de soluciones en entornos de Cloud (especialmente AWS). - Coordinación con el equipo Cloud para la implementación de infraestructuras seguras y escalables. - Colaboración con área de DevOps en la creación, optimización y migración de pipelines a Github Actions. - Planeación y gestión de costos asociados a la infraestructura Cloud. - Gestión de procesos de apagados y encendido programado de la infraestructura en base a la necesidad del negocio.

-
- Validación y aseguramiento de los pases a producción en coordinación con los equipos de desarrollo y Calidad QA.
 - Diseño y ejecución de pruebas de rendimiento (performance testing).
 - Participación en reuniones de Scrum (dailys, planificaciones, retrospectivas y pases a productivo).
 - Elaboración y mantenimiento de documentación técnica y de procesos.
-

Locorp Perú

(Marzo 2023 – Marzo 2025)

Cargo: Cloud DevOps and automation Specialist

Funciones:

- Funciones de SRE de aplicaciones, asegurando la estabilidad de múltiples aplicaciones críticas en entornos de producción.
 - Coordinación con IT Owners y equipos de desarrollo para alinear objetivos técnicos con los requerimientos del negocio.
 - Diseño de métricas de observabilidad mediante Grafana y New Relic, configurando monitoreo, métricas personalizadas y sintéticos de alertamiento para detectar incidentes.
 - Desarrollo e implementaciones de funciones serverless en AWS usando Node.js, Python y serverless Framework, con integración de Api Gateway y EvenBridge, para automatización de procesos y tareas repetitivas.
 - Gestión de Incidentes y análisis de Causa raíz (RCA), buscando una cultura de mejora continua.
-

Servex Perú (Junio 2022 – Febrero 2023)	Cargo: DESARROLLADOR JUNIOR NODEJS – REACTJS DE LA EMPRESA SERVEX Funciones: - Soporte al proyecto de Bot conversacional desarrollado con Node.js, brindando soporte y resolución de incidencias. - Implementar nuevas funcionalidades, entre ella la capacidad de escuchar y procesar audios de llamadas grabadas usando Node.js, Express y ReactJs. - Gestión de Base de datos relaciones y no relacionales, trabajando con SQL y mongo DB para consulta de información. - Desarrollo de microservicios, desplegados en contenedores Docker sobre sistemas Linux. - Levantamiento de entornos Linux, gestión básica de pipelines de entrega continua (CI/CD) y en despliegues. - Trabajo en equipo de forma colaborativa, dando soporte a la operación y usando metodologías ágiles para mejorar la eficiencia y calidad del servicio.
Dirección General de Posgrado - UNMSM (Marzo 2020 – Mayo 2022)	Cargo: DESARROLLADOR SEMISENIOR - PHP Funciones: - Actualización de páginas web con código PHP, mantenimiento de estas. - Escritura de código limpio, reutilizables y documentado siguiendo estándares de desarrollo y buenas prácticas de programación. - Desarrollo de nuevas aplicaciones web con PHP, JS, Bootstrap, MySQL.

-
- Realización de comunicación Front y Backend con Ajax Js.
 - Contribución a la seguridad de las aplicaciones, prácticas de protecciones contra ataques SQL, CSRF, entre otras vulnerabilidades.
-

Nota. Fuente: Elaboración Propia

La educación académica del autor se detalla en la Tabla 2:

Tabla 2
Educación Académica

Periodo	Educación Académica
2016-2021	Estudios de Pregrado de Ingeniería de Sistemas e Informática en la Universidad Nacional Mayor de San Marcos
2021	Bachiller en Ingeniería de Sistemas en la Universidad Nacional Mayor de San Marcos

Nota. Fuente: Elaboración Propia

Tabla 3
Cursos Realizados

Institución	Curso
--------------------	--------------

IBM	<i>Machine Learning with python</i>
CETI	Diplomado en Especialista en Desarrollo de aplicaciones con PHP 7.0, Laravel y MySQL
INAGEP	<i>Excel 2019 AVANZADO</i>
INAGEP	Diplomado Especializado en Gerencia de Proyectos bajo enfoque PMI (PMBOK 6ta edición)
Platzi	Serverless Framework
Platzi	Docker
Nota. Fuente: Elaboración Propia	

Tabla 4
certificaciones del autor

Periodo	Certificado	Educación Académica
2024	Azure – Azure Fundamentals (AZ- 900)	Azure
2023	AWS – Certified Cloud practitioner (CLF-C01)	AWS
2025	AWS – Certified Architec Asocciate	AWS

Nota. Fuente: Elaboración Propia

Tabla 5*Publicaciones realizadas*

Periodo	Articulo	Revista
Julio 19 – 21, 2023	IMPLEMENTATION OF A CHATBOT FOR VIRTUAL ATTENTION OF QUERIES CASE: POSTGRADUATE SCHOOL OF THE UNMSM	Latin American and Caribbean Consortium of Engineering Institutions - LACCEI

Nota. Fuente: Elaboración propia

Tabla 6

Idiomas estudiados por el autor

Periodo	Idioma	Nivel	Educación Académica
2018-2020	Ingles	C1	Asociación Cultural Peruano Británica

Nota. Fuente: Elaboración Propia

Tabla 7

Stack de tecnologías del autor

Tipo	Nombre
- Lenguaje de programación	- <i>PHP</i> - <i>JAVASCRIPT</i> - <i>PYTHON</i>
Servicios AWS	- Lambda

	- S3 - Ec2 - Api Gateway - Load Balancer - Dynamo DB - RDS - VPC - Cloufront - AutoScaling - Step Functions - ECS
Otros softwares	- <i>Github</i> - <i>Docker</i> - <i>New Relic</i> - <i>Grafana</i> - <i>Pagerduty</i>
Servicios Azure	- Virtual Machine - Azure Functions - Suscription - Bakups - SQL Azure - Cosmos DB

Nota. Fuente: Elaboración Propia

CAPÍTULO II – CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1. EMPRESA – ACTIVIDAD QUE REALIZA

La compañía especializada en cosmética y cuidado personal es una organización internacional con amplia trayectoria en el mercado. Su presencia se extiende a distintos países de América Latina y a otras regiones del mundo, consolidándose como una marca reconocida en el sector. Entre sus principales líneas destacan productos de maquillaje, fragancias y cuidado de la piel, siempre con una propuesta enfocada en la innovación y en atender las expectativas cambiantes de sus consumidores.

El modelo de negocio tiene como eje la venta directa, respaldada por una red de consultoras independientes que comercializan los productos a través de catálogos impresos y plataformas digitales. Con el avance tecnológico, la empresa ha integrado el comercio electrónico y espacios físicos de venta, lo que ha ampliado su alcance y fortalecido la conexión con sus clientes. Este sistema no solo impulsa las ventas, sino que también abre oportunidades de emprendimiento para miles de personas asociadas a la marca.

En materia de innovación, la compañía destina recursos a soluciones tecnológicas enfocadas en el manejo de datos, aplicaciones móviles y comercio digital, con el propósito de optimizar la experiencia de sus consumidores y de su fuerza de ventas. Gracias a esta estrategia, se ha consolidado como un referente en la industria cosmética internacional, respaldada por productos de calidad y campañas de marketing efectivas que sostienen su crecimiento.

2.2. VISIÓN

La visión de la empresa se proyecta hacia consolidarse como un referente en la industria de la belleza a nivel global, destacando por su capacidad de innovación, sostenibilidad y compromiso con las personas. La organización aspira a ser reconocida por generar valor tanto en el mercado como en la sociedad, ofreciendo productos y servicios que integren calidad, accesibilidad y responsabilidad social, de manera que impacten positivamente en la vida de millones de consumidores.

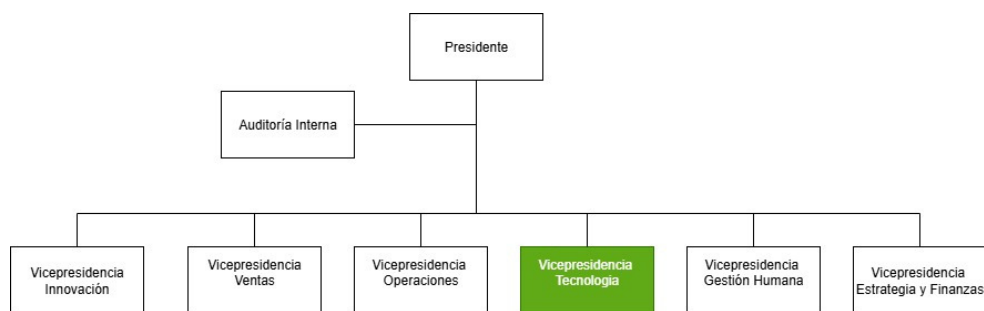
2.3. MISIÓN

La misión de la empresa de belleza consiste en ofrecer soluciones innovadoras en cuidado personal y cosmética, orientadas a mejorar la calidad de vida de sus clientes. La organización busca acompañar a las personas en su proceso de crecimiento y bienestar, entendiendo la belleza no solo como un atributo físico, sino también como una experiencia que fortalece la confianza, impulsa la autenticidad y contribuye al desarrollo integral de cada individuo.

2.4. ORGANIZACIÓN DE LA EMPRESA

Figura 1

Organigrama general de la empresa



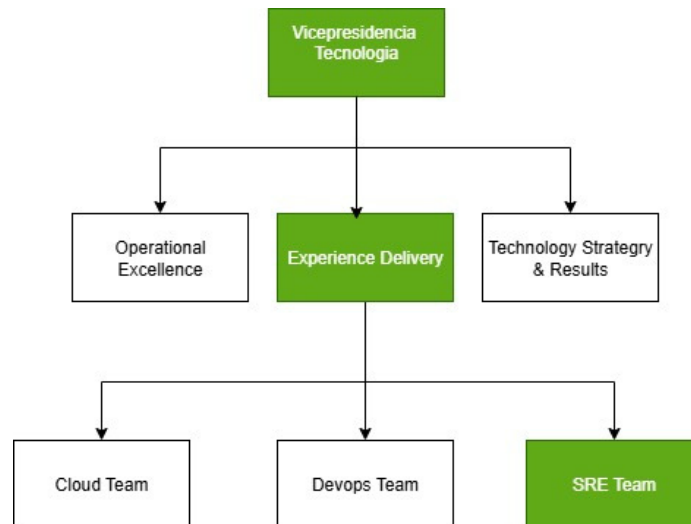
Nota. Fuente: elaboración propia

El área de la empresa encargada de gestionar todos los proyectos que implican cambios o gestión de la tecnología está bajo la vicepresidencia de tecnología.

2.5. ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS

Figura 2

Área que pertenece el autor



Nota. Fuente: elaboración propia

Mi participación dentro de la empresa en el rol de Site Reliability Engineering, en Experience Delivery dentro del subárea de SRE team, junto con el equipo, tuve el rol de mantener la ejecución de sistemas confiables en Producción haciendo uso tanto de la ingeniería de software así como buenas prácticas en infraestructura de AWS.

2.6. EXPERIENCIA PROFESIONAL REALIZADAS EN LA ORGANIZACIÓN

La experiencia del autor es de Site Reliability Engineer (SRE), las funciones que se tuvo fueron:

- Definición y mantenimiento de indicadores de nivel de servicio (SLI), establecimiento de objetivos de nivel de servicio (SLO) y realizando el cálculo de error, con seguimiento continuo para medir y mejorar la fiabilidad de los sistemas.
- Diseñar e implementar pipelines de observabilidad y monitoreo que agregan y procesan logs, métricas y trazas, facilitando la generación de reportes y alertas accionables para los equipos de la operación.
- Implementar soluciones de recopilación y almacenamiento de telemetría, como la centralización de registros (logs) y series temporales, para soportar el análisis

forense y el monitoreo en tiempo real de los sistemas, aplicando los principios de Site Reliability Engineering (SRE) descritos por Amazon Web Services (¿Qué es la ingeniería de fiabilidad del sitio (SRE)?, 2025)

- Identificación de tareas operativas manuales y creación de automatizaciones (scripts, jobs programados) para reducir trabajo repetitivo y minimizar errores humanos.
- Integración de prácticas de fiabilidad (SRE/DevOps) en el ciclo de vida del desarrollo: definición de runbooks, criterios de aceptación para resiliencia y revisiones de diseño orientadas a tolerancia a fallos.
- Participar en la gestión de incidentes mediante la coordinación de la respuesta ante fallos, la comunicación con los interesados (stakeholders), la documentación de los incidentes y la elaboración de informes posteriores a la incidencia (postmortem), conforme a las responsabilidades del enfoque Site Reliability Engineering (SRE) expuestas por Amazon Web Services Creación de documentos de análisis de causa raíz (RCA) y diseño de acciones de corrección y prevención para reducir la recurrencia de fallas. (¿Qué es la ingeniería de fiabilidad del sitio (SRE)?, 2025)
- Colaboración en planificación de capacidad y escalabilidad, realizando pruebas de carga y ajuste de configuraciones para garantizar desempeño ante crecimiento de la demanda.
- Optimización del rendimiento mediante ajustes de infraestructura, perfilado de aplicaciones y propuestas de arquitectura que mejoren latencias y rendimiento.
- Implementación y supervisión de procesos seguros de gestión del cambio (control de versiones, ventanas de despliegue y rollback), minimizando el riesgo durante actualizaciones de infraestructura y despliegues.
- Apoyo y mentoría a equipos de ingeniería en la adopción de buenas prácticas operacionales, creación de plantillas y guías para estandarizar procedimientos.
- Coordinación con áreas de producto y negocio para priorizar mejoras en fiabilidad y observar el impacto de las decisiones técnicas sobre los objetivos comerciales.

CAPÍTULO III – ACTIVIDADES DESARROLLADAS

3.1. Situación Problemática

Las empresas al día de hoy depende mucho de la velocidad de los procesos, en el contexto internacional, se enfrentan a importantes desafíos relacionados con los cuellos de botella en sus procesos de tecnologías de información y de negocio. La problemática es visible en áreas como marketing y publicidad digital, donde hay una necesidad de lanzar campañas comerciales de forma constante, como Navidad, San Valentín, Día de la Madre, Black Friday o Cyber Days, entre otros, lo cual exige la producción y edición constante y acelerada de gran cantidad de material gráfico y promocional.

Sin embargo, los tiempos de respuesta y calidad de ediciones pueden verse afectados por la dependencia de infraestructura ya sean ambientes locales (On-premises) o servidores en la nube. Ante ello, muchas empresas a nivel global están migrando hacia arquitecturas serverless, que les permiten tener una mayor escalabilidad, reducir gastos operativos y agilidad para responder a picos de demanda sin requerir mantener o gestionar constantemente servidores.

A nivel Nacional, las empresas peruanas también se están encontrando frente a una elevada presión competitiva que demanda ejecutar de forma constante campañas digitales. Sin embargo, editar imágenes genera grandes costos de tiempo, equipo (dinero) ya que se tiene una dependencia de un equipo especializado que impacta directamente en el inicio oportuno de las campañas promocionales.

En el ámbito local, particularmente en empresa con sede en Lima en sectores de cosméticos, retail y consumo masivo, la problemática se hace más evidentes. Ya que, estas organizaciones deben de adaptarse a un calendario comercial muy cargado con festividades, que demandan actualizaciones constantes de catálogos y anuncios. Este ritmo de trabajo genera sobrecarga y cuellos de botellas en la empresa, debido a las constantes coordinaciones entre equipos de diseño, marketing y TI. Como resultado, los tiempos de edición y distribución de contenido se prolongan y disminuyen la efectividad de las campañas lo que aumenta costos operativos.

3.1.1. Problema general:

En la empresa de Belleza, el proceso de edición de imágenes para

campañas publicitarias digitales afrontaba grandes tiempos y costos operativos debido a la dependencia de la edición de forma manual, y tener un equipo especializado para realizar estas ediciones. Esto ocasionaba retrasos y deficiencias al iniciar campañas de ventas.

3.1.2. Problemas específicos:

- **Excesivos Tiempos de edición:** El trabajo de edición manual con un equipo especializado tomaba varias horas o incluso días, además que no estaban disponibles en horario fuera de oficina.
- **Altos Costos operativos:** Mantener un equipo especializado a edición resulta costoso para la operación además de que trabajan principalmente por campañas.

3.2. Solución

3.2.1. Objetivos

3.2.1.1. Objetivo General

Implementar una solución serverless en AWS para edición automatizada de imágenes para las Campañas, aplicando la metodología de desarrollo de Kendall y Kendall (2023), con el fin de optimizar los tiempos y costos asociados de edición

3.2.1.2. Objetivos específicos

- Desarrollar una solución serverless que procese la edición de imágenes en base a parámetros definidos mediante archivos JSON.
- Integrar las automatizaciones con los entornos de almacenamiento utilizados por el equipo facilitando la carga y recuperación de las imágenes.

3.2.2. Alcance

3.2.2.1. Alcance funcional

El presente trabajo de suficiencia profesional, se limita al desarrollo e implementación de una solución automatizada para edición de imágenes usadas en campañas promocionales de una empresa que produce y vende artículos de belleza, el sistema permite que los

usuarios carguen las especificaciones de la edición en un formato JSON a través de una interfaz, junto con las imágenes que serán almacenadas en un Bucket de Amazon S3. Una vez que esté ejecutándose el proceso aplicará las ediciones definidas en el archivo JSON mediante funciones lambdas y orquestación con Step Functions, asegurando que las imágenes procesadas se guarden dentro del mismo bucket s3. El alcance incluye la automatización del flujo de edición y almacenamiento. Optimizando así tiempos y costos operativos. El presente trabajo no contempla el desarrollo de herramientas de diseño gráfico adicionales ni la gestión de catálogos externos.

3.2.2.2. Alcance técnico

La solución hace uso de servicios nativos de Amazon Web Services (AWS), entre ellos:

- Amazon S3 para la carga, almacenamiento y recuperación de imágenes.
- AWS Lambda para la ejecución automática de procesos de edición sin necesidad de servicios adicionales.
- Amazon SQS como sistema de colas para gestionar las solicitudes de procesamiento de imágenes.
- AWS Step Functions para orquestar las distintas funciones Lambdas involucradas en el flujo de edición.
- Api Gateway como punto de entrada para invocar el proceso.
- El archivo JSON cargado define los parámetros de edición (cómo redimensionamiento, compresión, cambio de formato).

El trabajo se limita al desarrollo de este flujo sin servidor (serverless) y no considera el desarrollo de entornos de diseño avanzados ni la integración con plataformas de terceros para publicación automática.

3.2.2.3. Alcance Organizacional

El sistema beneficiará principalmente al área de Marketing y al equipo de Diseño de la empresa de cosméticos, quienes podrán conseguir las imágenes editadas en menor tiempo y sin depender de procesos

manuales. Así mismo el equipo de TI será el responsable de mantener y supervisar la solución, garantizando la disponibilidad y escalabilidad. La organización en su conjunto se favoreció al reducir costos operativos y agilizar los tiempos de respuestas para el tiempo de campañas de ventas.

3.2.3. Etapas y metodología

Para el diseño e implementación de la solución de edición automática de imágenes de forma masiva se basó en las fases propuestas por Kendall & Kendall (2023) abarcando estudio, análisis, diseño, desarrollo, pruebas, despliegue y transferencia, pero se adaptó con el contexto de la empresa, los pasos se detallan en la Tabla 8.

Tabla 8
Plan de Trabajo

Semana	Etapas
1	Estudio preliminar y planificación de requerimientos
2	Análisis de requerimientos
3	Diseño lógico del sistema
4	Diseño físico y selección de herramientas
5	Construcción del prototipo inicial
6	Integración de componentes
7	Pruebas unitarias y funcionales
8	Implementación de monitoreo y alertamiento
9	Validación de desempeño y Costos
10	Documentación y pase a producción

Nota. Fuente: Elaboración Propia en base al plan de trabajo creado para el proyecto

- a. **Semana 1: Estudio preliminar y planificación del proyecto**
Se realizó la recopilación de información y análisis de factibilidad para identificar costos, beneficios y riesgos. Según Kendall y Kendall (2023), esta etapa permite establecer

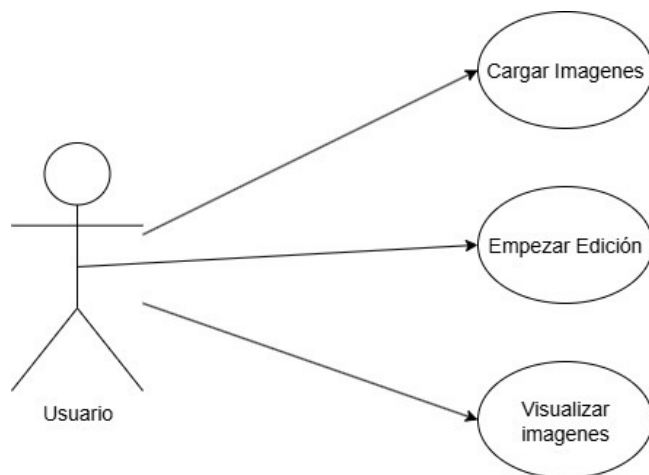
los lineamientos iniciales del sistema y comprender la problemática que se asocia a los cuellos de botella en procesos de TI y de negocio.

b. Semana 2: Análisis de requerimientos

Se levantan requerimientos funcionales y no funcionales en coordinación con las áreas de marketing y negocio. Werner (2004) menciona que las arquitecturas serverless es fundamental considerar desde el principio la elasticidad y la integración con orquestadores, lo cual permite cargar variables.

Figura 3

Casos de Usos



Nota. Fuente: Elaboración Propia basado en los casos de uso creados para la solución

c. Semana 3: Diseño lógico del sistema

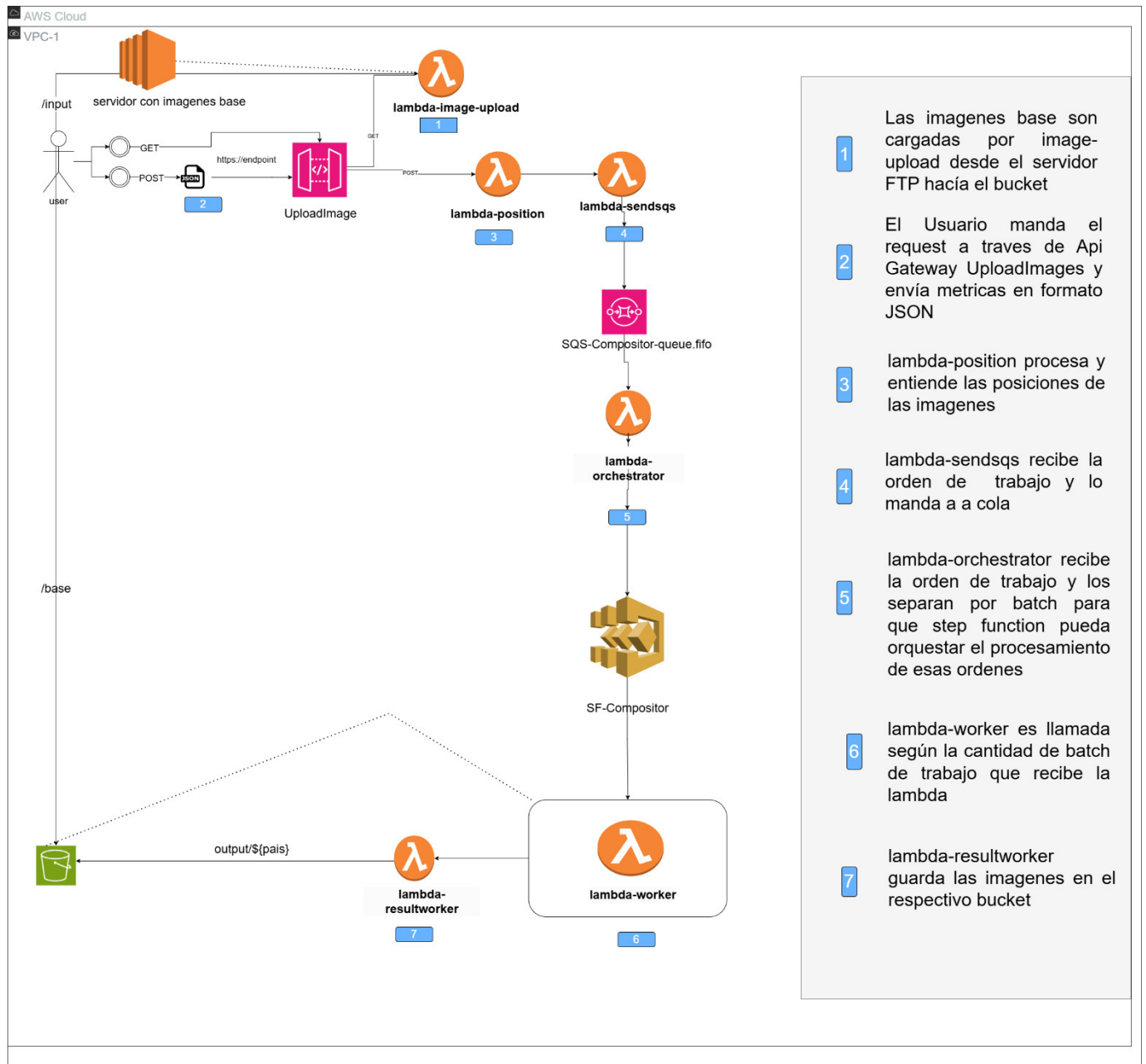
Se definen flujos lógicos del proceso: cargar de imágenes ejecución de Lambdas, orquestación con Step Functions y almacenamiento en SQS. Smith Lee (2025), señala que el uso de *infrastructure as Code (IaC)* facilita la consistencia y a replicar el proyecto para futuras modificaciones.

d. Semana 4: Diseño físico y selección de herramientas

Se seleccionó Serverless Framework como herramienta de despliegue, Lambdas desarrolladas en C#, almacenamiento en S3, Pahl, Jamshidi y Zimmermann (2025) destacan que el uso de IaC mejora la auditabilidad y confiabilidad de la infraestructura en entornos distribuidos.

Figura 4

Arquitectura de la solución



Nota. Fuente: elaboración propia en base a la arquitectura original

e. Semana 5: Construcción del prototipo inicial

Las primeras funciones Lambdas se desplegaron mediante Serverless framework, además se configura pipeline de CI/CD con Github actions para validar Versiones, de acuerdo con Github Engineering (2024) la integración de pruebas automatizadas en pipelines acelera la entrega y reduce errores.

f. Semana 6: Integración de componentes

Se conectan los servicios API Gateway, Lambdas, SQS, Step functions y almacenamiento en S3, validando el correcto funcionamiento de estos. Kendall y Kendall (2023) recomiendan que se deben realizar pruebas incrementales para asegurar la coherencia entre los subsistemas antes de la validación global.

g. Semana 7: Pruebas unitarias y funcionales

Se efectúan pruebas de las funciones Lambdas creadas y del flujo completo end to end de la edición de imágenes, evaluando los tiempos de respuestas respecto a la edición manual previa. GitHub Engineering (2024) sugiere incluir pruebas unitarias dentro de los pipelines para garantizar calidad continua.

h. Semana 8: Pruebas de rendimiento

Se configura métricas en New Relic y tableros en Grafana, además de alertas proactivas mediante pruebas sintéticas. New Relic (2024) enfatiza que monitorear métricas como inicios en frio y tiempos de ejecución es esencial para aplicaciones serverless.

i. Semana 9: Validación de desempeño y costos

Se realizan pruebas de rendimiento bajo cargas simuladas de campañas masivas y se comparan los costos frente a la situación anterior (equipo especializado). Werner (2024) señala que el análisis de costo-efectividad es un factor crítico en la adopción de arquitecturas serverless, pues garantiza escalabilidad con eficiencia financiera.

j. Semana 10: Documentación y pase a producción

Se documenta la arquitectura, procesos de operación, pruebas

realizadas en los ambientes de calidad, se coordina con DevOps y área de negocio el pase a producción. Kendall y Kendall (2023) subrayan que la documentación es clave como etapa final del ciclo de vida de un sistema, ya que asegura continuidad y soporte futuro.

3.2.4. Fundamentos utilizados

3.2.4.1. Fundamentos teóricos

a. Arquitectura serverless

La arquitectura serverless es un enfoca en el cual el desarrollo de una aplicación se apoya únicamente en servicios administrados por proveedores cloud, con funciones efímeras, base de datos y almacenamiento por separado. Principalmente abstrae la gestión de servidores, lo cual permite a los equipos enfocarse netamente en la lógica del negocio. Entre sus principales ventajas destacan:

- Escalabilidad automática
- Optimización de costos (Pago por consumo)
- Reducción de tareas operativas

Esta metodología también presenta retos como:

- Dependencia de la tecnología del proveedor
- Mayor complejidad en observabilidad y monitoreo

Werner, menciona esto como un modelo que combina el uso de Funciones como servicio (FaaS) y Backend como servicio (BaaS) para implementar soluciones ágiles y flexibles (Nightcore Project, 2021; Werner, 2024).

b. Infraestructura como código (IaC)

El concepto mismo de IaC se plantea que los recursos de cómputo y red se puedan gestionar por código o de manera programática, en lugar de realizar configuraciones manuales. Lo que implica que se deben de hacer uso de programación, con prácticas de ingeniería de software al aprovisionamiento de infraestructura, integrando versionado (Git), pruebas y despliegues automatizados. Entre los principales beneficios

destacan:

- Reducción de errores humanos
- Repetibilidad de entornos
- Trazabilidad de cambios

Pahl, Jamshidi & Zimmermann (2025), destacan la IaC con un pilar de la ingeniería de sistemas en la nube, al permitir desplegar entornos de forma coherente y escalable. Herramientas como Serverless Framework o Terraform son ejemplo de la aplicabilidad de este enfoque y de la importancia que están tomando.

c. Microservicios

Este paradigma consiste en dividir una aplicación grande de componentes independientes pero que se pueden comunicar entre sí. Esto promueve el desacoplamiento, la escalabilidad independiente de cada módulo y una mayor resiliencia en el sistema. Sin embargo, esto conlleva desafíos como:

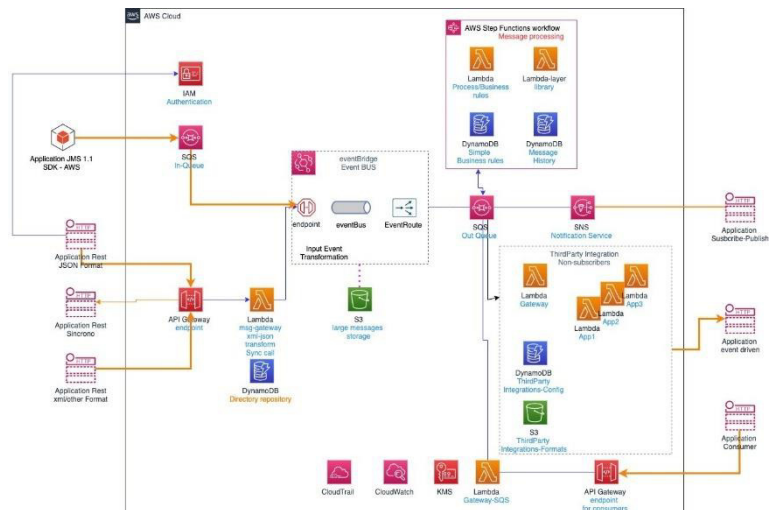
- Gestión de la comunicación distribuida
- Orquestación de servicios.

Ramasankar Molleti (2021) y Pahl, Jamshidi & Zimmermann (2024), han demostrado que la combinación de microservicios con tecnología serverless ofrece beneficios en cuanto a costos y rendimiento, especialmente para aplicaciones que son nativas de la nube.

En el blog oficial de AWS confirma que la arquitectura de microservicios como la de la imagen puede tener mejores beneficios

Figura 5

Integrando aplicaciones durante la modernización



Nota. Fuente: Blog de Amazon Web Services (AWS). Integrando aplicaciones durante la modernización.

Entre los principales beneficios que mencionan son:

- Cortos tiempo de entrega al mercado (TTM)
- Desacoplamiento de la arquitectura
- Integraciones síncronas (en vivo) y asíncronas
- Alta Flexibilidad
- Costos por uso, sin costos adicionales por licenciamiento

d. Pruebas de software

Estas pruebas son un proceso sistemático destinado a evaluar la calidad de un producto verificando de que se cumpla los requisitos definidos. Hay diferentes distintos niveles como:

- Pruebas unitarias
- Pruebas de integración
- Pruebas de sistemas
- Pruebas de aceptación

Este enfoque tiene como fin detectar problemas temprano.

3.2.4.2. Fundamentos metodológicos

a. Pruebas automatizadas

La automatización de pruebas con scripts y herramientas para ejecutar pruebas, evitando intervención manual. Su aplicación permite disminuir el tiempo de validación, aumentar la cobertura y mejorar la detección de fallas, la

página de GeeksforGeeks (2024) destacan estas metodologías se han convertido en un estándar en proyectos cloud. Tenemos las diferencias entre Testing manual y Testing automático en la tabla:

Tabla 9
Diferencia entre Testing Manual y automático

Parámetro	Testing Manual	Testing Automático
Definición	Es ejecutado por un QA Humano.	Es ejecutado por una herramienta de software
Tiempo del test	Consume tiempo	Es más rápido
Recursos requeridos	Requiere recursos humanos	Hace uso de herramientas de automatización y Trabajadores capacitados
Pruebas de rendimiento	Imprácticas para pruebas de rendimiento sistemáticas (p. ej., pruebas de carga, estrés y picos)	Admite pruebas de rendimiento (p. ej., pruebas de carga, estrés y picos) con herramientas como JMeter.
Pruebas por lotes:	En las pruebas manuales, las pruebas por lotes no son posibles	Se puede realizar varias pruebas por lotes para una ejecución más rápida.
¿Cuándo usarlas?	Las pruebas	Las pruebas

manuales son útiles automatizadas
 para pruebas son adecuadas
 exploratorias, de para pruebas de
 usabilidad y ad hoc regresión, de
 carga y de
 rendimiento.

Nota. Fuente: [geeksforgeeks.org](https://www.geeksforgeeks.org/software-testing/), Software testing

Como Podemos ver en la tabla es de para las pruebas de estrés se deben hacer uso de pruebas automatizadas.

b. Integración continua/Entrega Continua (CI/CD)

Es un pilar dentro de los equipos de desarrollo y de operaciones. Con CI/CD los cambios en el código se integran y se despliegan de forma automática, lo que favorece entregas frecuentes de mayor calidad. En entornos serverless, se pueden añadir pruebas, validaciones y despliegues sin intervención manual. Publicaciones de Github Engineering (2024) muestra cómo hacer uso de pipelines basados en Actions que genera una cadena de valor más eficiente.

3.2.5. Implementación de las áreas, procesos, sistemas y buenas practicas

Para el diseño e implementación se siguió el orden que está en el punto 3.2.3.1 donde se describen las etapas del proyecto.

Equipo de desarrollo

El equipo encargado de la implementación del proyecto, donde el autor tuvo el rol de SRE, en la siguiente Tabla de detalla los miembros del equipo

Tabla 10

Roles del proyecto

Rol	Descripción	Cantidad de miembros
Product	Responsable de definir los objetivos	1

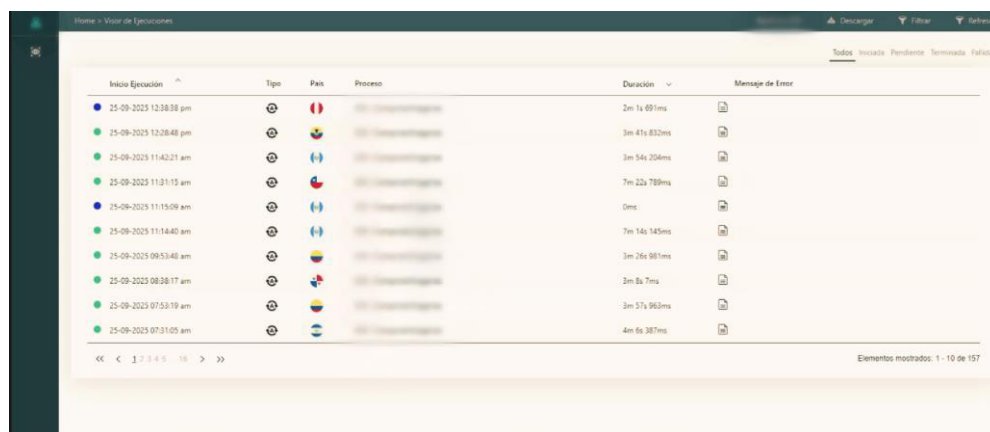
manager	del proyecto, que deben estar alineados con las necesidades del Negocio y de los usuarios. Supervisa la planificación, priorización de funcionalidades y asegura que el producto sea de valor al cliente.	
IT Owner	Responsable de la gestión y gobernanza de los recursos de tecnología, su principal función es garantizar que la infraestructura y los servicios de TI estén en constante alineamiento con los objetivos estratégicos de la organización, velando por seguridad, cumplimiento y eficiencia.	1
Software Developer	Profesional quien se encarga de diseñar, programar y mantener la aplicación.	3
QA	Responsable de la validación de la calidad del software. Crear y ejecuta pruebas para detectar errores, asegurar el cumplimiento de estándares y verificar que los entregables satisfagan los requisitos establecidos.	1
SRE (Site Reliability Engineer)	Responsable de combinar prácticas de desarrollo de software y buenas prácticas para construir sistemas confiables, escalables y de alto rendimiento. Automatiza procesos, gestiona incidentes y realiza mejora	1

	continua para garantizar la disponibilidad del sistema	
Technical Lead	Líder técnico del equipo, especial del equipo de desarrollo. Define arquitectura, buenas prácticas y lineamientos técnicos. Asegura la calidad del código haciendo uso de buenas prácticas al momento de programar.	1
Operation	Responsable de administrar, monitorear. Se encarga de asegurar la continuidad de los servicios.	1

Nota. Fuente: Elaboración propia en base Diseño del prototipo

Para llamar a las lambdas y realizar la carga del archivo JSON se desarrolló una interfaz de gestión cómo se ve en la siguiente Figura

Figura 6
Interfaz web de gestión de ejecuciones de ediciones



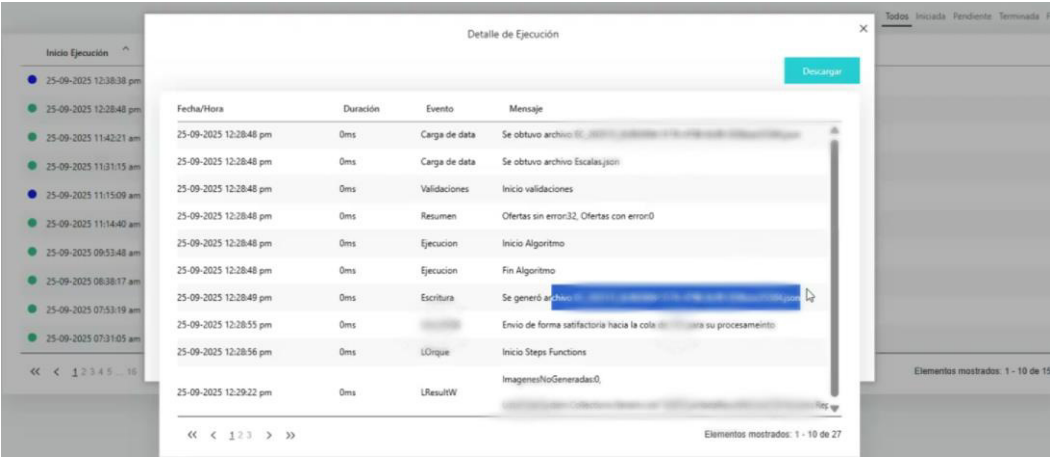
Inicio Ejecución	Tipo	País	Proceso	Duración	Mensaje de Error
25-09-2023 13:38:38 pm	🔗	🇵🇪	CC - Procesamiento	2m 1s 691ms	📄
25-09-2023 13:38:48 pm	🔗	🇵🇪	CC - Procesamiento	3m 41s 832ms	📄
25-09-2023 11:42:21 am	🔗	🇵🇪	CC - Procesamiento	3m 54s 208ms	📄
25-09-2023 11:31:15 am	🔗	🇵🇪	CC - Procesamiento	7m 22s 789ms	📄
25-09-2023 11:15:09 am	🔗	🇵🇪	CC - Procesamiento	0ms	📄
25-09-2023 11:14:40 am	🔗	🇵🇪	CC - Procesamiento	7m 14s 145ms	📄
25-09-2023 09:53:48 am	🔗	🇵🇪	CC - Procesamiento	3m 26s 981ms	📄
25-09-2023 08:38:17 am	🔗	🇵🇪	CC - Procesamiento	3m 8s 7ms	📄
25-09-2023 07:53:19 am	🔗	🇵🇪	CC - Procesamiento	3m 57s 963ms	📄
25-09-2023 07:31:05 am	🔗	🇵🇪	CC - Procesamiento	4m 6s 387ms	📄

Nota. Fuente: Elaboración propia basado en la web.

En la interfaz web se puede gestionar las ejecuciones de la edición de imágenes según las campañas que se vayan ejecutando y se puede ver el tiempo que demora en ejecutar, el país de la campaña y el resultado de la ejecución. Además,

se pueden observar los logs de cada ejecución cómo se muestra en la siguiente figura:

Figura 7
Interfaz web - logs



Nota. Fuente: elaboración propia basado en la interfaz web del proyecto

Flujograma del proceso

Imágenes

- **Imagen previa**

La imagen se carga en el Bucket s3, la cual es procesada para su posterior edición.

Figura 8
Imagen Simple 1



Nota. Fuente: Elaboración propia en base al catálogo.

Figura 9
Imagen Simple 2



Nota. Fuente: Elaboración propia en base al catálogo.

Como primera fase del proceso, cada imagen se redimensiona en tres tamaños:

- Small
- Medium (tamaño de la imagen Normal)
- Large

Cada imagen se va a usar según lo que describa un archivo JSON llamado Campania.json

- **Imagen Compuesta**

Es la imagen editada, con la oferta final que se pondrían en el catálogo y el uso para las campañas publicidad, la imagen se pueden duplicar, como se detalla en la siguiente imagen:

Figura 10

Imagen Compuesta



Nota. Fuente: Elaboración propia en base al catálogo.

O se pueden juntar con otras imágenes para generar una oferta en combo.

Figura 11

Imagen Compuesta 2



Nota. Fuente: Elaboración propia

Pruebas

Las pruebas que se realizaron para el proyecto son para garantizar el correcto funcionamiento de la aplicación en producción, sin errores.

Las **pruebas End-to-End (E2E)** constituyen un método integral para verificar el correcto funcionamiento de una aplicación en su totalidad, considerando el punto de vista del usuario final. Estas pruebas reproducen situaciones reales de uso, recorriendo los distintos niveles del sistema e interfaz, lógica de negocio, servicios y bases de datos. Con el propósito de comprobar que los componentes interactúan de manera adecuada. Su aplicación permite disminuir riesgos de fallos en entornos de producción y asegurar la confiabilidad del software en arquitecturas distribuidas y soluciones basadas en la nube (Alalfi et al., 2025).

En contraste, las **pruebas unitarias** se centran en validar pequeños fragmentos del código de manera aislada, como funciones o módulos específicos. Su mayor aporte radica en la detección temprana de errores durante las primeras fases del ciclo de desarrollo, lo cual favorece la mantenibilidad y la calidad del software. Este tipo de pruebas resulta esencial en metodologías ágiles y en la práctica de **integración y**

entrega continua (CI/CD), dado que garantiza estabilidad y confianza en los despliegues continuos (GeeksforGeeks, 2025).

Por su parte, las **pruebas de rendimiento** tienen como finalidad evaluar el desempeño del sistema bajo distintas condiciones de carga. Este tipo de validación permite identificar limitaciones, analizar la escalabilidad y verificar que los tiempos de respuesta se mantengan dentro de los parámetros esperados aun en escenarios de alta demanda. En la actualidad, donde la experiencia de usuario y la eficiencia tecnológica son factores estratégicos, este tipo de pruebas constituye un pilar clave para asegurar la competitividad y la solidez de las aplicaciones (Smith & Lee, 2025).

Archivo JSON

El archivo JSON (JavaScript Object Notation) es un formato que está siendo prácticamente un estándar para la comunicación web, es ligero de intercambio de datos. Está basado en texto plano y de ser bastante fácil de leer, tanto por humanos como para maquinas. Se componen de partes mediante pares clave-valor y listas ordenadas, esto facilita que la representación sea de forma jerárquica. Su simplicidad y compatibilidad con múltiples lenguajes de programación lo han permitido posicionarse como estándar para aplicaciones web, móviles y de sistemas distribuidos. Además, hacer uso de estructuras mediante el esquema JSON garantiza consistencia e integridad en entornos de software modernos (Attouche etc al., 2023).

Para el presente trabajo, se usaron 2 archivos JSON, para poder usarse en la lógica. Los archivos son listas de JSON según la cantidad de márgenes que se van a procesar.

- **Escala.json**

Es el archivo en el cual se va a definir la escala que tendrá la imagen, se puede ver un ejemplo en la Figura 10.

Figura 12
Archivo de escalas

```

1  {
2    "Escalas": [
3      {
4        "CodigoImagen": 200009089,
5        "Escala": 3,
6        "EscalaA": 134,
7        "EscalaB": 168
8      }
9    ]
10 }

```

Nota. Fuente: Elaboración propia en el archivo Original

El detalle del archivo se detalla en la Tabla:

Tabla 11
Detalle de JSON de Escalas

Nombre	Significado
CodigoImagen	Es el código identificador Único a la m imagen
Escala	Es el tamaño que se tendrá en la imagen: - 1: small - 2: medium - 3: largue
EscalaA	Es el punto donde se pondrá la imagen en el eje X
EscalaB	Es el punto donde iniciara la imagen en el eje Y

Nota. Fuente: Elaboración propia

- **Campania.json**

Este archivo JSON es el que define las características que tendrá la imagen final.

Figura 13
Archivo de Campañas

```

1  {
2    "Campaña": [
3      {
4        "Formato": "JPEG",
5        "Codigo_Oferta": "0123",
6        "Cantidad_de_Productos": 2,
7        "Nombre_Imagen_resultado": "pais_campana_numero",
8        "Productos": [
9          {
10           "CodigoImagen": "123456",
11           "CantidadRepeticion": 1,
12           "Tipo": "Fragancias"
13         },
14         {
15           "CodigoImagen": "1234567",
16           "CantidadRepeticion": 1,
17           "Tipo": "Fragancias"
18         }
19       ]
20     }
21   ]
22 }
23

```

Nota. Fuente: Elaboración propia basado en el archivo original

El archivo JSON se detallan en la Tabla

Tabla 12
Detalle de JSON de Campaña

Nombre	Descripción
Formato	Es el archivo de exportación de la imagen, pueden ser: - JPEG - PNG - web
Codigo_Oferta	Es el identificador de la imagen que se generaría.
Cantidad_de_Productos	Es la cantidad de imágenes que se cargarían para componer la imagen que se generaría.
Nombre_Imagen_resultado	Es el nombre que tendrá la imagen generada.

Productos	Es la lista de imágenes que se unen para generar la imagen exportada.
CodigoImagen	Es el identificador de la imagen simple que se usará en la composición de la nueva imagen.
CantidadRepeticion	Es la cantidad de veces que se repite el producto en la imagen final.
Tipo	Es el tipo de producto de la imagen, pueden ser: - Fragancias - Cosméticos - Joyería - Crema

Nota. Fuente: Elaboración propia

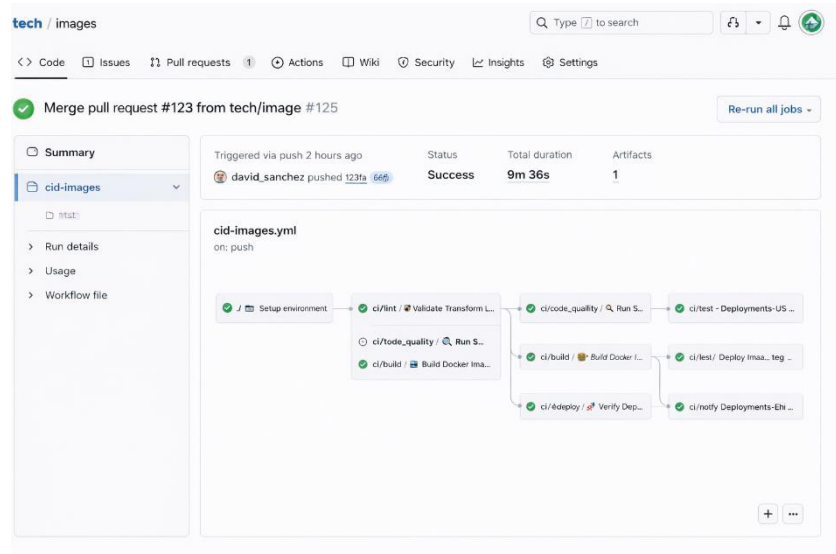
- **Integración Continua (CI)**

CI es una práctica continua que consiste en integrar de manera frecuente y automática el código de un repositorio, en conjunto con procesos de construcción y pruebas que aseguran la calidad del código antes que llegue a un ambiente productivo. Tiene como objetivo principal detectar errores en etapas tempranas, reducir el costo de corrección y garantizar entregas más confiables (Fowler, 2021).

- **Despliegue Continuo (CD)**

Despliegue Continuo automatiza la entrega de software hacia ambiente de Calidad (QA) y de producción después de la integración continua, lo cual reduce la intervención manual. Github Actions se posiciona como una de las herramientas más importantes por tener la capacidad de implementar pipelines directamente desde el repositorio de Github facilitando la construcción, prueba y despliegue de aplicaciones. (Github Engineering, 2024).

Figura 14
Git Actions



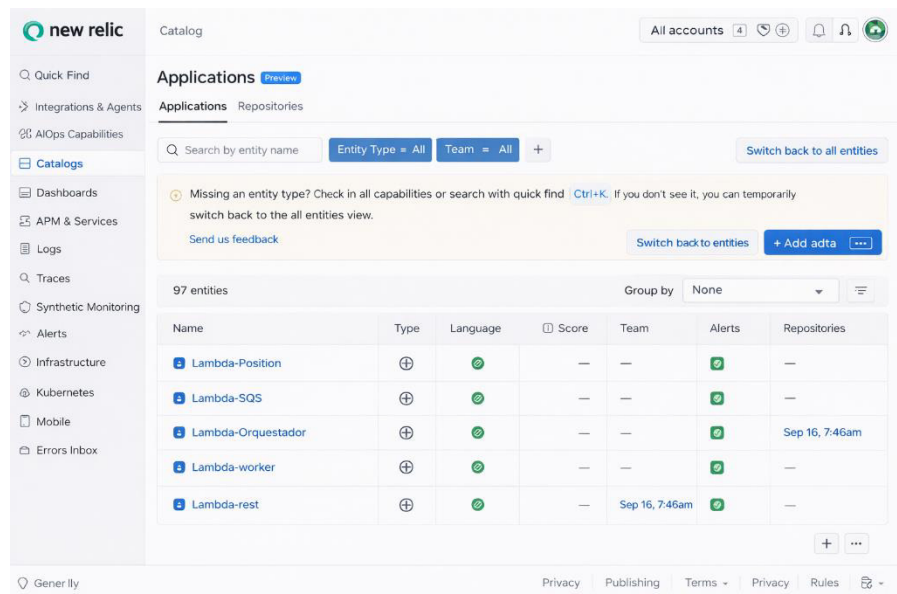
Nota. Fuente: repositorio propio basado en el del proyecto

- **Monitoreo y Observabilidad**

El monitoreo es la práctica de recopilar, procesar y analizar las métricas, logs y trazas de los sistemas para identificar el estado de las aplicaciones e infraestructuras en tiempo real.

La observabilidad, va un paso más allá, se centra a analizar la capacidad de un sistema para mostrar la suficiente información interna para comprender y diagnosticar problemas complejos. En este sentido, **New Relic** es una plataforma de observabilidad completa que centraliza métricas. Eventos, logs y Trazas (MELT), lo que facilita la detección temprana de anomalías, la optimización del rendimiento y la mejora de la experiencia del usuario final. Enfocarse en la observabilidad moderna permite a los equipos de DevOPS y SRE, más que encontrar fallos permite encontrar el motivo raíz hay detrás de ello, lo que habilita respuestas más rápidas y efectivas.

Figura 15
New Relic

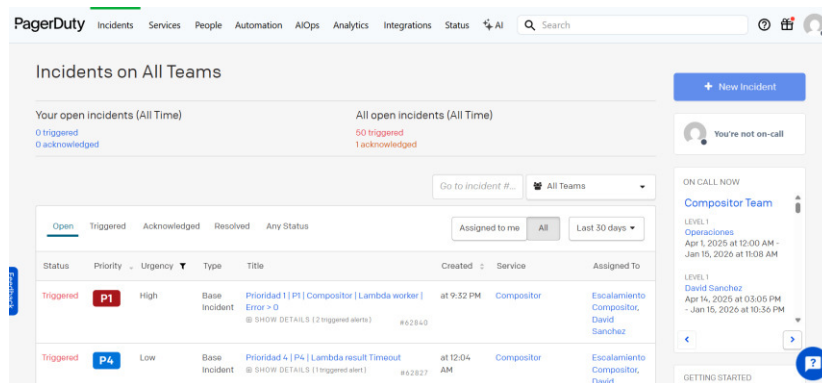


Nota. Fuente: New Relic de la empresa - editada

- **Alertamiento**

El alertamiento es considerado un componente critico dentro de las estrategias de monitoreo y observabilidad, ya que nos va a permitir notificar en tiempo real a los equipos responsables cuando ocurren incidentes o anomalías en las aplicaciones. Gestionar de forma eficaz no es solo generar notificaciones, sino que también se debe priorizar los eventos según el impacto que tenga en el sistema y asegura que los responsables adecuados sean informados para minimizar el tiempo de resolución. PagerDuty es una plataforma especializada en la gestión de incidentes que se integra con muchas herramientas de monitoreo como New Relic, o inclusive con AWS con CloudWatch, lo que permite crear notificaciones eficientes. Tiene un enfoque de automatización de flujos de trabajos que permite escalar alertas, reducir falsos positivos y coordinar la respuesta entre diferentes equipos. Al centralizar toda la gestión de alertas aplicando inteligencia operativa, PagerDuty ayuda a las organizaciones a mantener la continuidad del negocio y mejora los acuerdos de nivel de servicio (SLA) al permitir una respuesta más rápida y eficiente a problemas críticos (PagerDuty,2022).

Figura 16
Interfaz de PagerDuty



Nota. Fuente: Captura de PagerDuty de la empresa - Editada

- **Evaluación**

Evaluación económica

Para la Evaluación económica del proyecto se han considerado costos asociados al personal involucrado, además de los servicios en la nube y herramientas que usan para el desarrollo y operación de la solución.

Tabla 13

costos por mes del personal a cargo del proyecto

Rol	Costo mensual (s/.)
Product Manager	7500
IT Owner	8000
Software Developer	6000
QA	5000
SRE	7500
Technical Lead	9000
Operations	4500
Total, por mes	47500
Total, por proyecto (2.5 meses)	118,750

Nota. Fuente: elaboración propia en base a estimaciones del mercado TI

además de los costos del equipo, se realizó estimaciones en costos de Tecnología por el costo Anual.

Tabla 14
Calculo de gastos en infraestructura

Herramienta/Servicio	Costo anual (S/.)
AWS (Lambdas, S3, Route 53, Api Gateway, Logs, etc.)	5000
New Relic (monitoreo y observabilidad)	2000
Github (Actions + repositorios privados)	4,800
PagerDuty (alertamiento de incidentes)	3,200
Total	15000

Nota. Fuente: Elaboración propia, en base a estimaciones del proyecto

En base a esta información, el proyecto tuvo una inversión de S/. 133,750 nuevos soles durante las 10 semanas de duración, considerando S/. 118,750 en personal y S/. 15, 000, en servicios cloud y herramientas.

- **Evaluación de tiempos**

Se considera que el proyecto generará ahorros operativos y beneficios económicos que tendría.

Por campaña se pasan en promedio de 100 imágenes que antes se procesaban en 10 horas, se procesan en 5 minutos, lo que equivale al 99% de tiempo de operación.

Se estima que se tienen cerca de 20 campañas grandes en lo que va alrededor del año.

Tabla 15
Tiempo de Trabajo en edición

Ahorro	Tiempo anterior	Tiempo nuevo por campaña
Tiempo de preparación de imágenes para	10 horas	5 minutos

campaña	
Tiempo de trabajo de 10 horas	0 horas
equipo de edición	

Nota. Fuente: Elaboración propia basado en estimaciones del proyecto

- **Evaluación de Costos/Tiempo**

Durante alrededor del año se tienen aproximadamente 20 campañas grandes alrededor del año, lo cual lleva a un ahorro de tiempo grande a la operación, que se detalla en la siguiente tabla:

Tabla 16
Evaluación de Costo/Tiempo

Detalle	Tiempo	Costo asociado
Precio por hora de edición del equipo externo	1 Hora	S/. 600
Cantidad de Horas de Trabajo por campaña	10 horas	S/ 6,000
Cantidad de Horas de Trabajo por todas las 20 campañas durante el año	200 horas	S/. 120,000

Nota. Fuente: Elaboración propia basado en estimaciones del proyecto

En base a la información el proyecto tuvo un beneficio económico de 120,00 anuales.

- **Interpretación del VAN y TIR**

Como se sabe el proyecto tiene una inversión de S/. 133,750 y beneficios económicos que equivale a S/. 120,000 anuales durante los próximos 5 años con una tasa de descuento del 10% anual. En la Tabla se hallan los cálculos que se han estimado.

Tabla 17
Detalle de VAN y TIR

Año	Flujo Neto (FNE)	FNE (S/.)	Actual	FNE (S/.)	Acumulado
-----	------------------	-----------	--------	-----------	-----------

0	-133,750	-133,750.00	-133,750.00
1		95,454.55	-38,295.45
	105,000		
2		86,776.86	
	105,000		48,481.41
3		78,887.15	127,368.56
	105,000		
4		71,715.59	199,084.15
	105,000		
5			264,279.23
	105,000	65,195.08	

Nota. Fuente: Elaboración propia

Se puede concluir de los resultados obtenidos, que el VAN es de S/. 264,279.23 y **TIR $\approx$ 72%** (muy superior al 10% de la tasa de descuento). Con estos datos se determina que el proyecto es rentable ya que el VAN es positivo y la TIR supera ampliamente la tasa de descuento.

Capítulo IV Reflexión crítica de la experiencia

El autor del presente trabajo de suficiencia profesional (TSP) ha desempeñado el rol de SRE, y gracias a esa experiencia obtenida en el proyecto, ha obtenido las siguientes reflexiones:

- El presente proyecto se llevó a cabo utilizando un enfoque adaptado a la naturaleza de la solución propuesta, donde automatizar procesos mediante funciones Lambda serverless se constituyó como eje principal.

La elección de Serverless Framework como herramienta para levantar la infraestructura resultó ser fundamental, ya que permitió mantener el orden, trazabilidad y escalabilidad en cada despliegue. Esta decisión fue acertada con las mejores prácticas de infraestructura como código, lo que permitió que la implementación pueda repetirse con precisión en distintos entornos.

- Uno de los principales aprendizajes fue la necesidad de alinear la parte técnica con los objetivos del negocio. La automatización de campañas de ventas, además de enfocarse en reducir los tiempos de edición de materiales gráficos, sino también en ofrecer un valor tangible en las áreas de marketing y ventas. Lo que permitió validar que las decisiones técnicas respondieron directamente a las necesidades del proyecto, lo cual consolidó la aceptación del proyecto por parte de los stakeholders.
- Durante la duración del proyecto, se encontraron retos relacionados con la coordinación entre equipos técnicos y los responsables del negocio. Aunque el diseño de microservicios mediante Lambdas facilitó la modularidad y el escalamiento de la solución, fue necesario establecer una comunicación constante mediante reuniones diarias para alinear expectativas y realizar ajustes en los entregables. Este aspecto es evidencia de la importancia de contar con procesos bien definidos, especialmente cuando se introducen tecnologías emergentes en un entorno corporativo.
- Un punto que destaca de la experiencia fue la incorporación de buenas prácticas de automatización y monitoreo desde el inicio del proyecto. La integración de pipelines de CI/CD con GitHub Actions permitió acelerar la entrega de nuevas versiones sin comprometer la calidad del código. Además, la inclusión de herramientas de observabilidad garantizó tener el control sobre la disponibilidad y el rendimiento del sistema, reduciendo riesgos en la operación.
- Se demostró que trabajar los proyectos en cloud en contraste de ambientes onPremises, permite aumentar la agilidad en la creación de la arquitectura, pues permite crear recursos en cuestión de minutos.

- Finalmente, el proyecto demostró que el uso de arquitecturas serverless no solo representa una oportunidad de mejora tecnológica, sino también un cambio cultural en la forma de gestionar los recursos de TI. La experiencia dejó en claro que, para aprovechar al máximo esta propuesta, es necesario fomentar la capacitación continua y mantener una visión estratégica que permita extender este tipo de soluciones hacia otras áreas de la organización, promoviendo así un ecosistema digital más ágil, eficiente y sostenible.

Capítulo V conclusiones y recomendaciones

5.1. Conclusiones

La implementación de la solución serverless en AWS para la edición automatizada de imágenes permitió cumplir de manera efectiva el objetivo general del proyecto, logrando optimizar significativamente los tiempos y costos asociados al proceso de edición. La adopción de funciones AWS Lambda, junto con la orquestación mediante Step Functions y la mensajería a través de SQS, hizo posible la ejecución paralela de tareas, reduciendo de forma considerable los plazos que anteriormente requerían procesos manuales, en concordancia con la metodología de desarrollo propuesta por Kendall y Kendall (2023).

En relación con el primer objetivo específico, se desarrolló exitosamente una solución capaz de procesar la edición de imágenes a partir de parámetros definidos en archivos JSON, lo que permitió estandarizar el flujo de trabajo y facilitar la configuración de campañas sin necesidad de intervención técnica especializada. Esta estandarización contribuyó a una mayor eficiencia operativa y a la disminución de errores recurrentes durante períodos de alta demanda comercial.

Respecto al segundo objetivo específico, la integración de la automatización con los entornos de almacenamiento utilizados por el equipo, específicamente mediante el uso de Amazon S3, facilitó la carga, gestión y recuperación de las imágenes editadas. Esto permitió una comunicación fluida entre los equipos de marketing y tecnología, eliminando dependencias innecesarias y mejorando la

disponibilidad de los recursos gráficos para su uso inmediato en campañas publicitarias.

Asimismo, el uso de Serverless Framework para la gestión de la infraestructura bajo el enfoque de infraestructura como código (IaC) permitió un aprovisionamiento rápido, seguro y replicable, garantizando la escalabilidad y mantenibilidad de la solución en entornos productivos. Esto no solo redujo los costos asociados a la administración de servidores dedicados, sino que también demostró la capacidad de la plataforma para adaptarse a cargas de trabajo variables, especialmente en campañas estacionales como Navidad, Día de la Madre o San Valentín.

Finalmente, se concluye que la solución desarrollada constituye una herramienta robusta, flexible y alineada con buenas prácticas de ingeniería de software, automatización y gestión de proyectos. Su implementación elimina la dependencia de equipos externos de edición, reduce los cuellos de botella operativos y fortalece la capacidad de respuesta de la organización frente a las exigencias del mercado, aportando valor tanto a nivel técnico como estratégico.

5.2. Recomendaciones

En base al éxito del presente trabajo se recomienda considerar la replicación de este modelo de automatización en otros procesos que al igual que la edición de imágenes que demanden altos volúmenes de trabajo en plazos cortos. Otras áreas tal vez como generación de reportes, procesamiento de datos de forma masiva o personalización de materiales de marketing digital podrían beneficiarse de una arquitectura similar, optimizando tiempos y disminuyendo costos operativos.

También, es aconsejable continuar evaluando de manera periódica el desempeño de las funciones lambdas y flujos de Step Functions, con herramientas de observabilidad como New Relic, Grafana y configurar alertamientos para tener atención rápido y a fin de detectar alguna posible mejora en la configuración y aprovechar nuevas funcionalidades que los proveedores de Cloud vayan incorporando. Realizar este seguimiento constante garantizará que la plataforma evolucione según las nuevas tendencias y necesidades del negocio.

Finalmente, se recomienda difundir la documentación y resultados de este caso de éxito como referencia para que la organización lo tome como ejemplo y como referencia dentro de la organización, ya que constituye un ejemplo aplicable para una modernización de procesos. De esa forma, esta experiencia adquirida podría servir como guía para futuros proyectos de transformación digital.

5.3. Fuentes de información

- GitHub Engineering (2024). Building a serverless deployment pipeline with GitHub Actions. GitHub Blog. Recuperado el 24 de septiembre de 2025, de <https://github.blog>
- Kendall, K. E., & Kendall, J. E. (2023). Análisis y diseño de sistemas (10.^a ed.). Pearson.
- New Relic (2024). Best practices for monitoring serverless applications. New Relic Documentation. Recuperado el 24 de septiembre de 2025, de <https://newrelic.com>
- Pahl, C., Jamshidi, P., & Zimmermann, O. (2025). Infrastructure as code for cloud systems engineering. IEEE Software, 42(3), 45–53. Recuperado el 24 de septiembre de 2025, de <https://ieeexplore.ieee.org>
- Smith, R., & Lee, T. (2025). IAC-based serverless image processor with AWS & Terraform. Journal of Cloud Computing Research, 14(2), 45–60. Recuperado el 24 de septiembre de 2025, de <https://journalofcloudcomputing.springeropen.com>
- Werner, A. (2024). Serverless architectures and cloud-native strategies. Addison-Wesley. Recuperado el 24 de septiembre de 2025, de <https://www.pearson.com/store/p/serverless-architectures-and-cloud-native-strategies>
- Nightcore Project (2021). Efficient and scalable serverless computing para microservicios interactivos sensibles a latencia. ASPLOS. Recuperado el 24

de septiembre de 2025, de <https://www.cs.utexas.edu/~witchel/pubs/jia21asplos-nightcore.pdf>

- Chadha, M., et al. (2022). Migrating from Microservices to Serverless: estudio de caso en plataforma IoT. arXiv. Recuperado el 24 de septiembre de 2025, de <https://arxiv.org/pdf/2210.04212>
- Amazon Web Services (2021). Serverless Multi-Tier Architectures con API Gateway y Lambda (whitepaper). AWS Whitepaper. Recuperado el 24 de septiembre de 2025, de <https://docs.aws.amazon.com/pdfs/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/serverless-multi-tier-architectures-api-gateway-lambda.pdf>
- Fan, C. F., et al. (2020). Microservices vs Serverless: comparación de rendimiento en aplicación web cloud-native. ScitePress. Recuperado el 24 de septiembre de 2025, <https://doi.org/10.1016/j.future.2024.02.006> de <https://www.scitepress.org/Papers/2020/97927/97927.pdf>
- Serverless Architectures for Scalable Cloud-Based Microservices (2024). Performance y trade-offs de costos en microservicios serverless. ResearchGate. Recuperado el 24 de septiembre de 2025, de https://www.researchgate.net/publication/393453501_Serverless_Architectures_for_Scalable_Cloud-Based_Microservices_Performance_and_Cost_Trade-Offs
- Newman, S. (2015). Building Microservices. O'Reilly. Recuperado el 24 de septiembre de 2025, de <https://www.oreilly.com/library/view/building-microservices/9781491950340/>
- Richardson, C. (2018). Microservices Patterns: With examples in Java. Manning Publications. Recuperado el 24 de septiembre de 2025, de <https://www.manning.com/books/microservices-patterns>

- AWS Blog (s.f.). Integrando aplicaciones durante la modernización. *AWS Blog (Español)*. Recuperado el 24 de septiembre de 2025, de <https://aws.amazon.com/es/blogs/aws-spanish/integrando-aplicaciones-durante-la-modernizacion/>
- Alalfi, M., et al. (2025). Requirements-Driven Automated Software Testing: A Systematic Review (REDAST). arXiv. Recuperado el 24 de septiembre de 2025, de <https://arxiv.org/abs/2502.18694>
- Molleti, R. (2021). *Comparing Serverless and Microservices Architecture Patterns in FinTech. International Journal for Multidisciplinary Research (IJFMR)*. Recuperado el 24 de septiembre de 2025, de <https://www.ijfmr.com/papers/2021/6/2750.pdf>
- Pahl, C., Jamshidi, P., & Zimmermann, O. (2024). Microservices and serverless functions — lifecycle and deployment trade-offs. *Future Generation Computer Systems*, 152, 493–503. Recuperado el 24 de septiembre de 2025, de <https://www.sciencedirect.com/science/article/pii/S0167739X24000529>
- GeeksforGeeks. (2025, julio 19). *Automation Testing - Software Testing*. Recuperado el 24 de septiembre de 2025, de https://www.geeksforgeeks.org/software-testing/automation-testing-software-testing/?utm_source=chatgpt.com
- Attouche, L., Baazi, M.-A., Colazzo, D., Gheli, G., Sartiani, C., & Scherzinger, S. (2023). *Validation of Modern JSON Schema: Formalization and Complexity*. arXiv. Recuperado el 25 de septiembre de 2025, de <https://arxiv.org/abs/2307.10034>
- GitHub Engineering. (2024). *Building a serverless deployment pipeline with GitHub Actions*. *GitHub Blog*. Recuperado el 27 de septiembre de 2025, de <https://github.blog>
- New Relic. (2021). *The observability forecast: Defining the future of software*. New Relic. Recuperado el 27 de septiembre de 2025, de <https://newrelic.com/resources/ebooks/observability-forecast>
- PagerDuty. (2022). *The State of Digital Operations 2022*. PagerDuty. Recuperado el 27 de septiembre de 2025, de <https://www.pagerduty.com/resources/reports/state-of-digital-operations-2022>

- Universidad Nacional Mayor de San Marcos. (s. f.). *Perfil del egresado de Ingeniería de Sistemas*. Universidad Nacional Mayor de San Marcos. Recuperado el 2 de enero de 2026, de <https://www.unmsm.edu.pe/formacion-academica/carreras-de-pregrado/carrera-detalle/ingenieria-de-sistemas>
- Amazon Web Services. (s. f.). *¿Qué es Site Reliability Engineering (SRE)?*. Amazon Web Services. Recuperado el 2 de enero de 2026, de <https://aws.amazon.com/es/what-is/sre/>

5.4. Glosario

Glosario de términos

- **TI (Tecnología de la Información):** Conjunto de recursos, herramientas, equipos, programas informáticos, aplicaciones y dispositivos utilizados para la gestión, procesamiento y transmisión de información.
- **AWS (Amazon Web Services):** Plataforma de servicios en la nube ofrecida por Amazon, que proporciona infraestructura, almacenamiento, bases de datos, redes, inteligencia artificial y múltiples servicios bajo demanda.
- **Serverless:** Modelo de computación en la nube en el cual el proveedor administra la infraestructura de servidores, permitiendo al usuario enfocarse únicamente en el desarrollo de la lógica de negocio.
- **Bucket S3:** Recurso de Amazon S3 (Simple Storage Service) que permite almacenar y organizar objetos (archivos) en contenedores denominados “buckets”.
- **SRE (Site Reliability Engineering):** Disciplina de ingeniería que combina software y operaciones para crear sistemas escalables, confiables y eficientes. También hace referencia al puesto de trabajo especializado en aplicar esta metodología.
- **IaC (Infrastructure as Code):** Práctica que consiste en gestionar y aprovisionar la infraestructura tecnológica mediante código, evitando la configuración manual y permitiendo la automatización y reproducibilidad.
- **Serverless Framework:** Herramienta open source que facilita la creación y despliegue de aplicaciones serverless en múltiples proveedores de nube. En AWS permite implementar infraestructura bajo el enfoque de IaC.

- **Lambda (AWS Lambda):** Servicio de AWS que permite ejecutar código sin necesidad de aprovisionar ni administrar servidores, cobrando únicamente por el tiempo de ejecución utilizado.
- **API Gateway:** Servicio de AWS que permite crear, publicar, mantener, monitorear y asegurar APIs a escala. Es comúnmente utilizado junto con funciones Lambda.
- **SQS (Simple Queue Service):** Servicio de colas de mensajes totalmente administrado por AWS que facilita la comunicación entre componentes desacoplados de un sistema.
- **Cloud Computing:** Modelo de prestación de servicios tecnológicos a través de internet, en el cual los recursos como servidores, almacenamiento, bases de datos y software se entregan bajo demanda.
- **DevOps:** Conjunto de prácticas que combinan desarrollo de software (Dev) y operaciones de TI (Ops) con el fin de acortar el ciclo de vida del desarrollo y entregar software de alta calidad de manera continua.
- **EventBridge:** Servicio de AWS que facilita la conexión de aplicaciones mediante eventos, permitiendo la integración de sistemas internos y externos en tiempo real.
- **Route 53 Resolver:** Servicio de AWS que permite resolver y gestionar consultas DNS, incluyendo la posibilidad de aplicar reglas de seguridad mediante listas de dominios.
- **GuardDuty:** Servicio de AWS que proporciona detección de amenazas y monitoreo continuo para identificar actividades maliciosas o no autorizadas dentro de la cuenta de AWS.
- **Auto Scaling:** Funcionalidad que ajusta automáticamente la capacidad de los recursos de computación en la nube en función de la demanda, garantizando disponibilidad y optimización de costos.
- **New Relic:** Plataforma de observabilidad que permite monitorear el rendimiento de aplicaciones, infraestructura y servicios, identificando problemas en tiempo real.
- **CI/CD (Continuous Integration / Continuous Deployment):** Conjunto de prácticas de desarrollo de software que permiten la integración continua de cambios y su despliegue automatizado en diferentes entornos.

- **DevSecOps:** Enfoque que integra seguridad en cada fase del ciclo de vida del desarrollo de software dentro de la práctica DevOps.