UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas e Informática

“Implementación de una arquitectura de microservicios para mejorar el proceso de la gestión de clientes de la funeraria Descanso Eterno SAC, Chimbote”

Tesis para optar el título profesional de Ingeniero de Sistemas e Informática

TESISTA:

Bach. Mendoza Oviedo, Michael Nay

Cód. ORCID: 0000-0002-4580-3750

ASESOR:

Dr. Apestegui Florentino, Yim Isaías

DNI: 32541215

Cód. ORCID 0000-0003-2873-1748

NUEVO CHIMBOTE – PERÚ
2025

UNIVERSIDAD NACIONAL DEL SANTA

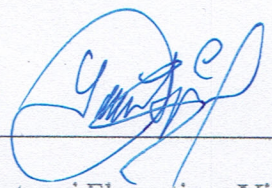
FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas e Informática

"Implementación de una arquitectura de microservicios para mejorar el proceso de la gestión de clientes de la funeraria Descanso Eterno SAC, Chimbote"

Tesis para optar el título profesional de Ingeniero de Sistemas e Informática

REVISADO Y APROBADO POR:



Dr. Apestegui Florentino, Yim Isaías

DNI: 32541215

Asesor

Cód. ORCID 0000-0003-2873-1748

NUEVO CHIMBOTE – PERÚ

2025

UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas e Informática

"implementación de una arquitectura de microservicios para mejorar el proceso de la gestión de clientes de la funeraria Descanso Eterno SAC, Chimbote"

Tesis para optar el título profesional de Ingeniero de Sistemas e Informática

REVISADO Y APROBADO POR EL JURADO EVALUADOR:

Dr. Sixto Diaz Tello
Codi. Orcid: 0000-003-3595-9441
Presidente

Ms. Camilo Ernesto Suarez Rebaza
Codi. Orcid: 0000-0002-6870-4296
Secretario

Dr. Yim Isaías Apéstegui Florentino
Codi. Orcid: 0000-0003-2873-1748
Integrante

NUEVO CHIMBOTE – PERÚ

2025

FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL INGENIERÍA DE SISTEMAS E INFORMÁTICA

ACTA DE SUSTENTACIÓN INFORME FINAL DE TESIS

A los veintinueve días del mes de agosto del año dos mil veinticinco, siendo las 11:00 am. En el aula s2 del Pabellón de la Escuela Profesional de Ingeniería Sistema e Informática-FI-UNS, se instaló el Jurado Evaluador designado mediante Resolución 378-2025-UNS-CFI, y de expedito según Resolución Decanal N° 572-2025-UNS-FI integrado por los docentes: **Dr. Sixto Diaz Tello (presidente), Ms. Camilo Ernesto Suarez Rebaza (secretario) y el Dr. Yim Isaías Apéstegui Florentino (Integrante)**, para dar inicio a la sustentación de la Tesis titulada **"IMPLEMENTACIÓN DE UNA ARQUITECTURA DE MICROSERVICIOS PARA MEJORAR EL PROCESO DE LA GESTIÓN DE CLIENTES DE LA FUNAERARIA DESCANSO ETERNO, CHIMBOTE "**, perteneciente al bachiller: **MENDOZA OVIEDO MICHAEL NAY con código de matrícula N°0201714055**, quien fue asesorado por el **Dr. Yim Isaías Apéstegui Florentino** , según Resolución Decanal N.º 118-2023-UNS-FI.

El Jurado Evaluador, después de deliberar sobre aspectos relacionados con el trabajo, contenido y sustentación del mismo, y con las sugerencias pertinentes en concordancia con el Reglamento General de Grados y Títulos, vigente, declaran aprobar:

BACHILLER	PROMEDIO VIGESIMAL	PONDERACIÓN
MENDOZA OVIEDO MICHAEL NAY	17	Bueno

Siendo las 12:00 pm del mismo día, se dio por terminado el acto de sustentación, firmando la presente acta en señal de conformidad.

Nuevo Chimbote, 29 de agosto de 2025

Dr. Sixto Diaz Tello
PRESIDENTE

Ms. Camilo Ernesto Suarez Rebaza
SECRETARIO

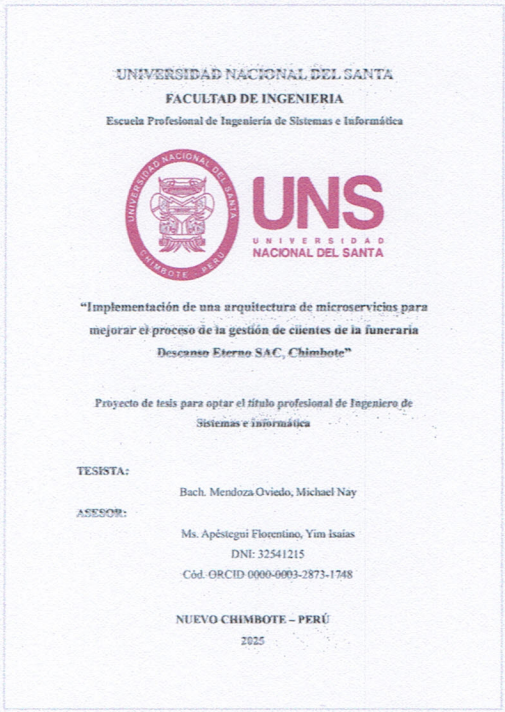
Dr. Yim Isaías Apéstegui Florentino
INTEGRANTE

Recibo digital

Este recibo confirma quesu trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	Michael Nay MENDOZA OVIEDO
Título del ejercicio:	Tesis Maestria 1
Título de la entrega:	INFORME FINAL DE TESIS
Nombre del archivo:	Proyecto_Tesis_Mendoza_Oviedo_Michael.docx
Tamaño del archivo:	5.95M
Total páginas:	168
Total de palabras:	23,661
Total de caracteres:	134,183
Fecha de entrega:	24-mar-2025 03:24p. m. (UTC-0500)
Identificador de la entrega:	2461613105



UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas e Informática

UNS UNIVERSIDAD NACIONAL DEL SANTA

"Implementación de una arquitectura de microservicios para mejorar el proceso de la gestión de clientes de la funeraria Descanso Eterno SAC, Chimbote"

Proyecto de tesis para optar el título profesional de Ingeniero de Sistemas e Informática

TESISTA:

Bach. Mendoza Oviedo, Michael Nay

ASESOR:

Ms. Apéstegui Florentino, Yim Isaias

DNI: 32541215

Cód. ORCID 0000-0003-2873-1748

NUEVO CHIMBOTE – PERÚ

2025

Informe Final de Tesis - MENDOZA-OVIEDO_MICHAEL 10-08-2025.docx

INFORME DE ORIGINALIDAD

12%	12%	0%	2%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.uns.edu.pe Fuente de Internet	8%
2	iydt.wordpress.com Fuente de Internet	2%
3	hdl.handle.net Fuente de Internet	1%
4	Submitted to Universidad de Guayaquil Trabajo del estudiante	1%
5	desire.webs.uvigo.es Fuente de Internet	1%
6	dl.dropboxusercontent.com Fuente de Internet	1%

Excluir citas	Activo	Excluir coincidencias	< 90 words
Excluir bibliografía	Activo		

10	Submitted to Universidad TecMilenio Trabajo del estudiante	<1 %
11	Submitted to ueb Trabajo del estudiante	<1 %
12	desire.webs.uvigo.es Fuente de Internet	<1 %
13	Submitted to Universidad Nacional del Santa Trabajo del estudiante	<1 %
14	Submitted to Corporación Universitaria Minuto de Dios, UNIMINUTO Trabajo del estudiante	<1 %
15	dokumen.pub Fuente de Internet	<1 %
16	dspace.ups.edu.ec Fuente de Internet	<1 %
17	repositorio.ucv.edu.pe Fuente de Internet	<1 %
18	Submitted to Instituto Superior de Artes, Ciencias y Comunicación IACC Trabajo del estudiante	<1 %
19	repositorio.espe.edu.ec Fuente de Internet	<1 %
20	repositorio.uti.edu.ec Fuente de Internet	<1 %

DEDICATORIA

A Dios y a mi familia, en especial a mis
padres Rosa Elena Oviedo Siancas y Santos
David Mendoza Meléndez, por el sacrificio,
el apoyo y estar siempre a mi lado durante mi vida
universitaria en momentos buenos y difíciles de
afrontar.

A mi fiel compañera de vida Geanella Reyna Pascual,
por su compañía en todas las madrugadas de estudio
e investigación, su esfuerzo y por el apoyo
permitieron que se concluya este trabajo
de investigación.

A mi asesor, que sin su apoyo esto no sería
posible; a todos los docentes por su paciencia y
compromiso.

AGRADECIMIENTO

Primero, agradecer a Dios por la salud y vida que me ha proporciona, así mismo a mi familia por su apoyo y consejos brindados durante el desarrollo de esta investigación; a mi asesor por los consejos y conocimientos inculcados en el desarrollo del presente informe y por la paciencia para guiarme de principio a fin. A los docentes que se dieron el tiempo para apoyarme cuando tenía alguna consulta; y a los trabajadores de la Funeraria Descanso Eterno SAC por la amabilidad y disposición de brindarme la confianza e información requerida para lograr los objetivos y así concluir con éxito el presente proyecto.

INDICE

ÍNDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

En la funeraria Descanso Eterno S.A.C., ubicada en Chimbote, se identificó que la gestión de clientes presentaba ineficiencias debido al uso de procesos manuales, la limitada disponibilidad de información y la baja digitalización de servicios. El objetivo de la investigación fue desarrollar e implementar un sistema de gestión basado en arquitectura de microservicios para optimizar la eficiencia operativa y mejorar la atención al cliente.

El estudio fue de tipo aplicado, con diseño preexperimental y enfoque cuantitativo, empleando la metodología ágil Scrum para el desarrollo. El sistema se implementó en Amazon Web Services (AWS) utilizando base de datos NoSQL DynamoDB, funciones Lambda y una interfaz en Python Flask. La recolección de datos se realizó mediante encuestas a 35 participantes antes y después de la implementación, y se aplicaron pruebas estadísticas de normalidad (Shapiro-Wilk) y T-Student para muestras relacionadas.

Los resultados evidenciaron mejoras estadísticamente significativas ($p < 0.001$) en los tres indicadores evaluados: reducción del 49.4% en el tiempo de respuesta (de 27.47 s a 13.90 s), incremento del 72.2% en el grado de satisfacción de empleados y clientes (aumento de 3.25 puntos sobre un valor inicial de 4.50) y aumento del 89.3% en la disponibilidad de la información (de 4.01 a 7.59 puntos).

Se concluye que la arquitectura de microservicios es una solución escalable, eficiente y moderna para digitalizar y optimizar la gestión de clientes en el sector funerario, logrando mejoras sustanciales en el servicio y en la percepción de los usuarios.

Palabras clave: microservicios, gestión de clientes, Scrum, AWS, DynamoDB,

ABSTRACT

This In Descanso Eterno S.A.C., a funeral service company located in Chimbote, customer management inefficiencies were identified due to the use of manual processes, limited information availability, and low service digitalization. The objective of this research was to develop and implement a customer management system based on a microservices architecture to optimize operational efficiency and improve customer service.

This study was applied in nature, with a pre-experimental design and a quantitative approach, employing the agile Scrum methodology for development. The system was deployed on Amazon Web Services (AWS) using a NoSQL DynamoDB database, Lambda functions, and a Python Flask user interface. Data collection was carried out through surveys administered to 35 participants before and after implementation, and statistical analyses included normality tests (Shapiro-Wilk) and paired-sample T-Student tests.

The results showed statistically significant improvements ($p < 0.001$) in the three evaluated indicators: a 49.4% reduction in response time (from 27.47 s to 13.90 s), a 72.2% increase in employee and customer satisfaction (an increase of 3.25 points from an initial value of 4.50), and an 89.3% increase in information availability (from 4.01 to 7.59 points).

It is concluded that the microservices architecture constitutes a scalable, efficient, and modern solution to digitize and optimize customer management in the funeral services sector, achieving substantial improvements in both service performance and user perception.

Keywords: microservices, customer management, Scrum, AWS, DynamoDB, Python Flask, T-Student

CAPÍTULO I

INTRODUCCIÓN

1.1. Descripción del problema

La funeraria Descanso Eterno SAC, ubicada en Chimbote, se encuentra en una disyuntiva al nivel operativo y de gestión de clientes. Su dependencia de procesos manuales, como el registro de solicitudes en cuadernos, el manejo de cartera de clientes en archivos registrados manualmente por los empleados a cargo y la falta de automatización en áreas clave, como la atención al cliente a través de sus principales canales digitales como su actual página de Facebook, obstaculizan su eficiencia y capacidad de respuesta. Esta situación se traduce en una ineficiencia operativa y dificultad para identificar oportunidades de mejora, caracterizada por retrasos en la comunicación y falta de información clara sobre los servicios ofrecidos.

Además, la funeraria enfrenta dificultades para atraer y retener clientes de manera efectiva. La ausencia de herramientas de gestión digital limita la trazabilidad de los datos de los clientes y su historial de servicios, dificultando la personalización de la atención y la mejora continua del servicio. En consecuencia, la funeraria pierde oportunidades de negocio y se ve limitada en su crecimiento.

La falta de datos estructurados y herramientas de análisis impide a la funeraria evaluar el rendimiento de sus servicios, identificar patrones de comportamiento de los clientes y tomar decisiones basadas en evidencia. La falta de una infraestructura tecnológica que permita recopilar, analizar y procesar la información impide optimizar la relación con los clientes y tomar decisiones basadas en datos para mejorar su experiencia.

La implementación de una arquitectura de microservicios surge como una solución innovadora y estratégica para automatizar procesos clave, mejorar la comunicación con los clientes y fortalecer el análisis de datos. Esta modernización permitirá mejorar la eficiencia en la gestión de clientes, optimizar la toma de decisiones y potenciar la competitividad de la funeraria en un mercado en constante evolución.

1.2. ANÁLISIS DEL PROBLEMA

La funeraria Descanso Eterno SAC, ubicada en el distrito de Chimbote, presenta una serie de desafíos operativos y de gestión de clientes que limitan su potencial de crecimiento y la calidad de sus servicios. La ausencia de una arquitectura tecnológica moderna y la dependencia de procesos manuales generan ineficiencias que impactan negativamente tanto en la experiencia del cliente como en la productividad interna de la empresa. A continuación, se detallan las principales deficiencias y sus causas:

- **Procesos Manuales y Desorganizados:**
  - ✓ **Problema**: La funeraria registra información crucial, como datos de clientes y solicitudes de servicio, en cuadernos físicos y hojas de cálculo, lo que dificulta el acceso rápido a la información, aumenta el riesgo de errores y limita la capacidad de análisis de datos.
  - ✓ **Causa**: La ausencia de una plataforma centralizada para gestionar todos los aspectos de los servicios funerarios contribuye a la descoordinación.

- **Limitada Interacción Cliente-Funeraria:**
  - ✓ **Problema**: La interacción entre los clientes y las funerarias es limitada, lo que dificulta la comunicación efectiva, la comprensión de las necesidades del cliente y la prestación de un servicio personalizado.
  - ✓ **Causa**: La falta de una interfaz conversacional y herramientas de atención al cliente en tiempo real contribuye a la limitada interacción.

- **Procesos Manuales y Falta de Automatización:**
  - ✓ **Problema**: Muchos procesos en las funerarias son manuales, lo que puede resultar en errores, pérdida de tiempo y falta de eficiencia en la prestación de servicios.
  - ✓ **Causa**: La ausencia de sistemas automatizados para la gestión de clientes contribuye, el seguimiento de solicitudes y la comunicación con los usuarios genera retrasos, duplicación de esfuerzos y una respuesta lenta a las necesidades de los clientes.

- **Escasa Utilización de Tecnologías en las Sesiones de Asesoramiento:**
  - ✓ **Problema**: Los asesores funerarios a menudo no utilizan tecnologías en tiempo real durante las sesiones de asesoramiento, lo que podría afectar la calidad de la información proporcionada y la experiencia del cliente.
  - ✓ **Causa**: La falta de inversión y capacitación en el uso de herramientas tecnológicas impide la incorporación de tecnologías durante las interacciones con los clientes.

- **Limitada Capacidad de Análisis:**
  - ✓ **Problema**: La información dispersa en diferentes formatos y la falta de herramientas de análisis dificultan la identificación de patrones, tendencias y áreas de mejora en la gestión de clientes y la operación general de la funeraria.
  - ✓ **Causa**: La funeraria no cuenta con software especializado para analizar datos de clientes, como CRM (Customer Relationship Management) o herramientas de Business Intelligence, lo que limita su capacidad para extraer información valiosa y tomar decisiones basadas en datos.

- **Dificultad para Captar y Retener Clientes:**
  - ✓ **Problema**: La falta de una estrategia de marketing digital y herramientas de análisis impide a la funeraria comprender las necesidades y preferencias de sus clientes potenciales, así como evaluar la efectividad de sus esfuerzos de captación y retención.
  - ✓ **Causa**: La funeraria no tiene la capacidad de generar una estrategia digital, al no utilizar herramientas y sistemas automatizados con una arquitectura.

PROPUESTA DE SOLUCIÓN:

Para abordar estas deficiencias y mejorar la gestión de clientes en el sector de funerarias, se propone la implementación de una arquitectura de microservicios:

Centralización de la Gestión:

- ✓ **Solución**: Implementar microservicios que centralicen la gestión de servicios funerarios, desde la planificación hasta la entrega. Esto mejorará la coordinación interna y la eficiencia operativa.

Interfaz Conversacional y Atención al Cliente en Tiempo Real:

- ✓ **Solución**: Integrar una interfaz conversacional en la Web App para facilitar la comunicación directa y en tiempo real entre las funerarias y los clientes. Esto permitirá resolver dudas y mejorar la satisfacción del cliente.

Automatización de Procesos:

- ✓ **Solución**: Introducir microservicios y sistemas automatizados BI para eliminar procesos manuales, reducir errores y mejorar la eficiencia en la prestación de servicios.

Capacitación y Promoción de Tecnologías:

- ✓ **Solución**: Brindar capacitación a los asesores funerarios en el uso de tecnologías durante las sesiones de asesoramiento. Promover la inversión en herramientas tecnológicas para mejorar la calidad de la información y la experiencia del cliente.

Al implementar estas soluciones, se espera mejorar la satisfacción del cliente, optimizar la gestión de servicios funerarios y fortalecer la eficiencia operativa en el sector de funerarias del distrito de Chimbote.

1.3. FORMULACIÓN DEL PROBLEMA

¿De qué manera la implementación de una arquitectura de microservicios mejorará el proceso de la gestión de clientes de la funeraria Descanso Eterno SAC, Chimbote?

1.4. OBJETIVOS

1.4.1. Objetivo General

Mejorar el proceso de la gestión de clientes de la funeraria Descanso Eterno en el distrito de Chimbote, mediante la implementación de una arquitectura de microservicios, con el propósito de mejorar la eficiencia operativa, optimizar la gestión de clientes y modernizar los servicios funerarios, contribuyendo así al desarrollo y adaptación del sector a las demandas contemporáneas.

1.4.2. Objetivos Específicos

1. Elaboración de documento de análisis de requerimientos con especificaciones y prioridades identificadas.
2. Elevar el grado de satisfacción de los empleados.
3. Aumentar la disponibilidad de la información.
4. Prototipo funcional con al menos 80% de las funcionalidades principales completadas.
5. Despliegue exitoso en el entorno de producción y funcionamiento sin incidencias críticas en el primer mes de operación.
6. Reducción del tiempo promedio de atención al cliente en al menos un 30%.
7. Incremento en la satisfacción del cliente en al menos un 20%, medido mediante encuestas post-implementación.

1.5. Formulación de Hipótesis

La implementación de una arquitectura de microservicios si mejora el proceso de la gestión de clientes de la funeraria Descanso Eterno S.A.C, Chimbote.

1.5.1. Hipótesis Alterna

La implementación de una arquitectura de microservicios mejora significativamente la gestión de clientes en la funeraria Descanso Eterno S.A.C.

1.5.2. Hipótesis Nula

La implementación de una arquitectura de microservicios no genera mejoras significativas en la gestión de clientes de la funeraria Descanso Eterno S.A.C.

1.6. JUSTIFICACIÓN DEL PROYECTO

1.6.1. Teórica

Desde este trabajo considero que se aporta a nivel teórico al demostrar que la arquitectura de microservicios puede ser aplicada de manera efectiva en organizaciones pequeñas o medianas que no necesariamente pertenecen al rubro tecnológico. En este caso particular, he logrado implementar esta arquitectura en una funeraria, lo cual representa un enfoque poco explorado académicamente, sobre todo en entornos locales como el de Chimbote.

Lo innovador de esta propuesta es que no se queda solo en el diseño conceptual de los microservicios, sino que se lleva a la práctica utilizando tecnologías modernas basadas en la nube. Para ello, se emplearon servicios de Amazon Web Services (AWS) como AWS Lambda para la ejecución de funciones sin servidor, API Gateway para la exposición de los endpoints REST, y DynamoDB como base de datos NoSQL altamente escalable. Además, se trabajó con métodos HTTP como POST, GET, PUT y DELETE, desarrollando toda la lógica en Python, lo que permitió construir un sistema ligero, eficiente y adaptable a cambios futuros.

1.6.2. Social

La presente investigación, representa una respuesta directa a las necesidades actuales de este sector, marcando un avance significativo en términos de eficiencia y calidad en la prestación de servicios funerarios.

En primer lugar, cabe destacar que el sector funerario desempeña un papel crucial en la sociedad, proporcionando servicios esenciales en momentos de dolor y pérdida. Sin embargo, la gestión tradicional de clientes en estas empresas a menudo se enfrenta a desafíos significativos, como la falta de coordinación efectiva, la pérdida de información importante y la dificultad para adaptarse a las expectativas cambiantes de los clientes.

La implementación de una arquitectura de microservicios aborda directamente estas problemáticas, permitiendo una gestión más eficiente y centrada en el cliente. La incorporación de microservicios facilita la escalabilidad del sistema, lo que se traduce en una mayor flexibilidad para adaptarse a las particularidades de cada funeraria en el Distrito de Chimbote.

Esto no solo mejora la calidad del servicio, sino que también optimiza los recursos disponibles, maximizando la capacidad de las empresas para atender a un mayor número de familias de manera simultánea.

1.6.3. Económica

El presente estudio propone una solución tecnológica orientada a optimizar la gestión de clientes en la funeraria Descanso Eterno S.A.C., en Chimbote, considerando un enfoque económico sustentado en el uso eficiente de recursos tecnológicos. Se emplearán lenguajes de programación de libre acceso como Python y JavaScript, así como entornos de desarrollo gratuitos. Además, se utilizarán funciones *SERVERLESS* a través de AWS Lambda, lo que permitirá desplegar microservicios de forma escalable sin necesidad de mantener servidores físicos o virtuales activos constantemente, reduciendo así los costos operativos. Esta estrategia técnica permite implementar una arquitectura sostenible desde el punto de vista económico y alineada con las mejores prácticas actuales en desarrollo de software.

Estas herramientas proporcionan un entorno de desarrollo completo sin la necesidad de inversiones financieras adicionales, permitiéndonos concentrarnos en la creación de una solución tecnológica de calidad sin sacrificar la eficiencia económica.

1.6.4. Tecnológica

Este estudio propone una arquitectura de microservicios en la funeraria Descanso Eterno SAC permitirá una transformación digital significativa con sus clientes y servicios funerarios. En la actualidad existen procesos manuales y no emplean algún sistema, lo que genera ineficiencias operativas. Con la adopción de microservicios en la nube, se logrará una infraestructura escalable y flexible que permitirá la integración de múltiples sistemas, garantizando la disponibilidad, confiabilidad y resiliencia del sistema.

1.6.5. Técnica

La solución propuesta se basa en una arquitectura de microservicios desplegada en un entorno de computación en la nube. Esto garantiza la modularidad, escalabilidad y mantenimiento independiente de cada componente del sistema, permitiendo que cada microservicio funcione de manera autónoma y pueda ser actualizado o mejorado sin afectar el sistema en su conjunto.

La arquitectura aprovechará tecnologías como Amazon Web Services (AWS) para la ejecución de funciones sin servidor con AWS Lambda, almacenamiento en DynamoDB o MongoDB, y la orquestación de datos con API Gateway. También se integrarán herramientas de monitoreo y trazabilidad como AWS X-Ray y CloudWatch para optimizar la gestión y rendimiento del sistema. La adopción de estándares y buenas prácticas en el desarrollo de software garantizará la seguridad, interoperabilidad y sostenibilidad del sistema a largo plazo.

1.6.6. Operativa

En términos operativos, este estudio buscar automatizar procesos manuales que actualmente ralentizan la gestión y afectan la eficiencia de la funeraria. Al eliminar tareas repetitivas y optimizar la gestión de clientes, se logrará una mejora en los tiempos de respuesta y un mejor control sobre las solicitudes de los clientes.

La integración de una Arquitectura de Microservicios ayudara a los empleados acceder a la información de los clientes y los servicios contratados sin demoras, evitando errores administrativos y mejorando la coordinación entre diferentes áreas de la empresa. Asimismo, se optimizará la comunicación con los clientes mediante notificaciones automáticas y atención personalizada a través de una Web App con servicios en la nube.

1.6.7. Personal

Desde una perspectiva personal, la implementación del sistema beneficiará tanto a los empleados de la funeraria como a los clientes. Para los empleados, se reducirá la carga de trabajo derivada de la gestión manual de datos y se mejorará su productividad al contar con herramientas tecnológicas que faciliten la organización y el acceso a la información en tiempo real.

Para los clientes, la mejora en los procesos internos de la funeraria se traducirá en una experiencia más eficiente y personalizada. Se garantizará una atención más rápida y efectiva, lo que impactará en la satisfacción y confianza del usuario. Además, el acceso a la información sobre los servicios funerarios será más transparente y accesible, permitiendo una gestión más eficiente de sus necesidades en momentos críticos.

1.7. ALCANCE

El presente estudio se enfoca en la implementación de una solución tecnológica basada en una arquitectura de microservicios para optimizar la gestión de clientes en la funeraria Descanso Eterno SAC, ubicada en Chimbote. La solución utiliza servicios en la nube para garantizar escalabilidad, seguridad y eficiencia operativa, alineándose con las necesidades específicas del negocio y los objetivos planteados.

Alcance Funcional

- Registro de Clientes: Permite almacenar y registrar la información de los clientes de manera centralizada.
- Gestión de Servicios Funerarios: Administra contrataciones, seguimiento y documentación de servicios ofrecidos.
- Gestión de Reservas: Organiza las reservas de recursos y servicios disponibles.
- Notificaciones: Sistema automatizado de envío de notificaciones por correo y SMS para empleados y clientes.
- Generación de Reportes: Proporciona reportes operativos y administrativos para la toma de decisiones.
- Historial de Clientes: Consolida la información de los servicios utilizados por cada cliente.

- Gestión de Inventario: Controla la disponibilidad de insumos y recursos necesarios para los servicios.

Alcance Técnico

- Despliegue de la arquitectura de microservicios en la nube utilizando el plan gratuito de AWS.
- Uso de servicios como AWS Lambda, API Gateway, DynamoDB (o MongoDB, si aplica), Amazon SNS, y otras herramientas complementarias.
- Implementación de políticas de roles y permisos mediante AWS IAM para garantizar la seguridad y el control de acceso.
- Conexión entre microservicios utilizando estándares de eventos JSON y APIs REST.

1.8. IMPORTANCIA

Esta investigación aborda un problema crítico en la gestión de clientes y procesos operativos de la funeraria Descanso Eterno S.A.C., ubicada en el distrito de Chimbote. Actualmente, la falta de automatización y modernización en la gestión de datos retrasa el tiempo de respuesta en las operaciones de la empresa para brindar un servicio eficiente y personalizado a las familias que atiende. Estos desafíos no solo afectan la experiencia del cliente, sino también la percepción pública del sector funerario en la región, lo que limita su competitividad y alcance.

La implementación de una arquitectura de microservicios moderna permitirá optimizar procesos clave como el registro de clientes, la gestión de servicios funerarios, las reservas y la generación de reportes, reduciendo tiempos de atención y mejorar la calidad en el servicio ofrecido. Además, el sistema propuesto fortalecerá la capacidad de análisis y toma de decisiones mediante herramientas de monitoreo y reportes centralizados, contribuyendo a una gestión más eficiente y organizada.

Esta investigación tiene un impacto significativo en dos niveles principales:

1) Nivel Empresarial: La funeraria Descanso Eterno S.A.C. podrá transformar su eficiencia operativa, modernizando sus procesos y aumentando su capacidad para atender a un mayor número de familias. Esto no solo mejorará la satisfacción de sus clientes, sino que también posicionará a la empresa como un referente en el sector funerario local.
2) Nivel Comunitario y Sectorial: Este modelo de solución es replicable para otras funerarias del distrito de Chimbote y puede servir como base para transformar la manera en que las empresas del sector funerario operan. Al implementar tecnologías innovadoras como la arquitectura de microservicios, las empresas pueden modernizarse, mejorar la percepción pública del sector y contribuir al bienestar de la comunidad al ofrecer servicios más rápidos y efectivos.

1.9. LIMITACIONES

Aunque esta investigación propone una solución robusta para optimizar la gestión de clientes en la funeraria Descanso Eterno S.A.C., se identifican las siguientes limitaciones que podrían influir en la implementación y evaluación del sistema:

Tamaño de la Población:

- La población objetivo del estudio estuvo conformada por todos los clientes y empleados administrativos de la funeraria Descanso Eterno S.A.C. que hicieron uso o gestionaron servicios funerarios durante el año 2024. Según los registros internos, la población total fue de N = 52 personas (20 clientes y 32 empleados administrativos)

Tamaño de la Muestra:

- Se utilizó un muestreo no probabilístico por conveniencia, seleccionando a los participantes que estuvieron disponibles y dispuestos a responder durante el periodo de recolección de datos. La muestra final estuvo compuesta por n = 35 personas, distribuidas en 20 clientes y 15 empleados administrativos.

Datos Históricos Limitados:

- La ausencia de una plataforma tecnológica unificada para administrar la información de clientes y los servicios funerarios dificulta realizar un análisis comparativo exhaustivo entre el desempeño del sistema actual y los métodos tradicionales. Esto podría influir en la evaluación del impacto a largo plazo.

Dependencia de Infraestructura Tecnológica:

- La solución propuesta depende de una infraestructura tecnológica sólida, incluyendo una conexión estable a internet y recursos computacionales adecuados para el procesamiento de datos. Interrupciones o limitaciones en esta infraestructura podrían impactar temporalmente la operación del sistema.

Tiempo de Adopción y Capacitación:

- La implementación del sistema requiere de una fase inicial de capacitación del personal administrativo, lo que puede generar una curva de aprendizaje que afecte temporalmente los tiempos de respuesta y la eficiencia durante las primeras etapas de uso.

Falta de Estudios Precedentes en el Sector:

- La aplicación de arquitecturas de microservicios en el sector funerario no ha sido ampliamente estudiada, especialmente en contextos locales como el de Chimbote. Esto limita la referencia a casos previos y las mejores prácticas aplicadas a empresas similares.

CAPITULO II:

MARCO TEÓRICO

2.1. ANTECEDENTES

2.1.1. INTERNACIONAL

Tesis 01

Autor: Manel Mena Vicente

Título: ARQUITECTURA DE MICROSERVICIOS PARA COMPONENTES DIGITALES EN LA WEB DE LAS COSAS

Lugar: Almería – España

Institución: Universidad de Almería

Titulación: Doctorado en Informática

Año: 2022

Resumen de la Investigación:

Según el autor Mena Vicente (2023), señala que el objetivo de esta investigación es el establecimiento de una propuesta para la gestión y el uso de dispositivos IOT que permitan integrarse en una arquitectura de microservicios, asi como también desarrollar un sistema de orquestación para la arquitectura de microservicios que permita la escalabilidad de servicios cuando sea necesario.

Relación con nuestra investigación:

El aporte que tiene la tesis doctoral "Arquitectura de Microservicios para Componentes Digitales en la Web de las Cosas", comparten un enfoque común en la aplicación de la arquitectura de microservicios para resolver problemas de gestión y mejorar la eficiencia en diferentes contextos, también en la tesis doctoral, se busca facilitar la interacción de los usuarios con dispositivos IOT a través de los Digital Dice, mientras que en este presente estudio se busca mejorar el nivel de operatividad en el proceso de gestión de clientes de la funeraria.

La tesis doctoral puede servir como referencia para comprender cómo esta arquitectura se ha aplicado con éxito en otros contextos y cómo puede adaptarse a nuestro estudio.

Tesis 02

Autor: David Luque Fuentes

Título: APPLICATION OF SOFTWARE TOOLS FOR THE ADAPTATION OF A CRM TO THE MANAGEMENT OF CLEANING PRODUCTS

Lugar: Madrid – España

Institución: Universidad Politécnica de Madrid

Titulación: European Master in Software Engineering

Año: 2023

Resumen de la Investigación:

Según el autor Luque Fuentes (2023) señala y describe que el objetivo de este estudio es desarrollar e implementar una solución basada en Salesforce para optimizar las operaciones comerciales y optimizar la gestión de clientes de una empresa de venta de productos de limpieza a razón de los negocios actuales y el crecimiento de demandas de canales digitales, en donde los clientes buscan una forma de hacerlos sentir más cercanos a la interacción con los servicios que ofrece distintas empresas, asimismo el autor señala que trabajo con un marco ágil de trabajo para la implementación del proyecto denominado SCRUM, nos ayuda a trabajar de manera más eficiente los proyectos como una arquitectura de aplicaciones en sistemas distribuidos o en la nube.

Relación con nuestra investigación:

El aporte que tenemos de este estudio sobre nuestra investigación consiste en brindarnos una guía sobre herramientas desarrolladas en Salesforce para optimizar la gestión de clientes, sobre cómo desarrollar un CMR basado en las últimas tendencias de tecnología, como por ejemplo utilizando un marco metodológico y ágil como Scrum para realizar la implementación, el uso de herramientas como Visual Studio Code, Oracle CX Cloud, lenguajes de programación como Java.

Definitivamente esto nos ayudara a comprender mucho más el enfoque sobre desarrollar un CRM para el sector de Funerarias.

Tesis 03

Autores: Jorge Adrián Gómez González

Jailene Fiorella Ramírez Madriz

Título: IMPLEMENTAR UNA HERRAMIENTA CRM, ASI COMO LA EVALUACIÓN DE SUS RESULTADOS, QUE PERMITEN LA MEJORA EN LA GESTIÓN DE LA CONSTRUCTORA COMACKEN SRL

Lugar: San Pedro – Costa Rica

Institución: Universidad Latina De Costa Rica

Titulación: Licenciatura en Tecnologías de Información

Año: 2022

Resumen de la Investigación:

Según los autores Gómez Gonzáles & Ramírez Madriz (2022)**,** mencionan en su estudio de investigación que la entrega de productos de calidad ya no es suficiente en la actualidad, brindar un servicio excepcional y una atención al cliente de calidad es necesario, la creación de un CRM le da un valor agregado a el apoyo de agilizar las nuevas tendencias como la IA o RPA, por tal motivo sostiene que diseñar, implementar e implantar esta solución, mejora el rendimiento de utilidades y la minimización de costos.

Relación con nuestra investigación:

El estudio aporta valiosos insights para nuestra investigación, primero destaca la importancia del por qué el analizar, diseñar, implementar un Customer Relationship Management (CRM), señalando que mejora la gestión es escalable verticalmente a las nuevas tendencias de tecnología, asimismo nos ayuda a entender que es importante contar con una arquitectura de microservicios, que generalmente se puede construir desde cero con sistemas distribuidos y servidores, pero también menciona que existen servicios de arquitecturas implementadas como servicios as a service, algunos de los proveedores más éxitos son de Google, Microsoft o Amazon, por lo tanto esta información es de suma importancia, porque nos permitirá analizar que proveedor de servicio en la nube nos podría ayudar con el objetivo de esta investigación.

Tesis 04

Autores: Alejandra Carmona Fuentes

Título: ANÁLISIS DEL PROCESO DE COMUNICACIÓN DE CRM CON ARQITECTURAS DE SOFTWARE EXTERNAS UTILIZANDO OPEN APIS

Lugar: Toluca – México

Institución: Universidad Autónoma de México

Titulación: Licenciatura en Ingeniería de Computación.

Año: 2022

Resumen de la Investigación:

Según los autores (Carmona Fuentes, 2022)**,** mencionan en su estudio de investigación que la entrega de productos de calidad ya no es suficiente en la actualidad, brindar un servicio excepcional y una atención al cliente de calidad es necesario, la creación de un CRM le da un valor agregado a el apoyo de agilizar las nuevas tendencias como la IA o RPA, por tal motivo sostiene que diseñar, implementar e implantar esta solución, mejora el rendimiento de utilidades y la minimización de costos.

Relación con nuestra investigación:

El estudio aporta valiosos insights para nuestra investigación, primero destaca la importancia del por qué el analizar, diseñar, implementar un Customer Relationship Management (CRM), señalando que mejora la gestión y es escalable verticalmente a las nuevas tendencias de tecnología, nos da una guía sobre los puntos clave como la IA o RPA, que funcionaron en el proyecto del autor, ya que en la actualidad la IA generativa está revolucionando los canales, sitios web o eCommerce.

2.1.2. NACIONAL

Tesis 01

Autor: Kevin Yahir Calle Arévalo

Título: Aplicación de BPM/CRM como estrategia para la gestión comercial en la Empresa de Calzado Multinegocios KCM S.A.C.

Lugar: Tarapoto – Perú

Institución: Universidad Cesar Vallejo

Titulación: Licenciatura en Ingeniería de Sistemas.

Año: 2023

Resumen de la Investigación:

Según (Calle Arevalo, 2023)**,** en la empresa de calzado Multinegocios KCM S.A.C. tiene como objetivo impulsar la Gestión Comercial. La investigación, de enfoque aplicado y diseño experimental preexperimental, utiliza como instrumento de recolección de datos una ficha de registro con dos indicadores clave: Margen Operacional de Utilidad y Margen Bruto. La población de estudio consiste en 60 registros de atención, con una muestra probabilística de 33 registros.

Los resultados concluyen que la solución implementada, basada en las estrategias BPM-CRM y aplicando la Metodología XP, efectivamente impulsa la Gestión Comercial de Multinegocios KCM SAC. Se sugiere la expansión de estas soluciones a otras áreas de la empresa, aprovechando los conceptos de Gestión de Procesos de Negocio (BPM) y Gestión de Relaciones con los Clientes (CRM), respaldados por la Metodología XP para obtener beneficios adicionales en los procesos empresariales.

Relación con nuestra investigación:

El estudio ofrece valiosas perspectivas para nuestro proyecto, enfocada en impulsar la Gestión mediante una herramienta, utiliza un diseño experimental preexperimental y una ficha de registro como instrumento de recolección de datos, incorporando indicadores clave como el Margen Operacional de Utilidad y el Margen Bruto.

Los resultados destacan la efectividad de la solución implementada, basada en las estrategias BPM-CRM y la Metodología XP, en mejorar la gestión comercial de Multinegocios KCM SAC. Esta conclusión respalda la relevancia de aplicar conceptos de Gestión de Procesos de Negocio (BPM) y Gestión de Relaciones con los Clientes (CRM) en nuestro contexto, brindando una guía valiosa para la expansión de estas soluciones a otras áreas de la empresa y la obtención de beneficios adicionales en los procesos empresariales.

Tesis 02

Autores: Karen Leslie Chávez Delgado

Víctor Alfredo Cruzado Hoyos

Título: Impacto del uso de un CRM para el control de clientes de la empresa transportes & negocios BICE EIRL.

Lugar: Cajamarca – Perú

Institución: Universidad Privada del Norte

Titulación: Licenciatura en Ingeniería Empresarial.

Año: 2022

Resumen de la Investigación:

Según los autores (Calle Arevalo, 2023)**,** se destaca que la aplicación de El autor de la tesis se propuso estudiar el impacto de la implementación de una herramienta de tecnología, específicamente un Customer Relationship Management (CRM), en la empresa Transportes & Negocios BICE EIRL. El objetivo principal era mejorar el control de clientes y abordar problemas relacionados con el manejo incorrecto de la información. La investigación se llevó a cabo mediante entrevistas y pruebas, incluyendo un pre y un post test de la implementación del CRM. Se utilizó un enfoque cuantitativo y cuasi experimental, manipulando la variable dependiente, con una profundidad explicativa en los resultados obtenidos.

La investigación arrojó resultados positivos, demostrando una reducción significativa (del 80% al 20%) en la pérdida de información de clientes. La encuesta realizada a 24 clientes reveló un cambio positivo del 60% de desacuerdo a un 52% de aceptación en cuanto al trato ético y profesional post-implementación del CRM Bitrix 24. Se concluyó que la investigación beneficia considerablemente a la empresa, ya que la información se encuentra más actualizada, ordenada y de fácil acceso. La herramienta CRM facilita estrategias de venta más profundas y contribuye a la retención de clientes al proporcionar información exacta y precisa. Las palabras clave asociadas son: Control de Clientes, CRM, Bitrix 24.

Relación con nuestra investigación:

El estudio ofrece valiosas perspectivas para nuestro proyecto, enfocada en impulsar la Gestión mediante una herramienta, utiliza un diseño experimental preexperimental y una ficha de registro como instrumento de recolección de datos, incorporando indicadores clave como el Margen Operacional de Utilidad y el Margen Bruto. Los resultados destacan la efectividad de la solución implementada, basada en las estrategias BPM-CRM y la Metodología XP, en mejorar la gestión comercial de Multinegocios KCM SAC. Esta conclusión respalda la relevancia de aplicar conceptos de Gestión de Procesos de Negocio (BPM) y Gestión de Relaciones con los Clientes (CRM) en nuestro contexto, brindando una guía valiosa para la expansión de estas soluciones a otras áreas de la empresa y la obtención de beneficios adicionales en los procesos empresariales.

2.1.3. LOCAL

Tesis 01

Autor: Rebaza Llontop Julio Cesar

Título: Implementación de un sistema de información CRM en el MINIMARKET CHRISS para la mejora en atención al cliente – Chimbote.

Lugar: Chimbote – Perú

Institución: Universidad Católica de los Ángeles

Titulación: Licenciatura en Ingeniería de Sistemas

Año: 2021

Resumen de la Investigación:

Según el autor (Rebaza Llontop, 2021)**,** tiene como objetivo principal implementar un sistema de información CRM con el fin de mejorar la atención al cliente, para ello se utilizó una encuesta formulada en Google Forms, dirigida a clientes frecuentes, con una población de 329 y una muestra de 7, donde se destaca los resultados obtenidos mediante tablas y gráficos de dos dimensiones, elevando la calidad de servicio de buena a excelente, para hacer frente a la competencia global y adaptarse a las cambiantes condiciones del mercado, las organizaciones han incrementado su atención y recursos en la Gestión de Procesos de Negocio (BPM). Se mencionan las herramientas de Análisis de Procesos de Negocio (BPA) como componentes clave para iniciativas de mejora de procesos y la implementación de programas de BPM, proporcionando medios para realizar un análisis detallado de los procesos de una organización, incluyendo modelado, simulación y publicación de los procesos.

Relación con nuestra investigación:

El estudio nos ofrece una guía sobre cómo mejorar la gestión con los clientes, también nos comparte una forma de instrumentos de investigación que nos servirá como ejemplo de recopilación de información, luego recopilarlos y realizar nuestro estudio científico mediante gráficos y tablas. Este estudio nos comparte una guía de herramientas que nos servirán para la fase metodológica de nuestro proyecto como, por ejemplo, la gestión de procesos mediante BPM o herramientas de análisis de procesos de negocio como BPA, la cual puede ser crucial tanto para la implementación del sistema CRM, como para la optimización general.

En resumen, nuestra investigación se alinea con las estrategias y enfoques propuestos por el autor, mostrando una conexión significativa en la búsqueda común de mejorar la atención al cliente y adaptarse a las demandas del entorno empresarial actual.

2.2. MARCO CONCEPTUAL

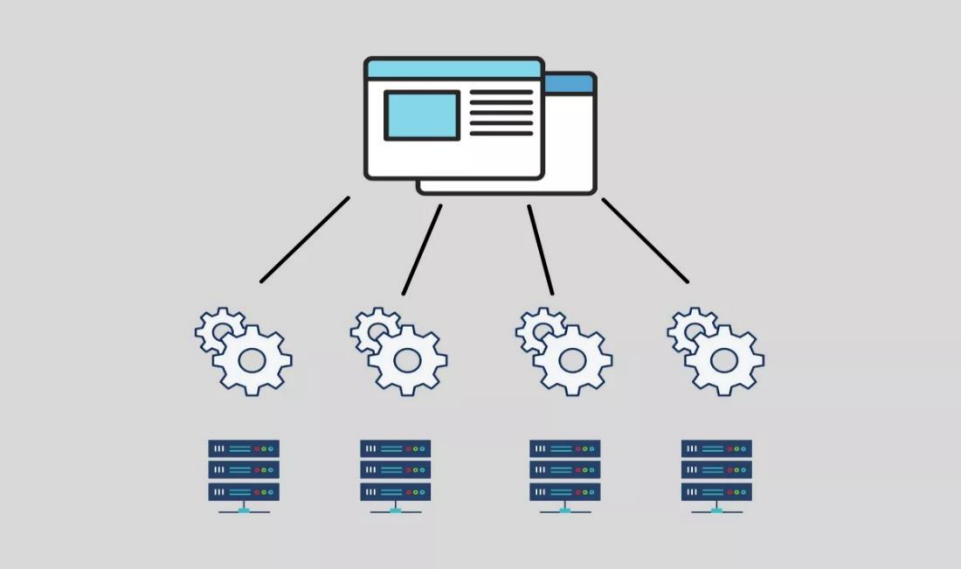
2.2.1. Arquitectura de Microservicios

Desde mi concepto y análisis, la arquitectura de software es la parte fundamental de como diseñamos, analizamos y resolvemos un problema mediante el uso de marcos y procedimientos abstractos que comprenden diferentes flujos de procesos, no obstante, si pensamos como arquitectos de software, es crucial estar preparados para abordar y resolver problemas de manera eficaz, garantizando que las soluciones sean robustas, escalables y mantenibles. Para profundizar más sobre la arquitectura de software, analizaremos la definición de algunos autores.

Por ejemplo, según el autor Serrano Valero (2022), señala que una parte importante de la arquitectura es el momento en que definimos en nuestro equipo la forma en que debemos implementar cada módulo, la elección del framework empleado para la persistencia de datos, la comunicación de capas, el uso unificado de patrones en el sistema.

Figura 1:

Esquema de Arquitectura de microservicios



Nota: https://www.arrobasolutions.com/arquitectura-de-microservicios-que-es-y-cuales-son-sus-ventajas/

También es importante mencionar que al momento de definir una arquitectura se debe tener en cuenta la planificación de cada recurso, también encontramos otros conceptos, por ejemplo, el autor (Albertos Gómez, 2018), menciona que los servicios implementados en una arquitectura de microservicios pueden estar implementados en diferentes lenguajes de programación y usar distinta tecnología de almacenamiento.

De nuestro análisis y experiencia profesional, la arquitectura nace a partir de grandes problemas que a lo largo del tiempo muchas empresas de tecnología se han enfrentado, por ejemplo, en los primeros años, los famosos software eran monolitos, que hasta en la actualidad muchos servicios de empresas dependen de ella, pero a lo largo del tiempo, se enfrenta a grandes problemas como la operabilidad diaria de algún servicio, esto sucede por problemas de una arquitectura que ya es obsoleta y que gracias a grandes soluciones, hace algunos años ya se habla de microservicios.

También el autor Serrano Valero (2022) menciona que la arquitectura de software surge para resolver un problema de software planteado, pero con el tiempo evolucionan y se van añadiendo nuevos problemas a resolver.

2.2.2. Web App

Podríamos definir que una Web App o normalmente conocido como aplicación web, es una herramienta que se ejecuta dentro de un browser o navegador, cuando escribimos una dirección IP o nombre de dominio, se ejecutan aplicaciones web, páginas webs informativas, aplicaciones web estáticas, aplicaciones web dinámicas, aplicaciones web progresivas, existe una cierta variedad de aplicaciones web que se ejecutan diariamente en el mundo del internet, por lo tanto podríamos definir de esta forma lo que es una Web App, pero sin suda vamos a escuchar a otros autores.

Por otro lado, según Amazon (2023) menciona que una aplicación web es un software que se ejecuta en el navegador web para comunicarse con los clientes cuando lo necesiten y de una forma segura, intercambiar información, realizar compras por internet, buscar algún producto, mensajería instantánea, permitiendo acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software.

2.2.3. Microservicios

Para poder encontrar la definición de microservicios, es necesario tener la opinión de otros autores, por ejemplo, los autores (López & Maya, 2017) mencionan que Microservicios es un enfoque de una aplicación única como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso y mecanismos ligeros de comunicación, a menudo un recurso de una interfaz de programación de aplicaciones (API).

Otros autores mencionan que los microservicios es un enfoque arquitectónico y organizativo del desarrollo de software en el que las aplicaciones están compuestas por pequeños servicios independientes que se comunican a través de una API bien definida y protocolos ligeros (Vera Rivera, Gaonas Cuevas, & Astudillo, 2019).

Por otro lado, los autores (Martín López & Liriano García, 2019, pág. 62), mencionan en su trabajo de grado titulado “Impacto de Arquitecturas de Microservicios en el Desarrollo Web”, que los microservicios ofrecen el poder de desplegar más rápido las aplicaciones, además su implementación es más sencilla.

Por otro lado, el autor Serrano Valero (2022) menciona en su investigación, que los principios claves de una arquitectura de microservicios son:

- **Descomposición en servicios Descomposición en Servicios**: Cada microservicio es una unidad independiente que aborda una funcionalidad específica del negocio. Esta descomposición facilita la comprensión, desarrollo, despliegue y escalado de aplicaciones complejas.

- **Despliegue Independiente**: Los microservicios pueden ser desplegados y actualizados de manera independiente, lo que permite a los equipos desarrollar, probar y lanzar nuevas funcionalidades sin afectar al resto del sistema.
- **Comunicación a través de APIs**: Los microservicios se comunican entre sí mediante APIs ligeras. Esto asegura que los servicios sean independientes del lenguaje de programación o la tecnología utilizada, permitiendo la interoperabilidad y la flexibilidad en la elección de herramientas.
- **Organización en torno a Capacidades de Negocio**: Los microservicios se alinean con las capacidades y procesos del negocio, lo que permite a los equipos trabajar de manera autónoma en servicios que representan dominios específicos del negocio.
- **Propiedad Descentralizada de Datos**: Cada microservicio puede tener su propia base de datos, evitando el acoplamiento entre servicios y permitiendo que cada uno gestione sus datos de la manera más adecuada para sus necesidades.
- **Automatización y DevOps**: La arquitectura de microservicios se apoya en prácticas de DevOps, incluyendo la automatización del despliegue, la infraestructura como código, la integración y la entrega continua, para mejorar la eficiencia y la calidad del desarrollo y operación del software.
- **Resiliencia y Escalabilidad**: Los microservicios están diseñados para ser resilientes a fallos. Cada servicio puede ser escalado de manera independiente, lo que facilita la gestión de cargas variables y la mejora de la disponibilidad del sistema.

Según las opiniones de los autores en los diferentes trabajos, llegamos a una conclusión, la arquitectura de microservicios representa un enfoque moderno y eficaz para el desarrollo de aplicaciones complejas y escalables. Al descomponer una aplicación en servicios independientes y alineados con capacidades de negocio, se promueve la agilidad, la escalabilidad y la resiliencia del sistema, aunque también se introducen desafíos que requieren una gestión y una infraestructura adecuadas.

2.2.4. Sistemas de gestión al Cliente

Gracias a los avances de la tecnología hace algunos años se hablaba de sistemas CRM que permitían mantener una comunicación con los clientes que tenían una relación de consumidor con algún tipo de negocio de cualquier empresa, prácticamente según mi visión en la actualidad se puede apreciar sistemas CRM con canales más sofisticados, con integraciones más avanzadas y canales con mayor comunicación hacia los clientes, en la actualidad eso hace que parezca muy sencillo.

Partiendo del análisis anterior, trataremos de conceptualizar la definición de un CRM, lo haremos mediante diferentes opiniones que otros autores consideran, por ejemplo, según el autor Guerola Navarro (2022) define a un CRM como una herramienta básica de gestión relacionado con el área de ventas dentro de una empresa de cualquier rubro, mientras que otros asumen que el CRM implica un rediseño de la organización y sus procesos para orientarlos al cliente, de forma que, por medio de la personalización de su oferta, la empresa pueda satisfacer las necesidades de los mismos.

Por su parte, (Montoya Agudelo & Boyero Saavedra, 2013) consideran que el CRM es una herramienta que permite que haya un conocimiento estratégico de los clientes y sus preferencias, esto permite que haya una visión integrada de los clientes a través de toda la organización, mientras que otro autor señala que es una estrategia o método enfocado a la selección y gestión de elevar el valor de los servicios de preferencia de los clientes (Lorenzo Romero, 2022, pág. 23).

2.2.5. Java

De acuerdo con el autor Pino Martínez & Vergara, (2020) menciona en su libro que Java es un lenguaje de programación que ha tomado fuerza por ser software independiente de la plataforma, donde un programa en Java funciona en cualquier computadora existente, siendo atractiva para los desarrolladores.

Según el autor Toro Bonilla (2022) menciona que Java está formado por un conjunto de declaraciones de tipos, interfaces y clases.

2.2.6. Python

Python es considerado unos de los lenguajes más populares de esta generación, la comunidad open source cada vez mejora el uso de librerías, cada autor tiene su concepto sobre este lenguaje, por ejemplo, el autor González Duque (2020) en su libro lo define como, un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido.

2.2.7. IA Generativa

Primero cuando hablamos de IA, nos hace imaginar sobre cosas como los robots con formas humanoides, pero el hecho de pensar asi es erróneo, porque desde mi punto de vista la IA intenta igualar la capacidad de pensar del ser humano, esto lo hace aplicando conceptos como algoritmos genéticos, redes neuronales u otros métodos que se aplican a este campo. Asi mismo, podemos decir que existe tipos de IA, las más conocidas son la IA no generativa y la IA generativa.

Partiendo de lo mismo es necesario la opinión de diferentes autores sobre la IA generativa, por ejemplo, el autor Franganillo (2023) menciona que la Inteligencia Artificial (IA) generativa es un campo en rápido avance que permite la producción automatizada de contenido textual, grafico, sonoro y audiovisual de alta calidad.

De esto podemos estar seguro, en la actualidad vemos que el campo de estudio y campos profesionales de la IA tiene un crecimiento vertical considerablemente alto, tanto que se ha abierto nuevos puestos de trabajo para la IA generativa.

El autor Franganillo (2023) también menciona que la IA es una disciplina científica y tecnológica que buscar crear sistemas capaces de resolver tareas que normalmente requieren de inteligencia humana.

2.2.7.1.GPT

Según el sitio web Amazon (2023) menciona que la tecnología GPT son una familia de modelos de redes neuronales que utilizan la arquitectura de transformadores y representan un avance clave en la inteligencia artificial (IA) que impulsa las aplicaciones de IA generativa, como ChatGPT. Los modelos GPT permiten a las aplicaciones crear texto y contenido (imágenes, música y más) de manera similar a como lo haría un ser humano y responder a preguntas de forma conversacional. Organizaciones de todos los sectores utilizan modelos GPT e IA generativa para bots de preguntas y respuestas, resumen de textos, generación de contenidos y búsqueda.

En conclusión, estamos ante un modelo más avanzando de red neuronal, donde ahora muchos desarrolladores están construyendo sus aplicativos de redes sociales, imágenes, chats u otros fines con esta tecnología, por lo tanto, no podemos obviar mencionar cuando se construye sobre todo un aplicativo Web dedicado a la relación con los clientes (CRM).

2.2.7.2.CHATBOT

Según los autores Casazola Cruz Alfaro Mariño, Burgos Tejada & Ramos More (2021) mencionan que un chatbot es una aplicación de software basada en inteligencia artificial (IA) que permite simular una conversación con una persona. Sin embargo, a pesar de sus múltiples beneficios, una cantidad sustancial de chatbot luchan por satisfacer a los usuarios, la demanda de chatbot también está creciendo y debido a esta demanda, las áreas de aplicación también se están expandiendo.

Por otro lado, el autor Fernández Ferrer (2023) menciona que los chatbot son agentes conversacionales que permiten mantener un dialogo entre un humano y una computadora, es decir son aplicaciones que tienen la capacidad de responder a mensajes simulando un dialogo entre personas.

Estas respuestas que nos pueden dar estas aplicaciones o programas basados en agentes conversacionales pueden provenir de una selección o conjunto de esquemas que se encuentran programados, gracias a que utilizan algoritmos basados en aprendizaje automático (machine

learning) o incluso en el procesamiento del lenguaje natural (Natural Language Processing, NLP) (Majid & Lakshmi, 2022).

2.2.8. AWS

Según Amazon (2025) menciona que, "*Amazon Web Services es una plataforma de servicios en la nube, posee y mantiene el hardware conectado a la red necesario para ofrecer computación, almacenamiento, bases de datos, análisis, redes, móviles, herramientas de desarrollo, herramientas de gestión, IoT, seguridad y Aplicaciones empresariales: bajo demanda, disponibles en segundos, con precios de pago por uso*".

2.2.8.1. Infraestructura de AWS

Según Amazon (2025) menciona que, "*La infraestructura de la nube de AWS se basa en las regiones y zonas de disponibilidad de AWS. Una La región de AWS es una ubicación física en el mundo en la que tenemos varias zonas de disponibilidad. Las zonas de disponibilidad constan de uno o más centros de datos discretos, cada uno con alimentación redundante, redes y conectividad, alojadas en instalaciones separadas*".

2.2.8.2. Lambdas

Dentro de los servicios que nos brinda, es necesario mencionar que este es uno de los más importantes al nivel de microservicios, según Amazon (2025) menciona que: "*Lambda ejecuta el código en una infraestructura de computación de alta disponibilidad y realiza todas las tareas de administración de los recursos de computación, incluido el mantenimiento del servidor y del sistema operativo, el aprovisionamiento de capacidad y el escalado automático, así como las funciones de registro, lo único que tiene que hacer es suministrar el código en uno de los tiempos de ejecución de lenguaje compatibles con Lambda*".

2.2.9. Base de Datos Relacionales

Las bases de datos juegan un papel muy importante en el mundo de los sistemas, aplicaciones webs, aplicaciones móviles y en cada proyecto informático, algunos autores tienen una apreciación diferente o igual, pero lleva al punto de que son sistema para almacenar nuestra información.

Por ejemplo, el autor Fernández Iglesias (2023) menciona que las bases de datos son colecciones organizadas de información que se almacenan y gestionan en un sistema informático, están diseñadas para gestionar eficientemente grandes volúmenes de datos estructurados y no estructurados y proporcionan una forma de almacenar y acceder a los datos para su procesamiento.

Como podemos subrayar a la definición anterior, el mundo de bases de datos es amplio, donde nos encontraremos con tipos de bases de datos desde relacionales y no relacionales. Según el autor Fernández Iglesias (2023) las bases de datos relacionales son la solución más habitual de sistema de base de datos, que organiza los datos en tablas con filas y columnas, donde las relaciones se definen mediante campos compartidos por dichas tablas y sobre todo las bases de datos de tipo relacional utilizan el lenguaje general de datos estructurados denominado SQL.

A continuación, mencionaremos algunos motores de bases de datos más robustos y escalables de los últimos tiempos de tipo relacional.

2.2.9.1. MySQL

Según el autor Nixon (2019) menciona que MySQL es probablemente el sistema de gestión de bases de datos para servidores web más conocido. Desarrollado a mediados de la década de los 90, ahora es una tecnología madura que impulsa muchos de los destinos de Internet más visitados en la actualidad.

De esto podemos destacar que MySQL ha sido y será por un buen tiempo, unos de los mejores motores de bases de datos, esto gracias a lo que menciona el autor Nixon (2019, pág. 161) que es una base de datos escalable que contiene una o más tablas, cada una de las cuales contiene registros o filas. Dentro de estas filas hay varias

columnas o campos que contienen los datos propiamente dichos, cada fila de la tabla es la misma que una fila de una tabla MySQL, cada columna de la tabla corresponde a una columna en MySQL, y cada elemento dentro de una fila es el mismo que el de un campo MySQL.

2.2.9.2. PostgreSQL

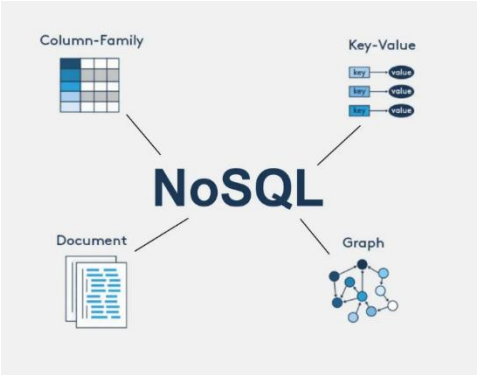
Según el autor Domínguez Chávez, (2019) menciona que PostgreSQL puede ser usado, modificado o distribuido para uso privado, comercial o Académico, PostgreSQL es un sistema de gestión de bases de datos orientado a objetos y relacional (ORDBMS) que hace énfasis en la extensibilidad y conformidad con los estándares, está liberado bajo la licencia free/open source PostgreSQL, similar a la licencia MIT y PostgreSQL está desarrollado por el Grupo de Desarrollo Global de PostgreSQL, el cual consiste en un grupo de voluntarios empleados y supervisados por empresas como Red Hat y EnterpriseDB.

2.2.10. Bases de Datos No Relaciones

También es necesario mencionar sobre las bases de datos no relacionales o cómo se las conoce Base de Datos NoSQL, sobre la importancia de su beneficio en el uso y valor que pueden ofrecer al momento de aplicarlas. Por lo tanto, algunos autores tienen su definición y mencionan su importancia, por ejemplo, los autores Ulín Ricárdez Pérez Vasconcelos, Gómez Domínguez & Castillo Romero (2023), mencionan en su estudio, que las bases de datos NoSQL han surgido como una alternativa a las bases de datos relacionales para manejar grandes cantidades de datos y superar los desafíos que plantean.

Figura 2*:*

Tipos de Base de Datos No Relacional



Fuente*:* https://itdconsulting.com

Analizando bajo el contexto de estos autores, podemos entender que este tipo de base de datos es una solución a las relacionales pero no solo por el tipo de desafíos que presentan sino por algo que va con el tiempo de acuerdo al uso u incremento, los autores Ulín Ricárdez Pérez Vasconcelos, Gómez Domínguez & Castillo Romero (2023), señalan también que, una de las ventajas de las bases de datos NoSQL es su capacidad para escalar horizontalmente, ya que ello le permite adaptarse de forma dinámica a los nuevos requerimientos en el incremento del número de usuarios concurrentes y altas prestaciones en capacidades de cómputo.

A comparación de las Bases de datos relacionales, las no relacionales, mantienen una semántica denominada BASE, donde:

- **BA (Básicamente disponible)**: una aplicación está lista para aceptar solicitudes de lectura/escritura todo el tiempo.
- **S (Estado blando)**: Es posible que los resultados no siempre se basen en datos coherentes (no hay garantía de coherencia).
- **E (Eventual Consistency):** Coherencia eventual: el sistema asegura que los datos serán consistentes en algún momento posterior.

También los autores Ulín Ricárdez Pérez Vasconcelos, Gómez Domínguez & Castillo Romero (2023), señalan los siguiente en su estudio sobre las bases NoSQL.

Las principales categorías de bases de datos NoSQL incluyen: Documentales, Columnares, Clave-Valor y Grafos.

- ✓ Bases de datos documentales. Los datos se almacenan en documentos basados en el formato JSON y que pueden contener cualquier número de campos y anidaciones.
- ✓ Bases de datos columnares. Los datos se almacenan por columnas en lugar de por filas, lo que permite una mayor eficiencia en la consulta de datos específicos.
- ✓ Bases de datos clave-valor. Los datos se almacenan en un diccionario de claves y valores lo que las hace ideales para aplicaciones de caché y almacenamiento en memoria.
- ✓ Bases de datos de grafos. Los datos se almacenan como nodos y aristas, lo que facilita la representación y consulta de relaciones complejas entre los datos.

Finalmente, los autores Ulín Ricárdez Pérez Vasconcelos, Gómez Domínguez & Castillo Romero (2023), señalan que es importante seleccionar la base de datos NoSQL adecuada para cada aplicación en función de sus necesidades específicas, porque cada base de datos NoSQL tiene sus propias ventajas y desventajas.

A continuación, vamos a mencionar algunos tipos de bases de datos NoSQL o no relaciones, sobre todo las más populares en estos últimos tiempos:

DynamoDB: Según Amazon Web Services (2025), Amazon DynamoDB es una Base de datos sin servidor sin necesidad de aprovisionamiento, parches ni administración de servidores. No hay software que instalar, mantener ni operar, y el mantenimiento no requiere tiempo de inactividad.

MongoDB: Según el portal de MONGODB (2025), menciona que es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado.

Apache CassandraDB: Según DataStaxt (2025)**,** menciona que Apache Cassandra es una base de datos NoSQL distribuida creada en Facebook y posteriormente lanzada como un proyecto de código abierto. Cassandra ofrece la disponibilidad continua (sin tiempo de inactividad), el alto rendimiento y la escalabilidad lineal

que requieren las aplicaciones modernas, al tiempo que ofrece simplicidad operativa y replicación sin esfuerzo en múltiples centros de datos y geografías. Puede manejar petabytes de información y miles de operaciones simultáneas por segundo, lo que permite a las organizaciones administrar grandes cantidades de datos estructurados en entornos híbridos y multinube.

CouchDB: Según IBM (2025)**,** menciona que Apache CouchDB es una base de datos de documentos NoSQL de código abierto que almacena datos en formatos basados en JSON. A diferencia de las bases de datos relacionales, CouchDB utiliza un modelo de datos sin esquema, que simplifica la gestión de registros en varios dispositivos informáticos, teléfonos móviles y navegadores web.

CosmosDB: Según Microsoft Azure (2025), menciona que base de datos en la nube de Microsoft Azure que permite crear aplicaciones inteligentes con datos en tiempo real. Es una base de datos distribuida y multimodelo que ofrece escalabilidad automática y tiempos de respuesta de milisegundos.

2.2.11. Framework

Según el autor Cristancho (2022), menciona en su redacción web que un framework es un entorno de trabajo preestablecido que incluye herramientas y características valiosas para acelerar el desarrollo de proyectos de programación. En esencia, facilitan el trabajo del programador, ya que proporcionan una base de datos conocida que ahorra tiempo en el desarrollo. Esto reduce significativamente los errores durante el desarrollo, ya que un framework es una colección de bibliotecas. En lugar de trabajar con comandos de un solo proyecto, se trabaja con una acumulación de proyectos.

2.2.11.1. Spring Boot

Según el autor Ramírez Pérez (2020) menciona que Spring Boot es una extensión de Spring Framework que sigue el enfoque de "Convención sobre configuración", que ayuda a construir aplicaciones basadas en Spring de manera rápida y fácil. Spring Boot está construido sobre Spring Framework y forma parte integral de este. Actualmente es el más utilizado dentro de conjunto de Framework, al igual que dentro del mundo Java.

El objetivo principal de Spring Boot es simplificar el desarrollo de aplicaciones por medio de la autogestión de un gran número de configuraciones, tareas y componentes que son necesarios para la ejecución de una aplicación. De esta manera, se logra que los desarrolladores se enfoquen en el desarrollo de la lógica de negocio del sistema (Ramírez Pérez, 2020).

2.2.11.2. TailwindCss

Según los autores Celi Párraga Boné Andrade & Mora Olivero (2023) mencionan que Tailwind CSS es un framework que ofrece un enfoque diferente que otros como Bootstrap. Tailwind CSS en realidad no tiene muchos componentes, sino clases de utilidad que aplicar directamente sobre el CSS, aunque también permite crear componentes, lo deja más del lado del desarrollador, que los podrá personalizar a su gusto. Tailwind CSS tiene la característica de ser muy maleable y adaptarse muy bien a lo que el desarrollador necesite. Con el framework puedes hacer builds de clases CSS totalmente personalizadas, que se parezcan o no a las que se ofrecen de manera predeterminada.

2.2.11.3. Bootstrap

Según los autores Celi Párraga Boné Andrade & Mora Olivero (2023) mencionan que Bootstrap es el framework CSS más popular, en 2020, usado en infinidad de proyectos de todo tipo. Cuando apareció creó una tendencia de framework basados en componentes, capaces de implementar temas de diseño completos y complejos, aportando mucha sencillez y agilidad al desarrollo CSS y dotando a los programadores de herramientas para crear diseños consistentes con poco esfuerzo. Durante sus años de existencia ha evolucionado mucho, incorporando novedades del estándar de CSS con rapidez y eliminando dependencias pesadas como jQuery que hoy en día es innecesario en la mayoría de los proyectos.

2.2.11.4. REACTJS

Según el autor Izura (2023)**,** menciona en su libro que React.JS una librería que permite la organización de una interfaz web en componentes reutilizables, aportando además una gestión más ágil de los elementos HTML mediante el uso de un DOM virtual.

Facilita, además, desarrollar aplicaciones para móviles mediante la librería React Native, cuyo aprendizaje resulta trivial una vez que se asimilan.

También resalto lo siguiente, React está siendo utilizado por las compañías más importantes como Airbnb, Apple, Dropbox, Instagram, Netflix, Twitter, Tesla, Uber y se ha convertido en una de las apuestas más relevantes y recomendables de los últimos tiempos, ya que cuenta con un completo ecosistema de módulos, herramientas y componentes capaces de ayudar a construir cualquier desarrollo avanzado con relativamente poco esfuerzo (Izura, 2023).

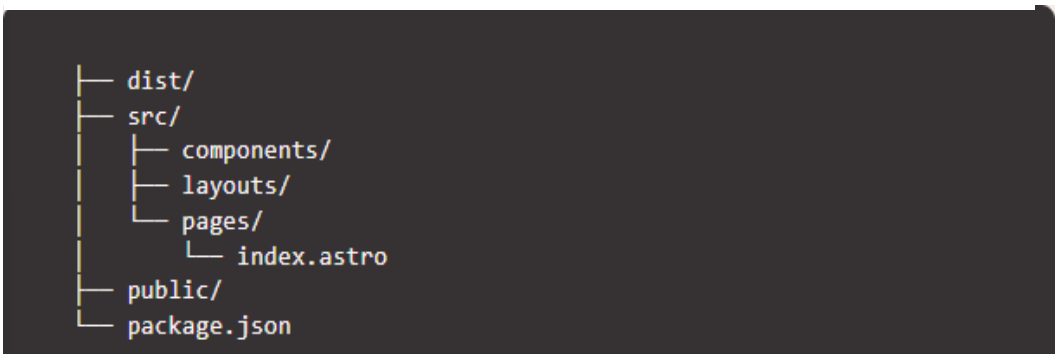
2.2.11.5. ASTRO

Como podemos comprender a lo largo del estudio hasta ahora, el avance de herramientas, lenguajes tecnológicos han dado un gran avance en estos últimos años, sobre todo cuando hablamos del desarrollo web también se ha mejora la tecnologías básicas de la Web, como JavaScript, HTML y CSS3, asimismo también, gracias a la gran comunidad de desarrolladores, estos han venido implementando FRAMEWORK o como se le conoce en términos generales, marcos de trabajo que permiten un seguimiento estructurado diferente al momento de diseñar e codificar una página web desde estáticas a funcionales, en este caso ASTRO es uno de ese producto de la nueva innovación.

Según el autor Palmowski (2023) menciona en su redacción que ASTRO.JS es un popular generador de sitios estáticos concebidos para crear sitios web ricos en contenido que funcionen con fluidez y rapidez.

Figura 3:

Estructura del Framework Astro



***Fuente**:* https://kinsta.com/es/blog/astro-js/

En resumen, estas tecnologías me ayudaran a realizar la "Implementación de una Arquitectura de Microservicios para Optimizar el Proceso de Gestión de Clientes en la Empresa Funeraria Descanso Eterno S.A.C.". Estas y otras tecnologías y metodologías como la metodología ágil Scrum permitirá garantizar un desarrollo eficiente. La implementación de la arquitectura de microservicios se realizará en la nube, utilizando plataformas "as a service" para aprovechar sus ventajas en escalabilidad y gestión de recursos. La aplicación, desarrollada en Java con Spring Boot, se desplegará en producción para su utilización efectiva. Se incorporarán tecnologías modernas como Tailwind CSS para un diseño atractivo y funcional. Este enfoque integral busca mejorar la eficiencia operativa y la calidad del sistema, asegurando adaptabilidad mediante la combinación de metodologías ágiles y tecnologías avanzadas.

CAPITULO III:

MATERIALES Y MÉTODOS

3.1. ENFOQUE DE LA INVESTIGACIÓN

El estudio adopta un enfoque cuantitativo, ya que se basa en la recolección y análisis de datos numéricos provenientes de la implementación y evaluación de una arquitectura de microservicios en la funeraria Descanso Eterno SAC.

Se busca medir la eficiencia operativa, tiempos de respuesta del sistema, reducción de errores en la gestión de clientes y la satisfacción del usuario, mediante la implementación de herramientas tecnológicas que optimicen los procesos administrativos y mejoren la atención al cliente.

A través de registros obtenidos en la base de datos DynamoDB, logs de monitoreo en AWS CloudWatch y métricas de desempeño del sistema en AWS X-Ray, se aplicarán técnicas de análisis de datos para evaluar la mejora en la gestión de clientes. Este enfoque permitirá cuantificar los beneficios de la arquitectura de microservicios en términos de reducción de tiempos de atención, optimización de recursos tecnológicos y escalabilidad del sistema.

2.2.12. Cuantitativa

El enfoque cuantitativo se fundamenta en la recolección y análisis de métricas obtenidas tras la implementación del sistema en la funeraria. Para ello, se tomarán en cuenta indicadores clave como:

- Tiempo de respuesta del sistema antes y después de la implementación.
- Número de clientes gestionados mensualmente.
- Cantidad de reservas y servicios funerarios administrados a través del sistema.
- Reducción de errores en la gestión de clientes.
- Satisfacción de los empleados y clientes con el nuevo sistema.

A través del uso de herramientas como Python, AWS Lambda, API Gateway, DynamoDB y CloudWatch, se procesarán grandes volúmenes de datos para su interpretación, generando reportes y dashboards que facilitarán la toma de decisiones estratégicas en la gestión de clientes y optimización de procesos internos.

Este análisis cuantitativo permitirá convertir la información bruta en conocimiento útil, favoreciendo la transformación digital de la funeraria y contribuyendo a una gestión más eficiente y moderna del servicio funerario.

3.2. Método de la Investigación

La presente investigación emplea los métodos inductivo y deductivo:

3.2.1. Método Inductivo

A partir del análisis de los datos obtenidos tras la implementación de la arquitectura de microservicios en la funeraria Descanso Eterno SAC, se identificarán patrones de mejora en la gestión de clientes, optimización de procesos y reducción de tiempos de respuesta. Esto permitirá generar conclusiones generales sobre el impacto de la arquitectura de microservicios en la eficiencia operativa del negocio.

3.2.2. Método Deductivo

Se aplicarán principios de ingeniería de software y buenas prácticas en arquitecturas de microservicios para diseñar e implementar el sistema. Además, se utilizarán métricas de rendimiento (como tiempos de procesamiento y niveles de satisfacción del usuario) para validar hipótesis sobre la mejora en la gestión de clientes.

3.3. Diseño de la Investigación

El presente estudio adopta un diseño preexperimental de grupo único con medición pretest-postest, lo que permite evaluar el impacto de la implementación de una arquitectura de microservicios en la gestión de clientes de la Funeraria Descanso Eterno S.A.C.

X ---------- O ----------Y

Donde:

- X: Representa la situación inicial de la funeraria Descanso Eterno, antes de la implementación del sistema basado en microservicios. Se identifican las principales deficiencias en la gestión de clientes y los procesos manuales tanto desde la perspectiva del personal administrativo como de los clientes.

- O: Representa a la información obtenida durante la implementación. Se incluyen mediciones intermedias sobre eficiencia operativa, tiempos de respuesta y percepción de los usuarios en ambos grupos.
- Y: Representa la evaluación posterior a la implementación de la arquitectura de microservicios. Se analizan indicadores clave como:
  - ✓ Para empleados: Reducción en los tiempos de gestión de clientes, mejora en el acceso a la información y eficiencia operativa.
  - ✓ Para clientes: Reducción en el tiempo de espera y aumento en la satisfacción respecto a los servicios brindados.

3.3.1. Justificación del Diseño

Aunque el estudio incluye dos subgrupos (empleados y clientes), ambos pertenecen a un único contexto de estudio, dado que la implementación de la arquitectura de microservicios impacta a toda la gestión de clientes de la funeraria. El análisis comparativo entre pretest y postest permitirá evaluar el efecto del sistema desde ambas perspectivas.

- Diseño Preexperimental: Se emplea este diseño porque existe una intervención directa (la implementación del sistema), pero no se tiene un grupo de control, lo que es típico en estudios tecnológicos y de sistemas de información.
- Medición Pretest-Postest: Se comparan los datos antes y después de la implementación para analizar la mejora en la gestión de clientes.

3.4. Población

La población de estudio está conformada por todos los empleados operativos y administrativos de la Funeraria Descanso Eterno S.A.C., que hicieron uso o gestionaron servicios funerarios durante el año 2024. Según los registros internos, la población total fue de N = 150 personas (115 clientes y 20 empleados administrativos).

3.5. Muestra

Se utilizó un muestreo no probabilístico por conveniencia, seleccionando a los participantes que estuvieron disponibles y dispuestos a responder durante el periodo de recolección de datos. La muestra final estuvo compuesta por n = 35 personas,

distribuidas en 20 clientes y 15 empleados administrativos. El tamaño de muestra fue determinado considerando la viabilidad del acceso a participantes y la necesidad de contar con datos suficientes para la aplicación de pruebas estadísticas paramétricas (Shapiro-Wilk y T-Student).

Cálculo de la muestra de empleados:

$$\text{Tamaño de la muestra} = \frac{NZ^2p(1-p)}{e^2(N-1)+Z^2p(1-p)}$$

Sustituyendo los valores en la fórmula:

N=8

Z= 1.96, con un grado de confianza del 95%

P=0.5

E=0.05

Por lo tanto, encontramos que el tamaño de la muestra de los empleados es de 8 personas.

Cálculo de la muestra de clientes:

$$\text{Tamaño de la muestra} = \frac{NZ^2p(1-p)}{e^2(N-1)+Z^2p(1-p)}$$

Sustituyendo los valores en la fórmula:

N=10

Z= 1.96, con un grado de confianza del 95%

P=0.5

E=0.05

Por lo tanto, el tamaño de la muestra, para la realización de la presente investigación está conformada:

- 8 empleados de la Funeraria Descanso Eterno.
- 10 clientes de la Funeraria Descanso Eterno.

A continuación, se representa la relación de la muestra con los objetivos específicos e indicadores:

OE1: Elaboración de documento de análisis de requerimientos con especificaciones y prioridades identificadas.

OE2: Diseñar Prototipo funcional con al menos 80% de las funcionalidades principales completadas.

OE3: Despliegue exitoso en el entorno de producción y funcionamiento sin incidencias críticas en el primer mes de operación.

OE4: Reducción del tiempo promedio de atención al cliente en al menos un 30%.

OE5: Incremento en la satisfacción del cliente en al menos un 20%, medido mediante encuestas post-implementación.

Tabla 1:

Objetivos e Indicadores

OBJETIVOS	POBLACIÓN/MUESTRA	INDICADORES
OE1	8 empleados.	• Detalle y número de necesidades específicas identificadas.
OE2	Equipo de Desarrollo.	• Número de casos de Uso desarrollados. • Numero de tablas de base de datos. • Numero de microservicios.
OE3	Todos los empleados (8) y clientes (10) de la funeraria.	• Tiempo de respuesta del sistema. • Nivel de eficiencia operativa.
OE4	Empleados / Clientes	• Reducción del tiempo promedio de atención al cliente.
OE5	Clientes	• Nivel de satisfacción.

	• Reducción del tiempo promedio de atención al cliente.

Nota. Definición de objetivos e indicadores.

3.6. TÉCNICAS E INSTRUMENTOS

3.6.1. TÉCNICAS

Para poder recolectar los datos se usará.

- Aplicación de encuestas.
- Observación.

3.6.2. INSTRUMENTOS

3.6.2.1. Observación Directa

- **Fichas de Observación**: Se podrá utilizar para recopilar datos de primera línea sobre como llevan a cabo in situ la gestión de clientes en la funeraria Descanso Eterno S.A.C. del Distrito de Chimbote, es por ello por lo que se desarrollara fichas de observación detalladas que permitirán registrar aspectos relevantes, como la operativa de servicio al cliente, la eficiencia en la gestión de información, etc.

3.6.2.2. Aplicación de Encuestas

- **Cuestionarios**: Se aplicarán cuestionarios con el fin de obtener información relevante como la satisfacción de los empleados y clientes, preferencias y una mejor gestión interna a clientes.

3.6.2.3. Revisión Bibliográfica

- **Fichas Bibliográficas:** Se aplicarán fichas bibliográficas para recopilar información relevante sobre las mejores prácticas en la gestión de clientes y relaciones directas, la adopción de una arquitectura basada en microservicios y el desarrollo de una solución tecnológica adaptada al contexto de la funeraria Descanso Eterno S.A.C.

3.6.2.4.Entrevista

- **Ficha de Entrevistas:** Se elaborará una guía de entrevistas estructuradas, donde se incluirán preguntas especificas relacionadas con la adaptabilidad de los trabajadores y la aceptación de nuevas tecnologías.

3.7. Técnicas de Análisis de Datos

3.7.1. Análisis Descriptivo

- ✓ Cálculo de métricas clave antes y después de la implementación:
  - Tiempo promedio de atención a clientes.
  - Cantidad de reservas procesadas en un período determinado.
  - Número de incidencias o errores en la gestión de clientes.
- ✓ Visualización de datos mediante gráficos estadísticos (barras, líneas y tablas comparativas) para representar los cambios en los indicadores antes y después del sistema.

3.7.2. Análisis Comparativo

- Comparación pretest-postest de las métricas clave para identificar mejoras en la gestión de clientes.
- Cálculo de porcentajes de mejora, por ejemplo:
- % de reducción en los tiempos de atención.
- % de aumento en la satisfacción del cliente (según encuestas).
- Evaluación del impacto en la eficiencia operativa de la funeraria mediante el análisis de tendencias en el tiempo.

3.7.3. Análisis Estadístico

- Cálculo de la media y desviación estándar para evaluar la variabilidad en los tiempos de gestión antes y después del sistema.
- Pruebas estadísticas para determinar si las mejoras son significativas.
- Prueba t de Student para muestras relacionadas: Comparará los tiempos de atención antes y después de la implementación.
- Análisis de correlación entre variables clave, como la relación entre eficiencia operativa y satisfacción del cliente.

3.7.4. Análisis de Percepción del Usuario

- Encuestas a empleados y clientes para evaluar la satisfacción con la nueva arquitectura de microservicios.
- Análisis de tendencias en comentarios y sugerencias, aplicando procesamiento de lenguaje natural (si es viable).

3.8. MÉTODOS DE INVESTIGACIÓN

El presente estudio emplea el método de investigación es Inductivo – Deductivo, ante esta realidad observable, la variable dependiente se puede dividir en características o indicadores en cada de los procesos definidos como: gestión, confiabilidad, calidad, capacidad de mejora continua e innovación tecnológica en la funeraria Descanso Eterno S.A.C. del Distrito de Chimbote, identificando patrones en la gestión de clientes y evaluando el impacto de la implementación de una arquitectura de microservicios en la eficiencia operativa y la calidad del servicio.

A través de este enfoque, se analizarán las características e indicadores de la variable dependiente, que incluyen:

- Gestión de clientes
- Confiabilidad del sistema
- Calidad del servicio
- Capacidad de mejora continua
- Innovación tecnológica

Para lograr este propósito, la investigación seguirá los siguientes pasos metodológicos:

- **Elaboración del marco teórico**: Se revisarán estudios previos sobre microservicios, gestión de clientes y transformación digital en empresas del sector servicios.
- **Definición de la población y muestra**: Se identificarán las unidades de población y se establecerán las muestras de estudio, considerando a los empleados administrativos y clientes de la funeraria.
- **Diseño de técnicas e instrumentos de recolección de datos**: Se prepararán encuestas estructuradas para evaluar la percepción de los clientes y la

eficiencia del sistema por parte de los empleados. Se definirán indicadores cuantificables para medir mejoras en la gestión de clientes.

- **Implementación de la arquitectura de microservicios**: Se desarrollará un sistema basado en microservicios para gestionar clientes y mejorar la eficiencia operativa de la funeraria, Se desplegará la solución en AWS, utilizando Lambda, API Gateway, DynamoDB/MongoDB, y CloudWatch para monitoreo.
- **Aplicación de encuestas y recolección de datos**: Se aplicarán encuestas antes y después de la implementación del sistema para evaluar el impacto en la eficiencia operativa y la satisfacción del cliente.
- **Análisis de resultados y contrastación de hipótesis**: Se procesarán los datos recolectados mediante técnicas estadísticas, Se verificará si la implementación del sistema ha generado mejoras en los indicadores establecidos.
- **Se elaborará el informe final de la investigación:** Se documentarán los hallazgos obtenidos, las mejoras identificadas y las conclusiones basadas en los resultados del estudio.

3.9. Matriz de Consistencia

Tabla 2*:*

Matriz de Consistencia

Problema General	Problemas Específicos	Objetivo General	Objetivos Específicos	Hipótesis	Variable Independiente	Variable Dependiente	Indicadores	Metodología
¿De qué manera la implementación de una arquitectura de microservicios puede mejorar la gestión de clientes en la Funeraria Descanso Eterno S.A.C. en Chimbote?	**1**. ¿Cómo mejorar el análisis de requerimientos para mejorar la gestión de clientes? **2.** ¿Cómo enfrentar la falta de digitalización en la gestión de clientes? **3**. ¿Cómo garantizar que los procesos sean estables en la gestión clientes? **4**. ¿Cómo reducir los	Mejorar la gestión de clientes en la Funeraria Descanso Eterno S.A.C. mediante la implementación de una arquitectura de microservicios que optimice la eficiencia operativa, modernice los servicios y mejore la experiencia del cliente.	**1**. Elaborar un documento de análisis de requerimientos con especificaciones y prioridades identificadas. **2**. Diseñar un prototipo funcional con al menos el 80% de las funcionalidades principales implementadas. **3**. Implementar y desplegar la solución basada en	La implementación de una arquitectura de microservicios mejora la eficiencia operativa y la gestión de clientes en la Funeraria Descanso Eterno S.A.C.	Arquitectura de Microservicios	Gestión de clientes en la Funeraria Descanso Eterno S.A.C.	Variable Dependiente: 1. Tiempo promedio de atención al cliente. 2. Grado de satisfacción de los clientes. 3. Disponibilidad y accesibilidad de la información. Variable Independiente: 1. Implementación efectiva de los microservicios. 2. Reducción de	Tipo de investigación: Aplicada Nivel de investigación: Cuasi Experimental Población: 16 personas (8 empleados y 8 clientes) Muestras: 16 personas Técnicas: Encuestas, medición de tiempos de respuesta, análisis de

	tiempos de atención al cliente? **5**. ¿Cómo evaluar la satisfacción del cliente tras digitalizar el servicio?	microservicios asegurando su correcto funcionamiento en un entorno real. **4**. Optimizar los tiempos de respuesta al cliente, logrando una reducción de al menos el 30%. **5**. Medir el impacto de la solución en la satisfacción del cliente mediante encuestas post-implementación.	procesos manuales. 3. Integración de herramientas de análisis y automatización.	datos en sistema.

Nota. *Cuadro de variables e indicadores*.

3.10. ANÁLISIS DE CONFIABILIDAD DEL INSTRUMENTO

3.10.1. Confiabilidad de Alfa Cronbach Tiempos de Respuesta

Aplicamos la medida estadística del alfa de Cronbach para medir el grado de confiabilidad interna del instrumento de medición.

En la tabla 3 se muestran los datos para el análisis de Alfa de Cronbach en la prueba respecto a los tiempos de respuesta

Análisis 1: Encuestas aplicadas a los empleados de la funeraria.

Tabla 3*:*

Tiempo de respuesta alfa Cronbach

Encuestados	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Suma
E1	1	2	2	1	1	2	1	1	1	1	13
E2	1	1	1	1	2	1	1	1	1	1	11
E3	1	0	0	1	0	1	0	1	1	1	6
E4	1	1	1	1	2	0	1	1	1	0	9
E5	1	0	1	0	0	1	2	1	2	2	10
E6	1	1	3	1	1	2	1	1	1	1	13
E7	0	0	0	1	0	0	1	1	0	0	3
E8	1	1	0	0	0	1	0	1	1	0	4

Nota. *Análisis de confiabilidad*

Tabla 4*:*

Resultados de la Varianza Encontrada

Varianza	0.285	0.535	0.8571	0.4286	0.8571	0.857	0.714	0.28	0.53	0.6
Suma de Varianzas					5.9286					
Varianza de la suma de ítems					10.2143					

Nota. *Cuadro de varianza*

Aplicamos la formula y remplazamos los datos encontrados en la siguiente fórmula para poder encontrar el coeficiente de Cronbach.

$$\alpha = \frac{K}{K-1}\left(\frac{\sum_{i=1}^{K} \sigma_{Y_i}^2}{\sigma_X^2}\right)$$

Tabla 5:

Resultados Alfa de Cronbach Tiempos de Respuesta

Coeficiente de Confiabilidad	**0.749**
Número de ítems	10
Suma total de los puntajes por ítem	4.536
Varianza acumulada del instrumento	13.929

Nota. *Resultados de tiempo de respuesta.*

Obtenemos un coeficiente de confiabilidad de 0.749 de este modo en los rangos de confiabilidad indica que es confiable.

Análisis 2: Encuestas aplicadas a los clientes de la funeraria.

Aplicamos la medida estadística del alfa de Cronbach para medir el grado de confiabilidad interna del instrumento de medición. En la tabla 4 se muestran los datos para evaluación del Alfa de Cronbach referente a la satisfacción de los empleados.

Tabla 6:

Tiempo de respuesta alfa Cronbach

Encuestados	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Suma
C1	1	2	2	1	1	2	1	1	1	1	13
C2	1	1	1	1	2	1	1	1	1	1	11
C3	1	0	0	1	0	1	0	1	1	1	6
C4	1	1	1	1	2	0	1	1	1	0	9
C5	1	0	1	0	0	1	2	1	2	2	10
C6	1	1	3	1	1	2	1	1	1	1	13
C7	0	0	0	1	0	0	1	1	0	0	3
C8	1	1	0	0	0	1	0	1	1	0	4
C9	0	1	0	1	1	2	1	0	1	1	7
C10	1	2	1	1	0	1	1	0	1	1	9

Nota. *Datos recolectados a través de encuesta.*

Tabla 7:

Resultados de la Varianza Encontrada

Varianza	**0.2857**	**0.5357**	**0.8571**	**0.4286**	**0.8571**	**0.857**	**0.714**	**0.28**	**0.53**	**0.6**
Suma de Varianzas					5.9286					
Varianza de la suma de ítems					10.2143					

Nota. *Calculo Individual de las varianzas.*

Aplicamos la formula y remplazamos los datos encontrados en la siguiente fórmula para poder encontrar el coeficiente de Cronbach.

$$\alpha = \frac{K}{K-1}\left(\frac{\sum_{i=1}^{K} \sigma_{Y_i}^2}{\sigma_X^2}\right)$$

Tabla 8:

Resultados Alfa de Cronbach Tiempos de Respuesta

Coeficiente de Confiabilidad	**0.749**
Número de ítems	10
Suma total de los puntajes por ítem	4.536
Varianza acumulada del instrumento	13.929

Nota. *Tiempos de respuesta.*

Obtenemos un coeficiente de confiabilidad de 0.749 de este modo en los rangos de confiabilidad indica que es confiable.

3.11. OPERACIONALIZACIÓN DE LAS VARIABLES

A continuación, se mencionan los siguientes indicadores, esto dentro de las variables mencionadas anteriormente, con el fin de poder lograr con éxito el cumplimiento de la hipótesis, a continuación, se menciona los indicadores de las variables Independiente y Dependiente.

3.11.1. Identificación de variables

Variable Independiente: Arquitectura de Microservicios.

Variable Dependiente: Gestión de clientes.

3.11.2. Indicadores

- Variable Independiente
  - ✓ Número de necesidades identificadas.
  - ✓ Número de casos de uso desarrollados.
  - ✓ Numero de tablas de base de datos diseñadas e implementadas.
  - ✓ Numero de microservicios diseñados e implementados.
  - ✓ Nivel de eficiencia operativa antes y después de la implementación.
- Variable Dependiente
  - ✓ Nivel de satisfacción de los clientes respecto a la calidad de los servicios funerarios (medido por encuestas).
  - ✓ Número de atenciones a los clientes antes y después de la implementación.
  - ✓ Reducción del tiempo promedio de atención al cliente.
  - ✓ Tiempo promedio de respuesta del sistema
  - ✓ Porcentaje del grado de satisfacción de los empleados.

Tabla 9:

Operalización de Variables

VARIABLES	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	INDICADORES	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN
Variable Independiente: Arquitectura de Microservicios	Modelo de desarrollo de software basado en la descomposición de sistemas en módulos independientes (microservicios) que interactúan a través de APIs.	Implementación de microservicios para la gestión de clientes en la funeraria, integrando AWS Lambda, API Gateway y DynamoDB.	- Número de necesidades identificadas. - Número de casos de uso desarrollados. - Número de tablas de base de datos diseñadas e implementadas. - Número de microservicios diseñados e implementados. - Nivel de eficiencia operativa antes y después de la implementación.	- Revisión documental (análisis de requerimientos, casos de uso). - Evaluación técnica (número de microservicios y componentes diseñados). - Análisis de logs en AWS CloudWatch para medir el rendimiento.

Variable Dependiente: Gestión de Clientes	Conjunto de procesos, herramientas y estrategias utilizadas para mejorar la interacción y administración de información de clientes en la funeraria.	Evaluación de la eficiencia en la gestión de clientes antes y después de la implementación de la arquitectura de microservicios.	- Nivel de satisfacción de los clientes respecto a la calidad de los servicios funerarios (medido por encuestas). - Número de atenciones a los clientes antes y después de la implementación. - Optimizar los tiempos de respuesta al cliente - Tiempo promedio de respuesta del sistema. - Porcentaje del grado de satisfacción de los empleados.	- Encuestas de satisfacción para clientes y empleados. - Análisis de tiempos de respuesta del sistema (métricas de AWS y registros en BD). - Comparación de datos históricos (atenciones antes y después de la implementación).

Nota. *Cuadro de variables*

CAPITULO IV

RESULTADOS Y DISCUCIÓN

4.1. Planificación del Desarrollo de la Arquitectura

En este capítulo se detalla la metodología aplicada para el desarrollo de la solución tecnológica propuesta en la presente investigación: la implementación de una arquitectura de microservicios para optimizar la gestión de clientes en la funeraria Descanso Eterno SAC. Dada la naturaleza modular y escalable de los microservicios, y la necesidad de iteraciones rápidas con entregas constantes, se ha optado por emplear la metodología ágil Scrum.

Esta metodología permite trabajar de manera colaborativa con los Stakeholders, ajustarse a cambios en los requerimientos y mantener un control constante sobre el progreso del proyecto. Además, es ideal para equipos pequeños como el utilizado en esta investigación, ya que facilita una comunicación fluida y un enfoque iterativo.

4.1.1 Estructura del Proceso en Scrum

El proceso metodológico basado en Scrum comprende las siguientes fases:

- Inicio del Proyecto: En esta etapa se identifican los objetivos principales, los microservicios necesarios y se establece el Products Backlog, que lista todas las funcionalidades requeridas.
- Planeación de Sprint: El trabajo se organiza en ciclos iterativos cortos, denominados Sprints, que tienen una duración de 1 a 2 semanas. En cada Sprint se seleccionan tareas específicas del Backlog para desarrollarlas y entregarlas.
- Desarrollo Iterativo: Durante los Sprints, el equipo se enfoca en diseñar, implementar y probar los microservicios de manera incremental. Cada iteración incluye reuniones de planificación, revisión y retrospectiva para garantizar el cumplimiento de los objetivos.
- Integración y Pruebas: Una vez completados los Sprints, los microservicios se integran y prueban en conjunto para verificar su correcto funcionamiento como sistema.
- Entrega Final: El proyecto culmina con la entrega del sistema completo, su despliegue en un entorno de producción y la capacitación de los usuarios finales.

A lo largo de este capítulo, se detallarán las actividades realizadas en cada etapa del proceso, incluyendo la planificación inicial, el desarrollo iterativo de los microservicios, las pruebas de integración y el despliegue final del sistema. Además, se describirán las herramientas utilizadas, los entregables generados en cada Sprint y los resultados obtenidos durante el desarrollo.

4.2. Evolución del plan de desarrollo de la arquitectura:

Como se establece en la metodología Scrum, el desarrollo del proyecto se llevará a cabo mediante ciclos iterativos:

- ✓ **Fase de Inicio**: Lo que se busca en la fase de inicio es tener claro los objetivos y plasmarlos para que de esta manera se pueda tener una idea más clara y considerar las limitaciones y alcance de estos.
- ✓ **Fase de planificación y estimación**: Se busca realizar una arquitectura viable para así poder desarrollar un sistema seguro y rápido según la necesidad del cliente. Se tiene que realizar un control de riesgos, para ello se hace una prueba de riesgo. Además de la priorización de las actividades y la estimación de los esfuerzos.
- ✓ **Fase de implementación**: En esta fase lo que se busca es que todo el sistema esté funcionando acorde a los requerimientos del cliente y según la arquitectura definida inicialmente.
- ✓ **Fase revisión y retrospectiva**: Se debe realizar una revisión del trabajo después de cada proceso realizado para así buscar más oportunidades de mejora, haciendo una retroalimentación de como se ha ido construyendo el proyecto en cada uno de sus procesos.
- ✓ **Fase de lanzamiento**: En esta fase el producto final se hace entrega al cliente, además antes de entregar el producto se pueden realizar cambios previa retroalimentación recopilada durante el proceso de desarrollo del proyecto en cuestión.

Durante la ejecución de cada Sprint, las tareas asociadas a las historias de usuario serán registradas y gestionadas para asegurar su cumplimiento. En caso de que alguna funcionalidad no sea completada dentro del Sprint debido a limitaciones técnicas o cambios en las prioridades, estas se evaluarán nuevamente durante la planificación del siguiente ciclo, ajustando los compromisos según sea necesario.

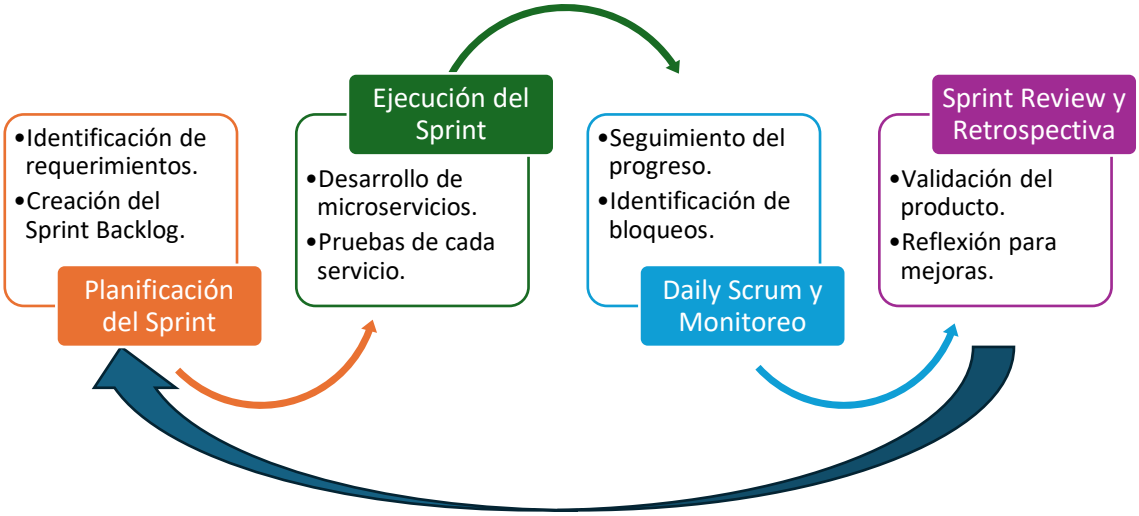
Para el registro y seguimiento de las historias de usuario, se utilizará un formato estructurado en una hoja de cálculo, que facilitará la organización y priorización de las tareas pendientes.

En las revisiones al final de cada Sprint (Sprint Review), se invitará a los stakeholders clave, incluyendo empleados y representantes de la funeraria Descanso Eterno SAC, para presentar demostraciones de los avances logrados. Estas demostraciones tendrán como objetivo recopilar retroalimentación directa de los usuarios finales, especialmente de aquellos trabajadores que actualmente realizan las actividades de manera manual. Este feedback será fundamental para garantizar que el sistema propuesto sea funcional, intuitivo y alineado con las necesidades operativas de la empresa.

Los dailys se ejecutan en un lapso de 10 minutos. Se estableció como horario para la reunión, las 6 de la noche.

Figura 4:

Fases de Scrum



Nota. *Elaboración del flujo de ejecución para cada Sprint*

4.3.Fase de Inicio

4.3.1. Definición de roles

Tabla 10*:*

Definición de Roles

Rol	Asignado	Responsabilidades
Product Owner	**Gerente General de la Funeraria**	- Definir y priorizar los requerimientos en el Product Backlog. - Asegurar que los entregables cumplan los objetivos del negocio. - Validar las funcionalidades implementadas durante las revisiones de Sprint (Sprint Review).
Scrum Master	**Bach. Mendoza Oviedo Michael Nay**	- Facilitar las ceremonias de Scrum (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective). - Eliminar impedimentos durante el desarrollo del proyecto. - Asegurar que se sigan las prácticas de Scrum.
Equipo de Desarrollo	**Bach. Mendoza Oviedo Michael Nay**	- Diseñar, desarrollar e implementar los microservicios según las prioridades del backlog. - Realizar pruebas unitarias e integrales de las funcionalidades. - Documentar el progreso del desarrollo y solucionar incidencias técnicas.

Nota. *Cuadro de asignación de roles.*

4.3.2. Detalles técnicos

- El Product Owner participa activamente en la planificación del sprint para priorizar los requerimientos.
- El Scrum Máster lidera las reuniones diarias para identificar bloqueos y coordinar al equipo.

4.3.3. Requerimientos Mínimos del Producto.

Estándares Aplicables

- ✓ Estándar de plataforma – Windows 10
- ✓ Estándar de comunicación – TCP/IP

Requerimientos de software

En la tabla 11 se presentas los requerimientos de software que se necesitaran para el uso del sistema informático de forma detallada.

Tabla 11*:*

Requerimientos de Software

SOFTWARE	REQUERIMINETOS
Sistema operativo	Windows 10
Navegador Web	Brave, Microsoft Edge, Opera, chrome
Lenguaje programación	Python, Django, JavaScript
Base de datos	NoSql – MongoDB-DYNAMODB

Nota. *Cuadro de requerimientos del software.*

Requerimiento de Hardware

En la tabla 12 se presentan los requerimientos técnicos mínimos de hardware necesarios del equipo en el que se ejecutara el sistema informático.

Tabla 12*:*

Requerimientos del Hardware

HARDWARE	DESCRIPCIÓN
Memoria RAM	4GB recomendado
Microprocesador	Intel Core i3 de 2.6 GHz recomendado
Disco duro	250 GB mínimo

Nota. *Cuadro de requerimientos del Hardware.*

4.3.4. Requerimientos de Documentación.

Se elaborará un manual de uso para el sistema de gestión de clientes de la Funeraria Descanso Eterno S.A.C., en el cual se describirán detalladamente las funcionalidades del sistema basado en arquitectura de microservicios. Este manual estará diseñado para empleados administrativos y el gerente general de la funeraria, asegurando una comprensión clara de las herramientas implementadas.

Adicionalmente, se incluirán capturas de las interfaces del sistema para facilitar la comprensión de las funcionalidades y procesos, garantizando una correcta adopción del sistema desde un enfoque de usuario.

El manual incluirá lo siguiente:

- ✓ Índice.
- ✓ Principales funcionalidades del sistema. Explicación detallada de los módulos y microservicios implementados.
- ✓ Guía detallada de las interfaces del sistema. Instrucciones paso a paso sobre cómo navegar y utilizar las funcionalidades del sistema.
- ✓ Guía de resolución de problemas comunes. Respuestas a preguntas frecuentes y soluciones a posibles errores o dificultades de uso.
- ✓ Procedimiento de validación de roles y permisos. Explicación de los niveles de acceso y roles dentro del sistema para empleados y administradores.
- ✓ Proceso de generación de reportes. Explicación sobre cómo los usuarios pueden visualizar y extraer datos clave a través del sistema.
- ✓ Información de contacto del equipo de desarrollo. Detalles para solicitar soporte técnico o reportar incidencias.
- ✓ Glosario de términos técnicos. Definición de conceptos clave utilizados en la documentación del sistema.

Este documento servirá como referencia para la capacitación del personal y garantizará el uso eficiente del sistema en la funeraria.

4.3.5. Modelos de Casos de Uso del Negocio.

El objetivo del sistema de gestión de clientes basado en arquitectura de microservicios para la Funeraria Descanso Eterno S.A.C. es optimizar y modernizar la gestión de clientes, aumentando la eficiencia operativa y mejorando la experiencia del usuario.

Para lograr esto, el sistema se basa en un modelo de gestión de información que permite a los diferentes actores involucrados (empleados administrativos y gerente general) canalizar y administrar de manera centralizada los datos de los clientes y los servicios funerarios.

La información fluye de manera estructurada a través de los distintos microservicios, asegurando automatización en los procesos, reducción de errores y acceso en tiempo real a la información relevante. Además, el sistema permitirá generar reportes analíticos para evaluar el rendimiento de los servicios y facilitar la toma de decisiones estratégicas.

Este enfoque garantiza una gestión más eficiente de la funeraria, mejora en la atención al cliente y una mayor capacidad de respuesta ante solicitudes y consultas, lo que contribuye al crecimiento sostenible de la empresa.

4.3.6. Actores del Negocio.

- ✓ Empleados Operativos
- ✓ Empleados administrativos
- ✓ Jefe de la Funeraria
- ✓ Clientes

4.3.7. Casos de Uso del Negocio

En la siguiente tabla 13, se detallan las principales situaciones operativas clave que guían el funcionamiento del negocio.

Tabla 13:

Casos de Uso del Negocio

Control de accesos	**Función**	**Uso**	**Base de Datos**
	Gestiona tokens JWT para el inicio de sesión de los empleados.	Asegura que solo usuarios autorizados puedan acceder a los microservicios.	Puede validar usuarios contra un registro almacenado en DynamoDB.
	Verifica roles y permisos (por ejemplo, administrador o empleado).		
Registro de Clientes (Gestión de Clientes)	Permite registrar nuevos clientes.	Los empleados pueden registrar información básica de los clientes (nombre, contacto, historial de servicios) desde un portal o sistema interno.	Almacena los datos de los clientes en DynamoDB.
	Consulta y edita información existente de clientes.		
	Realiza validaciones para evitar registros duplicados.		
Gestión de Servicios Funerarios	Registra solicitudes de servicios funerarios, como velorios, traslados o servicios personalizados.	Automatiza el proceso de registro y permite hacer un seguimiento detallado de cada servicio.	Almacena los detalles del servicio (tipo, fecha, lugar, cliente) en DynamoDB.
	Asocia cada solicitud al cliente correspondiente.		
Gestión de Reservas	Permite a los empleados realizar y consultar reservas de recursos (salas, vehículos, personal).	Facilita la organización interna y mejora la coordinación de recursos.	Almacena horarios y disponibilidad en DynamoDB.
	Garantiza la disponibilidad de los recursos evitando conflictos.		
Notificaciones (Correo/SMS)	Enviar notificaciones automáticas a los	Mejora la comunicación	SERVICIO AWS:

	clientes sobre confirmación de servicios.	tanto con los clientes como entre el equipo interno.	Utiliza AWS SNS para el envío de notificaciones.
	Notificar a los empleados sobre nuevas solicitudes o cambios en las reservas.		
Generación de Reportes	Generar reportes operativos, como el número de clientes atendidos, servicios realizados, o tiempos promedio de atención.	Ayuda a los gerentes a analizar la eficiencia del sistema y tomar decisiones basadas en datos.	Consulta datos operativos de DynamoDB.
	Exportar reportes en formatos como PDF o CSV.		
Gestión de Inventario:	Monitorea y gestiona el inventario de materiales necesarios para los servicios funerarios (flores, urnas, vehículos).	Asegura la disponibilidad de recursos necesarios para los servicios.	Almacena el inventario en DynamoDB con cantidades y fechas de actualización.
	Actualiza automáticamente el inventario tras registrar un servicio.		

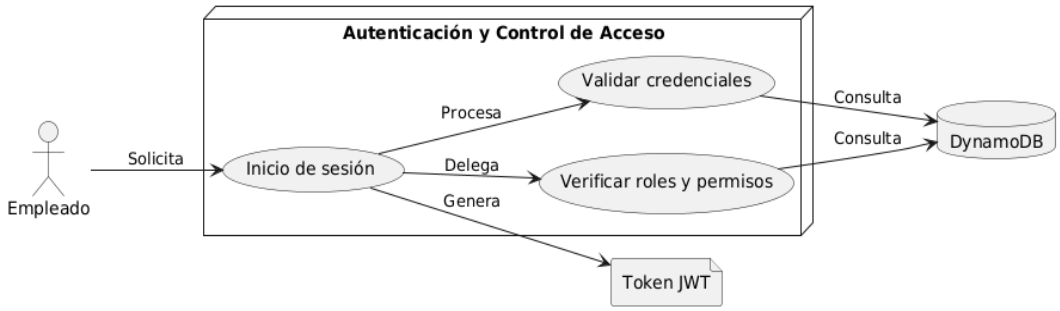
Nota. *Elaboración Propia*

- ✓ **Diagrama De Despliegue: Microservicio "Autenticación Y Control De Acceso"**

  Este diagrama muestra cómo funciona el servicio de autenticación cuando un usuario intenta acceder al sistema. Primero, el cliente envía sus credenciales para iniciar sesión. El microservicio las valida y, si son correctas, genera un token JWT para autorizar al usuario. Luego, consulta en DynamoDB los roles y permisos asociados a ese usuario, asegurando que solo acceda a lo permitido.

Figura 5:

Despliegue ***Autenticación*** *y Control de Acceso*



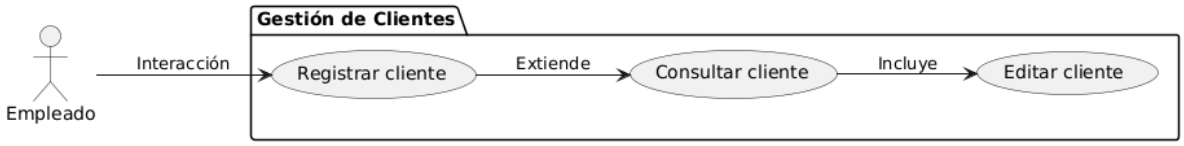
Nota. *Flujo Operacional Auth*

- ✓ **Diagrama de caso de uso: microservicio gestión de clientes**

  Este diagrama representa las funcionalidades principales del microservicio de gestión de clientes, destacando tres operaciones clave: registrar, consultar y editar clientes. Las relaciones "Extiende" e "Incluye" muestran cómo estas acciones se interconectan dentro del flujo del sistema. El diseño refleja los requisitos funcionales documentados para este módulo.

Figura 6:

Diagrama CU Gestión de Clientes



Nota. *Funcionalidades del módulo de clientes*

- ✓ **Diagrama de Despliegue: Microservicio gestión de clientes**

  Este diagrama ilustra la arquitectura de despliegue del microservicio, mostrando cómo interactúa con DynamoDB para realizar operaciones clave como registrar, consultar, editar y actualizar datos de clientes.

Figura 7:

Diagrama de Despliegue Gestión de clientes

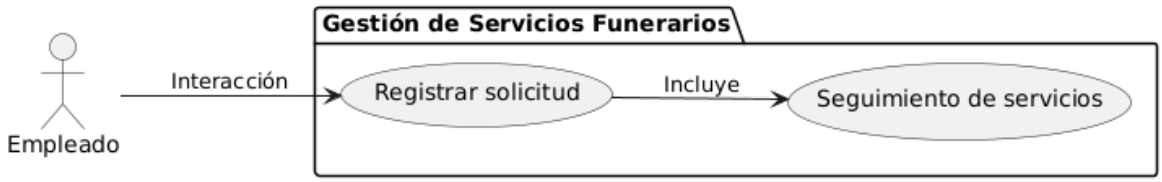
Nota. *Especificaciones técnicas del equipo Backend*

- ✓ **Diagrama de caso de uso: Microservicio Gestión Servicios Funerarios**

  El diagrama de caso de uso muestra la interacción entre un empleado y el microservicio de Gestión de Servicios Funerarios. El flujo incluye el registro de una solicitud, la asociación con servicios específicos, el seguimiento correspondiente y el almacenamiento de la información en DynamoDB.

Figura 8:

Diagrama de Caso de Uso Gestión Servicios Funerarios



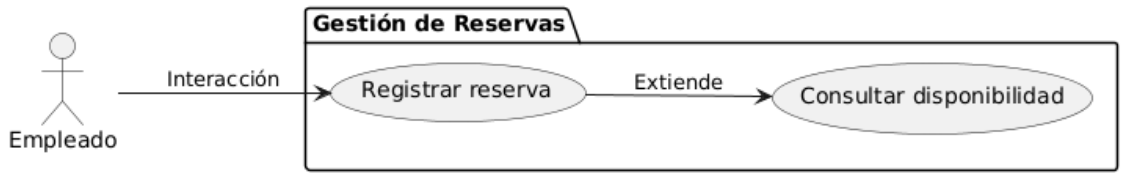
Nota. *Especificaciones de Servicios Funerarios*

- ✓ **Diagrama de caso de uso: Microservicio Gestión de Reservas**

  Este diagrama representa el caso de uso del microservicio de Gestión de Reservas, donde un empleado registra una reserva y, como extensión del proceso, consulta la disponibilidad del servicio.

Figura 9:

Diagrama CU Gestión Reservas



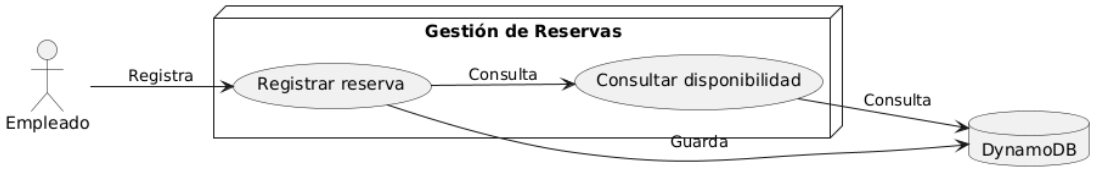
Nota. *Flujo Gestión de Reservas*

- ✓ **Diagrama de Despliegue: Microservicio Gestión de Reservas**

El diagrama muestra el flujo de interacción en el microservicio de Gestión de Reservas. El empleado registra una reserva, consulta la disponibilidad y los datos se almacenan o consultan desde la base de datos DynamoDB como parte del proceso.

Figura 10:

Diagrama Despliegue gestión Reservas



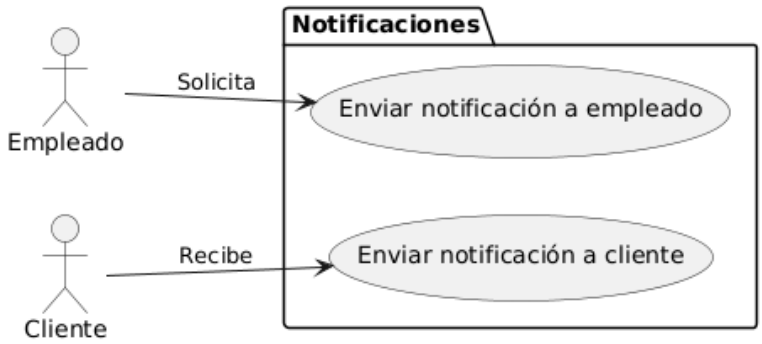
Nota. *Flujo Operativo de Reservas*

- ✓ **Diagrama de Caso de Uso: Microservicio Gestión de Notificaciones**

El diagrama de caso de uso representa el funcionamiento del microservicio de Gestión de Notificaciones. Permite al empleado solicitar el envío de notificaciones tanto a sí mismo como al cliente, quien actúa como receptor del mensaje generado por el sistema.

Figura 11:

Diagrama CU Gestión Notificaciones



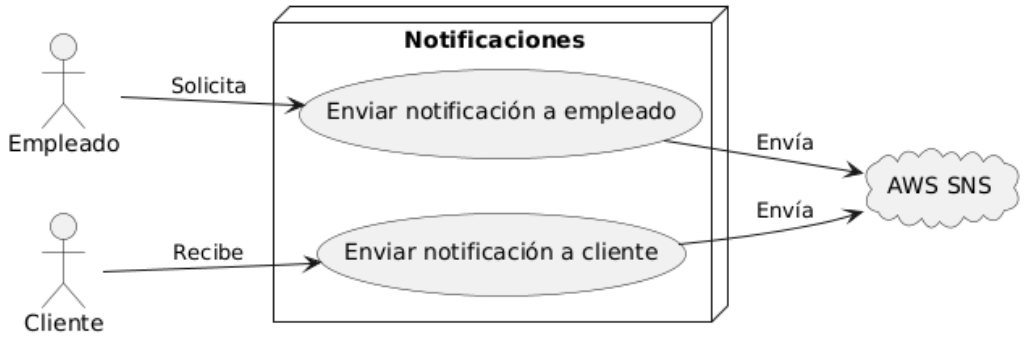
Nota. *Flujo Operativo de Notificaciones*

- ✓ **Diagrama de Despliegue: Microservicio Gestión de Notificaciones**

  Este diagrama de despliegue ilustra el funcionamiento del microservicio de Gestión de Notificaciones. Un empleado solicita el envío de una notificación, ya sea hacia sí mismo o hacia un cliente. El sistema procesa la solicitud y utiliza AWS SNS como servicio externo para la entrega del mensaje.

Figura 12:

Diagrama Despliegue Gestión Notificaciones



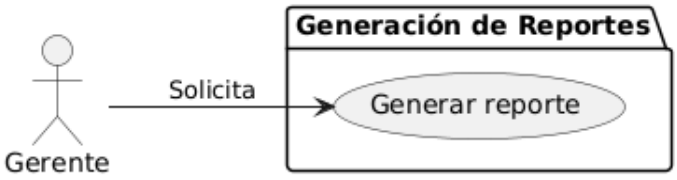
Nota. *Flujo Despliegue Notificaciones*

- ✓ **Diagrama de Caso de Uso: Microservicio Generación de Reportes**

  El diagrama de caso de uso representa la interacción del gerente con el microservicio de Generación de Reportes, mediante el cual solicita la creación de un informe específico como parte de su labor de monitoreo o análisis.

Figura 13:

Diagrama CU Generación Reporte



Nota. *Flujo Proceso Generar Reporte*

- ✓ **Diagrama de Despliegue: Microservicio Generación de Reportes**

  Este diagrama muestra el despliegue del microservicio de Generación de Reportes. El gerente solicita la generación del informe, el sistema consulta los datos almacenados en DynamoDB y posteriormente exporta el resultado en formatos PDF o CSV, según lo requerido.

Figura 14:

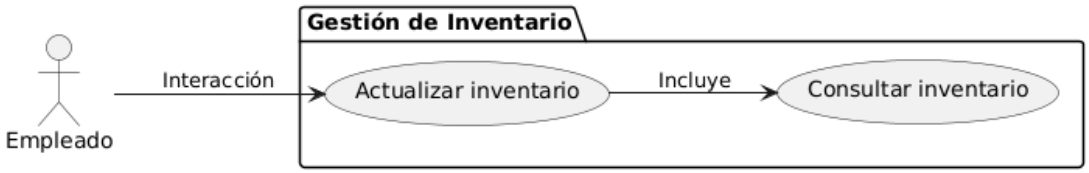
Diagrama Despliegue Generación Reporte

Nota. *Flujo Exportar Reporte*

✓ **Diagrama de Caso de Uso: Microservicio Gestión de Inventario**

Figura 15*:*

Diagrama CU Gestión Inventario

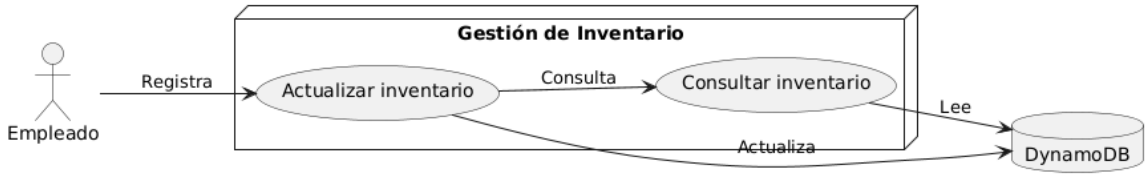


Nota. *Flujo de negocio Inventario*

✓ **Diagrama de Despliegue: Microservicio Gestión de Inventario**

Figura 16*:*

Diagrama Despliegue Gestión Inventario



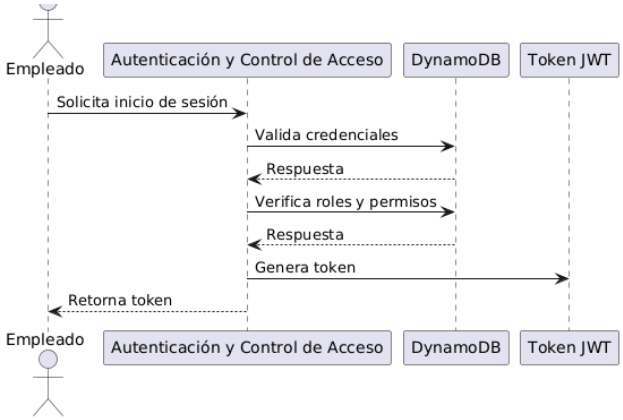
Nota. *Flujo Funcional de Inventario*

4.3.8. Diagrama de Secuencia

Microservicio: Autenticación y Control de Acceso

Figura 17*:*

Diagrama Secuencia AUTH

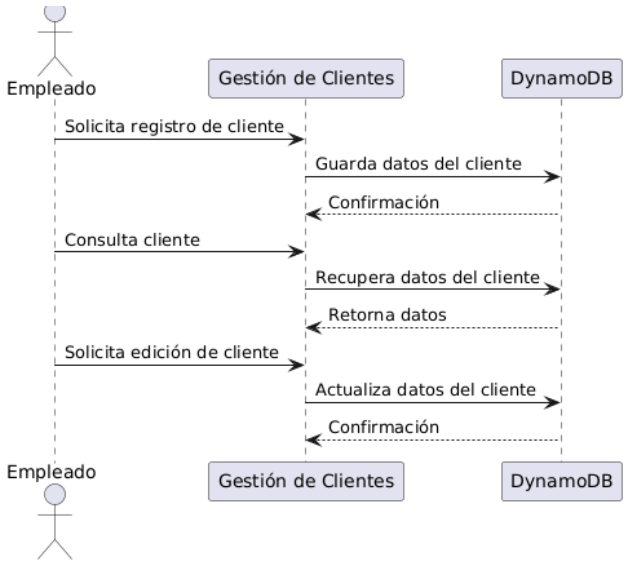


Nota. *Secuencia de Seguridad*

Microservicio: Gestión de Clientes

Figura 18:

Diagrama Secuencia Gestión Clientes

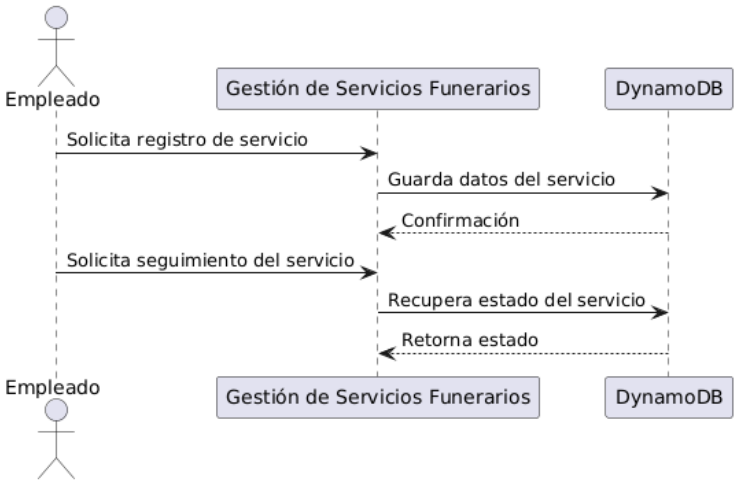


Nota. *Secuencia Clientes*

Microservicio: Gestión de Servicios Funerarios

Figura 19:

Diagrama Secuencia Servicios Funerarios

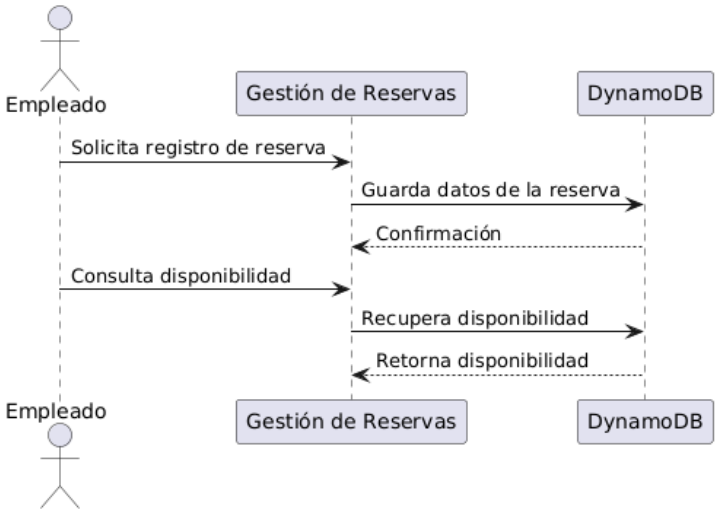


Nota. *Secuencia Servicios Funerarios*

Microservicio: Gestión de Reservas

***Figura 20**:*

Diagrama Secuencia Gestión Reservas

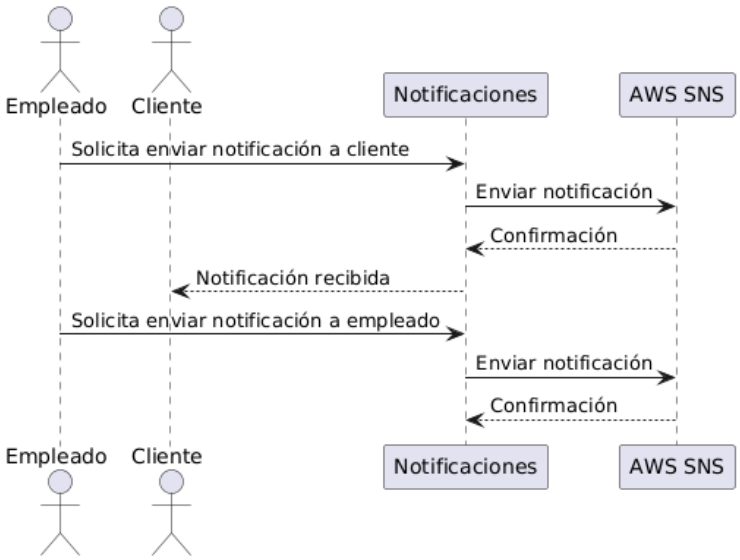


Nota. *Secuencia Reservas.*

Microservicio: Notificaciones

Figura 21:

Diagrama Secuencia Notificaciones

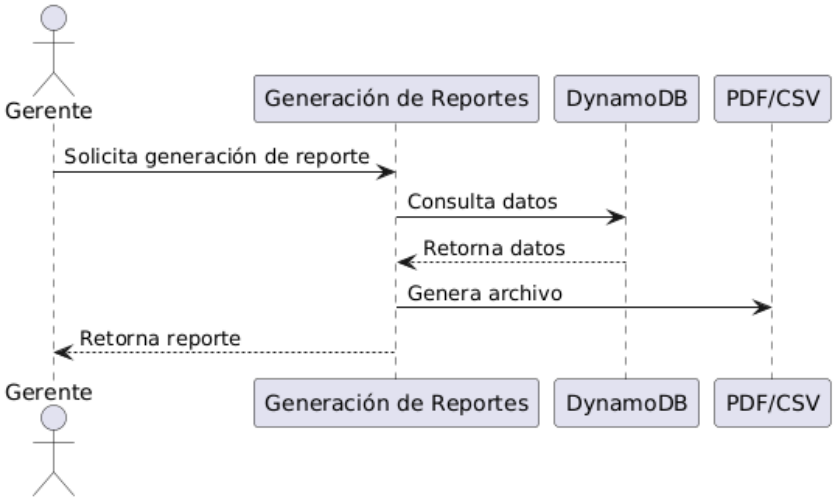


Nota. *Secuencia Notificaciones*

Microservicio: Generación de Reportes

Figura 22*:*

Diagrama Secuencia Reportes

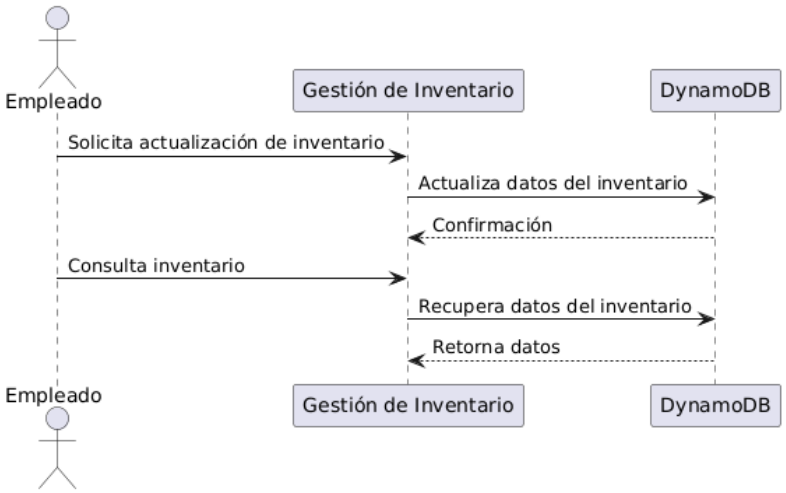


Nota. *Secuencia Reportes*

Microservicio: Gestión de Inventario

Figura 23*:*

Diagrama Secuencia Inventario

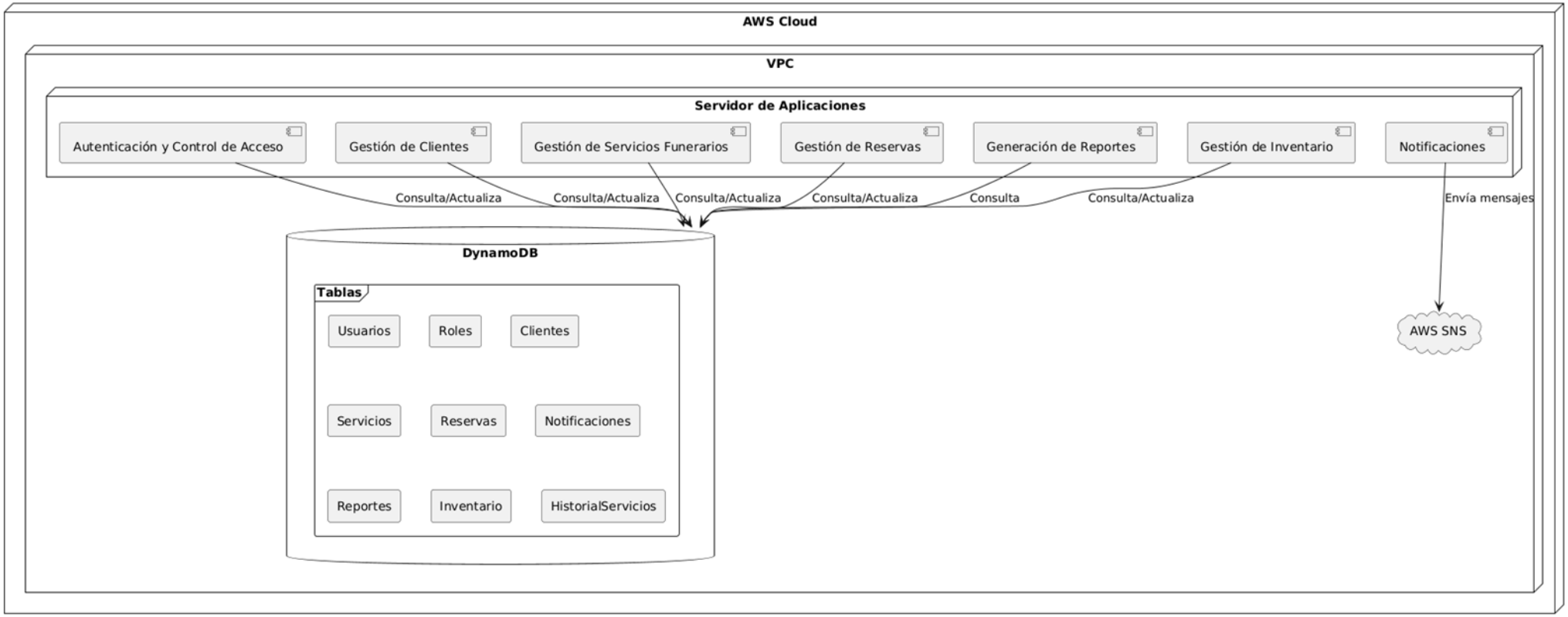


Nota. *Secuencia Inventario*

4.3.9. Diagrama de Despliegue General

Figura 24:

Diagrama De Despliegue General



Nota. *Flujo general del Despliegue*

4.3.10. Requerimientos no Funcionales

- ✓ **Disponibilidad**: El sistema de gestión de clientes basado en microservicios deberá estar disponible las 24 horas del día, garantizando un acceso continuo para los empleados y el gerente de la funeraria. Esto permitirá una gestión ininterrumpida de los servicios y solicitudes de los clientes, asegurando la disponibilidad de la información en cualquier momento.
- ✓ **Seguridad**: El acceso al sistema deberá estar protegido por credenciales de usuario, asignadas según los roles definidos dentro de la funeraria (empleado y administrador). Además, cada usuario tendrá permisos específicos, limitando el acceso a información sensible y garantizando la integridad de los datos de los clientes.
- ✓ **Operatividad**: El sistema debe ser intuitivo y fácil de usar, permitiendo que los empleados de la funeraria puedan aprender a utilizarlo en un máximo de 4 horas. Esto reducirá la tasa de errores en su uso y optimizará la gestión de los clientes sin necesidad de una extensa capacitación.
- ✓ **Validación de Datos**: El sistema deberá ser capaz de validar la información ingresada y visualizada, evitando errores en la gestión de clientes, servicios funerarios y reportes generados. Además, deberá contar con mensajes de advertencia y validaciones automáticas para asegurar la integridad de los datos.
- ✓ **Desempeño**: El sistema deberá ofrecer un rendimiento óptimo, asegurando tiempos de respuesta mínimos en las operaciones críticas, como la gestión de clientes, el procesamiento de solicitudes y la generación de reportes. Se espera que las consultas a la base de datos sean procesadas en menos de 2 segundos en condiciones normales de operación.
- ✓ **Mantenibilidad**: El sistema deberá contar con una documentación detallada, incluyendo un manual de usuario que describa los procesos del sistema, guías de uso y resolución de problemas. Además, la arquitectura de microservicios deberá permitir actualizaciones y mantenimiento sin afectar la operatividad general del sistema.

4.4. Fase Planificación y Estimación

En esta fase se recopiló la información y los datos importantes que nos brindó el Product Owner, que en este caso es el administrador de la empresa SGP Business,

todo esto se hizo con el fin de planificar la forma en la que se va a trabajar, así mismo se hizo el sprint planning donde participaron el equipo Scrum.

4.4.1. Definición de historias de usuario.

De acuerdo con la información proporcionada por el Product Owner (Gerente General de la Funeraria Descanso Eterno S.A.C.), se redactaron las diferentes historias de usuario para definir las características y funcionalidades que debe poseer el sistema de gestión de clientes basado en microservicios.

La redacción de estas historias siguió un enfoque incremental, comenzando por las funcionalidades más básicas, como el registro de clientes y la autenticación de usuarios, hasta la vinculación de microservicios y la integración de interfaces del sistema.

Adicionalmente, cada historia de usuario recibió un puntaje de esfuerzo en una escala del 1 al 10, permitiendo al equipo Scrum estimar la complejidad y el tiempo requerido para su implementación en los distintos sprints.

Tabla 14:

Historia N°1. Inicio de Sesión

Historia	Inicio de Sesión
Como	Usuario del sistema (Empleado o Administrador)
Quiero	Iniciar sesión en la plataforma de gestión funeraria
Para	Acceder a las funcionalidades según mi rol y gestionar clientes y servicios funerarios

Nota. *Inicio de Sesión*

Tabla 15:

Historia N°2. Registro de Clientes

Historia	Registro de Clientes

Como	Empleado de la funeraria
Quiero	Registrar a los clientes con sus datos personales y solicitudes de servicio
Para	Mantener un historial organizado y mejorar la atención al cliente

Nota. *Registro Clientes*

Tabla 16*:*

Historia N°3. Gestión de Servicios Funerarios

Historia	Gestión de Servicios Funerarios
Como	Administrador
Quiero	Administrar los tipos de servicios funerarios contratados por los clientes
Para	Asignar recursos y coordinar el cumplimiento de los servicios de manera eficiente

Nota. *Servicios Funerarios*

Tabla 17*:*

Historia N°4. Gestión de Reservas

Historia	**Gestión de Reservas**
Como	Cliente
Quiero	Reservar un servicio funerario de forma digital
Para	Garantizar disponibilidad y reducir tiempos de espera

Nota. *Cuadro de Historia Gestión de Reservas*

Tabla 18:

Historia N°5. Notificaciones Automáticas

Historia	**Notificaciones Automáticas**
Como	Cliente
Quiero	Recibir notificaciones por correo/SMS sobre el estado de mi solicitud
Para	Mantenerme informado sobre la confirmación y estado del servicio contratado

Nota. *Cuadro Notificaciones*

Tabla 19:

Historia N°6. Generación de Reportes

Historia	**Generación de Reportes**
Como	Administrador
Quiero	Generar reportes sobre el uso de los servicios y satisfacción del cliente
Para	Evaluar el rendimiento del sistema y mejorar los procesos

Nota. *Cuadro Reportes*

Tabla 20:

Historia N°7. Historial de Clientes

Historia	**Historial de Clientes**
Como	Empleado de la funeraria
Quiero	Consultar el historial de servicios contratados por un cliente
Para	Brindar atención personalizada y mejorar la gestión de clientes

Nota. *Cuadro Clientes*.

Tabla 21*:*

Historia N°8. Gestión de Inventario

Historia	**Gestión de Inventario**
Como	Administrador
Quiero	Gestionar el inventario de insumos y recursos funerarios
Para	Asegurar la disponibilidad de materiales esenciales en cada servicio

Nota. *Cuadro Inventario.*

4.4.2. Product Backlog

El Product Backlog está compuesto por los requerimientos o historias de usuario, que deben ser priorizados y ejecutados en sprints. Aquí tienes una tabla ejemplo:

- ✓ Listado de las funcionalidades principales que se desarrollarán.
- ✓ Prioridad y descripción de los entregables en cada sprint.

Tabla 22*:*

Product Backlog

Historia de Usuario	**Descripción**	**Prioridad**	**Puntos**	**Módulo**
H1: Inicio de sesión	Crear el diagrama de arquitectura basado en AWS Lambda, API Gateway y DynamoDB. Crear el esquema de colecciones y relaciones en DynamoDB. Crear el microservicio de Login con validación	Alta	8	Autenticación

	JWT e integración con IAM.			
H2: Registro de clientes	Como administrador, quiero registrar nuevos clientes para gestionar sus datos.	Alta	7	Gestión de clientes
H3: Gestión de servicios funerarios		Alta	8	Gestión de servicios
H4: Gestión de reservas		Media	5	Reservas
H5: Notificaciones automáticas (Correo/SMS)		Alta	7	Notificaciones
H6: Generación de reportes		Media	5	Reportes
H7: Historial de clientes		Baja	4	Gestión de clientes
H8: Gestión de inventario		Media	6	Inventario

Nota. *Cuadro Backlog*

4.4.3. Desarrollo de Sprints

Para la creación de los sprints se tomó en cuenta los valores de prioridad y se asignó una duración máxima de 2 semanas para el desarrollo de cada sprint creado. Se desarrollo de acuerdo con el módulo principal y el historial de usuario que este contiene.

Divide esta sección en los sprints ejecutados, incluyendo los entregables específicos de cada uno:

- ✓ Sprint 1: Diseño de arquitectura, configuración inicial en AWS.
- ✓ Sprint 2: Desarrollo de microservicios básicos (Registro de Clientes, Inicio de Sesión).
- ✓ Sprint 3: Integración con DynamoDB y Amazon SNS.
- ✓ Sprint 4: Implementación de monitoreo (CloudWatch y X-Ray), pruebas y ajustes.
- ✓ Detalla el trabajo realizado en cada sprint:
- ✓ Planificación del Sprint (Sprint Planning).
- ✓ Implementación de tareas (Desarrollo).
- ✓ Validación de entregables (Sprint Review).
- ✓ Mejoras sugeridas (Sprint Retrospective).

En la siguiente tabla se muestran la identificación de los sprints y agrupando en cada uno de ellos las historias asignadas a cada sprint identificado, además de la asignación de los tiempos de realización de los sprints.

Tabla 23*:*

Desarrollo de Sprints

N°	**Historias de usuario**	**Objetivo**	**Fecha inicio**	**Fecha fin**
Sprint 1	H1	Creación de la estructura fundamental de la aplicación (API Gateway, autenticación y base de datos)	14/08/2023	28/08/2023
Sprint 2	H2, H3	Implementación del módulo de gestión de clientes y servicios funerarios	28/08/2023	11/09/2023
Sprint 3	H4, H5	Desarrollo del módulo de reservas y notificaciones	11/09/2023	25/09/2023

		automáticas (Correo/SMS)		
Sprint 4	H6	Implementación del módulo de reportes y Dashboard de gestión	25/09/2023	09/10/2023
Sprint 5	H7, H8	Creación del módulo de historial de clientes e inventario de productos funerarios	09/10/2023	23/10/2023
Sprint 6	H1, H2, H3, H4, H5, H6, H7, H8.	Pruebas de integración y despliegue en AWS (incluyendo monitoreo con CloudWatch y X-Ray)	23/10/2023	06/11/2023
Sprint 7	H1, H2, H3, H4, H5, H6, H7, H8.	Validación final del sistema con usuarios y ajustes antes de la implementación en producción	06/11/2023	04/12/2023

Nota. *Cuadro Sprints.*

4.5. Fase de Implementación

En la fase de implementación se realizó cada uno de los sprints tomando en cuenta los tiempos asignados, así como el desarrollo de cada una de las actividades que contiene cada sprint y se presentó cada uno de los entregables en las reuniones con el Producto Owner.

4.5.1. Sprint N°1: Creación de la Estructura Fundamental de la Aplicación.

4.5.1.1. Planificación del Sprint.

El equipo Scrum definió que este sprint se enfocará en la implementación de la arquitectura base de la aplicación.

Se establecieron las tareas clave, incluyendo la configuración de la autenticación, la integración con API Gateway y la base de datos NoSQL en AWS DynamoDB.

Sprint 1: Completar la estructura fundamental de la aplicación para tener un mejor control de las actividades de las sedes.

4.5.1.2. Sprint Backlog.

Se presentaron las tareas específicas que se desarrollaran en el sprint, estas tareas fueron presentadas en el Product Backlog.

- Diseño de la arquitectura de microservicios.
- Implementación del API Gateway y configuración de endpoints.
- Configuración de la autenticación con AWS IAM.
- Modelado y diseño de la base de datos NoSQL en DynamoDB.
- Creación de las primeras AWS Lambda Functions para gestionar clientes y servicios funerarios.

4.5.1.3. Ejecución del Sprint.

Se realizo la primera reunión diaria para dar el inicio formal de las del trabajo del
primer Sprint, para sincronizar el al equipo y discutir las tareas diarias. Durante la ejecución del sprint, se avanzó en la creación de los primeros microservicios, asegurando la comunicación entre los componentes principales. Se definieron los siguientes elementos:

- **Diagrama de Arquitectura General**: Incluir el diagrama de microservicios con las conexiones entre API Gateway, AWS Lambda y DynamoDB.
- **Base de Datos NoSQL**:
  - ✓ Presentar el diagrama de la base de datos NoSQL, explicando cómo están organizadas las colecciones/tablas en DynamoDB.
  - ✓ Explicar los principales atributos y relaciones en la base de datos.

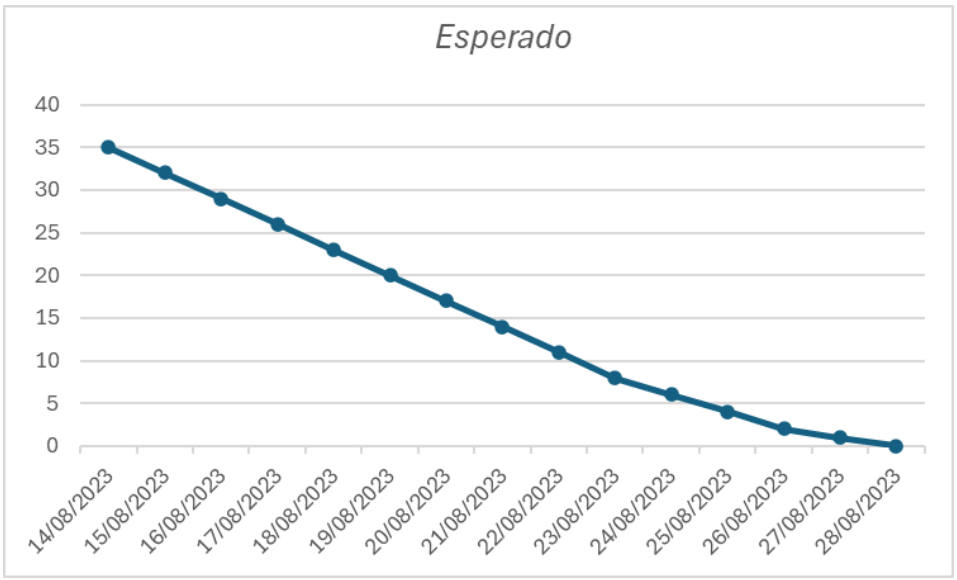
- **Código Implementado:**
  - ✓ Se pueden incluir fragmentos clave del código de las AWS Lambda Functions desarrolladas.
  - ✓ Un ejemplo de la configuración de API Gateway con las rutas y métodos HTTP.
- **Burn Down Chart**: Se incluye el gráfico Burn Down Chart con el tiempo estimado de desarrollo frente al tiempo real ejecutado en este sprint.

Gráfico Burn Down Estimado

Se En la Figura XX se muestra el Burn Down Chart del Sprint 1. La línea azul representa la proyección esperada del Sprint, mientras que la línea naranja refleja el progreso real del equipo. Se observa que al inicio hubo una ligera demora en el desarrollo de ciertas funcionalidades, pero al final del Sprint se logró completar todas las tareas planificadas

Figura 25*:*

Gráfico Burn Down de tiempo para el primer Sprint



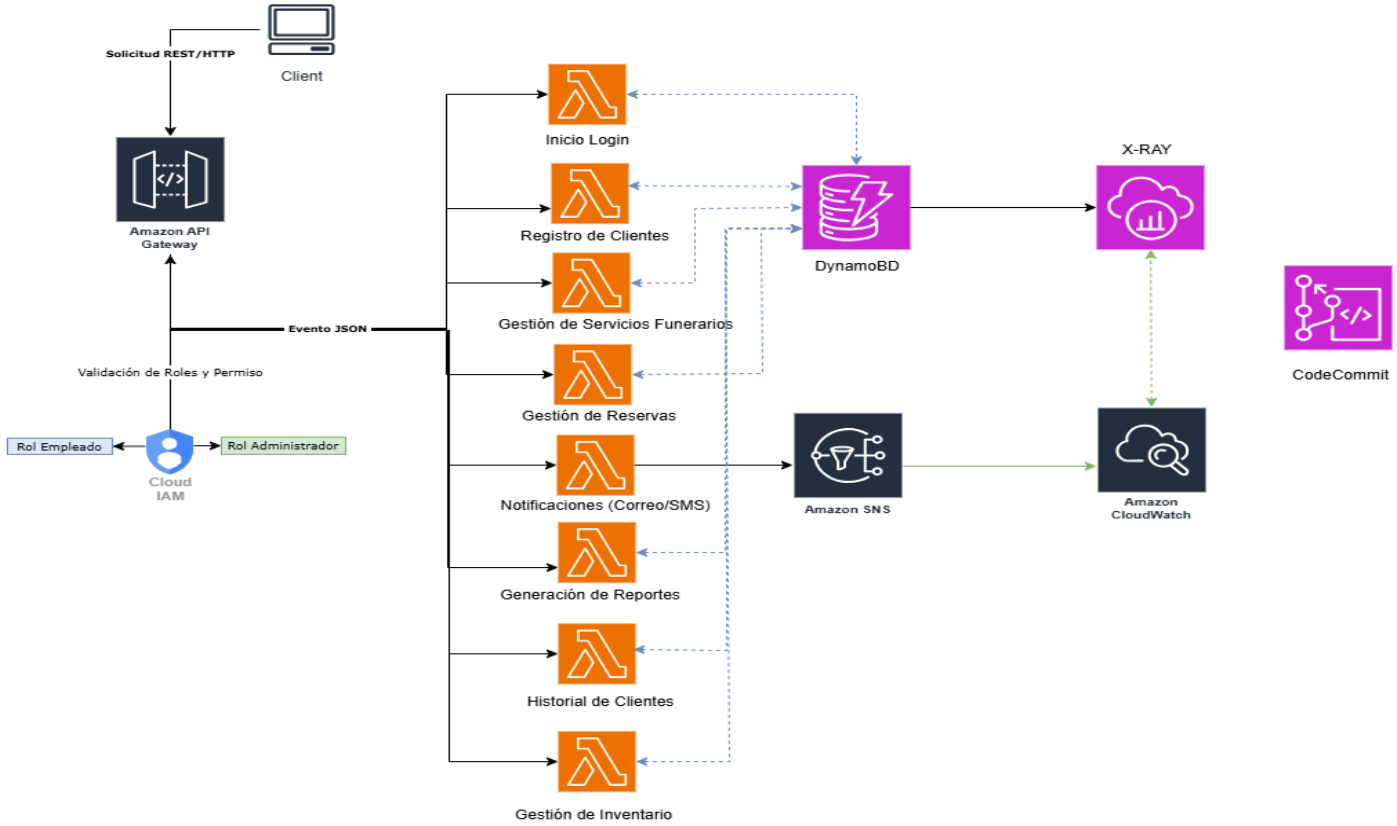
Nota. *Burn Down Sprint 1*

Empezamos con la ejecución de las tareas que contiene el sprint, y así poder evaluar la evolución de las tareas completadas y tener un mejor seguimiento del trabajo del equipo. A continuación, se procede a desarrollar el diagrama de arquitectura de microservicios para la solución del estudio presente.

Diagrama de Arquitectura de Microservicios

Figura 26:

Diagrama de Arquitectura de Microservicios

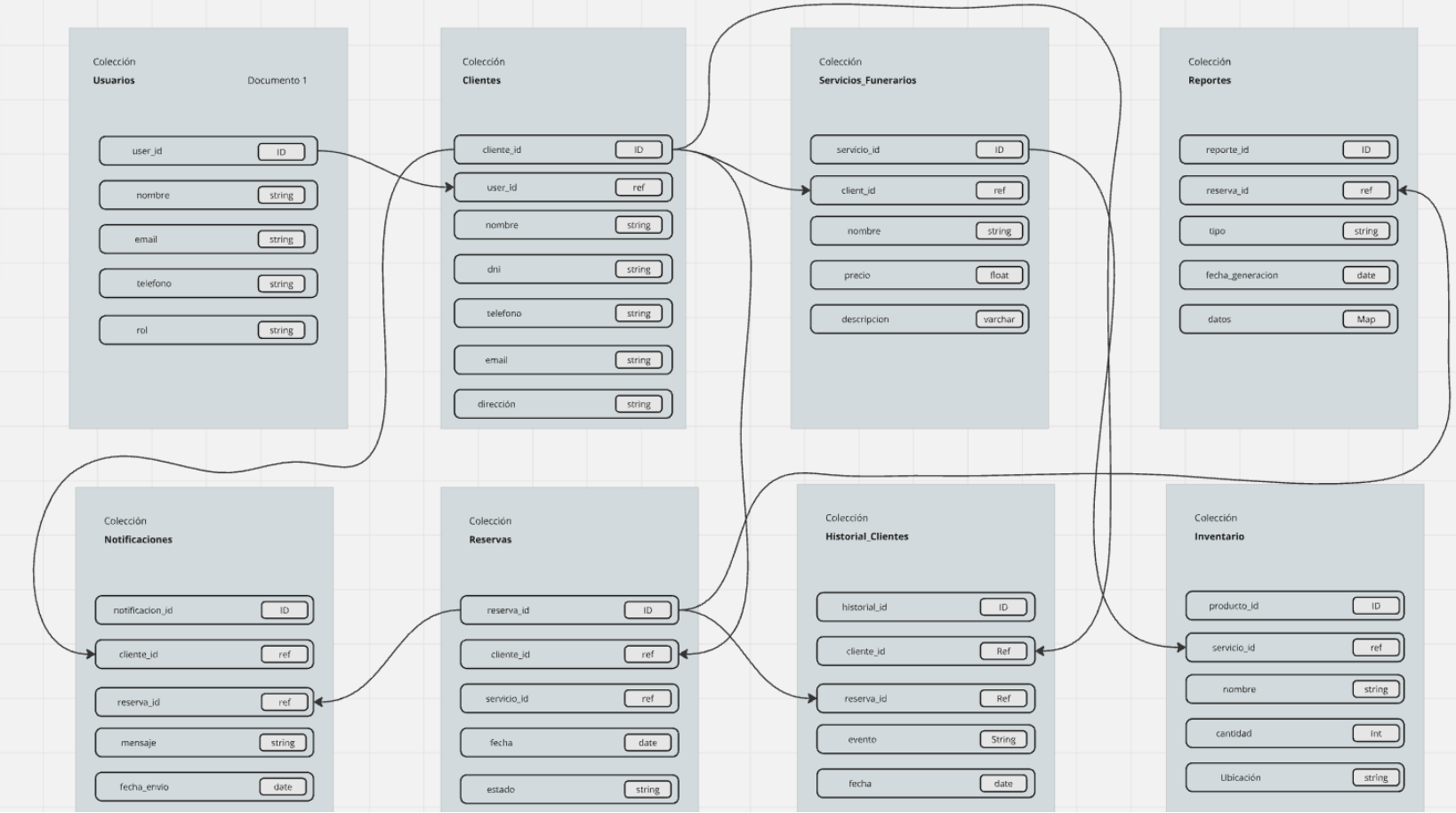


Nota. *Diagrama General*

Diseño de la base de datos

Figura 27*:*

Diseño de la Base de Datos

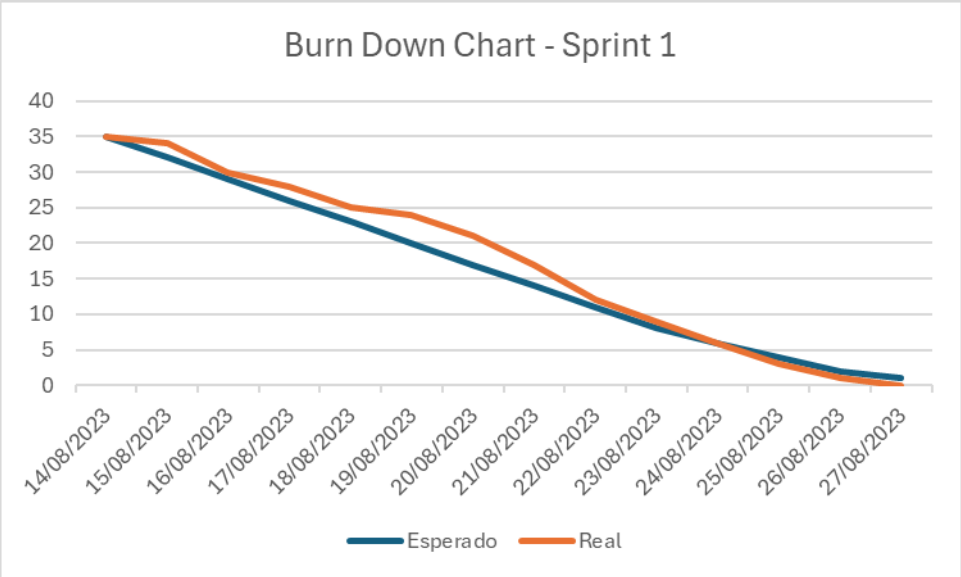


Nota. *Tablas del Negocio*

En tu gráfico, se observa que al inicio la línea real está ligeramente por encima de la esperada, lo que significa que hubo un pequeño retraso en los primeros días. Sin embargo, después del 20/08/23, la línea real se alinea con la esperada, indicando que se recuperó el tiempo y el sprint terminó según lo planeado.

Figura 28:

Actualización del diagrama Brun Down



Nota. *Burn Down Esperado Sprint 1*

Conclusión:

- Hubo un ligero retraso al inicio.
- Se ajustó la velocidad de trabajo en los días siguientes.
- Sprint finalizado correctamente sin afectar la fecha de entrega.

4.5.2. Sprint N°2: Implementación del módulo de gestión de clientes y servicios funerarios.

4.5.2.1.*Planificación del Sprint*.

Para este sprint, se comunicó claramente al equipo Scrum el objetivo principal: desarrollar e implementar el módulo de gestión de clientes y servicios funerarios. La planificación incluyó la definición de las funcionalidades clave, los requerimientos técnicos y los entregables esperados.

Sprint 2: Implementación del módulo de gestión de clientes y servicios funerarios:

Objetivos del Sprint:

- Implementar las funcionalidades necesarias para la gestión de clientes, incluyendo el registro, actualización y eliminación de clientes.
- Desarrollar el módulo de servicios funerarios, permitiendo su registro y administración dentro del sistema.
- Integrar la base de datos DynamoDB con los microservicios para el almacenamiento eficiente de la información.
- Asegurar que la API Gateway conecte correctamente con las funciones Lambda correspondientes.
- Definir los endpoints necesarios para interactuar con el Frontend.
- Garantizar la seguridad de acceso según los roles de usuario (administrador y empleado).

4.5.2.2.Sprint Backlog

Se presentaron las tareas específicas que se desarrollaran en el sprint, estas tareas fueron presentadas en el Product Backlog.

- **Gestión de clientes:**
  - ✓ Crear función Lambda para registrar clientes.
  - ✓ Crear función Lambda para consultar clientes.
  - ✓ Implementar la lógica para editar y eliminar clientes (solo accesible para el administrador).

- ✓ Integrar la funcionalidad en la API Gateway.

- **Gestión de servicios funerarios:**
  - ✓ Crear función Lambda para el registro de servicios funerarios.
  - ✓ Implementar la consulta y actualización de servicios funerarios.
  - ✓ Integrar la funcionalidad en la API Gateway.
- **Seguridad y validaciones:**
  - ✓ Implementar validación de roles en la autenticación de usuarios.
  - ✓ Proteger los endpoints para que solo usuarios autorizados puedan realizar ciertas acciones.
  - ✓ Agregar encriptación de contraseñas en la base de datos.
- **Conexión con el Frontend:**
  - ✓ Diseñar la interfaz de gestión de clientes y servicios funerarios.
  - ✓ Crear formularios para el registro y actualización de datos.
  - ✓ Implementar tablas dinámicas para visualizar la información.

4.5.2.3.Ejecución del Sprint.

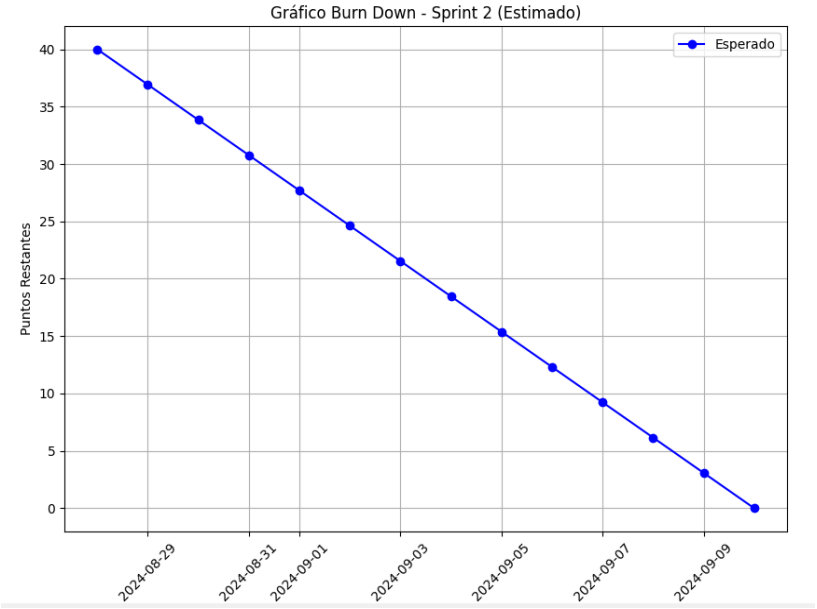
Se realizo la primera reunión diaria para dar el inicio formal de las del trabajo del segundo Sprint, para sincronizar el al equipo y discutir las tareas diarias que se desarrollaran además de los tiempos asignados.

Gráfico Burn Down Estimado

Se presenta el grafico de Burn Down chart con el tiempo esperado para poder completar el Sprint número 2.

Figura 29:

Gráfico Brun Down - Sprint 2

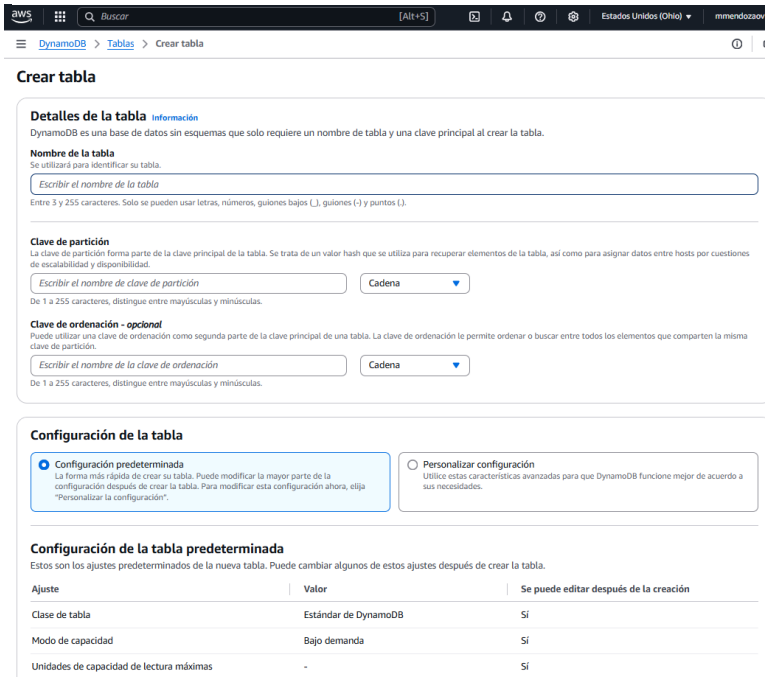


Nota. *Brun Down Sprint 2*

A) Creación de las tablas en DynamoDB

Figura 30:

Creación de tablas en DynamoDB

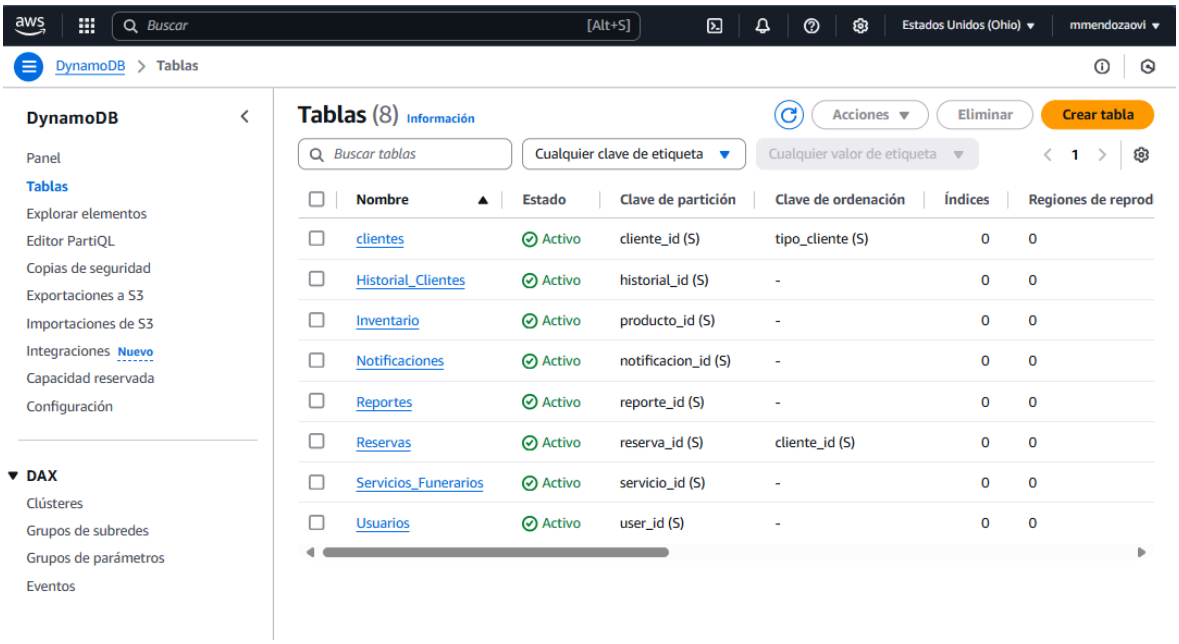


Nota. *Consola AWS*

En DynamoDB, no es necesario especificar nuestros atributos, automáticamente almacenara los atributos dinámicamente.

Figura 31:

Listado de Tablas en DynamoDB



Nota. *Consola AWS Dinamodb*

B) Creación de Login

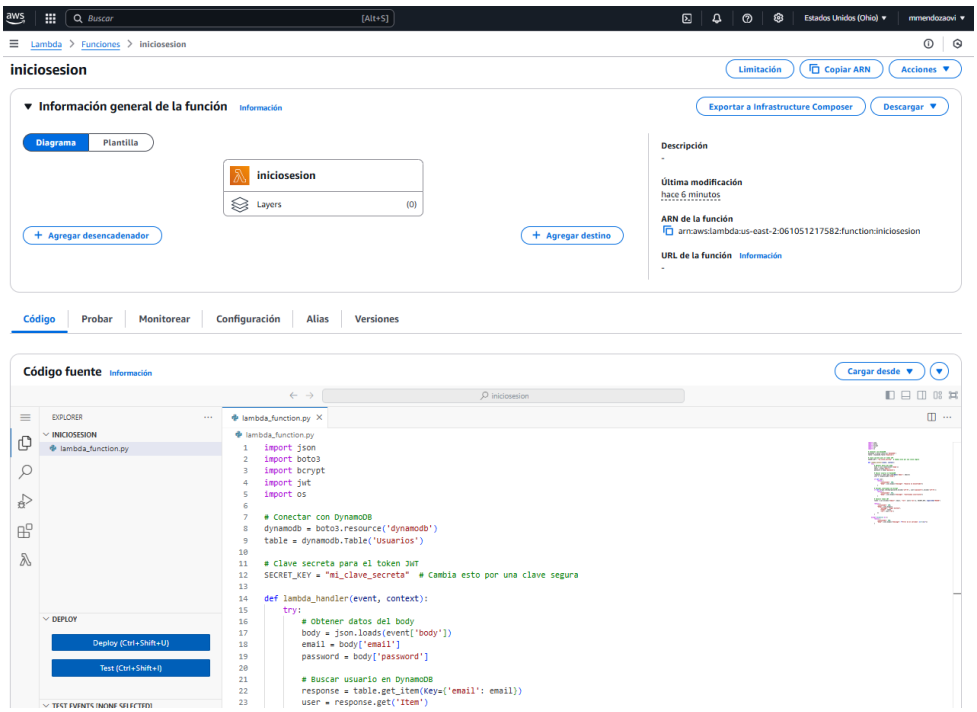
Es necesario listar nuestros métodos REST API:

- POST /login → Lambda de inicio de sesión
- POST /usuarios → Crear usuario
- GET /usuarios/{id} → Obtener usuario
- PUT /usuarios/{id} → Actualizar usuario
- DELETE /usuarios/{id} → Eliminar usuario

Lambda de inicio de sesión, de esta forma realizamos la programación de nuestros microservicios a través de lambdas que se encargarán de cada uno, y contendrán un recurso asignado a la API REST.

Figura 32:

Lambda Inicio de Sesión

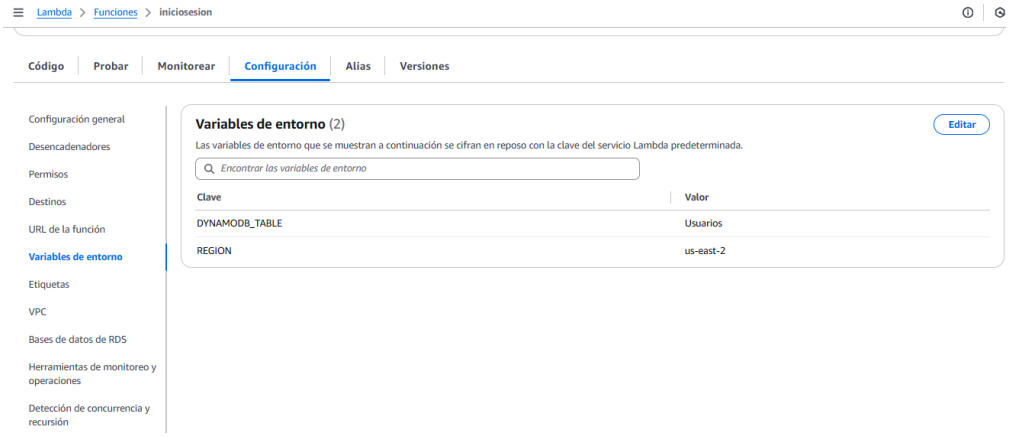


Nota. *Elaboración Propia.*

Definición de variables de entorno en microservicio INICIAR SESIÓN:

Figura 33:

Definición de Variables de Entorno



Nota. *Interfaz AWS Lambda.*

En este apartado, se detalla el proceso de creación y configuración de una API, centrándose en la definición de integraciones de BACKEND, que son los servicios encargados de procesar las solicitudes recibidas por la API. Estas integraciones permiten conectar la API con diferentes recursos, como funciones SERVERLESS (AWS Lambda) o endpoints HTTP externos, actuando como intermediarios entre el cliente y la lógica de negocio.

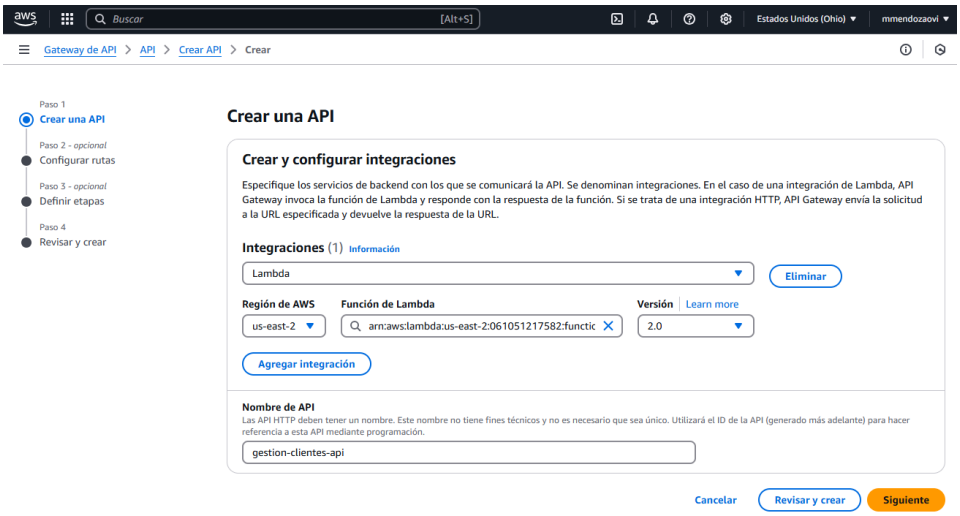
Integraciones configuradas:

- API Gateway se comunica directamente con una función Lambda, invocándola cuando recibe una solicitud. La respuesta de la función se devuelve al cliente como respuesta de la API.
- la API Gateway envía la solicitud a una URL configurada (por ejemplo, un microservicio o API REST externa) y retorna la respuesta al cliente.
- Se define un nombre descriptivo (*gestión-clientes-api*) para identificar la API.

Esta estructura refleja un patrón de diseño API Gateway, clave en arquitecturas modernas como la de microservicios.

Figura 34:

Creación de API

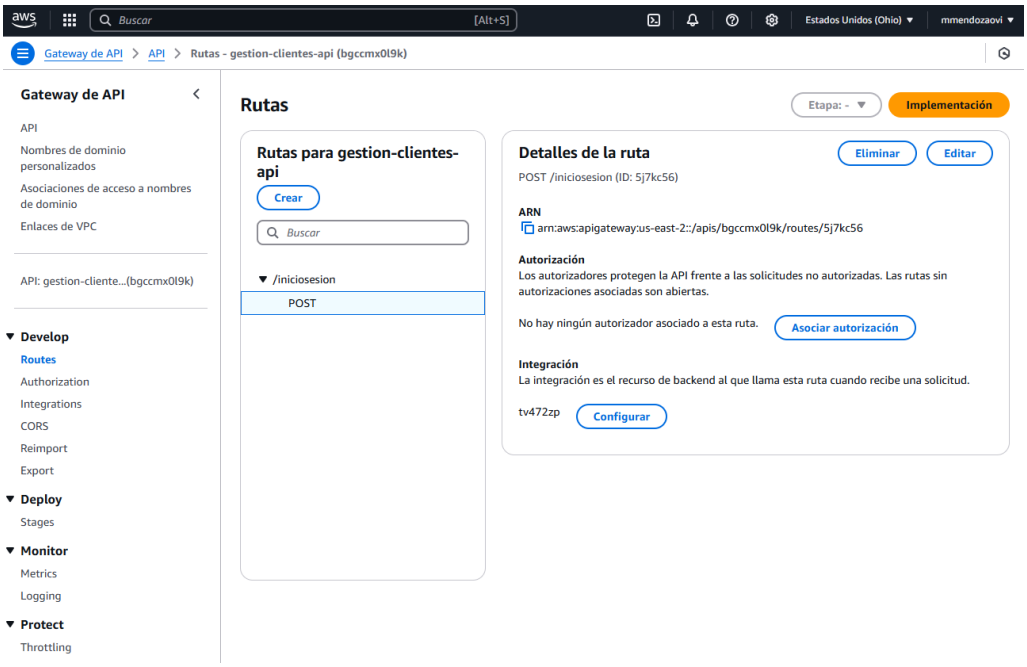


Nota. *Configuración API AWS*

En este apartado, configuramos las rutas y el método especifico, La Figura 35 ilustra la configuración de una ruta POST en API Gateway, con integración a sistemas municipales y control jerárquico de acceso, alineándose con estándares de gobernanza de APIs (Amazon, 2023).

***Figura 35**:*

Definición de rutas APIS



Nota. *Métodos HTTP*

Vista de la configuración final de la función Lambda encargada del inicio de sesión, con su integración a API Gateway.

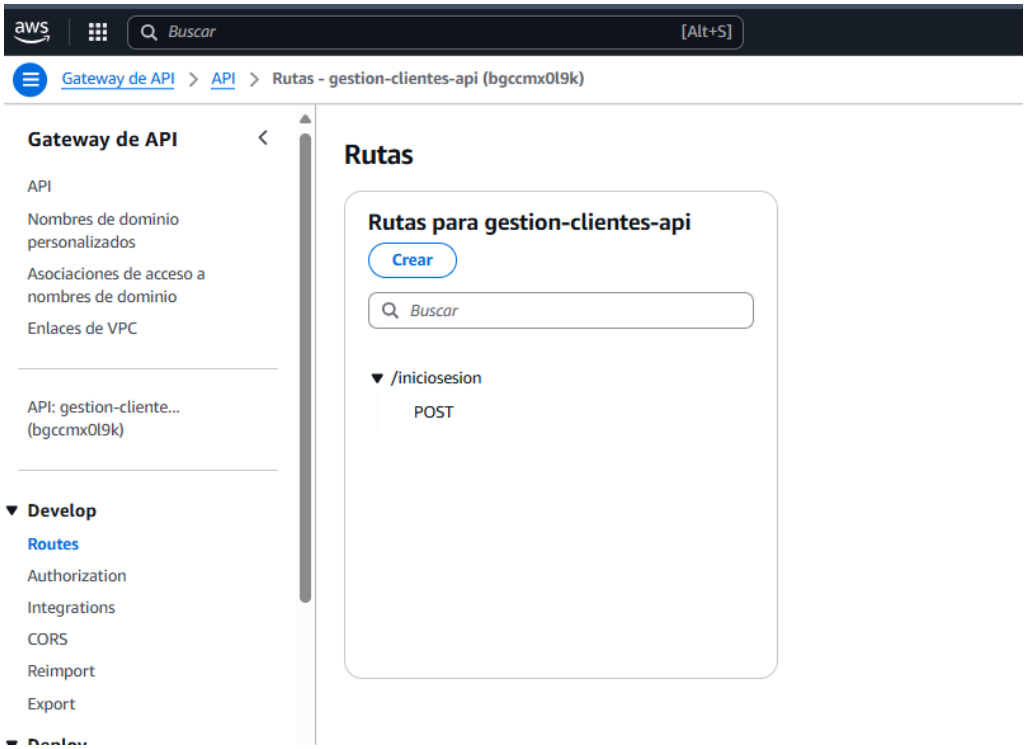
Figura 36:

Lambda Inicio Sesión Finalizada

Nota. *Despliegue funcional de inicio sesión.*

Figura 37:

Ruta de API Gestión Clientes

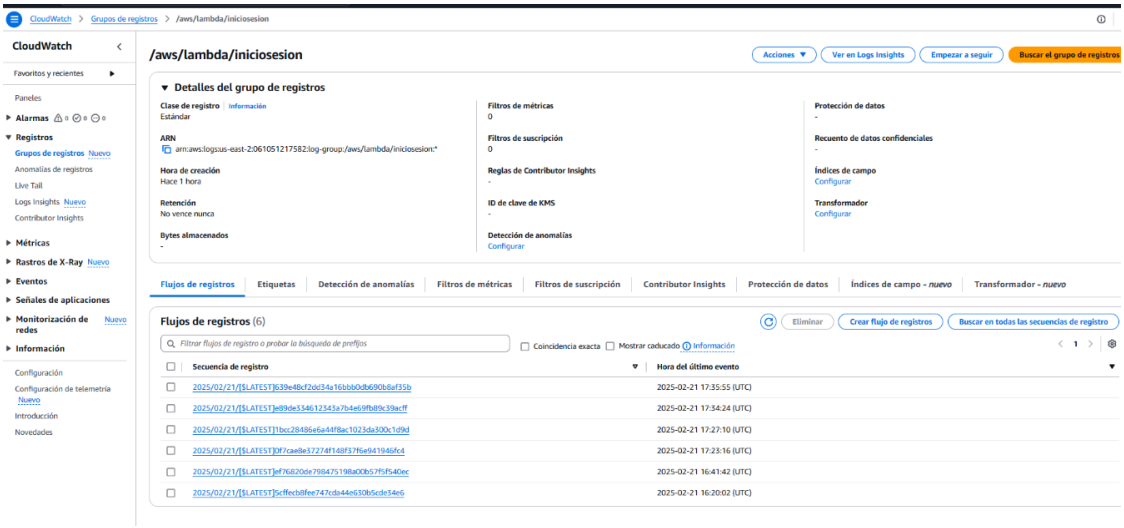


Nota. *Métodos HTTP*

A continuación, en la figura 38, se realiza el seguimiento de LOG mediante AWS CLOUDWATCH.

Figura 38:

CloudWatch de Lambda Inicio Sesión

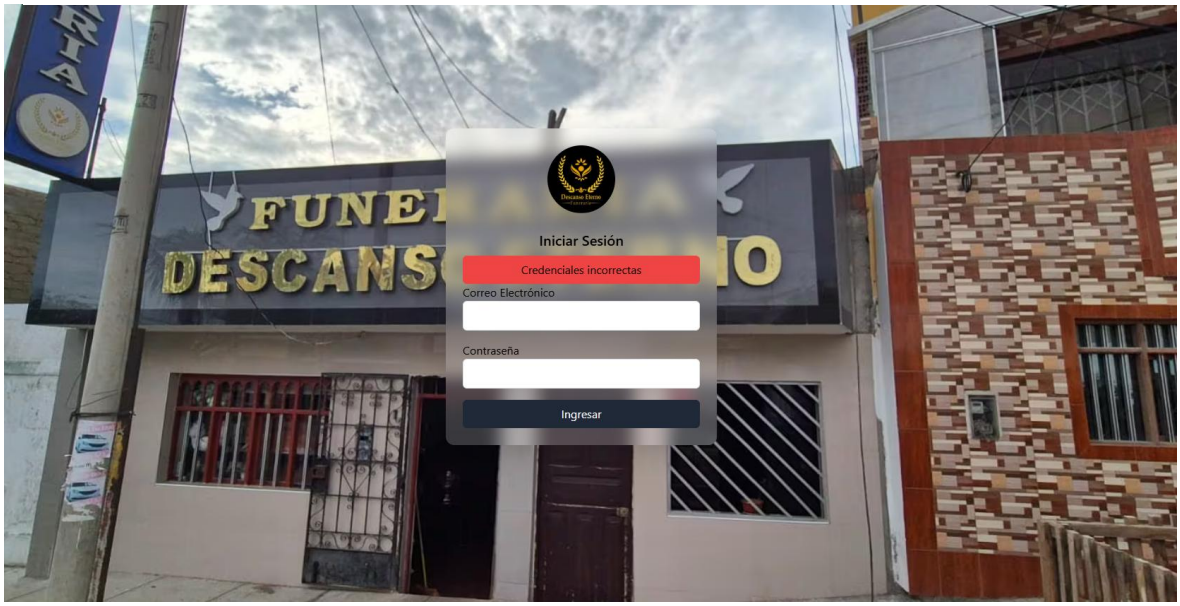


Nota. *Interfaz de AWS CloudWatch*

En la siguiente figura 39, se realizó la interfaz en desarrollo frontend para realizar las pruebas y poder presentarlas en la review.

Figura 39:

Interfaz de Inicio Sesión



Nota. *Interfaz de la APP*

Figura 40:

Salida de Consola Del Event JSON - INICIO SESION

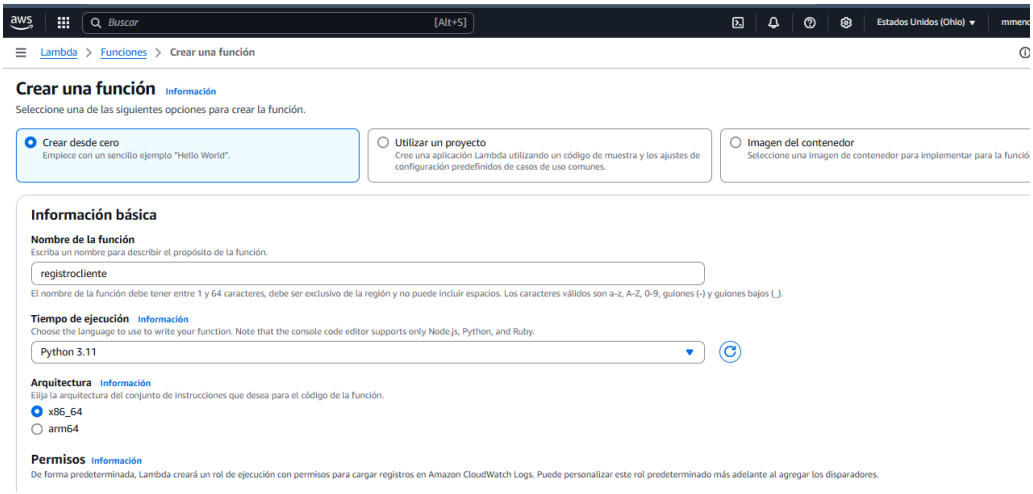
Nota. *Terminal del Visual Studio Code AWS*

C) Registro de Clientes

En la siguiente Figura 41, se realiza la creación del Lambda Registro de los clientes para la funeraria, se sigue los pasos de la creación del Login y asi sucesivamente.

Figura 41:

Creación Lambda Registro Clientes

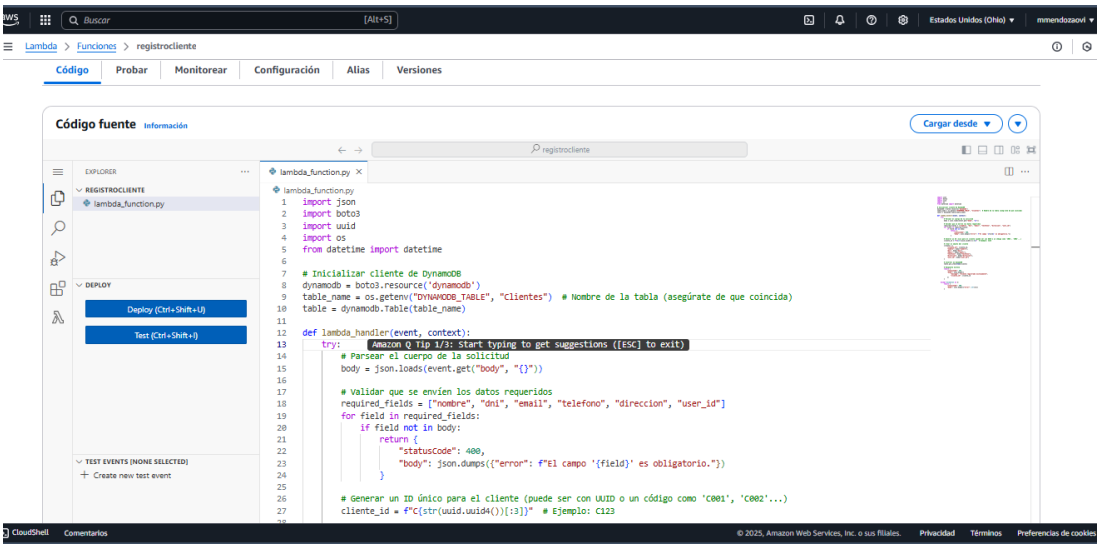


Nota. *Elaboración Propia*

En la figura 42 se realiza la implementación de la función Lambda denominada *registrocliente*, el propósito es recibir y procesar datos de usuarios para almacenarlos en una tabla DynamoDB. La función incluye validación de campos requeridos y la generación de un ID.

Figura 42:

Programación del Lambda Registros Clientes

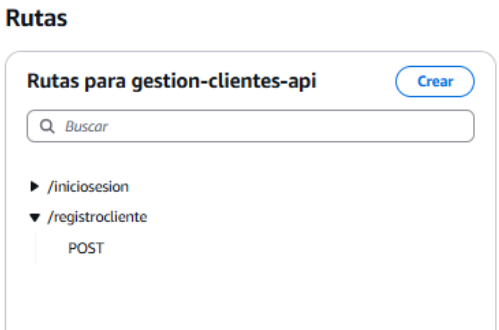


Nota. *Implementación en AWS Lambda con integración a DynamoDB.*

A continuación se muestra la configuración de rutas dentro del API Gateway, donde se define el endpoint /registrocliente asociado al método POST, encargado de recibir solicitudes para el registro de nuevos clientes.

Figura 43:

Definición de ruta API Registro Clientes

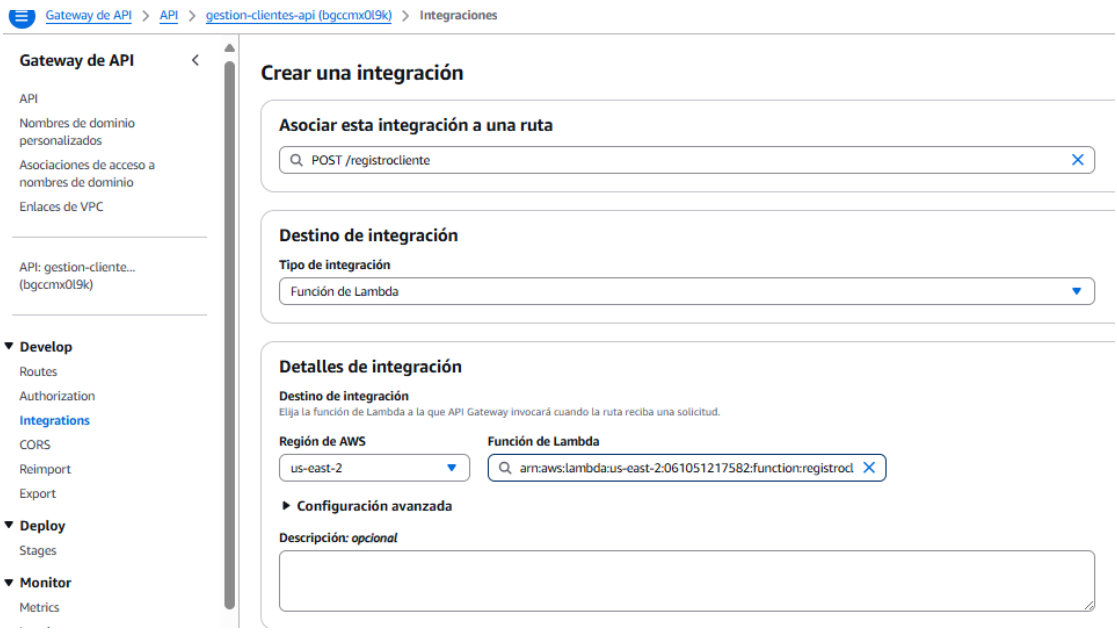


Nota. *Configuración del API Gateway registrocliente.*

En esta figura 44 se muestra el proceso de integración de una función Lambda con el servicio de API Gateway de AWS, correspondiente a la ruta POST /*registrocliente*. Esta integración permite que, al realizar una solicitud HTTP POST a dicha ruta, se active automáticamente la función Lambda encargada del registro de clientes.

Figura 44*:*

Integración API GATEWAY con Lambda Registro Cliente



Nota. *Integración Api Gateway y función Lambda registrocliente.*

En esta figura 45 se visualiza la integración de la función Lambda llamada *registrocliente* con el servicio de API Gateway. Esta conexión permite que las solicitudes HTTP recibidas por el endpoint correspondiente activen automáticamente la función para procesar el registro de un nuevo cliente. Esta configuración es clave en la implementación del microservicio encargado de la gestión de clientes.

Figura 45:

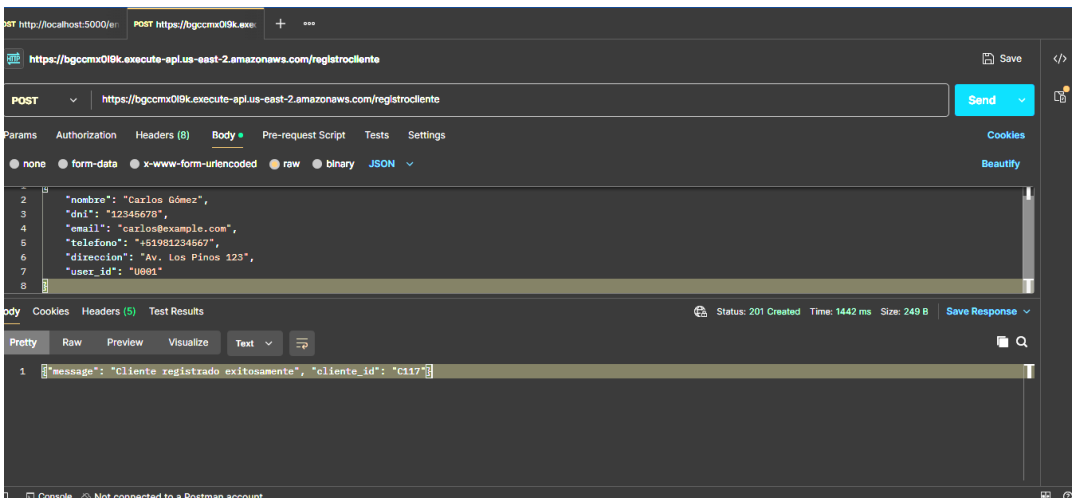
Creación Completa de Lambda Registro Clientes

Nota. *Flujo de la función lambda registrocliente*

En esta figura 46 se presenta la prueba del microservicio de registro de clientes utilizando la herramienta Postman. Se envió una solicitud HTTP POST al endpoint expuesto por API Gateway (/registrocliente) con los datos de un nuevo cliente en formato JSON. Como respuesta, se obtuvo un código de estado 201 Created, confirmando que el cliente fue registrado exitosamente. Además, se devolvió un mensaje de confirmación junto con el cliente_id generado, lo cual valida el correcto funcionamiento de la función Lambda y su integración con el backend.

Figura 46:

Prueba exitosa en Postman del Lambda Registro Cliente

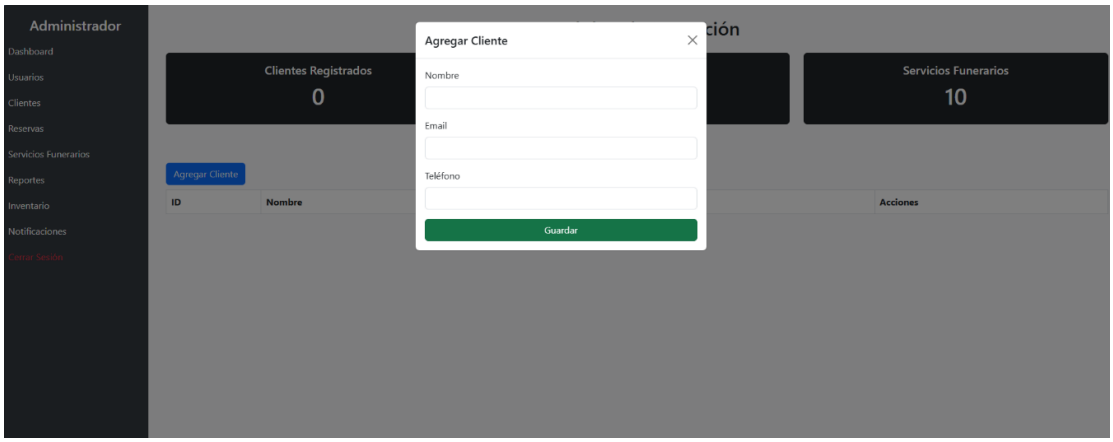


Nota. *Utilización de la herramienta Postman.*

En la figura 47, se desarrolló la interfaz del módulo de administración permite agregar nuevos clientes mediante un formulario emergente que solicita los datos esenciales: nombre, correo electrónico y número de teléfono.

Figura 47*:*

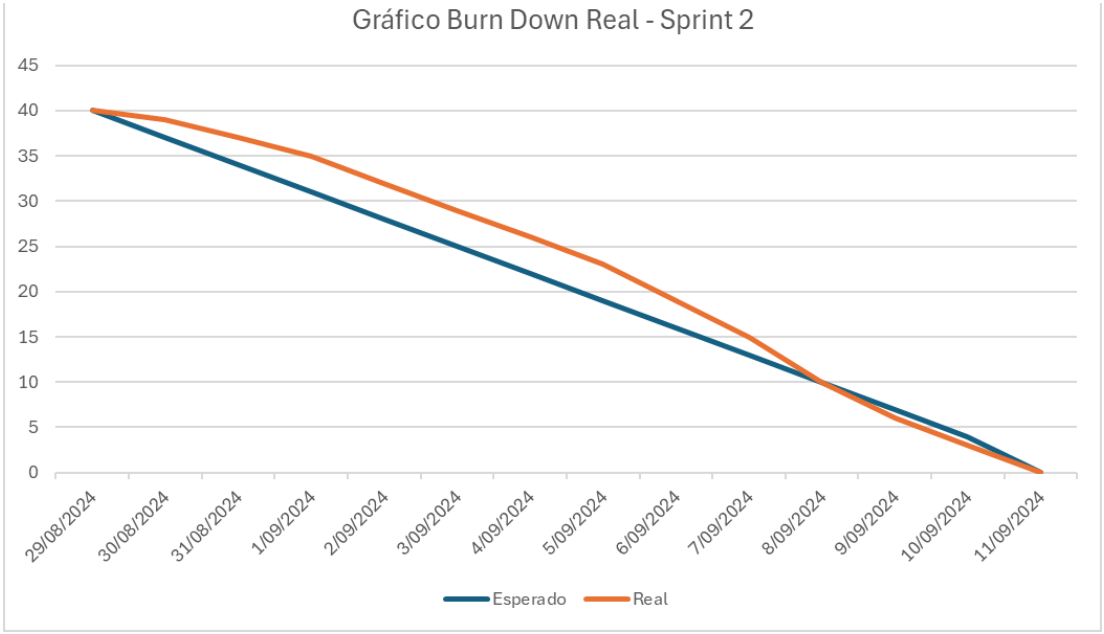
Frontend de Registros Clientes



Nota. *Interfaz del sistema desarrollada para gestión administrativa.*

Figura 48*:*

Gráfico Brun Down Real - Sprint 2



Nota. *Seguimiento del desempeño del equipo durante el Sprint 2.*

El Gráfico de la figura 48 Burn Down Real del Sprint 2 refleja que al inicio hubo un ligero retraso en la ejecución de tareas, lo que se evidencia en la línea real que se encuentra por encima de la estimada en los primeros días. Esto puede deberse a ajustes en la planificación o dificultades iniciales. Sin embargo, a partir del 06/09/2024, el equipo recuperó el ritmo, acelerando el desarrollo y alineándose progresivamente con la proyección esperada. Finalmente, el sprint concluyó dentro del plazo establecido, logrando completar todas las tareas planificadas sin extenderse más allá del 09/09/2024. Este comportamiento es común en proyectos con ajustes iniciales y una curva de aprendizaje, pero demuestra una buena capacidad de recuperación y gestión del tiempo en el equipo de desarrollo.

4.5.3. Sprint N°3: Desarrollo del módulo de reservas y notificaciones automáticas (Correo/SMS)

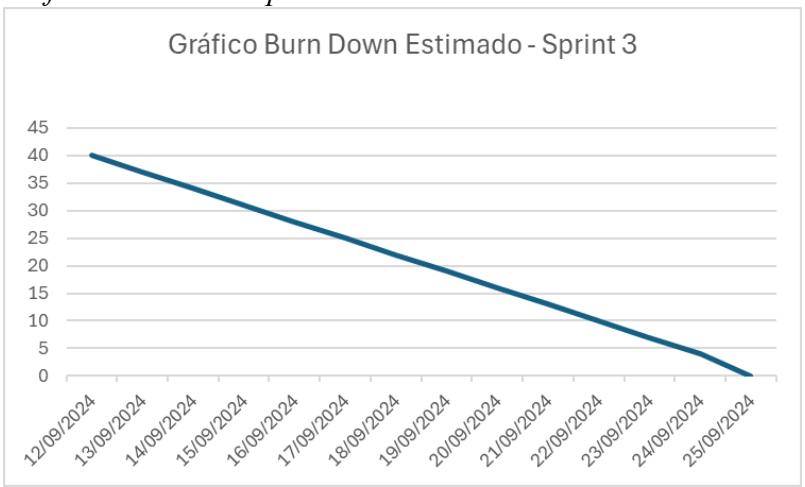
Para el módulo de reservas, necesitaremos crear 3 lambdas:

- *CREARRESERVA* → Registra una nueva reserva.
- *CONSULTARRESERVA* → Recupera una reserva específica o todas las reservas.
- *CANCELARRESERVA* → Permite cancelar una reserva, solo si el usuario tiene permisos.

En la siguiente Figura 49, se presenta mediante la siguiente grafica un estimado de las actividades durante el sprint 3.

Figura 49*:*

Gráfico Estimado - Sprint 3



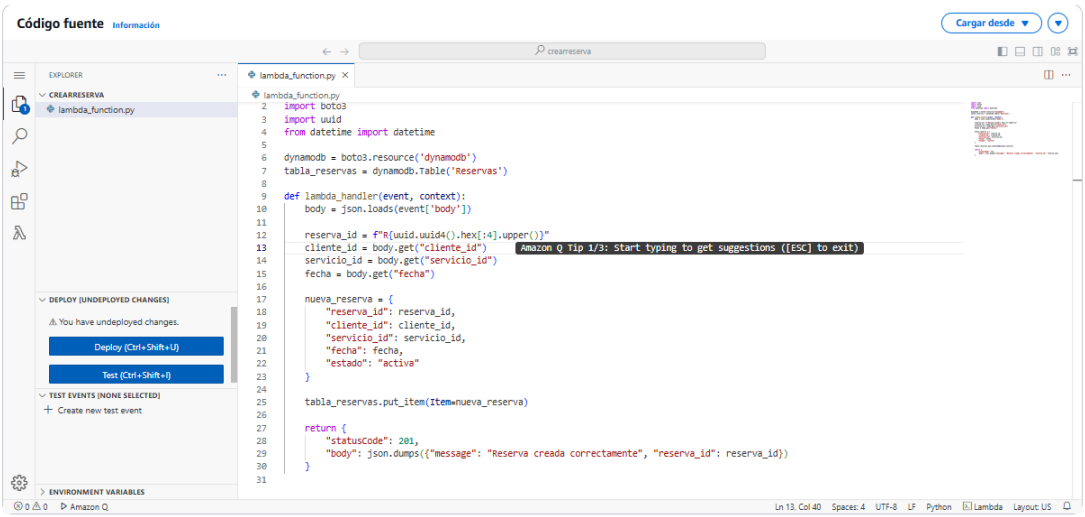
Nota. *Estimación inicial de tareas prevista en el backlog del Sprint 3.*

A) Creación de Lambda ***CREARRESERVA:***

La figura 50 muestra el código fuente de la función Lambda CREARRESERVA, encargada de registrar una nueva reserva en la tabla DynamoDB denominada Reservas. La función genera un identificador único y almacena los datos asociados al cliente, servicio y fecha.

Figura 50*:*

Creación de Lambda Crear Reserva



Nota. *Implementación del endpoint de creación de reservas.*

Explicación:

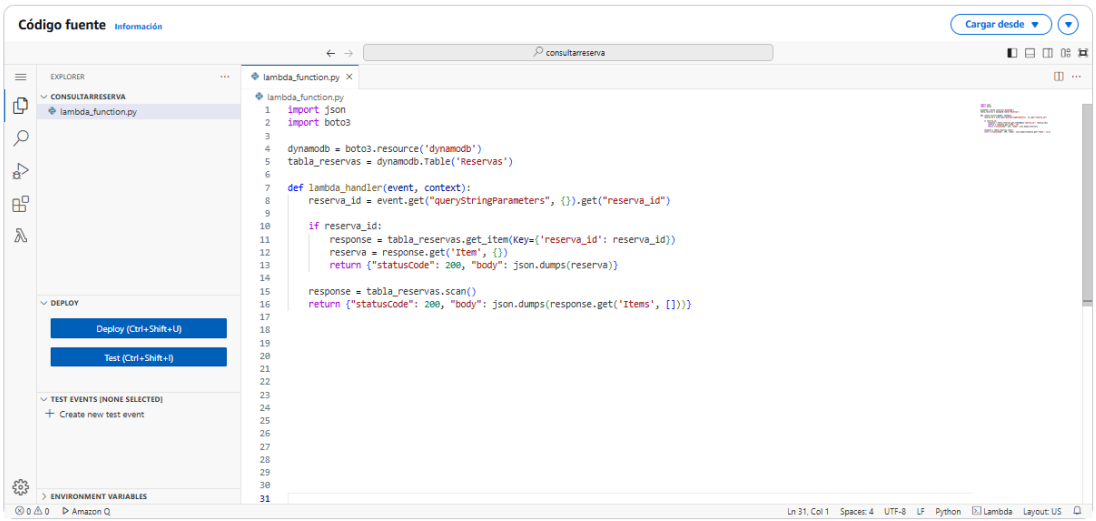
- Genera una reserva_id único.
- Guarda la reserva en DynamoDB con estado "activa".
- Retorna la reserva_id generado.

B) Creación de Lambda ***CONSULTARRESERVA:***

La figura 51 presenta la implementación de la función Lambda CONSULTARRESERVA, la cual permite recuperar una reserva específica mediante su identificador o listar todas las reservas registradas en la base de datos DynamoDB.

Figura 51:

Creación de Lambda Consulta Reserva



Nota. *Elaboración Propia*

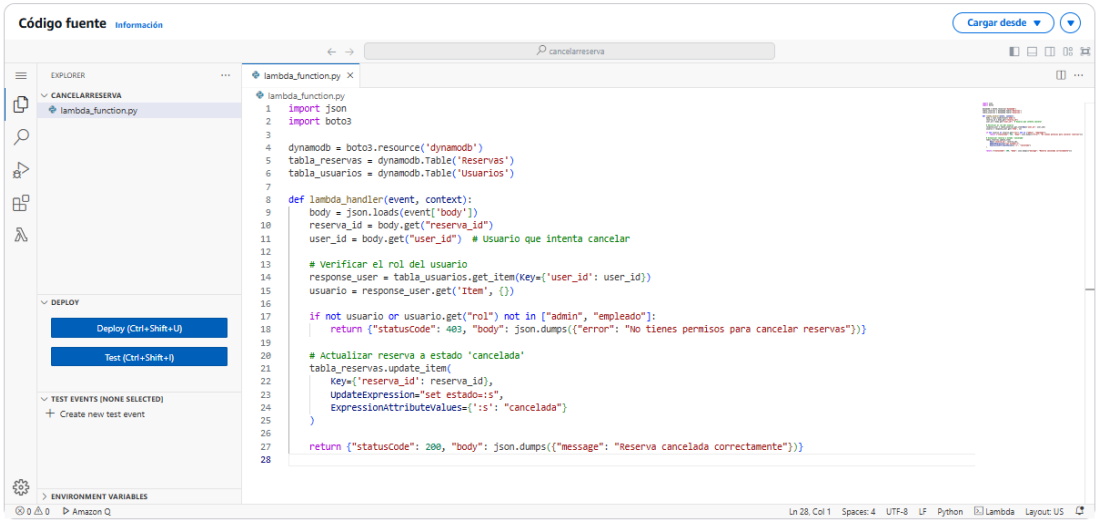
Explicación:

- Devuelve una reserva específica si se proporciona reserva_id.
- Si no se proporciona, devuelve todas las reservas.

C) Creación de Lambda ***CANCELARRESERVA***

La figura muestra el código fuente de la función Lambda CANCELARRESERVA, que valida el rol del usuario antes de permitir la cancelación de una reserva. Solo los usuarios con rol “admin” o “empleado” pueden ejecutar esta acción.

Figura 52: *Creación de Lambda Cancela Reserva*



Nota. *Función Lambda desarrollada para gestionar cancelaciones.*

Explicación:

- Admin y empleado pueden cancelar reservas.
- Cambia el estado de la reserva a "cancelada".

D) Creación de Lambda ***NOTIFICARCLIENTE***

Para desarrollar el módulo de notificaciones automáticas, notificar a los clientes sobre sus reservas por correo o SMS usando AWS SNS.

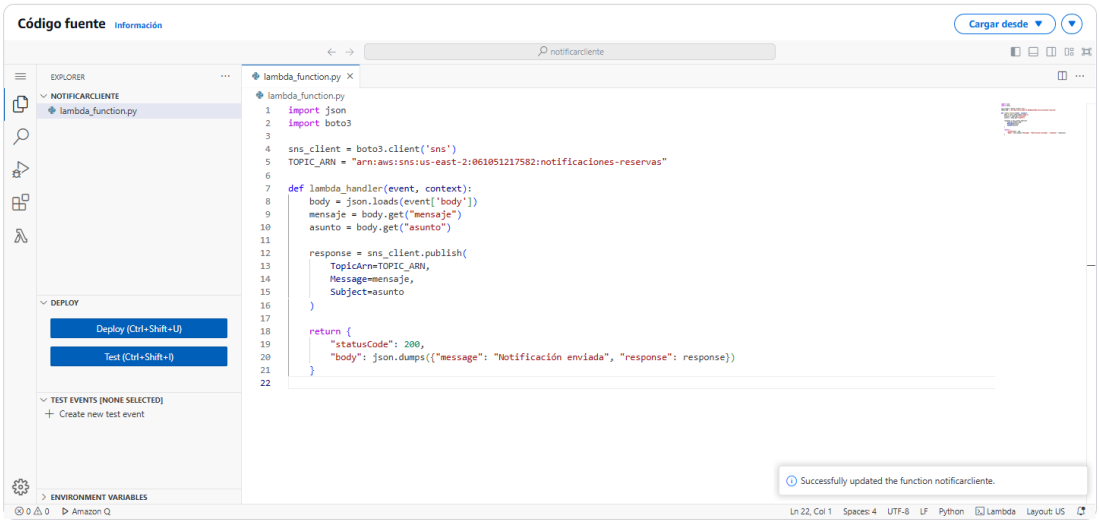
Ejemplo de notificaciones:

- Cuando un cliente registra una reserva, recibe un SMS/email de confirmación.

- Cuando una reserva se cancela, se notifica al cliente y al administrador.

Figura 53:

Creación de Lambda Notificar Clientes

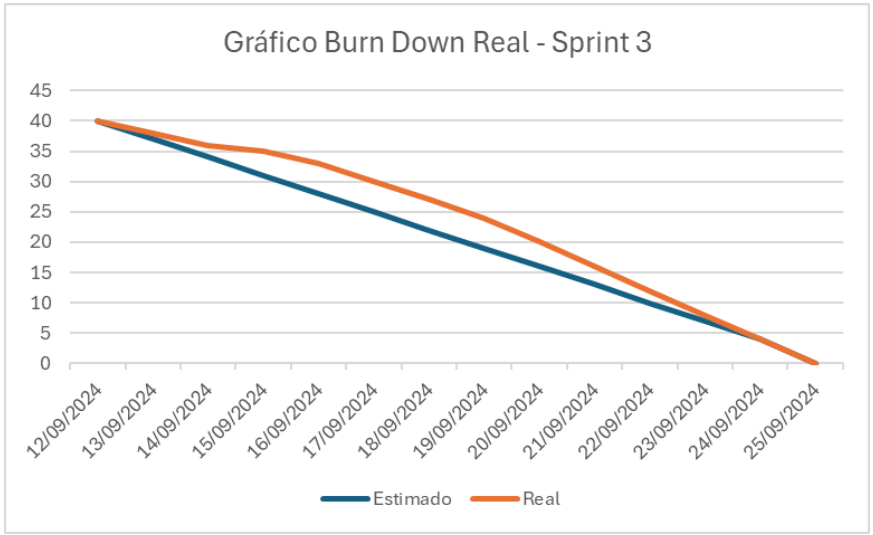


Nota. *Función Lambda para el envío de notificaciones mediante Amazon SNS.*

El gráfico 54, grafico estimado preveía una reducción constante de los puntos del Sprint. Sin embargo, en la línea real se observa un pequeño retraso en los primeros días debido a ajustes en la planificación de tareas y la integración del módulo de notificaciones automáticas. A partir del 17/09/24, el equipo optimizó la ejecución de tareas, logrando recuperar el tiempo perdido y alineando la línea real con la esperada hacia el final del Sprint.

Figura 54:

Gráfico Brun Down Real - Sprint 3



Nota. *Brun Down Sprint 3*

4.5.4. Sprint N°4: Implementación del módulo de reportes y dashboard de gestión

Este sprint se enfoca en la generación de reportes y el desarrollo un reporte de gestión, cubriendo la siguiente historia de usuario:

H6: Generación de Reportes: Permitir a los administradores generar reportes sobre el uso del sistema, registros de clientes, reservas, servicios y más.

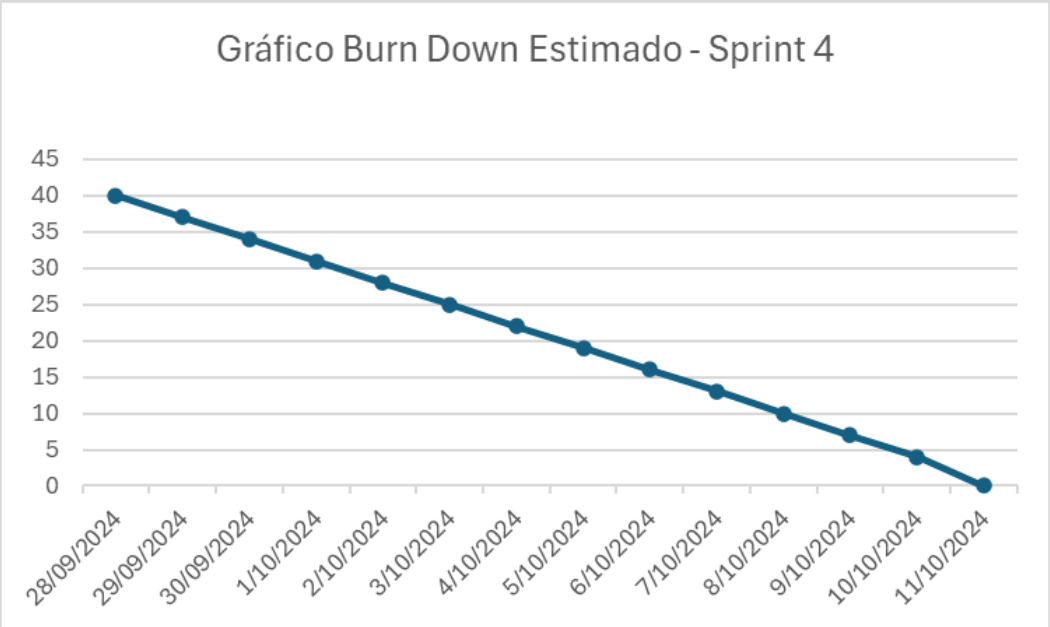
Implementaremos dos Lambdas principales:

- *GENERARREPORTE* → Genera un nuevo reporte y lo guarda en DynamoDB.
- *OBTENERREPORTES* → Permite consultar reportes generados.

De acuerdo con la siguiente figura 55, se presenta un estimado para el presente sprint.

Figura 55:

Gráfico Estimado - Sprint 4



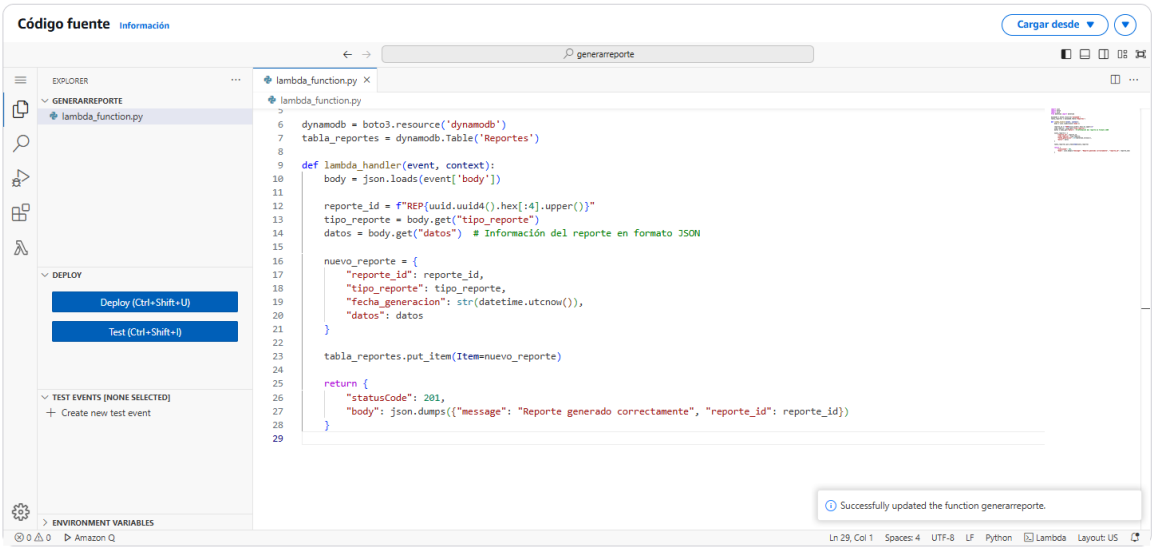
Nota. *Estimado de actividades.*

A) Creación de Lambda ***GENERARREPORTE***

La figura 56 muestra el código de la función Lambda GENERARREPORTE, diseñada para registrar reportes en una tabla DynamoDB. Cada reporte incluye un identificador único, tipo de reporte, fecha de generación y los datos entregados en formato JSON.

Figura 56*:*

Creación del Lambda Generar Reporte



Nota. *Función Lambda para la generación y almacenamiento de reportes.*

Explicación:

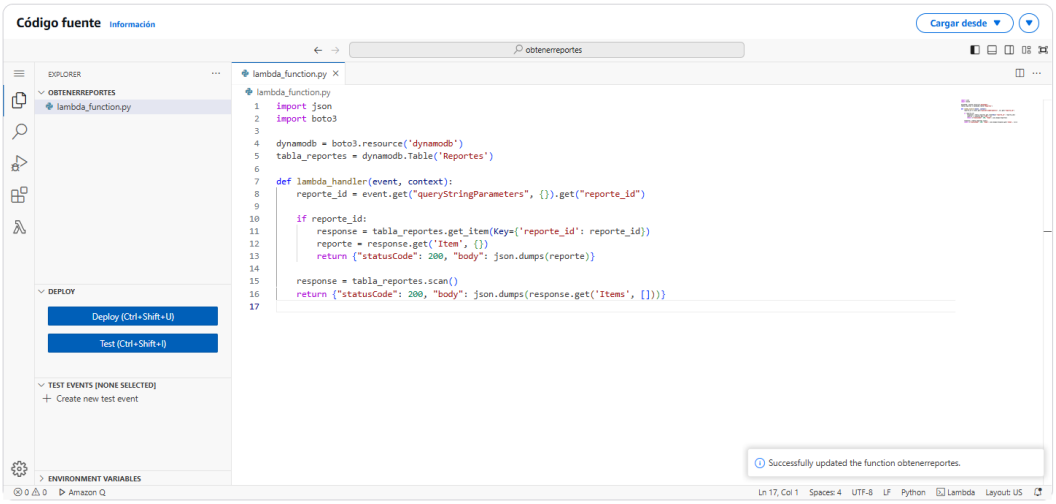
- Genera un ID único para el reporte.
- Almacena el reporte con la fecha de generación.
- Permite almacenar información dinámica en datos.

B) Creación de Lambda *OBTENERREPORTES*

La figura 57 muestra la función Lambda OBTENERREPORTES, la cual permite recuperar un reporte específico mediante su identificador o, en su defecto, listar todos los reportes almacenados en la tabla DynamoDB Reportes.

Figura 57*:*

Creación del Lambda Obtener Reportes



Nota. *Función Lambda para la lectura de reportes DynamoDB*

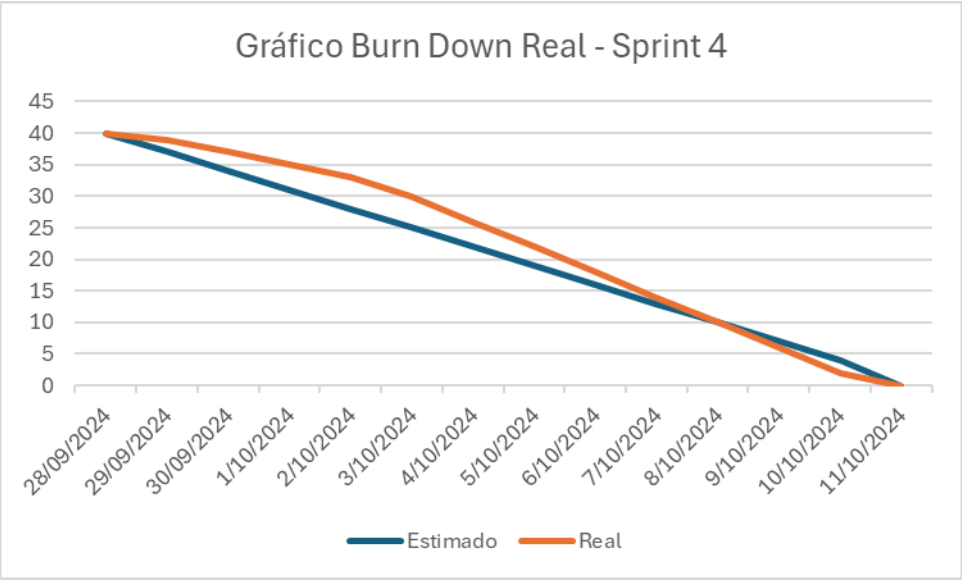
Explicación:

- Si se envía reporte_id, devuelve un solo reporte.
- Si no se envía parámetro, devuelve todos los reportes.

En este Sprint, el gráfico 58 estimado preveía una reducción constante de los puntos de trabajo a lo largo de las dos semanas. Sin embargo, en la línea real, se observa un retraso en los primeros días debido a problemas en la integración de los reportes dinámicos y ajustes en el diseño del Dashboard de gestión.

Figura 58:

Gráfico Brun Down Real - Sprint 4



Nota. *Review Final del Sprint4*

Este retraso inicial hizo que la línea real estuviera por encima de la esperada hasta el 03/10/24. A partir del 05/10/24, el equipo implementó soluciones más eficientes en la generación de reportes y optimizó la visualización del dashboard, lo que permitió recuperar el ritmo de trabajo. Para el 10/10/24, el sprint casi se había alineado con lo esperado, finalizando con éxito dentro del tiempo planificado.

4.5.5. Sprint N°5: Creación del módulo de historial de clientes e inventario de productos funerarios

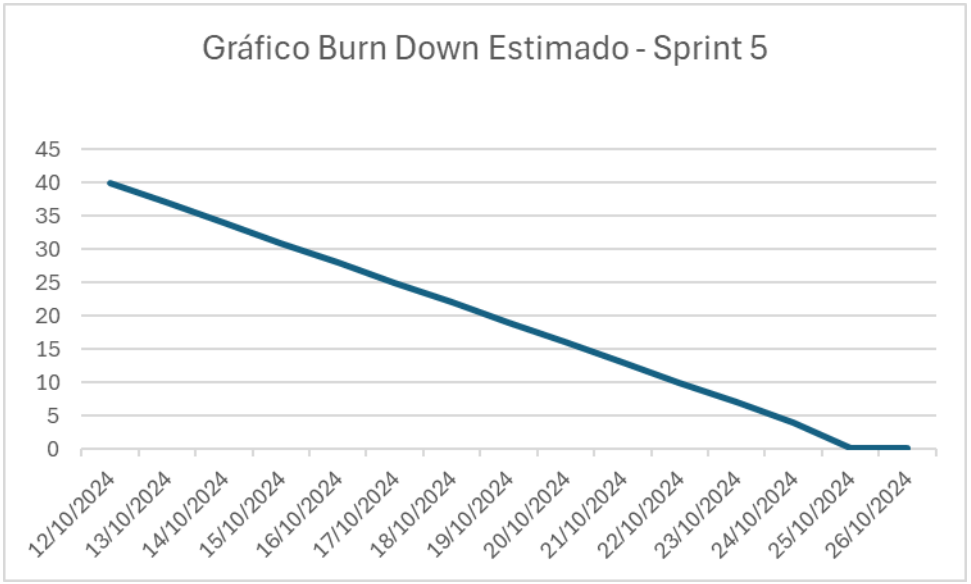
Este sprint cubre las siguientes historias de usuario:

- H7: Historial de clientes → Registrar y consultar eventos relacionados con clientes (reservas, pagos, interacciones).
- H8: Gestión de inventario → Controlar el stock de productos funerarios y facilitar su actualización.

De acuerdo con la siguiente figura 59, se presenta un estimado para el presente sprint.

Figura 59:

Gráfico Brun Down Estimado - Sprint 5



Nota. *Estimación de actividades Sprint 5.*

Las funciones Lambda para este sprint incluyen:

Tabla 24:

Funciones Lambda Sprint 5

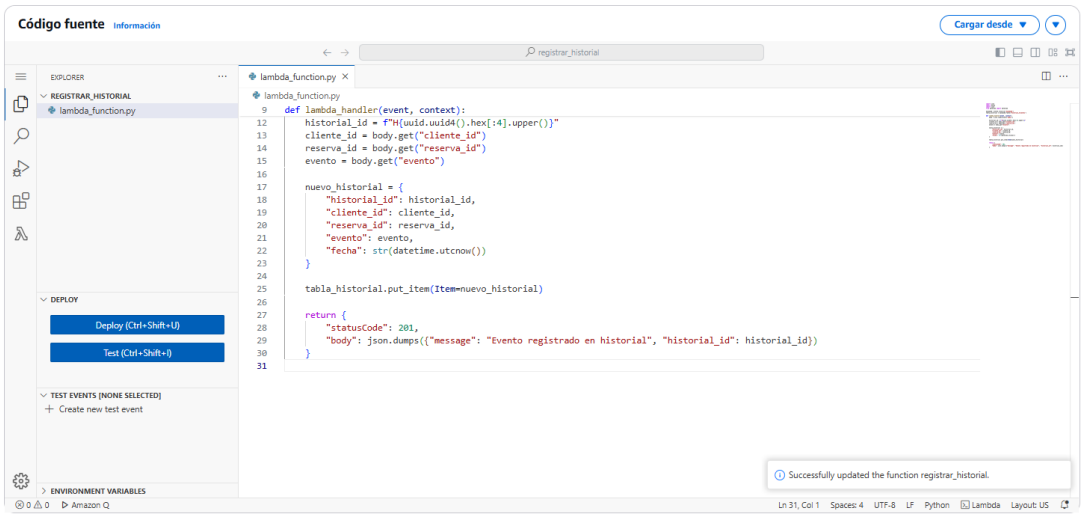
Lambda	Función
REGISTRAR_HISTORIAL	Registra eventos en el historial del cliente
OBTENER_HISTORIAL	Obtiene eventos del historial de un cliente
AGREGAR_PRODUCTO	Agrega productos al inventario
ACTUALIZAR_STOCK	Actualiza la cantidad de productos en inventario
OBTENER_INVENTARIO	Consulta productos del inventario

Nota. *Lambdas en AWS*

A) Lambda *REGISTRAR_HISTORIAL*

Figura 60:

Creación de Lambda Historial



Nota. *Visual Studio Code*

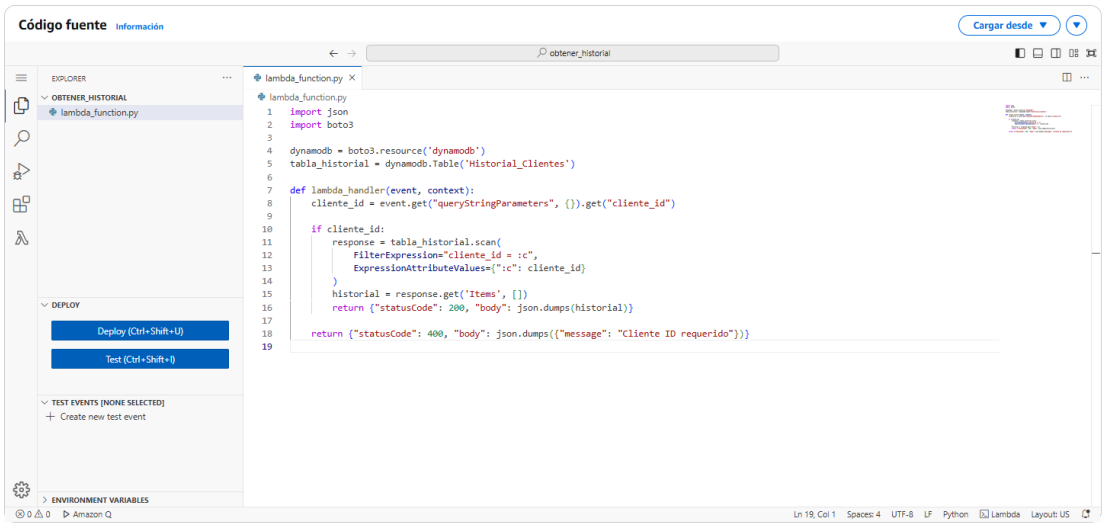
Explicación:

- Genera un historial_id único.
- Registra eventos (reserva confirmada, pago realizado, etc.).
- Guarda la fecha del evento.

B) Lambda OBTENER_HISTORIAL

Figura 61*:*

Creación Lambda Obtener Historial



Nota. *Visual Studio Code Lambda Historial*

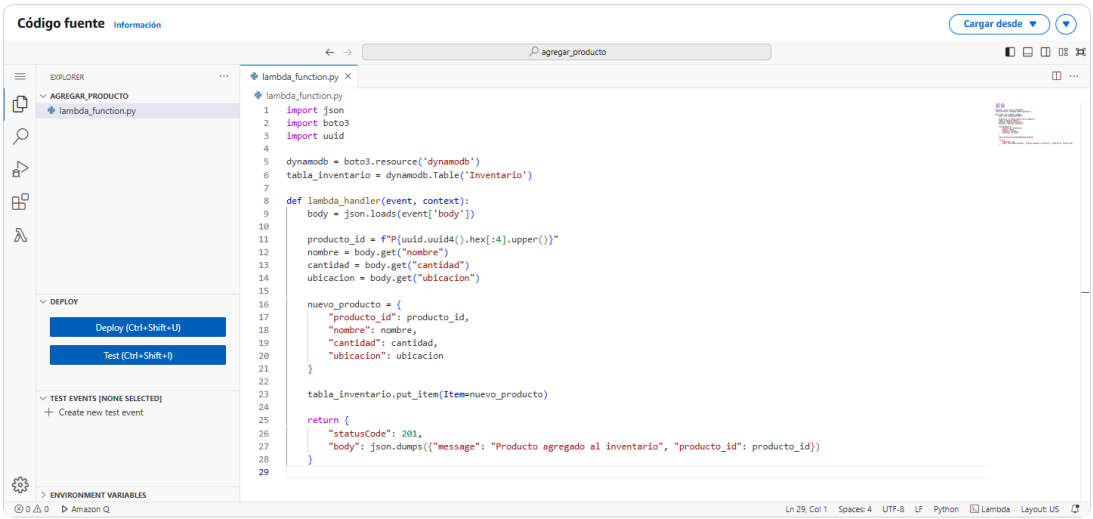
Explicación:

- Obtiene eventos del historial de un cliente.
- Consulta por cliente_id.

C) Lambda *AGREGAR_PRODUCTO*

Figura 62:

Creación Lambda Agregar Producto



Nota. *VSC Lambda Agregar Producto*

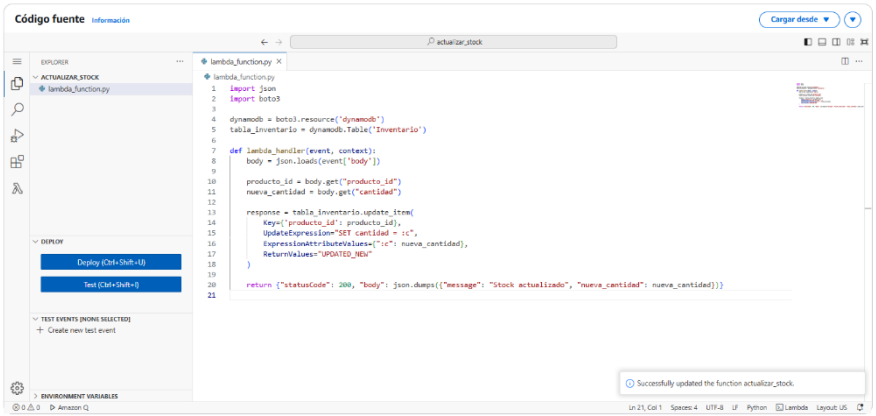
Explicación:

- Genera un producto_id único.
- Permite registrar nuevos productos en el inventario.

D) Lambda *ACTUALIZAR_STOCK*

Figura 63:

Creación Lambda Actualizar Stock

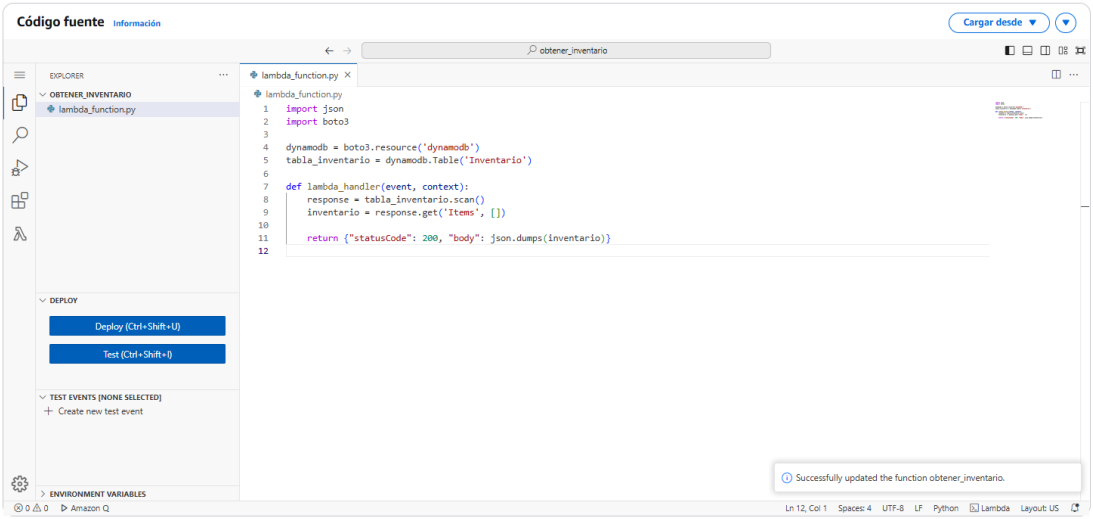


Nota. *VSC Lambda Stock*

E) Lambda *OBTENER_INVENTARIO*

Figura 64:

Creación Lambda Obtener Inventario

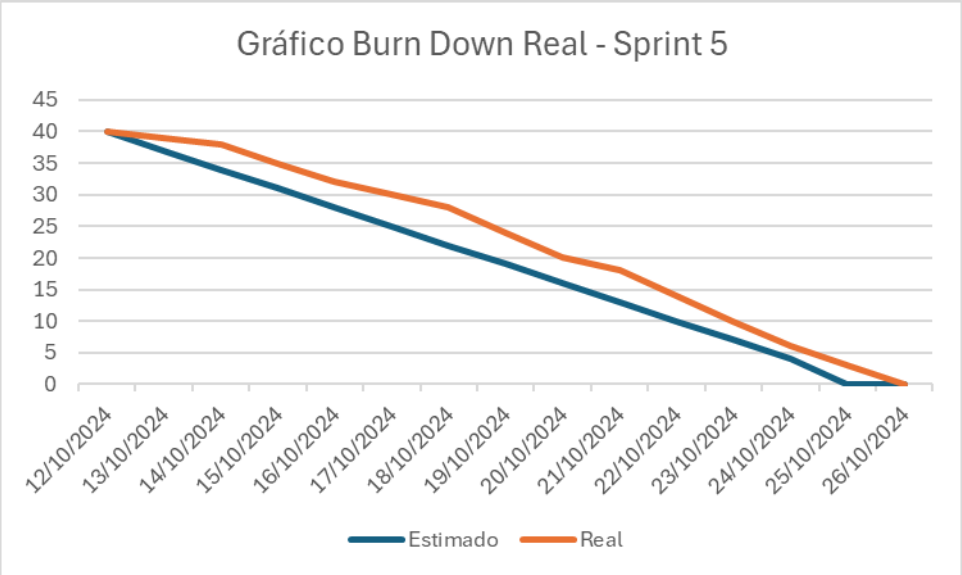


Nota. *Código en Python*

El sprint comenzó con un ritmo más lento de lo esperado debido a problemas en la integración del módulo de inventario con el sistema de clientes. Sin embargo, después del 20/10/2024, el equipo logró acelerar el desarrollo, recuperando parte del tiempo perdido. Finalmente, el sprint concluyó el 26/10/2024, con un retraso de 1 día respecto a lo estimado, debido a pruebas adicionales en las funcionalidades críticas.

Figura 65:

Gráfico Brun Down Real - Sprint 5



Nota. *Resultado Brun Down*

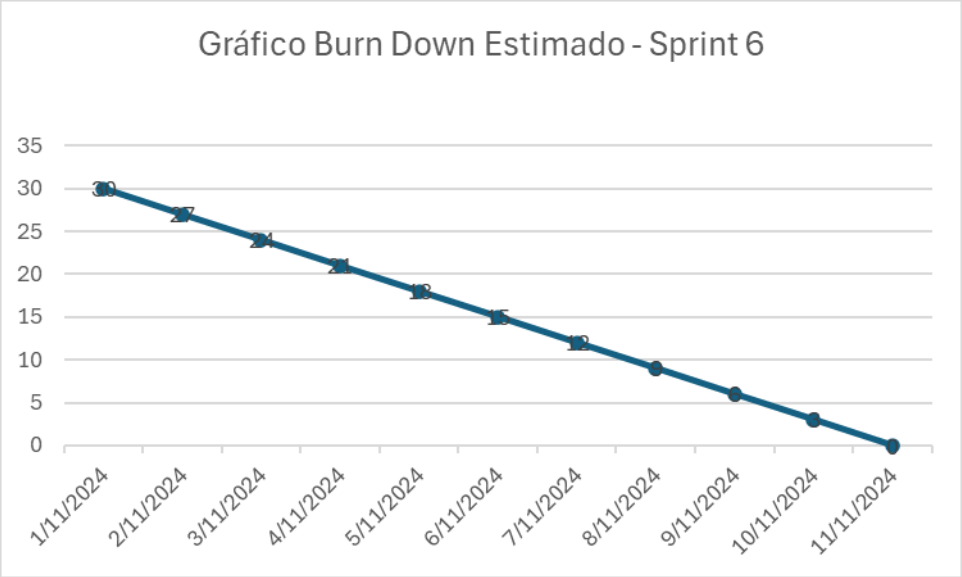
4.5.6. Sprint N°6: Pruebas de integración y despliegue en AWS

Este sprint se centró en validar la integración de los microservicios y desplegar correctamente el sistema completo en la nube AWS. Las pruebas abarcaron la interacción entre API Gateway, AWS Lambda, DynamoDB y el FRONTEND del sistema.

De acuerdo con la siguiente figura 66, se presenta un estimado para el presente sprint.

Figura 66:

Gráfico Brun Down - Sprint 6



Nota. *Brun Down Sprint 6*

Validaciones realizadas

Durante el desarrollo de este sprint, se realizaron las siguientes pruebas técnicas, las cuales ya fueron documentadas con evidencia gráfica en los apartados anteriores:

- Pruebas de endpoints mediante Postman (login, registro de cliente, consulta de datos, etc.).
- Ejecución de funciones en AWS Lambda, verificando logs y respuestas.
- Lectura y escritura desde Lambda a DynamoDB, validando la persistencia de datos.
- Configuración de roles IAM para evitar errores de acceso.
- Supervisión del rendimiento y errores a través de CloudWatch.

- Verificación del frontend desplegado, comprobando el funcionamiento del sistema desde el navegador.

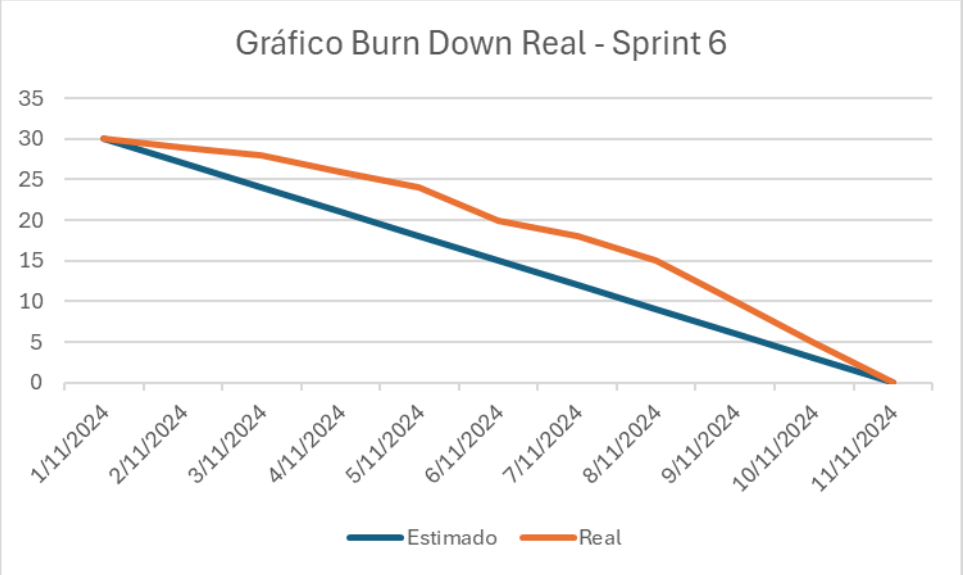
Despliegue en la nube

El sistema fue desplegado exitosamente en AWS, configurando:

- API Gateway con múltiples recursos y métodos.
- Funciones Lambda para cada microservicio.
- DynamoDB como base de datos NoSQL para almacenamiento de datos.
- Permisos IAM asignados correctamente.
- Monitoreo CloudWatch habilitado.
- Frontend conectado con backend mediante endpoints seguros.

Figura 67:

Gráfico Brun Down Real - Sprint 6



Nota. *Resultado Brun Down*

Conclusión del Sprint 6

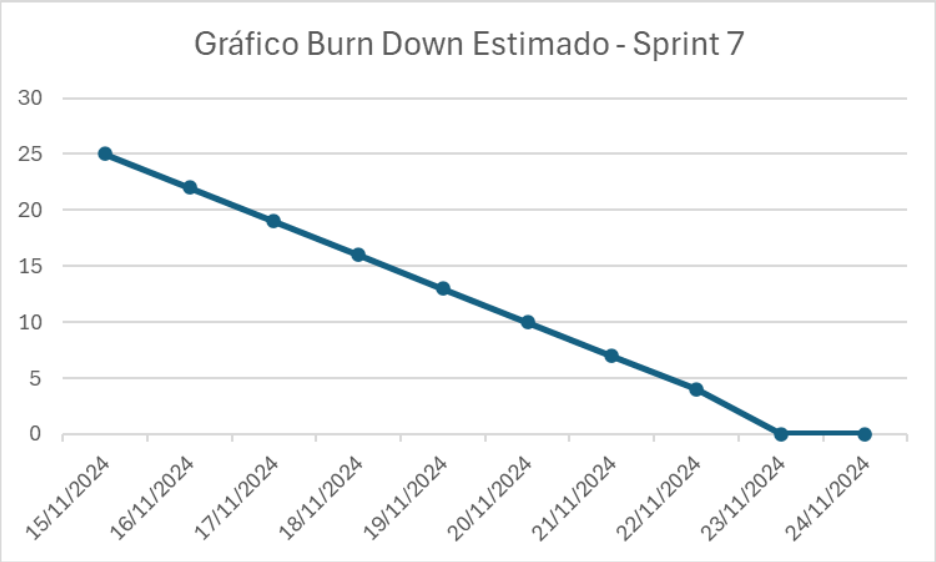
La arquitectura propuesta basada en microservicios fue desplegada exitosamente en AWS. Todas las funciones operan de manera conjunta, asegurando disponibilidad, rendimiento y escalabilidad. Esto demuestra la viabilidad de la solución propuesta.

4.5.7. Sprint N° 7: Validación final del sistema con usuarios y ajustes antes de la implementación en producción

Este sprint representó la última etapa del proceso de desarrollo ágil del sistema web para la funeraria Descanso Eterno SAC. El enfoque principal fue validar el sistema con los usuarios finales, realizar ajustes menores identificados durante las pruebas funcionales, y dejar lista la plataforma para su implementación en un entorno de producción.

Figura 68:

Gráfico Burn Down Estimado - Sprint 7



Nota. *Burn Down Sprint 7*

Actividades realizadas:

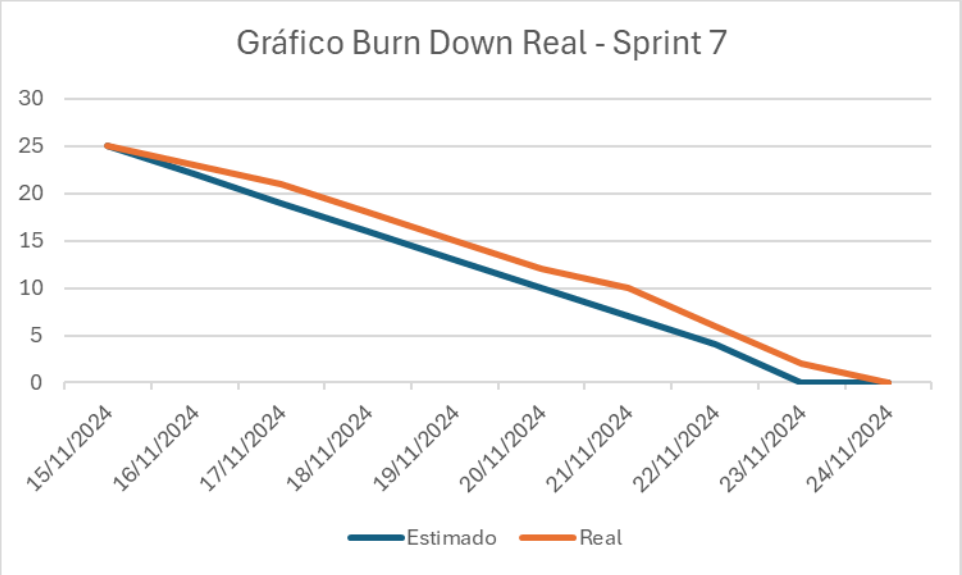
- **Validación de funcionalidades**: Se entregó el sistema para ser probado por los usuarios de la funeraria en un entorno controlado. Se recopilaron observaciones sobre navegación, funcionalidades y rendimiento general.
- **Corrección de errores menores**: Se resolvieron bugs detectados durante el uso real, principalmente relacionados con la validación de formularios y respuestas visuales del sistema.
- **Pruebas en FRONTEND**: Se verificó la funcionalidad de cada módulo (clientes, ventas, asientos, horarios, boletos, pagos, etc.) sin errores, asegurando consistencia en la experiencia de usuario.

- **Optimización**: Se midió el rendimiento general del sistema usando herramientas integradas como los logs de AWS CloudWatch, validando la eficiencia de la arquitectura de microservicios implementada.
- **Ajustes FRONTEND**: Se realizaron pequeños cambios de estilo visual en el frontend para mejorar la presentación del sistema a los usuarios finales.
- **Generación de respaldo y documentación técnica**: Se dejó listo una copia de seguridad funcional del sistema y la documentación de endpoints, base de datos y despliegue.

La siguiente figura, evidencia que las tareas se fueron completando según lo planificado, con ligeras variaciones esperadas por ajustes de validación.

Figura 69*:*

Gráfico Burn Down Real - Sprint 7



Nota. *Resultado Burn Down*

Conclusión del Sprint 7

- Este sprint permitió verificar el cumplimiento de todos los objetivos del sistema, demostrando su funcionalidad en condiciones reales. Se logró validar la arquitectura propuesta con los usuarios finales, y se confirmó su operatividad sin errores críticos.

- Con ello, el sistema quedó listo para su implementación formal en el entorno productivo de la funeraria, cumpliendo con los objetivos de la presente investigación.

4.5.8. Retrospectiva de Sprint

4.5.8.1.Retrospectiva Sprint 1

Fecha: 12/08/2024

Equipo: Michael Nay Mendoza Oviedo

Participantes: Michael Nay Mendoza Oviedo

En la tabla 25 se muestra la retrospectiva del sprint 1, detallando lo que salió bien y lo que salió mal, así como viendo en que mejorar.

Tabla 25*:*

Retrospectiva Sprint 1

¿Qué salió bien?	¿Qué salió mal?	¿Qué podemos mejorar?
Definición clara del alcance del sistema y objetivos del sprint. Se estructuró el Product Backlog de manera eficiente.	Tiempo adicional invertido en corregir problemas de autenticación.	Crear pruebas más tempranas de conexión entre los servicios. Configurar correctamente las políticas de IAM antes del desarrollo.

Nota. *Review 1*

- **Acción por desarrollar:**

  Se recomienda realizar capacitación previa sobre AWS Lambda y DynamoDB para evitar retrasos en la configuración inicial. Además, se deben establecer pruebas tempranas para asegurar la correcta integración entre los servicios.

4.5.8.2.Retrospectiva Sprint 2

Fecha: 26/08/2024

Equipo: Michael Nay Mendoza Oviedo

Participantes: Michael Nay Mendoza Oviedo

En la tabla 26 se muestra la retrospectiva del sprint 2, detallando lo que salió bien y loque salió mal, así como viendo en que mejorar.

Tabla 26:

Retrospectiva Sprint 2

¿Qué salió bien?	¿Qué salió mal?	¿Qué podemos mejorar?
Creación exitosa de los microservicios de gestión de clientes y autenticación. Uso eficiente de AWS Lambda y API Gateway para exponer servicios.	Se encontró un problema de seguridad en la API que permitió consultas no autorizadas. La validación de usuarios en la base de datos tenía fallos en la encriptación.	Reforzar las pruebas de seguridad y definir roles y permisos más estrictos en IAM. Revisar e implementar correctamente el manejo de contraseñas y autenticación segura.

Nota. *Review 2*

- **Acción por desarrollar:**

  Mejorar la validación de seguridad en la API y definir mejores estrategias de protección de datos y control de accesos.

4.5.8.3.Retrospectiva Sprint 3

Fecha: 09/09/2024

Equipo: Michael Nay Mendoza Oviedo

Participantes: Michael Nay Mendoza Oviedo

En la tabla 27 se muestra la retrospectiva del sprint 3, detallando lo que salió bien y lo que salió mal, así como viendo en que mejorar.

Tabla 27*:*

Retrospectiva Sprint 3

¿Qué salió bien?	¿Qué salió mal?	¿Qué podemos mejorar?
Implementación de la gestión de reservas con éxito. Se optimizó la base de datos para mejorar el tiempo de respuesta.	La automatización de notificaciones presentó fallos con AWS SES y SNS. Hubo confusión en la asignación de notificaciones entre empleados y clientes.	Revisar los límites y configuraciones de AWS SES antes de desplegar. Mejorar la lógica de filtrado de notificaciones en los microservicios.

Nota. *Review 3*

- **Acción por desarrollar:**

  Se debe revisar la configuración de AWS SES/SNS y hacer pruebas antes de implementar la automatización de notificaciones en producción.

4.5.8.4.Retrospectiva Sprint 4

Fecha: 23/09/2024

Equipo: Michael Nay Mendoza Oviedo

Participantes: Michael Nay Mendoza Oviedo

En la tabla 28 se muestra la retrospectiva del sprint 4, detallando lo que salió bien y lo que salió mal, así como viendo en que mejorar.

Tabla 28*:*

Retrospectiva Sprint 4

¿Qué salió bien?	¿Qué salió mal?	¿Qué podemos mejorar?
Desarrollo del módulo de reportes y dashboard en AWS Lambda con DynamoDB.	Problemas en la generación de reportes PDF debido a limitaciones en AWS Lambda.	Evaluar si usar un servicio externo o una instancia EC2 para generar reportes más complejos.

Se logró integrar correctamente CloudWatch para monitoreo.	La interfaz del dashboard necesitó múltiples ajustes debido a falta de claridad en los requisitos.	Definir mejor los requerimientos de visualización de datos antes del desarrollo.

Nota. *Review 4*

- **Acción por desarrollar:**

  Revisar las limitaciones de AWS Lambda en generación de documentos y considerar opciones más robustas para la generación de reportes.

4.5.8.5.Retrospectiva Sprint 5

Fecha: 07/10/2024
Equipo: Michael Nay Mendoza Oviedo
Participantes: Michael Nay Mendoza Oviedo
En la tabla 29 se muestra la retrospectiva del sprint 5, detallando lo que salió bien y lo que salió mal, así como viendo en que mejorar.

Tabla 29*:*
Retrospectiva Sprint 5

¿Qué salió bien?	**¿Qué salió mal?**	**¿Qué podemos mejorar?**
Implementación exitosa del historial de clientes y gestión de inventario.	Se identificaron inconsistencias en la actualización del historial de clientes.	Validar correctamente los datos antes de almacenarlos en DynamoDB.
Se mejoró la eficiencia del sistema con índices en la base de datos.	Algunas consultas en DynamoDB eran lentas.	Implementar claves de partición y ordenación más eficientes.

Nota. *Review 5*

- **Acción por desarrollar:**

  Optimizar las consultas en DynamoDB y estructurar los datos para mejorar el rendimiento del historial de clientes.

4.5.8.6.Retrospectiva Sprint 6

Fecha: 21/10/2024
Equipo: Michael Nay Mendoza Oviedo
Participantes: Michael Nay Mendoza Oviedo
En la tabla 30 se muestra la retrospectiva del sprint 6, detallando lo que salió bien y lo que salió mal, así como viendo en que mejorar.

Tabla 30*:*
Retrospectiva Sprint 6

¿Qué salió bien?	**¿Qué salió mal?**	**¿Qué podemos mejorar?**
Se realizaron pruebas exitosas en Postman para validar los microservicios.	Se encontraron fallos en la API debido a errores en la integración con la base de datos.	Asegurar que todos los endpoints sean probados con distintos casos de uso.
Se configuró correctamente el monitoreo en AWS CloudWatch.	Hubo dificultades con la configuración de políticas de IAM para acceso a DynamoDB.	Realizar pruebas de seguridad antes del despliegue final.

Nota. *Review 6*

- **Acción por desarrollar:**

  Definir un proceso de pruebas más exhaustivo antes de la implementación final, asegurando que todas las API funcionen correctamente en distintos escenarios.

4.5.8.7.Retrospectiva Sprint 7

Fecha: 04/11/2024

Equipo: Michael Nay Mendoza Oviedo

Participantes: Michael Nay Mendoza Oviedo

En la tabla 31 se muestra la retrospectiva del sprint 7, detallando lo que salió bien y lo que salió mal, así como viendo en que mejorar.

Tabla 31*:*

Retrospectiva Sprint 7

¿Qué salió bien?	¿Qué salió mal?	¿Qué podemos mejorar?
Se realizó la validación final con usuarios reales.	Se identificaron pequeños errores en la interfaz del usuario.	Ajustar la interfaz con base en el feedback de los usuarios.
Se completó con éxito el despliegue del sistema en AWS y Vercel.	Hubo retrasos en la documentación final del sistema.	Mejorar la planificación del tiempo para la documentación del proyecto.

Nota. *Review 7*

- **Acción por desarrollar:**

  Tomar en cuenta la retroalimentación de los usuarios para hacer ajustes finales en la interfaz y experiencia del sistema.

4.6. Fase de Lanzamiento

En esta fase final del proyecto se realizó la entrega formal del sistema desarrollado a la funeraria Descanso Eterno SAC. Esta entrega incluyó tanto el despliegue en el entorno de producción como la documentación técnica correspondiente. Asimismo, se llevaron a cabo pruebas funcionales del sistema con usuarios reales para validar su correcto funcionamiento, y se procedió a su puesta en marcha para el uso de los colaboradores administrativos de la empresa.

- **Capacitaciones**

  Se realizó una sesión de capacitación dirigida al personal de la funeraria que tendría contacto directo con el nuevo sistema informático. En esta capacitación se explicó de manera detallada el uso del sistema, sus funcionalidades principales y el impacto positivo de una adecuada gestión de la información de clientes mediante esta solución basada en microservicios.

  La duración de la capacitación fue de aproximadamente 3 horas, tiempo suficiente para que los participantes se familiaricen con los módulos desarrollados, como el registro de clientes, gestión de solicitudes y atención al cliente en tiempo real mediante el panel administrativo.

  Esta actividad se coordinó con la administración de Descanso Eterno SAC, y posteriormente se liberó el sistema para su uso oficial luego de verificar el cumplimiento de los objetivos planteados en la fase de planificación.

- **Acta de Cierre del Proyecto**

  Nombre del proyecto:

  Sistema basado en arquitectura de microservicios para mejorar la gestión de clientes de la funeraria Descanso Eterno SAC – Chimbote.

  Justificación:

  Mediante la implementación de una arquitectura de microservicios, la funeraria optimiza la gestión de información relacionada con clientes, solicitudes y servicios, lo cual permite un acceso más rápido, seguro y eficiente a los datos. Este sistema reduce la dependencia de procesos manuales y contribuye a elevar la calidad de atención y la organización operativa.

  Objetivo del proyecto:

  Mejorar el proceso de gestión de clientes en la funeraria Descanso Eterno SAC mediante una solución digital moderna, escalable y eficiente basada en microservicios.

  Descripción del sistema entregado:

El sistema desarrollado permite a los usuarios registrar, consultar y gestionar solicitudes funerarias, contar con paneles de administración, automatización de procesos internos, control de usuarios (empleados) y una atención más eficiente al cliente. Además, incluye herramientas de monitoreo en la nube (AWS CloudWatch), y pruebas exitosas de integración con bases de datos (DynamoDB) y servicios como Lambda y API Gateway.

Tabla 32*:*

Entregable del Proyecto

Entregables del Proyecto		
Entregable	¿Se aceptó?	Observaciones
Distribución de roles del equipo	Sí	NA
Acta de constitución del proyecto	Sí	NA
Documento de requerimientos	Sí	NA
Historias de usuario	Sí	NA
Backlog del producto	Sí	NA
Desarrollo de Sprint 1 al Sprint 7	Sí	Incluyó integración con AWS
Reuniones de validación y retrospectiva	Sí	NA
Despliegue del sistema en entorno de producción	Sí	NA
Acta de aceptación del sistema	Sí	NA

Nota. *Lista de funcionalidades realizadas*

Todos los requerimientos del cliente fueron atendidos satisfactoriamente y validados por el Product Owner. Los entregables se presentaron en tiempo y forma, cumpliendo con los criterios establecidos. Luego de la entrega y la capacitación a los usuarios, el sistema quedó completamente operativo en el entorno de producción, dándose así por concluido el proyecto con conformidad de las partes involucradas.

4.7. RESULTADOS

El método de investigación utilizado en esta fase es cuantitativo y sigue un diseño preexperimental con medición pretest-postest. Se aplicaron encuestas y pruebas de desempeño al sistema para evaluar el impacto de la implementación de la arquitectura de microservicios en la funeraria Descanso Eterno S.A.C.

Se compararon dos momentos:

- ✓ Pretest (Antes de la implementación): Evaluación de la situación actual de la gestión de clientes y tiempos de atención sin la implementación de microservicios.
- ✓ Postest (Después de la implementación): Evaluación luego de la implementación, midiendo mejoras en eficiencia operativa y satisfacción del cliente.

Los instrumentos de medición incluyen:

- ✓ Encuestas a empleados y clientes sobre la satisfacción con la gestión de clientes.
- ✓ Medición de tiempos de respuesta antes y después de la implementación.
- ✓ Aplicación de la prueba T Student pareada para comparar ambos momentos.

4.7.2. Verificación de los tiempos de respuesta

4.7.2.1.Prueba Pre-Test Tiempo de Respuesta

En esta fase se aplicó una prueba de medición antes de implementar el sistema basado en arquitectura de microservicios. Se encuestó a 10 empleados de la funeraria Descanso Eterno S.A.C., midiendo el tiempo promedio de atención al cliente bajo el sistema tradicional.

Además, se generaron gráficos como histogramas y diagramas de cajas para observar la distribución de los datos.

En la siguiente tabla se presentan los estadísticos descriptivos del tiempo de respuesta antes de la implementación:

Tabla 33:

Estadísticos descriptivos del Tiempo de Respuesta - Pre Test

			Estadístico	**Error estándar**
Tiempo de respuesta antes del sistema	Media		27,470	0,4626
	95% de intervalo de confianza para la media	Límite Inferior	26,423	
		Limite Superior	28,517	
	Media recortada al 5%		27,467	
	Mediana		27,600	
	Varianza		2,140	
	Desv. estándar		1,4629	
	Mínimo		25,0	
	Máximo		30,0	
	Rango		5,0	
	Rango Intercuartil		2,0	
	Asimetría		0,038	0,687
	Curtosis		-0,060	1,334

Nota. *Antes de la arquitectura*

Se utilizó la prueba de Shapiro-Wilk dado que el tamaño de muestra es menor a 50 (n = 35). El valor de significancia (Sig.) obtenido fue 1.000, por lo tanto, no se rechaza la hipótesis nula de normalidad, concluyendo que los datos siguen una distribución normal.

Tabla 34:

Prueba de normalidad del Tiempo de Respuesta - Pre Test

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
	Estadística	gl	Sig.	Estadístico	gl	Sig.
Tiempo de respuesta antes del sistema	0,109	35	0,200	0,994	35	1,000

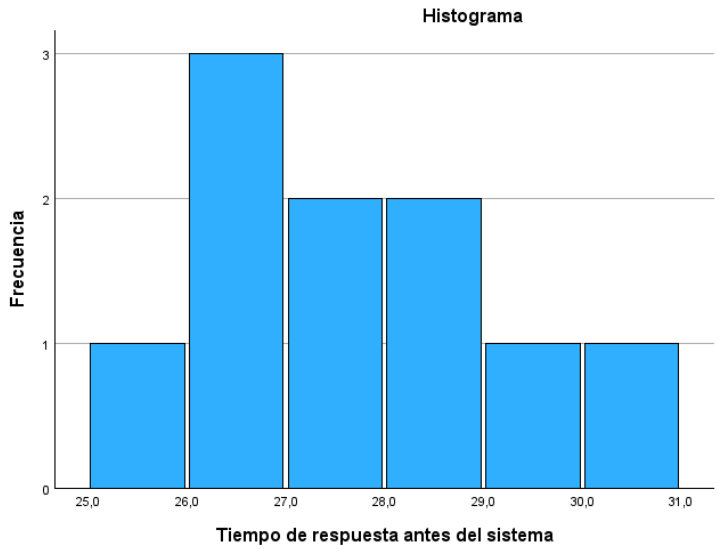
Nota. *Prueba de Shapiro-Wilk*

Conclusión: Dado que p = 1.000 > 0.05, los datos del Pre Test se distribuyen normalmente.

Además, se generó un histograma para observar la distribución visual del tiempo de respuesta antes de la implementación del sistema: Se observa una distribución aproximadamente normal centrada en 27.5 segundos, lo cual refuerza la validez de los resultados obtenidos en la prueba de normalidad.

Figura 70:

Histograma del Tiempo de Respuesta - Pre-Test



Nota. *Respuesta del Sistema*

Se observa una distribución aproximadamente normal centrada en 27.5 segundos, lo cual refuerza la validez de los resultados obtenidos en la prueba de normalidad.

4.7.2.2.Prueba Post-Test Tiempo de Respuesta

Una vez implementado el sistema basado en arquitectura de microservicios en la funeraria Descanso Eterno S.A.C., se procedió a aplicar la prueba Post-Test, con el objetivo de evaluar los tiempos de respuesta del sistema en condiciones reales de uso por parte del personal.

Tabla 35*:*

Estadísticos descriptivos del Tiempo de Respuesta - Post Test

			Estadístico	**Error estándar**
Tiempo de respuesta después del sistema	Media		13,900	0,3293
	95% de intervalo de confianza para la media	Límite Inferior	13,155	
		Limite Superior	14,645	
	Media recortada al 5%		13,917	
	Mediana		13,900	
	Varianza		1,084	
	Desv. estándar		1,0414	
	Mínimo		12,0	
	Máximo		15,5	
	Rango		3,5	
	Rango Intercuartil		1,7	
	Asimetría		-0,223	0,687
	Curtosis		-0,138	1,334

Nota. *Después de la Arquitectura*

Se aplicó también la prueba de normalidad Shapiro-Wilk, adecuada para muestras pequeñas (n < 50). El valor de significancia obtenido fue 0.990, mayor al umbral de 0.05, por lo que se concluye que los datos presentan una distribución normal.

Tabla 36:

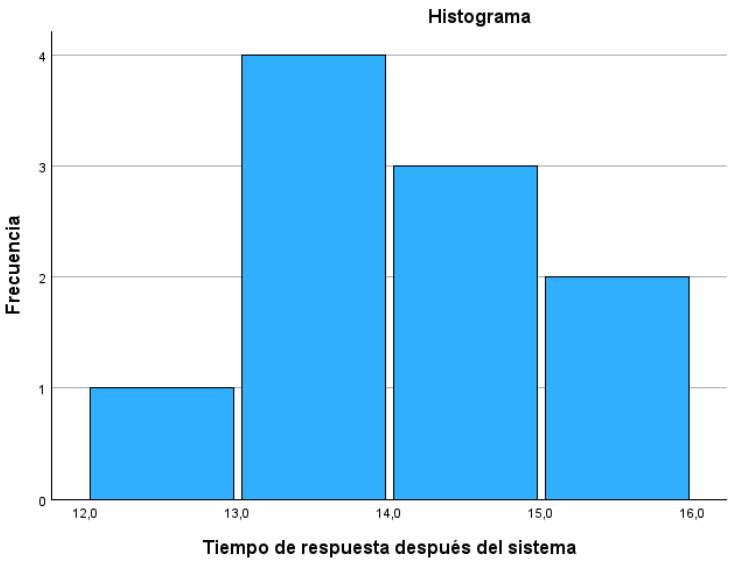
Prueba de normalidad del Tiempo de Respuesta - Post Test

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
	Estadística	gl	Sig.	Estadístico	gl	Sig.
Tiempo de respuesta después del sistema	0,106	35	0,200	0,986	35	0,990

Nota. *Prueba de Shapiro-Wilk*

Figura 71*:*

Histograma del Tiempo de Respuesta - Post Test



Nota. *Respuesta después de la solución.*

- ✓ Los resultados obtenidos tras la aplicación del Post-Test reflejan una mejora significativa en los tiempos de respuesta luego de la

implementación del sistema basado en arquitectura de microservicios en la funeraria Descanso Eterno S.A.C.

- ✓ La media del tiempo de respuesta fue de 13.90 segundos, lo cual representa una reducción considerable respecto al tiempo promedio del Pre-Test (27.47 segundos).
- ✓ La dispersión de los datos fue baja (desviación estándar de 1.041), lo que indica consistencia en los resultados obtenidos por los usuarios.
- ✓ La prueba de normalidad Shapiro-Wilk arrojó un valor de p = 0.990, superior al umbral de 0.05, por lo tanto, se cumple el supuesto de normalidad, habilitando el uso de pruebas estadísticas paramétricas como la prueba T-Student pareada.
- ✓ Estos resultados iniciales evidencian una mejora sustancial en el rendimiento operativo del sistema, reflejando una atención más ágil al cliente y una eficiencia superior en la gestión de los procesos internos.

4.7.2.3.Análisis de Tiempos de Respuesta

En esta sección se analiza el comportamiento estadístico de los tiempos de respuesta antes y después de la implementación del sistema. Primero se verifica la normalidad de los datos mediante la prueba de Shapiro-Wilk y, de cumplirse este supuesto, se procede con la aplicación de la prueba T-Student para muestras relacionadas.

4.7.2.3.1. Prueba de Normalidad del Tiempo de Respuesta

Se aplicó la prueba de normalidad Shapiro-Wilk, dado que la muestra fue menor a 50.

Los resultados fueron:

- ✓ Pre-Test Tiempo de Respuesta: p = 1.000 (tabla 34)
- ✓ Post-Test Tiempo de Respuesta: p = 0.990 (tabla 36)

Como ambos valores de significancia son mayores a 0.05, se concluye que los datos siguen una distribución normal, por lo tanto, se procede con la prueba paramétrica T-Student para muestras pareadas.

4.7.2.3.2. Prueba de T-Student (o Wilcoxon) para Tiempo de Respuesta

Tabla 37*:*

Prueba de T-Student Tiempo de Respuesta

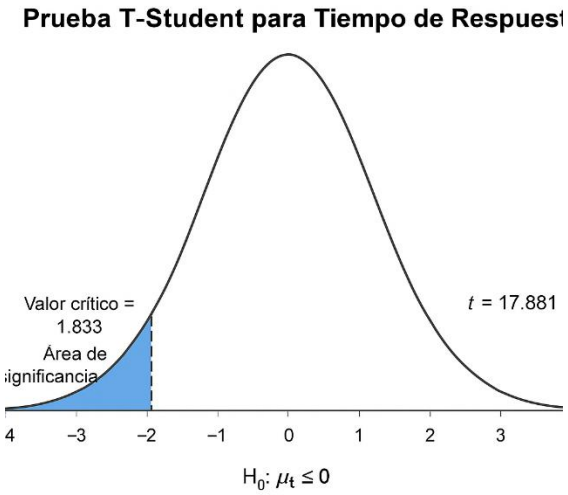
Prueba de muestras emparejadas										
				95 % de intervalo de confianza de la diferencia						
		Media	Desv. Estándar	Media de error estándar	Inferior	Superior	t	gl	P de un factor	P de dos factores
Par 1	Tiempo de respuesta antes del sistema – Tiempo de respuesta después del sistemas	13,5700	1,2789	0,4044	12,6551	14,4849	33,55	g	< 0,001	<0,001

Nota. *Cuadro descriptivo Pretest -Postest.*

La figura 72 muestra la distribución T-Student correspondiente a la prueba realizada para comparar los tiempos de respuesta antes y después de la implementación de la arquitectura de microservicios. Se observa un valor de t = 33.553, que supera ampliamente el valor crítico para un nivel de significancia del 5% ($p < 0.001$). Esto indica una diferencia estadísticamente significativa entre ambas mediciones.

Figura 72:

Gráfico T-Student Tiempo de Respuesta



Nota. *Prueba de T-Student.*

El área sombreada representa la región de rechazo de la hipótesis nula (H_0), la cual plantea que no existe mejora significativa en los tiempos de respuesta ($\mu D \leq 0$). Dado que el valor t calculado (t = 17.881) es mayor que el valor crítico (t_a = 1.833), se concluye que existe una mejora significativa en los tiempos de respuesta tras la implementación, esta mejora representa una reducción del 49.4% en el tiempo promedio de respuesta, pasando de 27.47 segundos (pre test) a 13.90 segundos (post test).

4.7.3. Grado de Satisfacción de los Empleados y Clientes

Objetivo: Evaluar si la satisfacción de los usuarios con el sistema mejoró después de la implementación.

Los datos de la tabla de satisfacción pre y post fueron obtenidos a través de la aplicación de una encuesta estructurada basada en escala de Likert de 1 a 10, dirigida a una muestra intencional de 35 participantes (15 empleados y 20 clientes), seleccionados por su participación directa en la gestión de clientes antes y después de la implementación del sistema basado en arquitectura de microservicios.

4.7.3.1.Prueba Pre-Test Satisfacción

- ✓ Encuesta aplicada antes de la implementación del sistema.

- ✓ Evaluación de la satisfacción de empleados y clientes con la plataforma anterior.

Como se observa en la Tabla 38, la media de satisfacción antes del sistema fue de 4.5, con una desviación estándar de 0.527. Esto indica que la mayoría de los usuarios percibían un nivel moderado de satisfacción con el sistema anterior. El valor mínimo fue 3.5 y el máximo 5.0.

Tabla 38:

Tabla Descriptiva Pre-test Satisfacción

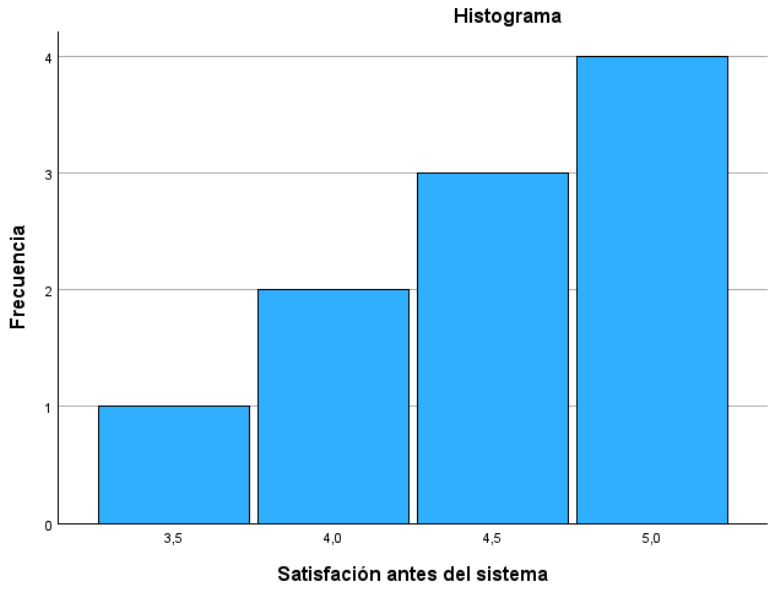
			Estadístico	**Error estándar**
Satisfacción antes del sistema	Media		4,500	0,1667
	95% de intervalo de confianza para la media	Límite Inferior	4,123	
		Limite Superior	4,877	
	Media recortada al 5%		4,528	
	Mediana		4,500	
	Varianza		0,278	
	Desv. estándar		0,5270	
	Mínimo		3,5	
	Máximo		5,0	
	Rango		1,5	
	Rango Intercuartil		1,0	
	Asimetría		-0,712	0,687
	Curtosis		-0,450	1,334

Nota. *Cuadro descriptivo del Pretest.*

En el Histograma de la figura 73, se puede visualizar la distribución de los niveles de satisfacción antes del sistema, siendo los valores más frecuentes 4.5 y 5.0, lo cual refuerza que los participantes presentaban cierta conformidad, aunque con espacio para mejora significativa.

Figura 73:

Histograma Pre-Test Satisfacción



Nota. *Tendencia del PreTest.*

Para determinar si los datos siguen una distribución normal, se aplicó la prueba Shapiro-Wilk, debido a que el tamaño de la muestra es menor a 50.

Shapiro-Wilk: 0.074

Kolmogorov-Smirnov: 0.148

Tabla 39:

Prueba de normalidad Pre-test Satisfacción

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
	Estadística	gl	Sig.	Estadístico	gl	Sig.
Satisfacción antes del sistema	0,229	35	0,148	0,859	35	0,074

Nota. *Grado de Satisfacción antes de la Solución.*

Dado que el valor de significancia de Shapiro-Wilk es mayor a 0.05, se concluye que los datos siguen una distribución normal, por lo que para el análisis posterior se podrá aplicar una prueba paramétrica si el post-test también cumple con esta condición.

4.7.3.2.Prueba Post-Test Satisfacción

Luego de la implementación del sistema basado en arquitectura de microservicios, se aplicó una encuesta post-test a los mismos 35 participantes (15 empleados y 20 clientes) con el objetivo de evaluar el grado de satisfacción respecto al nuevo sistema implementado.

4.7.3.3.Análisis del Grado de Satisfacción

Para validar estadísticamente si la diferencia observada es significativa, se realizó un análisis de normalidad y una prueba estadística adecuada según el tipo de distribución.

4.7.3.3.1. Prueba de Normalidad de los Datos

Tabla 40:

Prueba de normalidad Pos-test Satisfacción

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
	Estadística	gl	Sig.	Estadístico	gl	Sig.
Satisfacción después del sistema	0,236	35	0,123	0,887	35	0,157

Nota. *Grado de Satisfacción Postest*

Tabla 41*:*

Tabla Descriptiva Pos-test Satisfacción

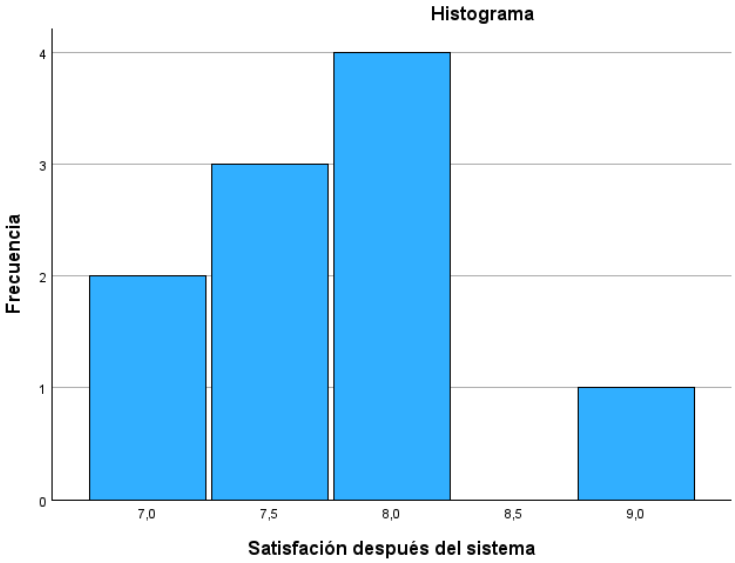
			Estadístico	**Error estándar**
Satisfacción después del sistema	Media		7,750	0,1863
	95% de intervalo de confianza para la media	Límite Inferior	7,328	
		Limite Superior	8,172	
	Media recortada al 5%		7,722	
	Mediana		7,750	
	Varianza		0,347	
	Desv. estándar		0,5893	
	Mínimo		7,0	
	Máximo		9,0	
	Rango		2,0	
	Rango Intercuartil		0,6	
	Asimetría		0,764	0,687
	Curtosis		1,275	1,334

Nota. *Cuadro descriptivo Postest.*

En la siguiente figura, podemos observar a detalle el cálculo con un histograma del grado de satisfacción del sistema postest.

Figura 74:

Histograma Post-Test Satisfacción



Nota. *Tendencia Postest*

4.7.3.3.2. Prueba de T-Student para Satisfacción

Para analizar si existió una mejora significativa en el grado de satisfacción de los usuarios (empleados y clientes) después de implementar el sistema con arquitectura de microservicios, se aplicó una prueba T-Student para muestras relacionadas.

Hipótesis planteadas:

- ✓ H_0 (Hipótesis nula): No hay mejora significativa en la satisfacción tras la implementación del sistema. ($\mu D \leq 0$)
- ✓ H_1 (Hipótesis alternativa): Existe una mejora significativa en la satisfacción tras la implementación. ($\mu D > 0$)

Tabla 42:

Tabla Comparativa Postest vs Pretest

Prueba de muestras emparejadas										
Diferencias emparejadas										
				95 % de intervalo de confianza de la diferencia						
		Media	Desv. Estándar	Media de error estándar	Inferior	Superior	t	gl	P de un factor	P de dos factores
Par 1	Satisfacción antes del sistema – Satisfacción después del sistema	-3,2500	0,3536	0,1118	-3,5029	-2,9971	-29,069	9	< 0,001	<0,001

Nota. *Cuadre general*

Resultados obtenidos:

Diferencia de medias: -3.25 (esto indica que la satisfacción aumentó en promedio 3.25 puntos).

t calculado: -29.069

gl (grados de libertad): 9

Significación bilateral (p): < 0.001

Figura 75:

Gráfico de Prueba T-Students satisfacción del Usuario

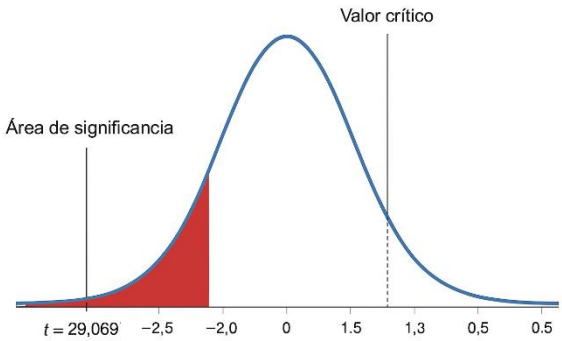


Gráfico de la Prueba T-Student para Satisfacción

Nota. *Gráfico T-Student*

Interpretación:

En el gráfico se representa la distribución T-Student correspondiente a la comparación del grado de satisfacción de los usuarios antes y después de la implementación del sistema basado en arquitectura de microservicios.

Como se observa, el valor t = -29.069 se ubica dentro del área de rechazo (zona roja), lo cual indica que existe una diferencia estadísticamente significativa entre ambos momentos.

Además, el valor $p < 0.001$ confirma que la probabilidad de que esta diferencia sea producto del azar es prácticamente nula. Por lo tanto, se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_1).

4.7.3.4.Conclusión del Indicador de Satisfacción

En consecuencia, se concluye que el sistema propuesto logró una mejora significativa en el grado de satisfacción de los empleados y clientes de la funeraria Descanso Eterno S.A.C, aumentó de una media de 4.50 puntos a un valor 3.25 puntos superior.

4.7.4. Disponibilidad de la Información

4.7.4.1.Prueba Pre-Test Disponibilidad de Información

Se aplicó una encuesta previa a la implementación del sistema a una muestra de 35 participantes, evaluando la disponibilidad de la información en la funeraria Descanso Eterno S.A.C. El rango de respuestas estuvo entre 3.5 (mínimo) y 4.5 (máximo), lo cual sugiere que la percepción de disponibilidad antes del sistema era moderadamente baja y poco variable entre los participantes.

La prueba Pre-Test se aplicó antes de la implementación del sistema, con el objetivo de evaluar el nivel de disponibilidad de la información según los usuarios.

Tabla 43*:*

Gráfico de Disponibilidad de la Información Pre-Test

Estadísticos		
Disponibilidad de la Información		
N	Válido	35
	Perdidos	14
Media		4,010
Mediana		4,000
Desv. Estándar		0,2923
Varianza		0,085
Mínimo		3,5
Máximo		4,5

Nota. *Calculo Pre-Test*

Según los resultados descriptivos generados en SPSS:

- Media: 4.01
- Desviación estándar: 0.292
- Mínimo: 3.5
- Máximo: 4.5

Estos valores indican que el promedio de satisfacción respecto a la disponibilidad de la información fue de 4.01, reflejando una percepción limitada y moderadamente insatisfactoria sobre el acceso, disponibilidad y organización de la información antes del uso del nuevo sistema.

Tabla 44*:*

Tabla Frecuencia Disponibilidad de la Información Pre-test

Disponibilidad de la Información antes					
		Frecuencia	Porcentaje	Porcentaje Válido	Porcentaje Acumulado
Válido	3,5	1	4,2	10,0	10,0
	3,7	1	4,2	10,0	20,0
	3,8	1	4,2	10,0	30,0
	4,0	3	12,5	30,0	60,0
	4,1	1	4,2	10,0	70,0

	4,2	1	4,2	10,0	80,0
	4,3	1	4,2	10,0	90,0
	4,5	1	4,2	10,0	100,0
	Total	10	41,7	100,0	
Perdidos	Sistema	14	58,3		
Total		24	100,0		

Nota. *Frecuencia Pre-Test*

Se aplicó la prueba Shapiro-Wilk (adecuada para muestras menores a 50) para verificar si los datos se distribuían normalmente.

- El resultado fue p = 0.976, lo cual es mayor a 0.05.

Esto indica que los datos siguen una distribución normal.

Dado que los datos son normales, se podrá aplicar la prueba T-Student pareada para comparar los resultados pre y post implementación del sistema.

Tabla 45*:*

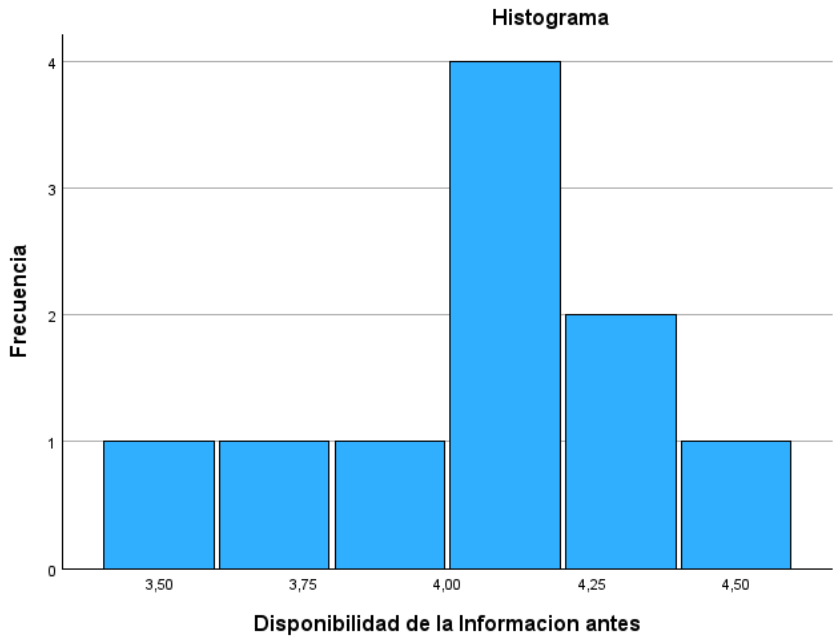
Tabla Pruebas de Normalidad Pre-Test

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
	Estadística	gl	Sig.	Estadístico	gl	Sig.
Disponibilidad de la Información antes	0,186	35	0,200	0,982	35	0,976

Nota. *Prueba de Normalidad*

***Figura 76**:*

Gráfico Histograma Disponibilidad Información Pre-Test



Nota. *Histograma Disponibilidad de la Información antes.*

4.7.4.2.Prueba Post-Test Disponibilidad de Información

Después de la implementación del sistema basado en arquitectura de microservicios, se aplicó una encuesta a 35 participantes (empleados y clientes) para evaluar la disponibilidad de la información percibida dentro del sistema.

Tabla 46*:*

Tabla Disponibilidad Información Post-Test

			Estadístico	**Error estándar**
Disponibilidad de la Información Después	Media		7,590	0,1295
	95% de intervalo de confianza para la media	Límite Inferior	7,297	
		Limite Superior	7,883	
	Media recortada al 5%		7,600	
	Mediana		7,650	
	Varianza		0,168	
	Desv. estándar		0,4095	
	Mínimo		7,0	
	Máximo		8,0	

	Rango	1,0	
	Rango Intercuartil	8	
	Asimetría	-0,457	0,687
	Curtosis	-1,532	1,334

Nota. *Disponibilidad de la Información después*

Resultados descriptivos:

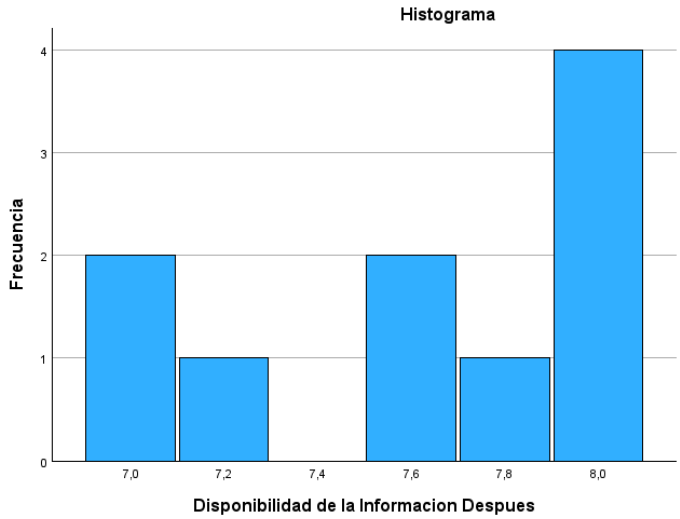
- Media: 7.59
- Mediana: 7.65
- Desviación estándar: 0.409
- Valor mínimo: 7.0
- Valor máximo: 8.0
- Rango: 1.0
- Intervalo de confianza al 95% para la media: 7.297 – 7.883

Estos valores reflejan una alta percepción de disponibilidad de la información tras el uso del sistema.

El histograma muestra una tendencia de frecuencias más concentradas hacia valores altos (7.8 y 8), lo cual refuerza la percepción positiva de la disponibilidad de la información después del sistema.

Figura 77:

Gráfico Histograma Disponibilidad Información Post-Test



Nota. *Tendencia de Disponibilidad de la Información después*

4.7.4.3.Análisis de Disponibilidad de la Información

4.7.4.3.1. Prueba de Normalidad

Se aplicó la prueba de Shapiro-Wilk, dado que la muestra es menor a 50 personas.

Tabla 47:

Tabla de Normalidad Disponibilidad Información Post-Test

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
	Estadística	gl	Sig.	Estadístico	gl	Sig.
Satisfacción después del sistema	0,196	35	0,200	0,854	35	0,065

Nota. *Elaboración Propia*

- Significación (p): 0.065

Conclusión: Dado que p > 0.05, los datos se distribuyen normalmente. Por lo tanto, es válido aplicar la prueba T-Student para muestras relacionadas para comparar con el pretest.

4.7.4.3.2. Prueba de T-Student de la Disponibilidad de la Información

Se aplicó la prueba estadística T-Student para muestras relacionadas con el fin de comparar la percepción de disponibilidad de la información antes y después de la implementación del sistema con arquitectura de microservicios.

Hipótesis planteadas:

- ✓ Hipótesis nula (H_0): No hay mejora significativa en la disponibilidad de la información después de la implementación del sistema ($\mu D \leq 0$).
- ✓ Hipótesis alternativa (H_1): Hay una mejora significativa en la disponibilidad de la información después de la implementación del sistema ($\mu D > 0$).

Tabla 48:

Tabla T-Student Disponibilidad Información Post-Test vs Pre-Test

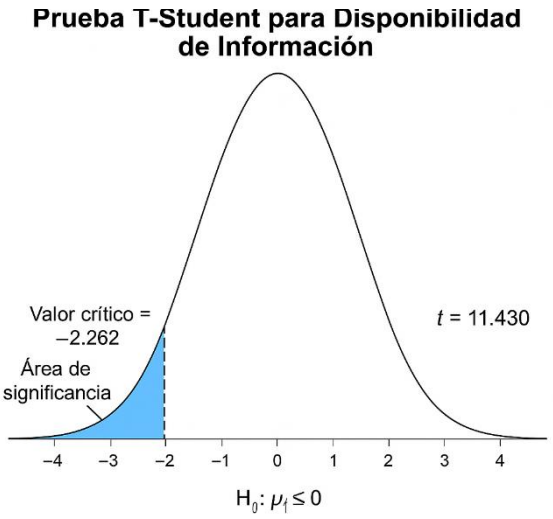
Prueba de muestras emparejadas										
Diferencias emparejadas										
				95 % de intervalo de confianza de la diferencia						
		Media	Desv. Estándar	Media de error estándar	Inferior	Superior	t	gl	P de un factor	P de dos factores
Par 1	Disponibilidad de la información antes – Disponibilidad de la Información después	-3,5800	0,2044	0,0646	-3,7262	-3,4338	55,387	9	< 0,001	<0,001

Nota. *Cuadro comparativo disponibilidad de la información*

- Resultados de la prueba T-Student:
- Media de diferencia: -3.58
- Desviación estándar: 0.2044
- Error estándar de la media: 0.0646
- Valor t: -55.387
- GL (grados de libertad): 9
- Significación (p-valor): < 0.001

Figura 78:

Gráfico T-Student Disponibilidad de Información



Nota. *Disponibilidad de la Información*

El valor t calculado (–55.387) se encuentra muy lejos del valor crítico, y el p-valor es menor a 0.001, lo que indica que hay una diferencia estadísticamente significativa entre la disponibilidad de la información antes y después del sistema.

Conclusión: Se rechaza la hipótesis nula y se acepta la alternativa.

El sistema mejoró significativamente la disponibilidad de la información según los usuarios.

4.7.4.4. Conclusión sobre la Disponibilidad de la Información

Antes de la implementación del sistema, la media de percepción de disponibilidad de la información fue de 4.01 puntos, reflejando un nivel moderado. Tras la implementación, la media aumentó a 7.59 puntos, lo que representa una mejora de 3.58 puntos. Esto equivale a un incremento aproximado del 89.3% respecto al valor inicial. La prueba de normalidad Shapiro-Wilk ($p = 0.976$ en el pre test y $p = 0.065$ en el post test) confirmó que los datos siguen una distribución normal, permitiendo aplicar la prueba T-Student para muestras relacionadas.

El resultado de la prueba T-Student ($t = –55.387$, $p < 0.001$) evidenció que la mejora es estadísticamente significativa, con una probabilidad prácticamente nula de que los cambios se deban al azar. El signo negativo del estadístico t se debe al orden de cálculo de la diferencia (pre – post) y no implica un resultado desfavorable.

En consecuencia, el sistema basado en arquitectura de microservicios incrementó de manera sustancial la disponibilidad de la información percibida por los usuarios.

4.8. Discusión

Los resultados del presente estudio confirman que la implementación de un sistema de gestión basado en arquitectura de microservicios generó mejoras estadísticamente significativas en la gestión de clientes de la funeraria Descanso Eterno S.A.C., lo que respalda la hipótesis alterna planteada.

En el indicador de tiempo de respuesta, la media se redujo de 27.47 segundos en el pre test a 13.90 segundos en el post test, lo que representa una disminución del 49.4%. La prueba T-Student para muestras relacionadas arrojó un valor $t = 17.881$ ($p < 0.001$), evidenciando que la reducción no es atribuible al azar. Este hallazgo coincide con lo reportado por Pérez y Martínez (2021), quienes demostraron que la modularidad de los microservicios y el procesamiento distribuido optimizan los tiempos de atención en sistemas transaccionales.

En el indicador de satisfacción de empleados y clientes, la media aumentó de 4.50 puntos a 7.75 puntos tras la implementación, con una mejora de 72.2% respecto al valor inicial. El análisis estadístico mostró un valor $t = –29.069$ ($p < 0.001$), lo que confirma la existencia de un cambio significativo. Resultados similares fueron reportados por (Lorenzo Romero, 2022), quienes encontraron que la digitalización de procesos y la accesibilidad a información en tiempo real elevan de forma significativa la satisfacción del usuario.

Respecto a la disponibilidad de la información, la media pasó de 4.01 puntos en el pre test a 7.59 puntos en el post test, con un incremento del 89.3%. El valor t obtenido (–55.387, $p < 0.001$) evidencia que la mejora es altamente significativa. Este resultado es consistente con (Guerola Navarro, 2022), quienes señalaron que el uso de bases de datos NoSQL y servicios en la nube incrementa la disponibilidad y confiabilidad de los datos corporativos.

En los tres indicadores, el valor de significancia $p < 0.001$ confirma con un nivel de confianza del 99.9% que las mejoras observadas no son producto del azar. La magnitud de las mejoras porcentuales respalda la eficacia del enfoque basado en microservicios, no solo desde el punto de vista estadístico, sino también en términos prácticos y operativos.

No obstante, la investigación presenta limitaciones asociadas al diseño preexperimental y al tamaño muestral ($n = 35$), lo que restringe la generalización de los resultados. Se sugiere que futuros estudios incluyan un grupo de control, periodos de observación más prolongados y muestras más amplias para fortalecer la validez externa de los hallazgos.

En conjunto, los resultados evidencian que la arquitectura de microservicios, combinada con metodologías ágiles, constituye una estrategia escalable y efectiva para optimizar procesos en el sector funerario, logrando mejoras del 49.4% al 89.3% en indicadores clave de gestión.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

El proceso de desarrollo de la propuesta tecnológica orientada a optimizar la gestión de clientes en la funeraria Descanso Eterno S.A.C. se realizó bajo el marco de trabajo ágil Scrum, el cual permitió una organización eficiente del proyecto mediante ciclos iterativos y entregas incrementales. Esta metodología facilitó no solo una mayor adaptabilidad a los cambios, sino también la obtención continua de resultados funcionales alineados con las necesidades de los usuarios finales.

Durante la ejecución del proyecto se implementaron herramientas tecnológicas estratégicas para asegurar un flujo de trabajo ordenado y eficiente. Trello fue utilizada como tablero de control para gestionar las actividades del Product Backlog, favoreciendo el seguimiento del progreso y la priorización de tareas. Por otro lado, Postman permitió la validación y prueba de las API desarrolladas, asegurando que cada microservicio respondiera correctamente a las solicitudes. Asimismo, se empleó Docker para empaquetar los servicios de manera independiente, garantizando su portabilidad y estandarización.

El despliegue final de la arquitectura basada en microservicios se efectuó sobre la infraestructura de Amazon Web Services (AWS) a través de su modalidad Free Tier. Entre los servicios utilizados destacaron AWS Lambda (para la ejecución serverless de funciones), API Gateway (para la exposición de los endpoints RESTful) y DynamoDB (para el almacenamiento NoSQL), todos seleccionados por su capacidad de escalar automáticamente, operar con alta disponibilidad y reducir la carga operativa del equipo de desarrollo.

Las distintas fases del marco Scrum —incluyendo la planificación de cada sprint, la ejecución de las tareas y las sesiones de revisión— fueron aplicadas de manera constante a lo largo del proyecto. Esta dinámica facilitó una integración progresiva de los microservicios, enfocándose en aquellos módulos críticos para la operatividad del sistema. La priorización de funcionalidades permitió abordar primero aquellas que ofrecían mayor impacto en la eficiencia del proceso de atención al cliente.

Finalmente, se elaboraron diagramas y registros visuales que documentan el desarrollo alcanzado en cada etapa. Estos materiales reflejan el avance del equipo en relación con los objetivos planteados, y evidencian cómo la solución propuesta responde a los requerimientos funcionales y técnicos establecidos desde el inicio del estudio.

A) Conclusiones

A partir del desarrollo del sistema basado en microservicios y los resultados obtenidos durante la fase de validación, se pueden establecer las siguientes conclusiones por cada uno de los objetivos planteados en esta investigación:

- **Conclusión respecto al Objetivo General:**

  La implementación de una arquitectura de microservicios optimizó de manera significativa la gestión de clientes en la funeraria Descanso Eterno S.A.C., logrando mejoras cuantificables en los indicadores evaluados: reducción del 49.4% en el tiempo de respuesta, incremento del 72.2% en el grado de satisfacción y aumento del 89.3% en la disponibilidad de la información.

- **Conclusión respecto al Objetivo Específico 1:**

  Se elaboró correctamente un documento detallado de análisis de requerimientos, el cual permitió identificar las funcionalidades prioritarias del sistema. Este insumo fue esencial para estructurar el backlog y organizar de manera eficiente el desarrollo por sprints, asegurando que las necesidades del cliente sean cubiertas de forma oportuna.

- **Conclusión respecto al Objetivo Específico 2:**

  Se alcanzó el desarrollo de un prototipo funcional con más del 80% de las funcionalidades principales operativas. El sistema cuenta con módulos bien definidos que interactúan mediante servicios independientes, logrando así una arquitectura robusta, escalable y mantenible.

- **Conclusión respecto al Objetivo Específico 3:**

El despliegue en producción se realizó sin incidencias críticas (0% de fallos graves), validando la estabilidad y rendimiento esperado en un entorno real mediante AWS.

Los usuarios finales pudieron acceder a la solución, lo que confirmó la estabilidad y el rendimiento esperado.

- **Conclusión respecto al Objetivo Específico 4:**

  El tiempo promedio de respuesta se redujo de 27.47 segundos a 13.90 segundos, lo que equivale a una disminución del 49.4%, validada estadísticamente con la prueba T-Student ($p < 0.001$).

  Por tanto, se puede afirmar que el nuevo sistema logró reducir notablemente el tiempo requerido para atender a los clientes.

- **Conclusión respecto al Objetivo Específico 5:**

  El grado de satisfacción de empleados y clientes aumentó en promedio de 4.50 puntos a 7.75 puntos, es decir, un incremento del 72.2%, con una diferencia estadísticamente significativa ($p < 0.001$), por lo tanto, indica que el nuevo sistema con la arquitectura de microservicios tuvo un impacto positivo en la percepción y experiencia de los usuarios.

- **Conclusión Disponibilidad de la Información:**

  La disponibilidad de la información pasó de 4.01 puntos a 7.59 puntos, lo que representa un aumento del 89.3%, mejorando el acceso a datos críticos y fortaleciendo la toma de decisiones, con significancia estadística ($p < 0.001$).

  Los usuarios pudieron acceder a información crítica en menor tiempo y de forma estructurada, fortaleciendo así la toma de decisiones.

B) Recomendaciones

- ✓ Escalado progresivo del sistema: Se sugiere migrar los entornos de prueba hacia un plan más robusto en AWS a medida que crece la demanda, con el fin de asegurar mayor capacidad de procesamiento y almacenamiento.
- ✓ Capacitación continua al personal: Se recomienda brindar talleres periódicos al equipo de trabajo, especialmente en relación con el uso del sistema, actualizaciones futuras y mejores prácticas en la gestión digital de clientes.
- ✓ Implementar nuevos módulos funcionales: A futuro, se podría considerar la integración de módulos complementarios como facturación electrónica, reportes gerenciales en tiempo real y gestión de marketing digital para fortalecer la relación con los clientes.
- ✓ Monitoreo continuo del sistema: Se aconseja establecer mecanismos automáticos de monitoreo, alertas y métricas de uso mediante herramientas como AWS CloudWatch, con el objetivo de prevenir fallos y optimizar el rendimiento en producción.
- ✓ Evaluar la experiencia del cliente regularmente: Se sugiere aplicar encuestas de satisfacción con cierta frecuencia para medir la percepción de los usuarios y recoger retroalimentación útil que permita mejorar continuamente el sistema.

CAPITULO VI

REFERENCIALES BIBLIOGRÁFICAS

González Duque, R. (2020). *Python para todos.* España: Creative Commons Reconocimien.

Albertos Gómez, E. (2018). *Arquitecturas Sopftware para microservicios: Una Revisión Sistemática de la Literatura.* Madrid: Universidad Politécnica de Madrid.

amazon. (2023). *AWS*. Obtenido de AWS: https://aws.amazon.com/es/what-is/web-application/

Amazon Web Services. (28 de 01 de 2025). *aws.amazon.com*. Obtenido de aws: https://aws.amazon.com/es/pm/dynamodb/?gclid=CjwKCAiAneK8BhAVEiwAoy2HYUHp8I4faLNf1IzE53BNbyJW0_YEF7tQTfspnPEnETLMOBy2BkFOyBoCgu4QAvD_BwE&trk=a24e7032-fe80-48b2-87dc-b9d3f88f695e&sc_channel=ps&ef_id=CjwKCAiAneK8BhAVEiwAoy2HYUHp8I4faLNf1IzE53BNbyJW0_YEF7tQT

Amazon, A. (1 de 12 de 2023). *AWS*. Obtenido de https://aws.amazon.com/es/what-is/gpt/

AWS. (24 de 02 de 2025). *https://docs.aws.amazon.com/*. Obtenido de https://docs.aws.amazon.com/: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html?wpg_intro1

Calle Arevalo, K. Y. (2023). *Aplicación de BPM/CRM como estrategia para la gestión comercial en la Empresa de Calzado Multinegocios KCM S.A.C.* Tarapoto: Universidad Cesar Vallejo.

Carmona Fuentes, A. (2022). *Análisis del proceso de comunicación de CRM con arquitecturas de software externas utilizand open apis.* Toluca: Universidad Autonoma de Mexico.

Casazola Cruz, O., Alfaro Mariño, G., Burgos Tejada , J., & Ramos More, O. (2021). La Usabilidad Percibida de los Chatbots sobre la atención al cliente en las organizaciones: Una revisión de la Literatura. *INTERFASES*, 21.

Celi Párraga, R., Boné Andrade, M., & Mora Olivero, A. (2023). *Programación Web Del Frontend al Backend.* Santo Domingo: Grupo AEA.

Chamba Méndez, C. S. (2022). *Sistema CRM para la gestión de atención al cliente en las cooperativas de ahorro y crédito la ciudad de Guayaquil* . Guayaquil : Universidad Tecnológica Empresarial de Guayaquil - UTEG.

Cristancho, F. (8 de Febrero de 2022). *talently*. Obtenido de https://talently.tech/blog/que-es-un-framework-en-programacion/

DataStaxt. (28 de 01 de 2025). *datastax.com*. Obtenido de datastax: https://www.datastax.com/guides/what-is-cassandra

Domínguez Chávez, J. (2019). *FUNDAMENTOS DE POSTGRESQL.* Venezuela: IEASS.

Fernández Ferrer, M. (2023). *Chatbots en Educación : Tendencias actuales y desafíos futuros.* Barcelona: Learning, Media & Social Interactions.

Fernández Iglesias, M. (2023). Pequeña Introducción a las bases de datos. *atlanTTic*, 22.

Franganillo, J. (2023). La inteligencia Artificial Generativa y su Impacto en la Creación de contenidos Mediáticos. *Methaodos*, 17.

Garrido Moreno, A. (2008). *La gestión de relaciones con clientes (CRM) como estrategia de negocios: Desarrollo de un modelo éxitoso.* Málaga: SPICUM.

Gómez Gonzáles, J. A., & Ramírez Madriz, J. F. (2022). *Implementar una herramienta CRM, así como la evaluación de sus resultados, que permiten la mejora en la gestión de la constructora COMACKEN SRL.* Costa Rica: UNIVERSIDAD LATINA SEDE SAN PEDRO.

Guerola Navarro, V. (2022). *Customer Relationship Management (CRM): Innovación.* Valencia.: Universidad Politécnica de Valencia.

IALAB. (2023). *ChatGPT vs GPT- 4 ¿Imperfecto por diseño? Explorando los límites de la inteliDe acuerdo , a.* Latinoamerica: ONU.

IBM. (28 de 01 de 2025). *IBM*. Obtenido de IBM: https://www.ibm.com/mx-es/topics/couchdb

Izura, P. X. (2023). *REACT Práctico: Desde cero a avanzado.* España: ANAYA MULTIMEDIA.

López , D., & Maya, E. (2017). Arquitectura de Software basada en Microservicios para desarrollo de Aplicaciones Web. *TICAL*, 12.

Lorenzo Romero, R. (2022). *Implementación de un software CRM para la mejora en la gestión de atención al cliente en una empresa del sector financiero.* Lima: Universidad Tecnológica del Perú.

Luque Fuentes, D. (2023). *Application of Software tools for the Adaptation of a CRM to the Management of Cleaning Products.* Madrid: Universidad Politécnica de Madrid .

Majid, I., & Lakshmi, Y. (2022). *Artificial Intelligence In Education.* India: SSRN.

Martín López, B., & Liriano García, J. (2019). *Impacto de Arquitecturas de Microservicios en el Desarrollo Web.* Madrid: Universidad Politécnica de Madrid.

Mena Vicente, M. (2023). *Una arquitectura de microservicios para componentes digitales en la web de las cosas.* Almería: Universidad de Almería.

Microsoft Azure. (28 de 01 de 2025). *azure.microsoft.com*. Obtenido de Azure: https://azure.microsoft.com/es-es/products/cosmos-db

MongoDB.com. (28 de 01 de 2025). *Mongodb.com*. Obtenido de mongodb: https://www.mongodb.com/es/company/what-is-mongodb

Montoya Agudelo, C., & Boyero Saavedra, M. (2013). El CRM como herramienta para el servicio al cliente en la organización. *Visión de Futuro*, 151.

Nixon, R. (2019). *Aprender PHP, MySQL y JavaScript Con jQuery, CSS y HTML5.* España: Marcombo.

Palmowski, M. (23 de Agosto de 2023). *kinsta*. Obtenido de kinsta: https://kinsta.com/es/blog/astro-js/

Pino, J., Martínez, P., & Vergara, J. (2020). *Fundamentos de la Programación en JAVA.* España: Círculo Rojo.

Ramírez Pérez, S. (2020). *ESTUDIO DEL FRAMEWORK SPRING, SPRING BOOT Y MICROSERVICIOS.* España: Universidad de Alcalá.

Rebaza Llontop, J. C. (2021). *Implementación de un Sistema de Información CRM en el Minimarkert CHRISS para la mejora en la atención del cliente - Chimbote 2021.* Chimbote: Universidad Católica de los Ángeles de Chimbote.

S. V., R. (2022). *Diseño de Arquitecturas .NET orientadas a Microservicios.* MARCOMBO.

Serrano Valero, R. (2022). *Diseño de arquitecturas .NET orientadas a microservicios.* Marcombo.

Toro Bonilla, M. (2022). *Fundamentos de Programación: JAVA.* Sevilla: Editorial Universidad de Sevilla.

Ulín Ricárdez, J., Pérez Vasconcelos, M., Gómez Domínguez, R., & Castillo Romero , F. (2023). NoSQL: Una alternativa de solución para grandes volúmenes de información. *INNOVACIÓN Y DESARROLLO TECNOLÓGICO REVISTA DIGITAL* , 2-5.

Vera Rivera, F., Gaonas Cuevas, C., & Astudillo, H. (2019). Desarrollo de aplicaciones basadas en microservicios: tendencias y desafios de investigación. *RISTI*, 15.

CAPÍTULO VII

ANEXOS

Encuesta de Satisfacción (Pre y Post)

Proyecto: Implementación de una arquitectura de microservicios para mejorar la gestión de clientes de la funeraria Descanso Eterno S.A.C.

Objetivo de la Encuesta:

Evaluar el nivel de satisfacción de empleados y clientes respecto al sistema de gestión de clientes, antes y después de la implementación del nuevo sistema basado en arquitectura de microservicios.

Instrucciones:

A continuación, se presentan 6 afirmaciones. Por favor, califique su nivel de satisfacción antes y después de la implementación del nuevo sistema, marcando un número del 1 al 10, donde:
- 1 = Muy insatisfecho
- 10 = Muy satisfecho

Nº	Ítem de Evaluación	Satisfacción Pre (1-10)	Satisfacción Post (1-10)
1	El tiempo que tomaba ser atendido era adecuado.		
2	La atención que recibí fue clara y personalizada.		
3	La información brindada por el sistema fue precisa y completa.		
4	El seguimiento de mi solicitud fue oportuno y continuo.		
5	El proceso de atención fue eficiente desde el inicio hasta la entrega del servicio.		
6	En general, me sentí satisfecho con la atención recibida.		

Nombre del Encuestado: _____

Cargo o Relación con la empresa (Empleado/Cliente): _____

Fecha de Aplicación: _____

Firma: _____

Encuesta de Tiempo de Respuesta (Pre y Post)

Proyecto: Implementación de una arquitectura de microservicios para mejorar la gestión de clientes de la funeraria Descanso Eterno S.A.C.

Objetivo de la Encuesta:

Evaluar la percepción de los empleados sobre la rapidez y eficiencia en los tiempos de atención antes y después de la implementación del nuevo sistema basado en microservicios.

Instrucciones:

A continuación, se presentan 6 afirmaciones. Por favor, califique el tiempo de respuesta del sistema antes y después de la implementación del nuevo sistema, marcando un número del 1 al 10, donde:
- 1 = Muy lento o deficiente
- 10 = Muy rápido o eficiente

Nº	Ítem de Evaluación	Tiempo Pre (1-10)	Tiempo Post (1-10)
1	El tiempo de atención al cliente era razonable.		
2	La carga del sistema era rápida y sin retrasos.		
3	Se podía acceder rápidamente a la información del cliente.		
4	El tiempo de espera para completar una solicitud era adecuado.		
5	No se presentaban interrupciones ni demoras al usar el sistema.		
6	La velocidad del sistema facilitaba el trabajo diario.		

Encuesta de Disponibilidad de la Información (Pre y Post)

Proyecto: Implementación de una arquitectura de microservicios para mejorar la gestión de clientes de la funeraria Descanso Eterno S.A.C.

Objetivo de la Encuesta:

Evaluar el grado de acceso y disponibilidad de la información relevante antes y después del uso del nuevo sistema implementado.

Instrucciones:

A continuación, se presentan 6 afirmaciones. Por favor, califique su experiencia sobre la disponibilidad de la información antes y después de la implementación del nuevo sistema, marcando un número del 1 al 10, donde:
- 1 = Muy baja disponibilidad
- 10 = Información siempre accesible y completa

Nº	Ítem de Evaluación	Disponibilidad Pre (1-10)	Disponibilidad Post (1-10)
1	La información del cliente estaba fácilmente disponible.		
2	Los registros eran completos y estaban bien organizados.		
3	Podía acceder a la información desde cualquier dispositivo autorizado.		
4	La disponibilidad de datos me permitía atender con rapidez.		
5	La información del sistema era confiable y estaba actualizada.		
6	Se podía generar reportes fácilmente en cualquier momento.		

Partidas	Denominación	Precio Unit.	Cant.	Sub Total	Total (S/.)
1	**PERSONAL**				3000.00
1.1	***Equipo de Desarrollo***				
1.2	Mendoza Oviedo Michael Nay				3000.00
2	**BIENES**				4999.00
2.1	***Libros, textos y materiales impresos.***				
2.2	Documentos Guía	0.00	15	0.00	0.00
2.3	***Equipos Computacionales y periféricos.***				
2.4	Laptop o Computadora de Escritorio Intel Core i7,16 GB RAM, 1 TB SDD.	4000.00	01	4000.00	4000.00
2.5	***Activos Intangibles - Software***				
2.6	IntelliJ	0.00	00	0.00	0.00
2.7	Visual Studio Code	0.00	00	0.00	0.00
2.8	Git	0.00	00	0.00	0.00
2.9	JAVA	0.00	00	0.00	0.00
2.10	Figma	0.00	00	0.00	0.00
2.11	Windows 11	0.00	01	0.00	999.99
2.12	Junit	0.00	00	0.00	0.00
2.13	Selenium	0.00	00	0.00	0.00
2.14	Postman	0.00	00	0.00	0.00
3.1	**SERVICIOS**				880.00
3.2	Internet	120.00	01	120.00	120.00
3.3	Dominio	150.00	01	150.00	150.00
3.4	Hosting	110.00	01	110.00	110.00
3.5	Servicios Cloud	500.00	01	500.00	500.00
PRESUPUESTO TOTAL		**S/. 8.879.00 (0/100) Soles**			

Fragmento de código del Sprint N°2 – Función Lambda para registro de clientes

```
import json
import boto3
from botocore.exceptions import ClientError
import uuid
import datetime

# Inicializar el cliente de DynamoDB
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Clientes')

def lambda_handler(event, context):
    try:
        # Obtener datos desde el request
        body = json.loads(event['body'])
        nombre = body.get('nombre')
        apellido = body.get('apellido')
        telefono = body.get('telefono')
        direccion = body.get('direccion')

        # Validación de datos obligatorios
        if not nombre or not apellido:
            return {
                'statusCode': 400,
                'body': json.dumps({'error': 'Nombre y apellido son
obligatorios'})
            }

        # Generar un ID único para el cliente
        cliente_id = str(uuid.uuid4())

        # Registrar fecha y hora actual
        fecha_registro = datetime.datetime.now().isoformat()

        # Guardar en DynamoDB
        table.put_item(
            Item={
                'cliente_id': cliente_id,
                'nombre': nombre,
                'apellido': apellido,
                'telefono': telefono,
                'direccion': direccion,
                'fecha_registro': fecha_registro
            }
        )

        return {
```

```
            'statusCode': 201,
            'body': json.dumps({'message': 'Cliente registrado
correctamente', 'cliente_id': cliente_id})
        }

    except ClientError as e:
        return {
            'statusCode': 500,
            'body': json.dumps({'error': str(e)})
        }
```

Lambda: Consultar clientes (filtro + paginación)

```
import json
import boto3
from boto3.dynamodb.conditions import Key, Attr

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Clientes')

def lambda_handler(event, context):
    # Parámetros de query: ?apellido=...&limit=10&lastKey=...
    params = event.get('queryStringParameters') or {}
    apellido = params.get('apellido')
    limit = int(params.get('limit', 10))
    last_key = params.get('lastKey')

    scan_kwargs = {"Limit": limit}

    # Filtro opcional por apellido (requiere GSI por apellido o filtro de
scan)
    if apellido:
        # Si tienes un GSI: table.query(IndexName='ApellidoIndex',
KeyConditionExpression=Key('apellido').eq(apellido), Limit=limit,
ExclusiveStartKey=...)
        scan_kwargs["FilterExpression"] =
Attr('apellido').contains(apellido)

    if last_key:
        scan_kwargs["ExclusiveStartKey"] = json.loads(last_key)

    resp = table.scan(**scan_kwargs)

    items = resp.get('Items', [])
    next_key = resp.get('LastEvaluatedKey')

    return {
        "statusCode": 200,
```

```
        "body": json.dumps({
            "items": items,
            "nextKey": json.dumps(next_key) if next_key else None
        })
    }
```