

REPOSITORIO ACADÉMICO UPC

Sistema de gestión de cotizaciones de servicios para empresas de telecomunicaciones mediante microservicios

Item Type	info:eu-repo/semantics/bachelorThesis
Authors	Sanchez Matos, Ernesto; Ore Quintana, Miguel Angel
Publisher	Universidad Peruana de Ciencias Aplicadas (UPC)
Rights	info:eu-repo/semantics/openAccess
Download date	05/09/2024 06:25:10
Item License	http://creativecommons.org/licenses/by-nc-sa/4.0/
Link to Item	http://hdl.handle.net/10757/671258



UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

PROGRAMA ACADÉMICO DE INGENIERÍA DE SOFTWARE

Sistema de gestión de cotizaciones de servicios para empresas de telecomunicaciones mediante microservicios

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el título profesional de Ingeniero de Software

AUTOR(ES)

Sanchez Matos, Ernesto	0009-0005-7904-8553
Ore Quintana, Miguel Angel	0009-0007-4278-6303

ASESOR(ES)

Subauste Oliden, Daniel Alejandro	0000-0003-1131-1384
-----------------------------------	---------------------

Lima, 16 de noviembre de 2023

DEDICATORIA

Este trabajo está dedicado a nuestras familiares y amigos.

AGRADECIMIENTOS

A nuestros profesores y colegas por su tutela y ejemplo brindados a lo largo de nuestra preparación profesional.

RESUMEN

El presente trabajo se centra en abordar el desafío central que enfrentan las empresas de telecomunicaciones: la gestión efectiva de las cotizaciones de servicios. Con una creciente demanda de servicios y cobertura, se hace evidente la necesidad de optimizar recursos existentes e intercambiar servicios como interconexión de redes, acceso a torres de telefonía, infraestructura de fibra óptica compartida, el uso conjunto de espectro de frecuencias y servicios de roaming; especialmente en un entorno de colaboración entre empresas que agrega complejidad debido a la negociación de acuerdos comerciales y la gestión de recursos compartidos.

Para enfrentar estos desafíos, se desarrolló un sistema de gestión de cotizaciones de servicios basado en microservicios. Se utilizaron tecnologías como Kafka para la mensajería y una arquitectura de microservicios con micro frontends para el backend y frontend. Las tecnologías clave incluyen Angular 15, Java 17, Oracle para la base de datos, Azure para servicios en la nube y Docker para la contenedorización. Se diseñaron arquitecturas de negocio, de datos, de aplicaciones, tecnológicas e integradas para garantizar la construcción de un sistema efectivo y especializado.

La migración exitosa del sistema de cotizaciones existente a esta nueva infraestructura tecnológica y arquitectura de microservicios conduce a mejoras significativas en términos de rendimiento, eficiencia y experiencia del usuario. Se lograron métricas notables en cuanto al tiempo de respuesta, disponibilidad, escalabilidad y seguridad, lo que demuestra la efectividad de este enfoque en sistemas de este rubro.

Palabras clave:

Microservicios; migración de sistema; rendimiento; cotizaciones de servicios; experiencia de usuario.

ABSTRACT

This work focuses on addressing the central challenge faced by telecommunications companies: the effective management of service quotes. With a growing demand for services and coverage, the need to optimize existing resources and exchange services such as network interconnection, access to telephone towers, shared fiber optic infrastructure, the joint use of frequency spectrum and roaming services is evident; especially in an environment of collaboration between companies that adds complexity due to the negotiation of commercial agreements and the management of shared resources.

To address these challenges, a microservices-based service quote management system was developed. Technologies such as Kafka were used for messaging and a microservices architecture with micro frontends for the backend and frontend. Key technologies include Angular 15, Java 17, Oracle for database, Azure for cloud services, and Docker for containerization. Business, data, applications, technological and integrated architectures were designed to guarantee the construction of an effective and specialized system.

The successful migration of the existing quoting system to this new technological infrastructure and microservices architecture leads to significant improvements in terms of performance, efficiency and user experience. Remarkable metrics were achieved in terms of response time, availability, scalability and security, which demonstrates the effectiveness of this approach in systems in this area.

Keywords:

Microservices; system migration; performance; service quotes; user experience.

u20171a420_Ore Quintana, Miguel Angel_Sistema de gestión de cotizaciones de servicios para empresas de telecomunicaciones mediante microservicios

INFORME DE ORIGINALIDAD

3%

INDICE DE SIMILITUD

2%

FUENTES DE INTERNET

1%

PUBLICACIONES

2%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1

Submitted to Universidad Peruana de Ciencias Aplicadas

Trabajo del estudiante

1%

2

dspace.utpl.edu.ec

Fuente de Internet

<1%

3

www.nobleprog.es

Fuente de Internet

<1%

4

Submitted to Universidad Tecnológica del Peru

Trabajo del estudiante

<1%

5

hdl.handle.net

Fuente de Internet

<1%

6

www.semanticscholar.org

Fuente de Internet

<1%

7

upc.aws.openrepository.com

Fuente de Internet

<1%

TABLA DE CONTENIDOS

1	CAPÍTULO 1: DEFINICIÓN DEL PROYECTO.....	1
1.1	Antecedentes	1
1.2	Descripción de la Organización	1
1.3	Análisis del Problema	2
1.4	Objetivos.....	2
1.4.1	General:	2
1.4.2	Específicos.....	3
1.5	Indicadores de Éxito	3
1.6	Planificación del Proyecto	3
1.6.1	Ciclo de vida del proyecto	3
1.6.2	Alcance del proyecto	4
1.6.3	Roles y responsabilidades.....	5
1.6.4	Registro de riesgos.....	6
2	CAPÍTULO 2: MARCO TEÓRICO	8
2.1	Relacionado a la industria y procesos de negocio	8
2.2	Relacionado a la Ingeniería de Sistemas.....	10
3	CAPÍTULO 3: DESARROLLO DEL PROYECTO	19
3.1	Análisis de tecnologías	19
3.1.1	Tecnologías de mensajería.....	19
3.1.2	Arquitecturas	20
3.2	Diseño de la Solución	21
3.2.1	Arquitectura de Negocio.....	21
3.2.2	Arquitectura de Datos.....	22
3.2.3	Arquitectura de Aplicaciones	24

3.2.4	Arquitectura Tecnológica	25
3.2.5	Arquitectura Integrada.....	27
3.3	Desarrollo de la Solución.....	27
3.4	Resultado del Proyecto	37
4	CONCLUSIONES Y RECOMENDACIONES	39
4.1	Conclusiones.....	39
4.2	Recomendaciones	40
5	REFERENCIAS	41
6	ANEXOS	44
6.1	Anexo 1	44
6.2	Anexo 2.....	45
6.3	Anexo 3.....	46
6.4	Anexo 4.....	47

ÍNDICE DE TABLAS

Tabla 1: Roles y responsabilidades en el proyecto	5
Tabla 2: Riesgos identificados en el proyecto	6
Tabla 3: Cloud Services.....	18
Tabla 4: Benchmarking de tecnologías de mensajería	19
Tabla 5: Benchmarking de arquitecturas	20
Tabla 6: Función de las tablas del diagrama de datos	23
Tabla 7: Ecosistema de aplicaciones de Corporation	24
Tabla 8: Comparación de métricas As-Is y To-Be	38
Tabla 9: Plan de costos	46

ÍNDICE DE FIGURAS

Figura 1: Ciclo de vida del sistema de cotizaciones	3
Figura 2: Proceso de negocio en empresas de telecomunicaciones.....	9
Figura 3: Arquitectura monolítica vs arquitectura de microservicios	10
Figura 4: Arquitectura de microservicios de un sistema de comercio electrónico.....	11
Figura 5: Equipos End-to-End con micro frontends.....	13
Figura 6: Arquitectura Docker.....	14
Figura 7: Arquitectura Kafka	16
Figura 8: Diagrama de negocio de Corporation	21
Figura 9: Arquitectura de datos	22
Figura 10: Arquitectura de Aplicaciones.....	24
Figura 11: Arquitectura Tecnológica.....	26
Figura 12: Arquitectura Integrada	27
Figura 13: Login As-Is	28
Figura 14: Login To-Be.....	28
Figura 15: Bandeja As-Is.....	29
Figura 16: Bandeja To-Be	29
Figura 17: Comparación de precio As-Is.....	30
Figura 18: Comparación de precio To-Be	30
Figura 19: Detalle de cotización: Clear Channel As-Is Parte 1	31
Figura 20: Detalle de cotización: Clear Channel As-Is Parte 2.....	31
Figura 21: Detalle de cotización: Clear Channel To-Be Parte 1	32
Figura 22: Detalle de cotización: Clear Channel To-Be Parte 2	33
Figura 23: Detalle de cotización: Internet As-Is Parte 1	33
Figura 24: Detalle de cotización: Internet As-Is Parte 2	34
Figura 25: Detalle de cotización: Internet To-Be Parte 1.....	34
Figura 26: Detalle de cotización: Internet To-Be Parte 2.....	35
Figura 27: Detalle de cotización: MPLS As-Is Parte 1	35
Figura 28: Detalle de cotización: MPLS As-Is Parte 2	36
Figura 29: Detalle de cotización: MPLS To-Be Parte 1	36

Figura 30: Detalle de cotización: MPLS To-Be Parte 2.....	37
Figura 31: Proceso de diseño en empresas de telecomunicaciones.....	44
Figura 32: Proceso de análisis jerárquico para empresas de telecomunicaciones	45
Figura 33: Plan de continuidad	47

1 CAPÍTULO 1: DEFINICIÓN DEL PROYECTO

1.1 Antecedentes

En la actualidad, las empresas de telecomunicaciones se enfrentan a grandes desafíos a medida que la demanda de servicios y la cobertura continúan expandiéndose, por lo que se ven presionados para mantener su competitividad en el mercado mediante la innovación de productos y servicios. Con 9 408 393 hogares peruanos que poseen al menos un servicio de telecomunicaciones, y con un 80.4% de estos mismos con más de un servicio contratado (Castillo & Chahuara, 1970), surge la necesidad de colaboración entre este tipo de empresas para optimizar el uso de los recursos existentes, reducir costos y mejorar la calidad de los servicios ofrecidos a los clientes.

Por ello, muchas de estas compañías han desarrollado sistemas de gestión de cotizaciones de servicios específicamente diseñados para este rubro. Estos sistemas permiten que las empresas compartan servicios entre sí, lo que incluye interconexión de redes, acceso a torres de telefonía, infraestructura de fibra óptica compartida, el uso conjunto de espectro de frecuencias y la posibilidad de ofrecer servicios de roaming.

En la búsqueda de optimizar estos sistemas, los microservicios se han establecido como una alternativa prometedora para la implementación y migración de estos sistemas. Con una gran capacidad para modernizar sistemas heredados, haciéndolos más flexibles y fáciles de mantener o modificar, los microservicios se presentan como una solución que merece una atención cuidadosa debido a su potencial para abordar los desafíos en la gestión de productos y servicios de manera más efectiva (Gan et al., 2019).

1.2 Descripción de la Organización

La organización que se analiza es un actor destacado en el competitivo sector de las telecomunicaciones, brindando un amplio catálogo de servicios que incluyen telefonía móvil y fija, servicios de Internet de alta velocidad, televisión por cable y streaming de contenido.

Esta organización se caracteriza por una gestión ágil y eficiente de su cartera de productos y servicios diversificados. Los procesos de desarrollo de nuevos servicios y productos se basan en una comprensión profunda de las necesidades cambiantes de los clientes y las tendencias del mercado. La agilidad es una prioridad, permitiendo la rápida adaptación a las demandas del mercado y la implementación de nuevos servicios innovadores.

La atención al cliente es una parte central de la estrategia de la organización. Se ha establecido un sistema de atención al cliente que abarca desde la gestión de consultas y solicitudes hasta la resolución de problemas de manera eficiente. Se utilizan sistemas de gestión de relaciones con el cliente (CRM) para garantizar un seguimiento integral de las interacciones y personalizar la experiencia del cliente en todos los servicios.

Internamente, la organización ha adoptado sistemas de planificación de recursos empresariales (ERP) para optimizar la gestión de recursos, la cadena de suministro y la logística, lo que garantiza la disponibilidad constante de equipos y componentes críticos.

En términos de seguridad y privacidad, la empresa prioriza la protección de datos mediante el uso de sistemas avanzados de cifrado y autenticación. Se realizan evaluaciones de seguridad y pruebas de penetración de manera regular para identificar y mitigar posibles vulnerabilidades. Además, la empresa cumple rigurosamente con las regulaciones de privacidad de datos y utiliza sistemas de gestión de la privacidad de los clientes para garantizar un manejo seguro y autorizado de la información personal.

1.3 Análisis del Problema

El problema central que enfrenta la empresa de telecomunicaciones es la gestión efectiva de las cotizaciones de servicios. En particular, la interacción y colaboración entre empresas agrega una capa adicional de complejidad a esta gestión, ya que involucra la negociación de acuerdos comerciales y la gestión de recursos compartidos, de por sí ya limitados. Las empresas de telecomunicaciones están sujetas a regulaciones gubernamentales que pueden limitar la forma en que colaboran y comparten recursos por lo que establecer los términos y condiciones, incluyendo los costos asociados, deben realizarse meticulosamente.

Para abordar estos desafíos, las empresas de telecomunicaciones deben contar con sistemas de gestión de cotizaciones de servicios efectivos y ágiles. Estos sistemas deben ser capaces de gestionar la complejidad de las negociaciones, garantizar el cumplimiento de las regulaciones gubernamentales, mantener la seguridad de la red y los datos, y facilitar la colaboración eficiente entre empresas.

1.4 Objetivos

1.4.1 General:

OG: Desarrollar un sistema de gestión para cotizaciones de servicios mediante microservicios para mejorar su intercambio entre empresas de telecomunicaciones.

1.4.2 Específicos

OE1: Analizar las opciones disponibles para la arquitectura y tecnologías de mensajería que se ajusten al propósito de la aplicación.

OE2: Diseñar la arquitectura de la aplicación considerando las tecnologías analizadas y su compatibilidad con las herramientas a usar.

OE3: Desarrollar el sistema de control de productos/servicios utilizando las tecnologías analizadas.

OE4: Validar el funcionamiento y mejora del rendimiento y mantenibilidad de la aplicación.

1.5 Indicadores de Éxito

OE1-I1: Benchmarking de tecnologías investigadas y establecidas para el proyecto.

OE2-I2: Diagramas de arquitectura de software para el sistema.

OE3-I3: Sistema WQTool desarrollado en entornos de desarrollo y calidad.

OE4-I4: Métricas de mejora de rendimiento con la migración a microservicios.

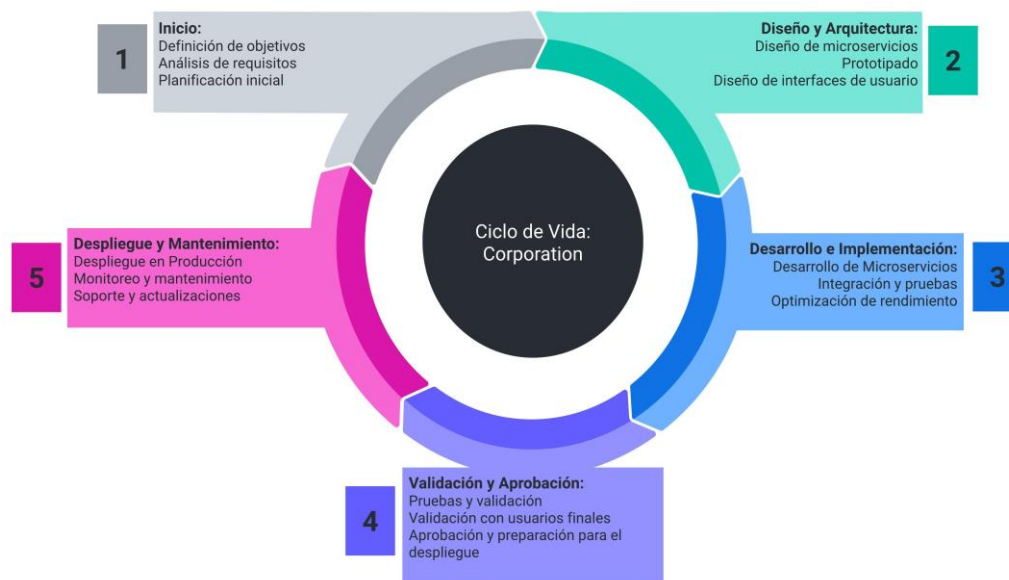
1.6 Planificación del Proyecto

1.6.1 Ciclo de vida del proyecto

La Figura 1 muestra el ciclo de vida del desarrollo del sistema de cotizaciones de Corporation, la empresa de telecomunicaciones. En ella se definen sus fases (inicio, diseño y arquitectura, desarrollo e implementación, validación y aprobación, y despliegue y mantenimiento), así como los elementos clave asociados a cada uno de ellos (definición análisis, planificación, diseño prototipado, etc.).

Figura 1

Ciclo de vida del sistema de cotizaciones



1.6.2 Alcance del proyecto

El objetivo principal del proyecto es diseñar, desarrollar e implementar un sistema de gestión de cotizaciones de servicios para empresas de telecomunicaciones mediante microservicios,

1.6.2.1 Descripción del Alcance:

- **Desarrollo de Microservicios:** Se desarrollarán una serie de microservicios que formarán el núcleo del sistema de gestión de cotizaciones. Cada microservicio se encargará de una función específica, como la gestión de clientes, productos, cotizaciones, etc.
- **Interfaz de Usuario (UI) basada en Angular 15:** Se creará una interfaz de usuario moderna y amigable basada en Angular 15, utilizando el enfoque de micro frontends para dividir la interfaz en componentes independientes.
- **Conexión con la Base de Datos Oracle:** Se establecerá una conexión con la base de datos Oracle, que se encuentra en un servidor dedicado, para almacenar y recuperar datos relacionados con las cotizaciones y otros elementos del sistema.
- **Contenedorización con Docker:** Todos los microservicios, la interfaz de usuario y cualquier otro componente necesario se contenerizarán utilizando Docker para facilitar el despliegue y la administración.
- **Integración con Azure:** El sistema se implementará y ejecutará en el entorno de Azure, aprovechando las capacidades de la plataforma para garantizar la escalabilidad, disponibilidad y seguridad del sistema.

- **Capacitación y Soporte:** Se proporcionará capacitación a los usuarios finales y al personal de soporte para garantizar un uso efectivo del sistema. Además, se ofrecerá soporte técnico continuo.

1.6.2.2 Exclusiones del Alcance

- **Integración con otros sistemas Legacy:** Este proyecto no abordará la integración con otros sistemas Legacy de la empresa de telecomunicaciones que no estén directamente relacionados con el sistema de gestión de cotizaciones (WQTool).
- **Desarrollo de Aplicaciones Móviles:** No se emprenderá el desarrollo de aplicaciones móviles nativas como parte del proyecto. Sin embargo, se implementará una interfaz web responsive para facilitar el acceso desde dispositivos móviles.
- **Rediseño de Procesos de Negocio:** No se llevará a cabo una revisión completa de los procesos de negocio de la empresa de telecomunicaciones en este proyecto. El sistema se adaptará a los procesos existentes en la medida de lo posible.
- **Ampliación de Funcionalidades a Largo Plazo:** Se considerarán adiciones significativas de funcionalidades que van más allá de la gestión de cotizaciones como parte de futuros proyectos, pero estas expansiones no están incluidas en el alcance actual.

1.6.3 Roles y responsabilidades

Dentro del proyecto, existen diferentes empleados con roles y responsabilidades específicas. La tabla 1 muestra estos roles y funciones que se desempeñan para llevar a cabo el desarrollo del sistema de cotizaciones.

Tabla 1

Roles y responsabilidades en el proyecto

Rol	Responsabilidad
Project Manager	<ul style="list-style-type: none"> ● Elabora, supervisa y confirma el logro de los objetivos del proyecto. ● Establece una comunicación continua con las partes interesadas en el proyecto para garantizar su visión y estado. ● Administra los requerimientos solicitados por el cliente.

Software Architect	<ul style="list-style-type: none"> • Diseña la arquitectura del proyecto tomando en cuenta las decisiones y atributos clave que se priorizan en el proyecto.
Software Developer	<ul style="list-style-type: none"> • Personal especializado en diferentes tecnologías (front-end, back-end) para implementar con éxito el proyecto.
Business Analyst	<ul style="list-style-type: none"> • Traduce los requisitos del negocio especificados por los stakeholders en requisitos técnicos para el equipo.
UI/UX Designer	<ul style="list-style-type: none"> • Diseñador de la interfaz de usuario, en base al cumplimiento de estándares de usabilidad.
Quality Assurance	<ul style="list-style-type: none"> • Responsable de testear el software para identificar los posibles errores y problemas de calidad que pueda presentar. • Emite su aprobación para el entregable.
Security Specialist	<ul style="list-style-type: none"> • Garantiza que el software sea seguro en cuanto a vulnerabilidades o posibles amenazas.
Data Engineer	<ul style="list-style-type: none"> • Administrador y optimizador de la base de datos.
Scrum Master	<ul style="list-style-type: none"> • Segura el cumplimiento de Scrum en el proyecto. • Facilita el proceso y el flujo de trabajo del equipo, garantizando una comunicación adecuada y eliminando posibles obstáculos en el flujo de trabajo.

1.6.4 Registro de riesgos

La Tabla 2 muestra la definición de los riesgos encontrados para el desarrollo del sistema de cotizaciones en su ciclo de vida.

Tabla 2

Riesgos identificados en el proyecto

Riesgo Identificado	Causa del Riesgo	Impacto del Riesgo	Posibles Respuestas
Cambios frecuentes en los Requisitos	Cambios frecuentes o definiciones poco	Retrasos en el cronograma, incremento de	Establecer un proceso sólido de gestión de cambios y mantener

	claras en los requisitos del cliente.	costos y descontento del cliente.	una comunicación efectiva con el cliente para la definición y documentación precisa de requisitos.
Falta de Experiencia en el equipo	Miembros del equipo con o sin experiencia en las tecnologías involucradas en el proyecto.	Baja calidad del código, retrasos y aumento de costos debido a errores y aprendizaje sobre la marcha.	Capacitar al equipo en las tecnologías requeridas y considerar la contratación de expertos o consultores externos según sea necesario.
Problemas de Integración	Dificultades para integrar los componentes o sistemas del proyecto.	Retrasos en el proyecto, problemas de funcionamiento y errores en la producción.	Realizar pruebas de integración de forma continua, seguir normativas de diseño estandarizadas y mantener una comunicación efectiva entre los equipos responsables de los diferentes componentes.
Falta de Recursos	Carencia de recursos humanos, financieros o tecnológicos para el proyecto.	Retrasos en el cronograma, baja calidad y posible cancelación del proyecto.	Realizar una evaluación inicial de los recursos, buscar financiamiento adicional en caso necesario y ajustar el cronograma en función de los recursos disponibles.
Vulnerabilidades de Seguridad	Brechas de seguridad en el software	Puede resultar en pérdida de datos,	Implementar pruebas exhaustivas de

	susceptibles de ser explotadas.	daño a la reputación y consecuencias legales adversas.	seguridad, adherirse a buenas prácticas de desarrollo seguro y mantener el software actualizado para abordar vulnerabilidades conocidas.
Cambios en la Tecnología	Modificaciones tecnológicas que requieren adaptaciones en el proyecto.	Necesidad de revisar el alcance, probabilidad de retrasos y aumento de costos.	Mantenerse informado acerca de las tendencias tecnológicas, realizar un análisis de impacto ante cambios tecnológicos y ajustar el plan del proyecto en consecuencia.

2 CAPÍTULO 2: MARCO TEÓRICO

2.1 Relacionado a la industria y procesos de negocio

Procesos de negocio en una empresa de telecomunicaciones:

En el ámbito de las telecomunicaciones, los procesos de negocio desempeñan un papel fundamental en la capacidad de una empresa para desarrollar y ofrecer productos y servicios de manera eficiente. La gestión efectiva de estos procesos es esencial para mantener la competitividad y satisfacer las demandas del mercado (Lubis et al., 2020).

Es importante considerar los diferentes tipos de innovación en este proceso. Estos incluyen la innovación incremental, que se centra en la mejora continua de los procesos de negocio para integrar funciones de manera más eficiente; la innovación responsiva, que implica explorar las demandas de los usuarios en profundidad para ofrecer procesos más sencillos y satisfactorios; la innovación disruptiva, que transforma procesos simples u offline en plataformas inmersivas en línea; y la innovación radical, que ofrece ideas arriesgadas que

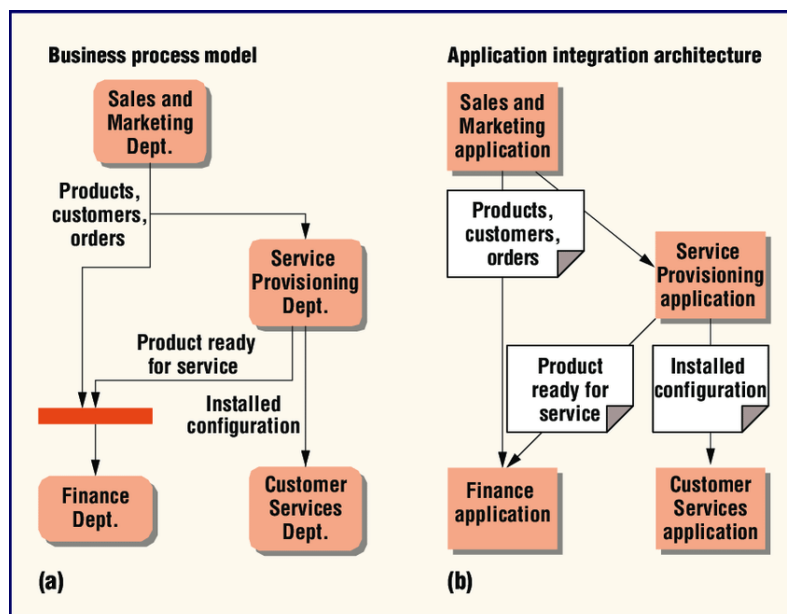
tienen como objetivo eliminar plataformas existentes a través de mejoras pioneras en la creación de valor (Viveros et al., 2019).

La selección de proveedores también es un aspecto crítico para garantizar la adquisición de equipos y servicios adecuados. Según la Fuente 2, esta elección debe considerar criterios como la calidad, la cantidad, las fechas de entrega y los métodos de pago. Para mejorar este proceso, se pueden utilizar enfoques de selección de proveedores basados en múltiples criterios, como el Análisis de la Jerarquía (AHP, por sus siglas en inglés), que permite comparar proveedores en función de múltiples dimensiones (Fuente 2), ampliando su espectro a otras empresas de telecomunicaciones.

El proceso de toma de decisiones en la selección de proveedores implica la identificación de criterios relevantes y la comparación de alternativas. Este proceso se basa en una jerarquía que implica la priorización de criterios y la evaluación de proveedores en función de esos criterios (Lubis et al., 2020). Sin embargo, es importante reconocer que algunos criterios pueden ser cualitativos y difíciles de cuantificar, como aspectos políticos, sociales y ambientales. En estos casos, se pueden utilizar métodos como el AHP con números triangulares difusos para representar la incertidumbre en la toma de decisiones (Viveros et al., 2019).

Figura 2

Proceso de negocio en empresas de telecomunicaciones



Nota. Flujo de negocio en una empresa de telecomunicaciones: (a) proceso de negocio principal y (b) integración de aplicaciones. De “Coordinating COTS Applications via a Business Event Layer”, por Lemahieu et al., 2005 (<https://doi.org/10.1109/ms.2005.90>).

2.2 Relacionado a la Ingeniería de Sistemas

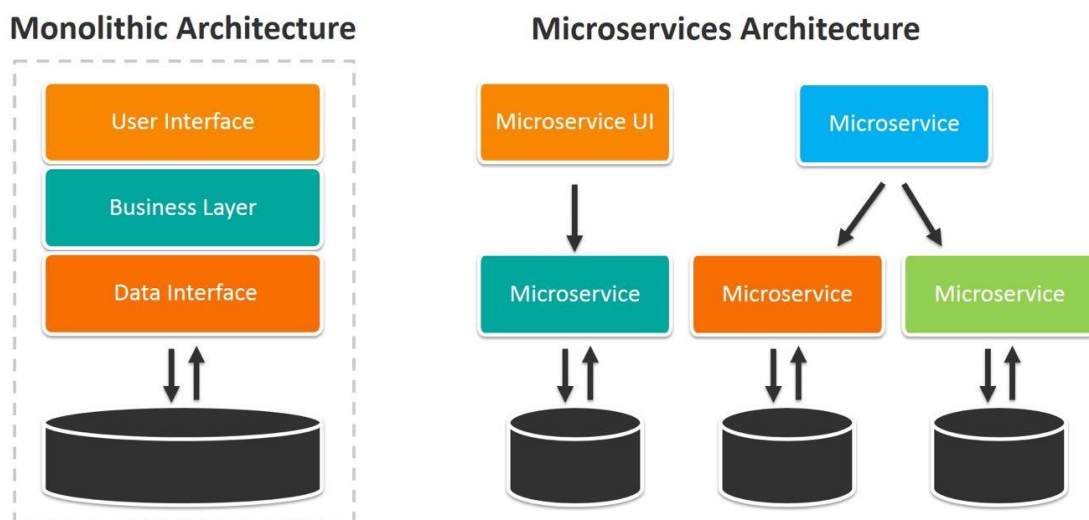
Arquitectura monolítica:

Es un enfoque de diseño de software en el que una aplicación se desarrolla como una única unidad monolítica. En este tipo de arquitectura, todas las funciones y componentes del sistema están integrados en una sola aplicación, lo que significa que todas las partes del software se ejecutan en el mismo proceso y comparten la misma base de código (Atlassian, 2022). Los sistemas monolíticos suelen caracterizarse por su simplicidad, ya que no involucran múltiples servicios o microservicios independientes, y suelen ser más fáciles de desarrollar e implementar en comparación con arquitecturas más distribuidas.

En la arquitectura monolítica, todas las funcionalidades están contenidas en una sola unidad de ejecución y todos los componentes comparten la misma base de código. Esta arquitectura posee un acoplamiento fuerte, lo que significa que los cambios en un área pueden afectar a otras partes del sistema (Lahtela & Kaplan, 2023). Además, debido a su escalabilidad limitada y despliegue unitario, los cambios pueden ser complicados ya que suelen implicar a toda la aplicación.

Figura 3

Arquitectura monolítica vs arquitectura de microservicios



Nota. Diferencias entre la arquitectura monolítica y de microservicios. De “Microservices vs. Monolithic Architectures”, por Rancher, 2021 (https://www.suse.com/c/rancher_blog/microservices-vs-monolithic-architectures/).

Microservicios:

Los microservicios representan una evolución fundamental en la arquitectura de software, diseñada para abordar los desafíos contemporáneos en el desarrollo y mantenimiento de aplicaciones (Gan et al., 2019). Esta arquitectura se caracteriza por la fragmentación de una aplicación monolítica en componentes autónomos y pequeños, conocidos como microservicios (Aragon et al., 2019). Cada microservicio encapsula una funcionalidad específica, operando como una entidad independiente con su propia lógica de negocio y base de datos.

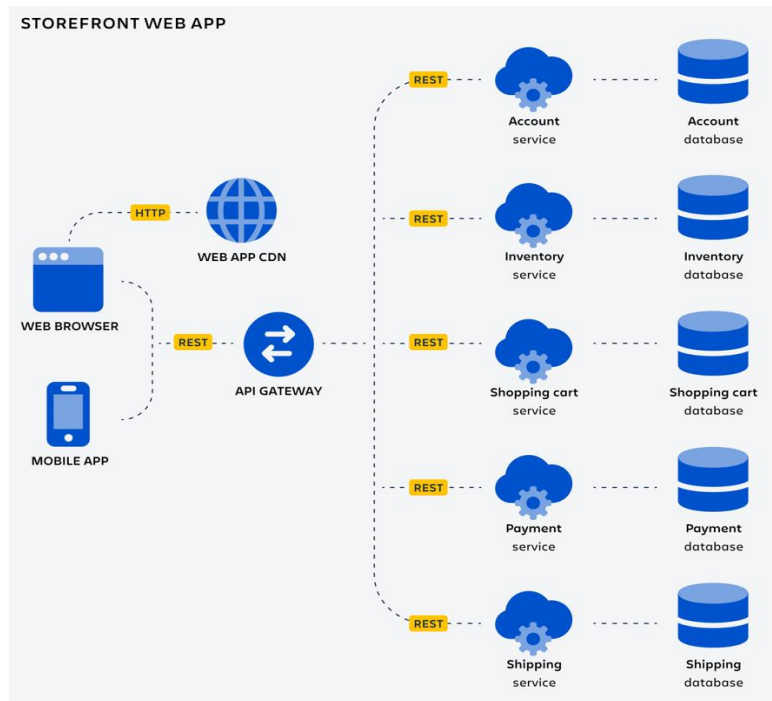
La independencia es una de las características más resaltantes de los microservicios. Cada uno puede desarrollarse, probarse y desplegarse de manera independiente. Además, la comunicación entre microservicios se realiza a través de protocolos ligeros como REST o RPC, lo que facilita la interoperabilidad (Aragon et al., 2019).

La adopción de microservicios ha aumentado debido a su capacidad para abordar los desafíos específicos de las aplicaciones basadas en la nube, como la heterogeneidad tecnológica y la escalabilidad (Aragon et al., 2019). Estos componentes desacoplados permiten implementaciones y despliegues independientes, lo que facilita la evolución continua de la aplicación.

La arquitectura de microservicios fomenta la adopción de prácticas modernas de desarrollo de software, como DevOps y CI/CD, al simplificar la gestión de código y permitir una mayor automatización en la implementación (Gan et al., 2019).

Figura 4

Arquitectura de microservicios de un sistema de comercio electrónico



Nota. Diseño de una arquitectura para una aplicación e-commerce. De “Arquitectura de microservicios”, por Atlassian, 2022

(<https://www.atlassian.com/es/microservices/microservices-architecture>).

Micro frontends:

La arquitectura de micro frontends representa un avance significativo en el desarrollo de aplicaciones web, diseñada para superar los desafíos de escalabilidad y mantenibilidad en entornos complejos. Esta estrategia implica la división de la interfaz de usuario en componentes independientes, conocidos como micro frontends, cada uno de los cuales abarca una parte específica de la interfaz (Geers, 2020).

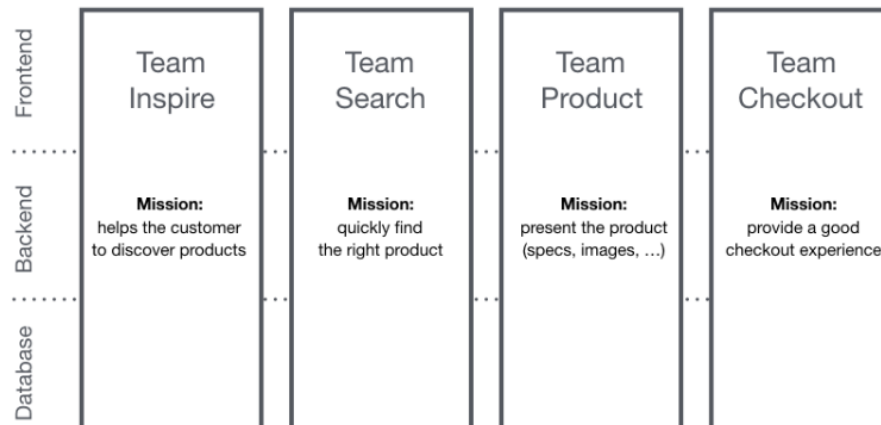
Un elemento distintivo de los micro frontends es su capacidad para operar de forma autónoma y ser desarrollados, probados y desplegados independientemente. Esto facilita la colaboración entre equipos de desarrollo y permite actualizaciones incrementales sin interrupciones en otras áreas de la aplicación (Harms et al., 2017).

La independencia de los micro frontends brinda a los equipos de desarrollo la flexibilidad para adoptar tecnologías y herramientas específicas para cada componente de la interfaz. Esto permite la incorporación de nuevas características y actualizaciones sin necesidad de realizar cambios a gran escala en toda la aplicación.

A pesar de sus ventajas, la arquitectura de micro frontends plantea desafíos en términos de gestión de dependencias y coordinación entre equipos. Sin embargo, estos desafíos pueden abordarse mediante la implementación de buenas prácticas de diseño y la adopción de enfoques de CI/CD (Geers, 2020).

Figura 5

Equipos End-to-End con micro frontends



Nota. Arquitectura de micro frontends con equipos de trabajo y objetivos diferentes. De “Micro frontends”, por M. Geers, 2022 (<https://micro-frontends-es.org/>)

Docker:

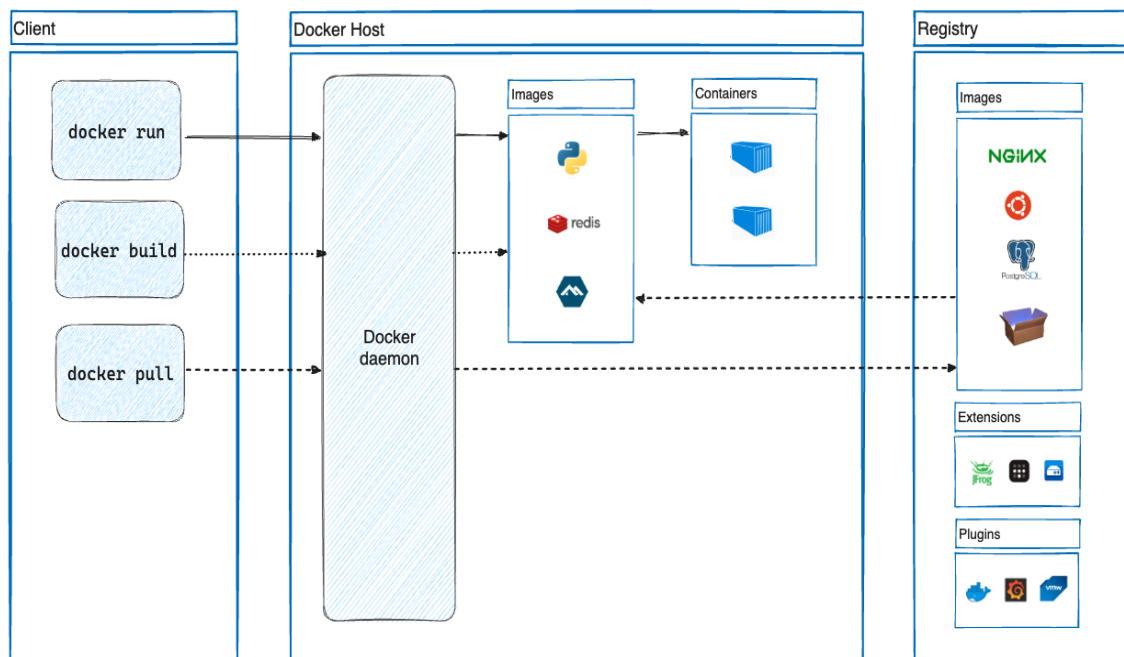
Docker se ha consolidado como una de las herramientas de contenerización más populares en la práctica de DevOps, con millones de usuarios y un elevado número de descargas (Docker, 2023).

La esencia de Docker radica en su capacidad para encapsular aplicaciones y todas sus dependencias en contenedores, lo que proporciona dos ventajas cruciales. En primer lugar, ofrece portabilidad, permitiendo que las aplicaciones se ejecuten de manera consistente en diversos entornos. Esto se traduce en un proceso de desarrollo y despliegue más fluido, ya que los desarrolladores pueden tener la confianza de que su aplicación funcionará igual en cualquier lugar donde se ejecute. En segundo lugar, simplifica significativamente la gestión de dependencias, ya que los contenedores pueden depender de otros contenedores, lo que crea relaciones de dependencia que se pueden gestionar con eficacia (Zhang et al., 2020).

La definición de un contenedor Docker se encuentra en un archivo conocido como "Dockerfile". Este archivo especifica los comandos de Docker y el orden en que deben ejecutarse para construir el contenedor deseado. Además, cada contenedor suele comenzar desde una "imagen base", que es un componente fundamental y puede influir en el desarrollo y la aplicación del contenedor (Docker, 2023).

Figura 6

Arquitectura Docker



Nota. Ejemplo de diseño de una arquitectura para una aplicación usando Docker. De “Docker overview”, por Docker, 2023 (<https://docs.docker.com/get-started/overview/#:~:text=Docker%20architecture,to%20a%20remote%20Docker%20daemon.>)

Spring Boot:

Spring Boot es un marco de desarrollo de aplicaciones Java popular en la creación de microservicios. Esta tecnología se basa en el marco de trabajo Spring, pero se distingue por su capacidad para simplificar la configuración y la implementación de aplicaciones empresariales escalables y autónomas (Spring, 2023). Una de las características clave que distingue a Spring Boot es su capacidad para incorporar un servidor web embebido, como Tomcat o Jetty, en la propia aplicación. Esto significa que un servicio Spring Boot se ejecuta

de manera autónoma sin requerir un servidor de aplicaciones externo. Además, ofrece un conjunto de herramientas para el empaquetamiento de aplicaciones como contenedores independientes, lo que facilita la implementación y la gestión de múltiples microservicios en entornos distribuidos.

Angular:

Angular es un marco de desarrollo de aplicaciones web de código abierto mantenido por Google y una comunidad activa de desarrolladores. Se basa en el lenguaje TypeScript y utiliza el patrón de diseño de arquitectura de componentes para crear aplicaciones web modernas y escalables. Angular proporciona un conjunto de bibliotecas y herramientas que simplifican la creación de aplicaciones web complejas al ofrecer una estructura organizativa sólida y un flujo de desarrollo coherente (Google, 2022).

Una de las características fundamentales de Angular es su enfoque en la reutilización de componentes, lo que permite dividir la interfaz de usuario en partes independientes y reutilizables. Estos componentes pueden comunicarse entre sí a través de una jerarquía de componentes, lo que facilita la gestión del estado de la aplicación y la creación de interfaces de usuario dinámicas.

Además, Angular ofrece un sistema de inyección de dependencias que simplifica la gestión de las dependencias y la configuración de los servicios necesarios para la aplicación. Esto fomenta la modularidad y la extensibilidad de las aplicaciones, lo que facilita la incorporación de nuevas características y la realización de pruebas unitarias.

RabbitMQ:

Es una plataforma de mensajería de código abierto que se utiliza para facilitar la comunicación entre aplicaciones y sistemas distribuidos (RabbitMQ, 2007). Actúa como un intermediario que permite a las aplicaciones enviar y recibir mensajes de manera eficiente y confiable. Basado en el protocolo de mensajería Advanced Message Queuing Protocol (AMQP), RabbitMQ proporciona una forma escalable y flexible de transmitir datos y eventos entre diferentes componentes de un sistema, lo que es fundamental en arquitecturas de software orientadas a microservicios y sistemas distribuidos.

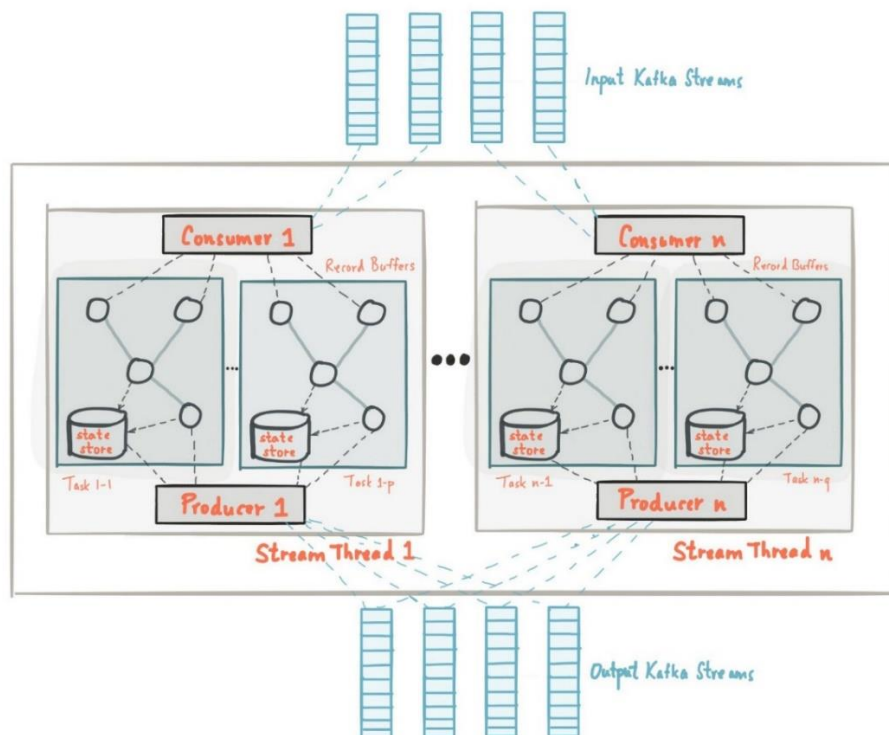
Apache Kafka:

Apache Kafka es una plataforma de transmisión de datos de código abierto que se utiliza para la ingesta, el almacenamiento y la transmisión de flujos de eventos en tiempo real (Kafka, 2016). Diseñado para manejar grandes volúmenes de datos y garantizar la escalabilidad y la tolerancia a fallos, Kafka permite a las aplicaciones y sistemas distribuidos compartir datos de manera eficiente y confiable. Utiliza un modelo de publicación-suscripción y almacena los eventos en registros inmutables, lo que lo convierte en una herramienta esencial para casos de uso como el procesamiento de registros, análisis en tiempo real y la integración de sistemas en arquitecturas de microservicios.

La figura 7 muestra la arquitectura de una aplicación que utiliza Kafka.

Figura 7

Arquitectura Kafka



Nota. De “Kafka Documentation”, por Apache Kafka, 2016

(<https://kafka.apache.org/documentation/>)

Cloud Services

Los servicios en la nube comprenden una amplia gama de servicios proporcionados a los clientes a través de Internet. Estos servicios suelen incluir infraestructuras, plataformas y

software que son alojados por proveedores de computación en la nube, quienes ponen a disposición de los usuarios estas herramientas y recursos.

Dada la creciente importancia de la tecnología en las operaciones empresariales, los usuarios buscan cualidades como calidad, seguridad, rendimiento y eficiencia. Los servicios en la nube pueden satisfacer estas demandas, ya que ofrecen un acceso sencillo y rentable a recursos tecnológicos (Red Hat, 2020).

Una de las ventajas más destacadas de la adopción de servicios en la nube es la gran reducción de costos que ofrecen a las empresas, ya que permite acceder a las organizaciones a recursos tecnológicos sin incurrir en gastos considerables relacionados con la infraestructura y el mantenimiento de servidores físicos (Citrix, 2021). Las empresas pueden aprovechar modelos de pago flexibles, como el pago por consumo o suscripciones, lo que les permite adaptar sus gastos de acuerdo con sus necesidades reales.

Tipos de Cloud Services:

- **Infraestructura como Servicio (IaaS):** Con IaaS, las organizaciones pueden desplegar servidores virtuales, almacenamiento y otros recursos de infraestructura sin la necesidad de invertir en hardware físico costoso y su mantenimiento (Oracle, 2020). Además, al poder escalar los recursos según las demandas cambiantes, las empresas pueden optimizar sus gastos y evitar gastos innecesarios. Esto se traduce en una mayor flexibilidad financiera y la capacidad de centrarse en inversiones estratégicas en lugar de la gestión de infraestructura.
- **Plataforma como Servicio (PaaS):** Ofrece a las empresas la posibilidad de desarrollar, desplegar y administrar aplicaciones sin preocuparse por la infraestructura subyacente (Red Hat, 2019). Esto conlleva una ventaja en la reducción de costos, ya que las organizaciones pueden evitar los gastos asociados con la compra y mantenimiento de servidores, sistemas operativos y software de desarrollo. Además, las empresas pueden acelerar el tiempo de comercialización y reducir los costos de desarrollo.
- **Software como Servicio (SaaS):** Con SaaS, las empresas pueden acceder a aplicaciones empresariales de alta calidad mediante suscripciones, lo que simplifica la gestión de software y reduce los costos relacionados con la adquisición y actualización de programas (IBM, 2020). Además, los proveedores de SaaS se

encargan del mantenimiento, la seguridad y las actualizaciones, liberando a las empresas de la carga de estos gastos y permitiéndoles enfocarse en sus operaciones esenciales.

La tabla 3 muestra algunas categorías de cloud services, así como los principales proveedores de estos servicios

Tabla 3

Cloud Services

Cloud Services	Google Cloud	Amazon Web	Microsoft Azure
	Services	Services	
IaaS	Compute Engine	Elastic Compute Cloud – EC2	Azure Virtual Machines
PaaS	App Engine Standard Environment, App Engine Flexible Environment	Elastic Beanstalk	App Service, Cloud Services
Servicio gestionado de Kubernetes	Kubernetes Engine	Elastic Container Service para Kubernetes (EKS)	Azure Kubernetes Service (AKS)
Registro de Contenedores Docker	Container Registry	EC2 Container Registry (ECR)	Azure Container Registry
Serverless	Cloud Functions	Lambda	Azure Functions
Balanceo de Carga	Cloud Load Balancing	Elastic Load Balancing	Load Balancer
API Management	Cloud Endpoints, Apigee Api Management	API Gateway	API Management

3 CAPÍTULO 3: DESARROLLO DEL PROYECTO

3.1 Análisis de tecnologías

3.1.1 Tecnologías de mensajería

Tabla 4

Benchmarking de tecnologías de mensajería

Métricas	Kafka	RabbitMQ
Rendimiento (MB/s)	605	38
Latencia (ms)	5	1
Repetición de mensajes	Durante el periodo de retención	No, los mensajes se eliminan luego de la entrega
ACK	Si	No
Diseño	Productor - Consumidor Inteligente	Productor Inteligente - Cosumidor
Arquitectura	Productor->intercambio- >reglas vinculantes->cola- >consumidor	Productor->tema- >broker->partición- >consumidor
Bibliotecas compatibles	Python, JAVA, Ruby, Nodo.JS	Spring. Python, PHP, .NET, C, Rubí

Nota. Benchmarking realizado entre Kafka y RabbitMQ: dos tecnologías de mensajería consideradas para el proyecto.

La Tabla 4 muestra el benchmarking de las tecnologías de mensajería realizado, del que se concluyó que Kafka es la herramienta idónea para el tipo de proyecto que se va a desarrollar debido a que se está priorizando el rendimiento, consistencia de entregas (ACK), y la repetición de mensajes. Cabe mencionar que el ACK es un mensaje que emite el receptor para confirmar la entrega, de esta manera se asegura que el mensaje fue entregado con éxito. El rendimiento se prioriza debido a que se manejan volúmenes de datos considerables, por lo cual es necesario tener una buena cantidad de MB/s en la transferencia de datos. Por otro lado, Kafka permite la retención de mensajes durante el periodo de retención, mientras que RabbitMQ no, debido a que estos se eliminan luego de la entrega.

3.1.2 Arquitecturas

Tabla 5

Benchmarking de arquitecturas

ATRIBUTOS	MONOLITO	MICROSERVICIOS
Implementación	Fácil	Compleja
Escalabilidad	Baja	Alta
Modificabilidad	Baja	Alta
Acoplamiento de funcionalidades	Alta	Baja
Latencia	Baja	podría ser alta
Portabilidad	Alta	Baja, es necesario realizar varios despliegues
Lenguaje de desarrollo	Se puede desarrollar en un lenguaje de backend y otro de frontend	Se puede hacer uso de múltiples lenguajes para cada microservicio
Escalamiento	Requiere escalar la aplicación completa	Se puede escalar de forma individual instanciando los microservicios múltiples veces en caso de que se encuentre algún cuello de botella
Tipo de escalamiento	Vertical, se contratan mejores recursos de computación	Horizontal, se puede escalar instanciando múltiples veces uno o más microservicios de forma individual

Nota. Benchmarking realizado entre arquitectura de monolito y microservicios: arquitecturas que guiarán el diseño de la aplicación.

La Tabla 5 muestra el benchmarking de las arquitecturas a considerar para el desarrollo de la solución. Originalmente, todos los sistemas de la empresa se encuentran bajo una arquitectura monolítica. Para el desarrollo del sistema de cotizaciones y para el ecosistema de aplicaciones de Corporation en general, se optó por una migración a una arquitectura de microservicios para el backend y micro frontend para el frontend debido a la alta escalabilidad, disponibilidad y performance. Los microservicios permiten un crecimiento

horizontal, el cual se traduce en ahorro de costos en infraestructura, por otro lado, si en caso se necesita priorizar algún servicio en específico, es posible aislarlo y tener múltiples instancias de este, lo cual mejora de forma considerable la disponibilidad, además que permite implementar una recuperación a fallos efectiva. Por otro lado, los micro frontends, de forma similar a los microservicios, nos permiten tener distintas aplicaciones para cada funcionalidad en concreto, lo cual también permite un crecimiento horizontal y performance.

3.2 Diseño de la Solución

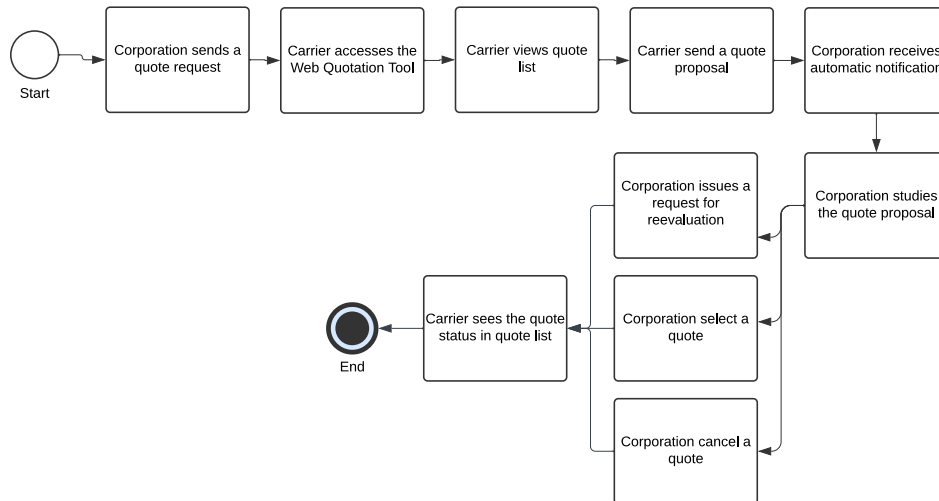
3.2.1 Arquitectura de Negocio

La Figura 8 muestra el flujo de procesos del negocio con respecto a las cotizaciones de servicios que la empresa de telecomunicaciones cliente, que desde ahora llamaremos “Corporation”, envía a otras empresas del mismo rubro, también llamados carriers.

El flujo del negocio inicia cuando Corporation desea adquirir un servicio de telecomunicaciones. Para ello, envía una solicitud de cotización a los carriers por medio del WQT (Web Quote Tool), sistema de gestión de cotizaciones propiedad de Corporation. Los carriers pueden visualizar la lista de solicitudes, que pueden ser filtradas por su ID de solicitud o por un estado específico: pendiente, respondida, preseleccionada, seleccionada, caducada y cancelada. Los carriers formalizan una cotización enviándola a Corporation por la misma herramienta. La empresa recibe una notificación automática para realizar una de tres acciones: reevaluar, preseleccionar o cancelar la cotización. Sea cual sea la acción emprendida por Corporation, los carriers pueden visualizar el status de la solicitud en la WQT.

Figura 8

Diagrama de negocio de Corporation

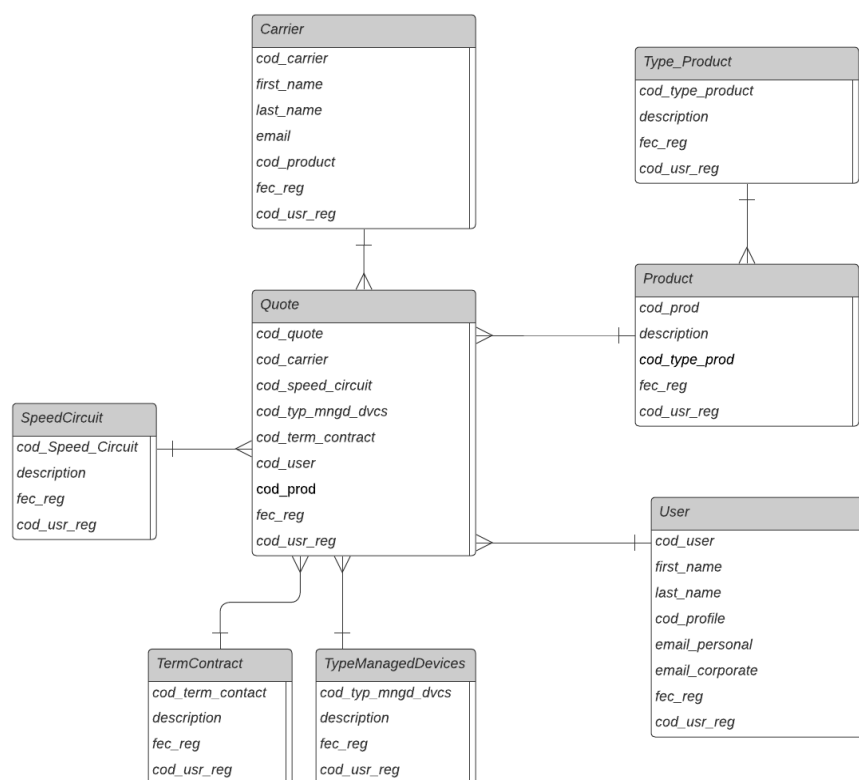


3.2.2 Arquitectura de Datos

La Figura 9 muestra la interacción de tablas, principalmente la relación con la tabla maestra Quote, la cual almacena información sobre las cotizaciones ingresadas mediante el sistema. Se observa que cada tabla adicional a sus campos específicos tiene dos campos adicionales: fec_reg que es la fecha de registro de la tupla y cod_usr_reg que es el código del usuario que realizó el registro, estos campos son importantes para realizar una auditoría a los datos.

Figura 9

Arquitectura de datos



La tabla 6 muestra la función de cada una de las tablas que conforman la arquitectura de datos del sistema de cotizaciones.

Tabla 6

Función de las tablas del diagrama de datos

Tabla	Función
User	Usuario de Corporation
Carrier	Representantes de otras empresas de telecomunicaciones
SpeedCircuit	Velocidad de conexión de servicio de telecomunicaciones
TermContract	Detalle del tiempo de contrato
TypeManagedDevice	Registro del tipo de instalación (los servicios pueden ser proveídos por diferentes empresas)
Product	Detalle del producto
Type_Product	Tipo de servicio (Internet, ClearChannel, MPLS, etc.)

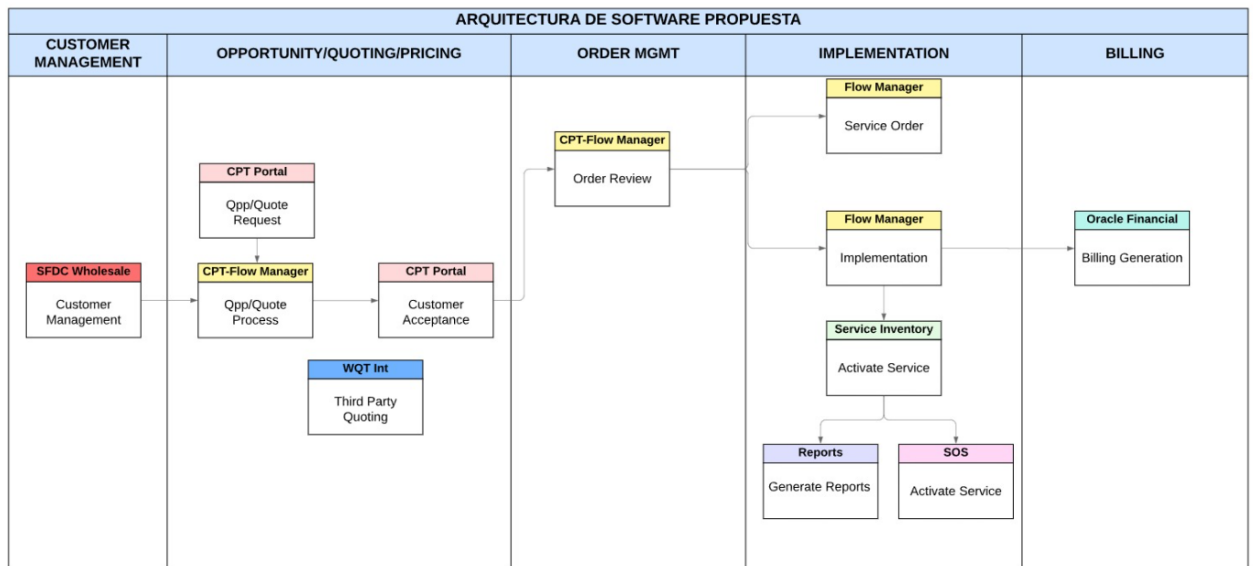
Quote	Detalle de la cotización
-------	--------------------------

3.2.3 Arquitectura de Aplicaciones

La Figura 10 describe la arquitectura de aplicaciones desde el punto de vista del ciclo de un servicio o producto. Este inicia con la gestión de clientes, donde el sistema Wholesale facilita la captación y gestión de clientes. Los sistemas Opportunity, permiten persuadir a los clientes para generar una primera cotización. La tercera fase es la aceptación de la cotización, en la que se reenvían ofertas hasta la aceptación final. Una vez que la cotización finaliza, se convierte en una orden. La fase de implementación es la instalación y seguimiento del servicio. Por último, la fase de Billing es la facturación del servicio.

Figura 10

Arquitectura de Aplicaciones



La tabla 7 define cada sistema que compone el ecosistema de aplicaciones de Corporation, sistemas referenciados en la arquitectura de aplicaciones.

Tabla 7

Ecosistema de aplicaciones de Corporation

Sistema	Definición
---------	------------

SFDC Wholesale	Sistema encargado de gestionar clientes, cumple la función de proporcionar información para contactar clientes, es un CRM en Salesforce, que permite una atención efectiva.
CPT Portal- Quote Request	Sistema desarrollado en Java 11, encargado de la iniciación de una cotización (fase 1 de cotización), se describe a alto nivel lo requerido.
CPT Flow Manager - Quote process	Sistema desarrollado en Java 17, encargado del proceso detallado de la cotización, se define características específicas del servicio.
CPT Portal - Customer Acceptance	Sistema desarrollado en Java 11, encargado de la aceptación de una cotización, se encarga del proceso de conversión de una cotización en una orden o pedido de servicio.
CPT Flow Manager - Order Review	Sistema desarrollado en Java 17, encargado de la revisión y aceptación de la orden por parte de los supervisores de la empresa, cada orden es revisada de forma individual por uno o más supervisores.
Flow Manager - Service Order	Sistema desarrollado en Java 17, encargado de la gestión a alto nivel de las órdenes, permite visualizar el estado de las órdenes en tiempo real y a los encargados de esta.
Flow Manager - Implementation	Sistema desarrollado en Java 17, encargado de realizar un seguimiento detallado a la implementación de la orden, muestra características específicas como velocidad de conexión, tipo de conexión, equipos a usar, etc.
Oracle Financial	Sistema desarrollado en Java 17, encargado de la facturación de los distintos servicios.

3.2.4 Arquitectura Tecnológica

La Figura 11 refleja el diseño estructural del sistema de cotizaciones y la interacción de los componentes tecnológicos que la conforman.

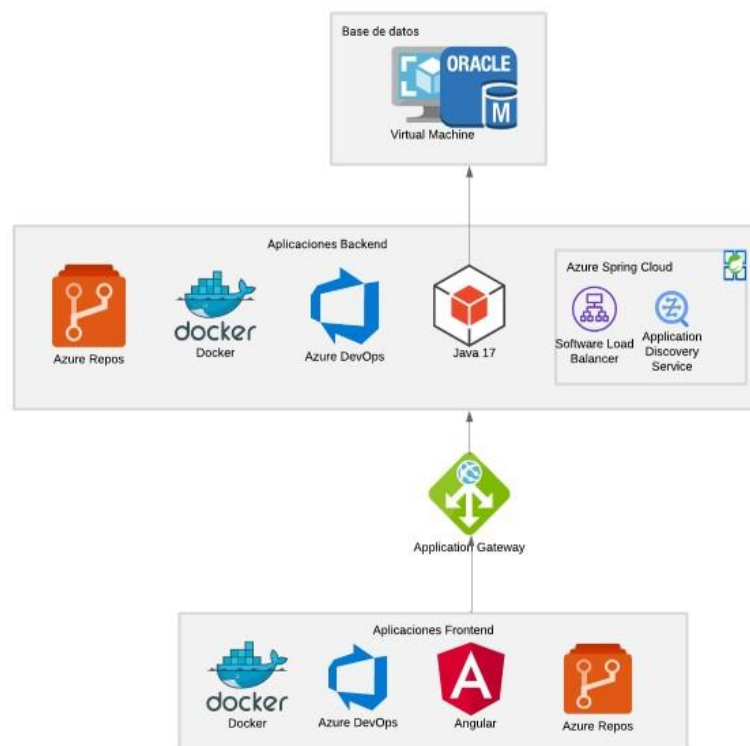
El frontend está bajo la arquitectura de Micro frontends, cuyo propósito es dividir un monolito en diversos micro proyectos de Angular 15. Cada proyecto tiene su propio repositorio y están alojados en la nube de Azure (Azure Repos), y se usa Gitflow como

modelo de flujo para la gestión de los repositorios. Cada uno de los proyectos emplea dockers en Azure Kubernetes Services (AKS) para su despliegue en producción, y Azure Container Instances (ACI) para el ambiente de desarrollo y QA. Las tareas relacionadas a frontend son gestionadas mediante Azure DevOps, con la que también se implementan flujos de trabajo para la entrega continua o delivery continuo (CD). El frontend se comunica con una API Gateway para poder consumir microservicios (backend).

El backend está bajo la arquitectura de microservicios, el cual consiste en dividir un proyecto extenso con funcionalidades complejas en pequeños proyectos, para ello se usó Java 17. La arquitectura se divide en microservicios consumidores y productores, los consumidores son a los que indirectamente consulta el frontend, cada uno de estos implementa la librería Ribbon como software para el balanceo de carga. Por otro lado, existe una aplicación Eureka Server para el registro de cada microservicio que permite el monitoreo del estado de salud de cada uno de ellos. Para el despliegue se usa docker en Azure (AKS) y los repositorios se encuentran alojados en Azure Repos. Las tareas relacionadas al backend se gestionan mediante Azure DevOps.

Figura 11

Arquitectura Tecnológica

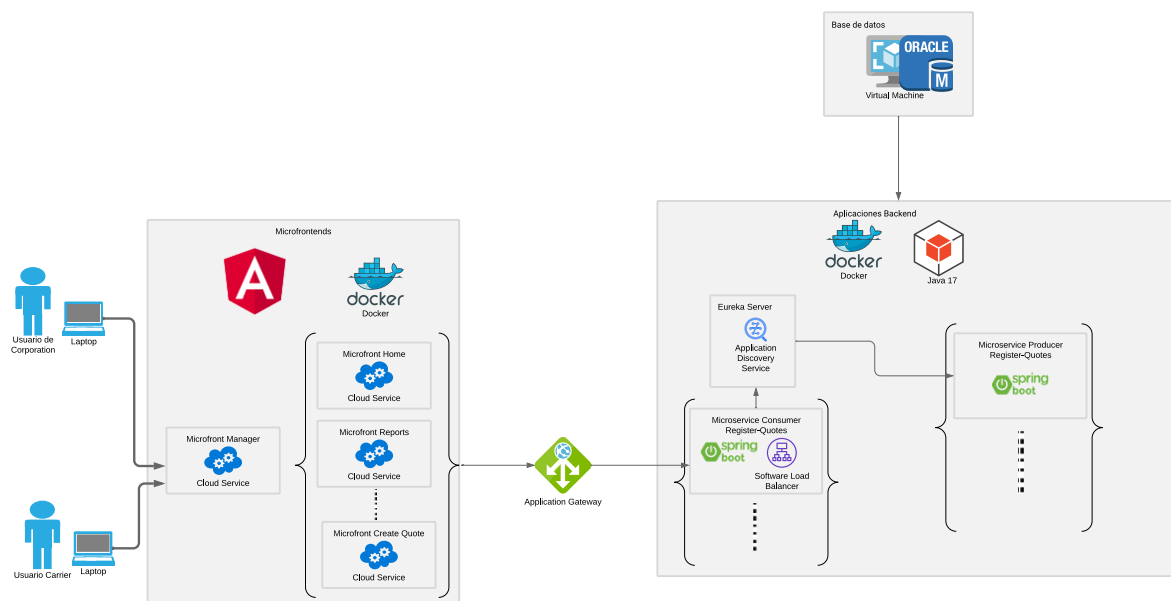


3.2.5 Arquitectura Integrada

La Figura 12 muestra la arquitectura integrada, en la que se refleja la interacción de componentes del sistema. Al entrar en la web de cotizaciones, los usuarios solicitan al servidor Frontend el micro frontend Manager que funciona como una plantilla compartida en varias páginas y que se encarga de renderizar el contenido de los demás micro frontends. Cuando el frontend realiza una solicitud al backend, recurre a un Api Gateway que permite el routing hacia un microservicio consumer, el cual se encarga del balanceo mediante LoadBalance. Este a su vez se conecta con Eureka para ver el listado de instancias disponibles y seleccionar una para atender la solicitud.

Figura 12

Arquitectura Integrada



3.3 Desarrollo de la Solución

Esta sección muestra el As-Is y el To-Be del sistema de cotizaciones para la empresa de telecomunicaciones Corporation. El As-Is fue desarrollado con JSP y JDeveloper, mientras que el To-Be en Angular 15, permitiendo un diseño más actualizado y de fácil implementación y mantenimiento.

Login

La figura 13 y la figura 14 muestran la pantalla de inicio de sesión para acceder a la aplicación WQTool

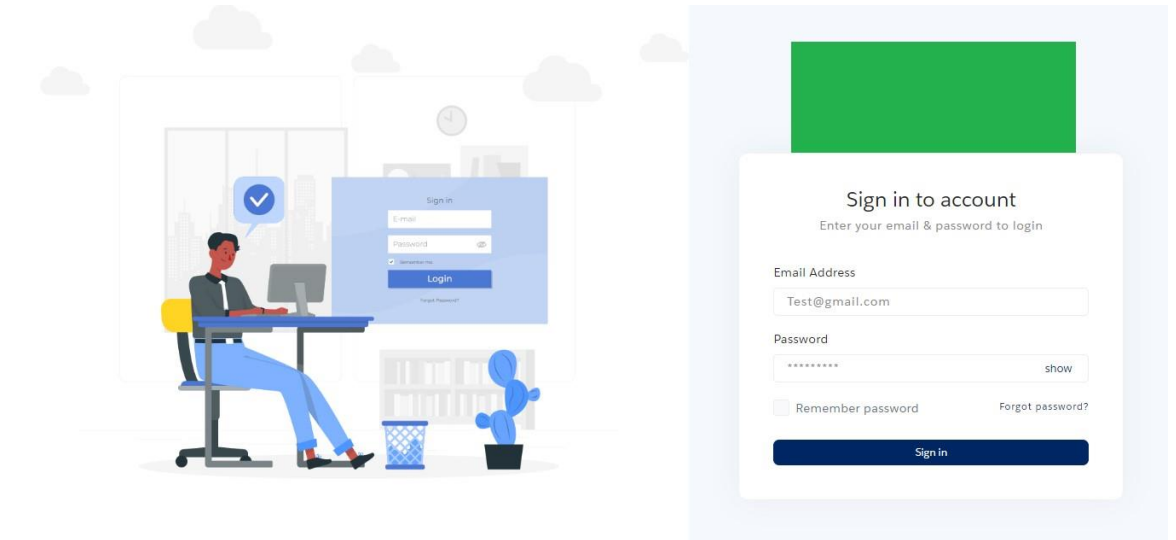
Figura 13

Login As-Is



Figura 14

Login To-Be



Bandeja

Esta sección muestra el listado de las cotizaciones. La figura 15 y la figura 16 muestran el detalle consta del tipo de producto ofrecido, el estado (pendiente, atendido, etc.), fecha de

registro y la compañía asociada. Las cotizaciones son realizadas por la empresa Corporation y son atendidas por los carriers, que son empresas externas.

Figura 15

Bandeja As-Is

WELCOME:JOAN MOSCOSO | USER GUIDE | LOGOUT | HOME

TRANSACTION

REPORT

MANAGEMENT

List of Quotes | < BACK

TYPE <--Show All--> ID FM ID QUOTE ID

Request ID Status <--Show All--> User <Show All> Search New Requirement

View Initial Requirement Compare Prices View Log of Emails Previous 1:10 of 703 Next 10

SELECT	REQUEST ID	PGEC ID	PGL ID	PM ID	QUOTE ID	STATUS	DATE	PRODUCT	COMPANY A	USER
<input checked="" type="radio"/>	0000036817			333	Quo3	Pending Quotes	Nov 5, 2023	MPLS/NNI	Company333	
<input type="radio"/>	0000036816			51	Quote53	Pending Quotes	Nov 3, 2023	CLEAR CHANNEL	Company52	JMOSCOSO_JT
<input type="radio"/>	0000036815			1	1	Pending Quotes	Nov 2, 2023	INTERNET	1	
<input type="radio"/>	0000036814			1	Q-3	Pending Quotes	Oct 31, 2023	INTERNET	Company2	
<input type="radio"/>	0000036813			22	Q-44	Pending Quotes	Oct 31, 2023	MPLS/NNI	Company33	
<input type="radio"/>	0000036812			143	Q-143	Pending Quotes	Oct 25, 2023	MPLS/NNI	COMPANY TEST	
<input type="radio"/>	0000036811			433	Q-3456	Pending Quotes	Oct 25, 2023	MPLS/NNI	COMP	
<input type="radio"/>	0000036810			123	Q-651	Pending Quotes	Oct 25, 2023	MPLS/NNI	COMP TEST	
<input type="radio"/>	0000036809			253	Q-123	Pending Quotes	Oct 25, 2023	MPLS/NNI	NEW COMPANY	
<input type="radio"/>	0000036808			175	Q-651	Pending Quotes	Oct 25, 2023	MPLS/NNI	NEW COM	

Figura 16

Bandeja To-Be

Ivan Gotelli Admin

List Of Quotes

Type ID FM Id Quote Id Request Id

Status User Search New Requirement

Request Id	FM Id	Quote Id	Status	Date	Product	Company A	User	Actions
36817	333	Quo3	Pending Quotes	2023-11-05 19:01:10	MPLS/NNI	Company333		
36816	51	Quote53	Pending Quotes	2023-11-03 15:05:56	CLEAR CHANNEL	Company52	JMOSCOSO_JT	
36815	1	1	Pending Quotes	2023-11-02 09:49:20	INTERNET	1		
36814	1	Q-3	Pending Quotes	2023-10-31 23:58:40	INTERNET	Company2		
36813	22	Q-44	Pending Quotes	2023-10-31 01:42:42	MPLS/NNI	Company33		
36812	143	Q-143	Pending Quotes	2023-10-25 15:51:10	MPLS/NNI	COMPANY TEST		

Comparaciones de precios

La figura 17 y la figura 18 muestran la comparación de precios se da en base de una cotización previa realizada por la empresa Corporation. Estos precios son mostrados en base

a los tipos de productos, este apartado es específicamente realizado para que los externos puedan tener una apreciación detallada de la cotización.

Figura 17

Comparación de precio As-Is

WELCOME:JOAN MOSCOSO | USER GUIDE | LOGOUT | HOME

						TRANSACTION	REPORT	MANAGEMENT			
Compare Prices < BACK											
Export Excel											
Previous 1-10 of 26 Next 10											
PRODUCT	LOCATION A	LOCATION B	CARRIER	TERM	BW	PRIMARY ACCESS LINK		CROSS CONNECT MRC	CROSS CONNECT OTC	TOTAL MRC COST	TOTAL OTC COST
						MRC	OTC				
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	4 YEARS	3 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	7 YEARS	3 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	2 YEARS	3 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	EXPEREO USA, INC.	4 YEARS	3 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	EXPEREO USA, INC.	7 YEARS	3 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	EXPEREO USA, INC.	2 YEARS	3 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	4 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	7 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	2 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	EXPEREO USA, INC.	4 YEARS	1 MBPS						

Figura 18

Comparación de precio To-Be

Emmy Walter Admin

←

Compare Prices

Export to Excel

Product	Location A	Location B	Carrier	Term	BW	Primary Access Link		Cross Connect MRC	Cross Connect OTC	Total MRC Cost	Total OTC Cost
						MRC	OTC				
CLEAR CHANNEL	****	Mexico,Chiapas,Tapachula., CRUCE FRONTERIZO MEXICO .	EXPEREO USA, INC.	6 YEARS	768 KBPS						
CLEAR CHANNEL	****	Mexico,Chiapas,Tapachula., CRUCE FRONTERIZO MEXICO .	EXPEREO USA, INC.	4 YEARS	768 KBPS						
CLEAR CHANNEL	****	Mexico,Chiapas,Tapachula., CRUCE FRONTERIZO MEXICO .	DIGICEL USA, INC.	6 YEARS	768 KBPS						
CLEAR CHANNEL	****	Mexico,Chiapas,Tapachula., CRUCE FRONTERIZO MEXICO .	DIGICEL USA, INC.	4 YEARS	768 KBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	EXPEREO USA, INC.	2 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	EXPEREO USA, INC.	7 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	EXPEREO USA, INC.	4 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	2 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	7 YEARS	1 MBPS						
CLEAR CHANNEL	****	Mexico,Chihuahua,Juarez.,CL. CENTRAL COPERNICO DE LA LABOR 6730 FRANCCIONAMIENTO EL CRUCERO	IG NETWORKS	4 YEARS	1 MBPS						

Detalle de cotización

Las figuras 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 muestran los formularios de detalle de cotización en WQTool, y que puede ser visualizado por los usuarios internos y externos. Es el input inicial de la cotización. Los 3 tipos de productos ofrecidos son:

MPLS (Multiprotocolo Label Switching): Es una técnica que permite unificar distintos tipos de datos a través de una misma red con el fin de superar limitaciones de velocidad.

Internet: Recurso que conecta dispositivos y redes en todo el mundo, permitiendo la transferencia de datos y comunicación entre usuarios.

Clear channel: Este producto permite que los datos se transmitan de un sitio remoto a un central por medio de un enlace dedicado para datos de punto a punto.

Figura 19

Detalle de cotización: Clear Channel As-Is Parte 1

WELCOME:JOAN MOSCOSO | USER GUIDE | LOGOUT | HOME

TRANSACTIONREPORTMANAGEMENT

Initial Requirement | BACK

Location A

Request ID0000036816

ProductCLEAR CHANNEL

PGEC ID51

FM IDQuote53

PGEC Quote

Request DateNov 3, 2023

CompanyCompany52

End Customer

Description55

AddressAddress54

Floor56

SuiteSuite57

CityCity58

Zip/Postal Codezip510

CountryCountry511

Customer Phone Number512

Product

CSU/DSUYes

CPEYes

PROVIDED BY OTHER THAN THE CARRIER

File Selected

Speed of Circuit

1 MBPS

4 MBPS

3 MBPS

Locations B

Select andRefresh

SELECT	CUSTOMER NAME	BRANCH NAME	FLOOR	SUITE	CITY	STATE	POSTAL CODE
<input checked="" type="radio"/>	CLARO ENTERPRISE SOLUTIONS, LLC.	CIUDAD JUAREZ-EL PASO-BC		FRANCCIONAMIENTO EL CRUCERO	Juarez	Chihuahua	33027
<input type="radio"/>	CLARO ENTERPRISE SOLUTIONS, LLC.	TAPACHULA-BC			Tapachula	Chiapas	

Target MRC15,142.00

Target OTC15,152.00

Configuration TypeLINEAR

Connecting Facility AssignmentYes

Carrier to provide CFA / LOACarrier 52222

Clear Channel Access Configuration – Linear

Without Access Resiliency. Carrier must provide the quote for a linear Access from Location A premises to Telmex's POP

Figura 20

Detalle de cotización: Clear Channel As-Is Parte 2

Request ID 0000036816
Product CLEAR CHANNEL
PGEC ID 51
FM ID 51
Quote ID Quote53
Request Date Nov 3, 2023
Company Company52
End Customer Description55
Address Address54
Floor 56
Suite Suite57
City City58
Zip/Postal Code zip510
Country Country511
Customer Phone Number 512

Select and Refresh

SELECT	CUSTOMER NAME	BRANCH NAME	FLOOR	SUITE	CITY	STATE	POSTAL CODE
<input checked="" type="radio"/>	CLARO ENTERPRISE SOLUTIONS, LLC.	CIUDAD JUAREZ-EL PASO-BC		FRANCCIONAMIENTO EL CRUCERO	Juarez	Chihuahua	33027
<input type="radio"/>	CLARO ENTERPRISE SOLUTIONS, LLC.	TAPACHULA-BC			Tapachula	Chiapas	

Product

CSU/DSU Yes
CPE Yes
PROVIDED BY OTHER THAN THE CARRIER File Selected
Speed of Circuit
1 MBPS
4 MBPS
3 MBPS
4 YEARS
7 YEARS
2 YEARS
Term of Contract
Carrier
EXPEREO U SA, INC.
IG NETWORKS
Description
Description5132

Target MRC 15,142.00
Target OTC 15,152.00
Configuration Type LINEAR
Connecting Facility Assignment Yes
Carrier to provide CFA / LOA Carrier 52222

Clear Channel Access Configuration – Linear

Without Access Resiliency. Carrier must provide the quote for a linear Access from Location A premises to Telmex's POP

Figura 21

Detalle de cotización: Clear Channel To-Be Parte 1

Initial Requirement

Location A

Request Id 36817
Product MPLS/NNI
PGEC ID
FM ID 333
Quote ID Quo3
PGEC Quote
Request Date 2023-11-05 19:01:10
Company Company333
End Customer
Description Description223
Address Address2
Floor 334
Suite 34
City 123
Zip Code 33
Country 3434
Customer Phone 3343

Location B

Description NNI Interconnection
Target MRC 222
Target OTC 2222
Configuration Type LINEAR
Connecting Facility Assignment No
Carrier to provide CFA / LOA

Product

CSU/DSU No
CPE No
Requirements REQUIRED BY THIRD PARTIES
Configuration Type STANDARD CONFIGURATION
Configuration ID Selected 31 - (CISCO1941/K9 - (1)T1, (2)10/100/1000, Security)

MPLS Access Configuration – Linear

Without Access Resiliency. Carrier must provide a MPLS quote with a linear Access from Location A premises to MPLS-NNI interconnection point.

Figura 22

Detalle de cotización: Clear Channel To-Be Parte 2

City56zip510Country511Customer Phone512

Product

CSU/DSUYes

RequirementsPROVIDED BY OTHER THAN THE CARRIER

CPEYes

Configuration TypeNON STANDARD CONFIGURATION

File selected : claro.png

Speed of Circuit

3 MBPS

4 MBPS

1 MBPS

Term of Contract

4 YEARS

2 YEARS

7 YEARS

Carrier

EXPEREO USA, INC.

IG NETWORKS

Description5132

Target MRC15142

Configuration TypeLINEAR

Carrier to provide CFA / LOACarrier 52222

Target OTC15152

Connecting Facility AssignmentYes

Clear Channel Access Configuration – Linear

Without Access Resiliency, Carrier must provide the quote for a linear Access from Location A premises to Telmex's POP

Location A

Customer Premises

Access

POP

Carrier Transport Network

PVC

Network Interconnection Point

Location Z

Network

Figura 23

Detalle de cotización: Internet As-Is Parte 1

WELCOME:JOAN MOSCOSO | USER GUIDE | LOGOUT | HOME

TRANSACTION

REPORT

MANAGEMENT

Initial Requirement | < BACK

Location A

Request ID000036814

ProductINTERNET

PGEC ID1

FM IDQ-3

PGEC Quote

Request DateOct 31, 2023

CompanyCompany2

End Customer

Description5

AddressAddress4

Floor6

SuiteSuite7

CityCity8

Zip/Postal Code10

CountryCountry11

Customer Phone Number12

Location B

DescriptionInternet Cloud

Product

CSU/DSUYes

CPEYes

RequirementsPROVIDED BY CLARO ENTERPRISE SOLUTIONS

NON STANDARD CONFIGURATION

Speed of Circuit

2 MBPS

4 MBPS

11 YEARS

Target MRC20.00

Target OTC21.00

Configuration TypeRING PROTECTED

Connecting Facility AssignmentYes

Carrier to provide CFA / LOACarrier to provide 22

Internet Access Configuration – Linear

Without Access Resiliency, Carrier must provide the quote for a ring Protected access solution ro its internet network.

Capacity has to be protected in the Carrier's backbone as well.

33

Figura 24

Detalle de cotización: Internet As-Is Parte 2

Location A

Request ID000036814

ProductINTERNET

PGECD

FM ID1

Quote IDQ-3

PGECD

Request DateOct 31, 2023

CompanyCompany2

End Customer

DescriptionDescription5

AddressAddress4

Floor6

SuiteSuite7

CityCity8

Zip/Postal Code10

CountryCountry11

Customer Phone Number12

Product

CSU/DSUYes

CPEYes

PROVIDED BY CLARO ENTERPRISE SOLUTIONS

NON STANDARD CONFIGURATION

Speed of Circuit

2 MBPS

4 MBPS

11 YEARS

5 YEARS

Term of Contract

11 YEARS

5 YEARS

Carrier

DIGICEL U SA, INC. comm DIGICEL U SA

China Telecom comm China Telecom

Description

Description18

Location B

DescriptionInternet Cloud

Target MRC20.00

Target OTC21.00

Configuration TypeRING PROTECTED

Connecting Facility AssignmentYes

Carrier to provide CFA / LOACarrier to provide 22

Internet Access Configuration – Linear

Without Access Resiliency. Carrier must provide the quote for a ring Protected access solution ro its internet network.

Capacity has to be protected in the Carrier's backbone as well.

Location A

Customer Premises

Ring Access

POP Provider

Location Z

Carrier Internet Network

Figura 25

Detalle de cotización: Internet To-Be Parte 1

Initial Requirement

Location A

Request Id36814

ProductINTERNET

PGECD

FM ID1

Quote IDQ-3

PGECD

Request Date2023-10-31 23:58:40

CompanyCompany2

End Customer

DescriptionDescription5

AddressAddress4

Floor6

SuiteSuite7

CityCity8

Zip Code10

CountryCountry11

Customer Phone12

Product

CSU/DSUYes

CPEYes

RequirementsPROVIDED BY CLARO ENTERPRISE SOLUTIONS

Configuration TypeNON STANDARD CONFIGURATION

File selected :

Location B

DescriptionInternet Cloud

Target MRC20

Target OTC21

Configuration TypeRING PROTECTED

Connecting Facility AssignmentYes

Carrier to provide CFA / LOACarrier to provide 22

Internet Access Configuration – Linear

Without Access Resiliency. Carrier must provide the quote for a ring Protected access solution ro its internet network.

Capacity has to be protected in the Carrier's backbone as well.

Location A

Customer

POP

Location Z

Figura 26

Detalle de cotización: Internet To-Be Parte 2

Address4

City6

Zip Code

10

6

Country11

Suite7

Customer Phone

12

Carrier to provide CFA / LOA

Carrier to provide 22

Product

CSU/DSU

Yes

CPE

Yes

Requirements

PROVIDED BY CLARO ENTERPRISE SOLUTIONS

Configuration Type

NON STANDARD CONFIGURATION

File selected :

Speed of Circuit

4 MBPS

2 MBPS

Term of Contract

5 YEARS

11 YEARS

Carrier

DIGICEL USA, INC.

comm DIGICEL USA

China Telecom

comm China Telecom

Description

Description18

Internet Access Configuration – Linear

Without Access Resiliency. Carrier must provide the quote for a ring Protected access solution ro its internet network. Capacity has to be protected in the Carrier's backbone as well.

Location A

Customer Premises

Ring Access

POP Provider

Location Z

Carrier Internet Network

Figura 27

Detalle de cotización: MPLS As-Is Parte 1

WELCOME:JOAN MOSCOSO | USER GUIDE | LOGOUT | HOME

TRANSACTION

REPORT

MANAGEMENT

Initial Requirement | < BACK

Location A

Request ID

0000036817

Product

MPLS/NNI

PGEC ID

FM ID

333

Quote ID

Quo3

PGEC Quote

Request Date

Nov 5, 2023

Company

Company333

End Customer

Description

Description223

Address

Address2

Floor

334

Suite

34

City

123

Zip/Postal Code

33

Country

3434

Customer Phone Number

3343

Location B

Description

NNI Interconnection

Product

CSU/DSU

No

CPE

No

Configuration ID Selected 1 - (CISCO1941/K9 - (1)T1, (2)10/100/1000, Security)

Configuration MPLS

SELECT	CARRIER	BANDWITH ACCESS	PORT	PROFILE	PACKAGE
<input checked="" type="radio"/>	CABLE & WIRELESS PANAMA, S.A.	128 KBPS	256 KBPS	Profile C&W	Pa02

MPLS Access Configuration – Linear

Without Access Resiliency. Carrier must provide a MPLS quote with a linear Access from Location A premises to MPLS-NNI interconnection point.

Location A

Location Z

Target MRC

222.00

Target OTC

2,222.00

Configuration Type

LINEAR

Connecting Facility Assignment

No

35

Figura 28

Detalle de cotización: MPLS As-Is Parte 2

Initial Requirement | < BACK

Location A

Request ID000036817

ProductMPLS/NNI

PGEC ID333

FM ID333

Quote IDQuo3

PGEC Quote

Request DateNov 5, 2023

CompanyCompany333

End Customer

DescriptionDescription223

AddressAddress2

Floor334

Suite34

City123

Zip/Postal Code33

Country3434

Customer Phone Number3343

Product

CSU/DSUNo

CPENo

Configuration ID Selected 1 - (CISCO1941/K9 - (1)T1, (2)10/100/1000, Security)

Configuration MPLS

SELECT	CARRIER	BANDWITH ACCESS	PORT	PROFILE	PACKAGE
<input checked="" type="radio"/>	CABLE & WIRELESS PANAMA, S.A.	128 KBPS	256 KBPS	Profile C&W	Pa02
<input type="radio"/>	VERIZON	128 KBPS	192 KBPS	Profile Verizon	Pa03_G15

Configuration MPLS Detalle

COS	BW
Min	max

Location B

DescriptionNNI Interconnection

Target MRC222.00

Target OTC2,222.00

Configuration TypeLINEAR

Connecting Facility AssignmentNo

MPLS Access Configuration – Linear

Without Access Resiliency. Carrier must provide a MPLS quote with a linear Access from Location A premises to MPLS-NNI interconnection point.

Location A

Customer Premises

Access

MPLS POP

Carrier MPLS Network

Location Z

Network Interconnection Point

Network

américa móvil

Figura 29

Detalle de cotización: MPLS To-Be Parte 1

Initial Requirement

Location A

Request Id36817

ProductMPLS/NNI

PGEC ID

FM ID333

Quote IDQuo3

PGEC Quote

Request Date2023-11-05 19:01:10

CompanyCompany333

End Customer

DescriptionDescription223

AddressAddress2

Floor334

Suite34

City123

Zip Code33

Country3434

Customer Phone3343

Product

CSU/DSUNo

CPENo

RequirementsREQUIRED BY THIRD PARTIES

Configuration TypeSTANDARD CONFIGURATION

Configuration ID Selected 31 - (CISCO1941/K9 - (1)T1, (2)10/100/1000, Security)

Location B

DescriptionNNI Interconnection

Target MRC222

Target OTC2222

Configuration TypeLINEAR

Connecting Facility AssignmentNo

Carrier to provide CFA / LOA

MPLS Access Configuration – Linear

Without Access Resiliency. Carrier must provide a MPLS quote with a linear Access from Location A premises to MPLS-NNI interconnection point.

Location A

Location Z

Figura 30

Detalle de cotización: MPLS To-Be Parte 2

Product

CSU/DSU
No

CPE
No

Requirements
REQUIRED BY THIRD PARTIES

Configuration Type
STANDARD CONFIGURATION

Configuration ID Selected 31 - (CISCO1941/K9 - (1)T1, (2)10/100/1000, Security)

Configuration MPLS

Select	Carrier	Bandwidth Access	Port	Profile	Package
<input type="radio"/>	CABLE & WIRELESS PANAMA, S.A.	128 KBPS	256 KBPS	Pa02	Profile C&W
<input type="radio"/>	VERIZON	128 KBPS	192 KBPS	Pa03_G15	Profile Verizon

Detail MPLS Configuration

COS

BW

Term of Contract

4 YEARS

Description

Description2233

MPLS Access Configuration – Linear

Without Access Resiliency, Carrier must provide a MPLS quote with a linear Access from Location A premises to MPLS-NNI interconnection point.

```
graph LR
    A[Customer Premises Location A] -- Access --> B[MPLS POP]
    B --- C[Carrier MPLS Network]
    C --> D[Network Interconnection Point Location Z]
    D --- E[Network]
```

3.4 Resultado del Proyecto

La construcción del sistema de gestión de cotizaciones de servicios para empresas de telecomunicaciones a una arquitectura basada en microservicios y micro frontends ha generado una serie de mejoras significativas en varios aspectos clave del sistema. Estas mejoras se han traducido en un rendimiento mejorado y una mayor eficiencia en la gestión de cotizaciones. A continuación, definimos los resultados más relevantes de esta migración:

Escalabilidad mejorada: Antes de la migración, el sistema enfrentaba limitaciones en cuanto a la escalabilidad, no estaba en la capacidad de gestionar un alto volumen de cotizaciones simultáneas. Con la adopción de microservicios, el sistema ahora puede escalar de forma horizontal, el cual, con una buena gestión de recursos permite ahorro en costos de infraestructura, sin degradar la performance.

Mayor disponibilidad y tolerancia a fallos: Los microservicios han permitido separar el sistema en distintos fragmentos de servicios encargados de una funcionalidad específica, lo cual permite que, si uno de estos fragmentos deja de funcionar, los demás sigan en funcionamiento. Además de ello, los patrones aplicados como Circuit Breaker permitieron crear componentes para recuperarse ante fallos, de modo que, si un microservicio falla, se

37

use una copia de respaldo. Como resultado, el sistema tiene alta disponibilidad, es resiliente y tolerante a fallos, tiene alto desacoplamiento y baja cohesión.

Tiempo de respuesta reducido: Con la implementación de micro frontends de la mano con microservicios, se logró una experiencia de usuario más fluida y tiempos de carga más cortos. Los usuarios experimentan una respuesta más rápida a las solicitudes de cotización, lo que mejora la satisfacción del cliente.

Flexibilidad en el desarrollo y despliegue: La arquitectura de microservicios permite el desarrollo y despliegue independiente de componentes, lo que ha acelerado el ciclo de desarrollo y ha facilitado la adopción de nuevas funcionalidades y actualizaciones sin afectar al sistema en su conjunto.

Mantenimiento simplificado: La granularidad de los microservicios ha simplificado el mantenimiento y la corrección de errores, ya que es más fácil identificar y resolver problemas específicos.

Métricas de Rendimiento (As-Is vs. To-Be)

La Tabla 8 presenta la comparación de métricas con respecto al rendimiento del sistema en su etapa previa a la migración y después de realizarla.

Tabla 8

Comparación de métricas As-Is y To-Be

Métrica	As-Is	To-Be
Tiempo promedio de respuesta (ms)	1500	1000
Disponibilidad (%)	90	99.9
Tasa de errores críticos por mes	5	< 1
Escalabilidad	Limitada	Alta
Costos de infraestructura	Altos	Optimizados
Velocidad de desarrollo de nuevos requerimientos	Bajo	Alto
Recuperación ante fallos	Bajo	Alto
Nivel de seguridad	Baja (Datos en sesión)	Alta (OAuth y JWT)
Cobertura de código	0%	85%

Errores reportados por semana en producción	5	<1
---	---	----

Nota. Para la comparación se seleccionaron diversos atributos de calidad, haciendo un énfasis en el performance.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Para el objetivo 1, se realizó un benchmarking para la elección de Kafka como tecnología de mensajería, ya que prioriza el rendimiento de la aplicación, tiene una mayor consistencia de entregas (ACK) y repetición de mensajes. También se escogió microservicios con micro frontend como la arquitectura que guiará el diseño del sistema, gracias a su alta escalabilidad, disponibilidad y performance. Además, se establecieron las siguientes tecnologías para su desarrollo: Angular 15 para el frontend, Java 17 para el backend, Oracle para la conexión a la base de datos, Azure para servicios de informática en la nube y Docker para la contenedorización.

Para el cumplimiento del objetivo 2, se diseñaron las arquitecturas de negocio, de datos, de aplicaciones, tecnológica e integrada para asegurar la construcción de un sistema efectivo y acorde a las necesidades del cliente. Se optó por implementar una arquitectura de microservicios en el desarrollo del sistema, ya que su capacidad para descomponer la aplicación en módulos independientes y altamente especializados contribuye a la mejora del rendimiento global del sistema de cotizaciones.

Para el cumplimiento del objetivo 3, se realizó la migración del sistema de cotizaciones de Corporation (WQTool) en base a las tecnologías establecidas y bajo las arquitecturas desarrolladas. Se realizó una reestructuración visual de los formularios críticos del sistema, tales como la creación de cotizaciones o la descripción de características. Para ello, se priorizó la usabilidad y la eficiencia del sistema, lo que a su vez contribuirá a optimizar el proceso de cotización y brindar una experiencia más intuitiva para los usuarios.

Para el cumplimiento del objetivo 4, se validó la correcta funcionalidad de los principales procesos de negocio. Se analizaron diferentes métricas como el tiempo promedio de respuesta, disponibilidad, Tasa de errores críticos por mes, escalabilidad, costo de infraestructura, velocidad de desarrollo de nuevos requerimientos, recuperación ante fallos, nivel de seguridad, cobertura de código, y errores reportados por semana en producción.

Estas métricas fueron analizadas tanto en la configuración actual ("As-Is") como en la configuración propuesta ("To-be") del sistema, revelando una mejora significativa con respecto a su estado previo.

4.2 Recomendaciones

En base al objetivo y resultados del presente trabajo, se presentan las siguientes recomendaciones para mejorar aún más el sistema de cotizaciones en futuros desarrollos:

a. Múltiples instancias de microservicios

Se recomienda tener múltiples instancias de un mismo microservicio con el fin de mejorar aún más la disponibilidad.

b. Bases de datos espejo

Configuración donde más de dos servidores de bases de datos se ejecutan en equipos independientes con el fin de mantener copias de la base de datos, sobre todo al trabajar con información sensible y relevante.

c. Bases de datos de lectura

Con esta adición se puede mejorar los tiempos de respuesta al realizar operaciones de lectura. Al estar desnormalizadas, suelen tener índices para agilizar las búsquedas. Este concepto aplicado a microservicios es denominado Command Query Responsibility Segregation (CQRS).

d. Redis

Es una base de datos en memoria, es decir, los datos se guardan principalmente en la memoria RAM, a diferencia de guardarla en el disco duro, es que el tiempo de respuesta es mucho menor. El inconveniente es que al apagar el servidor los datos se perderán si solo han sido almacenados en RAM, por ello es que Redis permite dentro de sus configuraciones que los datos sean guardados en el disco duro para que persistan. La utilidad de Redis radica en usar parte de la memoria RAM como si fuera un disco duro para reducir los tiempos de respuesta en la lectura de datos. Con su implementación, el flujo de datos iniciará buscando en la base de datos Redis y si en caso no se encuentra el documento o registro, procede a buscar en el servidor de base

de datos. Lo ideal es almacenar en Redis datos que son frecuentemente consultados con el fin de no saturar a la base de datos.

5 REFERENCIAS

Aragon, H., Braganza, S., Boza, E., Parrales, J., & Abad, C. (2019, mayo 13-17). *Workload characterization of a software-as-a-service web application implemented with a microservices architecture* [Conferencia]. The 2019 World Wide Web Conference, San Francisco, USA. <https://doi.org/10.1145/3308560.3316466>

Atlassian. (2022, 15 de mayo). *Arquitectura de microservicios*. Atlassian. Recuperado el 13 de septiembre de 2023, de <https://www.atlassian.com/es/microservices/microservices-architecture>

Atlassian. (2022, 23 de junio). *Microservices vs. monolithic architecture*. Atlassian. Recuperado el 27 de septiembre de 2023, de <https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith#:~:text=A%20monolithic%20architecture%20is%20a%20singular%2C%20large%20computing%20network%20with,of%20the%20service%2Dside%20interface.>

Castillo, C., & Chahuara, P. (1970, 1 de enero). *Análisis del Estado de la Demanda de los servicios públicos*. OSIPTEL. Recuperado el 4 de octubre de 2023, de <https://ideas.repec.org/p/opt/doctra/43.html>

Citrix. (2021, 13 de junio). *¿Qué es un servicio cloud?* Citrix. Recuperado el 29 de septiembre de 2023, de <https://www.citrix.com/es-mx/solutions/digital-workspace/what-is-a-cloud-service.html>.

Docker. (2023, 8 de septiembre). *Docker Overview*. Docker Documentation. Recuperado el 13 de septiembre de 2023, de <https://docs.docker.com/get-started/overview/#:~:text=Docker%20architecture,to%20a%20remote%20Docker%20daemon.>

Gan, Y., Zhang, Y., Cheng, D., Shetty, A., Rath, P., Katarki, N., Bruno, A., Hu, J., Ritchken, B., Jackson, B., Hu, K., Pancholi, M., He, Y., Clancy, B., Colen, C.,

- Wen, F., Leung, C., Wang, S., Zaruvinisky, L., ... Delimitrou, C. (2019, abril 13-17). *An open-source benchmark suite for microservices and their hardware-software implications for Cloud & Edge Systems* [Conferencia]. Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Providence, USA. <https://doi.org/10.1145/3297858.3304013>
- Geers, M. (2020, 13 de abril). *Microfrontends*. Micro Frontends. Recuperado el 27 de septiembre de 2023, de <https://micro-frontends-es.org/>
- Google. (2022, 18 de noviembre). *Update Angular to v15*. Angular. Recuperado el 20 de septiembre de 2023, de <https://angular.io/guide/update-to-version-15>
- Harms, H., Rogowski, C., & Lo Iacono, L. (2017, septiembre 4-8). *Guidelines for adopting frontend architectures and patterns in microservices-based systems* [Conferencia]. 11th Joint Meeting on Foundations of Software Engineering, Paderborn, Alemania. <https://doi.org/10.1145/3106237.3117775>
- IBM. (2020, 22 de enero). *IAAS*. IBM. Recuperado el 20 de septiembre de 2023, de <https://www.ibm.com/cloud/learn/iaas>.
- Kafka. (2016, 9 de octubre). *Kafka Documentation*. Apache Kafka. Recuperado el 12 de octubre de 2023, de <https://kafka.apache.org/documentation/>
- Lahtela, M., & Kaplan, P. (2023, 9 de junio). *¿Cuál es la diferencia entre la arquitectura monolítica y la de microservicios?* Amazon. Recuperado el 30 de agosto de 2023, de <https://aws.amazon.com/es/compare/the-difference-between-monolithic-and-microservices-architecture/>
- Lemahieu, W., Snoeck, M., Goethals, F., De Backer, M., Haesen, R., Vandenbulcke, J., Dedene, G. (2005). Coordinating Cots applications via a business event layer. *IEEE Software*, 22(4), 28–35. <https://doi.org/10.1109/ms.2005.90>
- Lubis, M., Fathoni, M., & Lubis, A. R. (2020, julio 17-19). *New Product Development Architectural Framework for sustainability and innovation within telecommunication industry* [Conferencia]. 8th International Conference on Computer and Communications Management, Singapur, Singapur. <https://doi.org/10.1145/3411174.3411197>

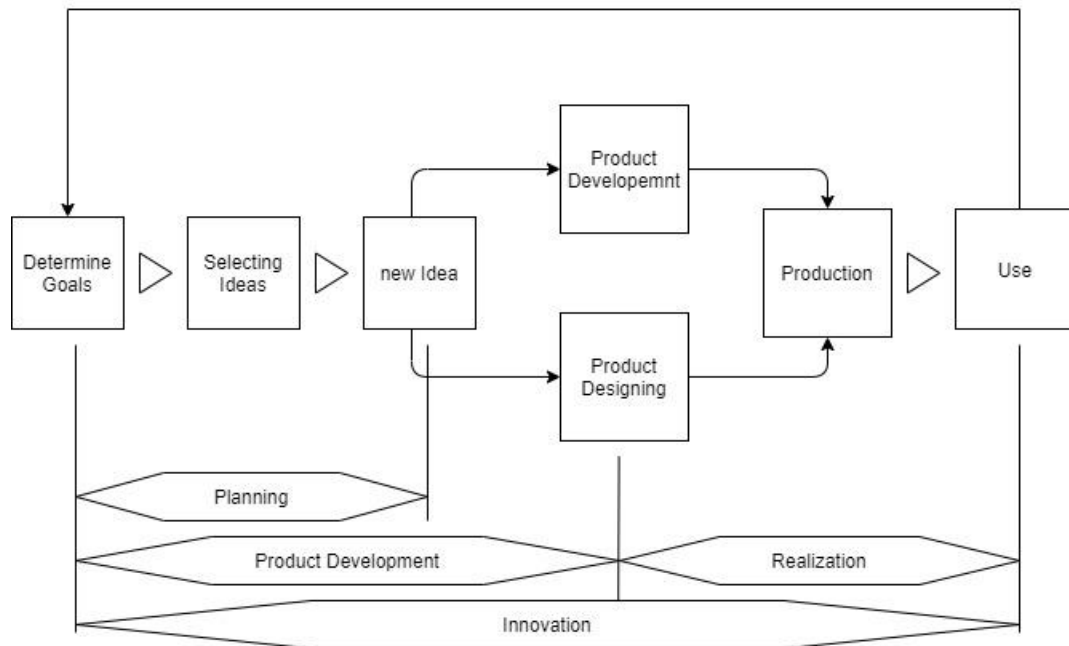
- Oracle. (2020, 5 de abril). *¿Qué es SaaS?* Oracle. Recuperado el 29 de octubre de 2023, de <https://www.oracle.com/mx/applications/what-is-saas/>.
- RabbitMQ. (2007, 9 de febrero). *Rabbit MQ Documentation*. RabbitMQ. Recuperado el 11 de octubre de 2023, de <https://www.rabbitmq.com/documentation.html>
- Rancher. (2021, 6 de diciembre). *Microservices vs. Monolithic Architectures*. SUSE Communities. Recuperado el 9 de octubre de 2023, de https://www.suse.com/c/rancher_blog/microservices-vs-monolithic-architectures/
- Red Hat. (2020, 3 de febrero). *What are cloud services?* Red Hat. Recuperado el 12 de octubre de 2023, de <https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services>.
- Red Hat. (2019, 30 de octubre). *What is PaaS?* Redhat. Recuperado el 12 de octubre de 2023, de <https://www.redhat.com/en/topics/cloud-computing/what-is-paas#benefits-of-paas>.
- Spring. (2023, 10 de febrero). *Spring Boot 3.1.4*. Spring Boot. Recuperado el 24 de agosto, de <https://spring.io/projects/spring-boot>
- Viveros, L. J., Rodríguez, J. C., & Sarmiento, D. A. (2019, diciembre 18-20). *Design of a multicriterio model with a diffuse hierarchical analysis process - FAHP, for the selection of suppliers in a mobile cellular telecommunications company* [Conferencia]. 2nd International Conference on Education Technology Management, Barcelona, España. <https://doi.org/10.1145/3375900.3375904>
- Zhang, Y., Zhang, Y., Wu, Y., Lu, Y., Wang, T., & Mao, X. (2020, mayo 12-14). *Exploring the dependency network of Docker containers: Structure, diversity, and relationship* [Conferencia]. 12th Asia-Pacific Symposium on Internetware, Singapur, Singapur. <https://doi.org/10.1145/3457913.3457927>

6 ANEXOS

6.1 Anexo 1

Figura 31

Proceso de diseño en empresas de telecomunicaciones

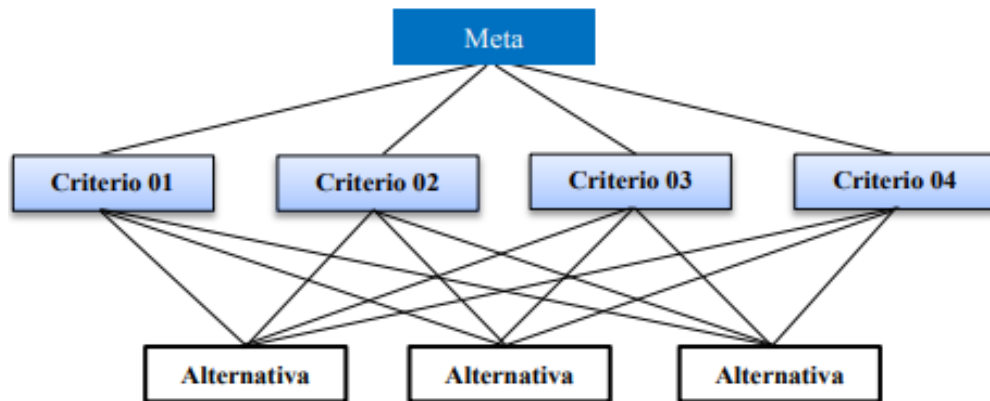


Nota. Fases e hitos del proceso de diseño para la concepción de nuevos productos y servicios en empresas de telecomunicaciones. De “New Product Development Architectural Framework for Sustainability and Innovation within Telecommunication Industry”, por Lubis et al., 2020 (<https://doi.org/10.1145/3411174.3411197>).

6.2 Anexo 2

Figura 32

Proceso de análisis jerárquico para empresas de telecomunicaciones



Nota. Diferencias entre la arquitectura monolítica y de microservicios. De “Design of a Multicriterio Model with a Diffuse Hierarchical Analysis Process - FAHP, for the Selection of Suppliers in a Mobile Cellular Telecommunications Company”, por Viveros et al., 2019 (<https://doi.org/10.1145/3375900.3375904>).

6.3 Anexo 3

Tabla 9

Plan de Costos

Plan de costos	
Costo estimado: Incertidumbre de ± 10 soles.	
Fuente: SalaryExpert - Economic Research Institute (ERI)	
Rol	Costo por Hora (PEN x Hora)
Project Manager	55
Software Architect	43
Developer	27
UI/UX Designer	25
Quality Assurance	25
Security Specialist	32
Data Engineer	34
Scrum Master	33

6.4 Anexo 4

Figura 33

Plan de continuidad

