

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE SEGUIMIENTO
DE OPERACIONES DE MANTENIMIENTO EN UNA EMPRESA
PROVEEDORA DE SERVICIOS DE TELECOMUNICACIONES**

Tesis para obtener el título profesional de Ingeniero de las Telecomunicaciones

AUTOR:

Percy De La Rosa Vera

ASESOR:

Cesar Stuardo Lucho Romero

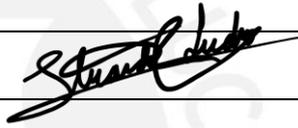
Lima, setiembre, 2024

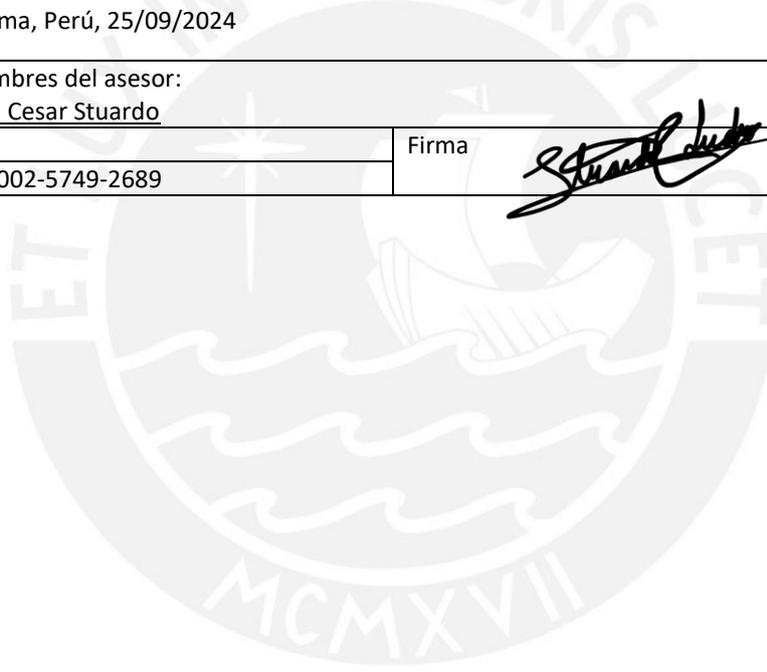
Informe de Similitud

Yo, Cesar Stuardo Lucho Romero, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada Diseño e implementación de una plataforma de seguimiento de operaciones de mantenimiento en una empresa proveedora de servicios de telecomunicaciones, del autor Percy De La Rosa Vera, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 8%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 25/09/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

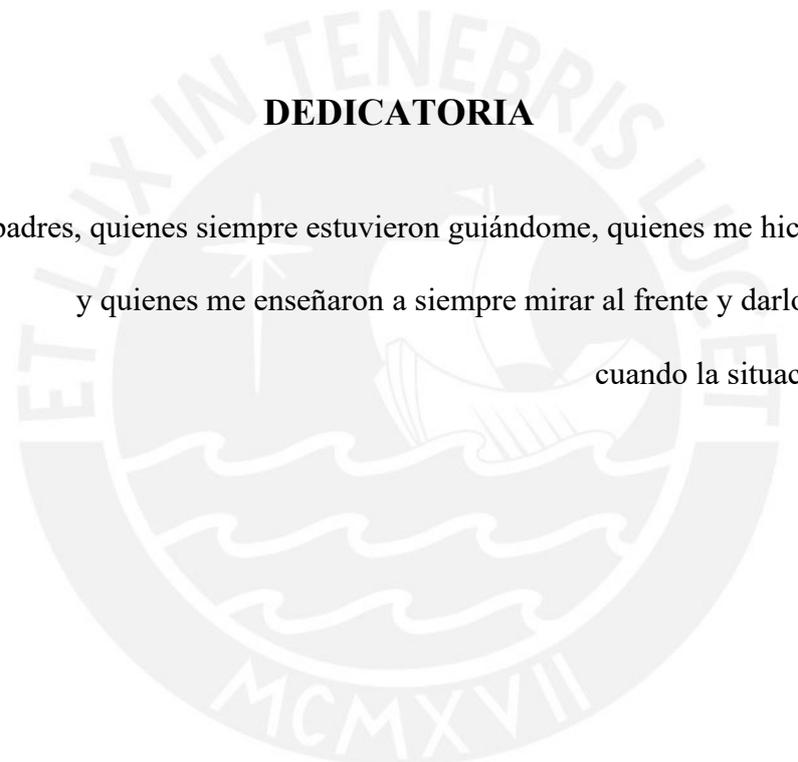
Lugar y fecha: Lima, Perú, 25/09/2024

Apellidos y nombres del asesor: <u>Lucho Romero, Cesar Stuardo</u>	
DNI: 70326404	Firma 
ORCID: 0000-0002-5749-2689	



RESUMEN

En Perú, uno de los desafíos presentes dentro de las empresas proveedoras de servicios de telecomunicaciones es la gestión y el monitoreo de operaciones de mantenimiento. Para realizar este tipo de operaciones a lo largo de todo el territorio nacional, se suele recurrir a los llamados “contratistas”, con quienes frecuentemente surgen problemas. Esto sucede principalmente debido a que las herramientas de gestión de operaciones y los medios de comunicación usados son muchas veces imprecisos, ineficientes, y difíciles de hacerle seguimiento. En respuesta a este problema, el objetivo de esta tesis es el de entregar una plataforma de alta disponibilidad en la nube, que permita agilizar las comunicaciones y centralizar la información de las operaciones de la empresa para usarlas en la toma de decisiones. Para esto, primero se identificó la problemática central, definiendo así el objetivo del trabajo y los requerimientos de la solución. Posteriormente, se eligió el servicio *open source* que más se adecuaba a estos requerimientos, se diseñó la arquitectura de la nube y se desplegó el servicio. Así mismo, la información podía visualizarse desde una plataforma web o desde un aplicativo móvil en Android desarrollado ad-hoc. Teniendo la solución desplegada, se procedió a realizar pruebas de rendimiento, y se analizó su factibilidad económica. Finalmente, se planearon conclusiones y planes para futuro.



DEDICATORIA

A mis padres, quienes siempre estuvieron guiándome, quienes me hicieron quien soy,
y quienes me enseñaron a siempre mirar al frente y darlo todo de mí aún
cuando la situación sea adversa.

AGRADECIMIENTOS

A mi asesor por convencerme de apostar por la carrera de Telecomunicaciones
allá por 2019 en una entrevista que le hizo mi grupo en Motiva.

A Eli, por acompañarme en las amanecidas y siempre buscar formas de hacerme reír.

ÍNDICE GENERAL

INTRODUCCIÓN.....	1
Capítulo 1. Estudio de la problemática de la gestión de operaciones de mantenimiento en empresas proveedoras de servicios de telecomunicaciones en el Perú.....	2
1.1 Área de las Telecomunicaciones.....	2
1.2 Situación actual.....	3
1.2.1 ITSM en el mundo	3
1.2.2 Empresas proveedoras de servicios de telecomunicaciones en el Perú	4
1.2.3 Gestión de incidencias dentro de las empresas proveedoras de servicios de telecomunicaciones en el Perú	5
1.3 Formulación del problema.....	6
1.4 Objetivos.....	7
1.4.1 Objetivo general	7
1.4.2 Objetivos específicos	7
1.5 Importancia.....	8
Capítulo 2. Marco teórico y estado del arte sobre gestión de operaciones en empresas de telecomunicaciones	9
2.1 Estado del Arte	9
2.2 Marco Teórico.....	12
2.2.1 Gestión de incidencias.....	12
2.2.2 Cloud Computing	18

2.2.3 Arquitectura de Alta Disponibilidad	22
2.2.4 Aplicaciones Móviles	25
2.2.5 APIs	29
Capítulo 3. Diseño e implementación de la solución con despliegue en la nube.....	32
3.1 Requerimientos de la Solución	32
3.2 Herramientas Utilizadas	34
3.2.1 Request Tracker	34
3.2.2 Draw.io	36
3.2.3 AWS	36
3.2.4 ClouDNS	38
3.2.5 Figma.....	38
3.2.6 Android Studio	38
3.2.7 Hopscotch.....	39
3.2.8 Apache Jmeter	39
3.3 Setup Local del Servicio	39
3.3.1 Instalación de las dependencias	40
3.3.2 Instalación básica de RT	40
3.3.3 Configuración del servidor Apache	42
3.4 Despliegue de Arquitectura en la Nube	44
3.4.1 Arquitectura Propuesta.....	44
3.4.2 Configuración de los Servicios	45
3.4.3 Habilitando HTTPS.....	55

3.5 Análisis de la Disponibilidad	58
3.6 Diseño e Implementación del Aplicativo Móvil	60
3.6.1 Diagrama de Flujo.....	60
3.6.2 Mockups	60
3.6.3 Resultados Finales.....	64
3.7 Desafíos encontrados	70
Capítulo 4. Pruebas y análisis económico de la solución desarrollada	73
4.1 Pruebas de Rendimiento	73
4.1.1 Configuración del Plan de Pruebas	73
4.1.3 Análisis de Resultados	77
4.2 Análisis Económico.....	78
4.2.1 CAPEX.....	78
4.2.2 OPEX	79
4.3 Recapitulación.....	80
CONCLUSIONES.....	81
RECOMENDACIONES Y OBSERVACIONES	82
BIBLIOGRAFÍA.....	83

ÍNDICE DE TABLAS

Tabla 1. Empresas proveedoras de servicios de telecomunicaciones en el Perú	4
Tabla 2. Cantidad de centros poblados con cobertura móvil	5
Tabla 3. Comparación de soluciones Open Source.....	11
Tabla 4. Comparación entre Help Desk y Service Desk	17
Tabla 5. Comparativa entre modelos de aprovisionamiento en la nube.....	20
Tabla 6. Comparación de los tipos de nube	21
Tabla 7. Requisitos funcionales de la solución	32
Tabla 8. Atributos de la instancia RDS	45
Tabla 9. Atributos de la instancia EC2.....	46
Tabla 10. Atributos del Grupo de Destino	48
Tabla 11. Parámetros del Balanceador de Carga.....	49
Tabla 12. Parámetros del bucket S3	50
Tabla 13. Parámetros de la zona alojada	52
Tabla 14. Parámetros del registro DNS creado	53
Tabla 15. Configuración de Cloudfront	54
Tabla 16. Parámetros del certificado ACM.....	55
Tabla 17. Adición de agente de escucha HTTPS al balanceador.....	57
Tabla 18. Petición para obtener información de usuario.....	65
Tabla 19. Configuración del grupo de hilos en Jmeter	74
Tabla 20. Configuración del HTTP <i>Test Script Recorder</i>	74
Tabla 21. Cálculo de la inversión inicial de la solución.....	78
Tabla 22. Cálculo de los costos de operación mensuales y anuales.....	80

ÍNDICE DE FIGURAS

Figura 1. Estructura ITIL	14
Figura 2. Conceptos de SLM.....	15
Figura 3. Análisis de beneficios tras implementar ITSM.....	16
Figura 4. Nombres que asigna cada empresa a su centro de soporte	17
Figura 5. Cuadrante de Proveedores de Nube - Año 2022.....	19
Figura 6. Ejemplo de arquitectura de Alta Disponibilidad.....	22
Figura 7. Lógica para el diseño de una arquitectura de Alta Disponibilidad	23
Figura 8. Parámetros influyentes en el Downtime	25
Figura 9. Línea de Tiempo del uso de los SO Móviles	27
Figura 10. Cuota de Mercado de los SO Móviles	27
Figura 11. Funcionamiento de una API	29
Figura 12. Página Principal de Request Tracker.....	35
Figura 13. Vista inicial al conectarse al servidor de prueba de RT.....	42
Figura 14. Conexión al servidor a través de HTTP usando el servidor Apache	44
Figura 15. Arquitectura propuesta.....	44
Figura 16. Creación de instancia EC2 usando una plantilla personalizada.....	47
Figura 17. Instancias EC2 creadas	48
Figura 18. Balanceador de Carga Creado.....	49
Figura 19. Prueba de conexión al balanceador de carga	50
Figura 20. Registros DNS generados para la zona alojada de CloudDNS.....	52
Figura 21. Registros DNS generados por la zona alojada de Amazon Route 53	53
Figura 22. Registros DNS modificados en CloudDNS	53
Figura 23. Acceso al servidor a través del dominio registrado	54
Figura 24. Vinculación del dominio con una dirección de correo	56

Figura 25. Adición de los registros MX.....	56
Figura 26. Correo de verificación para emitir el certificado	56
Figura 27. Certificado SSL emitido	57
Figura 28. Prueba de conectividad exitosa usando HTTPS	58
Figura 29. Diagrama de flujo de la aplicación	60
Figura 30. Mockups de inicio de sesión y de mensajes de error.....	61
Figura 31. Mockup de vista de listado de tickets	62
Figura 32. Mockup de vista de detalles del ticket.....	63
Figura 33. Mockup de vista de información de la cuenta	64
Figura 34. Vista de inicio de sesión	66
Figura 35. Vista de listado de tickets, junto a la búsqueda	68
Figura 36. Vista de detalles del ticket	68
Figura 37. Vista de información del usuario	70
Figura 38. Configuración del servidor proxy	75
Figura 39. Peticiones para las pruebas en Jmeter.....	75
Figura 40. Configuración final del plan de prueba en Jmeter	76
Figura 41. Resultados de la prueba de rendimiento con Jmeter.....	77

GLOSARIO

CAU: Centro de Atención del Usuario

KPI: Key Performance Indicator

IT: Information Technologies

ITIL:IT Infrastructure Library

ITSM: IT Service Management

NOC: Network Operation Center

OT: Orden de Trabajo

OC: Orden de Compra

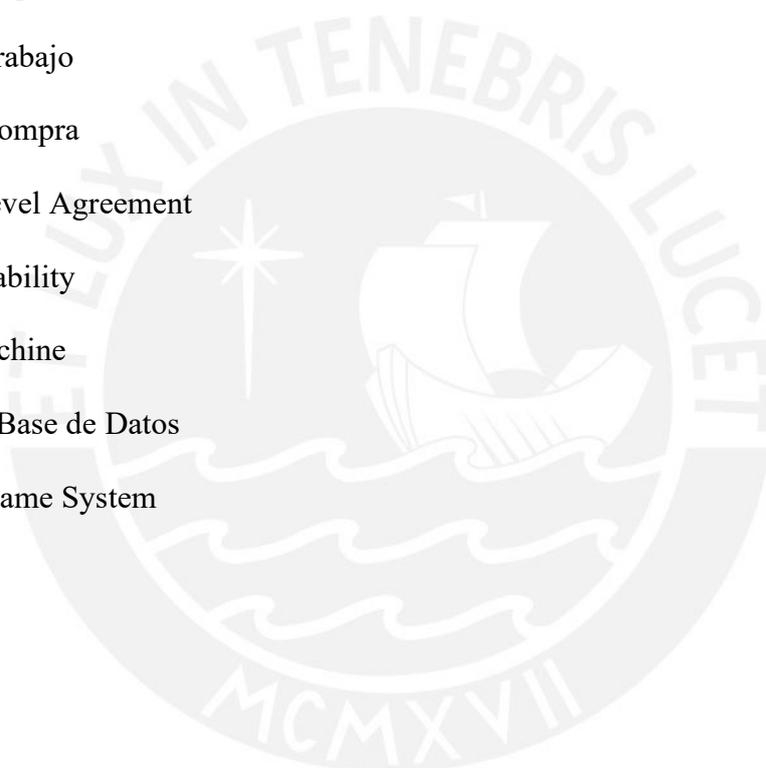
SLA: Service Level Agreement

HA: High Availability

VM: Virtual Machine

DB: Database o Base de Datos

DNS: Domain Name System



INTRODUCCIÓN

En el Perú, muchas empresas de telecomunicaciones medianas y grandes tienen flujos de comunicación y control de proyectos lentos, imprecisos e ineficientes. Estas presentan problemas para centralizar información registrada en distintas fuentes, afectando también a la elaboración de reportes y analíticas que puedan servir para formular presupuestos, proyecciones y tomar decisiones empresariales.

De esta forma, el concepto de *IT Service Management* (ITSM) surgió para agilizar la gestión empresarial a través de tecnologías orientadas a los servicios. En el mercado actual existe una gran cantidad de soluciones ITSM empresariales que buscan resolver este problema. Sin embargo, la mayoría de estas tienen sus funcionalidades limitadas, necesitando de pagos adicionales que pueden aumentar considerablemente los costos de operación para el servicio.

Con esta problemática en mente, el objetivo de esta investigación es el de diseñar e implementar una plataforma para la gestión de operaciones de mantenimiento en el contexto de una empresa proveedora de servicios de telecomunicaciones.

En este documento se describirá sobre el proceso de desarrollo y posterior análisis de la solución. En el primer capítulo se dará contexto al problema y se definirán objetivos a cumplir para la solución propuesta. En el segundo capítulo, se explicarán las bases teóricas para el desarrollo de la solución, tales como conceptos sobre la nube, alta disponibilidad, y APIs. En el tercer capítulo, se detalla el proceso de implementación de la solución, incluyendo el diseño de la arquitectura de alta disponibilidad, despliegue en la nube, y el diseño y desarrollo de una aplicación móvil que se integre a esta solución. Finalmente, en el cuarto capítulo se realizará un análisis de rendimiento y económico de la solución propuesta. Finalizando el documento, se darán conclusiones y consideraciones de la investigación, y se presentarán las fuentes bibliográficas empleadas.

Capítulo 1. Estudio de la problemática de la gestión de operaciones de mantenimiento en empresas proveedoras de servicios de telecomunicaciones en el Perú

En el presente capítulo se presentará el contexto actual de las labores de mantenimiento en las empresas de telecomunicaciones, identificando la problemática central. A partir de esta, se definirán los objetivos centrales del presente trabajo. Los contenidos de esta sección se basan en los definidos en el trabajo de investigación previo. [1]

1.1 Área de las Telecomunicaciones

El siguiente trabajo se desarrolla y enfoca en las áreas de software y de gestión de proyectos, correspondientes a la carrera de Ingeniería de las Telecomunicaciones. De esta forma, los conocimientos adquiridos en los cursos de *Ingeniería Web para las Telecomunicaciones*, *Gestión de Servicios de TICS* y *Servicios y Aplicaciones de IoT* son esenciales para la comprensión de los conceptos y para el diseño e implementación de la solución presentada al finalizar este trabajo. En estos cursos se aprendió sobre el desarrollo y despliegue de servicios

web, fundamentos del despliegue de programas y servicios en la nube, el uso y consumo de APIs, el desarrollo de aplicaciones móviles, así como otros conceptos relacionados con las tecnologías de la información y comunicación.

Asimismo, los cursos de *Formulación y Evaluación de Proyectos de Telecomunicaciones* y *Proyecto Integrador de Telecomunicaciones* brindaron los conocimientos necesarios para la planificación de un proyecto de telecomunicaciones orientado a empresas. Se aprendieron conceptos sobre el diagnóstico e identificación del problema central, análisis de alternativas de solución, y la evaluación de la propuesta de proyecto bajo los puntos de vista ambiental, social, económico y legal.

1.2 Situación actual

En esta sección se explicará el contexto actual sobre el que se realiza este trabajo. Primero, se explicará la tendencia a nivel mundial de adoptar las prácticas planteadas por el ITSM. Posteriormente, se hablará sobre la expansión de estas empresas de telecomunicaciones en el país, así de sus estrategias y métodos actuales para la gestión y monitorización de sus operaciones de mantenimiento.

1.2.1 ITSM en el mundo

En un inicio, el área de IT se enfocaba en la resolución de problemas técnicos dentro de distintos componentes dentro de la infraestructura IT de una empresa. Sin embargo, el crecimiento en los campos de conectividad debido a la llegada de Internet significó la llegada sistemas de administración IT mucho más complejos. En el sector empresarial y corporativo, surgió la necesidad de tener soluciones tecnológicas estandarizadas para poder simplificar y mejorar la disponibilidad y confiabilidad de los diferentes sistemas administrativos de la empresa. [2]

Ante el descontento de las empresas y corporaciones con una infraestructura IT poco transparente y que no se alinea con sus objetivos empresariales, surgió un creciente interés en el *IT Service Management*. Similar a como se integraron conceptos de gestión de proyectos en el área de ingeniería de software para la administración de grandes proyectos de desarrollo [2], el *IT Service Management* (ITSM) busca implementar una administración de IT orientada a los servicios, centrándose en proveer una administración no tan centrada en la innovación tecnológica, sino orientada a brindar servicios confiables a los usuarios. [3]

1.2.2 Empresas proveedoras de servicios de telecomunicaciones en el Perú

En la actualidad, el Perú tiene más de 32 millones de habitantes, de los cuales se estima que emplean más de 43 millones de líneas móviles, y más de 24 millones poseen conexión a Internet. [4] Todas estas personas, para poder satisfacer su necesidad de conectividad, recurren a acuerdos comerciales con compañías proveedoras de servicios de telecomunicaciones. En este país, para el año 2023, según la OSIPTEL predominan en el mercado las empresas Movistar, Claro, Entel y Bitel. [5]

Tabla 1. Empresas proveedoras de servicios de telecomunicaciones en el Perú

Operador	Propietario	Servicios	Tecnología/Espectro	Participación de mercado
Bitel	Vittel	Telefonía y Banda Ancha Móvil	900 MHz; 1900 MHz GSM / GPRS / EDGE; UMTS / HSPA+; LTE	19,5
Claro	América Móvil	Telefonía y Banda Ancha Móvil	700 MHz; 850 MHz; 1900 MHz GSM / GPRS / EDGE; UMTS / HSPA+; LTE	28,5
Entel	Entel Chile	Telefonía y Banda Ancha Móvil	700 MHz; 1900 MHz; 1700 MHz / 2100 MHz GSM / GPRS / EDGE; iDEN/ UMTS / HSPA+; LTE	20,6
Movistar	Telefónica	Telefonía y Banda Ancha Móvil	700 MHz; 850 MHz; 1900 MHz; 1700 MHz / 2100 MHz GSM / GPRS / EDGE; UMTS / HSPA+; LTE	30,6
Otros				0,8

Fuente: [5]

Con el objetivo de expandir su alcance, y por ende la cantidad de clientes potenciales, estas empresas realizaron y siguen realizando diversos proyectos de expansión en diferentes regiones del país, trayendo infraestructura y conectividad básica a diversos sectores. A partir de registros del OSIPTEL [6], [7], se tiene información estimada sobre el alcance que tiene cada operadora en el país.

Tabla 2. Cantidad de centros poblados con cobertura móvil

Características	Movistar	Claro	Entel	Bitel
Participación en el mercado 2023	29.45%	29.99%	22.96%	19.3%
Centros poblados con cobertura móvil 2019 - III	19780	22582	7587	27079
Centros poblados con cobertura móvil 2023 - II	24157	6143	14715	6351

Fuente: Elaboración propia, empleando la información obtenida de [6], [7]

1.2.3 Gestión de incidencias dentro de las empresas proveedoras de servicios de telecomunicaciones en el Perú

Sólo para el caso de la telefonía móvil, considerando la necesidad de la implementación de al menos 1 (una) radio base por centro poblado, una empresa de telecomunicaciones no es capaz de tener presencia ni dar soporte en todos estos centros poblados sin ayuda externa. De esta forma estas empresas recurren comúnmente a acuerdos comerciales con empresas externas con mayor presencia en la región, a las que se denomina contratistas. De esta forma, los contratistas se encargan de las labores de implementación y mantenimiento de los servicios en estos centros poblados, mientras que la empresa debe solo de gestionar y mantener la comunicación con estos grupos, realizando un seguimiento del desarrollo de estos proyectos. El presente trabajo se centrará en la gestión de las operaciones de mantenimiento.

A partir de las conversaciones que se han tenido con el Ing. Eduardo Vásquez [8], representante de la empresa de telecomunicaciones Gilat, las actividades de mantenimiento de una empresa se clasifican en 2 (dos) tipos, los cuales son:

- **Actividades Preventivas:** Se realizan de forma rutinaria para monitorear el correcto funcionamiento del sistema.
- **Actividades Correctivas:** Se realizan tras haberse reportado un fallo, ya sea por parte del cliente o del NOC.

Además, se tiene que los 3 (tres) flujos de atención comunes en una empresa son los siguientes:

- **CAU:** Recibe los reclamos y reportes de los usuarios. Es poco frecuente.
- **NOC:** Identifican un problema en la red (Revisión interna, chequeos). Intentan resolver los problemas identificados de forma remota.
- **Mantenimiento:** Se les asigna los casos que el NOC no puede resolver. Solucionan los problemas de forma presencial.

1.3 Formulación del problema

En muchas empresas en el Perú, es común que las actividades de mantenimiento sean monitorizadas de forma manual por medio de comunicación directa con los contratistas mediante chats de texto como WhatsApp o por medio de llamadas telefónicas. Por esto mismo, el proceso de revisiones es usualmente largo, trabajoso y disperso. Esto trae problemas de coordinación y de seguimiento de los procesos, lo que lleva a que suelen ser muy largos y sus resultados sean difíciles de medir. Otro problema es que la adquisición de información es lenta e imprecisa, además de que la empresa resulta muy afectada si por alguna razón pierden la comunicación con el contratista.

De forma concisa, la problemática de muchas empresas de telecomunicaciones en el Perú consiste en un manejo deficiente y desorganizado de la información al no poder gestionar correctamente sus operaciones de mantenimiento. Para solucionar este problema, lo que estas empresas necesitan es una solución transparente que permita realizar un seguimiento de las actividades y OTs tanto para los contratistas, clientes y para los mismos trabajadores.

1.4 Objetivos

1.4.1 Objetivo general

El objetivo general de la presente tesis es el de diseñar e implementar una plataforma para la gestión de operaciones de mantenimiento en el contexto de una empresa proveedora de servicios de telecomunicaciones. Para esto, se analizará el modelo de negocio y funcionalidades ofrecidas por cada alternativa.

1.4.2 Objetivos específicos

Los objetivos específicos de este trabajo de tesis son los siguientes:

- Levantar un servicio que soporte los roles de: cliente, operador del NOC, personal de mantenimiento y administrador.
- El servicio debe permitir la gestión de tickets, SLAs, usuarios, grupos y permisos.
- Desplegar el servicio en la nube, considerando una arquitectura de alta disponibilidad que incluya: redundancia de base de datos, balanceadores de carga, servidores de aplicación, etc.
- Permitir el acceso seguro por HTTPS al servicio.
- Diseñar e implementar una miniaplicación móvil que pueda ser usada para la creación, seguimiento y participación en tickets.

1.5 Importancia

La propuesta de este trabajo de tesis significará una mejora considerable en los procesos de gestión de operaciones de mantenimiento en las diferentes empresas proveedoras de servicios de telecomunicaciones. Esto pues le permitiría a cada empresa tener un flujo más eficiente y rápido de comunicación en sus operaciones. El tener un medio de información centralizado también les permitirá a las empresas visualizarla y procesarla de una forma más sencilla para usarla en la toma de decisiones importantes, el despliegue de cuadrillas, el manejo de presupuestos, etc.



Capítulo 2. Marco teórico y estado del arte sobre gestión de operaciones en empresas de telecomunicaciones

2.1 Estado del Arte

Antes de profundizar más en el tema, se evaluarán algunas soluciones ya presentes en el mercado para la gestión de operaciones de mantenimiento para el sector empresarial. Empezando por las soluciones de pago, estas pueden poseer tanto licenciamiento permanente como por suscripción, y brindando un soporte temporal. Entre algunas de las principales soluciones de este tipo se tienen:

- **Zendesk:** Es una solución ITSM desarrollada por la compañía del mismo nombre. Esta ofrece funcionalidades de mensajería y llamada con clientes, ticketing, analíticas, automatización e inteligencia artificial (IA), así como la integración con diferentes servicios. Esta ofrece planes desde los \$55 a \$115 mensuales por agente. [54]
- **Jira Service Management:** Es la plataforma de Atlassian para construir y entregar servicios digitales. En su solución permiten la interacción entre las áreas de

administración de proyectos, atención al cliente, recursos humanos, finanzas y marketing. Esta ofrece un plan gratuito con hasta 3 agentes, y de ahí un modelo de suscripción mensual de entre \$22 y \$50 por agente. [55]

- **Freshservice:** Es una herramienta ITSM de Freshworks, que permite la gestión de proyectos, el monitoreo de recursos, administración de servicios y operaciones, etc. Su modelo de negocios consiste en una suscripción desde \$19 a \$119 mensuales por agente mensualmente. [56]
- **SolarWinds Service Desk:** Es la alternativa de Solarwinds, la cual permite el manejo de tickets, la gestión de SLAs, administración de equipamiento, así como extensiones y una API. Sus suscripciones mensuales por agente varían entre los \$39 y \$99. [57]

Asimismo, se tienen soluciones de código abierto, las cuales ofrecen una solución gratuita y un soporte por parte de la comunidad. A continuación se muestran las soluciones de este tipo más populares:

- **Faveo:** Faveo es un software Open Source para el manejo de tickets lanzado por la empresa homónima. Faveo ofrece múltiples ediciones, dependiendo del caso de uso, funcionalidades, y nivel de servicio técnico deseado.
- **GLPI:** GLPI (o Gestor Libre de Activos Informáticos por sus siglas en francés) es una solución de código abierto para la administración IT. Este es parte de un proyecto de más de 20 años de trayectoria, el cual ofrece características y funcionalidades.
- **OsTicket:** OsTicket es otra alternativa de código abierto para el manejo de tickets, siendo una de las opciones más populares para el manejo de solicitudes debido a su simpleza.
- **Request Tracker:** Request Tracker (RT) es un sistema de *ticketing* proporcionado por la empresa *Best Practical*. Este es de código abierto y provee la funcionalidad de gestión de tickets e inventario, administración de SLAs, dashboards y reportes. RT

ofrece planes de servicio en la nube a través de una membresía, y el servicio para usarlo en un servidor propio. [41]

Comparando las funcionalidades de estas últimas alternativas, se tiene la siguiente tabla.

Tabla 3. Comparación de soluciones Open Source

Característica	Faveo	GLPI	OsTicket	Request Tracker
Gestión de Tickets	<ul style="list-style-type: none"> • CRUD • Prioridades • Flujo de trabajo • Búsqueda y filtrado 	<ul style="list-style-type: none"> • CRUD • Formularios personalizados • Enrutamiento • Creación de tickets recurrentes 	<ul style="list-style-type: none"> • CRUD • Formularios personalizados • Sistema de tareas requeridas para cerrar ticket 	<ul style="list-style-type: none"> • CRUD • Formularios personalizados • Flujos separados para reporte, manejo, investigación y contramedidas
Manejo de SLAs	<ul style="list-style-type: none"> • CRUD • Asignación manual y automática por departamento 	<ul style="list-style-type: none"> • CRUD • Asignación manual y automática por departamento • Alertas y notificaciones 	<ul style="list-style-type: none"> • CRUD • Asignación manual y automática por departamento • Alertas y notificaciones • Escalamiento 	<ul style="list-style-type: none"> • CRUD • Asignación manual y automática por departamento • Alertas y notificaciones • Escalamiento • Prioridades según SLA
Gestión de Permisos y Usuarios	<ul style="list-style-type: none"> • Clientes y agentes • Permisos estándar y de administrador • Agrupación en equipos y departamentos 	<ul style="list-style-type: none"> • 8 roles con diferentes permisos • Creación de grupos de forma jerárquica. 	<ul style="list-style-type: none"> • Usuario, agente y administrador • Permisos individuales para visualización, edición, borrado, asignación de agentes, etc. • Formación de departamentos y equipos entre departamentos 	<ul style="list-style-type: none"> • Usuario, agente y administrador • Permisos de visualización, edición y participación. • Agrupación en equipos y departamentos
Dashboards y Reportes	<ul style="list-style-type: none"> • Resumen del estado de tickets • Gráfico de puntos • Almacenamiento en base de datos relacional 	<ul style="list-style-type: none"> • Resumen del estado de tickets • Gráficos de puntos y de barras • Almacenamiento en base de datos relacional • Funcionalidades avanzadas a través de plugins 	<ul style="list-style-type: none"> • Dashboard básico: diagrama de puntos. • Almacenamiento en base de datos relacional 	<ul style="list-style-type: none"> • Dashboard configurable. • Personalizado por departamento y usuario. • Tablas, listas, y diagramas de barras y circular • Almacenamiento en base de datos relacional

APIs y Documentación	<ul style="list-style-type: none"> • API completa • Documentación actualizada 	<ul style="list-style-type: none"> • API completa • Documentación muy detallada y actualizada 	<ul style="list-style-type: none"> • API básica, con logueo y creación de ticket • Documentación actualizada 	<ul style="list-style-type: none"> • API completa • Documentación extensa y actualizada • Ejemplos de uso.
Funcionalidades adicionales	<ul style="list-style-type: none"> • <i>Plugins</i> para logueo e integración con mensajería instantánea. 	<ul style="list-style-type: none"> • Gestión de inventario físico y virtual • Gestión financiera • Gestión de proyectos • Canales RSS • Amplio ecosistema de <i>plugins</i> que mejoran la integración y agregan funcionalidades 	No tiene	<ul style="list-style-type: none"> • Gestión de inventario • Monitoreo de actividad • Canales RSS • Amplio ecosistema de <i>plugins</i> que mejoran la integración y agregan funcionalidades

Fuente: Elaboración Propia

Después de realizar el análisis comparativo, se llegó a la conclusión que la alternativa más adecuada es Request Tracker (RT), pues tiene la funcionalidad necesaria (descrito en la sección 3.1), dashboards y reportes configurables y personalizables, una documentación y API actualizadas y muy completas, además de ofrecer muchas funcionalidades adicionales a través de un sistema de extensiones creadas por la comunidad.

2.2 Marco Teórico

2.2.1 Gestión de incidencias

Como ya se explicó en el punto 1.2.1, el ITSM es un conjunto de técnicas con las cuales se puede maximizar la eficiencia de operaciones en una empresa. El ITSM logra esto por medio de una arquitectura TI orientada a servicios que satisfagan las necesidades del usuario. En este punto, se explorará este tema a mayor profundidad, considerando sus bases, los conceptos que trae consigo, los beneficios que trae su adopción, y las formas de aplicarlos en la gestión de operaciones de una empresa.

2.2.1.1 ITIL

La *Information Technology Infrastructure Library* (ITIL) comprende un conjunto de mejores prácticas de ITSM. Esta normativa fue introducida en el Reino Unido por la *Central Computer and Telecommunications Agency* (CCTA) en 1989 como un compendio de 30 volúmenes. En el año 2001, la segunda versión reorganizó las secciones para cada uno de los diferentes aspectos de la administración IT. En el 2007 fue publicada la versión 3, la cual condensó todo en 5 volúmenes. Finalmente, en el 2019 se publicó la versión 4, trayendo conceptos sobre transformación digital, IA, *cloud computing* y *DevOps*. [9]

El ITIL se centra en 5 aspectos principales del ITSM:

- **Estrategia de Servicio:** Da consideraciones al momento de diseñar, desarrollar e implementar ITSM.
- **Diseño del Servicio:** Da indicaciones sobre como diseñar diferentes tipos de servicios y procesos.
- **Transición de Servicios:** Explica como cambiar o modificar un servicio existente sin alterar al resto de procesos operando.
- **Operación de Servicio:** Da indicaciones sobre como verificar que los servicios estén operando correctamente.
- **Mejora Constante de Servicios:** Da indicaciones sobre como adaptar los servicios a los cambios en la empresa.



Figura 1. Estructura ITIL
Fuente: [10]

Se enfatiza que la ITIL no es una normativa, sino un compendio de recomendaciones para la implementación de ITSM. Es decir, la ITIL no emite certificaciones ni otorga méritos por su cumplimiento. Sin embargo, a partir de esta surgieron normativas que sí lo hacen. Entre ellas se tiene a la ISO 20000 y la BS15000, las cuales promueven aún más la adopción de estas consideraciones. [11]

2.2.1.2 Service Level Management

Una de los aspectos más importantes para alinear los esfuerzos en el departamento IT con los objetivos de la empresa es el *Service Level Management* (SLM). El SLM busca brindar servicios IT dirigidos a los clientes por medio de la negociación, optimización y seguimiento de SLAs. [11], [12]

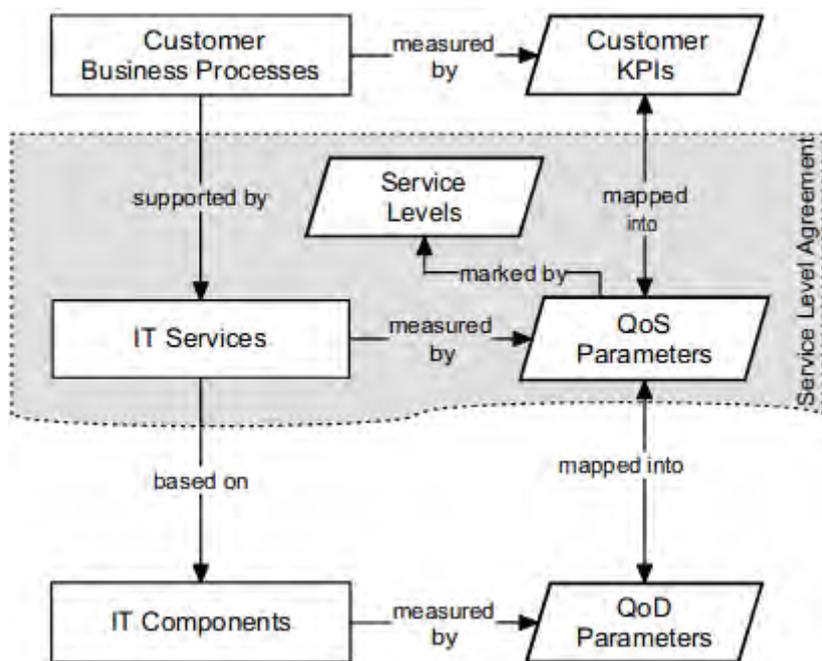


Figura 2. Conceptos de SLM

Fuente: [11]

Este considera múltiples elementos, entre ellos los siguientes [11], [12] :

- **Service Level Agreement (SLA):** Es un acuerdo escrito, el cual especifica de forma cuantitativa o cualitativa las condiciones con las que se proveerá un servicio. [6], [7]

En este se indica lo siguiente:

- Métricas para medir la calidad del servicio provisto.
- Penalidades en el caso que no se cumplan las condiciones acordadas.
- **Key Performance Indicator (KPI):** Es una medida cuantificable que sirve para definir con claridad el progreso o rendimiento de un proceso de la empresa, sirviendo como base para evaluar el progreso hacia un objetivo. [6], [7] Este puede ser usado en los siguientes casos:
 - Procesos, tecnologías, recursos o componentes de un negocio.
 - Criterios para la toma de decisiones como crecimiento, ingresos económicos, etc.

2.2.1.3 Beneficios

De acuerdo a un estudio realizado a múltiples empresas suizas [3], el uso de soluciones ITSM significó mejoras en las siguientes áreas:

- **Transparencia:** Al facilitar la monitorización de los procesos por parte de los empleados y los clientes.
- **Entendimiento:** La información recolectada de los múltiples servicios de la empresa permiten mejorar la calidad del servicio.
- **Eficiencia:** Al estandarizar, monitorizar y automatizar procesos.

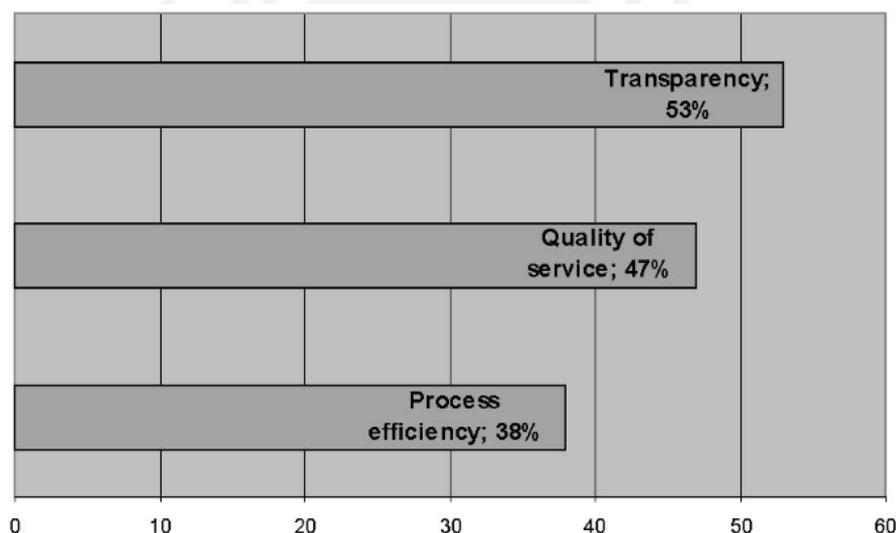


Figura 3. Análisis de beneficios tras implementar ITSM

Fuente: [3]

12.2.1.4 Acceso a Servicios IT

Existen 2 formas de acceso y administración de los servicios IT en una empresa: *help desks* y *service desks*. Comúnmente agrupados dentro de términos como soporte técnico, centros de atención al cliente o incluso *Call Centers*, estos términos suelen ser poco claros e interpretados de distinta forma por cada organización o individuo. [13] Esto se puede evidenciar en el siguiente gráfico:

The support center is referred to as...

Respondents were asked to select the closest match to their support center's title.

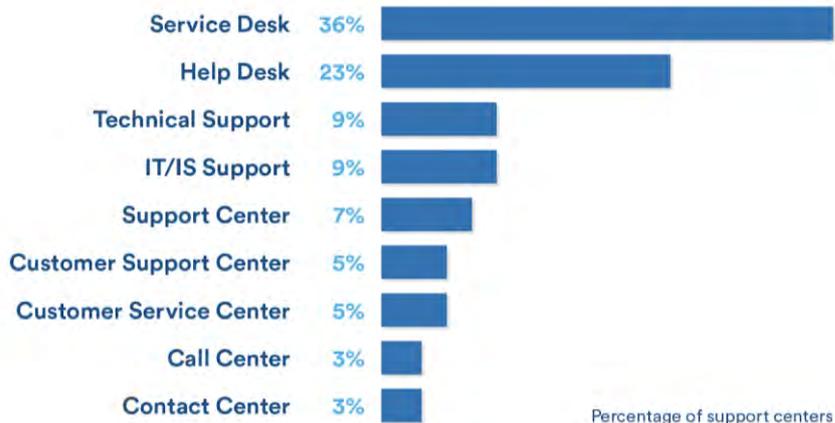


Figura 4. Nombres que asigna cada empresa a su centro de soporte

Fuente: [14]

Debido a esto, a continuación se explicará en qué consisten ambos términos y se compararán sus características y casos de uso [13], [15]:

- Help Desk:** Un *IT Service Desk* se centra en la resolución rápida de problemas de IT. Este aborda principalmente el proceso de manejo de incidencias dentro del esquema ITSM. Este tiene naturaleza reactiva, resolviendo un problema después que halla sido reportado. Su uso está destinado a empresas más pequeñas con una infraestructura IT reducida.
- Service Desk:** Por otro lado, un *IT Service Desk* ofrece una mayor cantidad de servicios. La ITIL lo define como un punto de contacto entre el usuario y el proveedor de servicios. Este tiene un mayor enfoque empresarial, centrándose no solo en la resolución de problemas. En cambio, esta se integra con más procesos ITSM, atiende las incidencias y solicitudes de los clientes y verifica el cumplimiento de SLAs. Se recomienda su uso en organizaciones con una infraestructura IT más compleja.

Tabla 4. Comparación entre Help Desk y Service Desk

	Help Desks	Service Desks

Enfoque	Usuario	Negocio
Forma de accionar	Reactiva, para solución de problemas a corto plazo	Proactiva, para plantear soluciones a largo plazo
Modelo	Solucionar errores específicos del usuario cuando ocurran.	Solucionar un problema en alineación con los objetivos de la empresa
Orientación	Orientado a tareas que satisfagan las necesidades del usuario	Orientado a procesos que mejoren y agilicen el funcionamiento de la empresa.

Fuente: [15] (Traducida)

2.2.2 Cloud Computing

El *Cloud Computing* es un nuevo paradigma para el consumo de recursos digitales, que apareció junto a los avances en la digitalización, según Amazon [16], el *Cloud Computing* consiste en la entrega de recursos de TI a través del Internet por medio de un modelo bajo demanda, de tal forma que servicios tanto de cómputo, almacenamiento como monitoreo puedan ser aprovisionados por un proveedor de nube.

2.2.2.1 Proveedores

En el mercado actual existe una gran cantidad de proveedores de servicios en la nube. Por esto mismo, junto con el apoyo del análisis de la consultora Gartner [17], se pudieron identificar a algunos de los principales proveedores de estos servicios de TI, las cuales fueron clasificadas en un cuadrante según su visión y su capacidad para alcanzarla.



Figura 5. Cuadrante de Proveedores de Nube - Año 2022

Fuente: [17]

De este cuadrante entonces, se identificaron a 3 compañías líderes en el sector, siendo estas:

- Amazon Web Services (AWS):** Es una compañía subsidiaria de Amazon. Este es el líder actual del mercado para este sector, superando en gran medida a sus otros competidores en usuarios y ganancias. Se caracteriza por ofrecer un gran y variado ecosistema de servicios y funcionalidades, los cuales permiten a los usuarios adaptar su arquitectura de nube a múltiples tipos de escenarios distintos de una forma sencilla y centralizada. Sin embargo, se conoce que AWS es poco flexible ante la interoperabilidad con la infraestructura de otros operadores, obligando a sus usuarios a trabajar únicamente con su ecosistema.
- Microsoft Azure:** Para el caso de la plataforma de Microsoft, esta centra sus esfuerzos en la integración con los otros servicios de la empresa, ofreciendo mayor soporte para entornos híbridos o de múltiples proveedores. Este suele ser usado mayormente en el

ámbito empresarial, por empresas medianas y grandes, pero con el paso del tiempo su base de usuarios está creciendo, con una tendencia a alcanzar los números de AWS.

- **Google Cloud Platform (GCP):** En el caso de Google, esta pone sus esfuerzos en la descentralización de su infraestructura, con el objetivo de diversificar a su base de clientes, proveyendo principalmente soluciones IaaS y PaaS.

2.2.2.2 Modelos de Aprovisionamiento

Cada uno de los servicios ofrecidos por estos proveedores puede ser clasificado según el nivel de control que posee el usuario sobre este, siendo el resto gestionado por el mismo proveedor, identificando así a 3 modelos principales. A partir de artículos de AWS [18], Redhat [19] e IBM [20], estos fueron descritos en la siguiente tabla.

Tabla 5. Comparativa entre modelos de aprovisionamiento en la nube

	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
Concepto	El proveedor da acceso al usuario a infraestructura de cómputo, siendo estos servidores dedicados o VMs. El proveedor se encarga únicamente de administrar el hardware.	Remueve la necesidad del usuario final de administrar el hardware y sistema operativo, centrándose en el desarrollo, despliegue y administración de la aplicación	El proveedor se hace cargo de la gestión de toda la aplicación. De esta forma, el usuario solo debe pensar en cómo usarla.
Orientado a	Desarrolladores	Desarrolladores	Usuarios Finales
Beneficios	<ul style="list-style-type: none"> • Reduce CAPEX • Facilita el escalamiento de recursos de acuerdo a la demanda • Permite tener una mayor disponibilidad al tener servidores distribuidos en distintas regiones. 	<ul style="list-style-type: none"> • Elimina la necesidad del usuario de mantener el sistema. • Ofrece acceso a la última versión de los recursos a utilizar. • Simplifica la colaboración. 	<ul style="list-style-type: none"> • El usuario final no debe preocuparse en mantener, administrar ni configurar el sistema. • Permite al usuario enfocarse en usar el producto. • Para escalar el sistema solo se le paga al proveedor.

Ejemplos	<ul style="list-style-type: none"> • Amazon EC2 • Azure Compute Services • Google Compute Engine 	<ul style="list-style-type: none"> • Plataformas de orquestación de recursos (Redhat OpenShift, AWS Beanstalk) • Plataformas de desarrollo (Heroku, Github Codespaces) 	<ul style="list-style-type: none"> • Servicios de correo, mensajería o videoconferencia. • Suites de oficina (Google Workspace, Microsoft 365) • Servicios de streaming (Spotify, Netflix, etc.)
-----------------	---	--	---

Fuente: Elaboración propia

2.2.2.3 Tipos de Nube

Así como se pueden agrupar las soluciones de nube de acuerdo al modelo en el que se basan para brindar sus servicios, también se les puede clasificar de acuerdo a quien administra la infraestructura física usada para soportar todos los servicios de nube ofrecidos.

Tabla 6. Comparación de los tipos de nube

	Nube Pública	Nube Privada	Nube Híbrida
Recursos	Se comparten entre distintos clientes	Dedicados para una sola organización	Usa recursos de ambos tipos según las necesidades de la empresa.
Operador	Proveedor externo	La misma organización	Una combinación de ambos.
Escalabilidad	Altamente escalable	It has predictability and consistency	It has scalability and flexibility by allowing organizations to use a combination of public and private cloud services.
Seguridad	Menor al ser accesibles al público	Depende de la infraestructura de la empresa	Menor en los puntos de interacción entre distintas nubes
Costos	Barato	Más caro	Depende de la administración de recursos de la empresa.

Fuente: [21] (Traducida)

2.2.3 Arquitectura de Alta Disponibilidad

En la actualidad, las aplicaciones más complejas pueden llegar a necesitar de múltiples componentes, siendo cada uno de estos un posible punto de falla [22]. Ante esta situación, la Alta Disponibilidad hace referencia a la habilidad de un sistema de operar de forma continua sin fallas incluso durante eventos inesperados. [23]

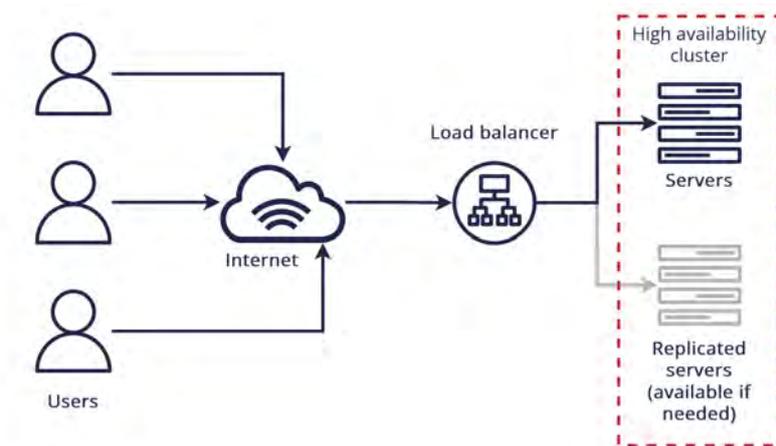


Figura 6. Ejemplo de arquitectura de Alta Disponibilidad

Fuente: [24]

2.2.3.1 Importancia

En la sociedad moderna existen algunos sistemas de misión crítica, como lo son los sistemas médicos, financieros y gubernamentales. Estos sistemas necesitan tener un alto nivel de disponibilidad debido a su importancia y alta demanda por parte de los usuarios. De esta forma el objetivo de las soluciones de alta disponibilidad en estos sistemas es el de brindar un servicio de forma óptima durante el 99.999% del tiempo (5 nueves). [25]

Sin embargo, con el aumento de la demanda de los servicios en línea, muchas organizaciones e industrias pueden desear tener una alta disponibilidad (del 99.99% o 99.999%). Esto pues la caída de sus servicios puede resultar en pérdidas de ingresos, reducción de la productividad, e incluso en daños a la imagen y reputación de la empresa. [25]

De esta forma, la importancia de la alta disponibilidad recae en los siguientes factores [25]:

- **Continuidad del Servicio:** Permite asegurar el funcionamiento de los sistemas críticos de la organización, de tal forma que el usuario experimenta una caída de servicio corta o nula.
- **Mejora en la experiencia de usuario:** Un sistema de alta disponibilidad es rápido y confiable, manteniendo a los usuarios del servicio satisfechos.
- **Ventaja Competitiva:** El ofrecer un servicio más rápido y confiable que la competencia puede servir como un factor diferenciador ante la competencia.

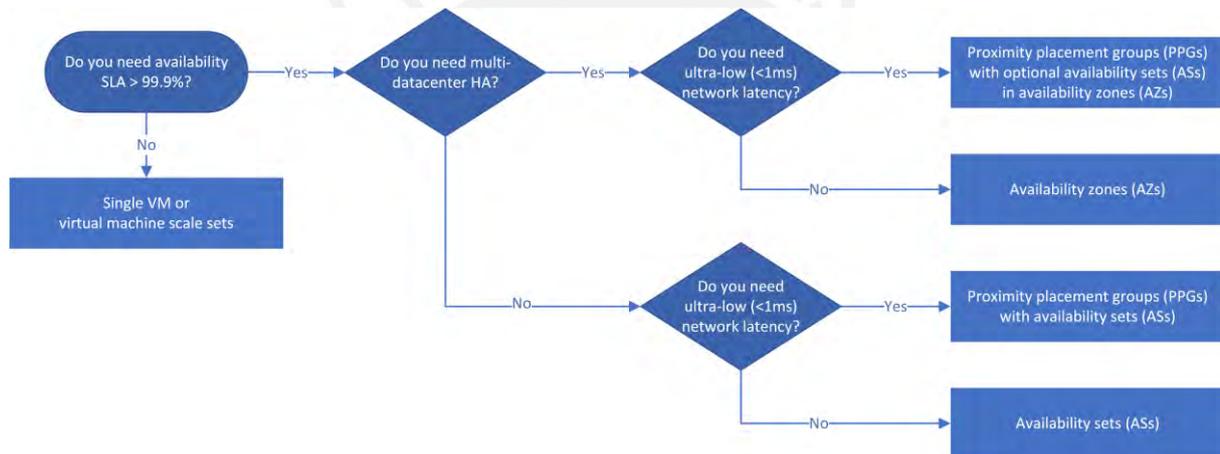


Figura 7. Lógica para el diseño de una arquitectura de Alta Disponibilidad

Fuente: [26]

2.2.3.2 Mecanismos empleados

Para lograr esta operación continua, los sistemas de alta disponibilidad usan distintos mecanismos de redundancia y *failover*, entre ellos teniendo [25]:

- **Redundancia:** Uso de más de un servidor, unidad de almacenamiento, fuente de alimentación o conexión de red, de tal forma que un componente caído pueda ser reemplazado rápidamente.

- **Monitoreo:** Uso de herramientas de monitoreo para lanzar alertas, notificando a los administradores del sistema para tomar acción lo más rápido posible.
- **Balanceo de Cargas:** Uso de un servidor especializado para repartir tráfico entre un grupo de servidores, mejorando el rendimiento de la aplicación. Además, permiten la redirección de tráfico si un servidor cae.
- **Sistemas de *Backup*:** Permite el rescate y recuperación de la información de un sistema en caso de falla.

2.2.3.3 Tipos de Alta Disponibilidad

Los sistemas de alta disponibilidad pueden diferenciarse de distintas formas. Sin embargo, la principal manera de diferenciarlos es según su modo de operación, tal que se tienen los siguientes tipos [25]:

- **Activo – Pasivo:** El sistema de respaldo se encuentra en modo de espera, mientras que el principal está activo. Cuando este cae, el sistema de respaldo es activado ya sea de forma manual o automatizada.
- **Activo – Activo:** Todos los sistemas corren de manera simultánea, teniéndose un elemento adicional que reparte la carga entre todos ellos. Al fallar uno de estos sistemas, el resto automáticamente recoge su carga. Estos sistemas son más complejos de implementar, pero llevan a una mejor rendimiento y escalabilidad.

2.2.3.4 Métodos de Medición

La disponibilidad de un sistema suele medirse usando los siguientes parámetros [27]:

- ***Mean Time Between Failures (MTBF)*:** Conocido también como *Uptime*.
- ***Mean Time to Repair (MTTR)*:** También conocido como *Downtime*.

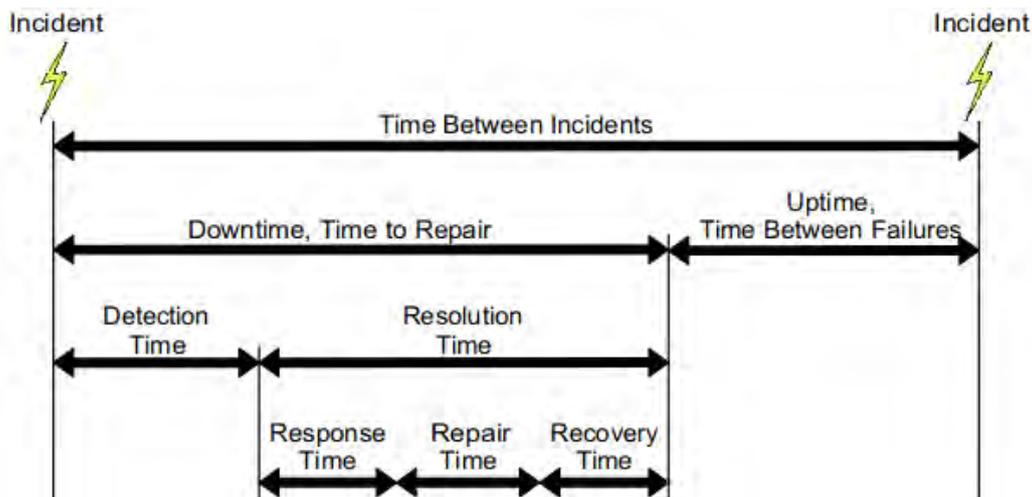


Figura 8. Parámetros influyentes en el Downtime

Fuente: [11]

De esta forma, la disponibilidad queda definida por la fórmula:

$$\text{Disponibilidad} = 1 - \frac{MTTR}{MTBF}$$

Ecuación 1. Fórmula de la Disponibilidad

Fuente: [11]

En la mayoría de los casos, la industria se enfoca en el MTTR para las soluciones de software, mientras que usa el MTBF para evaluar al hardware. De esta forma, un *Downtime* de 526 minutos al año implica una disponibilidad del 99.9%, mientras que una de 5 minutos tiene una disponibilidad del 99.999%. [27]

2.2.4 Aplicaciones Móviles

A lo largo de los años, aparecieron múltiples Sistemas Operativos (SOs) Móviles. Los primeros celulares con soporte de aplicaciones, y que no solo servían para hacer y recibir llamadas, usaban un SO llamado Symbian OS. Este fue desarrollado en un esfuerzo conjunto por Nokia, Ericsson y Motorola a finales de los 90, con el objetivo de tener un SO común e interoperable, pero con la suficiente capacidad de personalización para poder diferenciar a los

productos de los distintos fabricantes [28]. Con el auge de los smartphones en el 2007 tras el lanzamiento del iPhone, este sistema fue lentamente desplazado.

2.2.4.1 Sistemas operativos

De esta forma, en el mercado actual, se encuentran predominando los siguientes SO:

- **Android:**

“Android es un sistema operativo de código abierto para dispositivos móviles y un proyecto de código abierto correspondiente dirigido por Google”. [29] Este fue desarrollado a partir del kernel de Linux, y fue usado por primera vez en el año 2008. Su objetivo es el de desarrollar un SO móvil descentralizado, donde una sola empresa no puede tomar el control del desarrollo ni la dirección del proyecto.

Asimismo, el proyecto Android provee el código fuente y documentación de manera libre y gratuita, así como facilita el desarrollo de aplicaciones para su plataforma. Las aplicaciones nativas en Android se desarrollan en Java, y más recientemente, Kotlin. La última versión del SO existente hasta la fecha de este trabajo es Android 13.

- **iOS:**

Por otro lado, iOS es un sistema operativo propietario desarrollado por Apple, y empleado en toda su gama de dispositivos móviles. Este se lanzó en el año 2007 junto con el primer iPhone. [30]

Para el desarrollo de aplicaciones nativas en iOS se puede usar C, C++, Objective-C, y más recientemente, Swift. Sin embargo, debido a su naturaleza cerrada, la creación de aplicaciones en esta plataforma es más complicada, requiriéndose una Mac y el pago de licencias para poder acceder a las herramientas necesarias para desarrollar y

distribuir la aplicación. La última versión del sistema disponible hasta la fecha es iOS 17.

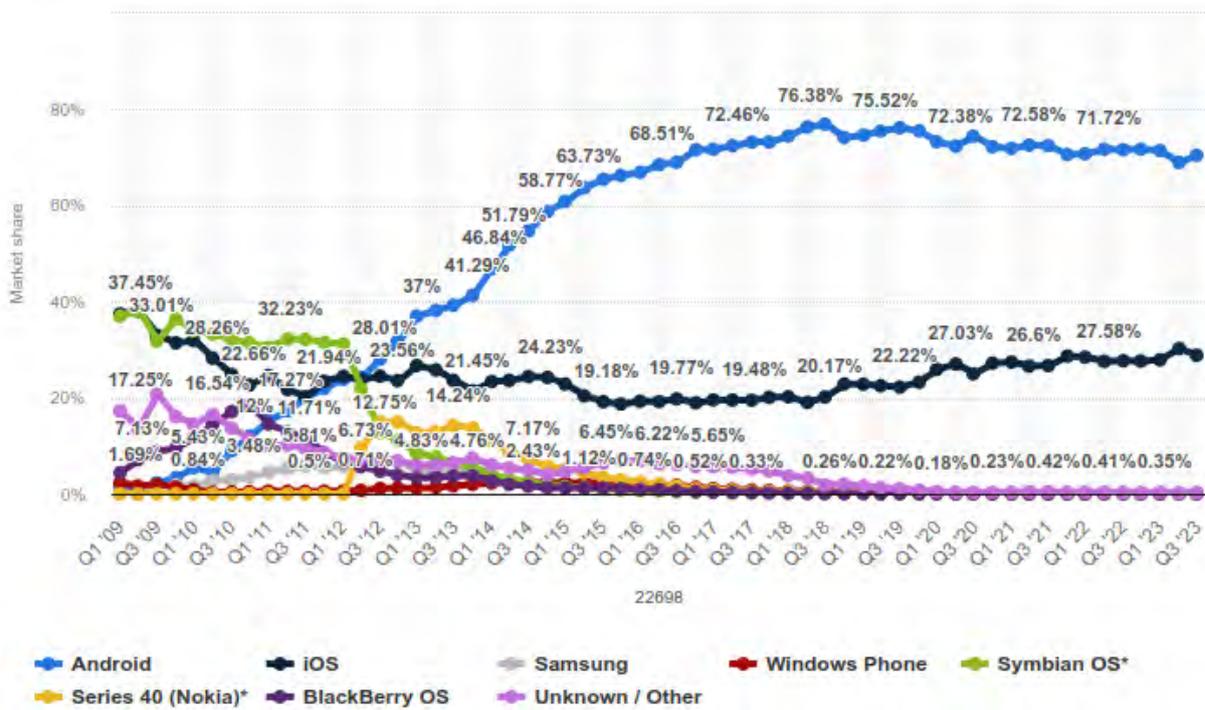


Figura 9. Línea de Tiempo del uso de los SO Móviles Fuente: [31]

Q3 '23	
Android	70.46%
iOS	28.83%
Samsung	0.37%
Windows Phone	0.02%
Symbian OS*	0%
Series 40 (Nokia)*	0.01%
BlackBerry OS	0%
Unknown / Other	0.15%

Figura 10. Cuota de Mercado de los SO Móviles Fuente: [31]

2.2.4.2 Tipos de Aplicaciones

El tipo de aplicación móvil hace referencia al conjunto de herramientas que permiten el desarrollo y la compilación de la aplicación. El elegir un tipo u otro puede significar tener una aplicación más rápida y confiable, una aplicación más fácil de desarrollar, u otros. A continuación, se explicarán los distintos tipos de aplicaciones móviles que se pueden encontrar en el mercado actual. [32]

- **Aplicaciones Nativas:** Estas son “las aplicaciones que son específicas para un determinado dispositivo o plataforma, como Android o iOS” [32]. Estas emplean los lenguajes y librerías propias de su respectivo SO, por lo que pueden aprovechar al máximo todas sus características. Sin embargo, se debería desarrollar otra aplicación aparte para soportar a otro SO.
- **Aplicaciones Multiplataforma:** Se basan en plataformas que permitan compilar un código fuente común a varios formatos de aplicación, de tal forma que se puedan ejecutar en distintos SO. La gran desventaja de este tipo de sistemas es la mayor complejidad de desarrollo para poder trabajar con la plataforma compiladora.
- **Aplicaciones Web Progresivas:** Estas aplicaciones se ejecutan dentro del navegador web del equipo móvil. De esta forma, permiten el uso de JS, CSS y HTML (herramientas estándar para el desarrollo de aplicaciones web) y la interacción con el SO del teléfono por medio del SDK del navegador. Este tipo de aplicación permiten a desarrolladores web empezar a desarrollar aplicaciones en equipos móviles con una menor curva de aprendizaje.
- **Aplicaciones Híbridas:** Este tipo de aplicaciones, por medio de un *framework*, permiten crear contenedores de aplicaciones web dentro del SO del equipo móvil, permitiendo así una mejor integración con el hardware y las funcionalidades del equipo. A pesar de estas ventajas, el uso de contenedores y capas de intermediarias

para la comunicación entre la aplicación y el SO afectan negativamente al rendimiento de la aplicación.

2.2.5 APIs

Una *Application Programming Interface* (API) es un conjunto de reglas definidas que permiten la interacción entre dos aplicaciones diferentes. [20] Estas permiten a las compañías abrir sus servicios e información para una integración sencilla con plataformas internas, clientes y desarrolladores independientes. En un contexto real, una API se encarga de recibir la solicitud (*request*) del usuario que la consume, relegando la información recibida a un servidor que la procesa. Finalmente, el servidor envía una respuesta (*response*) hacia el usuario a través de la misma API. [33]

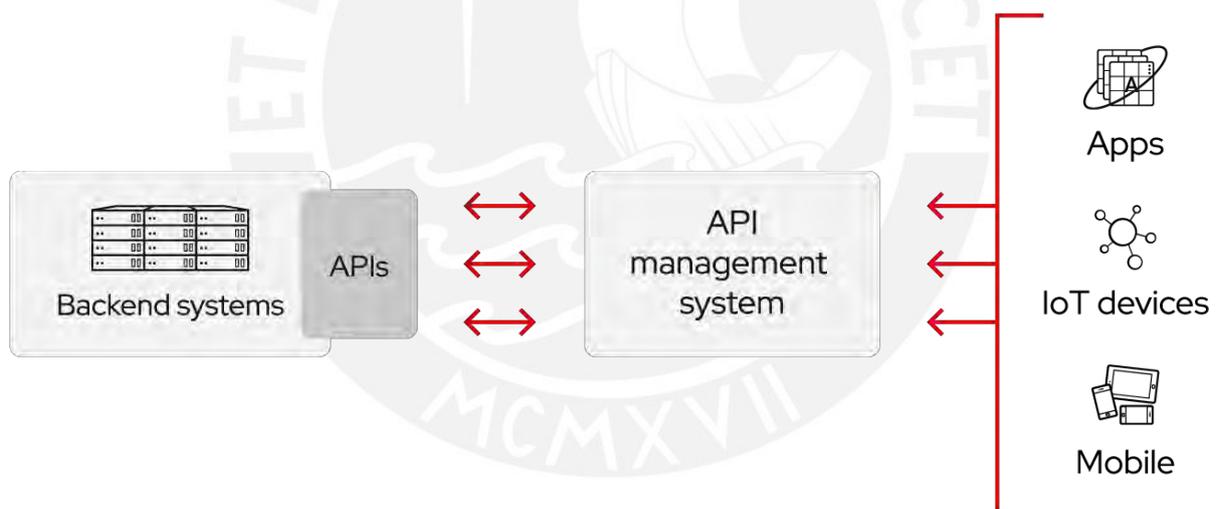


Figura 11. Funcionamiento de una API
Fuente: [33]

2.2.6.1 Especificaciones de APIs

Mientras el modelo de APIs empezó a volver cada vez más usado por las aplicaciones, se evidenció la importancia de estandarizar el intercambio de información. De esta forma, surgieron 2 (dos) especificaciones [20], [33]:

- **SOAP:** La especificación *Simple Object Access Protocol* (SOAP) permitió el intercambio de información por medio de archivos XML a través de HTTP o SMTP. Esta fue novedosa pues permitía el intercambio de información entre aplicaciones que operaban en diferentes entornos o lenguajes.
- **REST:** Por otro lado, la especificación *Representational State Transfer* (REST) no es un protocolo, sino un conjunto de principios de arquitectura, los cuales indican:
 - La arquitectura debe componerse por clientes, servidores y recursos. Además, la comunicación debe darse por medio de HTTP.
 - No se guarda información en el servidor entre peticiones. La información se queda del lado del cliente.
 - Se puede usar cacheado para evitar interacciones innecesarias.
 - Puede haber capas intermedias entre las interacciones del cliente, como balanceado de cargas, seguridad y cacheo.
 - Se deben usar interfaces uniformes, las cuales son fácilmente identificables entre sí, trabajan con recursos, y son autodescriptivas.

2.2.6.2 Usos de APIs

Las APIs se usan en un amplio y diverso rango de aplicaciones actuales, entre ellas:

- **Servicios de logueo:** Permite la integración entre diferentes plataformas (Google, Facebook, etc.).
- **Redes IoT:** La comunicación entre dispositivos se realiza mediante estas interfaces.
- **Aplicaciones de catálogos:** Permite la búsqueda y comparación entre servicios de diferentes plataformas (viajes, hoteles, eventos, etc.).

- **Mapas:** Las aplicaciones de mapas obtiene información sobre direcciones o establecimientos mediante el uso de APIs.
- **SaaS:** Las aplicaciones *Software as a Service* normalmente usan APIs para integrar sus funcionalidades con la infraestructura existente de una empresa.



Capítulo 3. Diseño e implementación de la solución con despliegue en la nube

Habiendo explorado los conceptos y las soluciones relevantes al enfoque de esta tesis, en este capítulo se describirán los requerimientos planteados para el desarrollo de la aplicación, así como el proceso de diseño e implementación de la solución propuesta.

3.1 Requerimientos de la Solución

En esta sección se señalarán los requerimientos funcionales que debe de cumplir la solución final.

Tabla 7. Requisitos funcionales de la solución

N°	Requerimientos	OB/OP
1	El usuario podrá iniciar sesión usando correo y contraseña.	Obligatorio
2	El sistema tendrá los siguientes roles: - Cliente/Contratista - NOC - Mantenimiento - Administrador	Obligatorio
3	Al instalar la aplicación se crea un usuario administrador por defecto	Obligatorio
4	Al iniciar sesión, el usuario podrá: - Gestión de usuarios	Obligatorio

	<ul style="list-style-type: none"> - Gestionar roles - Gestionar de SLAs 	
5	<p>Al crear un usuario, se especifican los siguientes campos:</p> <ul style="list-style-type: none"> - Nombres - Apellidos - Correo - Teléfono - Rol - Organización - Ubicación - Contraseña 	Obligatorio
6	<p>Se pueden modificar los datos de los usuarios:</p> <ul style="list-style-type: none"> - Nombres - Apellidos - Correo - Teléfono - Rol - Organización - Ubicación - Contraseña 	Obligatorio
7	<p>Se pueden dar de baja a usuarios del sistema, mediante un estado de deshabilitado, y mediante un borrado lógico.</p>	Obligatorio
8	<p>Al crear un SLA, se definen los siguientes campos:</p> <ul style="list-style-type: none"> - Nombre - Plazo de Tiempo en horas - Descripción (opcional) 	Obligatorio
9	<p>Luego de crear el SLA, el usuario puede automatizar un ticket indicando:</p> <ul style="list-style-type: none"> - Prioridad asignada (Baja, media, alta) - Rol del creador - Departamento - Fecha de creación 	Obligatorio
10	<p>Se pueden editar los siguientes campos del SLA, siempre y cuando no esté siendo usado.</p> <ul style="list-style-type: none"> - Nombre - Plazo de Tiempo en horas - Descripción (opcional) 	Obligatorio
11	<p>Un SLA no puede ser eliminado mientras sea usado en algún ticket.</p>	Obligatorio
12	<p>Los clientes/contratistas y trabajadores en el NOC pueden crear tickets de atención especificando los siguientes criterios:</p> <ul style="list-style-type: none"> - Título - Descripción - Archivos Adjuntos (pdf, docx, xlsx, pptx, png, jpeg, txt, etc.) de hasta un máximo de 20 MB 	Obligatorio
13	<p>Tras haberse creado el ticket, los trabajadores del NOC y de mantenimiento pueden agregar los siguientes campos:</p> <ul style="list-style-type: none"> - Asignación manual de SLA - Asignación de Departamento - Designación de agente encargado para el ticket (1 o más) - Asignación de permisos a trabajadores (visualización, modificación, participación) 	Obligatorio

	- Prioridad (Baja/media/alta)	
14	Un ticket tendrá un hilo de conversaciones tipo foro y actualizaciones sobre las operaciones del mantenimiento entre todos los participantes	Obligatorio
15	Los usuarios con permisos de visualización en un ticket podrán visualizar los contenidos de su hilo de conversación	Obligatorio
16	Una Todo List puede ser visualizada por los participantes con permiso de visualización en el ticket.	Obligatorio
17	Los ítems de esta Todo List pueden ser marcados como completados por el trabajador del NOC y Mantenimiento que tengan el permiso de modificación.	Deseable
18	Un ticket, al satisfacerse todos sus objetivos, podrá ser cerrado por un miembro del NOC con permiso de modificación.	Obligatorio
19	Los miembros del NOC y del equipo de mantenimiento pueden observar en su ventana principal un dashboard que muestre información actual como: - Tickets abiertos - Tickets sin ser asignados - Tickets por vencer	Obligatorio
20	Se tendrá una aplicación móvil para Android, la cual usará la API de RT, y permitirá realizar el seguimiento de los tickets.	Deseable
21	Los roles soportados por la aplicación son: - NOC - Mantenimiento	Deseable
22	El usuario iniciará sesión indicando su usuario, contraseña, y la dirección IP o el URL del servidor con RT ejecutándose.	Deseable
23	Los usuarios/contratistas podrán: - Visualizar un listado de sus tickets activos - Presionar un botón y crear un nuevo ticket mediante un formulario. - Seleccionar un ticket y ver el historial de mensajes y escribir uno nuevo	Deseable
24	Los operarios del NOC podrán: - Visualizar un listado de los tickets creados, y con permiso de visualización - Presionar un botón y crear un nuevo ticket mediante un formulario. - Seleccionar un ticket y ver el historial de mensajes. - Si se tiene permiso de participación, se puede escribir un nuevo mensaje. - Se tendrá una opción para cerrar el ticket	Deseable
25	Los miembros del personal de mantenimiento podrán: - Visualizar un listado de los tickets asignados con permiso de visualización - Seleccionar un ticket y ver el historial de mensajes. - Si se tiene permiso de participación, se puede escribir un nuevo mensaje. - Tendrá una opción para cerrar el ticket	Deseable

3.2 Herramientas Utilizadas

3.2.1 Request Tracker

Request Tracker (RT) es un sistema de *ticketing* proporcionado por la empresa *Best Practical*.

Este es de código abierto y proporciona de una plataforma altamente configurable de acuerdo

a las necesidades de la empresa. RT ofrece planes de servicio en la nube a través de una membresía, y el servicio para usarlo en un servidor propio. [41]

Asimismo, RT ofrece un amplio catálogo de funcionalidades, como las siguientes:

- Soporte en equipos móviles (aplicación y de escritorio (interfaz web)).
- Integración con correo electrónico.
- Manejo de grupos, usuarios y roles.
- Funcionalidad de gestión de ticket y de inventario
- Administración de SLA y de horario laboral.
- Dashboards personalizables y exportación de datos.
- Herramientas de automatización de procesos.
- API detallada para la interacción con otros programas
- Gran variedad de extensiones.

The screenshot displays the Request Tracker (RT) main dashboard. At the top, there is a navigation bar with links for Home, Search, Reports, Assets, Tools, and Logged in as artie. A search bar and a 'Create new ticket' button are also present. The main content area is titled 'RT at a glance' and contains several sections:

- 10 highest priority tickets I own (Found 2 tickets):** A table with columns for #, Subject, Priority, Queue, and Status. Two tickets are listed: #492 'Test new Bootstrap themes in RT' and #501 'Publish blog post about upcoming release of RT 5'.
- 10 newest unowned tickets (Found 401 tickets):** A table with columns for #, Subject, Queue, Status, and Created. Four tickets are listed with subjects like 'perspiciatis facere dolores ab rem eum doloremque est voluptas saepe'.
- My reminders:** A section for managing reminders.
- Queue list:** A table showing ticket counts for different queues. The 'General' queue has 300 new tickets and 105 open tickets.
- Dashboards:** A section for managing dashboards.
- Refresh:** A button to refresh the page, with a 'Go!' button below it.

Figura 12. Página Principal de Request Tracker
Fuente: [41]

Se optó por usar este programa ya que, de acuerdo a un estudio previo [1], es el que tenía una comunidad más activa, una mejor documentación y soporte, además de tener funcionalidades

que se amoldan con las requeridas para este trabajo. La última versión disponible al momento de realizar este trabajo es la 5.0.5.

3.2.2 Draw.io

Draw.io es un software para la creación de diagramas de todo tipo. Dada su interfaz simple, su fácil acceso a través de un navegador, y sus múltiples integraciones en otros programas, este ha sido el programa elegido para diseñar los diferentes diagramas que se verán a lo largo del siguiente trabajo, incluyendo la arquitectura de la solución y otros esquemas.

3.2.3 AWS

Como ya se habló en la sección 2.2.2.1, AWS es un proveedor de *Cloud Computing*, el cual será usado en este trabajo para el despliegue de recursos en la nube. Se optó por recurrir a este proveedor pues es el más predominante en el mercado y, por ende, el que más probabilidad tiene de ser adoptado en un entorno empresarial. Asimismo, ofrece todos los servicios necesarios para este despliegue, además de una buena documentación.

Entre los servicios a usar se tienen:

- **Amazon Route 53:** Es un servicio web ofrecido parte de AWS el cual se encarga de proveer un servidor DNS de alta disponibilidad y escalabilidad. Entre sus funcionalidades este permite el registro de dominios, el enrutamiento DNS, y el monitoreo de los recursos vinculados al servicio. [34]
- **Amazon Cloudfront:** Es otro servicio web, el cual ofrece una *Content Delivery Network* (CDN), compuesta por cientos de PoPs (puntos de presencia por sus siglas en inglés) a lo largo del mundo. Con esta red, el servicio permite el cacheado de recursos provenientes tanto de servidores web como de buckets S3. Gracias a esto, se

disminuye la latencia de las comunicaciones entre usuario y servidor, así como la cantidad de peticiones que llegan a los servidores centrales. [35]

- **Amazon Load Balancer:** Un balanceador de carga distribuye el tráfico entrante a un servicio entre un grupo de objetivos, pudiendo estos ser recursos pertenecientes a la nube, o externos. Este elemento permite agrupar recursos físicos y en la nube, aumentando la flexibilidad y escalabilidad del servicio brindado. [36]
- **Amazon EC2:** Las instancias EC2 son recursos de cómputo en la nube, los cuales siguen un modelo de IaaS. Para la configuración de este servicio, AWS ofrece diversos sistemas operativos, y distintos niveles de recursos según las necesidades del usuario. De esta forma, el servicio permite el despliegue rápido y seguro de aplicaciones. [37] Durante este trabajo, las instancias EC2 se usarán para alojar al servidor de tickets, a sus archivos de configuración, y a su servidor web.
- **Amazon RDS:** Es un servicio PaaS, con el cual se puede crear y administrar una base de datos relacional. Entre las opciones se tienen bases de datos como MySQL, MariaDB, Amazon Aurora, PostgreSQL y Microsoft SQL Server. Esta es una manera más rápida y sencilla de tener una base de datos accesible y actualizada. [38] En este trabajo se usará para almacenar la información de la plataforma de tickets.
- **Amazon S3:** Es un servicio de almacenamiento de objetos, los cuales pueden ser usados para cualquier propósito. Esta solución ofrece seguridad, disponibilidad y escalabilidad para el acceso a los datos. [40] Para el caso de este trabajo, aquí se guardarán tanto archivos adjuntos como imágenes que puedan estar en el hilo de un ticket.

Para el caso de este trabajo de tesis, se usó la plataforma de AWS Academy. A través de esta plataforma se tuvo acceso a un laboratorio de pruebas, el cual dispone de \$100 en créditos para el acceso a una gran cantidad de recursos de AWS, incluyendo a los anteriormente

mencionados, con fines educativos. Al iniciar una sesión de laboratorio, esta dura 3 horas, tras las cuales los servicios como instancias EC2 y buckets S3 dejarán de funcionar, y deberán ser inicializados nuevamente. Sin embargo, recursos como Amazon RDS y Route 53 siguen operativos hasta ser eliminados o detenidos manualmente. Cabe resaltar que este entorno posee ciertas limitaciones en cuanto al uso de algunos servicios [44], por lo que ciertas funcionalidades como el registro de un nuevo dominio están bloqueadas, y se tuvieron que usar servicios externos.

3.2.4 CloudDNS

CloudDNS es un proveedor de servicios DNS centralizados alrededor de una arquitectura en la nube. Esta les otorga a los servicios de la empresa una gran velocidad de funcionamiento, así como redundancia y tolerancia a fallos. Asimismo, ofrece una plataforma para el monitoreo y administración y optimización de los servicios ofrecidos. Para el caso de este trabajo, se usará su nivel gratuito para crear la zona DNS de la que se obtendrá el dominio para la solución.

3.2.5 Figma

Figma es una plataforma de diseño vectorial accesible desde Internet. Para este trabajo, Figma será usado para el diseño de los mockups correspondientes a las diferentes vistas de la aplicación móvil.

3.2.6 Android Studio

Debido a tener un mercado más extenso, y por lo ende, ser más usado, además de tener un ecosistema de desarrollo más abierto, se optó por desarrollar una app nativa para Android. Por esto mismo, el IDE recomendado, y el que tiene mayor soporte, es Android Studio. Android Studio a su vez permite el desarrollo de programas con 2 lenguajes de programación: Java y Kotlin. Entre ambas opciones, se optó por emplear Java, pues es un lenguaje con una amplia

documentación y es bastante usado en la industria, además de que al momento de desarrollar esta tesis tenía una mayor familiaridad con este. De esta forma, será usado para desarrollar y realizar pruebas en la aplicación móvil.

3.2.7 Hoppscotch

Hoppscotch es un programa de código abierto que sirve para diseñar, depurar, y consumir APIs. Se eligió este programa en vez de otras alternativas como Postman o Insomnia debido a ser completamente gratis y no tener funcionalidades, como la sincronización entre dispositivos, bloqueadas por un plan de pago. Además, su interfaz es simple y de fácil acceso a través de cualquier navegador.

3.2.8 Apache Jmeter

Apache Jmeter es un programa escrito en Java, cuya principal función es la de evaluar el funcionamiento y el rendimiento de una aplicación o servicio ante una alta carga de tráfico. [39] Su función en este trabajo será la de evaluar las funciones de alta disponibilidad de la solución final.

3.3 Setup Local del Servicio

Para realizar algunas pruebas y configuraciones con la herramienta Request Tracker, se optó por primero realizar una instalación local de la herramienta en una máquina virtual con características similares a la de la instancia EC2 de AWS en la que se desplegaría eventualmente. De esta forma, se levantó una máquina virtual con Ubuntu 22.04 LTS. Tras algunas pruebas, se determinó que los requerimientos mínimos del equipo para realizar una instalación funcional eran de 2 núcleos del procesador, 4 GB de RAM y 8 GB de almacenamiento.

El proceso de instalación y configuración básica, así como la conexión a base de datos, se registró y automatizó empleando un conjunto de scripts en bash. Estos fueron los siguientes:

3.3.1 Instalación de las dependencias

Antes de instalar RT en el equipo es necesario que sus principales dependencias se encuentren instaladas en el equipo. Para eso se tiene el siguiente script de bash:

```
#!/bin/bash

# Se actualiza el sistema
sudo apt update -y
sudo apt upgrade -y

# Se instalan dependencias básicas como Apache y un cliente MariaDB
sudo apt install build-essential apache2 libapache2-mod-fcgid \
  libssl-dev libexpat1-dev libmysqlclient-dev libcrypt-ssleay-perl \
  liblwp-protocol-https-perl mariadb-server mariadb-client -y

# Se realiza una configuración básica de Perl
sudo /usr/bin/perl -MCPAN -e shell

# Se instalan las dependencias de Perl necesarias para el proyecto
sudo cpan install HTML::FormatText HTML::TreeBuilder \
  HTML::FormatText::WithLinks HTML::FormatText::WithLinks::AndTables \
  DBD::mysql LWP::Protocol::https
```

Archivo 1. preparar.sh

3.3.2 Instalación básica de RT

Teniendo las dependencias listas, se procede con la instalación de RT y de todos sus módulos esenciales. Posteriormente se le da una configuración básica y se levanta el servicio dentro de la máquina virtual de prueba.

```
#!/bin/bash

# Variables importantes
dbHost="localhost"
dbPwd="password"
IP="localhost"
srcDir=$(pwd)

# Descarga y descomprime el código fuente
cd /tmp
wget https://download.bestpractical.com/pub/rt/release/rt-5.0.5.tar.gz
```

```

tar xzf rt-5.0.5.tar.gz
[ -d rt-5.0.5/ ] && rm -rf rt-5.0.5.tar.gz || exit
cd rt-5.0.5/

# Configuración de las dependencias
sudo ./configure

# Se verifica si existen las dependencias
if ! sudo make testdeps; then
    yes | sudo make fixdeps
fi

# Se realiza la instalación
if ! sudo make testdeps; then
    exit
fi
sudo make install

# Se carga la configuración inicial de RT
echo "
# Redirección
Set(\$WebSecureCookies, 0);
Set(\$CanonicalizeRedirectURLs, 1);
Set(\$CanonicalizeURLsInFeeds, 1);

# Logs
Set(\$LogToSyslog, 'warning');
Set(\$LogToSTDERR, 'warning');

# Perl expects to find this 1 at the end of the file.
1;
" | sudo tee /opt/rt5/etc/RT_SiteConfig.d/basic_config.pm

# Configuraciones del servidor web
echo "
# Configuración Web
Set(\$WebDomain, q{\$IP});
1;" | sudo tee /opt/rt5/etc/RT_SiteConfig.d/domain_config.pm

# Se registran las credenciales de la DB
echo "
# Configuración de la Base de Datos
Set(\$DatabaseHost, q{\$dbHost});
Set(\$DatabaseRTHost, q{\$dbHost});
Set(\$DatabaseAdmin, q{admin});
Set(\$DatabasePassword, q{\$dbPwd});
1;" | sudo tee -a /opt/rt5/etc/RT_SiteConfig.d/db_config.pm

# Se inicializa la base de datos (Solo lo debe hacer la primera máquina)
sudo mysql -h \$dbHost -u admin -p -e "ALTER USER 'rt_user'@'%' IDENTIFIED BY '\${dbPwd}';commit;"
sudo mysql -h \$dbHost -u admin -p -e "grant all privileges on *.* to rt_user;flush privileges;"
sudo make dropdb
sudo make initialize-database

# Se realiza la prueba de rt5
sudo /opt/rt5/sbin/rt-server --port 8080

```

Archivo 2. instalar-rt.sh

Al culminar esta configuración, como se observa en la última línea del fragmento de código mostrado, es posible acceder a una interfaz web de pruebas a través del puerto 8080 de la

instancia. A continuación se muestra la vista obtenida al realizar una conexión a través del navegador.

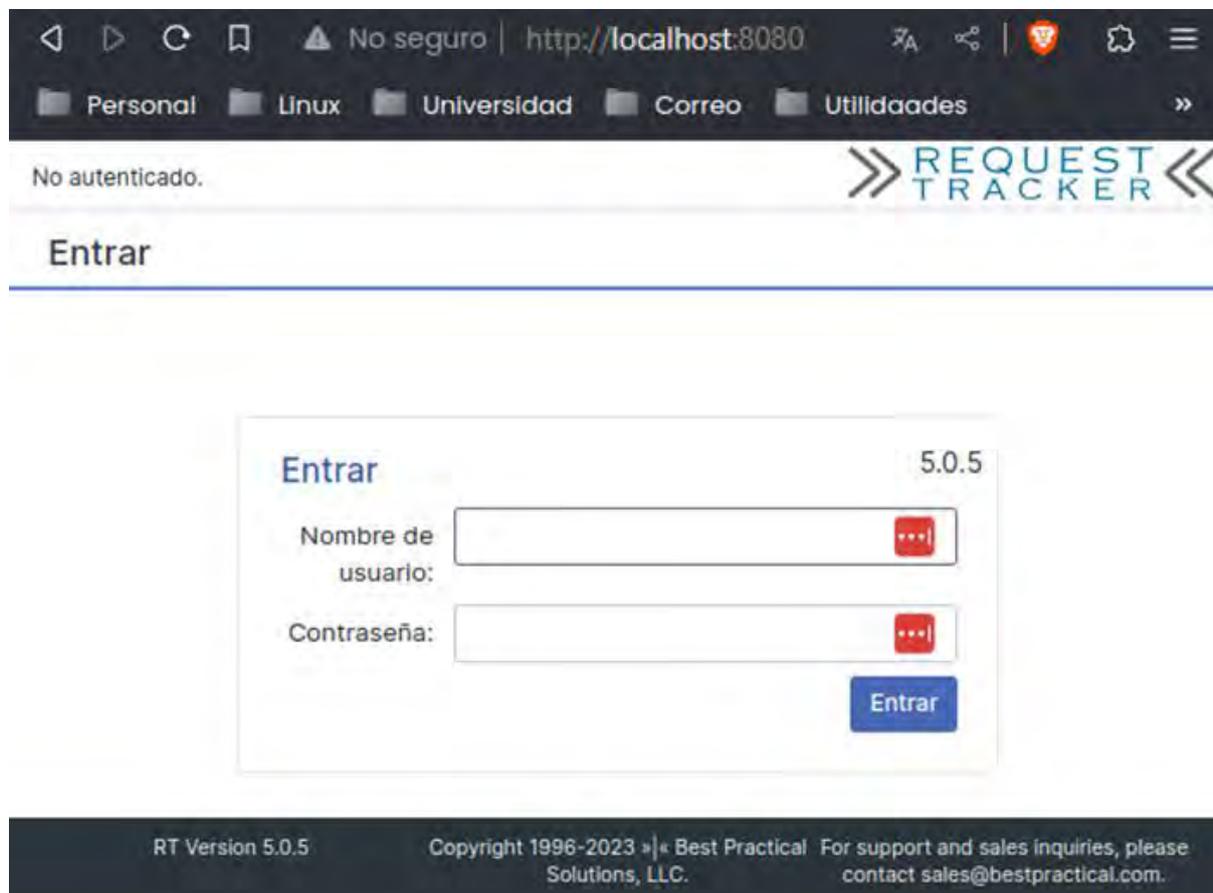


Figura 13. Vista inicial al conectarse al servidor de prueba de RT
Fuente: Elaboración propia

3.3.3 Configuración del servidor Apache

Para poder acceder a todas las funcionalidades del programa, así como a la encriptación por SSL, es necesario configurar un servidor web dedicado, ya sea Apache o Nginx. Para este caso se eligió Apache, ya que es el servidor web recomendado por los desarrolladores de RT.

De esta forma, su configuración es la siguiente:

```
#!/bin/bash

# Variables importantes
srcDir=$(pwd)

# Crear el archivo del sitio de RT5
echo '
<VirtualHost *:80>
# Ensure that your log rotation scripts know about these files
```

```

ErrorLog /opt/rt5/var/log/apache2.error
TransferLog /opt/rt5/var/log/apache2.access
LogLevel debug

AddDefaultCharset UTF-8

ScriptAlias / /opt/rt5/sbin/rt-server.fcgi/

ServerName rt5
ServerAlias rt5.tesis.cloudns.ph

DocumentRoot "/opt/rt5/share/html"
<Location />
    Require all granted
    Options +ExecCGI
    AddHandler fcgid-script fcgi
</Location>
</VirtualHost>' | sudo tee /etc/apache2/sites-available/rt5.conf

# Cargar apache y deshabilitar el sitio por defecto
sudo a2ensite rt5
sudo a2dissite 000-default

# Configuración para permitir el uso de la API
sudo sed -i '10i SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1' /etc/apache2/sites-enabled/rt5.conf

# Cargando módulos de apache
sudo a2dismod mpm_event
sudo a2dismod mpm_worker
sudo systemctl restart apache2
sudo a2enmod mpm_prefork
sudo systemctl restart apache2

# Activar HTTP 2
sudo a2enmod http2

## Corrección mod_fcgid.conf
sudo sed '3 i FcgidMaxRequestLen 1073741824' /etc/apache2/mods-available/fcgid.conf

# Reiniciar apache
sudo systemctl reload apache2
sudo systemctl restart apache2

```

Archivo 3. configure-apache.sh

De igual forma que en el caso anterior, el contenido del archivo `rt5.conf` se puede encontrar en los anexos. De esta forma, tras esta configuración, se observa que ya es posible realizar una conexión HTTP a través del puerto 80 con el servidor, usando el servidor Apache.

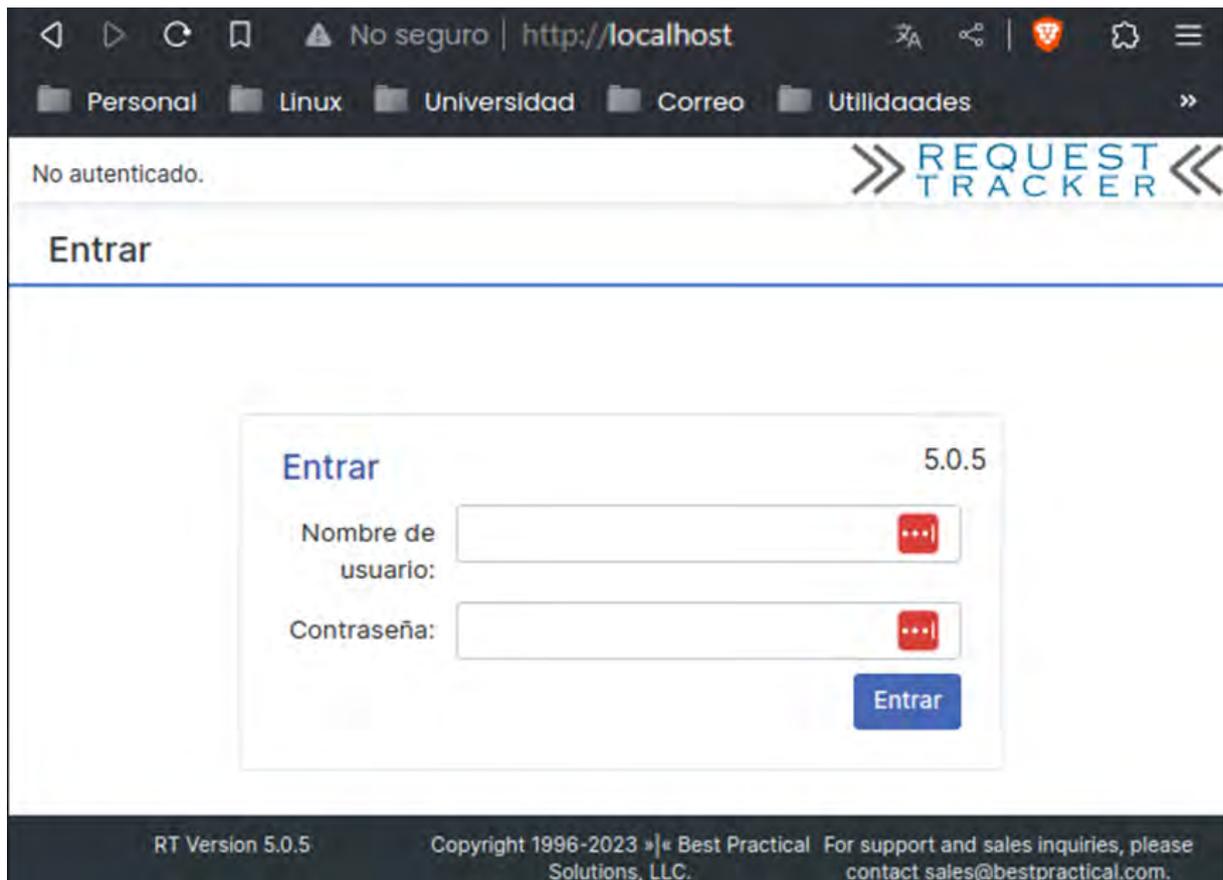


Figura 14. Conexión al servidor a través de HTTP usando el servidor Apache
Fuente: Elaboración propia

3.4 Despliegue de Arquitectura en la Nube

3.4.1 Arquitectura Propuesta

La arquitectura propuesta para el despliegue en la nube es la siguiente:

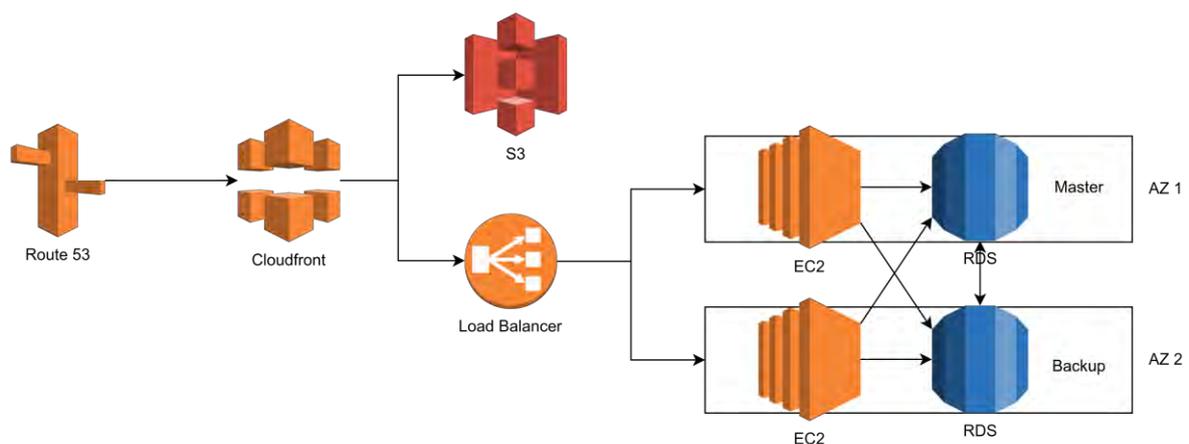


Figura 15. Arquitectura propuesta
Fuente: Elaboración propia

Esta arquitectura se encuentra enfocada en brindar una alta disponibilidad. Para lograr esto, los puntos claves son:

- **Balanceador de Cargas:** Permite repartir el tráfico entrante entre 2 instancias EC2, estando cada una alojada en una zona de disponibilidad distinta.
- **Base de Datos:** La base de datos está configurada con un Backup en otra zona de disponibilidad. La configuración es de tipo Activa-Pasiva, es decir, al conectarse al *endpoint* de la DB, el tráfico es redirigido siempre a la misma base de datos “*master*”, estando la otra DB “*backup*” en reposo. Sin embargo, al momento de existir cualquier falla con el *master*, el *backup* toma su lugar de forma inmediata, ya que tiene sus contenidos siempre sincronizados con la otra.
- **Cloudfront:** El servicio CDN de Amazon Cloudfront permite realizar el cacheado de algunos recursos del servidor web, brindándolos de forma más rápida a los usuarios. Gracias a esto, se aligera también la carga de procesamiento de los servidores de aplicación.

3.4.2 Configuración de los Servicios

a. Amazon RDS

Se creó una instancia RDS con replicación Activo - Pasivo, para alojar la información que será empleada por la aplicación de Request Tracker. A continuación, se muestran las configuraciones que fueron aplicadas a la DB.

Tabla 8. Atributos de la instancia RDS

Grupo	Atributo	Valor
Características Físicas	Tipo de instancia	db.t3.small (2 vCPU y 2 GB de RAM)
	Almacenamiento	20 GB
Características de	Backend	MySQL 8.0.33

la DB	Puerto	3306
	Usuario principal	admin
Configuraciones de AWS	Nombre	database-1
	Zona de disponibilidad	us-east-1d
	Réplicas	1 (Pasiva)
	Reglas de Entrada	Únicamente las instancias EC2 registradas.

Fuente: Elaboración propia

b. Amazon EC2

Las instancias Amazon EC2 fueron usadas para alojar y desplegar el servicio de Request Tracker. Debido a los requerimientos de hardware para el servidor de Request Tracker, que necesita al menos 4 GB de memoria RAM para poder ser inicializado correctamente, no se pudieron emplear instancias t2.micro ni t2.small, que poseen solo 1 y 2 GB de RAM respectivamente. A las instancias empleadas para esta función se les atribuyeron las siguientes características.

Tabla 9. Atributos de la instancia EC2

Grupo	Atributo	Valor
Características de la Instancia	Tipo de instancia	t2.medium (2 vCPU y 4 GB de RAM)
	Almacenamiento	20 GB
	Imagen de SO (AMI)	Ubuntu 22.04 LTS / Request Tracker
Configuraciones de AWS	Nombre	rt-1 y rt-2
	Reglas de Entrada	Permiten acceso desde cualquier equipo: <ul style="list-style-type: none"> • SSH (Puerto 22) • HTTP (Puerto 80) • HTTPS (Puerto 443) • Pruebas (Puerto 8080)
	Zona de disponibilidad	us-east-1c y us-east-1e

Fuente: Elaboración propia

Una vez creada la instancia, se pasa a desplegar el servicio de Request Tracker ahí, teniendo como base a lo scripts de la sección 3.3. Sin embargo, cabe resaltar que se realizaron algunas modificaciones en las variables iniciales, para diferenciar una instalación local de una en la nube. De esta forma, se actualizó el hostname y la contraseña de la DB creada en la sección anterior, y se obtuvo de forma dinámica la dirección IP pública de la instancia EC2.

```
# Variables importantes
dbHost="database-1.cgsgmbarsagq.us-east-1.rds.amazonaws.com"
dbPwd="H2zLNKqf"
IP=$(curl http://checkip.amazonaws.com)
```

Archivo 4. instalar-rt.sh modificado

Una vez se finalizó el proceso de instalación y configuración en la primera instancia, se procedió a crear una plantilla para reducir el tiempo de levantar el servicio en una nueva instancia, teniendo así el software necesario y una configuración por defecto inmediatamente tras ser creadas. Asimismo, el uso de estas plantillas trae la posibilidad de introducir flujos de automatización y escalabilidad, introduciendo por ejemplo al servicio de Amazon EC2 Auto Scaling [43], que permite escalabilidad horizontal de cómputo de acuerdo a la demanda.

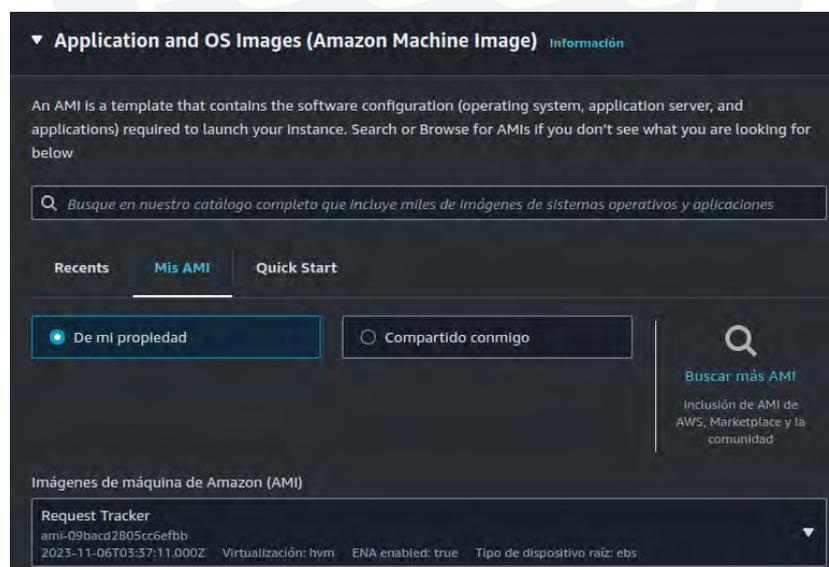


Figura 16. Creación de instancia EC2 usando una plantilla personalizada
Fuente: Elaboración propia

De esta forma, se acaban teniendo 2 instancias EC2, cada ejecutando el servicio de ticketing, y que se comunican con la misma base de datos para mostrar información sincronizada.

Name	ID de la ins...	Estado de la i...	Tipo de ...	Comproba...	Estado de l...	Zona de...	DNS de ...	Direcció...
rt-2	i-050c4aa7...	En ejecución	t2.medium	2/2 comprob	Sin alarmas+	us-east-1c	ec2-54-90-...	54.90.97.207
rt-1	i-0d3f6b29...	En ejecución	t2.medium	Inicializando	Sin alarmas+	us-east-1e	ec2-3-80-1...	3.80.18.178

Figura 17. Instancias EC2 creadas
Fuente: Elaboración propia

c. Amazon Load Balancer

Para el servicio de balanceador de carga, AWS ofrece distintas opciones, de acuerdo al tipo de tráfico y al destino de este. De forma concisa, se tiene:

- **Balanceador de Aplicación:** Opera sobre la capa de aplicación (capa 7), estando optimizado para el tráfico HTTP y HTTPS.
- **Balanceador de Red:** Opera en la capa de transporte (capa 4), y es capaz de trabajar con tráfico TCP y SSL. Este es más flexible y configurable, a costa de ser más complejo.

Ya que el objetivo es el de balancear la carga de tráfico HTTP y HTTPS entre 2 instancias EC2, se opta por crear un balanceador de carga de aplicación. Sin embargo, primero se debe de crear un grupo de destino para definir los equipos a los cuales el balanceador de carga hará una redirección, además de definir parámetros para su monitoreo. La configuración realizada se muestra en la tabla 10.

Tabla 10. Atributos del Grupo de Destino

Atributo	Valor	Atributo	Valor
Tipo de destino	Instancias	Destinos	<ul style="list-style-type: none"> • rt-1 • rt-2
Nombre	GD-rt5-HTTP		
Protocolo	HTTP (Puerto 80)	Dirección de Monitoreo	http://{ip}/

Fuente: Elaboración propia

En la tabla 11, se muestra la configuración realizada para crear el balanceador de carga de aplicación, tras haber definido el grupo de destino.

Tabla 11. Parámetros del Balanceador de Carga

Grupo	Atributo	Valor
Configuraciones del Balanceador	Tipo de Balanceador	Aplicación
	Esquema	Expuesto a Internet
	Agentes de Escucha	GD-rt5-HTTP
Configuraciones de AWS	Nombre	ELB-rt5-HTTP
	Mapeo de Red	Donde se ubican las instancias EC2 <ul style="list-style-type: none"> us-east-1e us-east-1c
	Grupos de Seguridad	Por defecto

Fuente: Elaboración propia

Tras esto, ya se tiene el balanceador de carga, y se puede encontrar en el panel de AWS, junto con el DNS asignado para su uso.

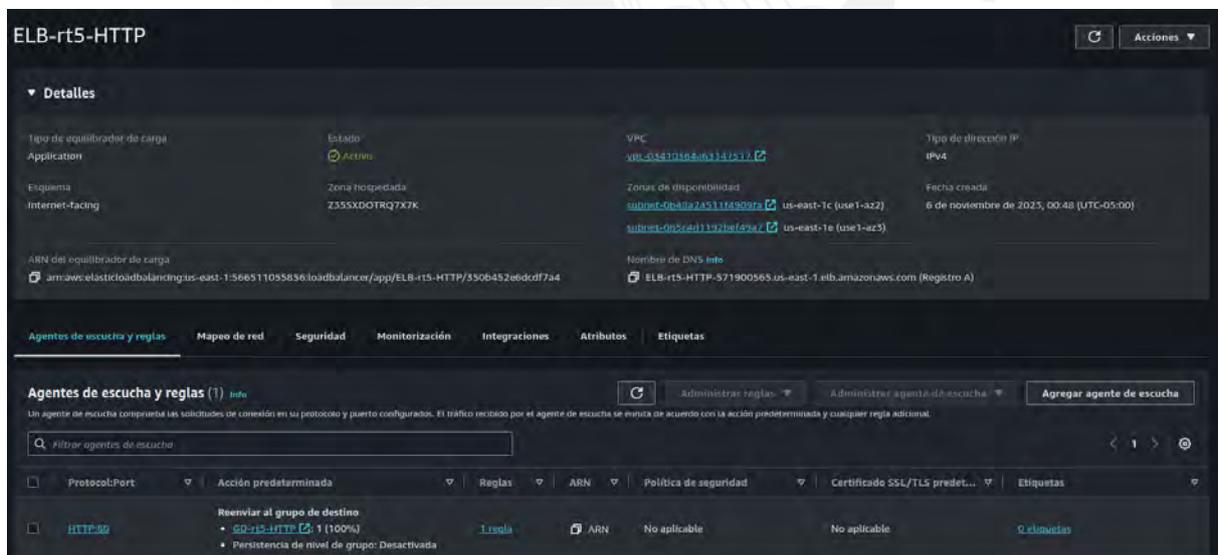


Figura 18. Balanceador de Carga Creado

Fuente: Elaboración propia

Finalmente, se realiza la prueba de conectividad usando el balanceador de carga, ingresando al servidor HTTP previamente configurado.

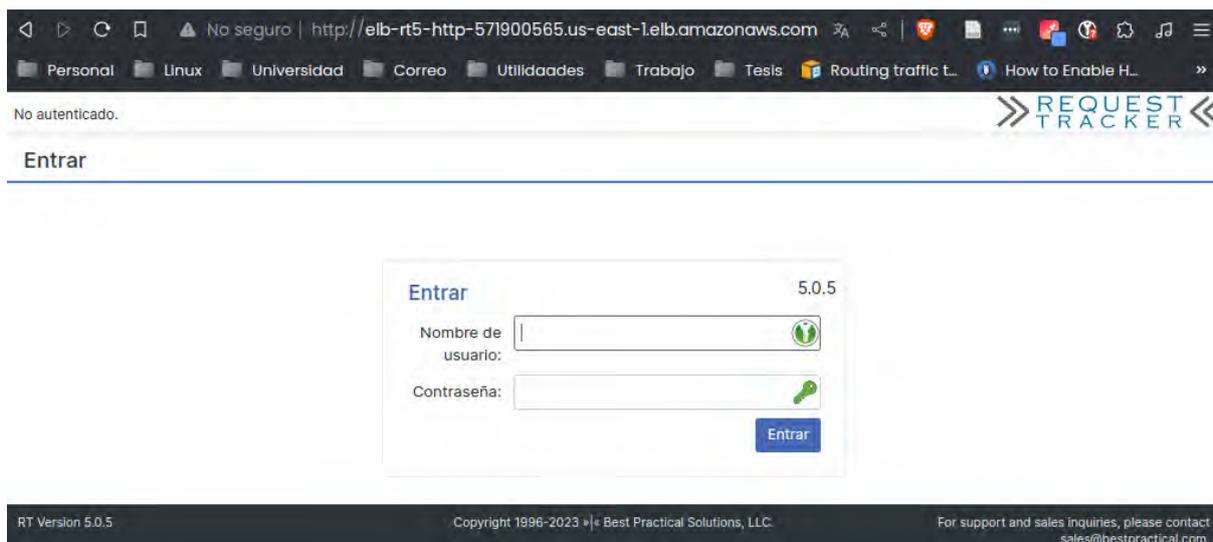


Figura 19. Prueba de conexión al balanceador de carga
Fuente: Elaboración propia

d. Amazon S3

Para almacenar archivos adjuntos y demás que puedan ser usados por el servicio, se opta por el uso de una instancia S3. Request Tracker provee soporte integrado para usar esta forma de almacenamiento, y detalla la configuración e información necesaria para hacerlo.

Lo primero que se realiza es crear el bucket en S3, para los que se usaron los parámetros detallados a continuación.

Tabla 12. Parámetros del bucket S3

Atributo	Valor
Nombre	rt5
Región	us-east-1
Acceso	Solo rt-1 y rt-2

Fuente: Elaboración propia

Después de crear este elemento, se procede a realizar la configuración del servidor de RT, donde se utilizó el siguiente script para instalar las dependencias necesarias y vincular el servicio con la base de datos.

```
#!/bin/bash
```

```

# Credenciales para el acceso a servicios de AWS
accessKey="<AWSaccessKey>"
secretAccessKey="<AWSsecretAccessKey>"

# instala el módulo de Perl necesario
sudo cpan install Amazon::S3

# Se registra el bucket S3
echo "
# Configuración del bucket S3
Plugin('Amazon::S3');

Set(%ExternalStorage,
  Type      => 'AmazonS3',
  AccessKeyId => '$accessKey',
  SecretAccessKey => '$secretAccessKey',
  Bucket    => 'rt5'
);

1;" | sudo tee /opt/rt5/etc/RT_SiteConfig.d/s3_config.pm

# Carga los archivos en la nube cada hora
echo "#!/bin/bash
/opt/rt5/sbin/rt-externalize-attachments -v
" | sudo tee /etc/cron.hourly/sync-s3.sh

# Recarga apache
sudo systemctl reload apache2
sudo systemctl restart apache2

```

Archivo 5. configure-s3.sh

e. Amazon Route 53

En Amazon Route 53 se realizará la configuración DNS que permitirá el redireccionamiento del tráfico entrante a Internet a los recursos desplegados en la nube de AWS. Además, es necesario usar un nuevo dominio para poder generar los certificados SSL empleados para configurar HTTPS, pues los dominios de AWS son identificados por los agentes de certificación como dominios inseguros.

Para iniciar este proceso, debido a que AWS Academy no permite el registro de dominios, se empleó el servicio de CloudDNS. De esta forma, tras crear la cuenta respectiva, se creó una zona alojada gratuita, en este caso de nombre 'tesis.cloudns.ph'. A continuación, se pueden observar los registros DNS generados tras registrar la zona alojada.



Figura 20. Registros DNS generados para la zona alojada de CloudDNS
Fuente: Elaboración propia

A partir de este punto, se siguió la documentación de AWS para la vinculación entre un servicio DNS externo y Amazon Route 53. [42] Lo primero que se realizó fue crear una zona alojada con el nombre del subdominio 'rt5' en Route 53, con los parámetros mostrados en la tabla 13.

Tabla 13. Parámetros de la zona alojada

Atributo	Valor
Nombre de dominio	rt5.tesis.cloudns.ph
Tipo de Zona Alojada	Pública

Fuente: Elaboración propia

Los registros generados por defecto al crear la zona alojada se pueden observar en la figura 21.

Cabe resaltar que la estructura de estos registros es similar a los de CloudDNS.

Nombre del registro	Tipo	Política...	Difer...	Alias	Valor/Dirigir el tráfico a	TTL (s...)	ID de c...
rt5.tesis.cloudns.ph	NS	Simple	-	No	ns-727.awsdns-26.net, ns-1390.awsdns-45.org, ns-1623.awsdns-10.co.uk, ns-61.awsdns-07.com.	172800	-
rt5.tesis.cloudns.ph	SOA	Simple	-	No	ns-727.awsdns-26.net. awsd...	900	-

Figura 21. Registros DNS generados por la zona alojada de Amazon Route 53
Fuente: Elaboración propia

A continuación, se procede a agregar los registros NS de Route 53 al subdominio registrado en CloudDNS, para que así Route 53 pueda administrar el subdominio e integrarlo con los otros servicios de AWS.

Host	Type	Points to	TTL
tesis.cloudns.ph	NS	ns41.cloudns.net	1h
tesis.cloudns.ph	NS	ns42.cloudns.net	1h
tesis.cloudns.ph	NS	ns43.cloudns.net	1h
tesis.cloudns.ph	NS	ns44.cloudns.net	1h
rt5.tesis.cloudns.ph	NS	ns-1390.awsdns-45.org	1h
rt5.tesis.cloudns.ph	NS	ns-1623.awsdns-10.co.uk	1h
rt5.tesis.cloudns.ph	NS	ns-61.awsdns-07.com	1h
rt5.tesis.cloudns.ph	NS	ns-727.awsdns-26.net	1h

Figura 22. Registros DNS modificados en CloudDNS
Fuente: Elaboración propia

Cabe destacar que, dependiendo del TTL de los NS, es posible que la migración de los *Name Servers* pueda tardar hasta 48 horas. Por esto mismo se le optó por cambiar en ambos servicios su valor a 1 hora. Tras haber cambiado los registros, se agrega un registro A en el panel de AWS, el cual apunte el subdominio raíz al balanceador de carga.

Tabla 14. Parámetros del registro DNS creado

Atributo	Valor
Nombre del registro	rt5.tesis.cloudns.ph
Tipo de registro	A
Alias	Sí
Tipo de Punto de Enlace	Alias de Classic Load Balancer o de aplicación
Región	us-east-1
Balanceador	dualstack.ELB-rt5-HTTP-571900565.us-east-

	1.elb.amazonaws.com
Política de Direccionamiento	Direccionamiento sencillo

Fuente: Elaboración propia

Finalmente, se prueba ingresando al nuevo dominio registrado usando HTTP, obteniéndose lo siguiente:

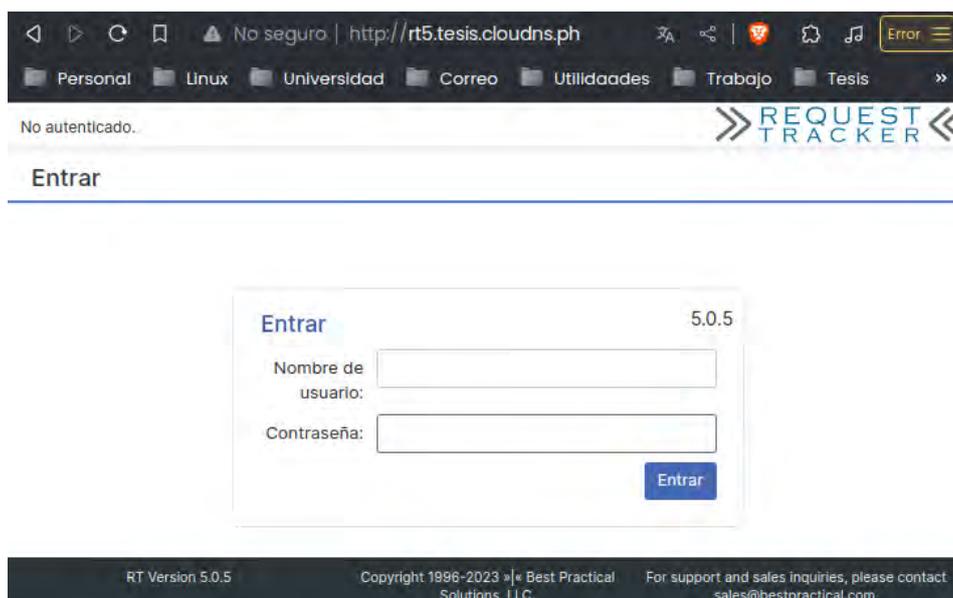


Figura 23. Acceso al servidor a través del dominio registrado

Fuente: Elaboración propia

f. Amazon Cloudfront

El uso de Cloudfront en esta solución será principalmente el de cacheado para reducir la carga entrante de los servidores. Para la configuración de Cloudfront, primero se recolectó la información de todos los otros servicios ya configurados en pasos anteriores. De esta forma, se crea una distribución.

Tabla 15. Configuración de Cloudfront

Atributo	Valor
Dominio de origen	rt5.tesis.cloudns.ph
Protocolo	Solo HTTP
Nombre	CF_Tesis

Fuente: Elaboración propia

3.4.3 Habilitando HTTPS

Aunque el servicio ya se encuentre funcionando, la comunicación entre el cliente y el servidor no es segura, pues la información no se encuentra encriptada. Para solucionar este problema, se procede con la configuración de la comunicación HTTPS, que usa un certificado TLS versión 1.2 para transmitir información encriptada de forma rápida y segura.

a. Amazon Certificate Manager (ACM)

Para poder soportar HTTPS, el sitio web debe de contar con los certificados TLS. Para vincularlos con los servicios de AWS se tienen distintas alternativas:

- **Uso de ACM:** Integrado con AWS, y ofrece monitoreo y renovación automática de los certificados.
- **Importación de certificados externos:** Importar certificados obtenidos de otras autoridades certificadoras.

Debido a su fácil administración y opciones de renovación simplificadas, se optó por el uso de un certificado de ACM, donde se realizó la configuración señalada en la tabla 16.

Tabla 16. Parámetros del certificado ACM

Atributo	Valor
Dominios	<ul style="list-style-type: none"> • tesis.cloudns.ph • rt5.tesis.cloudns.ph
Método de Validación	Por Correo
Algoritmo de la clave	RSA 2048

Fuente: Elaboración propia

De esta forma, ACM envía un correo de verificación a la dirección de correo registrada junto al dominio, para así corroborar que el dueño del dominio está emitiendo el certificado. Para poder recibir el correo, en primer lugar, se debe de crear un alias, que vincule un correo

existente con un correo perteneciente al dominio recientemente creado. Cabe resaltar que, por defecto, AWS enviará un correo a las direcciones `admin@`, `administrator@`, `hostmaster@`, `postmaster@` y `webmaster@`, por lo que se debe definir el alias de correo del dominio con uno de estos nombres de usuario.



Figura 24. Vinculación del dominio con una dirección de correo
Fuente: Elaboración propia

Asimismo, se deben de agregar los registros MX en el servidor DNS para permitir esa vinculación. Estos se encargan de redirigir el tráfico de una dirección de correo del dominio a una dirección de correo conocida, como es el correo PUCP.

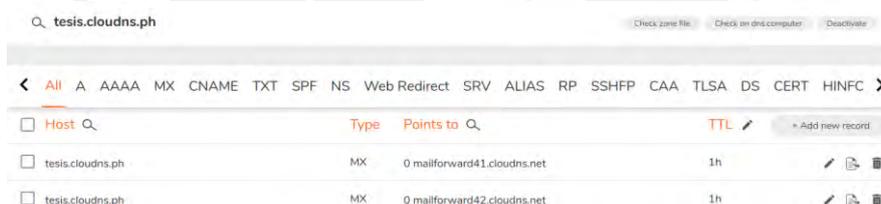


Figura 25. Adición de los registros MX
Fuente: Elaboración propia

Completando estos pasos, se logra recibir el correo de verificación en la cuenta PUCP, donde se accede a un enlace y se da por terminada la validación.



Figura 26. Correo de verificación para emitir el certificado
Fuente: Elaboración propia

Tras esto, se observa que el certificado ya fue emitido y validado para los dominios ingresados.

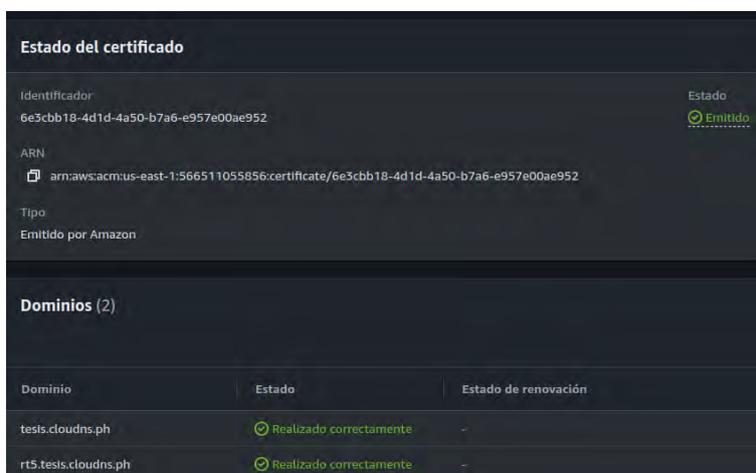


Figura 27. Certificado SSL emitido
Fuente: Elaboración propia

b. Balanceador de Cargas

Para permitir que el balanceador de carga reciba tráfico HTTPS, es necesario agregarle un nuevo agente de escucha, y asignarle el certificado que fue configurado en la sección anterior.

Tabla 17. Adición de agente de escucha HTTPS al balanceador

Atributo	Valor
Agente de Escucha	HTTPS (443)
Acciones de Enrutamiento	Reenviar a Grupo de destino
Grupo de Destino	GD-rt5-HTTP
Origen del Certificado	ACM
Certificado	tesis.cloudns.ph

Fuente: Elaboración propia

Con esta acción se ha configurado la conexión segura desde el cliente hasta el balanceador de cargas. Sin embargo, la conexión entre el balanceador de carga y las instancias EC2 sigue empleando HTTP. Se tomó esta decisión pues, en la documentación de AWS [50], se declara que no es necesario implementar HTTPS entre ambos medios, pues la encriptación es una medida de seguridad ante la interceptación de paquetes en algún punto de la red pública. Sin

embargo, al ser la nube de Amazon privada, y al acuerdo de privacidad de AWS, se puede considerar que el tráfico de información dentro de la red privada es seguro. De esta manera, ya no es necesario realizar configuraciones a otros servicios de la plataforma.

Se debe considerar también que, para evitar problemas de dominio con el servidor de Request Tracker, se realiza un ligero cambio en la configuración del archivo `‘/opt/rt5/etc/RT_SiteConfig.d/domain_config.pm’`, cambiando el dominio del servidor por `‘rt5.tesis.cloudns.ph’`.

De esta manera, tras haber seguido estos pasos, se realiza una prueba de conectividad intentando ingresar al dominio usando HTTPS, observándose resultados favorables

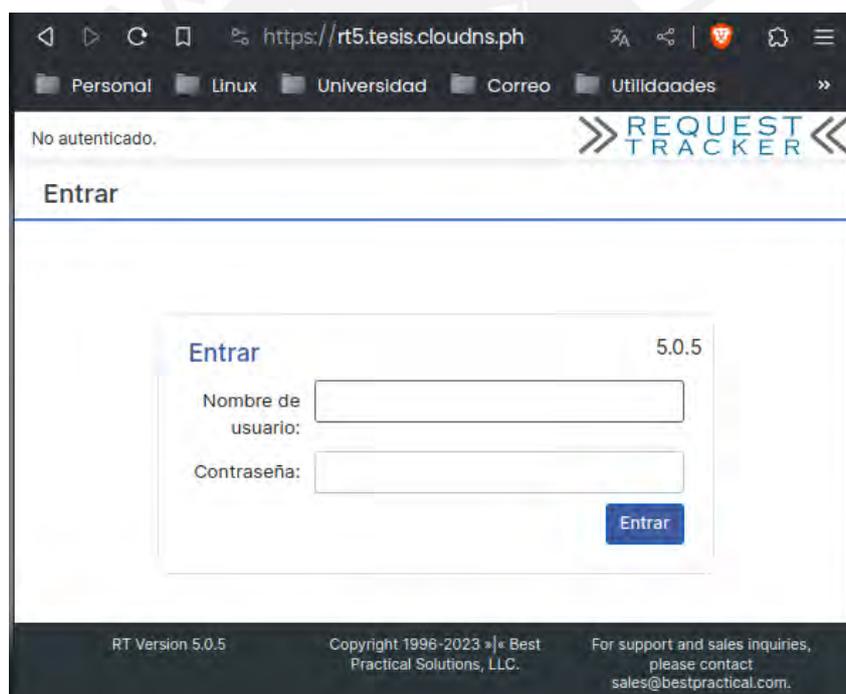


Figura 28. Prueba de conectividad exitosa usando HTTPS
Fuente: Elaboración propia

3.5 Análisis de la Disponibilidad

Como se observó en la figura 15, la arquitectura de la solución se encuentra compuesta por distintos servicios de la nube de AWS. Para hallar la disponibilidad del servicio, se realizará un análisis de cada elemento de esta arquitectura.

- **Amazon Route 53:** Este servicio, por su misma naturaleza de difundir las zonas hospedadas DNS en todo el mundo, posee una arquitectura redundante y descentralizada. Según la documentación del servicio [45], este tiene comunicación con 83 servidores DNS alrededor del mundo, teniendo así también una alta resiliencia y disponibilidad. Asimismo, según su SLA, este declara que posee una disponibilidad del 100%.
- **Amazon Cloudfront:** Revisando la información de disponibilidad declara en el SLA de Amazon Cloudfront [49], se determina que su disponibilidad es del 99.9%.
- **Amazon ELB:** Al ser 2 instancias en distintas zonas de disponibilidad que emplean un balanceador de carga para operar de forma Activa-Activa, su SLA [46] declara que se tiene una disponibilidad del 99.99%.
- **Amazon RDS:** Para la base de datos, se tiene una arquitectura de múltiple zona de disponibilidad, aumentando la resiliencia ante cualquier falla de la base de datos. Esta redundancia es del tipo Activa-Pasiva, por lo que su SLA [47] indica que tiene una disponibilidad del 99.95%.
- **Amazon S3:** Para el servicio de almacenamiento, su SLA [48] declara que posee una disponibilidad del 99.9%.

Para poder hallar la disponibilidad total de un sistema, se realiza la multiplicación de la disponibilidad individual de cada elemento. De esta forma, tras obtener la SLA de los distintos componentes, se procede a hallar la disponibilidad total del sistema, realizando la siguiente operación.

$$\text{Disponibilidad} = 100\% * 99.9\% * 99.99\% * 99.95\% * 99.9\% = 99.74\%$$

Así se tiene entonces que la disponibilidad total del sistema es del 99.74%.

3.6 Diseño e Implementación del Aplicativo Móvil

Tras haber levantado la arquitectura en la nube, se prosiguió con el diseño e implementación de un aplicativo móvil sencillo, basado en Android, con el cual se puede visualizar información sobre los tickets activos en la plataforma.

3.6.1 Diagrama de Flujo

Se planeó un flujo de uso para la aplicación

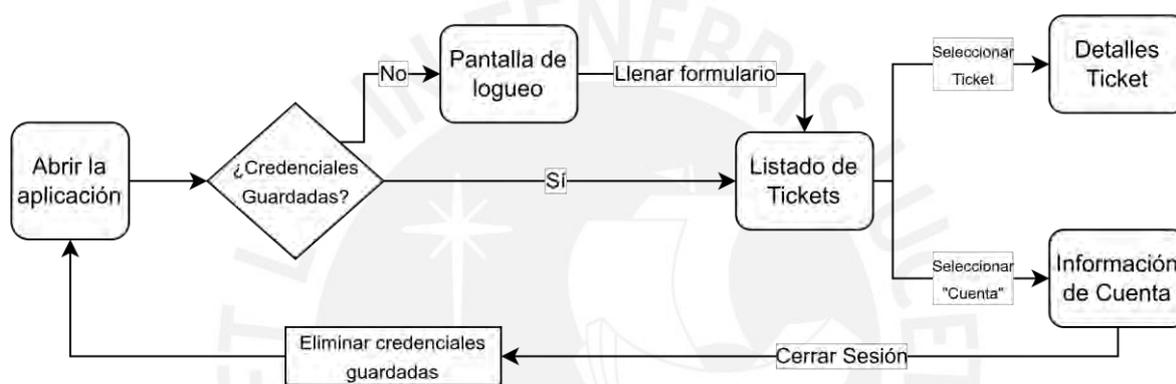


Figura 29. Diagrama de flujo de la aplicación
Fuente: Elaboración propia

3.6.2 Mockups

A partir del diagrama de flujo planteado para la aplicación, se procedió con el diseño de los mockups de la aplicación. Para su elaboración, se usó la herramienta Figma, previamente mencionada en la sección 3.2.5.

- **Interfaz de Inicio de Sesión:** En primer lugar, se definió la vista de la pantalla de inicio de sesión. En esta un usuario previamente creado, ingresa el dominio donde se encuentran alojados los servidores de Request Tracker. Posteriormente, el usuario ingresa sus credenciales previamente registradas en el sistema para iniciar sesión. Además, se muestran mensajes de error al haber problemas de conexión, y al ingresar credenciales inválidas.

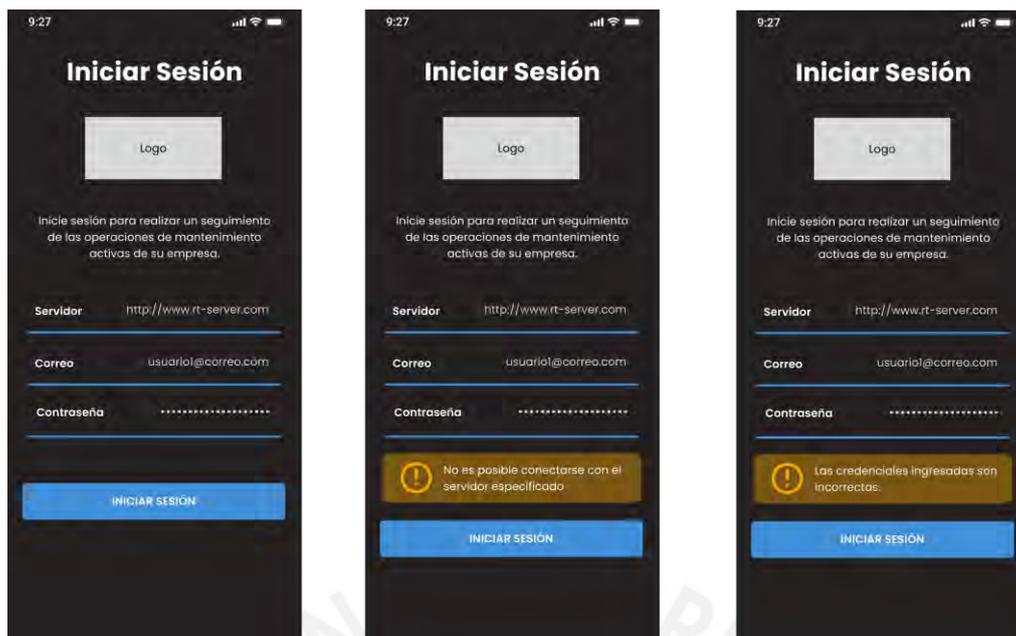


Figura 30. Mockups de inicio de sesión y de mensajes de error

Fuente: Elaboración propia

- **Vista de Listado de Tickets:** Tras haberse logueado de forma exitosa, el usuario se verá ante una pestaña de inicio, donde se observará una barra de búsqueda, un listado de tickets, y una barra de navegación. Empezando por el listado, este mostrará información de todos los tickets creados o asignados a este usuario, y que aún no estén cerrados. Para cada ticket se mostrará su asunto, el tiempo transcurrido tras su creación, y su prioridad, además de un botón para visualizar más detalles. Por otro lado, la barra de búsqueda permitirá el filtrado de tickets según su asunto. Finalmente, la barra de navegación permitirá cambiar a otra vista, la cual será explicada más adelante.

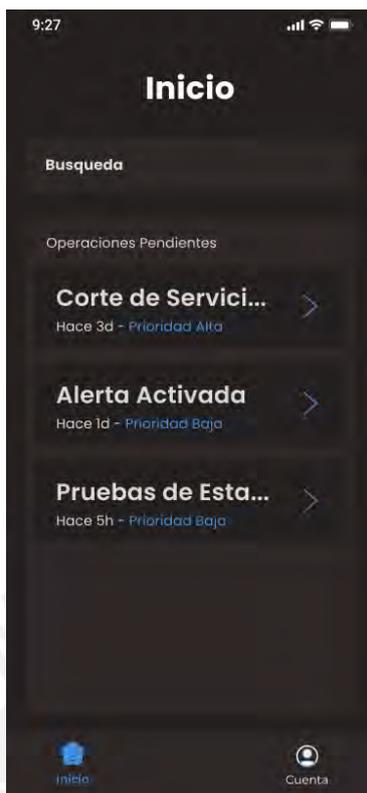


Figura 31. Mockup de vista de listado de tickets
Fuente: Elaboración propia

- **Vista de Detalles de Ticket:** Al seleccionar un ticket del listado anterior, se accederá a una nueva vista, donde se visualizará una mayor información del ticket. Entre estas se verá su asunto, su fecha de creación y su fecha límite de cierre, e información del creador y todos los participantes en la atención del ticket.



Figura 32. Mockup de vista de detalles del ticket
Fuente: Elaboración propia

- **Vista de Cuenta:** Al seleccionar la otra opción en la barra de navegación inferior, se podrá visualizar información del usuario con el que se inició sesión. Se podrá visualizar información de identidad, de contacto, de ubicación geográfica, y de la organización a la que pertenece y el rol que desempeña en esta. Asimismo, en esta vista se mostrará un botón que permitirá cerrar sesión, el cual regresará al usuario a la vista de inicio de sesión. De esta manera este podrá conectarse a otro servidor o ingresar al mismo, pero desde otra cuenta.

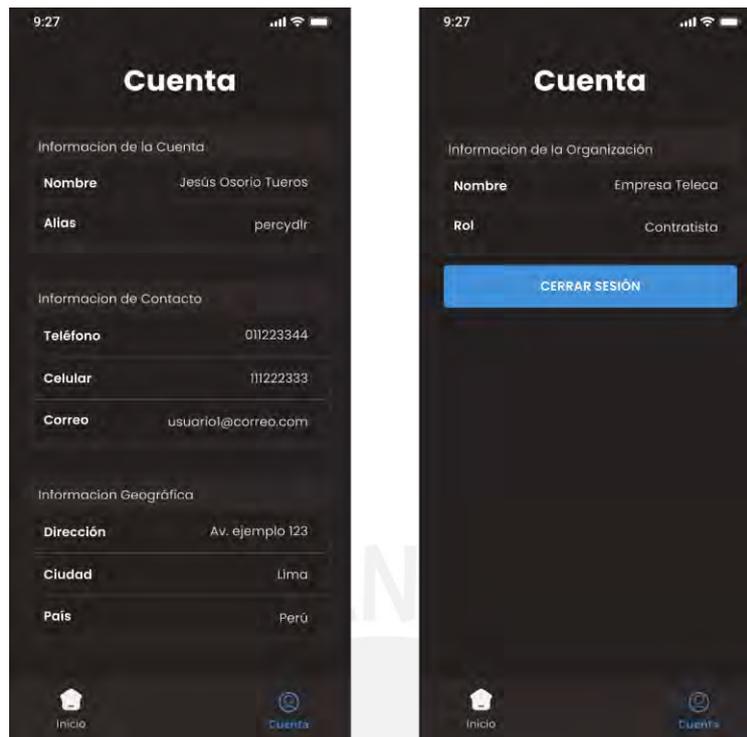


Figura 33. Mockup de vista de información de la cuenta
Fuente: Elaboración propia

3.6.3 Resultados Finales

A partir de los mockups mostrados, se empezó con el desarrollo del aplicativo móvil. Como se explicó anteriormente en la sección 3.2, se va a usar Android Studio como IDE, y Java como el lenguaje de programación sobre el que se hará la lógica de la aplicación. Para interactuar con el servicio de Request Tracker que se encuentra operando en los servidores de AWS, se usa la API proporcionada por Request Tracker, y explicada en su documentación. El código fuente de la aplicación desarrollada se puede encontrar en:

<https://github.com/PercyDLR/RT-Companion>

A continuación, se mostrarán las vistas de la aplicación desarrollada, explicando la forma en la que interactúan con el servidor ya existente. Cabe mencionar que se desarrolló un modo claro y un modo oscuro, los cuales serán mostrados de forma alternada durante la explicación.

- **Vista de Inicio de Sesión:** Esta vista pide especificar el dominio donde se aloja el servidor, y las credenciales de acceso para una cuenta creada previamente. Para

verificar la conexión exitosa, y la validez de los datos, se realiza una petición para obtener toda la información del usuario cuyas credenciales se ingresaron, la cual se muestra a continuación.

Tabla 18. Petición para obtener información de usuario

Tipo de Petición	GET
URI	/REST/2.0/user/{usuario}
Cabeceras	Authorization: {usuario}:{pwd}
Resultado Esperado	<pre> Status: 200 { "Address1": "Av. Ejemplo 123", "LastUpdated": "2023-12-06T05:42:06Z", "Address2": "", "Creator": { "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/root", "type": "user", "id": "root" }, "Privileged": 1, "EmailAddress": "usuario1@correo.com", "PagerPhone": "Percy De La Rosa Vera", "Disabled": "0", "Zip": "15331", "Timezone": "America/Lima", "_hyperlinks": [{ "id": "40", "type": "user", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/40", "ref": "self" }, { "ref": "history", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/40/history" }], "Created": "2023-12-03T18:52:10Z", "Gecos": "root", "Memberships": [{ "type": "group", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/group/42", "id": "42" }], "State": "", "CustomFields": [], "RealName": "Percy De La Rosa Vera", "Lang": "", "Name": "percydlr", "FreeformContactInfo": "", "MobilePhone": "111222333", "City": "Lima", "LastUpdatedBy": { "id": "root", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/root", "type": "user" }, "HomePhone": "5262851", "NickName": "Percy", "Country": "Perú", "Comments": "", "Organization": "RT Prueba", </pre>

	<pre>"id": 40, "WorkPhone": "" }</pre>
Resultados Fallidos	<ul style="list-style-type: none"> • Connection Time Out: La dirección ingresada no es válida • Status: 401: Credenciales inválidas

Fuente: Elaboración propia

Al tener un registro exitoso, se almacena la información de logueo en un archivo encriptado dentro de la memoria interna del teléfono, de tal manera que los siguientes inicios de sesión sean automáticos. A continuación, se muestra la vista de inicio de sesión junto con todos los mensajes de error que puede recibir.

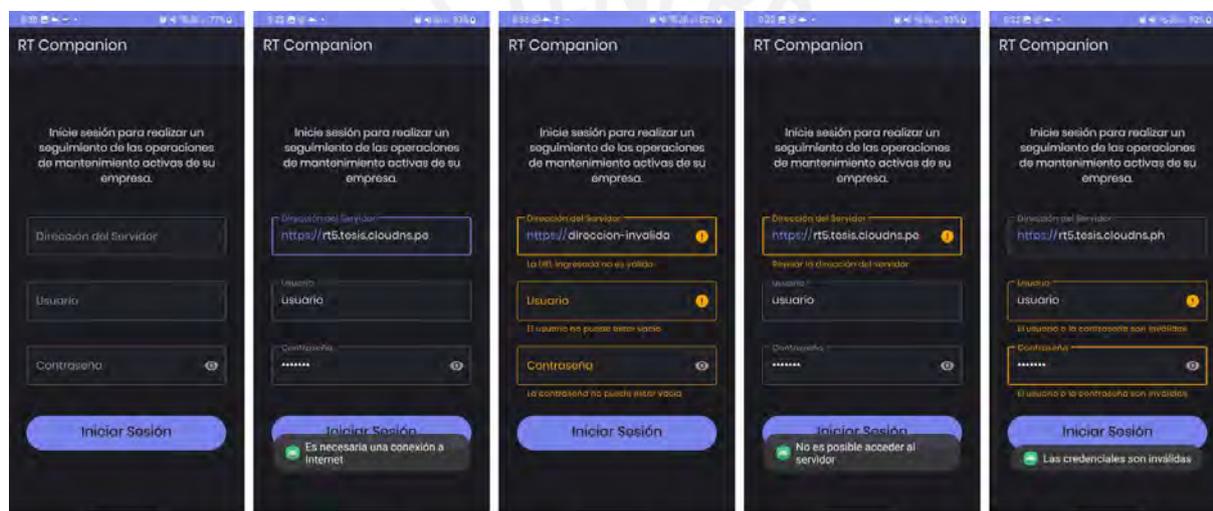


Figura 34. Vista de inicio de sesión

Fuente: Elaboración propia

- **Vista del listado de tickets:** Tras un logueo exitoso, se accede a esta vista, donde se observarán todos los tickets con estados ‘Nuevo’, ‘Abierto’ y ‘Pendiente’ que fueron creados por o asignados al usuario. Para obtener esta información se usó la siguiente petición HHTP a la API de Request Tracker. En cada ítem se puede visualizar la misma información del ticket señalada en los mockups.

Tipo de Petición	GET
Argumento	Se necesita especificar una búsqueda, donde “” muestra todo.
URI	/REST/2.0/tickets?query=(Subject like '{busqueda}')&fields=Owner,Status,Created,Subject,Queue,CustomFie

	Ids,Requestor,Cc,AdminCc,CustomRoles,Due,Priority
Cabeceras	Authorization: {usuario}:{pwd}
Resultado Esperado	<pre> Status: 200 { "items": [{ "Priority": "100", "Subject": "Prueba para la tesis", "Created": "2023-11-07T05:19:39Z", "Status": "open", "CustomRoles": {}, "Requestor": [{ "type": "user", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/percydlr", "id": "percydlr" }], "id": "1", "Queue": { "id": "1", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/queue/1", "type": "queue" }, "Due": "2023-12-25T05:00:00Z", "Cc": [{ "id": "percydlr", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/percydlr", "type": "user" }], "CustomFields": "", "type": "ticket", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/ticket/1", "Owner": { "id": "root", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/root", "type": "user" }, "AdminCc": [{ "id": "percydlr", "type": "user", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/percydlr" }] }], } </pre>
Resultados Fallidos	<ul style="list-style-type: none"> • Connection Time Out: Se perdió la conexión • Status: 401: El usuario fue modificado

Fuente: Elaboración propia

Empleando esta API, es posible listar todos los tickets en los que está asignado un trabajador, ya sea del NOC o de Mantenimiento. Además, es posible realizar una búsqueda simple para encontrar un ticket en específico.

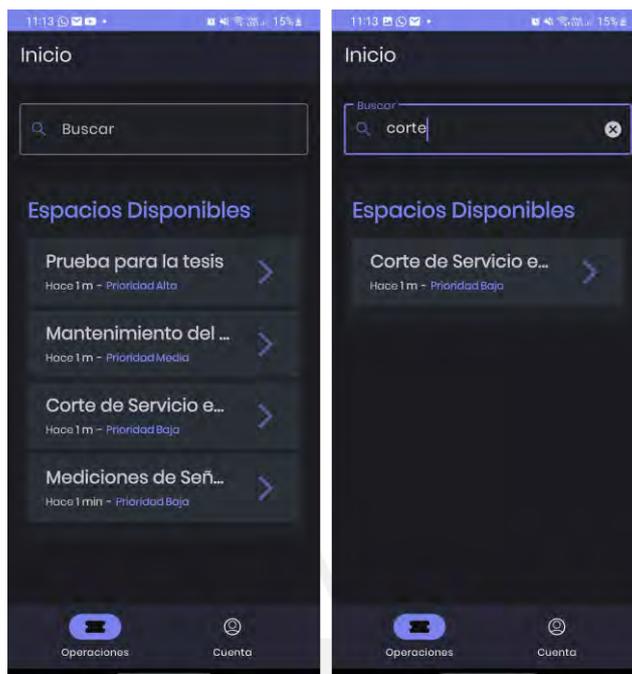


Figura 35. Vista de listado de tickets, junto a la búsqueda
Fuente: Elaboración propia

- **Vista de detalle de Ticket:** Al seleccionar un ticket, se observa una mayor cantidad de información. Además, debido a que ya se obtuvo la información deseada con la petición anterior que da información de todos los tickets, no se necesita realizar una nueva consulta a la API para entrar a esta vista. De esta manera, la pantalla se observa a continuación.

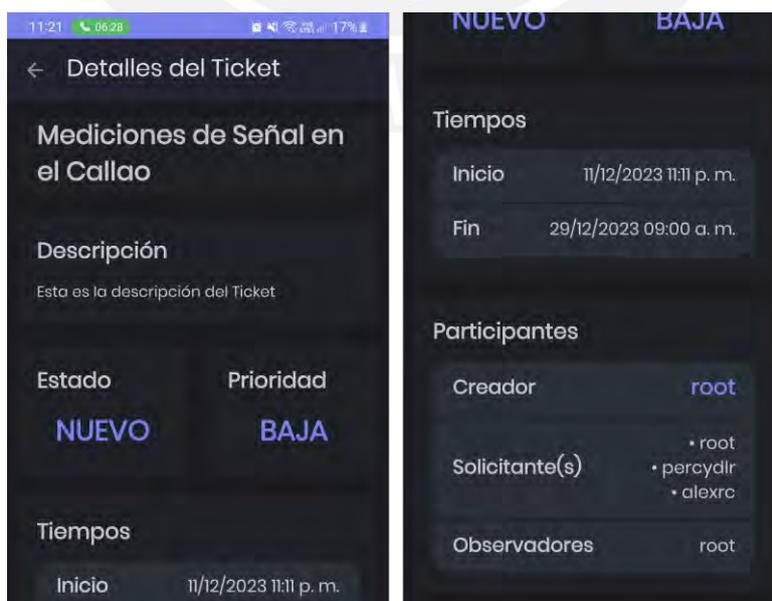


Figura 36. Vista de detalles del ticket
Fuente: Elaboración propia

- Vista de Información de la Cuenta:** Al seleccionar el otro elemento de la barra de navegación inferior, al igual que en los mockups, se observa una vista con la información del usuario con el que se inició sesión. Se muestran su nombre, su usuario en el sistema, e información geográfica, de contacto y de su organización y su rol en ella. Sin embargo, a pesar que en el proceso de inicio de sesión se obtuviera la mayor parte de la información del usuario, es necesario realizar una solicitud adicional para obtener la información de la organización y del rol del usuario. Esto sucede porque la primera petición solo brinda el id del rol, mas no su nombre. De esta forma, los detalles de la solicitud a la API se muestran a continuación.

Tipo de Petición	GET
URI	/REST/2.0/group/{idGrupo}
Cabeceras	Authorization: {usuario}:{pwd}
Resultado Esperado	<p>Status: 200</p> <pre> { "Instance": "0", "id": 42, "_hyperlinks": [{ "id": "42", "type": "group", "ref": "self", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/group/42" }, { "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/group/42/history", "ref": "history" }, { "ref": "members", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/group/42/members" }], "Creator": { "id": "root", "type": "user", "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/root" }, "Disabled": "0", "LastUpdated": "2023-12-03T18:54:43Z", "CustomFields": [], "Name": "Contratistas", "Domain": "UserDefined", "Members": [{ "_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/percydlr", "type": "user", "id": "percydlr" }], "Created": "2023-12-03T18:54:23Z", "LastUpdatedBy": { </pre>

	<pre>"_url": "http://rt5.tesis.cloudns.ph/REST/2.0/user/root", "type": "user", "id": "root" } }</pre>
Resultados Fallidos	<ul style="list-style-type: none"> • Connection Time Out: Se perdió la conexión • Status: 401: El usuario fue modificado

Fuente: Elaboración propia

Asimismo, la vista final se ve de la siguiente forma.



Figura 37. Vista de información del usuario

Fuente: Elaboración propia

3.7 Desafíos encontrados

Durante la implementación de este trabajo de tesis se encontraron ciertas trabas que retrasaron el despliegue del servicio, e incluso hicieron reiniciar el desarrollo varias veces. Estas serán explicadas a continuación:

- En primer lugar, hubo un problema al levantar el servidor de Request Tracker. Tras haber realizado la instalación y configuración básica, como estaba señalada hasta la sección 3.3.2, se experimentó un error extraño. De acuerdo con la documentación de

Request Tracker, tras realizar un logueo con las credenciales predeterminadas (usuario: *root* y contraseña: *password*) debía ser posible acceder al dashboard de la plataforma. Sin embargo, al ingresar las credenciales, se volvía a la vista de *login* sin ningún mensaje de error. Sin embargo, al ingresar unas credenciales incorrectas sí aparecía un mensaje de error de autenticación. Por esto, al creer que era un error en la instalación, probé reinstalando y reconfigurando el servicio 3 veces, cambiando ligeramente el procedimiento en cada oportunidad. Finalmente al tercer intento, y tras investigar en los foros de ayuda de Request Tracker [51], encontré la solución. Esta era incluir el argumento “Set(\$WebSecureCookies, 0);” en el archivo de configuración `RT_SiteConfig.pm`. Lo que este hace es deshabilitar una medida de seguridad de Request Tracker, la cual previene la creación de *cookies* en conexiones inseguras (HTTP). Esta configuración se dejó activa durante la fase de pruebas y configuración, es decir, hasta antes de llegar a la sección 3.4, donde se habilitó HTTPS.

- Tras terminar la configuración del servidor, y comenzar a preparar las consultas para la API de Request Tracker, se observó que, ante cualquier solicitud, a pesar de seguir las pautas de la documentación de Request Tracker, el servidor devolvía un error HTTP 401, indicando un error de autenticación. Tras una investigación en los foros de Request Tracker [52], se identificó que el problema era que el servidor Apache no leía el encabezado *Authorization*, por lo que este no procesaba las credenciales enviadas junto con la petición. Para solucionarlo, se agregó la línea “SetEnvIf Authorization "(.*)” HTTP_AUTHORIZATION=\$1” en la configuración del sitio en el servidor Apache.
- Habiendo configurado el servidor S3 según indicaba la documentación de Request Tracker, se observó que los archivos adjuntos no se cargaban en la nube, y solo se quedaban con su copia local en el sistema. De esta forma, volviendo a consultar los

foros de RT [53], se identificó que para transferir los archivos a un bucket S3 es necesario ejecutar el comando `‘/opt/rt5/sbin/rt-externalize-attachments’`. De esa forma, se procedió con automatizar la ejecución de este comando, para que así se ejecute cada hora.



Capítulo 4. Pruebas y análisis económico de la solución desarrollada

4.1 Pruebas de Rendimiento

Como se mencionó en la sección 3.2.8, se usará Apache JMeter para realizar las pruebas de rendimiento a la solución desarrollada en este trabajo.

4.1.1 Configuración del Plan de Pruebas

Dentro de Jmeter, un Plan de Pruebas contiene al conjunto de elementos que definen a una o más pruebas. Asimismo, en este se definen algunos parámetros globales, como variables de entorno y librerías adicionales.

a. Grupo de Hilos

Este elemento es el encargado de realizar las conexiones usadas para las pruebas. Es posible modificar el comportamiento de estas conexiones, mediante los siguientes parámetros:

- **Número de Hilos:** Indica la cantidad total de conexiones que se realizarán.

- **Ramp-up Period:** Define la cantidad de conexiones que se realizarán por segundo. Si se indican valor de 0, Jmeter realizará una conexión simultánea con todos los hilos.
- **Iteraciones:** Se define la cantidad de conexiones que realizará cada hilo durante la prueba.

De esta forma, considerando un total de hasta 100 usuarios activos, con un promedio de 6 conexiones simultáneas, se muestran los valores configurados para esta prueba en la tabla 19.

Tabla 19. Configuración del grupo de hilos en Jmeter

Parámetro	Valor
Nombre	Pruebas de Conectividad
Número de Hilos	100
Ramp-up Period	6
Iteraciones	1

Fuente: Elaboración propia

b. HTTP(S) Test Script Recorder

Este componente permite la creación y registro sencillo de las solicitudes que se realizarán para hacer las pruebas. Este funciona creando un servidor proxy, el cual, al ser configurado dentro de un navegador, recibe y registra todas las peticiones HTTPS que realiza el usuario en un periodo de tiempo, y las vincula con un grupo de hilos.

Tabla 20. Configuración del HTTP Test Script Recorder

Parámetro	Valor
Puerto	8888
Destino	Plan de Prueba > Pruebas Conectividad
Patrones URL a Incluir	rt5.tesis.cloudns.ph

Fuente: Elaboración propia

Se ingresa la dirección del servidor proxy “http://localhost:8888” en la configuración del navegador, y se navega por la interfaz para capturar peticiones para las pruebas.

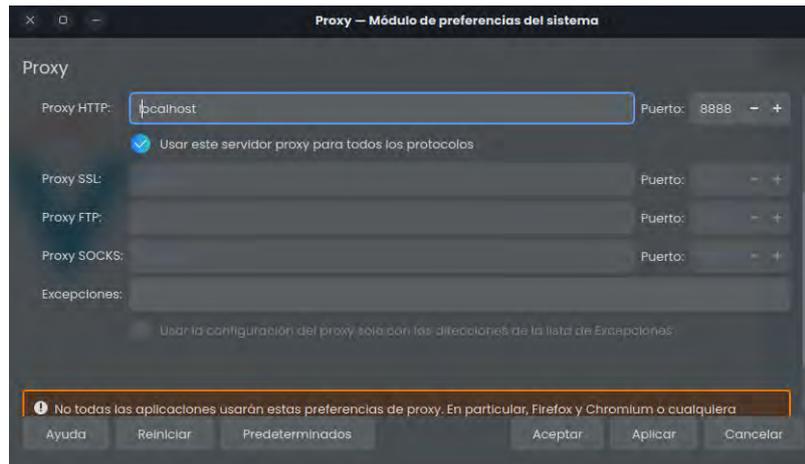


Figura 38. Configuración del servidor proxy
Fuente: Elaboración propia

Al detener la captura, se tienen 10 peticiones, entre las que se tienen el cargado de listados, la visualización de información de cuenta, y la descarga de recursos, como un libro de Excel y una imagen. Estas 10 peticiones serán entonces ejecutadas por cada uno de los 100 hilos.

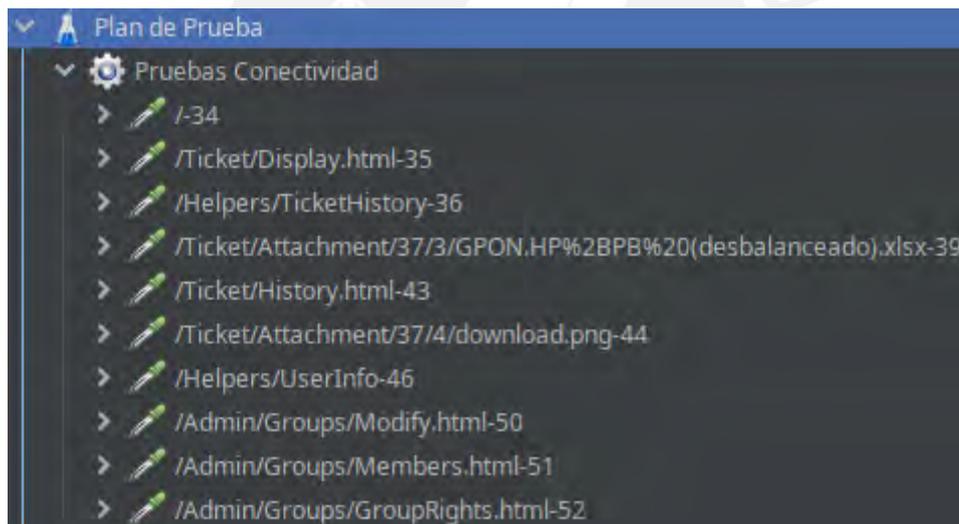


Figura 39. Peticiones para las pruebas en Jmeter
Fuente: Elaboración propia

c. Agentes de Escucha

Teniendo ya configuradas las pruebas que se van a realizar, se procede con la reacción de agentes de escucha que permitan visualizar información sobre los resultados de cada petición, y puedan procesarlos para determinar valores estadísticos útiles para un par un análisis posterior. Los agentes de escucha configurados fueron los siguientes:

- **View Results Tree:** Permite visualizar la información de cada solicitud, como su código de respuesta, el tamaño de los distintos campos en la solicitud y respuesta HTTPS, la duración de la conexión, etc. Este sirvió como una herramienta de monitoreo para identificar problemas en la configuración de la prueba, así como problemas en la red.
- **Summary Report:** Genera estadísticas respecto a las solicitudes generadas en la prueba. Muestra información como el retardo mínimo y máximo, junto con su desviación estándar, el número de solicitudes atendidas por segundo, el porcentaje de peticiones fallidas, y el tamaño promedio de la respuesta.

De esta forma, se tiene que la configuración final del plan de pruebas es como se muestra en la figura 40 a continuación.

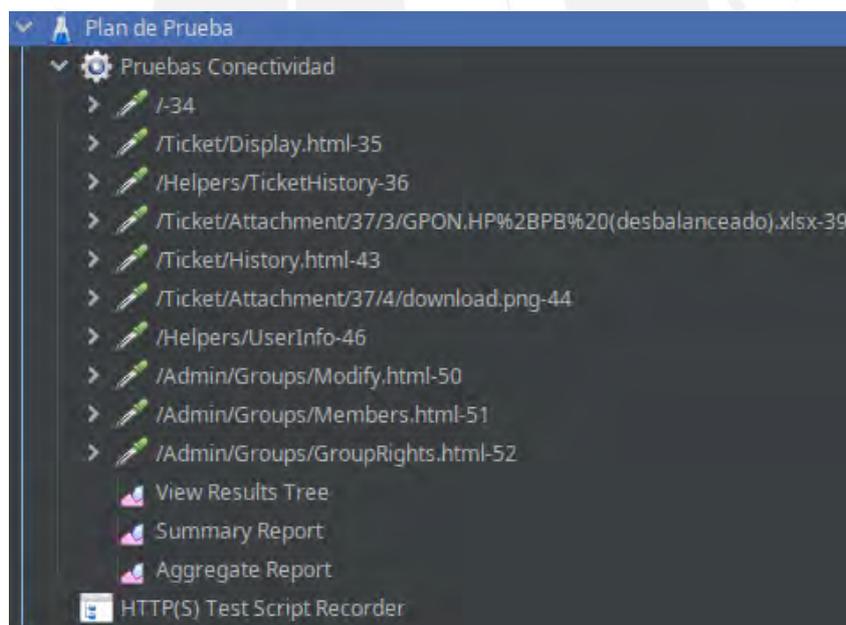


Figura 40. Configuración final del plan de prueba en Jmeter
Fuente: Elaboración propia

4.1.3 Análisis de Resultados

Al acabar la ejecución del plan de pruebas, y verificar que las peticiones han llegado correctamente al destino usando el *View Results Tree*, se procede a analizar la información procesada en el *Summary Report*. Esos resultados se pueden apreciar en la figura 41.

Label	# Samples	Average	Min ↓	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
/-34	100	4387	0	37030	6690.46	1.00%	2.3/sec	5.71	0.97	2489.7
/Ticket/Display.html-35	100	6540	0	60117	10090.69	5.00%	1.6/sec	4.44	1.38	2778.3
/Helpers/TicketHistory-36	100	9210	0	65431	15642.30	10.00%	1.3/sec	3.39	1.01	2657.5
/Ticket/Attachment/37/3/GPON_HP%2BPB%20(desbalancead...	100	7130	0	71582	15909.19	9.00%	1.3/sec	3.42	1.21	2690.1
/Ticket/History.html-43	100	5080	0	60093	11609.72	3.00%	1.3/sec	3.59	1.17	2831.4
/Ticket/Attachment/37/4/download.png-44	100	3160	0	60238	9538.55	3.00%	1.2/sec	3.20	0.86	2839.7
/Helpers/UserInfo-46	100	2093	0	29384	5321.18	0.00%	1.2/sec	3.27	0.91	2904.3
/Admin/Groups/Modify.html-50	100	1663	0	29235	4920.05	0.00%	1.2/sec	3.41	1.13	2904.3
/Admin/Groups/Members.html-51	100	1479	0	60090	7513.45	2.00%	1.2/sec	3.36	1.12	2861.1
/Admin/Groups/GroupRights.html-52	100	1738	0	60228	8322.58	2.00%	1.2/sec	3.30	1.10	2861.0
TOTAL	1000	4248	0	71582	10553.41	3.50%	10.0/sec	27.22	8.20	2781.7

Figura 41. Resultados de la prueba de rendimiento con Jmeter
Fuente: Elaboración propia

En estos resultados se pueden destacar los siguientes aspectos:

- **Latencia:** La latencia promedio de las respuestas ante esta demanda fue de 4.248 s, con una desviación estándar de hasta 10.5 s. Se observa que gran parte de esta desviación se dio por el tiempo de descarga de los archivos XLSX y PNG. Este resultado es esperado debido al tamaño de estos archivos, los cuales tienen una descarga más lenta.
- **Porcentaje de Error:** De las 1000 peticiones que se realizaron, 35 obtuvieron un mensaje de error del servidor, lo que significa un 3.5%. Observando detenidamente las pruebas, se tiene que la mayoría de errores se dieron al descargar varias veces el archivo XLSX. Al tener una descarga más lenta, se pudo haber estado descargando el mismo archivo muchas veces al mismo tiempo, generando más tráfico en la red.
- **Throughput:** El sistema fue capaz de responder en promedio 10 solicitudes por segundo.

4.2 Análisis Económico

Para realizar el análisis de costos de la aplicación, es necesario clasificar los costos en la inversión inicial (CAPEX), y los costos de operación (OPEX).

4.2.1 CAPEX

Dentro del CAPEX se consideran todas las inversiones iniciales. Al ser un proyecto que emplea servicios de la nube, y que usa servicios y aplicaciones gratuitos o de código abierto, no hay muchos gastos en esta etapa. De esta forma, se considera principalmente el pago para la mano de obra que participó en la planificación, diseño y despliegue de la solución final. En los cálculos se considera que:

- La mano de obra la realiza un practicante que cobra el sueldo mínimo, es decir, S/. 1050 al mes. Considerando que se trabaja de lunes a viernes por 4 semanas y 2 días, se tienen 22 días de trabajo. Si, además, el practicante trabaja durante 6 horas diarias, se tiene que su pago es de:

$$1050 \div 22 \div 6 \approx 7.95 \text{ soles/hora}$$

- Para diciembre del 2023, el tipo de cambio es de aproximadamente S/. 3.75.
- No hay costos por licencias ya que todo el software empleado es *Open Source* o de elaboración propia.

Tabla 21. Cálculo de la inversión inicial de la solución

Categoría	Tiempo Total	Soles/hora	Total Soles	Total Dólares
Diseño Arquitectura	15 horas	S/. 7.95	S/. 119.25	\$ 31.80
Despliegue AWS	65 horas	S/. 7.95	S/. 516.75	\$ 137.80
Diseño App Móvil	4 horas	S/. 7.95	S/. 31.80	\$ 8.48
Implementación App	45 horas	S/. 7.95	S/. 357.75	\$ 95.40
Total				\$ 273.48

Fuente: Elaboración propia

4.2.2 OPEX

Dentro de los costos de operación, se consideran los pagos recurrentes que se realizarán para garantizar el funcionamiento de la solución. Al ser el caso de una solución basada en la nube, se tiene que estos gastos corresponden a los costos de los servicios empleados. Se considera que estos servicios son usados por 100 empleados durante 40 horas a la semana (8 horas diarias de lunes a viernes). Los costos mostrados a continuación fueron estimados a partir de la calculadora de precios de AWS .

- **EC2:** Considerando 2 instancias t2.medium de 20 GB de almacenamiento alojadas en diferentes zonas de disponibilidad dentro de una misma región, y que además son usadas de lunes a viernes durante 8 horas diarias.
- **RDS:** Considerando una instancia db.t3.small de 20 GB de almacenamiento, con un sistema de redundancia Activo-Pasivo en otra zona de disponibilidad. Es usada de lunes a viernes durante 8 horas diarias, o 40 horas semanales.
- **Route 53:** Considerando solo 1 zona alojada, con 1 solo récord DNS adicional.
- **S3:** Considerando que cada usuario cargue 2 archivos diariamente en el sistema, con un tamaño promedio de 1 MB entre imágenes y documentos, se realizan 200 operaciones de carga al día y 4000 al mes (equivalente a 4 GB al mes) Asimismo, se considera que cada trabajador abre al menos 5 archivos al día, lo que dan 500 lecturas diarias y alrededor de 10000 lecturas al mes (equivalente a 10 GB). Para este caso se asume que ya hay 20 GB de información almacenados.
- **Load Balancer:** Considerando un balanceador de carga de aplicación, que procesa hasta 4 GB de información de entrada y salida por hora. Además, se considera una duración de conexión corta (de menos de 10 segundos) al solo ser usada para servicios web y para APIs.

- Al analizar el consumo de otros servicios del sistema, como Amazon Certificate Manager, Amazon Cloudfront, y CloudDNS, se observa que su consumo cae dentro del nivel gratuito, por lo que no son considerados como gastos de operación.

Tabla 22. Cálculo de los costos de operación mensuales y anuales

Servicio	Cantidad	Costo Unitario	Costo Mensual
Amazon EC2	2	\$ 12.32	\$ 24.64
Amazon RDS	2	\$ 8.21	\$ 16.42
Amazon Route 53	1	\$ 0.50	\$ 0.50
Amazon S3	1	\$ 0.47	\$ 0.47
Amazon ELB	1	\$ 39.79	\$ 39.79
Total			\$ 81.82

Fuente: Elaboración propia

4.3 Recapitulación

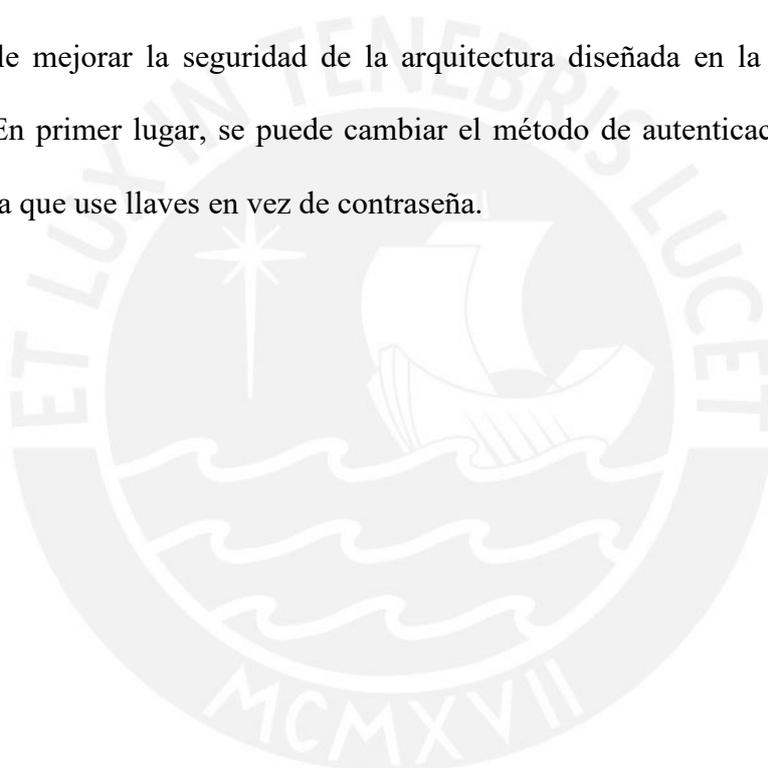
Con lo visto en este capítulo, se puede observar que la solución planteada es capaz de operar en un ambiente con 100 usuarios activos, considerando que existen 6 conexiones simultáneas en promedio. De esta forma, puede adaptarse a los requerimientos y a la demanda de uso de una empresa proveedora de servicios de telecomunicaciones. Se observa también que, al usar una arquitectura en la nube, y usar software de código abierto, existe un ahorro considerable en la inversión inicial (consiste únicamente de la mano de obra) y los costos de mantenimiento y operación (solo se paga por los servicios desplegados en la nube).

CONCLUSIONES

- Se diseñó e implementó una plataforma de gestión de operaciones de mantenimiento orientada hacia las compañías proveedoras de servicios de telecomunicaciones. Esta solución no satisface los requerimientos iniciales de facilitar el seguimiento de operaciones de mantenimiento, administrar usuarios y grupos, gestionar el inventario, y centralizar la información de las operaciones para análisis posteriores.
- El sistema desplegado logró una alta disponibilidad, al emplear distintas técnicas como la redundancia en múltiples zonas de disponibilidad para levantar un respaldo cuando el equipo primario falle, el uso de balanceadores de carga que distribuyan el tráfico de forma equitativa entre múltiples equipos, y el uso de un servidor CDN que agilice la carga de contenido repetitivo.
- Se estableció una conexión HTTPS entre el usuario y el balanceador de carga, demostrando así que la solución planteada puede ser empleada por proveedores de servicios de telecomunicaciones con pocos riesgos a la seguridad.
- La aplicación móvil desarrollada como complemento a la plataforma brindada permite visualizar los tickets asignados a un usuario en específico, así como visualizar un mayor detalle de cada ticket, y de revisar la información con la que está registrado este usuario en el sistema.

RECOMENDACIONES Y OBSERVACIONES

1. Se puede hacer más robusta la arquitectura en la nube, agregando servicios para *logging* con Amazon CloudWatch, y extendiendo la API de Request Tracker con Amazon Lambda.
2. Es posible aumentar la funcionalidad de la aplicación móvil, agregando más filtros para el listado de tickets, permitiendo la creación de estos, y ofreciendo la opción de editar la información de la cuenta.
3. Es posible mejorar la seguridad de la arquitectura diseñada en la nube de diversas formas. En primer lugar, se puede cambiar el método de autenticación de la base de datos para que use llaves en vez de contraseña.



BIBLIOGRAFÍA

[1]P. De La Rosa Vera, «Estudio comparativo de plataformas web para el seguimiento de operaciones de mantenimiento en una empresa proveedora de servicios de telecomunicaciones», Trabajo de Investigación, Pontificia Universidad Católica del Perú, Lima, Perú.

[2]E.-M. Kern, H.-G. Hegering, y B. Brügge, *Managing Development and Application of Digital Technologies: Research Insights in the Munich Center for Digital Technology & Management (CDTM)*. Springer Science & Business Media, 2006.

[3]A. Hochstein y W. Brenner, «Implementation of Service-Oriented it Management: an Empirical Study on Swiss it Organizations», en 2006 International Conference on Service Systems and Service Management, oct. 2006, pp. 91-97. doi: 10.1109/ICSSSM.2006.320594.

[4]«The World Factbook: Peru», The World Factbook. Central Intelligence Agency, 11 de julio de 2023. Accedido: 14 de julio de 2023. [En línea]. Disponible en: <https://www.cia.gov/the-world-factbook/countries/peru/>

[5]«Panorama de Mercado - Perú», TeleSemana.com. Accedido: 14 de julio de 2023. [En línea]. Disponible en: <https://www.telesemana.com/panorama-de-mercado/peru/>

[6]«Cantidad de Centros Poblados con Cobertura Móvil - [OSIPTel] | Plataforma Nacional de Datos Abiertos». Plataforma Nacional de Datos Abiertos, 19 de noviembre de 2019. Accedido: 14 de julio de 2023. [En línea]. Disponible en: <https://www.datosabiertos.gob.pe/dataset/cantidad-de-centros-poblados-con-cobertura-m%C3%B3vil-osiptel>

[7]«COBERTURA DE SERVICIO MÓVIL POR EMPRESA OPERADORA | Plataforma Nacional de Datos Abiertos». Plataforma Nacional de Datos Abiertos, 17 de agosto de 2023. Accedido: 14 de julio de 2023. [En línea]. Disponible en: <https://www.datosabiertos.gob.pe/dataset/cobertura-de-servicio-m%C3%B3vil-por-empresa-operadora>

[8]H. Vasquez, S. Lucho Romero, y P. De La Rosa, «Conversaciones con el Ingeniero representante de Gilat», 24 de mayo de 2023.

[9]IBM, «ITIL - IT Infrastructure Library | IBM», IBM. Accedido: 15 de junio de 2023. [En línea]. Disponible en: <https://www.ibm.com/topics/it-infrastructure-library>

[10]Bluemara Solutions, «ITIL V4: Introducción a la actualización ITIL 2019», Bluemara Solutions. Accedido: 15 de junio de 2023. [En línea]. Disponible en:

<https://bluemarasolutions.com/itil-v4-actualizacion-2019/>

[11]M. Brenner, «Classifying ITIL Processes; A Taxonomy under Tool Support Aspects», en 2006 IEEE/IFIP Business Driven IT Management, Vancouver, Canada: IEEE, 2006, pp. 19-28. doi: 10.1109/BDIM.2006.1649207.

[12]M. Raza, «SLA vs KPI: What's the Difference?», BMC Blogs. Accedido: 16 de junio de 2023. [En línea]. Disponible en: <https://www.bmc.com/blogs/sla-vs-kpi/>

[13]K. Magowan, «Help Desk vs Service Desk: What's The Difference?», BMC Blogs. Accedido: 15 de junio de 2023. [En línea]. Disponible en: <https://www.bmc.com/blogs/help-desk-vs-service-desk-whats-difference/>

[14]«Service desk vs help desk vs ITSM: What's the difference?», Atlassian. Accedido: 15 de junio de 2023. [En línea]. Disponible en: <https://www.atlassian.com/itsm/service-request-management/help-desk-vs-service-desk-vs-itsm>

[15]A. Smithers, «Help desk vs service desk: What's the difference?», Zendesk. Accedido: 15 de junio de 2023. [En línea]. Disponible en: <https://www.zendesk.com/blog/help-desk-vs-service-desk-whats-name/>

[16]«What is Cloud Computing», Amazon Web Services, Inc. Accedido: 29 de octubre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/what-is-cloud-computing/>

[17]Raj Bala, Dennis Smith, Kevin Ji, David Wright, y Miguel Angel Borrega, «Magic Quadrant for Cloud Infrastructure and Platform Services». Gartner, 18 de octubre de 2022. Accedido: 29 de octubre de 2023. [En línea]. Disponible en: <https://www.gartner.com/doc/reprints?id=1-2AOZQAQL&ct=220728&st=sb>

[18]«Types of Cloud Computing - SaaS vs PaaS vs IaaS - AWS», Amazon Web Services, Inc. Accedido: 29 de octubre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/types-of-cloud-computing/>

[19]«Diferencias entre IaaS, PaaS y SaaS», Red Hat. Accedido: 1 de noviembre de 2023. [En línea]. Disponible en: <https://www.redhat.com/es/topics/cloud-computing/iaas-vs-paas-vs-saas>

[20]«What is an Application Programming Interface (API)? | IBM», IBM. Accedido: 15 de junio de 2023. [En línea]. Disponible en: <https://www.ibm.com/topics/api>

[21]«Public Cloud vs Private Cloud vs Hybrid Cloud», GeeksforGeeks. Accedido: 1 de noviembre de 2023. [En línea]. Disponible en: <https://www.geeksforgeeks.org/public-cloud-vs-private-cloud-vs-hybrid-cloud/>

[22]P. Corona y V. Del Angel, Arquitecturas de Alta Disponibilidad | Capítulo 1, (6 de mayo de 2020). Accedido: 7 de septiembre de 2023. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=Dsip7sn0sTU>

[23]«What is high availability?», Red Hat. Accedido: 7 de septiembre de 2023. [En línea]. Disponible en: <https://www.redhat.com/en/topics/linux/what-is-high-availability>

[24]A. Novotný, «AWS High Availability Architecture: How to Create it? | StormIT», StormIT. Accedido: 7 de septiembre de 2023. [En línea]. Disponible en: <https://www.stormit.cloud/blog/aws-high-availability-architecture/>

[25]«What Is High Availability?», NGINX Resources. Accedido: 7 de septiembre de 2023. [En línea]. Disponible en: <https://www.nginx.com/resources/glossary/high-availability/>

[26]S. Croucher, «High availability and disaster recovery for IaaS apps - Azure Architecture Center», High availability and disaster recovery scenarios for IaaS apps. Accedido: 7 de septiembre de 2023. [En línea]. Disponible en: <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/infrastructure/iaas-high-availability-disaster-recovery>

[27]«High Availability Concepts and Best Practices», Oracle Docs. Accedido: 7 de septiembre de 2023. [En línea]. Disponible en: https://docs.oracle.com/cd/A91202_01/901_doc/rac.901/a89867/pshavdtl.htm

[28]D. D. Garcia-Swartz y M. Campbell-Kelly, Cellular: An Economic and Business History of the International Mobile-Phone Industry. The MIT Press, 2022. doi: 10.7551/mitpress/11542.001.0001.

[29]«Android Open Source Project», Android. Accedido: 19 de octubre de 2023. [En línea]. Disponible en: <https://source.android.com/?hl=es-419>

[30]«What is Apple iOS? | Definition from TechTarget», Mobile Computing. Accedido: 2 de noviembre de 2023. [En línea]. Disponible en: <https://www.techtarget.com/searchmobilecomputing/definition/iOS>

[31]«Mobile OS market share worldwide 2009-2023», Statista. Accedido: 19 de octubre de 2023. [En línea]. Disponible en: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>

[32]«¿Qué es el desarrollo de aplicaciones móviles? | Microsoft Azure», Microsoft Azure. Accedido: 19 de octubre de 2023. [En línea]. Disponible en: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-mobile-app-development/>

[33]«What is an API?», Red Hat. Accedido: 15 de junio de 2023. [En línea]. Disponible en: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>

[34]«What is Amazon Route 53? - Amazon Route 53», Amazon Web Services, Inc. Accedido: 2 de noviembre de 2023. [En línea]. Disponible en: <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>

[35]«What is Amazon CloudFront? - Amazon CloudFront», Amazon Web Services, Inc. Accedido: 5 de noviembre de 2023. [En línea]. Disponible en: <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>

[36]«Load Balancer - Amazon Elastic Load Balancer (ELB) - AWS», Amazon Web Services, Inc. Accedido: 5 de noviembre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/elasticloadbalancing/>

[37]«Secure and resizable cloud compute – Amazon EC2 – Amazon Web Services», Amazon Web Services, Inc. Accedido: 5 de noviembre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/ec2/>

[38]«Fully Managed Relational Database - Amazon RDS - Amazon Web Services», Amazon Web Services, Inc. Accedido: 5 de noviembre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/rds/>

[39]«Apache JMeter - Apache JMeter™», Apache Software Foundation. Accedido: 5 de noviembre de 2023. [En línea]. Disponible en: <https://jmeter.apache.org/>

[40]«Cloud Object Storage – Amazon S3 – Amazon Web Services», Amazon Web Services, Inc. Accedido: 5 de noviembre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/s3/>

[41]«Request Tracker: The open-source ticket management system», Best Practical Solutions. Accedido: 5 de junio de 2023. [En línea]. Disponible en: <https://bestpractical.com>

[42]«Making Route 53 the DNS service for an inactive domain - Amazon Route 53», Amazon Web Services, Inc. Accedido: 1 de noviembre de 2023. [En línea]. Disponible en: <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/migrate-dns-domain-inactive.html#migrate-dns-create-records-domain-inactive>

[43]«Escalado automático de instancias - Amazon EC2 Auto Scaling - AWS», Amazon Web Services, Inc. Accedido: 7 de diciembre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/es/ec2/autoscaling/>

[44]«Service usage and other restrictions», Amazon Web Services, Inc.

[45]«Amazon Route 53 Service Level Agreement», Amazon Web Services, Inc. Accedido: 10 de noviembre de 2023. [En línea]. Disponible en:

<https://d1.awsstatic.com/legal/AmazonRoute53SLA/Amazon%20Route%2053%20Service%20Level%20Agreement-ES.pdf>

[46]«Amazon Elastic Load Balancing Service Level Agreement», Amazon Web Services, Inc. Accedido: 10 de noviembre de 2023. [En línea]. Disponible en: https://d1.awsstatic.com/legal/AmazonElasticLoadBalancing/Amazon_Elastic_Load_Balancing_Service_Level_Agreement_2022-07-25_ES-ES.pdf

[47]«Amazon RDS Service Level Agreement», Amazon Web Services, Inc. Accedido: 10 de noviembre de 2023. [En línea]. Disponible en: https://d1.awsstatic.com/legal/amazonrdsservice/Amazon_RDS_Service_Level_Agreement__2022-03-09_Spanish.pdf

[48]«Amazon S3 Service Level Agreement», Amazon Web Services, Inc. Accedido: 10 de noviembre de 2023. [En línea]. Disponible en: https://d1.awsstatic.com/legal/amazons3service/Amazon_S3_Service_Level_Agreement_Spanish.pdf

[49]«Amazon CloudFront Service Level Agreement», Amazon Web Services, Inc. Accedido: 12 de noviembre de 2023. [En línea]. Disponible en: https://d1.awsstatic.com/legal/amazon-cloudfront-sla/Amazon_CloudFront_Service_Level_Agreement-ES.pdf

[50]«Create a Classic Load Balancer with an HTTPS listener - Elastic Load Balancing», Amazon Web Services, Inc. Accedido: 16 de noviembre de 2023. [En línea]. Disponible en: <https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-create-https-ssl-load-balancer.html>

[51]«RT 5.0.4 upgrade - Can't log in now - Login page just refresh - RT Users», Request Tracker Community Forum. Accedido: 12 de diciembre de 2023. [En línea]. Disponible en: <https://forum.bestpractical.com/t/rt-5-0-4-upgrade-cant-log-in-now-login-page-just-refresh/38114/7>

[52]«REST2 Api issue on RT5.0.1», Request Tracker Community Forum. Accedido: 12 de septiembre de 2023. [En línea]. Disponible en: <https://forum.bestpractical.com/t/rest2-api-issue-on-rt5-0-1/36017>

[53]«How to store Files not in database», Request Tracker Community Forum. Accedido: 12 de diciembre de 2023. [En línea]. Disponible en: <https://forum.bestpractical.com/t/how-to-store-files-not-in-database/37931>

[54]«Zendesk: The Complete Customer Service Solution», Zendesk. Accedido: 24 de marzo de 2024. [En línea]. Disponible en: <https://www.zendesk.com/>

[55]Atlassian, «Jira Service Management | A new take on ITSM software», Atlassian. Accedido: 21 de abril de 2024. [En línea]. Disponible en: <https://www.atlassian.com/software/jira/service-management>

[56]Feshworks, «Freshservice | ITSM», Freshworks. Accedido: 10 de abril de 2024. [En línea]. Disponible en: <https://www.freshworks.com/freshservice/>

[57]Solarwinds, «IT Service Desk Software | SolarWinds», Solarwinds. Accedido: 18 de marzo de 2024. [En línea]. Disponible en: <https://www.solarwinds.com/service-desk>

