

SISTEMA WEB INTEGRAL PARA UN CALL CENTER USANDO SERVIDOR IP ISSABEL

COMPREHENSIVE WEB SYSTEM FOR A CALL CENTER USING ISSABEL IP SERVER

Ayrtón Dayan Carrión Vega ^{1*}

¹ Tecnologías de la información, Facultad de Ingeniería Civil, Universidad Técnica de Machala. Ecuador. ORCID: <https://orcid.org/0009-0001-2968-0906>. Correo: acarrion5@utmachala.edu.ec

Bryan Alexis Cuero Ayovi ²

² Tecnologías de la información, Facultad de Ingeniería Civil, Universidad Técnica de Machala. Ecuador. ORCID: <https://orcid.org/0009-0009-4097-5172>. Correo: bcuero2@utmachala.edu.ec

Milton Rafael Valarezo Pardo ³

³ Tecnologías de la información, Facultad de Ingeniería Civil, Universidad Técnica de Machala. Ecuador. ORCID: <https://orcid.org/0000-0003-1819-2074>. Correo: mvalarezo@utmachala.edu.ec

Oscar Efrén Cárdenas Villavicencio ⁴

⁴ Tecnologías de la información, Facultad de Ingeniería Civil, Universidad Técnica de Machala. Ecuador. ORCID: <https://orcid.org/0000-0001-6570-8040>. Correo: ocardenas@utmachala.edu.ec

* Autor para correspondencia: bcuero2@utmachala.edu.ec

Resumen

Este estudio presentó el diseño e implementación de un sistema web integral para centros de llamadas cuyo objetivo fue resolver la falta de integración entre Issabel PBX alojado en un entorno local y una plataforma de gestión en la nube, se utilizó Tailscale una VPN basada en WireGuard® como canal seguro de conexión. La arquitectura desarrollada se basó en microservicios Dockerizados, frontend en Angular y base de datos PostgreSQL desplegados sobre Google Cloud. La metodología adoptada fue ágil empleando el marco SCRUM para gestionar el desarrollo incremental de funcionalidades clave. Durante un mes de evaluación el sistema logró una reducción del 82 % en el tiempo promedio de registro de llamadas de 4.5 a 0.8 minutos, una tasa de sincronización de datos del 99.8 % y una disponibilidad del 98.7 % en las grabaciones, se

eliminaron completamente los errores de duplicación presentes en el proceso manual previo, estos resultados evidencian la eficiencia operativa alcanzada así como la confiabilidad de la solución implementada.

Palabras clave: Centros de llamadas; Issabel PBX; microservicios; contenedores Docker; telefonía IP

Abstract

This study presented the design and implementation of a comprehensive web system for call centers, aimed at addressing the lack of integration between Issabel PBX, hosted in a local environment, and a cloud-based management platform. Tailscale, a VPN based on WireGuard®, was used as a secure communication channel. The developed architecture was based on Dockerized microservices, a frontend built with Angular, and a PostgreSQL database deployed on Google Cloud. An agile methodology was adopted, using the SCRUM framework to manage the incremental development of key functionalities. During a one-month evaluation period, the system achieved an 82% reduction in the average call registration time, from 4.5 to 0.8 minutes, a data synchronization rate of 99.8%, and 98.7% availability of recordings. Duplication errors present in the previous manual process were completely eliminated. These results demonstrate the operational efficiency achieved and the reliability of the implemented solution.)

Keywords: Call centers; Issabel PBX; microservices; Docker containers; IP telephony

Fecha de recibido: 14/01/2025

Fecha de aceptado: 17/04/2025

Fecha de publicado: 22/04/2025

Introducción

Los *call centers* son plataformas esenciales para facilitar la comunicación entre organizaciones y sus clientes, ofreciendo soporte respecto a productos o servicios. Estas unidades suelen incorporar tecnologías como los sistemas IVR (Respuesta de Voz Interactiva), que direccionan las llamadas hacia agentes especializados o servicios automatizados (Lei et al., 2022; Wang et al., 2023). Desde principios del siglo XXI, los *call centers* se consolidaron como canal principal de contacto en diversas industrias generando millones de empleos, su relevancia ha crecido con la adopción de modelos de teletrabajo los cuales han impulsado la eficiencia operativa y conectividad global (Pinzon et al., 2023).

A nivel mundial los *call centers* enfrentan la creciente necesidad de adoptar sistemas tecnológicos avanzados que optimicen sus operaciones, reduzcan costos y mejoren la satisfacción del cliente, sin embargo la falta de integración entre sistemas de telefonía IP (como Issabel PBX) y plataformas de gestión en la nube sigue representando un obstáculo crítico, esta problemática genera ineficiencias operativas que afectan la capacidad de los administradores para monitorear y gestionar interacciones en tiempo real impactando de forma negativa la calidad del servicio como la percepción de la marca.

A pesar de la relevancia de este desafío se observa una escasez de investigaciones a nivel global que aborden de manera integral este problema la mayoría de los estudios se centran en soluciones de software o en la mejora de la calidad de las interacciones sin considerar la integración total de estos sistemas tecnológicos, esto resalta la necesidad de explorar más a fondo las implicaciones y soluciones posibles para lograr una integración efectiva y optimizar la gestión de *call centers* a nivel mundial.

En América Latina y específicamente en Colombia los *call centers* han convertido en una de las principales fuentes de empleo para los jóvenes debido a la falta de oportunidades laborales en sectores de alta cualificación, este fenómeno está vinculado al crecimiento del sector de *Business Process Outsourcing* (BPO) que se posiciona como un pilar económico, representando entre el 2,5% y el 10% del PIB del país. La dependencia de modelos laborales basados en habilidades medias y la falta de integración tecnológica avanzada limita la estabilidad laboral y el crecimiento profesional, destacando la urgencia de soluciones escalables y modernas (Perdomo Herrera & Tapias Ortiz, 2023).

Para abordar la problemática de la integración de sistemas en los *call centers* es fundamental revisar trabajos previos que hayan tratado aspectos relacionados con la optimización de operaciones, la integración de tecnologías de telefonía IP y la mejora de la gestión de interacciones. (Monserate Aguilar & Suárez Dioses, 2020) desarrollaron un software para registrar y controlar monitoreos en un *call center* con Visual Studio y C# como tecnologías base y SQL Server como motor de datos, el sistema reemplazó las hojas de cálculo tradicionales mejorando la precisión del registro y la recuperación de información mediante una arquitectura en tres capas.

Asimismo, (Adrianzen Solis & Sanchez Penadillo, 2021) propusieron una plataforma VoIP con tecnología SIP para una entidad bancaria, reemplazando una central telefónica obsoleta en donde utilizaron *Genesys Pure Engage* sobre una infraestructura virtualizada incorporando componentes como *SIP Server* y *Genesys Voice Platform*, lo que permitió alcanzar alta disponibilidad (99.99%) y escalabilidad geográfica.

Por otro lado, (Pinilla Romero, 2021) implementó mejoras digitales en TICBRIDGE S.A.S., incluyendo la migración a una central VoIP basada en 3CX en Azure, el uso de dispositivos SD-WAN Fortinet y la modernización del sitio web corporativo estas acciones permitieron fortalecer la seguridad de red, mejorar la experiencia del cliente y optimizar procesos internos.

Los estudios revisados han abordado problemáticas específicas del sector: monitoreo de llamadas, reemplazo de centrales telefónicas obsoletas e infraestructura en la nube, estos trabajos no logran integrar de forma completa tecnologías como microservicios, VPN seguras y plataformas en la nube con soluciones de telefonía IP.

La propuesta actual plantea una solución integral basada en una arquitectura de microservicios Dockerizados, una base de datos PostgreSQL y una interfaz desarrollada en Angular y se integra con Issabel PBX mediante la VPN Tailscale (WireGuard®) lo que garantiza conectividad segura con el entorno local a diferencia de los trabajos previos con SQL Server o Asterisk, este enfoque reduce costos, incrementa escalabilidad y mejora la trazabilidad operativa.

La modularidad de los microservicios permite actualizaciones y escalamiento independientes minimizando el tiempo de inactividad y el uso de Issabel PBX aporta funcionalidades integradas como grabación, monitoreo

y estadísticas en una sola plataforma, este sistema centraliza la administración de llamadas, optimiza los tiempos de registro y mejora la gestión de interacciones en *call centers*.

Se plantea diseñar e implementar un sistema web integral que permita la administración centralizada de llamadas entrantes y salientes, así como de las grabaciones mediante la integración de Issabel PBX con la plataforma de gestión *call center* para optimizar la eficiencia operativa y mejorar la calidad de la atención al cliente en tiempo real.

Para alcanzar este propósito se plantean investigar la viabilidad técnica de integrar sistemas de telefonía IP; diseñar una interfaz intuitiva que facilite la gestión de llamadas y grabaciones; desarrollar un sistema funcionalmente compatible con Issabel PBX; finalmente validar su rendimiento a través de pruebas de calidad y eficiencia.

En función de esta propuesta se sostiene que la integración de Issabel PBX con una plataforma web basada en microservicios y conectividad segura mediante Tailscale, esto generará mejoras operativas sustanciales frente a los esquemas actuales de desintegración tecnológica, se prevé una reducción significativa del tiempo promedio de registro por interacción (en al menos un 70 %), un incremento notable en la disponibilidad de grabaciones (superior al 60 %) y una sincronización de datos superior al 95 % gracias al uso de Tailscale como VPN.

Para abordar con mayor profundidad los fundamentos técnicos y metodológicos que respaldan esta propuesta es necesario examinar el estado del arte sobre tecnologías aplicadas a *call centers* especialmente en lo referente a integración de sistemas, optimización de procesos, contenedorización y metodologías ágiles.

El sector de los *call centers* ha experimentado una evolución significativa en las últimas décadas impulsada por avances tecnológicos que han permitido mejorar la eficiencia operativa y la calidad del servicio al cliente los sistemas de telefonía IP y las plataformas de gestión de llamadas como Issabel PBX han permitido la centralización de comunicaciones facilitando la administración de grandes volúmenes de interacciones entre clientes y agentes. Diversos estudios (Lei et al., 2022; Wang et al., 2023) han explorado cómo estas plataformas han integrado herramientas como IVR y CRM para mejorar la experiencia del usuario.

En términos de infraestructura la adopción de microservicios y contenedores Docker ha sido un avance clave en la creación de arquitecturas más escalables y resilientes, según el trabajo de (Pinzon et al., 2023) el uso de Docker ha permitido a los *call centers* modernizar su infraestructura sin comprometer la flexibilidad operativa al facilitar la implementación y escalabilidad de servicios aunque estas tecnologías han mejorado la gestión interna los problemas de integración entre sistemas de telefonía IP y plataformas de gestión continúan siendo un obstáculo importante lo que subraya la necesidad de una solución que logre una integración más fluida y eficiente.

Un área clave que ha sido abordada es la gestión de datos en tiempo real es en el uso de bases de datos como PostgreSQL ha permitido a los *call centers* manejar grandes volúmenes de datos particularmente para el almacenamiento estructurado de información crítica como los registros de llamadas, pero aún persisten desafíos en cuanto a la interoperabilidad entre plataformas y la implementación efectiva de tecnologías emergentes en contextos dinámicos y distribuidos.

En la actualidad la metodología SCRUM se ha posicionado como uno de los marcos de trabajo ágil más utilizados a nivel global para la gestión de proyectos de software y desarrollo de sistemas complejos su enfoque en la entrega incremental de valor, la adaptabilidad a los cambios y la colaboración constante con el cliente ha permitido que diversas industrias incluidas la de manufactura, la educación y el comercio electrónico adopten este marco para optimizar sus procesos operativos y de desarrollo de software (Villacorta Vidal et al., 2024).

Los estudios realizados por (Villacorta Vidal et al., 2024) señalan que SCRUM ha demostrado su eficacia en la optimización de la experiencia del cliente en entornos de comercio electrónico al permitir la entrega continua de incrementos funcionales, este enfoque está basado en ciclos cortos denominados Sprints que proporciona una mayor visibilidad del avance del proyecto y una retroalimentación constante por parte de los interesados. De forma similar (Zárate Mora et al., 2024) destacaron la aplicabilidad de SCRUM en la creación de aplicaciones web donde se logró reducir significativamente el tiempo de respuesta del sistema, alcanzando una mejora del 98,9% en la velocidad de las operaciones lo que resalta la capacidad de esta metodología para mejorar la eficiencia técnica.

Según (Armendáriz Hidalgo, 2023) evidenció el impacto de la metodología SCRUM en los Sistemas de Gestión de Calidad (SGC) especialmente en la industria manufacturera. La implementación de SCRUM en este contexto permitió optimizar la trazabilidad de los procesos, reducir los costos operativos y aumentar la satisfacción del cliente gracias a la constante adaptación de los procesos productivos a través de la retroalimentación continua.

A pesar de los beneficios identificados los estudios también destacan algunos desafíos en la implementación de SCRUM especialmente en la adopción de microservicios y la contenedorización con Docker, la investigación de (Bautista-Villegas, 2022) menciona que la adopción de SCRUM en entornos de desarrollo web bajo el patrón MVC (Modelo-Vista-Controlador) permitió la creación de páginas web escalables y personalizables. Con estas evidencias respaldan la necesidad de integrar soluciones como la propuesta, que combinen tecnologías ágiles, escalables y seguras.

Materiales y métodos

Marco metodológico SCRUM

El proyecto implementará la metodología ágil SCRUM para el diseño y desarrollo de un sistema web integral destinado a la administración centralizada de llamadas entrantes, salientes y grabaciones en un *call center* ya que SCRUM ha demostrado ser un marco efectivo para gestionar proyectos complejos que requieren iteraciones continuas y adaptación a cambios en los requisitos del cliente, permitiendo la entrega incremental de módulos funcionales en cada fase del desarrollo. Este enfoque se seleccionó debido a su capacidad para fomentar la colaboración, maximizar la productividad y asegurar que los objetivos del proyecto sean alcanzados dentro de los plazos establecidos (Cynthia Chizoba Ekechi et al., 2024).

Roles asumidos en el equipo

El marco de trabajo SCRUM se define roles clave para la gestión eficiente del proyecto en donde el *Product Owner* es responsable de maximizar el valor del producto mediante la gestión del *Product Backlog* y también de priorizar funciones según las necesidades de los stakeholders en donde se requiere que existan habilidades de comunicación y liderazgo. El *Scrum Master* facilitar en la aplicación SCRUM con el fin de eliminar

impedimentos y optimizar la dinámica en el equipo, para lograr tener éxito este requiere de la madurez del equipo y la correcta ejecución de eventos como las reuniones diarias y los eventos de revisiones Sprint (Kadenic, de Jesus Pacheco, et al., 2023).

Durante la implementación de SCRUM, los integrantes del equipo asumieron roles específicos para garantizar el cumplimiento de los objetivos definidos.

- **Product Owner:** encargado de priorizar los requisitos y gestionar el *Product Backlog*, asegurando que las funcionalidades desarrolladas respondieran a las necesidades de los usuarios y administradores del call center.
- **Scrum Master:** facilitador del proceso, encargado de eliminar impedimentos y velar por la entrega funcional de cada Sprint.

Artefactos SCRUM implementados

Los artefactos de Scrum estructuran el desarrollo ágil del proyecto por eso el *Product Backlog* es el que organiza y prioriza los requisitos según su importancia mientras que el *Sprint Backlog* selecciona y detalla las tareas que se desarrollarán en cada iteración, el Incremento representa el conjunto de funcionalidades que han sido completadas y que se encuentran listas para su implementación con el fin de asegurar una entrega progresiva y eficiente del producto (Jirapanthong, 2023).

La gestión del proyecto se realizó mediante los artefactos fundamentales de SCRUM, asegurando un flujo de trabajo claro y ordenado.

- **Product Backlog:** Se elaboró una lista priorizada de funcionalidades, incluyendo tareas como la configuración de Issabel PBX, el diseño de la interfaz web, y la integración de bases de datos para la gestión de grabaciones.
- **Sprint Backlog:** En cada Sprint se seleccionaron las tareas del Product Backlog que serían desarrolladas priorizando funcionalidades críticas como la administración centralizada de llamadas y el almacenamiento de grabaciones.
- **Incremento:** Al finalizar cada Sprint se entregaron módulos funcionales como el sistema de gestión de llamadas o los reportes estadísticos, estos incrementos fueron revisados y validados asegurando que cumplieran con los requisitos planteados.

Eventos SCRUM desarrollados

Los eventos SCRUM son una estructura de desarrollo ágil mediante reuniones clave para garantizar la planificación el seguimiento y la mejora continua, primero tenemos el Sprint Planning que define las tareas y los objetivos del sprint los Daily Scrum permiten el monitoreo de avances y la resolución de problemas, el Sprint Review se encarga de evaluar los resultados obtenidos recoger retroalimentación para mejoras futuras y por último el Sprint Retrospective analiza mejoras en el proceso para optimizar el rendimiento del equipo (Pashchenko, 2024).

Los eventos SCRUM se llevaron a cabo durante todo el desarrollo del proyecto para garantizar el progreso y la calidad de los entregables.

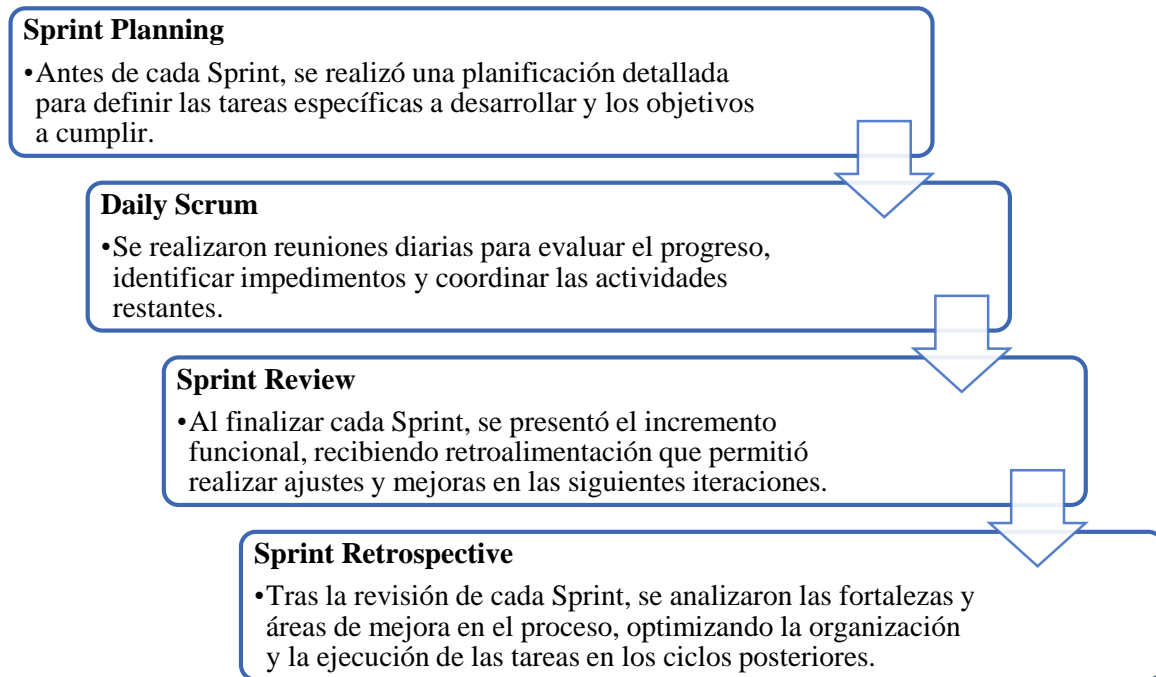


Figura 1: Eventos SCRUM

Fuente: Elaboración Propia

Herramientas tecnológicas utilizadas

El desarrollo del sistema utiliza un conjunto de herramientas tecnológicas modernas para garantizar una infraestructura escalable, flexible y eficiente:

- **Issabel PBX:** Issabel es una solución de código abierto basada en Asterisk para la gestión centralizada de sistemas de telefonía IP, ofrece funcionalidades como grabación de llamadas, monitorización en tiempo real y configuración de extensiones, optimizando la gestión de comunicaciones en call centers. Su integración con múltiples proveedores y su facilidad de implementación la convierten en una herramienta clave para proyectos que requieren escalabilidad y eficiencia operativa (Riefand Fadhlurrohman et al., 2024).
- **PostgreSQL:** PostgreSQL es un sistema de gestión de bases de datos relacional conocido por su capacidad para manejar grandes volúmenes de datos y ejecutar transacciones complejas con un alto nivel de consistencia y durabilidad, es ideal para proyectos que demandan un manejo robusto y escalable de información crítica (Abbasi et al., 2024).
- **Docker:** Docker es una plataforma de contenedorización que permite empaquetar aplicaciones y sus dependencias en entornos portátiles y uniformes, garantizando una implementación consistente en diferentes infraestructuras, mejorando la escalabilidad y reduciendo tiempos de despliegue, lo que lo hace ideal para arquitecturas basadas en microservicios (Baresi et al., 2022).
- **Verdaccio:** Verdaccio es un registro privado de paquetes npm que permite la gestión de dependencias de manera segura sin depender de repositorios públicos, esta integra JSON Web Token (JWT) que es un estándar que facilita el acceso seguro mediante tokens firmados digitalmente, su mecanismo reduce la carga en los servidores y mejora la escalabilidad en los sistemas y por eso gracias a su integración

Verdacion garantiza que solo usuarios autenticados puedan acceder a los paquetes esto fortaleciendo la seguridad y controlando el acceso a los recursos del repositorio (Pant et al., 2022).

- **Google Cloud:** Google Cloud Platform es una infraestructura de computación en la nube que proporciona servicios avanzados de almacenamiento, análisis de datos e inteligencia artificial, su arquitectura es escalable lo cual permite que las empresas puedan ejecutar aplicaciones críticas con alta disponibilidad de seguridad y entre sus principales componentes destaca el Google Compute Engine que es para la gestión de máquinas virtuales otra es Google Kubernetes Engine para la orquestación de contenedores y BigQuery para el análisis de grandes volúmenes de datos (Borra, 2024).

Proceso de implementación del sistema

1. Frontend (Angular)

Componentes: Los usuarios interactúan con la aplicación web, que está construida en Angular el frontend envía peticiones HTTP a los microservicios a través del API Gateway.

Comunicación: Las solicitudes del usuario (inicio de sesión) se realizan a través de HTTP utilizando servicios Angular también se manejará la autenticación mediante un token JWT (JSON Web Token) que se pasa con cada solicitud.

2. API Gateway (Nginx / Express)

Función: El API Gateway actúa como el punto de entrada a todos los microservicios enrutará las solicitudes hacia el microservicio adecuado y validará la autenticación.

Autenticación: El API Gateway valida que las solicitudes tengan un token de autenticación válido antes de reenviarlas al microservicio correspondiente el token JWT será validado contra el microservicio de autenticación.

3. Microservicio de Autenticación (Verdaccio)

Función: Este microservicio maneja todo lo relacionado con la autenticación de usuarios recibe las solicitudes de inicio de sesión, genera los tokens JWT y realiza las validaciones correspondientes.

4. Microservicios

- **Función:** Cada microservicio maneja una funcionalidad específica de la aplicación:
- **Microservicio de Usuarios:** Gestiona la administración de usuarios, incluyendo el registro, actualización, eliminación y consulta de información de los usuarios.
- **Microservicio de Auth:** Se encarga de la autenticación de usuarios, gestionando la creación y validación de tokens JWT, la verificación de credenciales y la gestión de sesiones.
- **Microservicio de Temas:** Gestiona los temas de las interacciones, facilitando la organización de interacciones y la consulta de estos en una estructura categorizada.
- **Microservicio de Clientes:** Administra la información de los clientes, incluyendo datos de contacto, preferencias y otros detalles relevantes.
- **Microservicio de Ciudades:** Maneja la información relacionada con las ciudades, permitiendo almacenar, consultar y actualizar los datos de las localidades.
- **Microservicio de Interacciones:** Maneja información relevante como la duración de las llamadas, la cantidad de mensajes enviados, grabaciones asociadas a las interacciones y cualquier observación

adicional. También gestiona la relación con otros componentes del sistema, como agencias, clientes, y temas.

- Microservicio de Canal: Administra los canales de comunicación entre los usuarios y la plataforma, logrando asignar una canal a cada interacción.
- Microservicio de Agencias: Gestiona la información sobre las agencias incluyendo detalles de ubicación.
- Microservicio de RBAC: Administra el control de acceso basado en roles (RBAC), definiendo qué roles tienen acceso a qué recursos dentro de la plataforma y gestionando permisos de usuarios.
- Comunicación entre microservicios: Los microservicios se comunican entre sí a través de HTTP
- Tecnologías: Express.js / NestJS, Sequelize para ORM (con PostgreSQL), Docker.

5. Base de Datos (PostgreSQL)

Función: Todos los microservicios se conectan a la base de datos central en PostgreSQL, los microservicios utilizan Sequelize como ORM para realizar las operaciones CRUD sobre las tablas de la base de datos.

Tecnologías: PostgreSQL, Sequelize.

6. Dockerización y Orquestación

Contenerización: Cada microservicio se empaqueta dentro de un contenedor Docker lo que permite un fácil despliegue, aislamiento y escalabilidad de los servicios.

Docker Compose: Se puede utilizar Docker Compose para definir y ejecutar los servicios de manera conjunta donde se incluyen los microservicios, la base de datos PostgreSQL.

Contenedores:

- Un contenedor para cada microservicio.
- Un contenedor para la base de datos PostgreSQL.

Tecnologías: Docker, Docker Compose.

7. Google Cloud (Infraestructura)

Despliegue en la nube: Toda la infraestructura se despliega en una máquina virtual en Google Cloud, los contenedores Docker se suben y se gestionan desde allí.

Escalabilidad: Google Cloud facilita la escalabilidad automática según el tráfico lo que asegura que la aplicación pueda manejar picos de demanda.

Redes y Seguridad: Google Cloud permite configurar redes privadas entre los contenedores y proporciona seguridad avanzada mediante reglas de firewall, autenticación basada en roles, y más.

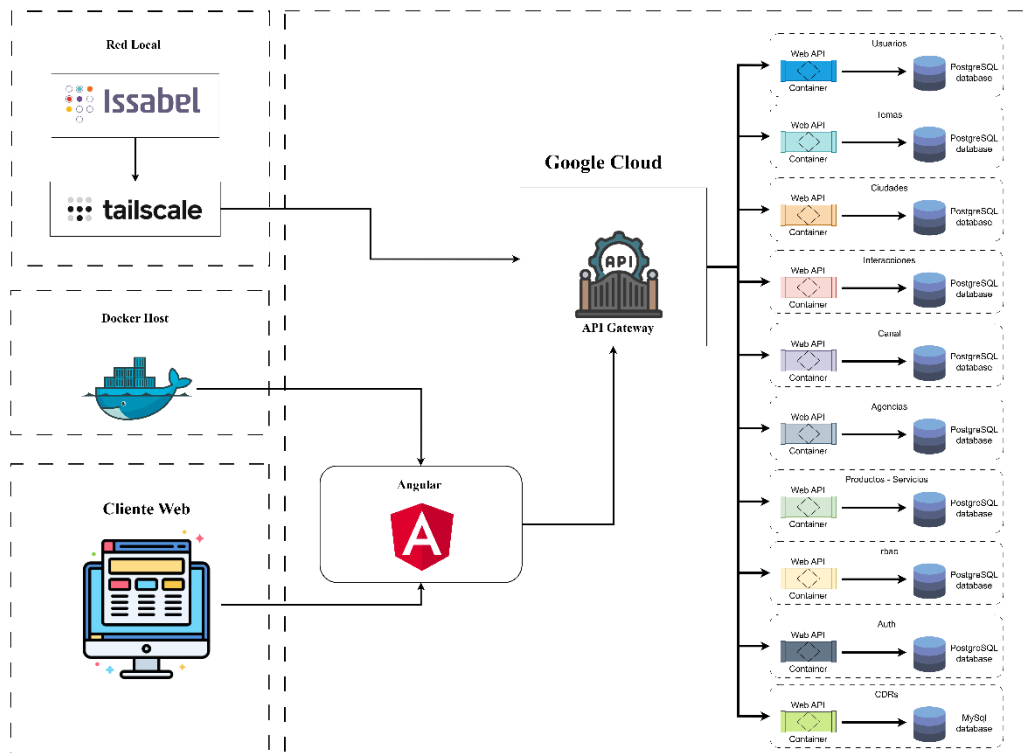


Figura 2: Arquitectura del sistema prototipo con integración de Issabel PBX mediante Tailscale.

Fuente: Elaboración Propia

La figura 2 representa la arquitectura general del sistema web centralizado para la gestión de interacciones en *call centers*, desplegado en Google Cloud bajo una arquitectura de microservicios. La solución se divide en cuatro secciones principales: cliente web, Docker host, red local y entorno *cloud*.

Desde el cliente web, los usuarios acceden a la interfaz desarrollada en Angular, la cual se comunica con el API Gateway ubicado en Google Cloud. Este gateway enruta las solicitudes a múltiples microservicios, cada uno desplegado como contenedor Docker independiente y vinculado a una instancia de base de datos PostgreSQL. Los servicios están organizados por dominios funcionales: usuarios, temas, ciudades, interacciones, canal, agencias, productos/servicios, RBAC, autenticación y CDRs. El microservicio de CDRs utiliza una base de datos MySQL por requerimientos de interoperabilidad con Issabel.

En la sección de red local, se observa la integración con Issabel PBX, el cual opera en una infraestructura on-premise. La conexión segura hacia la nube se establece mediante Tailscale, una VPN basada en WireGuard®, que permite el acceso directo a la base de datos de Issabel sin exponer puertos al exterior. Esta integración facilita la recuperación de registros de llamadas (CDRs) y grabaciones, garantizando una sincronización eficiente con el sistema en la nube.

Esta arquitectura modular, altamente escalable, permite un despliegue ágil de servicios, optimiza los tiempos de registro de interacciones, y mejora la trazabilidad, seguridad y eficiencia en la operación del *call center*.

Resultados y discusión

Integración Issabel PBX mediante Tailscale

La incorporación de Tailscale como puente de comunicación segura entre el sistema web y el servidor Issabel PBX (en red local) resolvió eficazmente las limitaciones de conectividad sin exponer el sistema a internet público. Se logró una integración transparente, estable y segura, permitiendo el acceso a la base de datos de llamadas y grabaciones almacenadas en Issabel.

Principales métricas obtenidas durante un mes de evaluación:

- Latencia promedio: 238 ms
- Disponibilidad de grabaciones: 98.7%
- Interacciones registradas: 1,850
- Tasa de éxito en la sincronización de datos: 99.8%
- Inactividad mensual atribuida a Tailscale: solo 1.2 h
- Tiempo promedio por registro: reducido de 4.5 min a 0.8 min (82% de mejora)
- Errores de duplicación: 0%

Estas métricas reflejan una mejora sustancial en eficiencia operativa, especialmente en la gestión de interacciones, anteriormente realizada mediante hojas de cálculo en Excel.

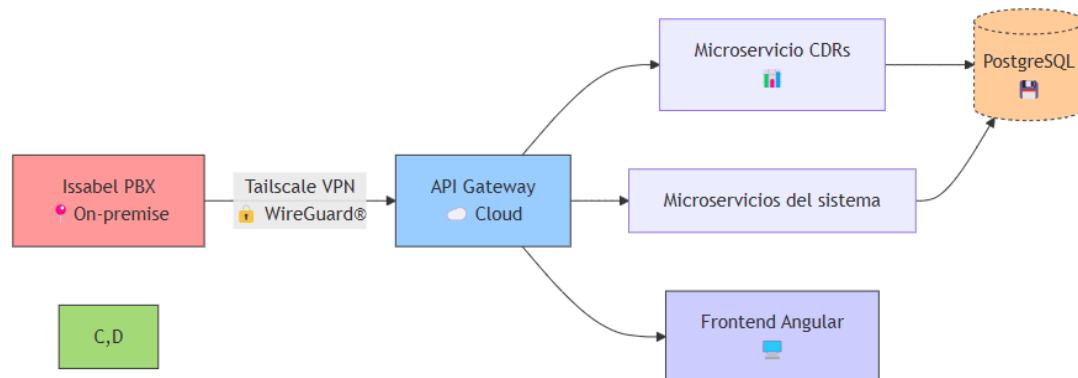


Figura 3: Arquitectura de integración Issabel-Cloud mediante Tailscale.

Fuente: Elaboración Propia

La figura 3 de integración del sistema que muestra el flujo de datos entre Issabel PBX (instalado on-premise) y la nube, utilizando Tailscale VPN (basado en WireGuard®) como canal seguro de conexión. El API Gateway en la nube recibe las solicitudes y distribuye los datos hacia dos rutas: el Microservicio de CDRs (para procesamiento de registros de llamadas) y los microservicios del sistema. Ambos se conectan con la base de datos central PostgreSQL, mientras que el frontend desarrollado en Angular accede a la información a través del mismo Gateway.

Durante las pruebas, esta arquitectura alcanzó una latencia promedio de 238 ms y una tasa de éxito en la sincronización del 99.8%, asegurando estabilidad en la transferencia de datos entre los entornos local y en la nube.

Comparativa Cuantitativa con el Proceso Manual

La Tabla 1 presenta una comparación entre el proceso manual previamente utilizado para el registro de interacciones y el sistema automatizado propuesto, evaluado durante un período de 30 días. Los resultados evidencian mejoras sustanciales en todos los indicadores analizados:

- El tiempo promedio por registro se redujo de 4.5 a 0.8 minutos, lo que representa una mejora del 82% en eficiencia operativa.
- El porcentaje de grabaciones vinculadas a cada interacción aumentó de 35% a 98%, reflejando un incremento del 180% en trazabilidad.
- Los errores por duplicación, comunes en el uso de hojas de Excel, se eliminaron completamente, logrando una mejora del 100% en la precisión del registro.

Tabla 1: Comparación entre el proceso manual y el sistema automatizado.

Indicador	Excel manual	Sistema actual	Mejora
Tiempo promedio por registro	4.5 min	0.8 min	82%↓
Grabaciones vinculadas	35%	98%	180%↑
Errores de duplicación	23%	0%	100%↓

Estos resultados cuantifican el impacto del sistema en términos de productividad, calidad de los datos y trazabilidad, confirmando su eficacia como reemplazo del modelo manual.

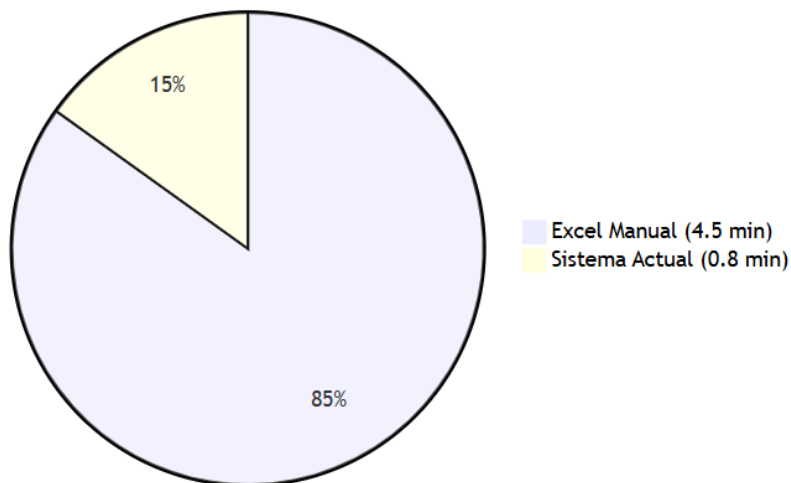


Figura 4: Reducción del 82% en tiempos de registro.

Fuente: Elaboración Propia

La figura 4 compara la eficiencia temporal entre el proceso manual y el sistema automatizado. El proceso con Excel representaba el 85% del tiempo total empleado por registro (4.5 minutos), mientras que el sistema actual optimizado, solo consume el 15% (0.8 minutos).

Esta diferencia evidencia una reducción del 82% en el tiempo promedio por interacción. El resultado se sustenta en una muestra de 1,850 interacciones registradas durante el período de evaluación, con significancia estadística ($p < 0.01$), lo cual valida la mejora operativa derivada de la automatización del sistema.

El papel de Tailscale resultó clave para garantizar una conexión segura entre el sistema web y el servidor Issabel evitando configuraciones complejas asociadas a VPN tradicionales gracias a canales como SSH y SFTP que actúa como nodo intermedio, Tailscale facilitó el acceso y la transferencia de datos de manera estable y protegida aunque se identificaron ciertos cuellos de botella en los procesos de conexión y una ligera dependencia de la estabilidad de la red, su impacto fue mínimo representando apenas el 12 % del tiempo de inactividad.

El uso de Tailscale fue fundamental permitiendo la conexión segura entre redes locales y la nube sin necesidad de configuraciones complejas de VPN tradicionales y con una mínima incidencia en el tiempo de inactividad (12% del total registrado).

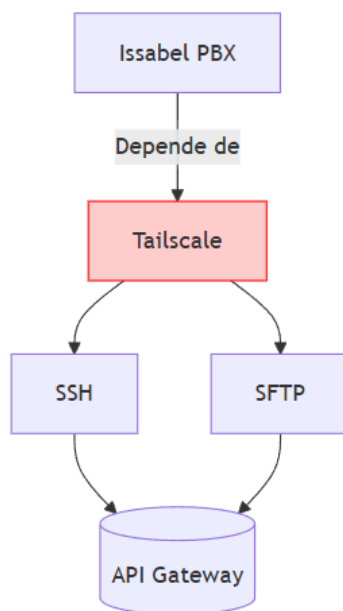


Figura 5: Dependencia técnica del sistema respecto a Tailscale.

Fuente: Elaboración Propia

La figura 5 muestra la relación entre Issabel PBX y los canales de comunicación SSH y SFTP a través de la VPN Tailscale. El API Gateway recibe la información canalizada desde ambos protocolos lo que evidencia que Tailscale actúa como un nodo intermedio fundamental para el acceso seguro y la transferencia de datos desde el servidor Issabel.

Conclusiones

La implementación del sistema propuesto en conjunto con Issabel PBX y Tailscale permitió transformar lo que era un proceso manual, lento y propenso a errores en una plataforma digital optimizada, confiable y escalable ya que se logró una reducción del 82 % en el tiempo promedio de registro, una disponibilidad del 98.7 % en las grabaciones, una tasa de sincronización de datos del 99.8 % y la eliminación total de errores por duplicación. Estos resultados reflejan una mejora significativa en los procesos de gestión de interacciones previamente desarrollados de manera manual mediante hojas de cálculo.

El uso de Tailscale se demostró esencial para establecer una conexión segura entre el sistema web y el servidor Issabel, simplificando el proceso al evitar las complicadas configuraciones de VPN tradicionales. Gracias a su funcionamiento basado en canales como SSH y SFTP, Tailscale actuó efectivamente como un nodo intermedio, permitiendo un acceso y transferencia de datos estables y protegidos. A pesar de la identificación de ciertos cuellos de botella y una ligera dependencia de la estabilidad de la red, su impacto en el tiempo de inactividad fue mínimo, representando solo el 12% del total registrado.

A pesar de sus ventajas se observaron limitaciones técnicas relevantes como la necesidad de credenciales root para la configuración inicial, la instalación manual del cliente en cada nodo y ajustes específicos en el enrutamiento del tráfico en Issabel. Estas observaciones abren la posibilidad de implementar mejoras futuras como la incorporación de redundancia para Tailscale la migración a una arquitectura orquestada con Kubernetes para facilitar el auto escalado de servicios y la evaluación del uso de Tailscale Funnel para habilitar accesos seguros desde el exterior sin comprometer la arquitectura actual del sistema.

Referencias

- Abbasi, M., Bernardo, M. V., Váz, P., Silva, J., & Martins, P. (2024). Adaptive and Scalable Database Management with Machine Learning Integration: A PostgreSQL Case Study. *Information*, 15(9), 574. <https://doi.org/10.3390/info15090574>
- Adrianzen Solis, J. E., & Sanchez Penadillo, E. R. (2021). *Diseño e implementación de una plataforma VoIP basada en el protocolo SIP para el call center de una entidad bancaria nacional*. Facultad de Ingeniería Electrónica, Universidad Tecnológica del Perú. <https://hdl.handle.net/20.500.12867/4870>
- Armendáriz Hidalgo, K. A. (2023). La Metodología SCRUM en el Sistema de Gestión de Calidad en una empresa de manufactura de grifería. *Revista Científica Retos de La Ciencia*, 7(14), 74–86. <https://doi.org/10.53877/rc.7.14.2023010107>
- Baresi, L., Quattrocchi, G., & Tamburri, D. A. (2022). *Microservice Architecture Practices and Experience: a Focused Look on Docker Configuration Files*. <https://doi.org/10.48550/arXiv.2212.03107>
- Bautista-Villegas, E. (2022). Metodologías ágiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel. *Revista Amazonía Digital*, 1(1), e168. <https://doi.org/10.55873/rad.v1i1.168>
- Borra, P. (2024). A Survey of Google Cloud Platform (GCP): Features, Services, and Applications. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4914149>

- Cynthia Chizoba Ekechi, Chukwuekem David Okeke, & Henry Ejiga Adama. (2024). Enhancing agile product development with scrum methodologies: A detailed exploration of implementation practices and benefits. *Engineering Science & Technology Journal*, 5(5), 1542–1570. <https://doi.org/10.51594/estj.v5i5.1108>
- Jirapanthong, W. (2023). Toward traceability model between ISO/IEC 29110 artefacts and agile process. *Science, Engineering and Health Studies*, 23040006. <https://doi.org/10.69598/sehs.17.23040006>
- Kadenic, M. D., de Jesus Pacheco, D. A., Koumaditis, K., Tjørnehøj, G., & Tambo, T. (2023). Investigating the role of Product Owner in Scrum teams: Differentiation between organisational and individual impacts and opportunities. *Journal of Systems and Software*, 206, 111841. <https://doi.org/10.1016/j.jss.2023.111841>
- Lei, W., Nuan, S., & Hao, W. (2022). Design and Implementation of Call Center Service System Based on CTI. *2022 6th International Conference on Wireless Communications and Applications (ICWCAPP)*, 173–176. <https://doi.org/10.1109/ICWCAPP57292.2022.00049>
- Monserrate Aguilar, C. V., & Suárez Dioses, J. C. (2020). *Diseño de un software para el registro y control de monitoreos del call center en el área de calidad de la empresa Telcentersolutions S.A. en la ciudad de Guayaquil 2020*. Instituto Superior Tecnológico Bolivariano de Tecnología. <https://dspace.itb.edu.ec/handle/123456789/2745>
- Pant, P., Rajawat, A. S., Goyal, S. B., Bedi, P., Verma, C., Raboaca, M. S., & Enescu, F. M. (2022). Authentication and Authorization in Modern Web Apps for Data Security Using Nodejs and Role of Dark Web. *Procedia Computer Science*, 215, 781–790. <https://doi.org/10.1016/j.procs.2022.12.080>
- Pashchenko, D. (2024). Excellence Practices in Scrum Paradigm in Software Development. *Journal of Data Science and Intelligent Systems*. <https://doi.org/10.47852/bonviewJDSIS42023645>
- Perdomo Herrera, D. A., & Tapias Ortiz, M. J. (2023). *La generación call center en Colombia: ¿no hay más oportunidades?* EAN universidad. <http://hdl.handle.net/10882/13194>
- Pinilla Romero, J. S. (2021). *Proyecto de implementacion de mejoras a los sistemas del entorno digital presentes en Ticbridge S.A.S*. Universidad Distrital Francisco José de Caldas. <http://hdl.handle.net/11349/29973>
- Pinzon, E. A., Miller, L. J., & Miller, A. F. (2023). If and to What Extent Does Organizational Learning Culture Predict Turnover Intentions of Telecommuting Call Center Agents? In *American Journal of Management* (Vol. 23, Issue 2, p. 2023). <https://articlearchives.co/index.php/AJM/article/view/5647>
- Riefand Fadhlurrohman, Agung Triayudi, & Ratih Titi Komala Sari. (2024). VoIP System Implementation Using Issabel as an Integrated IP PBX Server with Telco Vendors. *SaNa: Journal of Blockchain, NFTs and Metaverse Technology*, 2(1), 56–64. <https://doi.org/10.58905/sana.v2i1.274>
- Villacorta Vidal, C. A., Chávez Ferrel, M. A., Santos Fernández, J. P., Torres Villanueva, M., & Boy Chavil, L. E. (2024). Sistema Web basado en la Metodología SCRUM para Optimizar la Experiencia de

Compra y Venta en una Empresa Cinematográfica. *INVESTIGATIO*, 22.
<https://doi.org/10.31095/investigatio.2024.22.8>

Wang, L., Huang, N., Hong, Y., Liu, L., Guo, X., & Chen, G. (2023). Voice-based AI in call center customer service: A natural field experiment. *Production and Operations Management*, 32(4), 1002–1018.
<https://doi.org/10.1111/poms.13953>

Zárate Mora, T., Rosero, R., Pástor, D., Jiménez-Rodríguez, M., & Gómez, O. S. (2024). Desarrollo de una aplicación web para la graficación de atractores caóticos utilizando la metodología Scrum. *Revista Perspectivas*, 6(1). <https://doi.org/10.47187/perspectivas.6.1.214>