

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERÍA Y ARQUITECTURA
Escuela Profesional de Ingeniería de Sistemas



**Informe de desempeño profesional desarrollado en WELL
DONE SOLUTIONS (2023-2026)**

Trabajo de Suficiencia Profesional para Obtener el Título Profesional de
Ingeniero de Sistemas

Autor:

Michael Soncco Ramos

Asesor:

Mtro. Gómez Apaza, Roel Dante

Juliaca, agosto de 2026

DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE SUFICIENCIA PROFESIONAL

Yo Mg. **Roel Dante Gómez Apaza**, docente de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que la presente investigación titulada: **“INFORME DE DESEMPEÑO PROFESIONAL DESARROLLADO EN WELL DONE SOLUTIONS (2023-2026)”** del autor Michael Soncco Ramos, tiene un índice de similitud de 18 % verificable en el informe del programa Turnitin, y fue realizada en la Universidad Peruana Unión bajo mi dirección.

En tal sentido asumo la responsabilidad que corresponde ante cualquier falsedad u omisión de los documentos como de la información aportada, firmo la presente declaración en la ciudad de Juliaca, a los 18 días del mes de agosto del año 2026.



Mg. Roel Dante Gómez Apaza

ACTA DE SUSTENTACIÓN DE TRABAJO DE SUFICIENCIA PROFESIONAL



En Puno, Juliaca, Villa Chullunquiani, a 16 día(s) del mes de agosto del año 2023 a las 10:10 horas, se reunieron los miembros del jurado en la Universidad Peruana Unión Campus Juliaca, bajo la dirección del (de la) presidente(a):

Mg. Nancy Esther Casildo Bedón, el (la) secretario(a): Mg. David Mamani Pati
y los demás miembros: Mg. Helida Huaman Pazo
Mg. Moises Condeni Chañi y el (la) asesor(a) Mg. Rael Dante Gomez Apaza

con el propósito de administrar el acto académico de sustentación del trabajo de suficiencia profesional titulado:

"Informe de desempeño profesional desarrollado en WELL DONE SOLUTIONS (2023-2026)"

del(los) bachiller/es: a) Soncco Ramos Michael

b).

conducente a la obtención del título profesional de:

Ingeniero de Sistemas

(Denominación del Título Profesional)

El Presidente inició el acto académico de sustentación invitando al (a la) / a (los) (las) candidato(a)s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por al (a la) / a (los) (las) candidato(a)s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Bachiller (a): Soncco Ramos Michael

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
<u>Aprobado</u>	<u>18</u>	<u>A-</u>	<u>Muy Bueno</u>	<u>Sobresaliente</u>

Bachiller (b):

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al (a la) / a (los) (las) candidato(a)s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

[Firma]
Presidente/a
[Firma]
Asesor/a
[Firma]
Bachiller (a)

[Firma]
Miembro

[Firma]
Secretario/a
[Firma]
Miembro

Bachiller (b)

AGRADECIMIENTO

Agradezco en primer lugar a Dios, por darme la fortaleza, sabiduría y perseverancia necesarias para culminar esta etapa de mi formación profesional.

A mis padres, por su esfuerzo y sacrificio constante para brindarme la oportunidad de una educación superior, y por inculcarme los valores de responsabilidad y perseverancia que hoy guían mi desempeño profesional.

A mi familia en general, por su apoyo incondicional, comprensión y aliento durante todo este proceso, siendo un pilar fundamental para alcanzar esta meta.

A los docentes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Peruana Unión, por compartir sus conocimientos y experiencia a lo largo de mi formación académica, contribuyendo así a mi desarrollo profesional.

DEDICATORIA

Dedico este trabajo a mis padres, quienes con su amor, esfuerzo y sacrificio constante fueron el motor principal para alcanzar esta meta profesional. Gracias por creer siempre en mí y por su apoyo incondicional a lo largo de todo mi camino académico.

Índice

1. Datos generales de la empresa o institución	1
1.1. Razón social, RUC, dirección contacto	1
1.2. Actividad económica principal de la empresa o institución	1
1.3. Reseña histórica	1
1.4. Visión y Misión.....	3
1.4.1. Visión	3
1.4.2. Misión	3
1.5. Descripción donde el bachiller realizó sus actividades	3
2. Descripción del cargo y actividades del bachiller	4
2.1. Descripción del cargo	4
2.2. Descripción de las actividades del bachiller	5
2.3. Responsabilidades	6
2.4. Procesos donde se intervino como bachiller	7
2.5. Herramientas y metodologías utilizadas	8
2.5.1 Lenguajes y frameworks de desarrollo	9
2.5.2 Bases de datos.....	11
2.5.3 Infraestructura y servicios en la nube.....	13
2.5.4 Integración y procesamiento de datos.....	14
2.5.5 Calidad, pruebas y seguridad	15
2.5.6 Seguridad y control de acceso	17
2.5.7 Control de versiones y despliegue continuo	18
2.5.8 Metodologías de trabajo.....	19
3. Principales logros del bachiller.....	20
3.1. Proyectos o programas ejecutados	20
Figura 1.....	22
Figura 2.....	23
Figura 3.....	24
3.2. Metodología utilizada	25
3.3. Documentos, informes o manuales elaborados	28
3.4. Utilidad generada.....	29
3.5. Innovaciones.....	30
3.6. Premios	31
4. Conclusiones y recomendaciones.....	32
5. Referencias	34
6. Anexos	38
Anexo A: Copia de certificado de los tres años de experiencia laboral.....	38
Anexo B: Certificado del Curso de Análisis de Datos de GOOGLE	39
Anexo C: Copia de la dictaminación favorable del informe	40
Anexo D: Copia del Título Profesional de Bachiller.....	41
Anexo E: Repositorio "Highlands" en GitHub - Well Done Solutions	42

Anexo F: Arquitectura de microservicios del proyecto "Highlands"	43
Anexo G: Repositorio "raven-be" - Proyecto Phronesys	44
Anexo H: Estructura de proyecto backend y Makefile - Talma Click.....	44
Anexo I: Documentos de Buenas Prácticas de Seguridad elaborados como Líder de Proyecto	45
Anexo J: Reunión de coordinación de proyecto - Phronesys	45
Anexo K: Ejecución de pruebas automatizadas con Pytest - Proyecto Phronesys	46
Anexo L: Estructura de proyecto Java - Pacífico Seguros	46
Anexo M: Instancia de Keycloak desplegada en AWS EC2	47
Anexo N: Informe mensual de actividades - Proyecto Talma Click.....	47
Anexo Ñ: Informe mensual de actividades - Proyecto Phronesys	48
Anexo O: Herramienta SAST y DAST	49

1. Datos generales de la empresa o institución

1.1. Razón social, RUC, dirección contacto

Razón Social : Well Done Solutions SAC

RUC : 20603175876

Dirección : Pje 1ero de Mayo 112 – Oficina 401

Promotor : Roel Dante Gómez Apaza

Celular : 936148819

1.2. Actividad económica principal de la empresa o institución

Well Done Solutions S.A.C. es una empresa peruana dedicada al desarrollo de software a la medida y a la prestación de servicios de consultoría en tecnologías de la información, fundada en mayo de 2018. Su actividad económica principal consiste en el diseño, desarrollo y mantenimiento de sistemas de información para terceros, brindando soluciones tecnológicas orientadas a la automatización de procesos empresariales y al manejo de información para empresas de distintos rubros. La empresa genera ingresos principalmente mediante contratos de desarrollo de software y soporte técnico prestados a sus clientes corporativos.

1.3. Reseña histórica

Well Done Solutions S.A.C nace en mayo del 2018 del espíritu emprendedor de dos profesionales con amplia trayectoria en consultoría y desarrollo de proyectos tecnológicos. Tras años de experiencia por caminos separados, aunaron sus conocimientos para establecer una fábrica de software dedicada a brindar soluciones innovadoras que impulsen a las empresas a mejorar sus procesos internos.

Antes de fundar la empresa, ambos habían trabajado en proyectos por separado, lo que les dio una perspectiva de primera mano sobre lo que el mercado realmente

necesita y lo que enfrentan las empresas. Ese camino inicial fue fundamental para dar el siguiente paso: establecer una empresa más estratégica, lista para abordar proyectos más complejos y construir un equipo técnico leal.

"El equipo de Well Done Solutions es joven, enérgico y colaborativo". La empresa desarrolla un ambiente de trabajo en el que el desarrollo profesional y el aprendizaje continuo son parte de su cultura. Gracias a este método, el equipo alcanza los objetivos marcados con altos estándares de calidad y en un ambiente motivador.

En más de una ocasión, han tenido que colaborar con proveedores que tienen la mentalidad de "si funciona, no lo toques". Pero en Well Done Solutions pensamos diferente: "Siempre se puede mejorar". Esta disposición a cambiar, a revisar lo que hacemos, ha dado lugar a mejores soluciones, más escalables y más sostenibles en el tiempo.

Uno de sus cofundadores, Helard Gómez, graduado de la Pontificia Universidad Católica del Perú, trae la perspectiva empresarial desarrollada en empresas como Interbank y SOA Profesional. Su visión complementa la de los técnicos y refuerza el liderazgo de la empresa.

Hoy, Well Done Solutions es parte de la nueva generación de empresas tecnológicas peruanas: ágiles, innovadoras y enfocadas en crear valor para sus clientes. Su mayor activo es su equipo humano y su futuro pasa por la colaboración, la mejora continua y el crecimiento en conjunto.

La empresa se inició con el propósito de brindar soluciones tecnológicas de alto nivel en el sur del Perú; hoy continúa expandiéndose para contribuir al desarrollo tecnológico del sur del Perú.

1.4. Visión y Misión

1.4.1. Visión

Satisfacer con excelencia técnica y de servicio las necesidades de nuestros clientes en el campo de las Tecnologías de Información, aportándoles el máximo valor y apoyándose en su Plan Informático.

1.4.2. Misión

Los servicios que presta Well Done Solutions involucran tecnología de punta y un cuidadoso estudio de las necesidades y ventajas que obtendrá con la nueva solución.

1.5. Descripción donde el bachiller realizó sus actividades

Realicé mis actividades laborales en Well Done Solutions S.A.C. bajo la modalidad de trabajo remoto (home office), utilizando mi propia estación de trabajo y accediendo de forma remota a los entornos y herramientas de la empresa. Esta modalidad implicó organizar mi propio tiempo y espacio de trabajo, sin perder por ello el ritmo ni la disciplina que exigía cada proyecto. La coordinación con el equipo se sostenía principalmente a través de reuniones diarias de seguimiento por videoconferencia, donde se compartía el avance del día y se identificaban posibles bloqueos, lo que permitía mantener una comunicación constante con el resto del equipo pese a no compartir un mismo espacio físico. A lo largo de mi permanencia en la empresa, no tuve necesidad de presencia física en oficinas, desarrollando la totalidad de mis funciones bajo este esquema de trabajo distribuido.

2. Descripción del cargo y actividades del bachiller

2.1. Descripción del cargo

En Well Done Solutions S.A.C., me desempeñé inicialmente como Desarrollador Backend, formando parte del área de desarrollo de software de la empresa. Mi función consistió en la implementación de funcionalidades y componentes del lado del servidor para distintos proyectos de la empresa, entre ellos Pacífico Seguros, Talma Click y Phronesys, cada uno con sus propias particularidades: mientras que en los dos primeros me integraba a sistemas que ya estaban en producción y en constante evolución, en Phronesys tuve la oportunidad de construir el backend prácticamente desde cero, lo que me exigió un nivel distinto de responsabilidad sobre las decisiones técnicas del proyecto.

A lo largo de estos proyectos, trabajé de la mano con el equipo de Control de Calidad (QA) y con el líder de proyecto, lo que permitió mantener un flujo de trabajo alineado con los estándares de calidad de la empresa y una comunicación constante sobre el avance y las incidencias detectadas en cada etapa del desarrollo. Esta dinámica de trabajo cercano con otras áreas fue, con el tiempo, una de las bases que me ayudó a prepararme para asumir mayores responsabilidades dentro de la empresa.

Durante los últimos meses de mi permanencia, asumí el rol de Líder de Proyecto en el desarrollo de Highlands, responsabilizándome de la coordinación del equipo de desarrollo, la planificación y priorización de tareas, y el seguimiento del avance del proyecto. Este cambio de rol representó una transición natural dentro de mi trayectoria: pasé de ejecutar tareas dentro de un flujo ya establecido, a ser quien definía cómo debía organizarse ese flujo para el equipo a mi cargo.

2.2. Descripción de las actividades del bachiller

Como Desarrollador Backend, mi trabajo se organizaba en torno a reuniones diarias de seguimiento con el equipo, donde se coordinaban las tareas del día y se reportaba el avance de los requerimientos en curso. Este ritmo diario terminó siendo, en la práctica, la columna vertebral de cómo se organizaba mi trabajo: cada mañana sabía en qué estaba parado y qué se esperaba que avanzara antes de la siguiente reunión. Para cada requerimiento asignado, mi actividad consistía en analizar lo solicitado, implementar la funcionalidad correspondiente y organizarla de forma que resultara clara y fácil de mantener dentro del sistema, algo especialmente relevante en proyectos como Phronesys, donde al construir el backend desde cero, las decisiones que tomaba sobre cómo estructurar el código terminaban afectando todo lo que se desarrollaba después. En algunos casos, esta labor incluía también la configuración inicial del entorno de trabajo del proyecto.

Una vez implementada una funcionalidad, elaboraba las pruebas necesarias para validar su correcto funcionamiento, verificando que cubrieran el comportamiento esperado antes de considerarla finalizada. Dependiendo del proyecto, esta validación podía requerir además una revisión adicional de calidad y seguridad del código, así como la aprobación del equipo de control de calidad antes de su paso a producción, un flujo que se hacía más riguroso en sistemas como el de Pacífico Seguros, donde cualquier cambio debía pasar por varios filtros antes de ser aprobado. El despliegue de los cambios validados hacia los distintos entornos, dependiendo del proyecto, estaba a cargo de un equipo especializado o formaba parte de mis propias actividades.

Como Líder de Proyecto, mis actividades se ampliaron hacia la definición de la estructura general de las soluciones a desarrollar, evaluando cómo debían organizarse y comunicarse los distintos componentes del sistema antes de su implementación, un

cambio de perspectiva importante, ya que dejé de pensar únicamente en cómo resolver una tarea puntual para empezar a pensar en cómo debía comportarse el sistema en su conjunto. Asimismo, coordinaba el trabajo del equipo, distribuyendo y dando seguimiento a las tareas según su prioridad e impacto en el proyecto, y revisaba las incidencias reportadas para asegurar que fueran atendidas de forma oportuna. También lideré la actualización de los lineamientos internos de desarrollo, con el fin de fortalecer las prácticas de seguridad aplicadas en los proyectos de la empresa, estableciendo criterios que el equipo siguió aplicando más allá de mi propia participación directa en cada tarea.

2.3. Responsabilidades

Como Desarrollador Backend, era responsable de que cada funcionalidad que entregaba contara con las pruebas necesarias para respaldar su correcto funcionamiento, no bastaba con que el código compilara o pareciera funcionar: debía demostrarlo. Esa exigencia se volvió más estricta en proyectos como Talma Click, donde ser responsable de una funcionalidad implicaba también responder por la solidez de las pruebas que la validaban. Asimismo, era responsable de resolver de forma rápida y confiable los errores reportados en producción dentro de mi área de trabajo, consciente de que un fallo no corregido a tiempo podía tener un impacto significativo en el negocio del cliente, un riesgo que se sentía de forma particular en un sistema como el de Pacífico Seguros, donde cualquier error mal manejado podía traducirse en pérdidas económicas considerables. En esta etapa, mi responsabilidad se centraba en proponer soluciones ante los problemas técnicos que se presentaban, aunque las decisiones de mayor impacto para el proyecto, como cambios de arquitectura, quedaban a cargo del líder de proyecto.

Al asumir el rol de Líder de Proyecto, mis responsabilidades cambiaron de naturaleza, no solo de alcance. Pasé a ser responsable de la definición de la estructura de las soluciones a desarrollar, teniendo que anticipar decisiones técnicas antes de que el equipo empezara a escribir código, así como de la organización y priorización del trabajo, asegurando que cada tarea se atendiera en el orden adecuado y no simplemente conforme iba surgiendo. También era responsable de dar seguimiento a las incidencias reportadas hasta su resolución, y de que los procesos internos de desarrollo del equipo se mantuvieran actualizados y alineados a buenas prácticas de seguridad, una responsabilidad que asumí de forma activa al establecer lineamientos propios sobre cómo debía gestionarse el trabajo en el repositorio. A diferencia de la etapa anterior, en este rol contaba con autonomía para tomar decisiones de forma directa sobre el proyecto a mi cargo, sin necesidad de que fueran validadas por alguien más.

2.4. Procesos donde se intervino como bachiller

A lo largo de mi experiencia en Well Done Solutions S.A.C., intervine en distintas fases del ciclo de vida del desarrollo de software, aunque el peso de mi participación en cada una fue cambiando conforme fui asumiendo mayores responsabilidades dentro de la empresa.

La fase de desarrollo fue, sin duda, en la que más tiempo invertí a lo largo de mi trayectoria. Ahí se concentraba el día a día: tomar un requerimiento, entenderlo y convertirlo en código funcional dentro del sistema correspondiente. Ya fuera integrando un nuevo servicio externo, corrigiendo la forma en que se manejaban las migraciones de base de datos, o construyendo un backend completo desde cero, esta fue la actividad que más definió mi rol como desarrollador.

En la fase de pruebas, mi participación fue más allá de simplemente verificar que el código funcionara. Intervine en la elaboración de pruebas que realmente validaran el comportamiento esperado de cada funcionalidad, y en algunos proyectos también en evaluar qué tan buenas eran esas mismas pruebas, identificando aquellas que no detectaban errores reales aunque parecieran estar cumpliendo su función. Antes de liberar cualquier cambio, formaba parte de este proceso corregir lo que fuera necesario hasta asegurar que la funcionalidad estuviera lista.

En la fase de mantenimiento, participé en la resolución de incidencias reportadas en sistemas que ya estaban en producción, brindando soporte correctivo a los proyectos en los que estuve involucrado. Este tipo de intervención exigía un cuidado distinto al del desarrollo inicial, ya que cualquier cambio debía hacerse sin afectar a un sistema que ya estaba siendo usado activamente por el cliente.

Como Líder de Proyecto, mi participación se amplió hacia fases en las que antes no intervenía directamente: el análisis y el diseño. Pasé a involucrarme en la definición de la estructura de las soluciones antes de que se implementara una sola línea de código, así como en la planificación del trabajo del equipo, organizando y priorizando las tareas a lo largo del desarrollo del proyecto. Este cambio de rol me permitió ver el ciclo de vida del software de forma más completa, no solo desde la ejecución, sino también desde las decisiones que la anteceden.

2.5. Herramientas y metodologías utilizadas

Durante mi desempeño profesional en Well Done Solutions S.A.C., utilicé una variedad de herramientas y metodologías orientadas a garantizar la calidad, eficiencia y escalabilidad de los sistemas desarrollados, organizadas en las siguientes categorías:

2.5.1 Lenguajes y frameworks de desarrollo

- **Java**



Es un lenguaje de programación orientado a objetos cuyo código se ejecuta sobre una máquina virtual (JVM) para lograr independencia de la plataforma sobre la que corre. Fue desarrollado por un equipo liderado por James Gosling en Sun Microsystems y presentado públicamente el 23 de mayo de 1995 (MDPI Encyclopedia, 2025). Lo utilicé en las versiones 11 (LTS, 2018) a 21 (LTS, 2023), según la antigüedad de cada proyecto, principalmente para el desarrollo de microservicios en Pacífico Seguros y Highlands.

- **Spring Boot**



Es un framework para Java que simplifica la configuración de aplicaciones empresariales mediante convenciones automáticas y servidores embebidos. Fue desarrollado por el equipo de Spring en Pivotal Software, con su primera versión estable lanzada el 1 de abril de 2014 (LearnCodeWithDurgesh, 2025). Lo utilicé como base para construir y mantener los microservicios en Java de los distintos proyectos.

- **Python**



Es un lenguaje de programación de alto nivel conocido por su sintaxis simple y legible. Fue creado por el programador neerlandés Guido van Rossum, quien publicó su primera versión pública en febrero de 1991 (Computer History Museum, 2021). Trabajé con las versiones 3.8 a 3.13, según la antigüedad y compatibilidad de cada proyecto, principalmente en Talma Click y Phronesys.

- **FastAPI**



Es un framework web para Python orientado a la construcción de APIs de alto rendimiento, con validación y documentación automática. Fue creado por el desarrollador colombiano Sebastián Ramírez, quien publicó su primera versión el 24 de diciembre de 2018 (So, 2024). Lo utilicé para desarrollar servicios y endpoints en Talma Click y Phronesys.

- **Django**



Es un framework web para Python de alto nivel, orientado al desarrollo rápido de aplicaciones con un enfoque de "baterías incluidas" (ORM propio, panel de administración, seguridad predeterminada). Fue creado por Adrian Holovaty y Simon Willison en el periódico *Lawrence Journal-World*, con primera versión pública liberada el 21 de julio de 2005 (Django Software Foundation, 2025). Lo utilicé junto con su ORM nativo para la persistencia de datos en Talma Click y Phronesys.

- **Liquibase**



Es una herramienta de código abierto para versionar y controlar cambios en esquemas de bases de datos. Fue creada en 2006 por Nathan Voxland (Liquibase, 2021). La introduje en Pacífico Seguros para dejar las migraciones de base de datos documentadas y reproducibles, algo con lo que el proyecto no contaba previamente.

2.5.2 Bases de datos

- **PostgreSQL**



Es un sistema de gestión de bases de datos relacionales de código abierto. Se originó en 1986 como el proyecto POSTGRES en la Universidad de California, Berkeley (PostgreSQL Global Development Group, 2026). Lo utilicé como motor relacional para la presentación de información al cliente final en Highlands.

- **MySQL**



Es un sistema de gestión de bases de datos relacionales. Fue desarrollado a partir de 1994 por Michael "Monty" Widenius y David Axmark, quienes fundaron MySQL AB en Suecia en 1995 (Buytaert, 2025). Lo utilicé como alternativa relacional en algunos de los proyectos en los que participé.

- **Cloud SQL**



Es el servicio de bases de datos relacionales completamente administrado de Google Cloud Platform para MySQL, PostgreSQL y SQL Server, que automatiza tareas de administración como copias de seguridad y alta disponibilidad (Google Cloud, 2021). Tuve contacto con este servicio en proyectos desplegados sobre infraestructura de GCP.

- **MongoDB**



Es una base de datos NoSQL orientada a documentos. Fue fundada el 24 de noviembre de 2007 en

Nueva York por Dwight Merriman, Eliot Horowitz y Kevin Ryan bajo el nombre 10gen (MatrixBCG, 2026). La utilicé para el procesamiento de información en Highlands.

- **DynamoDB**



Es el servicio de base de datos NoSQL de AWS, lanzado el 18 de enero de 2012 (Amazon Science, 2025).

Tuve contacto con este servicio en proyectos que requerían alta escalabilidad sobre infraestructura de AWS.

- **ArangoDB**



Es un sistema de base de datos NoSQL de código abierto de tipo "multimodelo", capaz de manejar documentos, grafos y pares clave-valor dentro de un mismo motor. Fue lanzada originalmente como AvocadoDB en 2011 por los ingenieros alemanes Claudius Weinberger y Frank Celler, cambiando de nombre a ArangoDB un año después (VentureBeat, 2021). La utilicé en Highlands como parte del procesamiento de información dentro de la arquitectura de microservicios.

- **Redis**



Es un sistema de almacenamiento en memoria usado como base de datos, caché y broker de mensajes.

Fue creado en 2009 por el programador italiano Salvatore Sanfilippo (Redis, 2026). Lo utilicé como sistema de caché en distintos proyectos.

- **Google BigQuery**



Es el data warehouse serverless de Google Cloud, anunciado en mayo de 2010 en Google I/O y construido

sobre Dremel, el motor de consultas interno de Google (Stacksync, 2026). Lo utilicé para el manejo de grandes volúmenes de información en proyectos sobre GCP.

2.5.3 Infraestructura y servicios en la nube

- **Amazon Web Services (AWS)**



Es la plataforma de servicios en la nube de Amazon.

Amazon S3, su primer servicio de almacenamiento, se lanzó el 14 de marzo de 2006, y AWS Lambda, su servicio de cómputo sin servidor, fue anunciado el 13 de noviembre de 2014 (AWS News Blog, 2026; Amazon Web Services, 2014). Utilicé S3 y Lambda en Phronesys para el procesamiento de datos históricos, y EC2 para el despliegue de la instancia de Keycloak.

- **Microsoft Azure**



Es la plataforma de servicios en la nube de

Microsoft. Fue presentada en octubre de 2008 y alcanzó

disponibilidad general el 1 de febrero de 2010 (Microsoft, 2010). Tuve contacto con esta infraestructura en el proyecto de Pacífico Seguros.

- **Google Kubernetes Engine (GKE)**



Es el servicio administrado de Kubernetes de Google Cloud, lanzado en 2015 (Google Cloud, 2025). Lo utilicé para la orquestación de contenedores en proyectos desplegados sobre GCP.

- **Terraform**



Es una herramienta de infraestructura como código (IaC) que permite definir y gestionar recursos de distintos proveedores cloud mediante un mismo lenguaje declarativo. Fue creada por Mitchell Hashimoto y anunciada por HashiCorp el 28 de julio de 2014, alcanzando su versión 1.0 (estabilidad de API) en junio de 2021 (Build5Nines, 2026). La utilicé en Pacífico Seguros para la gestión de la infraestructura desplegada en Azure.

- **Docker**



Es una plataforma que permite empaquetar aplicaciones y sus dependencias en contenedores. Fue presentada públicamente por Solomon Hykes el 13 de marzo de 2013 (Docker, 2024). Lo utilicé para estandarizar los entornos de desarrollo y despliegue en distintos proyectos.

- **Kubernetes**



Es un sistema de orquestación de contenedores de código abierto. Fue anunciado por Google el 6 de junio de 2014, y en junio de 2024 la comunidad celebró oficialmente su décimo aniversario (Kubernetes, 2024). Tuve contacto con esta herramienta para la orquestación de servicios en proyectos que lo requerían.

2.5.4 Integración y procesamiento de datos

- **Apache Kafka**



Es una plataforma de mensajería distribuida usada para la comunicación asíncrona entre servicios. Fue desarrollada originalmente en LinkedIn a partir de 2010 y liberada como código abierto en junio de 2011 (Factor House, 2026). La utilicé en Highlands para la comunicación entre microservicios bajo un enfoque orientado a eventos.

- **Debezium**



Es una plataforma de código abierto para la captura de cambios en bases de datos (CDC). Fue fundada en 2016 por Randall Hauch, ingeniero de Red Hat (Morling, 2024). La utilicé en Highlands para resolver la sincronización de datos entre MongoDB y PostgreSQL.

- **Apache Spark / PySpark**



Junto con su interfaz en Python (**PySpark**), es un motor de procesamiento de datos a gran escala. Se originó en 2009 como proyecto de investigación en el AMPLab de la Universidad de California, Berkeley, liderado por Matei Zaharia (Flexera, 2026). Lo utilicé en Highlands para el procesamiento de grandes volúmenes de datos.

2.5.5 Calidad, pruebas y seguridad

- **SonarQube**



Es una herramienta de análisis estático de código (SAST) que identifica errores, vulnerabilidades y malas prácticas. Fue creada por Simon Brandhof en 2007, y la empresa SonarSource se fundó al año siguiente (Sonar, 2025). La utilicé en Pacífico Seguros y otros proyectos como parte del control de calidad previo al despliegue.

- **JUnit**



Es un framework de pruebas unitarias para Java. Fue creado por Kent Beck y Erich Gamma en 1997 (GOTO Copenhagen, 2025). Lo utilicé para elaborar pruebas unitarias en los proyectos desarrollados en Java.

- **Pytest**



Es un framework de pruebas para Python. Se originó dentro del proyecto PyPy, cuando Holger Krekel comenzó a desarrollarlo en 2004 (pytest documentation, s.f.). Lo utilicé para elaborar pruebas unitarias en los proyectos desarrollados en Python.

- **Mutmut**



Es una herramienta de pruebas de mutación para Python que evalúa la calidad real de las pruebas automatizadas de un proyecto, más allá de si estas simplemente se ejecutan sin errores. Es mantenida por el programador sueco Anders Hovmöller (Hovmöller, 2019). La incorporé en Talma Click, donde ayudó a reducir la frecuencia de errores en producción.

- **OWASP ZAP**



Es una herramienta de código abierto para pruebas dinámicas de seguridad (DAST). Fue creada por Simon Bennetts, con primera versión pública el 6 de septiembre de 2010 (DevOpsSchool, 2026). La utilicé para identificar vulnerabilidades en aplicaciones ya en ejecución.

- **Postman**



Es una herramienta para el desarrollo y prueba de APIs. Fue creada en 2012 por Abhinav Asthana (So, 2024; ver nota más abajo). La utilicé para validar manualmente el comportamiento de los endpoints desarrollados.

2.5.6 Seguridad y control de acceso

- **Amazon Cognito**



Es el servicio de gestión de identidad de AWS, anunciado el 10 de julio de 2014 (AWS News Blog, 2024).

Tuve contacto con este servicio como alternativa de gestión de identidad en proyectos sobre AWS.

- **Azure Active Directory / Microsoft Entra ID**



Es el servicio de gestión de identidades de Microsoft Azure, introducido en octubre de 2008 y renombrado en julio de 2023 (Microsoft, 2024). Tuve contacto con este servicio en el proyecto de Pacífico Seguros.

- **Google Cloud IAM**



Es el servicio de gestión de identidad y acceso de Google Cloud Platform (Google Cloud, s.f.). Tuve contacto con este servicio en proyectos desplegados sobre GCP.

- **Keycloak**



Es una solución de código abierto para la gestión de autenticación, autorización y control de acceso, basada en estándares como OpenID Connect y OAuth 2.0. Se originó en 2013 dentro de la comunidad JBoss de Red Hat. La implementé desde cero sobre infraestructura de AWS para centralizar el control de accesos entre sistemas.

2.5.7 Control de versiones y despliegue continuo

- **GitHub**



Es una plataforma de control de versiones basada en Git. Fue lanzada oficialmente el 10 de abril de 2008 por Tom

Preston-Werner, Chris Wanstrath, PJ Hyett y Scott Chacon (Stacksync, 2026). La utilicé como repositorio principal de código en todos los proyectos.

- **GitLab**



Es una plataforma de control de versiones y DevOps. Nació el 8 de octubre de 2011 con el primer commit de Dmitriy Zaporozhets (GitLab, 2023). La utilicé como alternativa de control de versiones en algunos proyectos.

- **Jenkins**



Es una herramienta de automatización para integración y despliegue continuo. Se originó en 2004 bajo el nombre "Hudson", bifurcándose como Jenkins en enero de 2011 (Grokikipedia, 2026). La utilicé para automatizar los procesos de integración y despliegue.

- **Visual Studio Code**



Fue anunciado por Microsoft el 29 de abril de 2015 en la conferencia Build, liderado por Erich Gamma el mismo coautor de JUnit desde el laboratorio de Microsoft en Zúrich; su código fue liberado como proyecto de código abierto en noviembre de ese mismo año (Wikipedia contributors, citado en Microsoft, 2020; en su lugar: Microsoft, 2020).

- **IntelliJ IDEA**



Fue lanzado por JetBrains (entonces llamada IntelliJ Software), empresa fundada en Praga, República Checa, por los ingenieros Sergey Dmitriev, Valentin Kipyatkov y Eugene Belyaev; su primera versión salió en enero de 2001 como uno de

los primeros IDE para Java con navegación de código avanzada y refactorización integrada (JetBrains, 2021).

2.5.8 Metodologías de trabajo

El equipo de desarrollo organizó su trabajo bajo el método Kanban, visualizando el flujo de tareas en un tablero dividido por estados y sosteniendo el ritmo de trabajo mediante reuniones diarias de seguimiento, lo que permitía priorizar tareas de forma dinámica según su urgencia e impacto. Esto se complementó con algunas buenas prácticas del framework Scrum, como reuniones periódicas de revisión de avances con el equipo. El origen, la definición formal de cada uno y el detalle de cómo se aplicaron en el día a día se desarrollan en la sección 3.2 del presente informe.

3. Principales logros del bachiller

3.1. Proyectos o programas ejecutados

Entre los distintos proyectos en los que participé, destaco los siguientes por su complejidad e innovación, los cuales se encuentran actualmente desplegados:

- **Pacífico Seguros.**

Sistema desarrollado bajo una arquitectura de microservicios. Mi participación se centró en el desarrollo de nuevos requerimientos y la resolución de incidencias dentro de microservicios específicos del sistema. Entre los aportes más relevantes, integré un servicio externo que permitía obtener información vehicular a partir del número de documento del cliente, automatizando un proceso que antes requería validación manual. Asimismo, corregí incidencias relacionadas con la base de datos e incorporé Liquibase para el versionado y control de las migraciones del esquema, funcionalidad con la que el proyecto no contaba hasta entonces.

- **Talma Click.**

Sistema desarrollado bajo una arquitectura de microservicios para un escenario logístico. Mi participación consistió en el desarrollo de nuevos requerimientos reportados como tickets, varios de ellos relacionados con información que no se cargaba correctamente y que en ocasiones debía procesarse manualmente así como en la corrección de errores del sistema. En este proyecto, la incorporación de Mutmut resultó especialmente relevante, ya que ayudó a reducir la frecuencia con la que surgían nuevos errores en la producción.

- **Phronesys.**

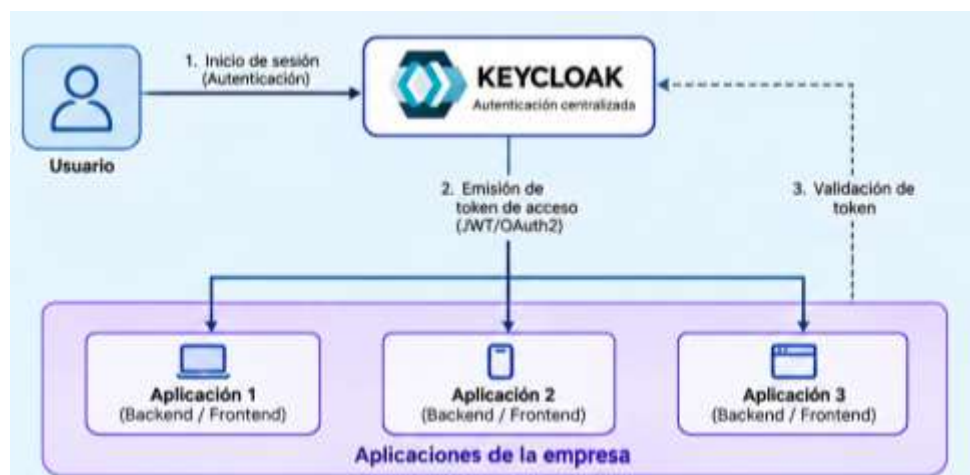
Proyecto que inicié desde cero, donde estuve a cargo de prácticamente la totalidad del desarrollo backend, combinando un backend principal con funciones serverless puntuales para escenarios específicos del sistema. Uno de los principales retos fue diseñar un microservicio dedicado exclusivamente a la carga de archivos con información histórica, un proceso que debía manejar volúmenes considerables de datos sin afectar el rendimiento del resto del sistema. Para resolverlo, apoyé este microservicio en AWS Lambda y S3, permitiendo procesar y almacenar la data histórica de forma desacoplada del backend principal. Al tratarse de un proyecto construido desde su etapa inicial, las decisiones que tomé sobre la estructura del código y la organización de estos servicios sentaron las bases sobre las cuales se continuó desarrollando el sistema en adelante.

- **Implementación de seguridad Keycloak.**

Iniciativa puntual que desarrollé desde cero como parte de mis funciones como desarrollador, orientada a resolver la falta de un sistema centralizado de autenticación y control de acceso entre los distintos sistemas de la empresa. Mi trabajo se centró en integrar Keycloak con las aplicaciones ya existentes, tanto en el backend como en el frontend, de modo que estas dejaran de gestionar su propia autenticación de forma independiente y pasarán a validar el acceso de los usuarios contra este servicio centralizado. Esto implicó adaptar la lógica de autenticación de cada aplicación para que reconociera y validará correctamente los tokens emitidos por Keycloak, en lugar de mantener mecanismos de autenticación propios y desconectados entre sí.

Figura 1

Arquitectura de autenticación centralizada mediante Keycloak



Nota. Autenticación centralizada mediante Keycloak, la emisión de tokens JWT/OAuth 2.0 y su posterior validación para permitir el acceso seguro a las aplicaciones empresariales.

- **Visualizador de indicadores.**

Mi participación se centró principalmente en la corrección de errores del microservicio: existían valores que no se cargaban correctamente, lo que impedía el funcionamiento adecuado del sistema. Al investigar la causa, identifiqué que el problema se originaba en el mapeo y transformación de los datos, donde ciertos campos no se estaban parseando correctamente antes de ser almacenados o mostrados, generando valores erróneos en la información presentada. El trabajo consistió en corregir esta lógica de transformación de datos, estabilizando el microservicio existente y asegurando que los indicadores reflejaran la información real de forma consistente.

Figura 2

Visualizador de indicadores



Nota. Diagrama de flujo del proceso de corrección, transformación y validación de datos en el microservicio para la visualización de indicadores.

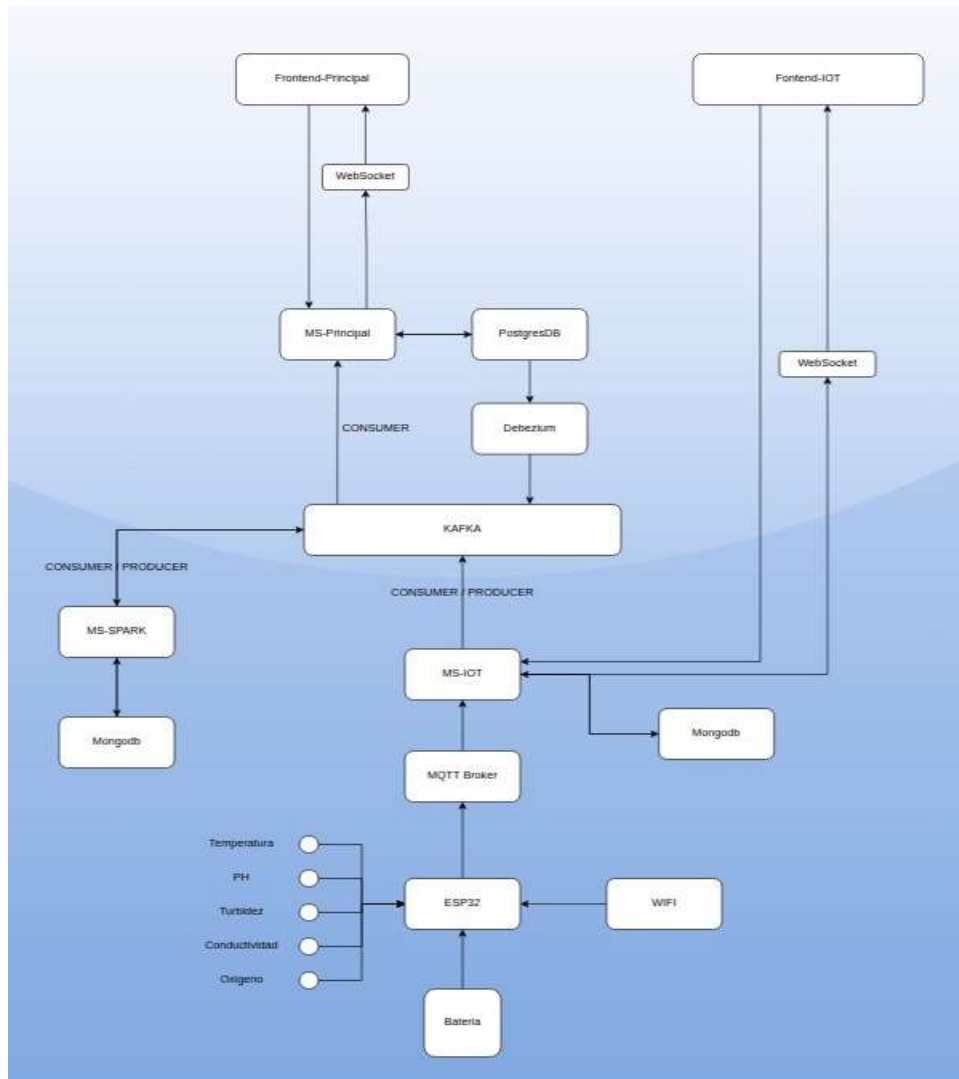
- **Highlands.**

Proyecto interno de mayor envergadura, en el que asumí el rol de Líder de Proyecto, complementando esta función con la de arquitecto de la solución. En esta doble responsabilidad, lideré la implementación de Apache Spark y Kafka dentro de una arquitectura de microservicios, utilizando MongoDB como base de datos para el procesamiento de información y PostgreSQL para la presentación de datos al cliente.

final. La incorporación de Debezium permitió resolver una problemática de sincronización de datos entre ambos motores, garantizando que la información procesada estuviera disponible de forma consistente para su visualización.

Figura 3

Arquitectura Highlands



Nota. Arquitectura del sistema Highlands, incluyendo la interacción entre los componentes frontend, microservicios, Kafka, bases de datos, dispositivos IoT y protocolos de comunicación.

3.2. Metodología utilizada

- **Kanban**

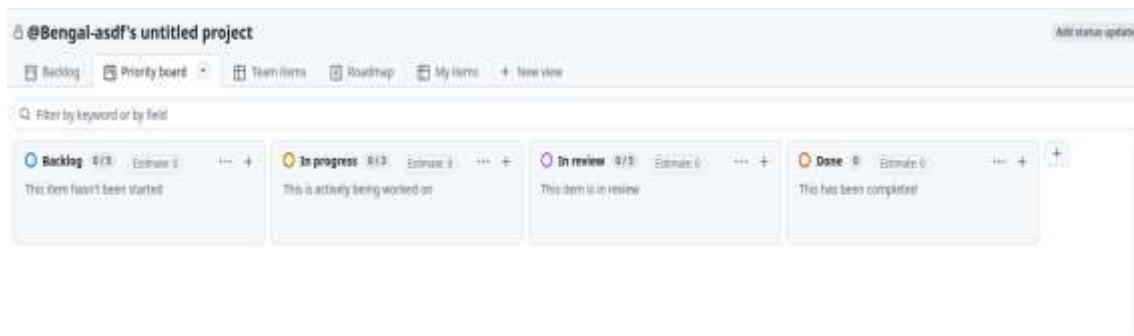
Es un método de gestión visual del flujo de trabajo, formalizado oficialmente por David J. Anderson en su libro *Kanban: Successful Evolutionary Change for Your*

Technology Business. Según la Guía Oficial del Método Kanban, publicada por Kanban University, la motivación central detrás de su creación fue encontrar una forma de gestionar y mejorar negocios de servicios profesionales, a la vez que se ofrecía un método de cambio "humano" para las organizaciones; a diferencia de la manufactura tradicional, de donde provienen sus raíces (Lean Manufacturing), el Método Kanban está pensado para gestionar trabajo de conocimiento que produce bienes y servicios intangibles o virtuales, aceptando la variabilidad en la entrega del trabajo como algo inherente al proceso (Kanban University, 2021). El método utiliza un sistema de extracción (*pull system*) con límites al trabajo en curso como mecanismo central para exponer los problemas del proceso y estimular la colaboración del equipo para resolverlos.

En el equipo de desarrollo en el que trabajé, esto se traducía en un tablero de tareas dividido por estados (por hacer, en progreso, en revisión y hecho), donde cada requerimiento avanzaba de una columna a otra conforme se completaban las actividades correspondientes. El límite al trabajo en curso se reflejaba en la práctica al no tomar más tareas de las que el equipo podía atender de forma realista al mismo tiempo, evitando así la sobrecarga y los cuellos de botella. El ritmo de trabajo se sostenía mediante reuniones diarias de seguimiento, en las que cada integrante compartía brevemente su avance y los posibles bloqueos que impidieran continuar, lo que permitía priorizar las tareas de forma dinámica según su urgencia e impacto, en lugar de seguir un plan rígido establecido de antemano.

Figura 4

Tablero Kanban del equipo de desarrollo



Nota. Tablero Kanban utilizado por el equipo, gestionado mediante GitHub, Projects (GitHub, s.f.).

- **Scrum**

A diferencia de lo que se suele afirmar de manera común, no es en sí mismo una metodología: la propia Scrum Guide, escrita por sus creadores Ken Schwaber y Jeff Sutherland, lo define explícitamente como "un framework ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos" (Scrum Guides, s.f.). Esta definición es deliberada: Scrum establece un conjunto mínimo de roles, eventos, artefactos y reglas que los vinculan entre sí, pero deja en manos de cada equipo la elección de las técnicas y metodologías concretas con las que llenará ese marco general (Mountain Goat Software, 2026). Schwaber y Sutherland presentaron Scrum públicamente por primera vez en la conferencia OOPSLA de 1995 en Austin, Texas, inspirándose en investigaciones previas sobre desarrollo ágil de productos en empresas japonesas y estadounidenses.

En nuestro caso, no adoptamos la totalidad de los roles y ceremonias formales de Scrum (no existía, por ejemplo, la figura formal de un Scrum Master ni sprints cerrados de duración fija), sino que se incorporaron de forma complementaria a Kanban algunas de sus buenas prácticas, como reuniones periódicas de revisión de avances con el equipo. Estas reuniones cumplían una función similar a la de una revisión de sprint, permitiendo mostrar el trabajo completado, recibir retroalimentación y ajustar las

prioridades del equipo en función de los resultados obtenidos, sin necesidad de ceñirnos a la estructura completa y formal que Scrum prescribe.

Figura 5

Scrum del equipo de desarrollo



Nota. Se presentan los principales componentes de Scrum y la dinámica de la Daily Scrum durante el desarrollo de un Sprint.

3.3. Documentos, informes o manuales elaborados

La documentación, informes y manuales que elaboré durante mi desempeño variaron según las necesidades particulares de cada proyecto y etapa de mi carrera dentro de la empresa. En proyectos como Phronesys y Talma Click, entregué informes mensuales siguiendo un formato estándar establecido por la empresa, detallando las actividades realizadas día a día, los requerimientos atendidos y el estado de avance de cada uno, con el fin de mantener informados tanto al jefe de proyecto como al cliente sobre el progreso del trabajo.

A partir de las herramientas de pruebas automatizadas utilizadas en cada proyecto, extraía además reportes de validación que evidenciaban el comportamiento correcto de las funcionalidades desarrolladas, los cuales se presentaban al área de seguridad como parte del proceso de aprobación previo al despliegue a producción. De forma similar, en los proyectos desarrollados bajo arquitecturas de microservicios, elaboré reportes que detallaban las versiones de las librerías y dependencias utilizadas en cada servicio, permitiendo identificar dependencias desactualizadas o con vulnerabilidades conocidas y facilitando su seguimiento y actualización oportuna por parte del área de seguridad, contribuyendo así a reducir la superficie de exposición ante posibles ataques.

Asimismo, elaboré manuales técnicos detallados por proyecto, documentando aspectos como la configuración del entorno de desarrollo, la estructura del código y las convenciones adoptadas por el equipo, con el objetivo de estandarizar el trabajo y facilitar la incorporación de nuevos desarrolladores. Como parte de mis funciones de mayor responsabilidad al asumir el rol de Líder de Proyecto, contribuí también a la mejora de los lineamientos de seguridad aplicados durante el desarrollo de software en la empresa, estableciendo criterios como el control del tamaño y el contenido de los cambios enviados al repositorio, según se detalla en la sección de Innovaciones del presente informe.

3.4. Utilidad generada

Más allá del cumplimiento de las tareas asignadas, mi participación en cada proyecto generó mejoras que permanecieron vigentes una vez finalizada mi intervención directa en ellos. En Pacífico Seguros, la integración del servicio externo de datos vehiculares eliminó un proceso que anteriormente se realizaba de forma manual, mientras que la incorporación de Liquibase para el versionado de las migraciones de

base de datos evitó que los futuros cambios de esquema se realizarán sin un control adecuado, como ocurría previamente. En Talma Click, la corrección sostenida de incidencias y la incorporación de Mutmut no solo resolvieron problemas puntuales, sino que modificaron la forma en que el equipo evaluaba la confiabilidad de sus propias pruebas, exigiendo que estas demostraran efectivamente su capacidad de detectar errores, y no únicamente que se ejecutarán sin fallos.

En Phronesys, al desarrollar el backend desde su etapa inicial, las decisiones adoptadas respecto a la estructura del código y al manejo del procesamiento de datos históricos mediante Lambda y S3 sentaron las bases sobre las cuales se continuó desarrollando el sistema. De manera similar, la implementación de Keycloak permitió centralizar un proceso de autenticación que anteriormente cada sistema gestionaba de forma independiente, reduciendo tanto la duplicación de esfuerzos como el riesgo de una gestión inconsistente de usuarios y permisos entre distintas plataformas. En el visualizador de indicadores, si bien mi participación fue puntual, la corrección de los valores que no se cargaban correctamente resultó determinante para que el sistema pudiera ser utilizado con confianza por los usuarios finales.

Como Líder de Proyecto en Highlands, el impacto generado fue de mayor alcance: no se trató de resolver un problema específico, sino de definir la arquitectura sobre la cual se construyó todo el sistema. La selección de Spark, Kafka y Debezium para resolver la sincronización entre MongoDB y PostgreSQL constituyó una decisión estructural que condicionó el desarrollo posterior del proyecto. Asimismo, los lineamientos de seguridad que contribuí a mejorar, como los límites establecidos sobre el tamaño de los commits, quedaron incorporados como práctica permanente del equipo, más allá de su aplicación puntual.

3.5. Innovaciones

Una de las prácticas que introduje, poco habitual en proyectos similares, fue evaluar no solo si el código funcionaba correctamente, sino qué tan buenas eran realmente las pruebas que lo validaban. En Talma Click, donde surgían con frecuencia errores en producción a pesar de contar con pruebas unitarias, apliqué una técnica que introduce pequeños cambios deliberados en el código y verifica si las pruebas existentes son capaces de detectarlos. Esto permitió identificar pruebas débiles o insuficientes que, de otra forma, habrían pasado desapercibidas, elevando el estándar de calidad más allá de simplemente "que las pruebas pasen", y reduciendo de forma notoria la cantidad de nuevos errores que llegaban a producción.

De forma complementaria, integré la revisión de seguridad como parte del propio ciclo de desarrollo, en lugar de tratarla como una validación aislada al final del proyecto. Esto permitió detectar vulnerabilidades tanto en el código como en la aplicación en ejecución de forma temprana, evitando que estos riesgos llegaran a producción, particularmente en proyectos donde las dependencias se actualizaban con frecuencia y podían introducir fallos de seguridad sin que el equipo lo advirtiera de inmediato.

En Pacífico Seguros, introduje el uso de Liquibase para el versionado de las migraciones de base de datos, una práctica con la que el proyecto no contaba previamente y que permitió que los cambios de esquema quedarán documentados, ordenados y reproducibles, en lugar de aplicarse de forma manual y sin trazabilidad.

Finalmente, como Líder de Proyecto, establecí lineamientos de control sobre el propio proceso de desarrollo del equipo, más allá de las prácticas convencionales de seguridad. Esto incluyó reglas como límites en la cantidad de líneas modificadas por commit y en el peso de los archivos permitidos al hacer push al repositorio, buscando

mantener un historial de cambios más ordenado, trazable y fácil de revisar, algo que muchas empresas no llegan a formalizar dentro de sus propios equipos de desarrollo.

3.6. Premios

Durante mi desempeño en la empresa, recibí bonos económicos como reconocimiento por la culminación exitosa de determinados proyectos, otorgados en función del cumplimiento de los objetivos establecidos y de la calidad del trabajo entregado al cliente. Este tipo de reconocimiento se aplicaba de forma diferenciada según el impacto y la complejidad de cada proyecto, y buscaba valorar tanto el cumplimiento de plazos como la ausencia de incidencias posteriores a la entrega.

Adicionalmente, la empresa reconocía el esfuerzo sostenido del equipo mediante viajes otorgados al alcanzar metas y objetivos específicos, una práctica que trascendía el reconocimiento individual para poner en valor el trabajo colaborativo detrás de cada proyecto. Este tipo de iniciativas reflejaba una cultura organizacional orientada no solo a resultados, sino también a fortalecer la motivación y el sentido de pertenencia del equipo, aspectos que, en mi experiencia, incidían directamente en la calidad del trabajo entregado y en la disposición del equipo para asumir nuevos retos técnicos.

4. Conclusiones y recomendaciones

La experiencia profesional desarrollada en Well Done Solutions S.A.C. permitió consolidar y fortalecer las competencias adquiridas durante la formación universitaria, aplicándolas en un entorno laboral real y exigente. El desempeño como Desarrollador Backend, trabajando con tecnologías como Java y Python en distintos proyectos para clientes como Pacífico Seguros, Talma Click y Phronesys, permitió desarrollar sólidas habilidades técnicas en el desarrollo, mantenimiento y aseguramiento de la calidad de servicios y microservicios.

La incorporación de prácticas como pruebas de mutación con Mutmut, pruebas de seguridad SAST y DAST, y la implementación de Keycloak como solución de autenticación y control de acceso, reflejan un compromiso constante con la calidad y la seguridad del software desarrollado, superando en varios aspectos las prácticas convencionales de la industria.

Asimismo, la posterior asunción del rol de Líder de Proyecto representó un salto cualitativo importante en mi desarrollo profesional, al asumir mayores responsabilidades relacionadas con el diseño de soluciones, la coordinación de equipos, la gestión de tareas y la mejora de procesos internos de desarrollo de software, en coordinación directa con el arquitecto y el jefe de proyecto. Esta experiencia permitió desarrollar no solo competencias técnicas, sino también habilidades blandas fundamentales como el liderazgo, la comunicación efectiva y la toma de decisiones bajo un entorno de trabajo remoto.

En conjunto, la experiencia adquirida evidencia la aplicación práctica de conocimientos de ingeniería de sistemas en la resolución de problemas reales, contribuyendo tanto al desarrollo profesional personal como a la generación de valor tangible para la empresa y sus clientes.

A la empresa, se recomienda continuar fortaleciendo la incorporación de prácticas de aseguramiento de calidad y seguridad, como las pruebas de mutación y los análisis SAST/DAST, extendiéndolas de manera estandarizada a todos los proyectos de la organización, con el fin de mantener un nivel de calidad homogéneo. Asimismo, se recomienda seguir promoviendo espacios de crecimiento profesional interno, como la transición de roles técnicos a roles de liderazgo, dado el impacto positivo que esto genera tanto en el desarrollo del colaborador como en los resultados de los proyectos.

A la Universidad Peruana Unión y a la Escuela Profesional de Ingeniería de Sistemas, se recomienda reforzar en la malla curricular contenidos relacionados con arquitecturas de microservicios, buenas prácticas de seguridad en el desarrollo de software (DevSecOps) y metodologías ágiles de gestión de proyectos, ya que constituyen competencias altamente demandadas en el entorno laboral actual. De igual forma, se sugiere fomentar espacios de práctica orientados al desarrollo de habilidades blandas, como liderazgo y trabajo en equipos distribuidos, dado que estas resultan determinantes para el desempeño profesional en contextos de trabajo remoto, cada vez más comunes en el sector tecnológico.

5. Referencias

Amazon.com. (2012, 18 de enero). *Amazon Web Services launches Amazon DynamoDB* [Comunicado de prensa]. <https://press.aboutamazon.com/2012/1/amazon-web-services-launches-amazon-dynamodb-a-new-nosql-database-service-designed-for-the-scale-of-the-internet>

Amazon Science. (2025). *Amazon DynamoDB: A decade of internet-scale database innovation*. <https://www.amazon.science/>

Amazon Web Services. (2014, 13 de noviembre). *Amazon Web Services announces AWS Lambda* [Comunicado de prensa]. <https://press.aboutamazon.com/2014/11/amazon-web-services-announces-aws-lambda>

Asthana, A. (2021, 5 de abril). *How we built Postman, the product and the company*. Postman Blog. <https://blog.postman.com/how-we-built-postman-product-and-company/>

AWS News Blog. (2024, 10 de julio). *Amazon Cognito: A decade of secure customer identity*. <https://aws.amazon.com/blogs/aws/>

AWS News Blog. (2026, 13 de marzo). *Twenty years of Amazon S3 and building what's next*. <https://aws.amazon.com/blogs/aws/twenty-years-of-amazon-s3-and-building-whats-next/>

Build5Nines. (2026). *Terraform history and evolution*. <https://build5nines.com/>

Buytaert, D. (2025, 24 de noviembre). *The history of MySQL AB*. <https://dri.es/the-history-of-mysql-ab>

CNCF. (2023, 11 de abril). *Keycloak joins CNCF as an incubating project*. <https://www.cncf.io/blog/2023/04/11/keycloak-joins-cncf-as-an-incubating-project/>

Colling, M. (2024, 10 de octubre). *Mutation testing in Python using Mutmut*. Medium. <https://medium.com/@dead-pixel.club/mutation-testing-in-python-using-mutmut-a094ad486050>

Computer History Museum. (2021, 10 de febrero). *Guido van Rossum*. <https://computerhistory.org/profile/guido-van-rossum/>

DevOpsSchool. (2022, 4 de abril). *What is Jenkins and how does it work? An overview and its use cases*. <https://www.devopsschool.com/blog/what-is-jenkins-and-how-does-it-work-an-overview-and-its-use-cases/>

DevOpsSchool. (2026). *From OWASP ZAP to ZAP: The complete evolution, history and milestones of the Zed Attack Proxy*. <https://www.devopsschool.com/blog/from-owasp-zap-to-zap-the-complete-evolution-history-and-milestones-of-the-zed-attack-proxy/>

Django Software Foundation. (2025). *Django celebrates 20 years*.
<https://www.djangoproject.com/>

Docker. (2024, 16 de julio). *11 years of Docker: Shaping the next decade of development*. <https://www.docker.com/blog/docker-11-year-anniversary/>

Factor House. (2026, 30 de mayo). *How LinkedIn uses Apache Kafka in production*.
<https://factorhouse.io/articles/linkedin-kafka-architecture/>

Flexera. (2026). *Apache Spark history and origin*. <https://www.flexera.com/>

GitLab. (2023). *Company history*. GitLab Handbook.
<https://handbook.gitlab.com/handbook/company/history/>

Google Cloud. (2021). *Cloud SQL documentation*. <https://cloud.google.com/sql/docs>

Google Cloud. (2025, 23 de julio). *Celebrating 10 years of GKE*.
<https://cloud.google.com/blog/products/containers-kubernetes/10-years-of-gke-ebook/>

Google Cloud. (s.f.). *Identity and Access Management (IAM)*. Recuperado el 12 de agosto de 2026, de <https://cloud.google.com/products/iam>

GOTO Copenhagen. (2025). *Kent Beck: Speaker bio*. <https://gotocph.com/>

JetBrains. (2021). *20 years of IntelliJ IDEA*. <https://www.jetbrains.com/lp/intellijidea-20-anniversary/>

Kanban University. (2021, 1 de marzo). *The official guide to the Kanban method*.
<https://kanban.university/kanban-guide/>

Kubernetes Blog. (2024, 6 de junio). *Kubernetes: 10 years of open source innovation*.
<https://kubernetes.io/blog/>

LearnCodeWithDurgesh. (2025, 9 de diciembre). *History of Spring Boot: Evolution from Spring Framework*. <https://learncodewithdurgesh.com/tutorials/spring-boot-tutorials/spring-boot-history-and-version-history>

Liquibase. (2021, junio). *Liquibase celebrates 15 years of open source database change*. <https://15years.liquibase.com/>

MatrixBCG. (2026). *MongoDB: History and evolution of a NoSQL database*.
<https://matrixbcg.com/>

MDPI Encyclopedia. (2025, 21 de mayo). *May 23: Release of the Java programming language*. <https://encyclopedia.pub/entry/58353>

Microsoft. (2021). *VS Code Day: An event for an editor?*. Visual Studio Code Blog.
<https://code.visualstudio.com/blogs/2023/04/13/vscode-day>

Microsoft. (2024, 12 de julio). *Azure AD is being renamed to Microsoft Entra ID*. Microsoft Entra Identity Platform Blog. <https://devblogs.microsoft.com/identity/aad-rebrand/>

Morling, G. (2024, 4 de noviembre). *Thoughts on moving Debezium to the Commonhaus Foundation*. <https://www.morling.dev/blog/thoughts-on-moving-debezium-to-commonhaus-foundation/>

Mountain Goat Software. (2026). *The Scrum framework: Roles, meetings, artifacts, and how Scrum works*. <https://www.mountaingoatsoftware.com/agile/scrum>

Nave. (2023, 15 de marzo). *From supermarkets to software: History of Kanban*. <https://getnave.com/blog/kanban-history/>

PostgreSQL Global Development Group. (2026). *A brief history of PostgreSQL*. PostgreSQL Documentation. <https://www.postgresql.org/docs/current/history.html>

ProjectManager. (2025, 28 de abril). *Kanban history: Origin & expansion across industries*. <https://www.projectmanager.com/blog/kanban-history>

pytest. (s.f.). *History*. pytest documentation. Recuperado el 12 de agosto de 2026, de <https://docs.pytest.org/en/stable/history.html>

Redis. (2026, 21 de mayo). *Redis then & now: Adapting with developers through every era*. <https://redis.io/blog/redis-then-and-now-adapting-with-developers-through-every-era/>

Scrum Guides. (s.f.). *Home*. Recuperado el 12 de agosto de 2026, de <https://scrumguides.org/>

So, N. (2024, 23 de abril). *Keeping an open-source mind*. Sequoia Capital. <https://sequoiacap.com/article/sebastian-ramirez-spotlight/>

Sonar. (2025, 13 de noviembre). *Seventeen years later, code quality is more relevant than ever*. <https://www.sonarsource.com/blog/sonars-17-year-anniversary/>

Stacksync. (2026a, 23 de marzo). *The \$300,000 he turned down from Microsoft: The origin story of GitHub*. <https://www.stacksync.com/blog/the-300-000-he-turned-down-from-microsoft-the-origin-story-of-github>

Stacksync. (2026b). *Google BigQuery: From Dremel to serverless analytics*. <https://www.stacksync.com/>

VentureBeat. (2021). *ArangoDB: Multi-model database origins*. <https://venturebeat.com/>

GitHub. (s.f.). *About Projects*. GitHub Docs. Recuperado el 12 de agosto de 2026, de <https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>

6. Anexos

Anexo A: Copia de certificado de los tres años de experiencia laboral



<https://www.wds.com.pe>

Jr 1ero de Mayo 112

Certificado de Trabajo

El que suscribe:

Ing. Roel Dante Gómez Apaza, Gerente General de la Empresa **Well Done Solutions SAC**. Identificado con **RUC N° 20603175876**, domicilio fiscal en el Pje. 1ero de mayo 112 oficina 401 de la Ciudad de Juliaca departamento de Puno.

CERTIFICA:

Por medio del presente, se certifica que el señor **Soncco Ramos, Michael**, identificado con DNI N.º **60632940**, ha laborado en nuestra empresa en el cargo de **Desarrollador Back End**, en el área de **Desarrollo de Software**, desde el **02 de enero del 2023** hasta el **13 de abril del 2026**.

Durante el periodo de su vínculo laboral, el señor Soncco Ramos participó activamente en proyectos internos y externos de la empresa. Entre los principales proyectos en los que fue asignado destacan sus labores como **Desarrollador Backend** en los proyectos **Talma** y **Phronesys**, brindando soporte técnico y contribuyendo al desarrollo de soluciones eficientes y escalables.

Se expide el presente certificado a solicitud del interesado, para los fines que estime conveniente.

Juliaca, 14 de abril 2026.


GERENTE GENERAL WDS SAC
Gómez Apaza, Roel Dante
DNI 40071297



9 cursos

- Foundations: Data, Data, Everywhere
- Ask Questions to Make Data-Driven Decisions
- Prepare Data for Exploration
- Process Data from Dirty to Clean
- Analyze Data to Answer Questions
- Share Data Through the Art of Visualization
- Introduction to Data Analysis Using Python
- Google Data Analytics Capstone: Complete a Case Study
- Accelerate Your Job Search with AI



Jun 15, 2026

Michael Soncco Ramos

ha completado correctamente el Certificado profesional en línea

Google Data Analytics

Those who earn the Google Data Analytics Professional Certificate have completed nine courses, developed by Google, that include hands-on, practice-based assessments and are designed to prepare them for introductory-level roles in Data Analytics. They are competent in tools and platforms including spreadsheets, SQL, Tableau, and Python. They know how to prepare, process, analyze, and share data for thoughtful action.



Amanda Broghy
Global Director of
Google Career
Certificates

Este certificado acreditativo al ser el alumno competente en programas en línea impartidos a través de Coursera. No constituye una inscripción formal en ninguna universidad o entidad, ni otorga por sí misma credenciales académicas, titulaciones o un título. Los acreditados en su organización pueden, a su discreción, reconocer este logro obteniendo un título, premio, programa o credenciales.

Verifica este certificado en:
<https://coursera.org/verify/professional/certificate/5YYZ0FD181E>

Anexo C: Copia de la dictaminación favorable del informe



"Año de la Esperanza y el Fortalecimiento de la Democracia"

RESOLUCIÓN N° 1030-2026/UPeU-FIA-CF

Lima, Ñaña, 04 de agosto de 2026

VISTO:

El expediente del (de la) bachiller **Soncco Ramos Michael** identificado(a) con código universitario N° **201520621**, de la Escuela Profesional de Ingeniería de Sistemas de la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión;

CONSIDERANDO:

Que la Universidad Peruana Unión tiene autonomía académica, administrativa y normativa, dentro del ámbito establecido por la Ley Universitaria N° 30220 y el Estatuto de la Universidad;

Que la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión, mediante sus reglamentos académicos y administrativos, ha establecido las formas y procedimientos para la declaratoria de expedito para la sustentación Informe de Trabajo de Suficiencia Profesional;

Que el Comité Dictaminador ha emitido su dictamen aprobando el Informe de Trabajo de Suficiencia Profesional titulado "Informe de desempeño profesional desarrollada en WELL DONE SOLUTIONS (2023-2026)", presentado por el (la) bachiller **Soncco Ramos Michael**, reuniendo de esta manera las condiciones previas para la declaratoria de expedito para la programación de la sustentación;

Estando a lo acordado en la sesión del Consejo de la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión, celebrada el 04 de agosto de 2026, y en aplicación del Estatuto y el Reglamento General de investigación de la Universidad;

SE RESUELVE:

Artículo 1. – DECLARAR EXPEDITO al (a la) bachiller **Soncco Ramos Michael**, para que sustente el Informe de Trabajo de Suficiencia Profesional "Informe de desempeño profesional desarrollada en WELL DONE SOLUTIONS (2023-2026)", conducente a la obtención del título profesional de Ingeniero de Sistemas, el 16 de agosto, a las 10:10 horas, en la modalidad presencial, en el Auditorio Pedro Kalbermatter.

Artículo 2. – DESIGNAR EL JURADO DE SUSTENTACIÓN, encargado de gestionar la sustentación respectiva, el mismo que queda constituido por los siguientes miembros:

Presidente: Mg. Nancy Esther Casildo Bedón
Secretario: Mg. David Mamani Pari
Asesor: Mg. Roel Dante Gómez Apaza
Vocal 1: Mg. Néida Huamán Paco
Vocal 2: Mg. Moisés Condori Chañi

Artículo 3. – DISPONER que la presente Resolución sea registrada, comunicada a las instancias correspondientes y archivada conforme a ley.

Regístrese, comuníquese y archívese.



Dra. Erika Inés Acuña Salinas
DECANA

Cel:
Andrés
Jorge
García
García

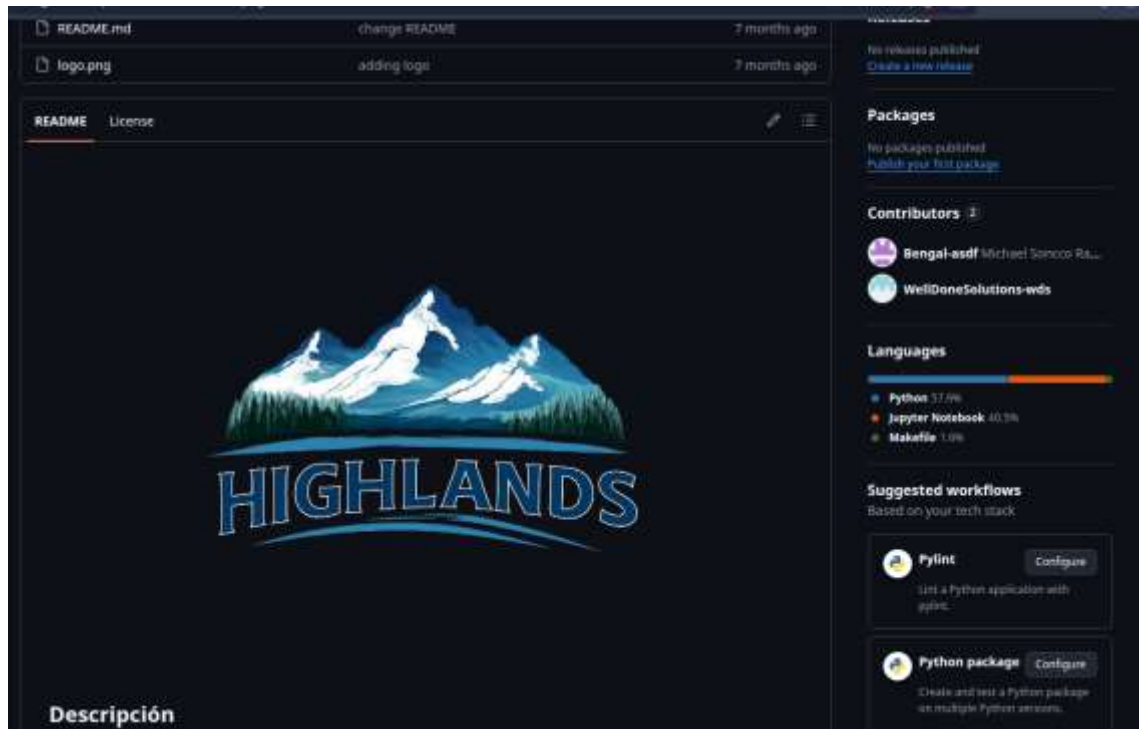


Mg. Sc. Gina Marita Tito Tolentino
SECRETARIA ACADÉMICA

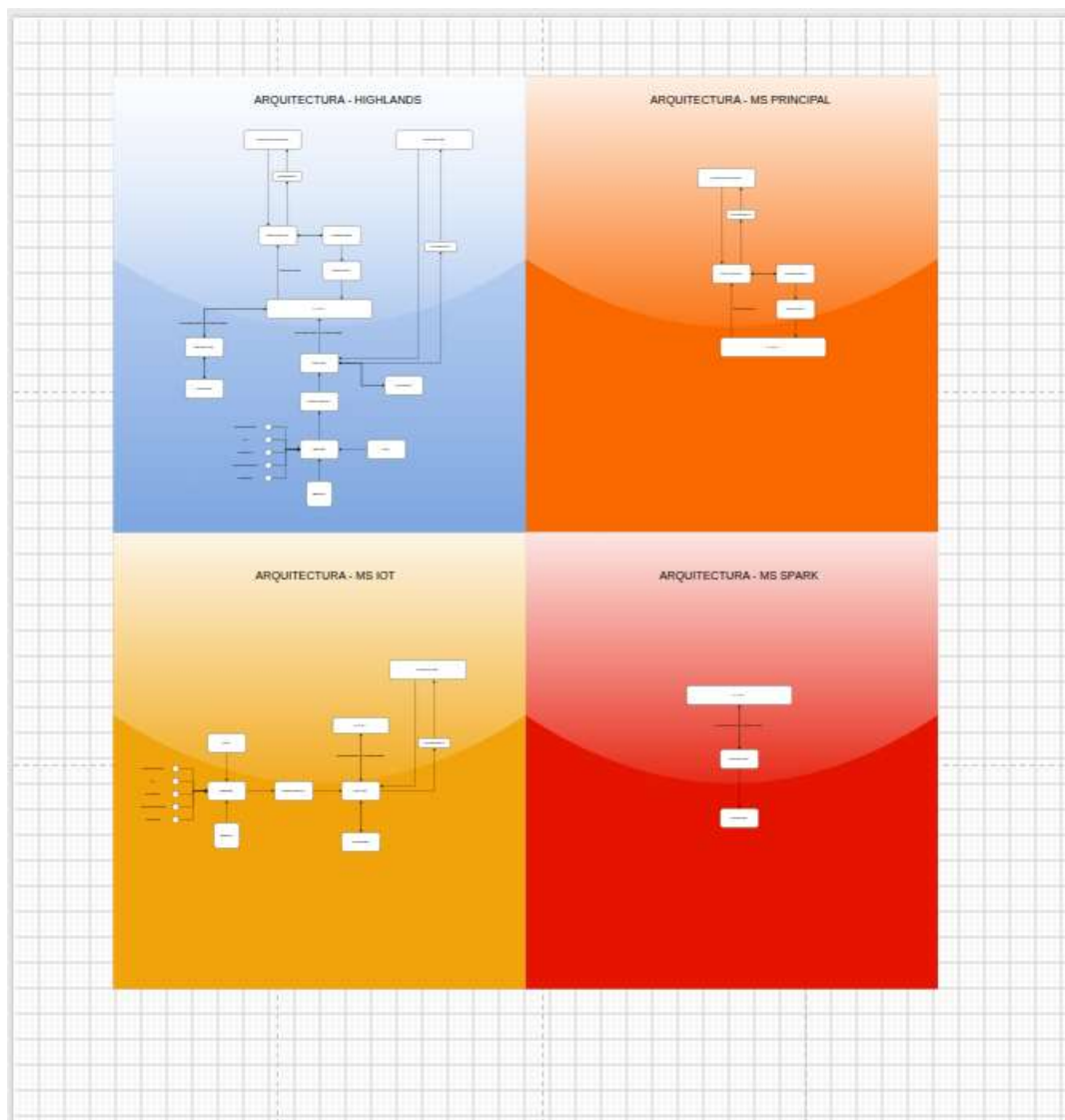
Anexo D: Copia del Título Profesional de Bachiller



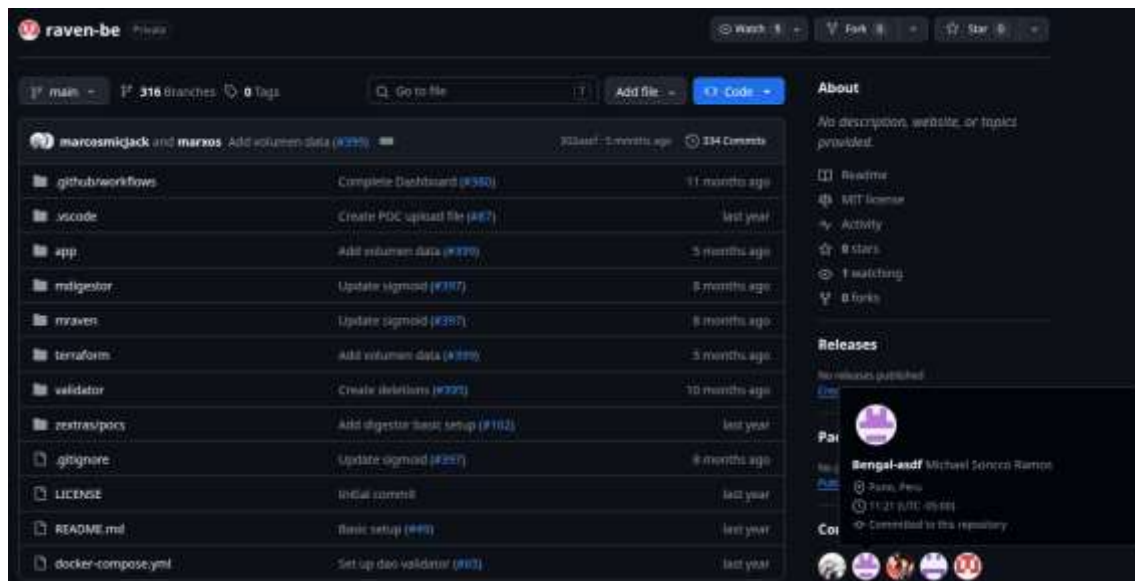
Anexo E: Repositorio "Highlands" en GitHub - Well Done Solutions



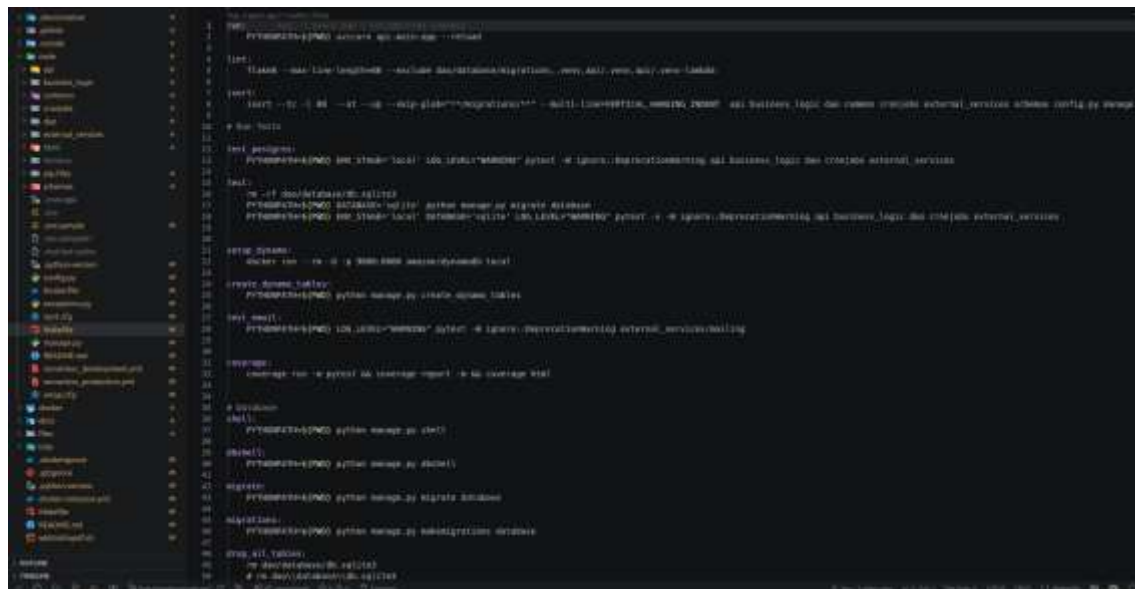
Anexo F: Arquitectura de microservicios del proyecto "Highlands"



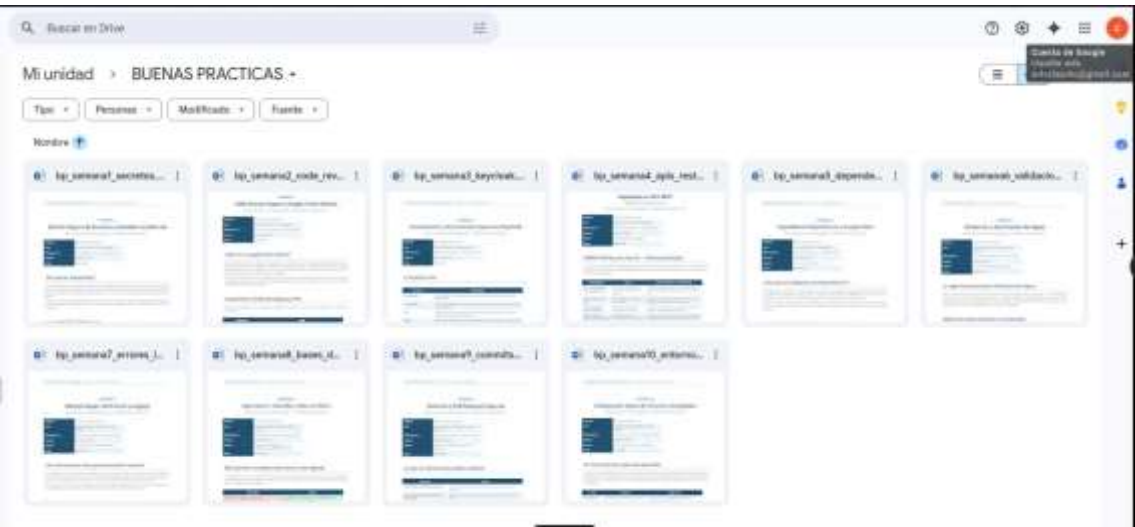
Anexo G: Repositorio "raven-be" - Proyecto Phronesys



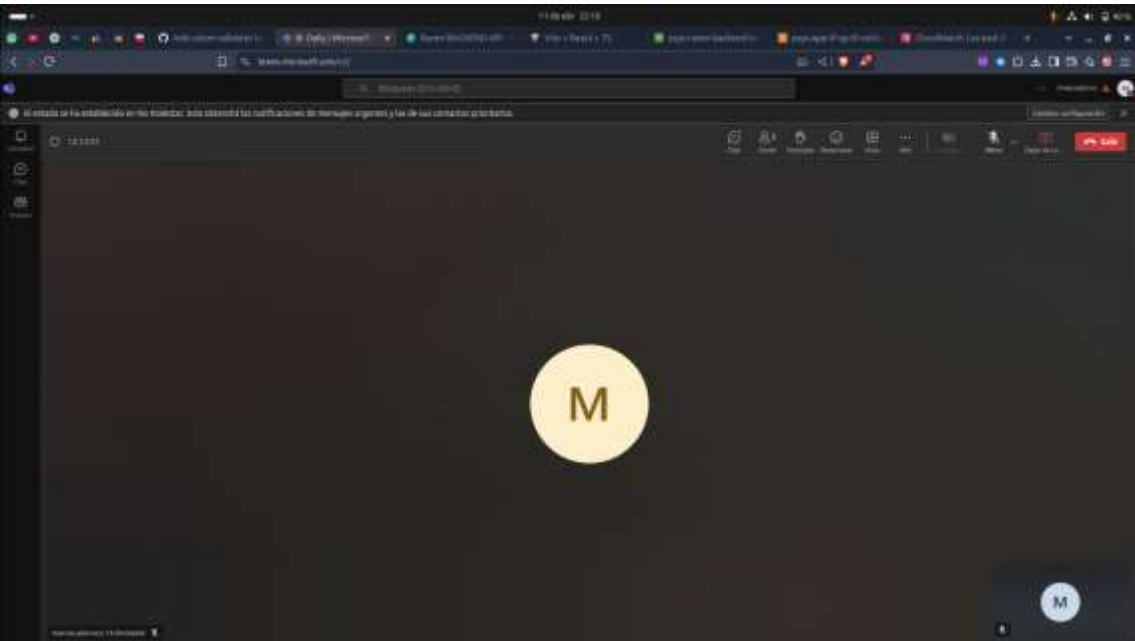
Anexo H: Estructura de proyecto backend y Makefile - Talma Click



Anexo I: Documentos de Buenas Prácticas de Seguridad elaborados como Líder de Proyecto



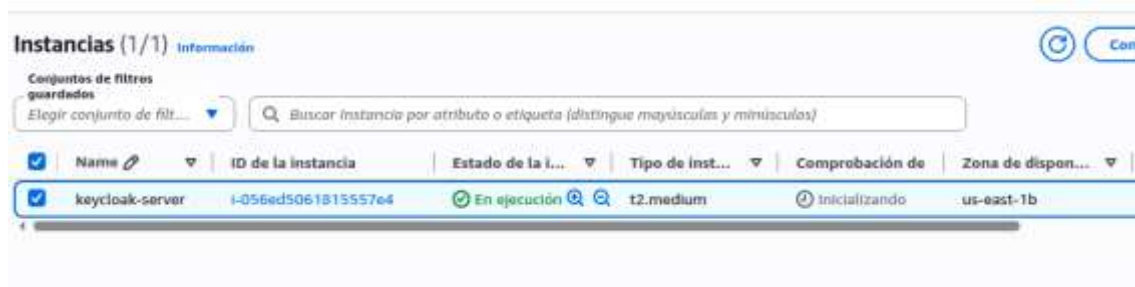
Anexo J: Reunión de coordinación de proyecto - Phronesys



[illegible]

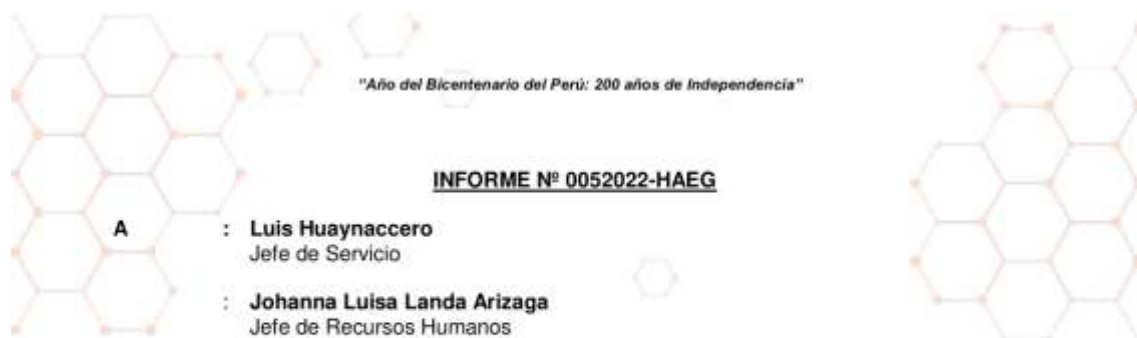
```
1 package pe.com.pacifico.kuntur;
2
3 import com.pacifico.kuntur.core.launcher.ApplicationLauncher;
4 import com.pacifico.kuntur.core.starter.web.runner.StarterWebApplication;
5 import org.springframework.boot.SpringApplication;
6 import org.springframework.boot.autoconfigure.SpringBootApplication;
7
8 /**
9  * <b>Class</b>: LaunchApplication <br/>
10  * <b>Copyright</b>: 2023 Pacifico Seguros - La Chakra <br/>.
11  *
12  * <b>Author</b> 2023 Pacifico Seguros - La Chakra <br/>
13  * <b>Service Provider</b>: Soluciones Digitales <br/>
14  * <b>Developed by</b>: La Chakra developer <br/>
15  * <b>Changes</b>: <br/>
16  * <ul>
17  * <li>
18  * January 9, 2023 Creación de Clase.
19  * </li>
20  * </ul>
21  */
22 @SpringBootApplication
23 @ApplicationLauncher
24 public class LaunchApplication extends StarterWebApplication {
25
26     public static void main(String[] args) {
27         SpringApplication.run(LaunchApplication.class, args);
28     }
29 }
30
31
```

Anexo M: Instancia de Keycloak desplegada en AWS EC2



Nombre	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de integridad	Zona de disponibilidad
keycloak-server	i-056ed5061815557e4	En ejecución	t2.medium	Inicializando	us-east-1b

Anexo N: Informe mensual de actividades - Proyecto Talma Click



A : **Luis Huaynaccero**
Jefe de Servicio

: **Johanna Luisa Landa Arizaga**
Jefe de Recursos Humanos

De : **Michael Soncco Ramos**
Contratado(a) por Locación de Servicios

Asunto : Entregable – N°00385, de fecha 6 de enero 2023.

Fecha : 21 de febrero de 2023.

Me es grato dirigirme a usted, para saludarlo muy cordialmente y, en atención al asunto que se indica, informarle lo siguiente:

I. ANTECEDENTES:

- 1.1. Mediante el Contrato N°00385, notificada a mi persona el 6 de enero del 2023, se efectuó la contratación de mis servicios para realizar **labores de Gestión del Proyecto de WFH y Certificación ISO-27001 en la modalidad de CONSULTORIA** para el cliente **BRACING INTERMEDIA** la fecha de mi contrato vence el 31 de marzo del 2023.

II. DESCRIPCIÓN DEL SERVICIO PRESTADO

- 2.1. Durante el servicio brindado en el mes de febrero 2022, dentro del plazo pactado para el entregable, se han elaborado las siguientes actividades:
 - Pruebas unitarias, integrales de las funcionales del requerimiento RQ-F2.

Anexo Ñ: Informe mensual de actividades - Proyecto Phronesys

INFORME MENSUAL DE ACTIVIDADES

Fecha: 23/05/2025

Título del Proyecto:	Desarrollo de una plataforma utilizando modelos bioestadísticos de dinámicas poblacionales que permitirá mejorar la estimación de calibre y volumen de la palta Hass, impactando en la reducción de emisiones asociadas.
Contrato / Convenio:	304-PROINNOVATE-PIEC1CC-2024
Período del Hito del Proyecto:	Del 26/06/2024 al 25/06/2025
Entidad Ejecutora:	PHRONESYS
Coordinador General del Proyecto:	Christian Giancarlo Gonzales

1. Datos Generales del Personal
Sección que deberá ser llenada por el Personal

Nombres y Apellidos:	Michael Soncco Ramos		
Recursos Humano Adicional	X	Entidad Asociada (colaboradora)	
Rol en el proyecto:	Desarrollador Backend		
Mes de las actividades:	Abril		

1.1 Actividades realizadas y resultados alcanzados en el mes que se informa.		
Actividad detallada	Fecha	Lugar
Se actualizó el método get_plots en el módulo validator, adecuándolo a la nueva estructura definida. Asimismo, se realizaron las modificaciones correspondientes en la capa DAO para garantizar la correcta obtención de los datos (ver anexo A).	28-04-2025	Puno
Se implementó un nuevo endpoint		

Anexo O: Herramienta SAST y DAST

```
ast-ds v0.2.0 - Application Security Testing

Detectando endpoints... 9 encontrados

POST /api/trout/Sampling/measurement 1 vulnerabilidad 100% 0:00:00
POST /api/trout/farm/ 1 vulnerabilidad 100% 0:00:00
GET /api/trout/farm/ 0 vulnerabilidades 100% 0:00:00
GET /api/trout/farm/summary/ 0 vulnerabilidades 100% 0:00:00
GET /api/trout/fundo/ 0 vulnerabilidades 100% 0:00:00
POST /api/trout/fundo/cohort/fundo/ 1 vulnerabilidad 100% 0:00:00
GET /api/trout/fundo/summary 0 vulnerabilidades 100% 0:00:00
GET /api/trout/sector/ 0 vulnerabilidades 100% 0:00:00
GET /api/trout/sector/summary 0 vulnerabilidades 100% 0:00:00

Endpoints analizados: 9 | Vulnerabilidades detectadas: 3 | Confirmadas: 3

POST /api/trout/Sampling/measurement
[ALTO] Ausencia de control de autenticación - CONFIRMADO
OWASP: API2-2023 - Broken Authentication

POST /api/trout/farm/
[ALTO] Ausencia de control de autenticación - CONFIRMADO
OWASP: API2-2023 - Broken Authentication

POST /api/trout/fundo/cohort/fundo/
[ALTO] Ausencia de control de autenticación - CONFIRMADO
OWASP: API2-2023 - Broken Authentication
```