



Gestión del conocimiento en proyectos tecnológicos ágiles en Sistemas GyG S.A.

Jorge Alberto Márquez Camargo

Henry Alfonso Guío Ávila

**Tunja
30/04/2026**

CONTENIDO

1. FICHA TÉCNICA DE LA PASANTÍA.....	4
2. ANTECEDENTES DE LA EMPRESA.....	5
3. PLANTEAMIENTO DEL PROBLEMA.....	6
4. JUSTIFICACIÓN	9
5. OBJETIVOS	11
5.1. OBJETIVO GENERAL	11
5.2. OBJETIVOS ESPECÍFICOS	11
6. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.....	12
6.1. LISTADO DE ACTIVIDADES REALIZADAS EN EL PROYECTO.....	12
6.2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO	14
6.3. DESARROLLO DE ACTIVIDADES POR OBJETIVO ESPECÍFICO	15
6.3.1. INDUCCIÓN INSTITUCIONAL Y CAPACITACIÓN TÉCNICA.....	16
6.3.2. PROTOTIPO 1: PROCESADOR AUTOMATIZADO DE ARCHIVOS EXCEL CON PROGRAMACIÓN CONCURRENT.	24
6.3.3. PROTOTIPO 2: API DE ENVÍO AUTOMATIZADO DE CORREOS ELECTRÓNICOS.	29
6.3.4. ASEGURAMIENTO DE CALIDAD: DISEÑO Y EJECUCIÓN DE PRUEBAS FUNCIONALES	39
6.3.5. AJUSTES A PROYECTO ANGULAR/NODE.JS	42
6.3.6. DOCUMENTACIÓN TÉCNICA DE LOS PROTOTIPOS.....	43
6.3.7. MANUAL DE USUARIO PROCESADOR AUTOMATIZADO DE ARCHIVOS EXCEL.	44
6.3.8. CONFIGURACIÓN INICIAL.....	44
6.3.9. PREPARAR EL ARCHIVO EXCEL DE ENTRADA.....	44
6.3.10. EJECUTAR EL PROGRAMA	45
6.3.11. INTERPRETAR LOS RESULTADOS	45
6.3.12. SOLUCIÓN DE PROBLEMAS FRECUENTES	46
7. CONCLUSIONES	47

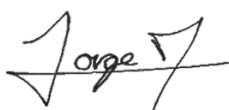
<u>8.</u>	<u>TABLA DE ILUSTRACIONES.....</u>	<u>48</u>
<u>9.</u>	<u>FIGURAS.....</u>	<u>48</u>
<u>10.</u>	<u>TABLAS</u>	<u>50</u>
<u>11.</u>	<u>REFERENCIAS</u>	<u>51</u>
<u>12.</u>	<u>ANEXOS.....</u>	<u>52</u>

1. FICHA TÉCNICA DE LA PASANTÍA

Título	Gestión del conocimiento en proyectos tecnológicos ágiles en Sistemas GyG S.A.
Nombre Estudiante	Jorge Alberto Márquez Camargo
Correo electrónico estudiante	jorge.marquez@usantoto.edu.co
Director	Henry Alfonso Guío Ávila
Tutor Empresarial	David Ricardo Munévar
Entidad o sector donde realizará la pasantía	Sistemas GyG S.A.
Ciudad donde se ubica la empresa	Motavita, Boyacá
Lugar de ejecución de la pasantía	Sede Principal - Fábrica de Desarrollo Carrera 4 Calle 1 Centro Motavita, Boyacá, Colombia
Modalidad de la pasantía	Mixta (presencial y teletrabajo)
Duración	4 meses
Línea de investigación	☒ Ingeniería del software ☐ Inteligencia Artificial ☐ Transformación Digital
Palabras clave	Aseguramiento de calidad, automatización de procesos, integración de sistemas, metodologías ágiles, prototipado

Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente (incluir nombres y firmas de estudiantes y director).

Firma del autor



Nombre autor

Jorge Alberto Márquez Camargo

Firma del director

Nombre director pasantía

Henry Alfonso Guío Ávila

2. ANTECEDENTES DE LA EMPRESA

Sistemas GyG S.A. es una empresa colombiana con más de 40 años de experiencia especializada en el desarrollo de software para el sector financiero, con sede principal en Motavita, Boyacá. La organización se ha consolidado como pionera en soluciones financieras, siendo el Core Bancario número 1 en Colombia, atendiendo a entidades que prestan servicios de crédito, depósito de dinero, captaciones e inversiones.

Su producto principal es el **Sistema Integrado de Información Financiera (SIIF)**, un Core Bancario modular en la nube que permite optimizar procesos operativos y administrativos de productos financieros, desde la producción hasta la cobranza. La plataforma cuenta con integraciones con centrales de riesgo, biometría, Deceval, digitalización y algoritmos de inteligencia artificial para análisis de score crediticio, procesando variables demográficas y financieras para reducir la probabilidad de impago.

La empresa se estructura en diferentes gerencias especializadas, entre ellas la Gerencia de Innovación, cuya misión es liderar proyectos de investigación aplicada, desarrollo de prototipos, validación de tecnologías emergentes y mejora continua de los productos existentes. Esta área trabaja bajo metodologías ágiles y estándares del Sistema de Gestión Integrado (SGI, SGSI, SG-SST, SGC), garantizando calidad, seguridad y cumplimiento normativo en cada iniciativa.

En el contexto de transformación digital del sector financiero colombiano, (Sistemas GyG S.A., 2026) ha establecido convenios estratégicos con instituciones académicas para fortalecer sus capacidades de innovación mediante la incorporación de talento en formación, facilitando la transferencia de conocimiento universidad-empresa y el desarrollo de proyectos que impulsen la competitividad organizacional.

3. PLANTEAMIENTO DEL PROBLEMA

Sistemas GyG S.A. es líder en software financiero en Colombia, pero su Gerencia de Innovación enfrenta un desafío importante: no tienen una forma estructurada y ágil de validar si las nuevas ideas tecnológicas realmente funcionarán antes de invertir recursos significativos en ellas. Esto está afectando su capacidad para responder rápidamente a las necesidades del mercado y mantener su ventaja competitiva en un sector que cambia cada vez más rápido.

Análisis Causa-Efecto

Primera causa: No hay prototipos rápidos para probar ideas

Cuando el equipo de innovación tiene una idea para mejorar el sistema SIIF o integrar una nueva tecnología, no existe un proceso claro para crear una versión pequeña y rápida que permita ver si funciona o no. Simplemente se debate teóricamente si pudiese servir, y muchas veces se decide avanzar sin tener certeza real de que la solución será viable. Esto genera que inviertan semanas o meses desarrollando algo que al final no cumple con lo esperado, o peor aún, que descarten ideas buenas por falta de evidencia técnica que las respalde. El resultado es tiempo perdido, recursos desperdiciados y oportunidades de mercado que se escapan mientras la competencia avanza más rápido.

Segunda causa: Las pruebas de los desarrollos no están organizadas

Cuando alguien del equipo crea un nuevo componente o funcionalidad, generalmente esa misma persona lo prueba de manera informal, sin documentar qué exactamente probó, cómo lo hizo ni qué resultados obtuvo. No hay casos de prueba escritos ni una lista clara de qué debe funcionar correctamente antes de considerar algo terminado. Esto hace que muchos errores solo se descubran cuando el sistema ya está en producción y los clientes empiezan a usarlo, generando reclamos, pérdida de confianza y la necesidad de hacer arreglos de emergencia que interrumpen otros proyectos. Además, cuando otra persona necesita modificar ese código más adelante, no tiene forma de saber si sus cambios rompieron algo que antes funcionaba.

Tercera causa: El conocimiento técnico no se documenta ni se comparte

Cada vez que alguien investiga una tecnología nueva, resuelve un problema complejo o aprende algo importante durante un proyecto, ese conocimiento queda en su cabeza o en notas personales que nadie más puede consultar. No existe una biblioteca de conocimiento técnico donde se guarden las lecciones aprendidas, las configuraciones que funcionaron bien, los errores que se cometieron y cómo se solucionaron, o las razones por las que se tomaron ciertas decisiones técnicas. Cuando esa persona se va de vacaciones, cambia de área o renuncia, todo ese aprendizaje se pierde y el siguiente que trabaje en algo similar tiene que empezar desde cero, cometiendo los mismos errores que ya se habían solucionado antes. Esto hace que el equipo no crezca en capacidades técnicas como organización, sino que dependa demasiado de personas específicas.

Cuarta causa: Falta tiempo para explorar qué hay de nuevo en tecnología

El día a día del equipo está dominado por tareas operativas urgentes: arreglar bugs, atender solicitudes de clientes, hacer ajustes al sistema existente. Esto deja muy poco tiempo para investigar qué tecnologías nuevas están surgiendo en el sector financiero mundial, qué están haciendo los competidores, qué herramientas podrían hacer el trabajo más eficiente, o qué tendencias regulatorias vienen en camino y requerirán adaptaciones del sistema. Sin esta mirada hacia adelante, la empresa corre el riesgo de quedarse atrás tecnológicamente, descubrir tarde que sus competidores ya adoptaron soluciones mejores, o verse obligada a hacer cambios apurados cuando una nueva regulación entra en vigor sin haberla anticipado.

Magnitud del Problema

Todo esto se traduce en consecuencias concretas para la empresa: los proyectos de innovación toman más tiempo del necesario porque no hubo validación temprana, las funcionalidades nuevas requieren múltiples rondas de correcciones porque no se probaron adecuadamente, se invierte dinero en explorar tecnologías que al final no sirven, y hay oportunidades de negocio que se pierden porque otros actores del mercado son más rápidos en innovar. En un sector como el financiero, donde la confianza y la estabilidad son fundamentales, estos problemas no solo afectan la eficiencia interna sino también la reputación de la empresa frente a sus clientes.

Oportunidad de Mejora

Aquí es donde toma valor el desarrollo de esta pasantía. Durante cuatro meses se propone trabajar en resolver estos cuatro problemas: creación de prototipos rápidos que permitan validar ideas antes de invertir en desarrollos completos, establecer procesos claros de testing con casos de prueba documentados, estructurar un sistema de gestión del conocimiento donde se guarde todo lo que se aprenda en cada proyecto, e investigar sobre nuevas tecnologías aplicables al negocio. Todo esto siguiendo los estándares de calidad que la empresa ya tiene establecidos en su Sistema de Gestión Integrado. El objetivo es que, al terminar la pasantía, la Gerencia de Innovación cuente con herramientas y procesos concretos para innovar de manera más rápida, segura y efectiva, manteniendo el liderazgo de Sistemas GyG en el mercado colombiano.

4. JUSTIFICACIÓN

Esta pasantía en Sistemas GyG S.A. busca resolver un problema real que enfrenta la empresa: cuando quieren probar nuevas tecnologías o funcionalidades para su sistema financiero SIIF, no tienen una forma clara y rápida de validar si esas ideas realmente funcionarán antes de invertir tiempo y dinero en desarrollarlas completamente. Esto significa que a veces avanzan con proyectos que luego no cumplen las expectativas, generando pérdida de recursos y retrasos. Mi trabajo consistirá en crear prototipos y pruebas rápidas que permitan saber desde el principio si una tecnología vale la pena o no, además de establecer procesos claros de testing para asegurar que lo que se desarrolle funcione bien desde el inicio. También documentaré todo lo que se aprenda en cada proyecto para que ese conocimiento no se pierda cuando las personas cambien de rol o salgan de la empresa, sino que quede disponible para todos.

El impacto de este trabajo será tangible tanto para la empresa como para el sector. Sistemas GyG podrá contar con un análisis más claro sobre qué tecnologías evaluar o adoptar en el desarrollo de nuevas funcionalidades, lo que contribuirá a una toma de decisiones más informada y alineada con las necesidades del producto principal. Esto es importante porque en el sector financiero la competencia es cada vez mayor, especialmente con la llegada de startups tecnológicas que se adaptan rápidamente a los cambios del mercado. Al documentar procesos y resultados de manera organizada, la empresa también podrá construir una base de conocimiento que facilite la consulta por parte de los miembros del equipo, permitiendo aprender de experiencias previas y mejorar continuamente las prácticas de desarrollo. Para Boyacá y Colombia, este tipo de iniciativas contribuye a fortalecer empresas nacionales que compiten en el sector tecnológico, promoviendo el desarrollo de talento y la innovación en la región.

Esta oportunidad me motiva porque voy a trabajar en proyectos reales que afectan a miles de usuarios del sistema financiero colombiano, experimentando con tecnologías como inteligencia artificial, automatización y desarrollo en la nube, pero aplicándolas a problemas concretos de negocio. No se trata solo de aprender teoría, sino de ver cómo se toman decisiones técnicas en una empresa consolidada, cómo se manejan proyectos bajo estándares de calidad exigentes y cómo se innova de manera responsable en un sector tan regulado como el financiero. Reconozco que habrá limitaciones: cuatro meses no son suficientes para proyectos muy complejos, algunos datos serán confidenciales por tratarse

de información financiera, y dependeré de la disponibilidad de herramientas y ambientes de prueba. Sin embargo, esas son las condiciones normales del mundo laboral real, y enfrentarlas será parte del aprendizaje. En resumen, esta pasantía ayudará a que Sistemas GyG innove de forma más inteligente, rápida y documentada, manteniendo su posición de liderazgo en el mercado colombiano.

5. OBJETIVOS

5.1. Objetivo General

Contribuir a la implementación de proyectos de innovación tecnológica en Sistemas GyG S.A., mediante la construcción de dos prototipos funcionales, la aplicación de procesos de aseguramiento de calidad y la elaboración de documentación técnica, apoyados en metodologías ágiles, herramientas de desarrollo en Go y estándares del Sistema de Gestión Integrado, con el fin de validar soluciones tecnológicas y mejorar los procesos internos del área de trabajo.

5.2. Objetivos específicos

Tabla 1. Objetivos específicos

Nro.	Objetivos específicos
1	Desarrollar dos prototipos funcionales y sus respectivas pruebas de concepto para proyectos de innovación tecnológica, mediante la aplicación de metodologías ágiles y el uso del lenguaje de programación Go, con el fin de validar su viabilidad técnica a través de pruebas controladas y evidencias funcionales documentadas.
2	Aplicar procesos de aseguramiento de calidad a los desarrollos tecnológicos propuestos durante la pasantía, mediante el diseño y ejecución de casos de prueba, la validación funcional y la identificación de riesgos técnicos, con el fin de verificar su correcto funcionamiento, estabilidad y cumplimiento de los requerimientos antes de su despliegue.
3	Elaborar la documentación técnica de los prototipos desarrollados durante la pasantía, incluyendo análisis, diseño, pruebas y resultados, mediante el uso de estándares del Sistema de Gestión Integrado (SGI) y herramientas colaborativas institucionales, con el fin de garantizar la trazabilidad de los procesos y facilitar la transferencia de conocimiento dentro del equipo de trabajo.

6. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

En esta sección se presenta el detalle de las actividades ejecutadas durante los cuatro meses de pasantía en Sistemas GyG S.A., desde el 19 de agosto hasta el 17 de diciembre de 2025. Las actividades se organizan en torno a los tres objetivos específicos planteados en la propuesta aprobada y se complementan con actividades de inducción, formación técnica y trabajo colaborativo.

6.1. Listado de actividades realizadas en el proyecto

A continuación, se presenta el consolidado de las actividades desarrolladas durante la pasantía, con sus fechas de inicio y entrega, y el producto concreto que se obtuvo de cada una. Las actividades se ordenan cronológicamente e incluyen tanto las proyectadas en la propuesta de pasantía como las complementarias que surgieron durante la ejecución del proyecto. En la columna de producto entregado se indica la subsección del documento donde se encuentra la descripción detallada y las evidencias correspondientes.

Tabla 2. Listado de actividades realizadas en el proyecto.

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Capacitación en el lenguaje de programación Go mediante curso en Udemy (secciones 1 a 22): fundamentos, variables, control de flujo, estructuras de datos, manejo de errores, POO, concurrencia, genéricos, API RESTful con Gin, persistencia con MySQL y GORM, y pruebas unitarias.	01/09/2025	01/10/2025	Certificado de finalización del curso de Go (Udemy, 2025) y cursos de la empresa. Inducción institucional y capacitación técnica.
Desarrollo del Prototipo 1: Procesador automatizado de archivos Excel con programación concurrente en Go. Incluye validación de datos, llenado de resultados mediante mapas y ejecución paralela con goroutines (máximo 10 hilos simultáneos). Publicación del proyecto en GitLab.	15/09/2025	03/10/2025	Prototipo funcional del procesador Excel con goroutines. Prototipo 1: Procesador automatizado de archivos Excel con programación concurrente.
Desarrollo del Prototipo 2: API de envío	03/10/2025	17/10/2025	Servicio Go funcional para

automatizado de correos electrónicos en Go. Primera fase: estructura base de la API y lógica de envío de correos con soporte para múltiples destinatarios.			envío de correos. Prototipo 2: API de envío automatizado de correos electrónicos.
Integración de RabbitMQ al Prototipo 2: configuración de colas de mensajes para el procesamiento asíncrono de solicitudes de correo. Ajustes de lógica para recepción y procesamiento de mensajes desde el bróker.	06/10/2025	24/10/2025	Servicio con RabbitMQ. Integrado. Prototipo 2: API de envío automatizado de correos electrónicos.
Integración de mensajes desde AS400: desarrollo del módulo para recibir y parsear tramas provenientes del sistema AS400, extracción de campos desde archivos TXT y mapeo al cuerpo del correo.	27/10/2025	07/11/2025	Módulo de integración AS400 funcional. Prototipo 2: API de envío automatizado de correos electrónicos.
Integración con MongoDB al Prototipo 2: implementación del CRUD completo para gestión de plantillas de correo (archivos TXT). Lógica de búsqueda: primero en MongoDB y, si no existe, en la ruta del sistema; almacenamiento posterior en la base de datos.	03/11/2025	05/12/2025	CRUD de plantillas en MongoDB funcional e integrado. Prototipo 2: API de envío automatizado de correos electrónicos.
Diseño y ejecución de pruebas funcionales para ambos prototipos: validación de comportamientos esperados, identificación de errores y análisis de resultados. Depuración de código y ajustes derivados de las pruebas.	25/09/2025	12/12/2025	Casos de prueba ejecutados y registro de resultados. Aseguramiento de calidad: diseño y ejecución de pruebas funcionales.
Elaboración de documentación técnica de los prototipos: análisis, diseño, registro de pruebas y resultados, bajo los estándares del Sistema de Gestión Integrado (SGI).	01/10/2025	17/12/2025	Documentación técnica de los prototipos.
Inducción institucional, acceso a herramientas de trabajo y plataformas de Sistemas GyG S.A. (Teams, GitLab, entornos de desarrollo). Firma del contrato y recepción de equipo de trabajo.	19/08/2025	22/08/2025	Inducción institucional y capacitación técnica.
Investigación sobre programación concurrente en Go: canales, actores y memoria compartida. Comparativa técnica con el lenguaje Rust.	20/08/2025	21/08/2025	Documento de investigación comparativa Go vs. Rust. Inducción institucional y capacitación técnica.
Investigación sobre IBM iMerlin (IBM, 2026): características, funcionalidades,	22/08/2025	02/09/2025	Inducción institucional y capacitación técnica.

instalación, herramientas RPGLE y GitLab. Presentación ante el equipo de Gerencia de Innovación.			
Realización de cursos de inducción corporativa: Inducción a la Compañía, Sistema de Gestión de Calidad (SGC), Sistema de Gestión de Innovación (SGI), Sistema de Gestión de Seguridad de la Información y Ciberseguridad (SGSI) y Desarrollo Gráfico.	21/08/2025	29/08/2025	Cursos completados bajo los estándares del SGI de Sistemas GyG S.A. Inducción institucional y capacitación técnica.
Ajustes a proyecto Angular/Node.js de un compañero: revisión del proyecto, instalación de Node.js, comprensión de la arquitectura Angular y aplicación de correcciones funcionales solicitadas por el jefe de área.	05/12/2025	17/12/2025	Ajustes a proyecto Angular/Node.js.

6.2. Herramientas utilizadas en el desarrollo del proyecto

El desarrollo de los prototipos y las actividades complementarias de la pasantía requirió el uso de un conjunto de herramientas tecnológicas especializadas. En la siguiente tabla se presenta cada herramienta, su versión y la función que cumplió dentro del proyecto. La selección de estas herramientas estuvo determinada por los estándares tecnológicos de Sistemas GyG S.A. y los requerimientos técnicos de cada desarrollo.

Tabla 3. Herramientas utilizadas en el desarrollo del proyecto.

Herramienta utilizada	Versión	Para qué la utilizó
Go (Go, 2026)	1.25.x	Lenguaje principal para el desarrollo de ambos prototipos funcionales: procesador Excel y API de correos. (Kernighan, 2015)
Visual Studio Code (Visual Studio Code, 2025)	1.118	Entorno de desarrollo integrado (IDE) utilizado para escribir, depurar y ejecutar el código fuente en Go.
GitLab (GitLab, 2026)	Cloud	Plataforma de control de versiones y gestión de repositorios. Usada para compartir el código con el equipo y gestionar ramas del proyecto.
RabbitMQ (RabbitMQ, 2026)	4.3.0	Message broker utilizado en

		el Prototipo 2 para gestionar colas de mensajes asíncronos entre los productores (AS400) y el servicio de correos.
MongoDB (MongoDB, 2025)	MongoDB 8.0	Base de datos NoSQL utilizada en el Prototipo 2 para almacenar y consultar plantillas de correo (archivos TXT) mediante operaciones CRUD.
Gin (framework Go)	Go 1.25	Framework web para Go utilizado en el desarrollo de la API REST del servicio de envío de correos.
GORM	Go 1.25	ORM para Go utilizado en el módulo de persistencia con MySQL durante la fase de capacitación.
Udemy (Udemy, 2025)	Web	Plataforma de cursos en línea. Se completó el curso completo de Go (22 secciones) con certificado de finalización.
Microsoft Teams	Cloud	Herramienta de comunicación y colaboración corporativa utilizada para reuniones de seguimiento con el equipo de Gerencia de Innovación.
AS400 (IBM i)	N/A	Sistema heredado de la empresa que genera tramas de datos consumidas por el Prototipo 2 para construir el cuerpo de correos automatizados.

6.3. Desarrollo de actividades por objetivo específico

A continuación, se describe en detalle cómo se desarrolló cada actividad durante la pasantía, organizando el contenido en función de los objetivos específicos establecidos en la propuesta. Por razones de confidencialidad propia del sector financiero, el código fuente de los desarrollos no se incluye en este documento; en su lugar, se presentan

diagramas de flujo y de secuencia que describen de forma precisa la lógica y arquitectura de cada prototipo, así como capturas de pantalla de los resultados funcionales obtenidos.

6.3.1. Inducción institucional y capacitación técnica.

Durante las primeras semanas de la pasantía, entre el 19 de agosto y el 1 de octubre de 2025, se realizó un proceso de inducción integral que abarcó tanto los aspectos corporativos de Sistemas GyG S.A. como la formación técnica requerida para los desarrollos que vendrían en las fases siguientes.

Inducción institucional. Desde el primer día se gestionó el acceso a las plataformas corporativas (Microsoft Teams, GitLab, entornos de desarrollo internos) y se asignó el equipo de trabajo. Durante los primeros días se elaboró la hoja de vida corporativa solicitada por la empresa y se participó en la firma del contrato de pasantía. Los cursos de inducción obligatorios completados fueron: Inducción a la Compañía, Sistema de Gestión de Calidad (SGC), Sistema de Gestión de Innovación (SGI), Sistema de Gestión de Seguridad de la Información y Ciberseguridad (SGSI) y Desarrollo Gráfico. Estos cursos sentaron las bases conceptuales para trabajar bajo los estándares del SGI de la organización.

Figura 1

Certificado curso Inducción a la Compañía.



Certificado curso Inducción SGC.

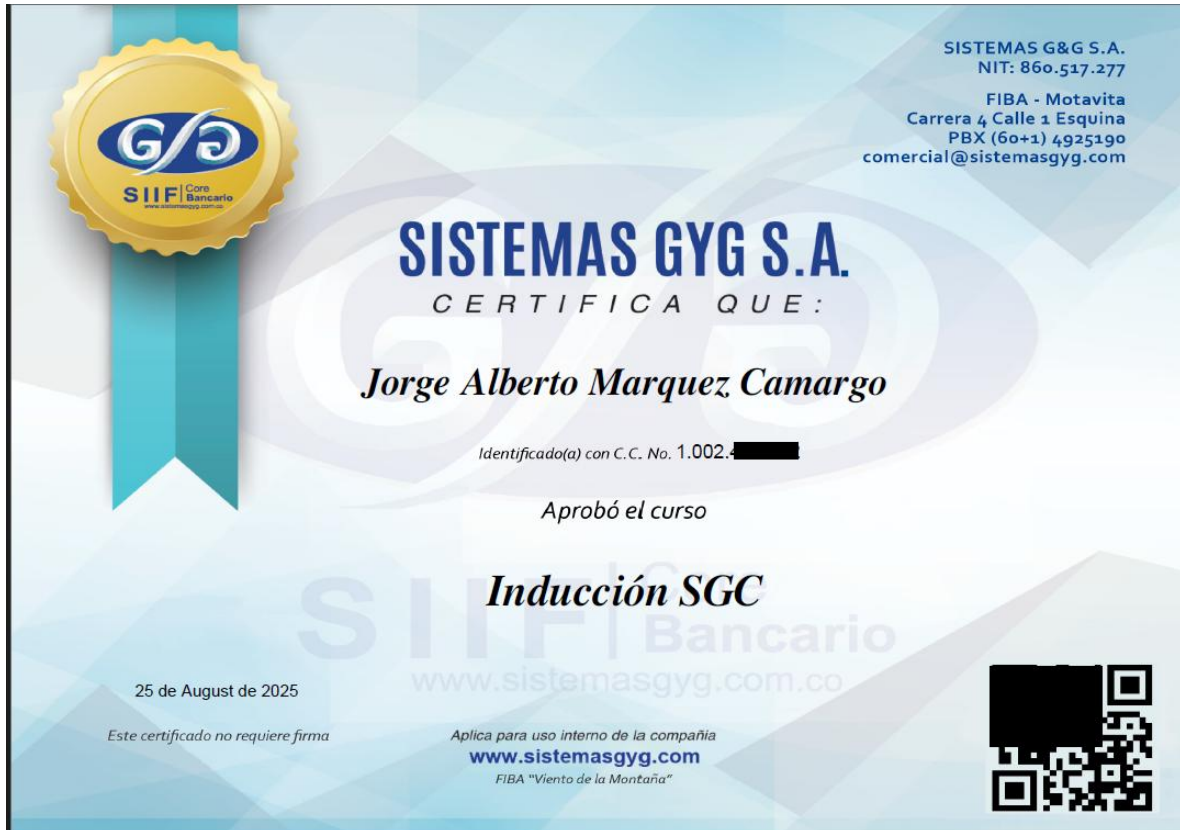
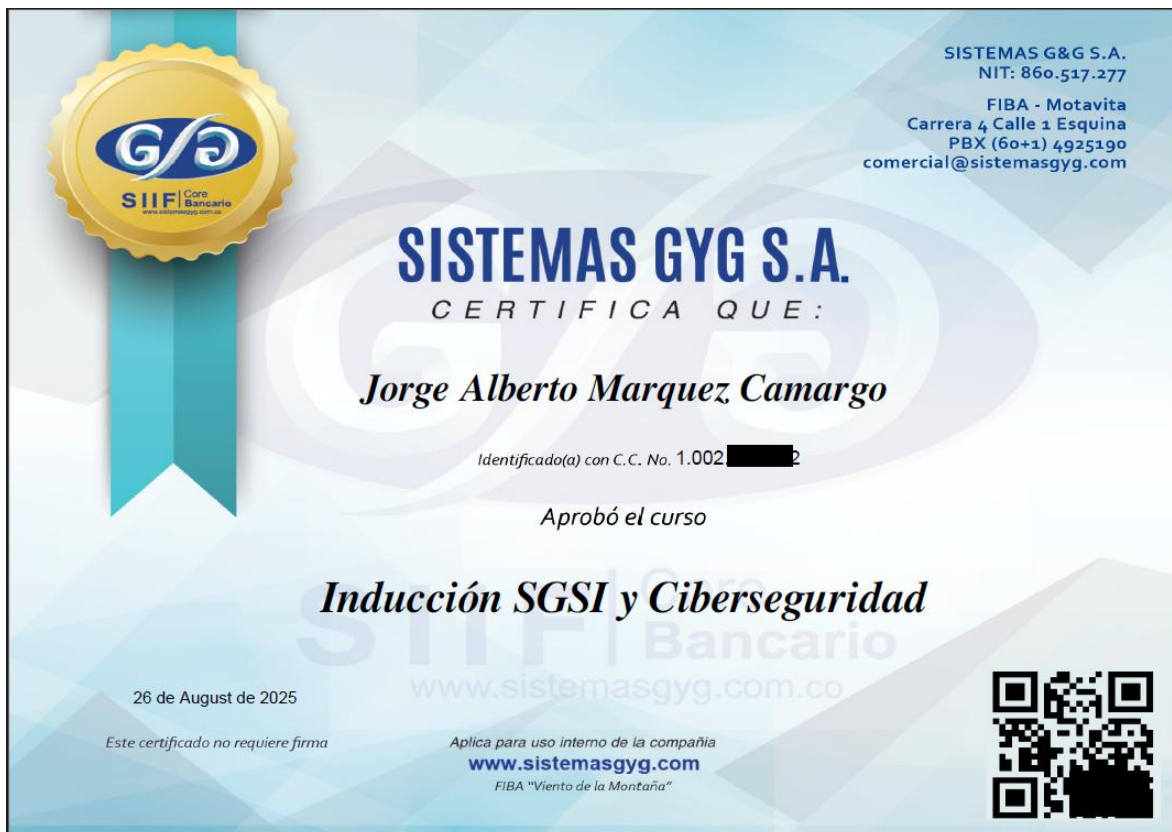


Figura 3

Certificado curso Inducción SGSI y Ciberseguridad.



Certificado curso Inducción SGC.



Figura 5

Certificado curso Desarrollo Gráfico.



Investigaciones técnicas asignadas. Paralelamente a los cursos, se asignaron dos investigaciones técnicas. La primera consistió en un análisis comparativo de técnicas de programación concurrente en Go (canales, actores y memoria compartida) frente al lenguaje Rust, con énfasis en rendimiento y seguridad de memoria; esta investigación fue presentada ante el grupo de Gerencia de Innovación. La segunda investigación se centró en IBM iMerlin, cubriendo sus características, funcionalidades e integración con herramientas como RPGLE y GitLab, considerando su aplicabilidad en los procesos de instalación de los sistemas de la empresa. Esta segunda investigación también fue expuesta formalmente al equipo de innovación.

Figura 6

Investigación Técnicas de Programación.



Nota: El documento de la presentación está en los anexos.

Figura 7

Investigación IBM.



Nota: El documento de la presentación está en los anexos.

Capacitación en Go. Entre el 1 de septiembre y el 1 de octubre de 2025 se completó un curso completo de Go en la plataforma Udemy, que comprendió 22 secciones: desde los fundamentos del lenguaje (variables, control de flujo, estructuras de datos, manejo de errores) hasta temas avanzados como concurrencia con goroutines y canales, programación orientada a objetos e interfaces, genéricos, desarrollo de APIs RESTful con el framework Gin, persistencia de datos con MySQL usando GORM, pruebas unitarias y desarrollo de aplicaciones web en Go. Al finalizar el curso se obtuvo el certificado, el cual se adjunta como Anexo 1 de este documento. (GO, 2026)

Figura 8

Certificado de finalización del curso Go – Udemy.



6.3.2. Prototipo 1: Procesador automatizado de archivos Excel con programación concurrente.

El primer prototipo funcional desarrollado durante la pasantía fue un programa en Go orientado a automatizar el procesamiento de archivos Excel. Esta herramienta surgió como respuesta a una necesidad real del área: el equipo de Gerencia de Innovación manejaba archivos Excel con grandes volúmenes de datos que requerían validaciones y transformaciones que antes se hacían manualmente. El desarrollo se llevó a cabo entre el 15 de septiembre y el 3 de octubre de 2025.

Descripción del problema. El área necesitaba un mecanismo que leyera archivos Excel con información tabular, validara los datos fila por fila según reglas de negocio predefinidas, consolidara los resultados en un nuevo archivo de salida y realizara todo

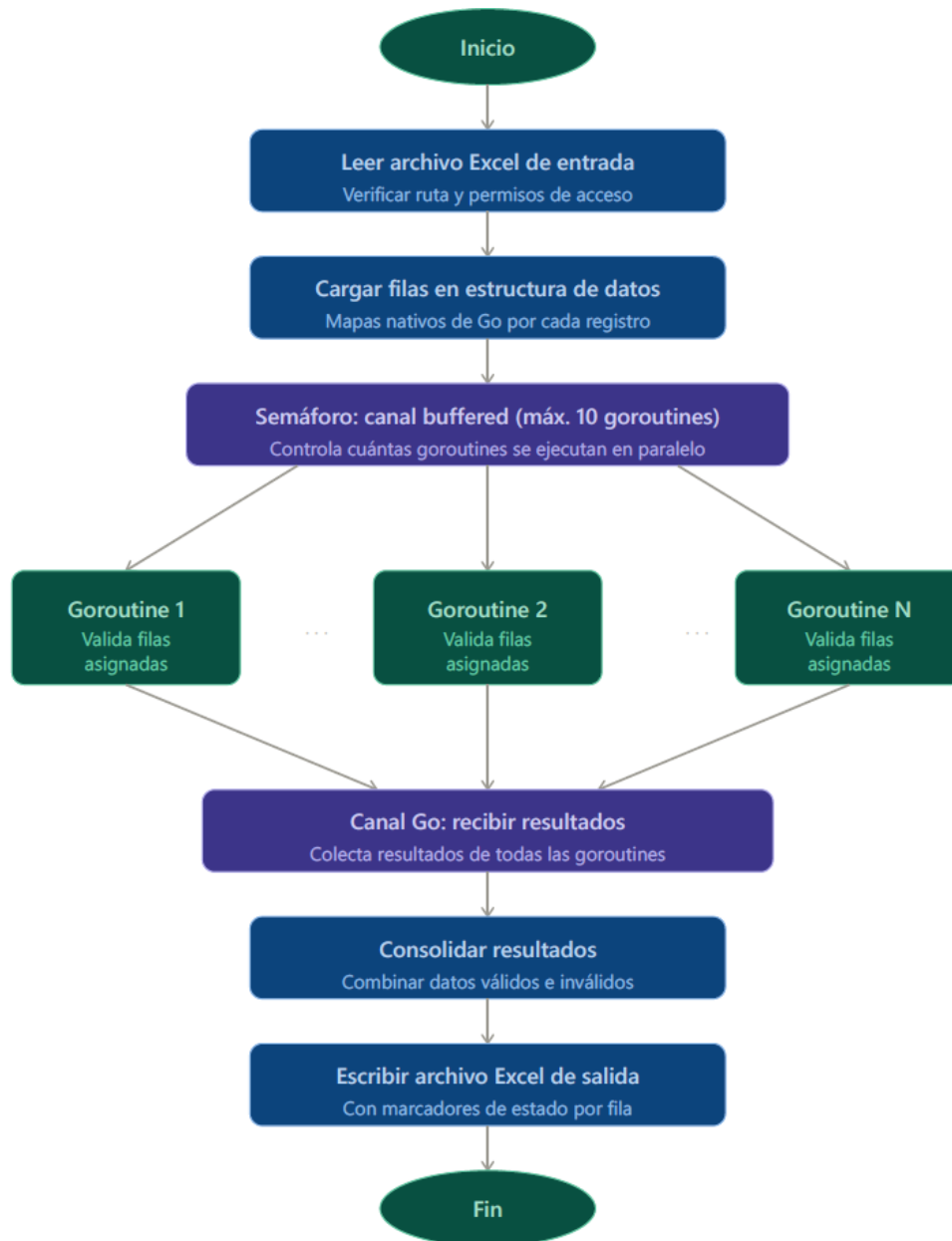
este proceso de la manera más eficiente posible, aprovechando el procesamiento paralelo.

Solución técnica implementada. Se desarrolló un programa en Go que, en primera instancia, lee el archivo Excel de entrada y carga los registros en estructuras de datos nativas del lenguaje (mapas). Una vez cargados los datos, el programa distribuye el trabajo entre un conjunto de goroutines (Cutajar, 2023) —unidades de ejecución ligeras de Go— limitadas a un máximo de 10 hilos concurrentes, controladas mediante un semáforo implementado con un canal de tipo buffered. Cada goroutine procesa las filas asignadas de manera independiente: aplica las validaciones correspondientes y envía sus resultados a un canal de salida. El programa principal recibe los resultados de todas las goroutines a medida que están disponibles, los consolida y finalmente escribe el archivo Excel de salida con los datos procesados. El código se publicó en un repositorio de GitLab para revisión del jefe de área

El siguiente diagrama de flujo ilustra la lógica general del prototipo, desde la lectura del archivo hasta la escritura del resultado final:

Figura 9

Diagrama de flujo – Prototipo 1: Procesador de Excel con goroutines.



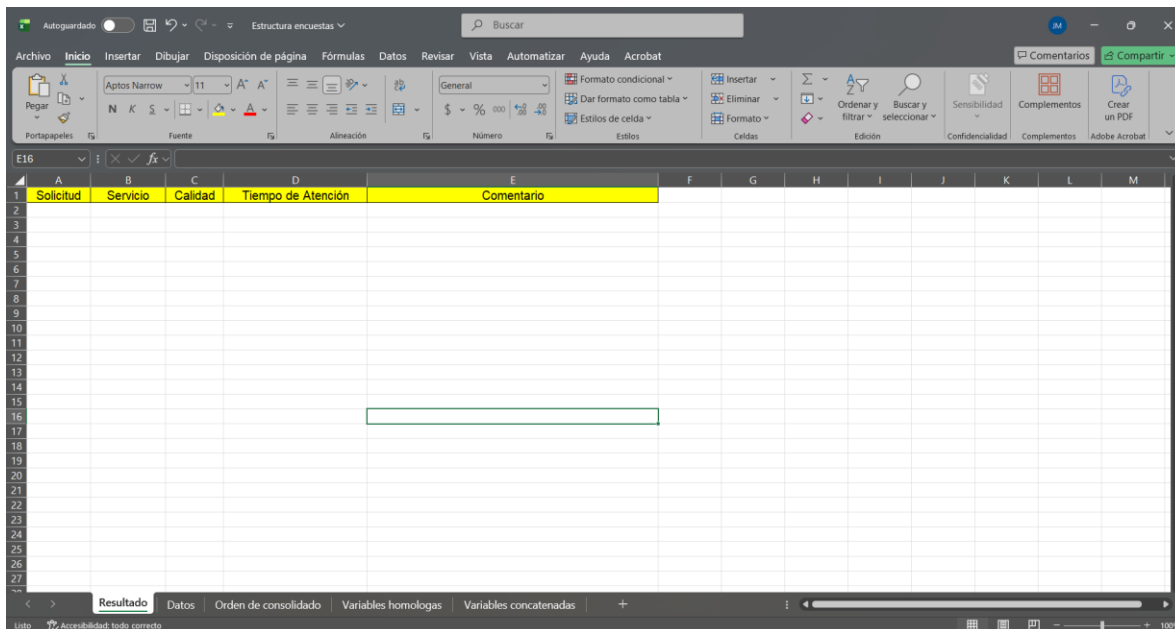
Aspectos técnicos destacados. La elección de goroutines como mecanismo de concurrencia se fundamenta en su bajo costo de creación en comparación con hilos del sistema operativo (el run time de Go puede gestionar miles de goroutines con un consumo mínimo de memoria). El uso de un canal buffered como semáforo garantiza que nunca se

supere el límite de 10 goroutines activas simultáneamente, evitando la saturación de recursos. Los canales de comunicación entre goroutines eliminan la necesidad de bloqueos explícitos (mutex), haciendo el código más seguro y legible.

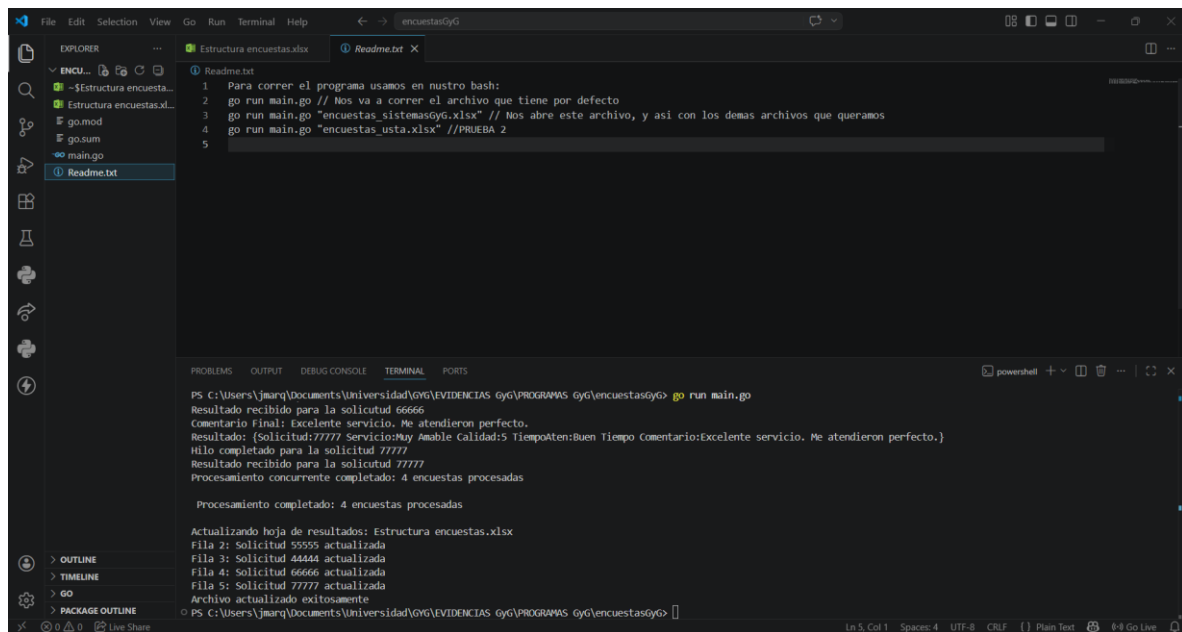
Ajustes durante el desarrollo. Tras la primera entrega, el jefe de área solicitó dos ajustes principales: primero, la creación de un repositorio en GitLab para facilitar la revisión del código; segundo, la implementación del control de concurrencia (límite de 10 goroutines) que no estaba presente en la versión inicial. Ambos ajustes se realizaron durante la semana del 22 al 26 de septiembre de 2025 y fueron revisados y aprobados en reunión con el jefe de innovación el 3 de octubre de 2025.

Figura 10

Captura de pantalla – Ejecución del Prototipo 1



Nota: Esta captura de pantalla es el archivo Excel antes de que se corre el programa.

Figura 11*Captura de pantalla – Ejecución del Prototipo 1*

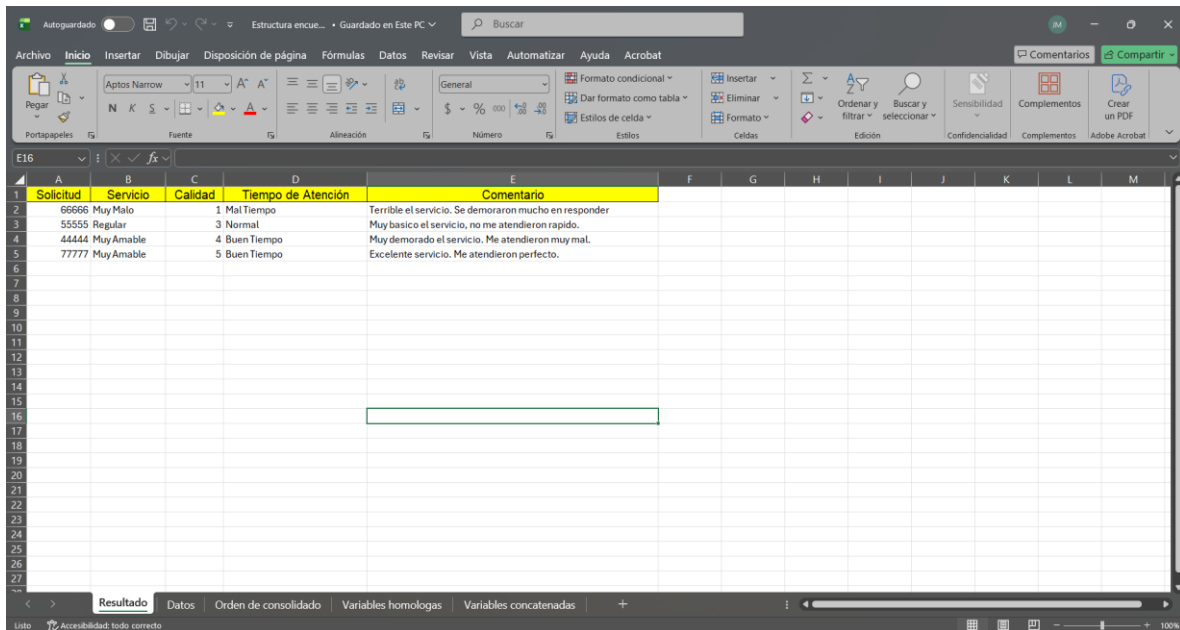
```
1 Para correr el programa usamos en nuestro bash:
2 go run main.go // Nos va a correr el archivo que tiene por defecto
3 go run main.go "encuestas sistemasgyg.xlsx" // Nos abre este archivo, y así con los demas archivos que queramos
4 go run main.go "encuestas_usta.xlsx" //PRUEBA 2
5
```

```
PS C:\Users\jmarq\Documents\Universidad\GYG\EVIDENCIAS GyG\PROGRAMAS GyG\encuestasGyG> go run main.go
Resultado recibido para la solicitud 66666
Comentario Final: Excelente servicio. Me atendieron perfecto.
Resultado: {Solicitud:77777 Servicio:Muy Amable Calidad:5 Tiemposten:Buen Tiempo Comentario:Excelente servicio. Me atendieron perfecto.}
Hilo completado para la solicitud 77777
Resultado recibido para la solicitud 77777
Procesamiento concurrente completado: 4 encuestas procesadas

Procesamiento completado: 4 encuestas procesadas

Actualizando hoja de resultados: Estructura encuestas.xlsx
Fila 2: Solicitud 55555 actualizada
Fila 3: Solicitud 44444 actualizada
Fila 4: Solicitud 66666 actualizada
Fila 5: Solicitud 77777 actualizada
Archivo actualizado exitosamente
PS C:\Users\jmarq\Documents\Universidad\GYG\EVIDENCIAS GyG\PROGRAMAS GyG\encuestasGyG>
```

Nota: En esta captura se corre el programa.

Figura 12*Captura de pantalla – Ejecución del Prototipo 1*


Solicitud	Servicio	Calidad	Tiempo de Atención	Comentario
66666	Muy Malo	1	Mal Tiempo	Terrible el servicio. Se demoraron mucho en responder
55555	Regular	3	Normal	Muy basico el servicio, no me atendieron rapido.
44444	Muy Amable	4	Buen Tiempo	Muy demorado el servicio. Me atendieron muy mal.
77777	Muy Amable	5	Buen Tiempo	Excelente servicio. Me atendieron perfecto.

Nota: En esta captura está el Excel luego de que se corre el programa.

6.3.3. Prototipo 2: API de envío automatizado de correos electrónicos.

El segundo y más complejo prototipo desarrollado durante la pasantía fue un servicio en Go para el envío automatizado de correos electrónicos, integrado con un sistema de mensajería asíncrona (RabbitMQ), una base de datos NoSQL (MongoDB) y el sistema heredado AS400. El desarrollo de este prototipo se extendió entre el 3 de octubre y el 5 de diciembre de 2025, con múltiples iteraciones de ajuste y extensión de funcionalidades.

Descripción del problema. Sistemas GyG S.A. requería un mecanismo que permitiera a sus sistemas internos, en particular al sistema AS400, disparar envíos de correo electrónico de forma automática sin intervención manual. Los correos debían construirse a partir de plantillas predefinidas, soportar múltiples destinatarios y ser procesados de forma asíncrona para no bloquear los procesos del sistema origen.

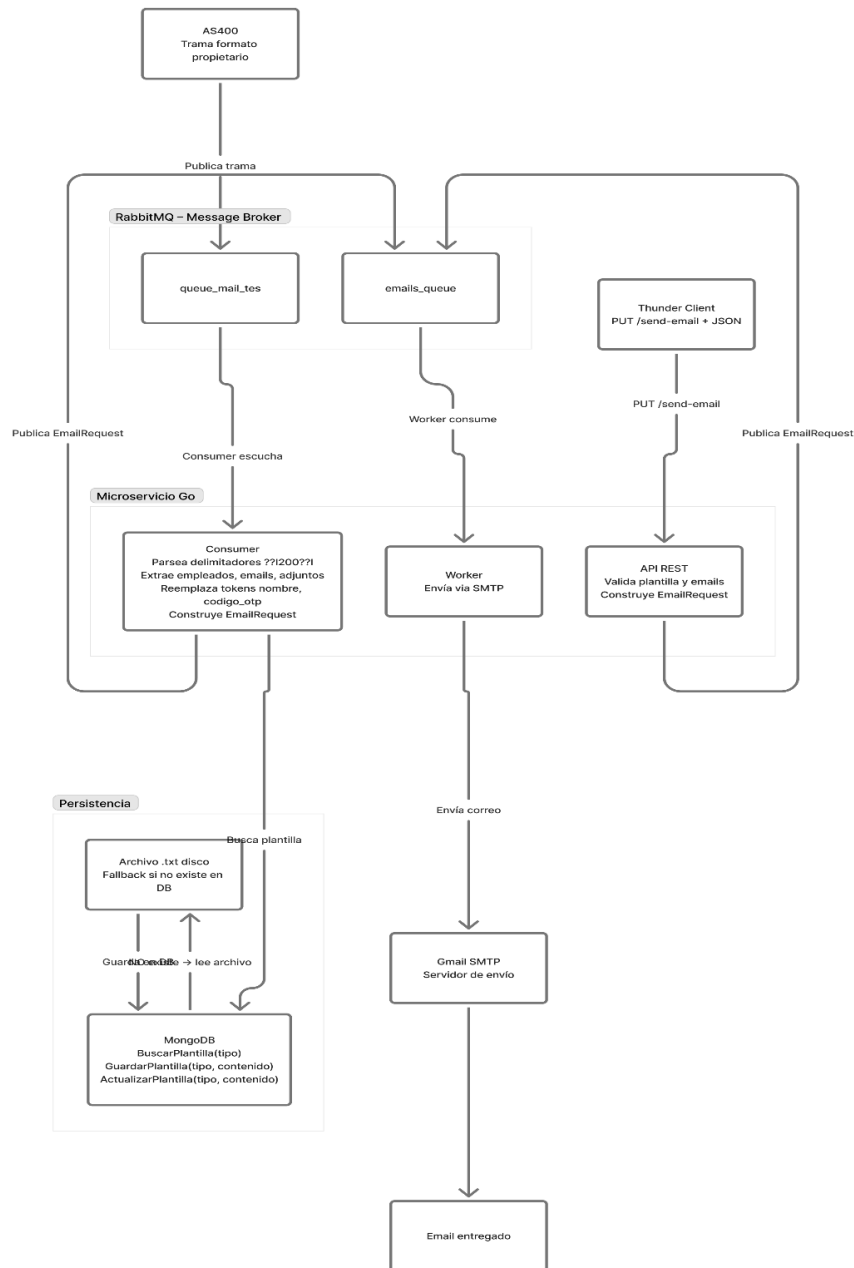
Arquitectura de la solución. La solución se diseñó como un microservicio en Go con los siguientes componentes: (1) un consumidor de RabbitMQ que escucha la cola de mensajes y procesa cada solicitud de correo de manera asíncrona; (2) un módulo de

parseo que interpreta las tramas provenientes de AS400, extrayendo los campos necesarios (destinatarios, asunto, datos para el cuerpo del mensaje); (3) un módulo de gestión de plantillas que consulta primero en MongoDB si existe la plantilla requerida y, si no la encuentra, la busca en el sistema de archivos local (TXT), la lee y la persiste en MongoDB para optimizar consultas futuras; (4) un módulo de construcción y envío del correo que combina los datos del mensaje con la plantilla y lo despacha via SMTP. Adicionalmente, se expuso una API REST en Go con Gin para la gestión CRUD de las plantillas almacenadas en MongoDB. (Newman, 2021).

El siguiente diagrama de secuencia muestra el flujo completo de procesamiento de un mensaje, desde que AS400 lo publica en la cola hasta que el correo es enviado y se confirma el procesamiento:

Figura 13

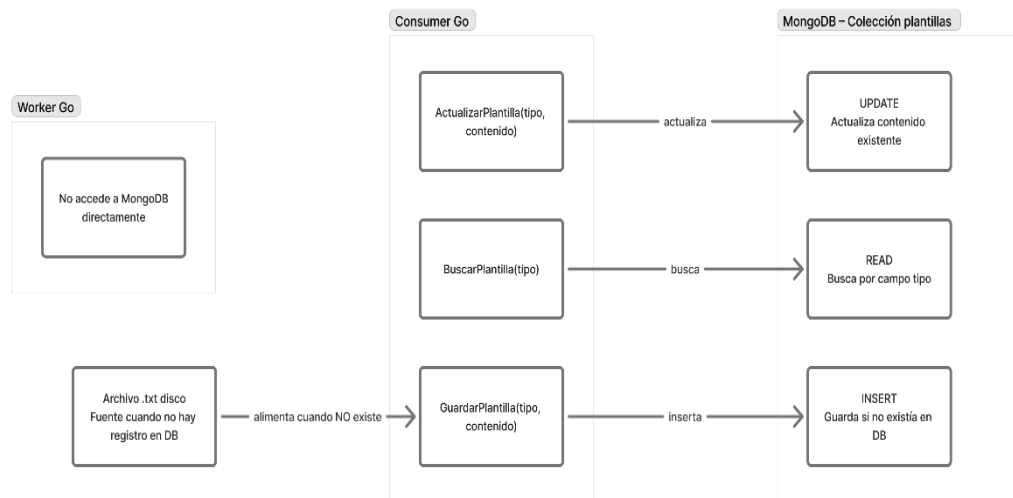
Diagrama Arquitectura general del microservicio de envío automatizado de correos electrónicos.



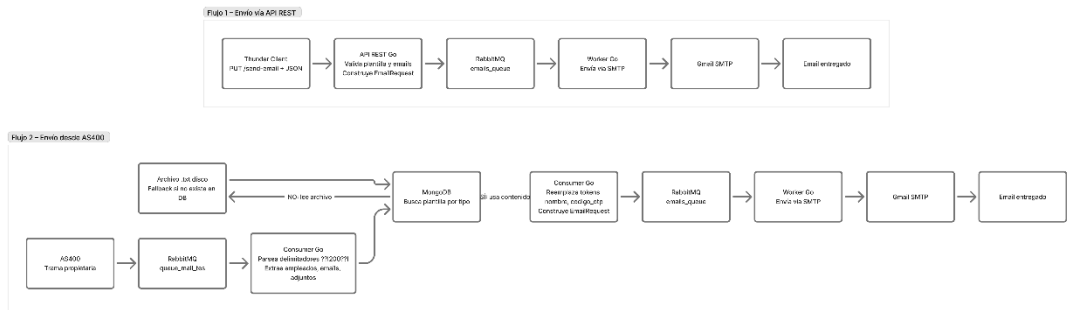
Nota: vista completa de todos los componentes: los dos orígenes (Thunder Client y AS400), el microservicio Go con sus tres roles (API REST, Consumer, Worker), las dos colas de RabbitMQ, MongoDB con su fallback a disco, Gmail SMTP y el email final.

Figura 14

Diagrama Flujos del Prototipo 2

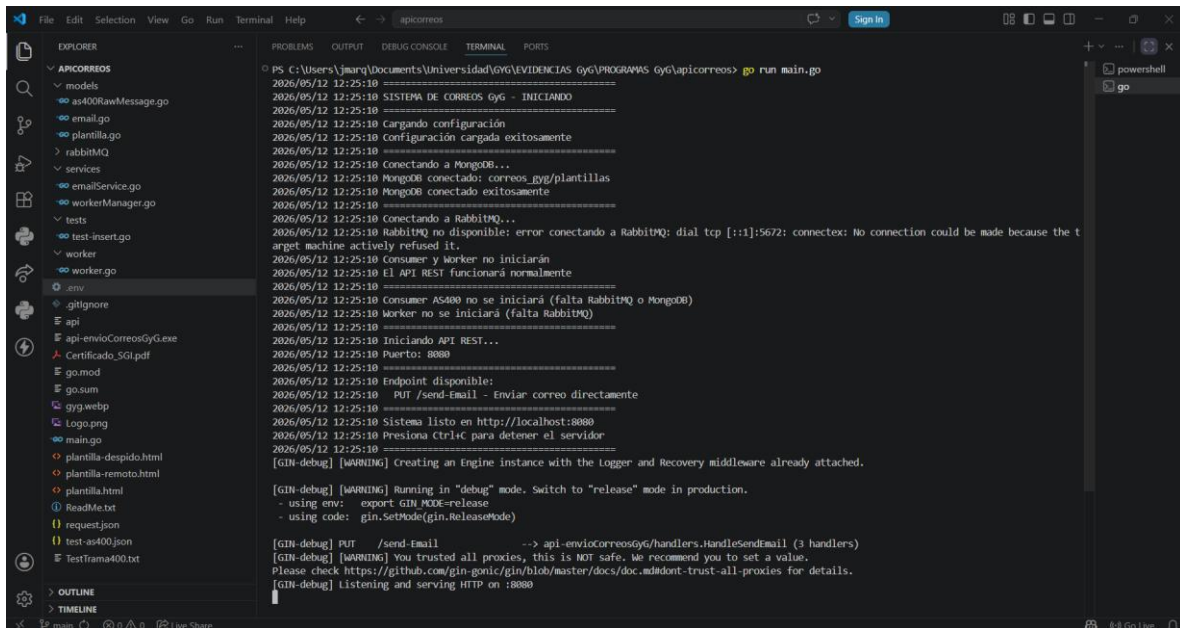


Nota: los dos caminos de entrada comparados visualmente, con la lógica de decisión de MongoDB (Sí/NO existe plantilla) bien marcada en el Flujo 2.

Figura 15*Diagrama Interacciones MongoDB*

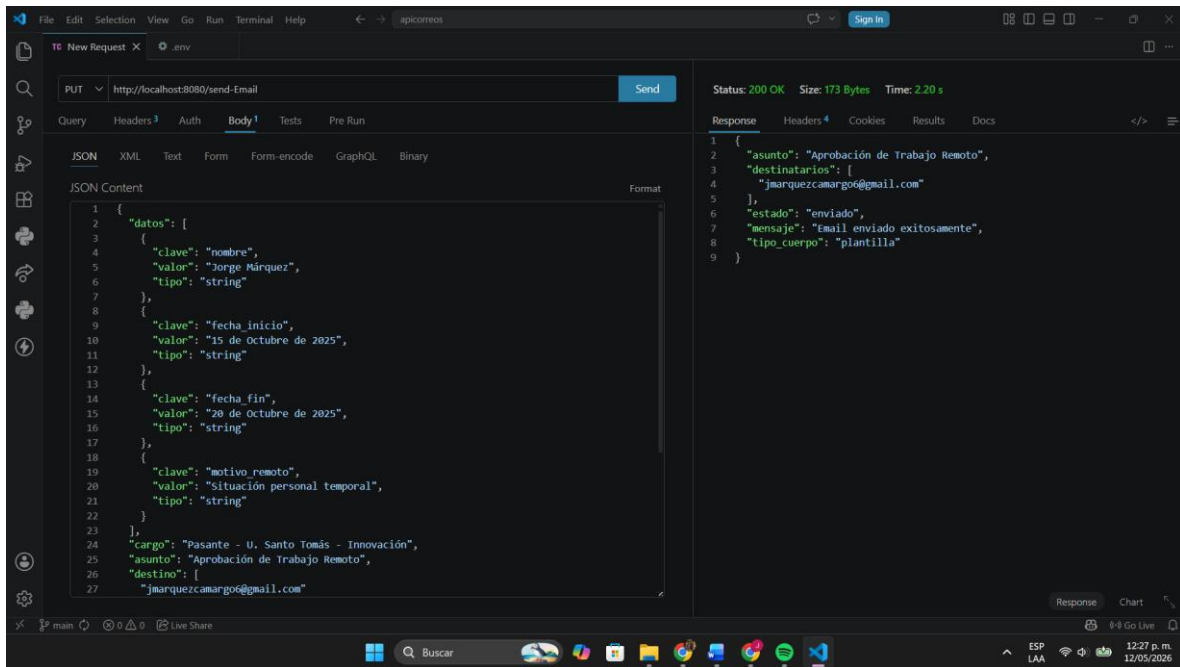
Nota: Detalle técnico de las tres operaciones CRUD (BuscarPlantilla, GuardarPlantilla, ActualizarPlantilla).

Desarrollo iterativo. El prototipo se construyó de forma incremental a lo largo de tres meses, con revisiones periódicas con el jefe de innovación. La primera fase cubrió la estructura base del servicio y la lógica de envío de correos. La segunda fase incorporó RabbitMQ como capa de mensajería asíncrona. La tercera fase añadió el módulo de integración con AS400, incluyendo el parseo de tramas con separadores codificados y la lectura de archivos TXT desde rutas del sistema. La cuarta y última fase integró MongoDB mediante un CRUD completo para las plantillas, implementando además la lógica de búsqueda con fallback al sistema de archivos y almacenamiento posterior en la base de datos.

Figura 16*Ejecución Inicial del prototipo 2*

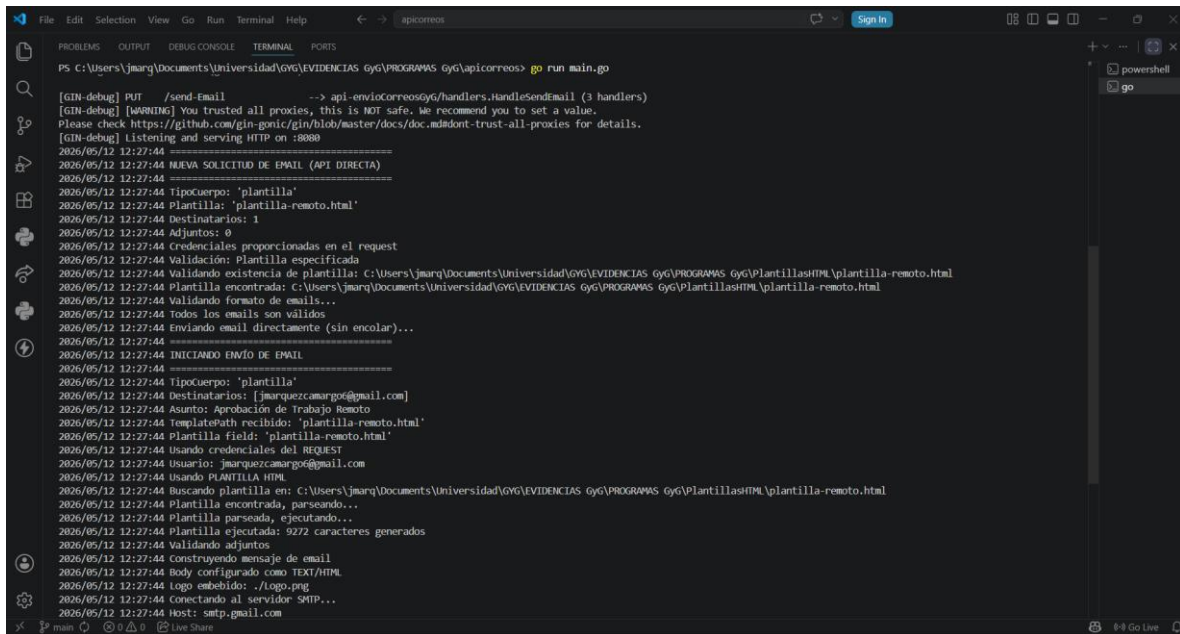
```
PS C:\Users\jmarq\Documents\Universidad\GyG\EVIDENCIAS GyG\PROGRAMAS GyG\apicorreos> go run main.go
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 SISTEMA DE CORREOS GyG - INICIANDO
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 Cargando configuración
2026/05/12 12:25:10 Configuración cargada exitosamente
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 Conectando a MongoDB...
2026/05/12 12:25:10 MongoDB conectado: correos_gyg/plantillas
2026/05/12 12:25:10 MongoDB conectado exitosamente
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 Conectando a RabbitMQ...
2026/05/12 12:25:10 RabbitMQ no disponible: error conectando a RabbitMQ: dial tcp [::1]:5672: connect: No connection could be made because the target machine actively refused it.
2026/05/12 12:25:10 Consumer y Worker no iniciarán
2026/05/12 12:25:10 El API REST funcionará normalmente
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 Consumer AS400 no se iniciará (falta RabbitMQ o MongoDB)
2026/05/12 12:25:10 Worker no se iniciará (falta RabbitMQ)
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 Iniciando API REST...
2026/05/12 12:25:10 Puerto: 8080
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 Endpoint disponible:
2026/05/12 12:25:10 PUT /send-email - Enviar correo directamente
2026/05/12 12:25:10 =====
2026/05/12 12:25:10 Sistema listo en http://localhost:8080
2026/05/12 12:25:10 Presiona ctrl+c para detener el servidor
2026/05/12 12:25:10 =====
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env: export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)
[GIN-debug] PUT    /send-email      --> api.envioCorreosGyG/handlers.handleSendEmail (3 handlers)
[GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. We recommend you to set a value.
Please check https://github.com/gin-gonic/gin/blob/master/docs/doc.md#dont-trust-all-proxies for details.
[GIN-debug] Listening and serving HTTP on :8080
```

Nota: En esta captura se puede visualizar la terminal en el programa Visual Studio Code que es donde se corre el programa, además de esto se puede observar que en el momento no se encuentra disponible la conexión a RabbitMQ puesto que este prototipo se realizó con una conexión local.

Figura 17*Prueba del Prototipo 2 - Thunder Client*

Nota: En esta captura podemos visualizar el contenido tipo JSON para el envío del correo por medio de Thunder Client (extensión de Visual Studio).

Figura 18

Estado primera ejecución del Prototipo 2

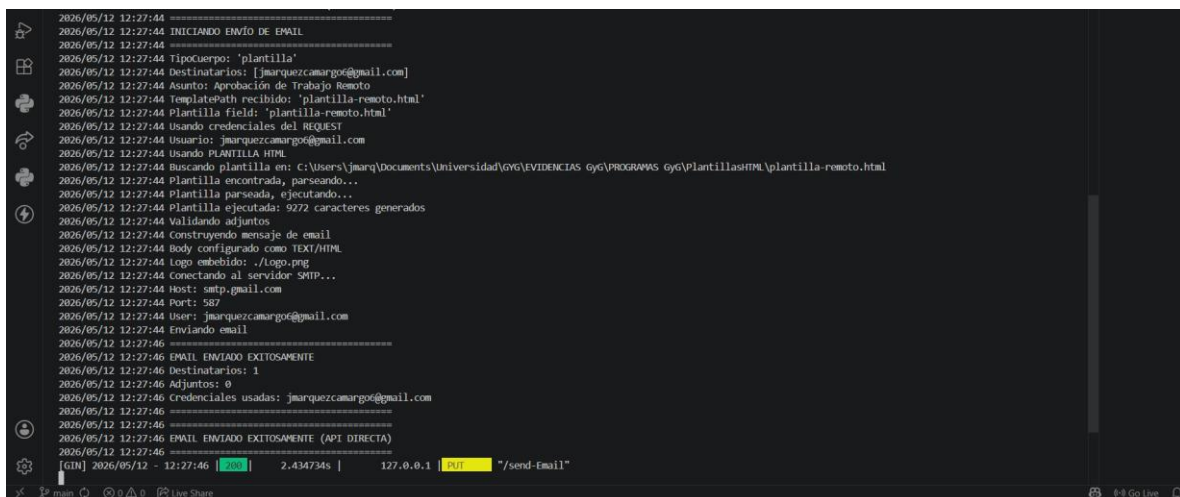
```
PS C:\Users\jmarq\Documents\Universidad\GyG\EVIDENCIAS GyG\PROGRAMAS GyG\apicorreo> go run main.go

[GIN-debug] PUT    /send-Email          --> api-envioCorreoGyG/handlers.HandleSendEmail (3 handlers)
[GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. We recommend you to set a value.
Please check https://github.com/gin-gonic/gin/blob/master/docs/doc.md#dont-trust-all-proxies-for-details.
[GIN-debug] Listening and serving HTTP on :8080

2026/05/12 12:27:44 =====
2026/05/12 12:27:44 NUEVA SOLICITUD DE EMAIL (API DIRECTA)
2026/05/12 12:27:44 =====
2026/05/12 12:27:44 Tipocuerpo: 'plantilla'
2026/05/12 12:27:44 Plantilla: 'plantilla-remoto.html'
2026/05/12 12:27:44 Destinatarios: 1
2026/05/12 12:27:44 Adjuntos: 0
2026/05/12 12:27:44 Credenciales proporcionadas en el request
2026/05/12 12:27:44 Validación: Plantilla especificada
2026/05/12 12:27:44 Validando existencia de plantilla: C:\Users\jmarq\Documents\Universidad\GyG\EVIDENCIAS GyG\PlantillasHTML\plantilla-remoto.html
2026/05/12 12:27:44 Plantilla encontrada: C:\Users\jmarq\Documents\Universidad\GyG\EVIDENCIAS GyG\PlantillasHTML\plantilla-remoto.html
2026/05/12 12:27:44 Validando formato de emails...
2026/05/12 12:27:44 Todos los emails son válidos
2026/05/12 12:27:44 Enviando email directamente (sin encolar)...
2026/05/12 12:27:44 =====
2026/05/12 12:27:44 INICIANDO ENVÍO DE EMAIL
2026/05/12 12:27:44 =====
2026/05/12 12:27:44 Tipocuerpo: 'plantilla'
2026/05/12 12:27:44 Destinatarios: [jmarquezcarmargo@gmail.com]
2026/05/12 12:27:44 Asunto: Aprobación de Trabajo Remoto
2026/05/12 12:27:44 TemplatePath recibido: 'plantilla-remoto.html'
2026/05/12 12:27:44 Plantilla Field: 'plantilla-remoto.html'
2026/05/12 12:27:44 Usando credenciales del REQUEST
2026/05/12 12:27:44 Usuario: jmarquezcarmargo@gmail.com
2026/05/12 12:27:44 Usando PLANTILLA HTML
2026/05/12 12:27:44 Buscando plantilla en: C:\Users\jmarq\Documents\Universidad\GyG\EVIDENCIAS GyG\PlantillasHTML\plantilla-remoto.html
2026/05/12 12:27:44 Plantilla encontrada, parseando...
2026/05/12 12:27:44 Plantilla parseada, ejecutando...
2026/05/12 12:27:44 Plantilla ejecutada: 9272 caracteres generados
2026/05/12 12:27:44 Validando adjuntos
2026/05/12 12:27:44 Construyendo mensaje de email
2026/05/12 12:27:44 Body configurado como TEXT/HTML
2026/05/12 12:27:44 Logo embebido: ./logo.png
2026/05/12 12:27:44 Conectando al servidor SMTP...
2026/05/12 12:27:44 Host: smtp.gmail.com
```

Nota: En esta captura podemos visualizar la terminal de Visual Studio Code, la cual nos muestra el estado del envío del correo que en este caso fue exitoso.

Figura 19

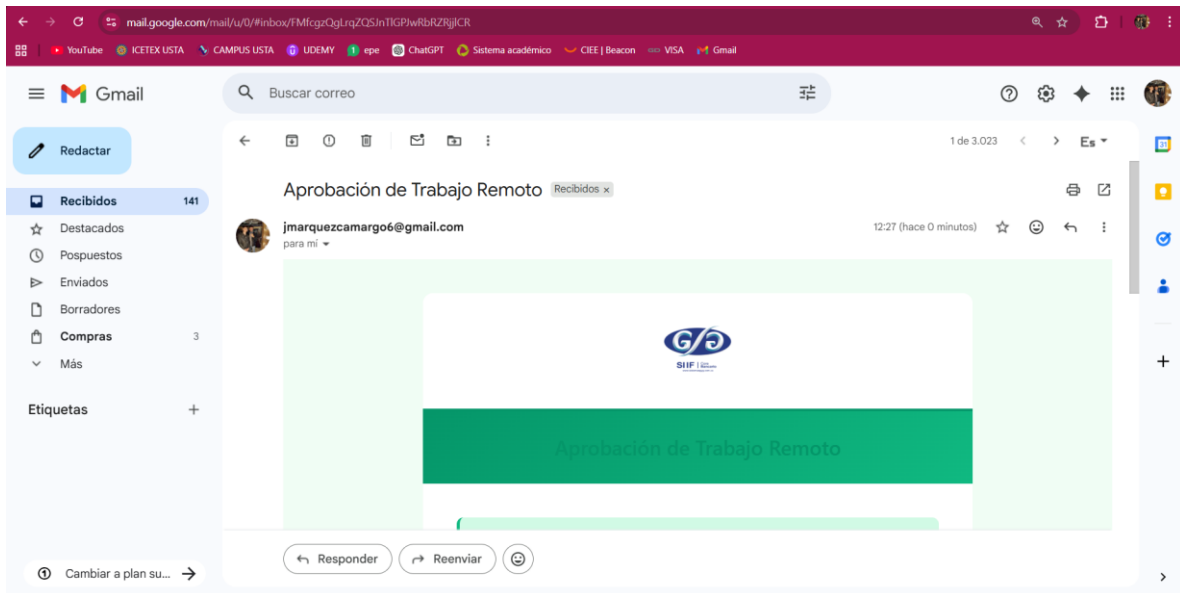
Verificación de la ejecución 1 del Prototipo 2

```
2026/05/12 12:27:44 =====
2026/05/12 12:27:44 INICIANDO ENVÍO DE EMAIL
2026/05/12 12:27:44 =====
2026/05/12 12:27:44 Tipocuerpo: 'plantilla'
2026/05/12 12:27:44 Destinatarios: [jmarquezcarmargo@gmail.com]
2026/05/12 12:27:44 Asunto: Aprobación de Trabajo Remoto
2026/05/12 12:27:44 TemplatePath recibido: 'plantilla-remoto.html'
2026/05/12 12:27:44 Plantilla Field: 'plantilla-remoto.html'
2026/05/12 12:27:44 Usando credenciales del REQUEST
2026/05/12 12:27:44 Usuario: jmarquezcarmargo@gmail.com
2026/05/12 12:27:44 Usando PLANTILLA HTML
2026/05/12 12:27:44 Buscando plantilla en: C:\Users\jmarq\Documents\Universidad\GyG\EVIDENCIAS GyG\PlantillasHTML\plantilla-remoto.html
2026/05/12 12:27:44 Plantilla encontrada, parseando...
2026/05/12 12:27:44 Plantilla parseada, ejecutando...
2026/05/12 12:27:44 Plantilla ejecutada: 9272 caracteres generados
2026/05/12 12:27:44 Validando adjuntos
2026/05/12 12:27:44 Construyendo mensaje de email
2026/05/12 12:27:44 Body configurado como TEXT/HTML
2026/05/12 12:27:44 Logo embebido: ./logo.png
2026/05/12 12:27:44 Conectando al servidor SMTP...
2026/05/12 12:27:44 Host: smtp.gmail.com
2026/05/12 12:27:44 Port: 587
2026/05/12 12:27:44 User: jmarquezcarmargo@gmail.com
2026/05/12 12:27:44 Enviando email
2026/05/12 12:27:46 =====
2026/05/12 12:27:46 EMAIL ENVIADO EXITOSAMENTE
2026/05/12 12:27:46 Destinatarios: 1
2026/05/12 12:27:46 Adjuntos: 0
2026/05/12 12:27:46 Credenciales usadas: jmarquezcarmargo@gmail.com
2026/05/12 12:27:46 =====
2026/05/12 12:27:46 EMAIL ENVIADO EXITOSAMENTE (API DIRECTA)
2026/05/12 12:27:46 =====
[GIN] 2026/05/12 - 12:27:46 | 200 | 2.434734s | 127.0.0.1 | PUT    "/send-Email"
```

Nota: En esta captura podemos visualizar la terminal de Visual Studio Code, la cual nos muestra el estado del envío del correo que en este caso fue exitoso.

Figura 20

Evidencia primera ejecución del Prototipo 2



Nota: En esta captura podemos visualizar que el envío del correo fue exitoso, es decir se cumplió con los requisitos.

Figura 21

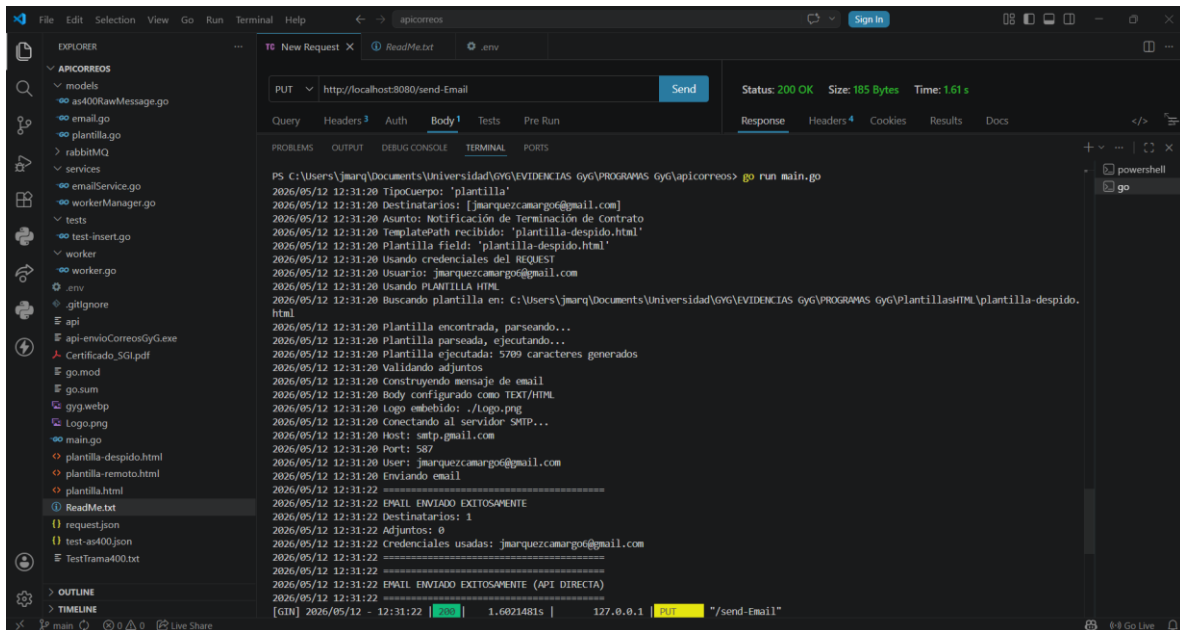
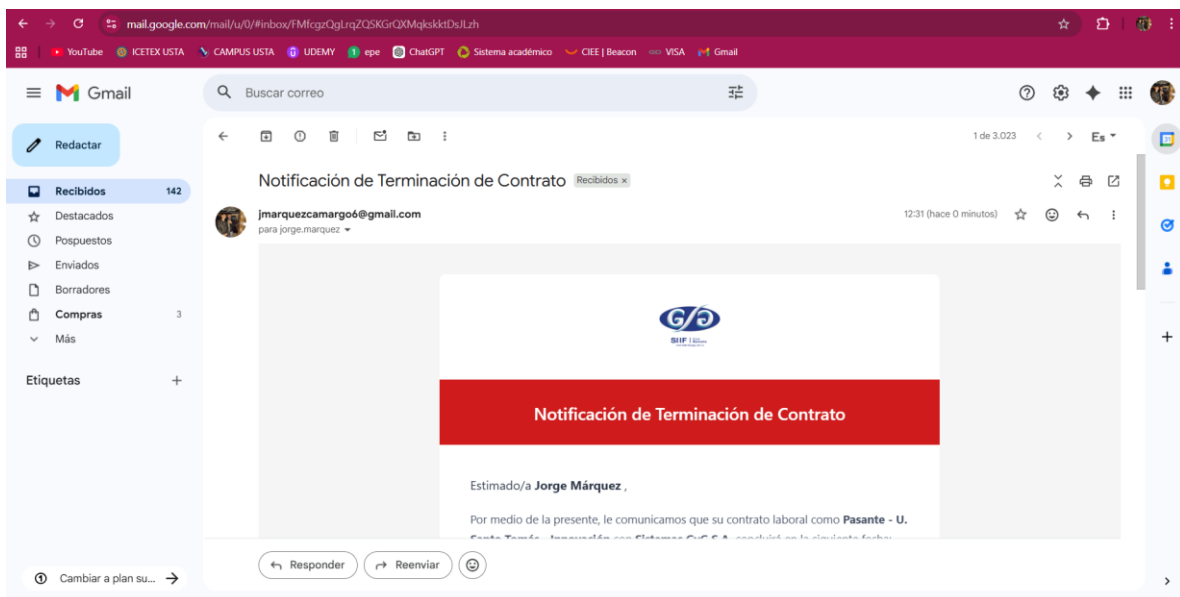
Verificación de la ejecución 2 del Prototipo 2

Figura 22

Evidencia primera ejecución del Prototipo 2

6.3.4. Aseguramiento de calidad: diseño y ejecución de pruebas funcionales

En paralelo con el desarrollo de ambos prototipos, se aplicaron procesos de aseguramiento de calidad que incluyeron el diseño de casos de prueba, su ejecución en entornos de desarrollo y el análisis de los resultados obtenidos (Felderer & Ramler, 2020). Este proceso siguió los lineamientos del Sistema de Gestión Integrado de Sistemas GyG S.A. y se realizó de forma iterativa: cada vez que se completaba una nueva funcionalidad, se ejecutaban las pruebas correspondientes antes de presentarla al jefe de área.

Metodología de pruebas. Las pruebas realizadas fueron de tipo funcional, enfocadas en verificar que cada componente del sistema se comportara según los requerimientos definidos. Para el Prototipo 1 se validó la correcta lectura del archivo Excel, el comportamiento del sistema de goroutines bajo distintas cargas, la validación de datos por fila y la escritura del archivo de salida. Para el Prototipo 2 se validaron todos los componentes del flujo: la recepción de mensajes desde RabbitMQ, el parseo de tramas AS400, la búsqueda y fallback de plantillas, el CRUD de MongoDB y el envío efectivo de correos a múltiples destinatarios. Las pruebas se ejecutaron utilizando herramientas de cliente REST (Postman) para los endpoints de la API, registros de consola (logs) para la validación del flujo interno del servicio, y revisiones manuales de los archivos de salida generados.

Tabla 4. Casos de prueba – Prototipo 1: Procesador de Excel con goroutines.

ID	Descripción del caso	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-01	Lectura correcta del archivo Excel	Archivo Excel con 100 filas de datos válidos	El programa carga todas las filas sin error	100 filas cargadas correctamente en el mapa Go	✓ PASS
CP-02	Ejecución concurrente con goroutines	Archivo con 50 filas; límite de 10 goroutines	Se crean máximo 10 goroutines simultáneas	10 goroutines ejecutadas en paralelo; resto en espera	✓ PASS
CP-	Validación de	Fila con	El programa	Fila marcada	✓ PASS

ID	Descripción del caso	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
03	datos por fila	campo obligatorio vacío	identifica la fila como inválida y la registra	con error de validación en la salida	
CP-04	Escritura de resultados en Excel de salida	Procesamiento exitoso de 100 filas	Excel de salida contiene resultados consolidados	Archivo Excel generado con todos los resultados	✓ PASS
CP-05	Manejo de archivo no encontrado	Ruta de archivo inválida o inexistente	El programa retorna error descriptivo y termina limpiamente	Error capturado y registrado en consola sin panic	✓ PASS
CP-06	Rendimiento: tiempo de procesamiento	500 filas de datos con goroutines	Tiempo menor al procesamiento secuencial equivalente	Reducción de tiempo de procesamiento verificada con métricas	✓ PASS

Tabla 5.Casos de prueba – Prototipo 2: API de envío de correos

ID	Descripción del caso	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-07	Recepción de mensaje desde cola RabbitMQ	Trama enviada desde AS400 al broker	El servicio consume el mensaje de la cola sin pérdida	Mensaje consumido y procesado correctamente	✓ PASS

ID	Descripción del caso	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-08	Parseo de trama AS400	Trama con separadores y campos codificados	Extracción correcta de campos: destinatario, asunto, cuerpo	Todos los campos extraídos con valores correctos	✓ PASS
CP-09	Búsqueda de plantilla en MongoDB	ID de plantilla existente en la base de datos	El servicio retorna la plantilla almacenada	Plantilla recuperada de MongoDB exitosamente	✓ PASS
CP-10	Fallback a archivo TXT en disco	ID de plantilla no existente en MongoDB	El servicio lee el archivo TXT desde la ruta de disco	Archivo TXT leído y cuerpo del correo construido	✓ PASS
CP-11	Almacenamiento de plantilla en MongoDB tras lectura de disco	Plantilla leída desde TXT (no estaba en DB)	La plantilla se guarda en MongoDB para consultas futuras	Plantilla insertada correctamente en la colección	✓ PASS
CP-12	Envío de correo a múltiples destinatarios	Lista de 3 destinatarios válidos en la trama	El correo llega a los 3 destinatarios	Correos enviados y confirmados a todos los destinatarios	✓ PASS
CP-13	CRUD: crear plantilla vía API REST	POST con cuerpo JSON de nueva plantilla	Plantilla creada y almacenada en MongoDB con ID	Plantilla creada, respuesta 201 con ID	✓ PASS

ID	Descripción del caso	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
			generado	asignado	
CP-14	CRUD: actualizar plantilla existente	PUT con ID válido y nuevo contenido	Plantilla actualizada en MongoDB	Registro actualizado; respuesta 200 con datos nuevos	✓ PASS
CP-15	Manejo de destinatario con formato inválido	Dirección de correo malformada en la trama	El servicio rechaza el envío y registra el error	Error capturado; log registrado; RabbitMQ recibe NACK	✓ PASS

Errores encontrados y resolución. Durante las pruebas del Prototipo 2 se identificaron errores en el CRUD de MongoDB: el método de actualización no estaba persistiendo correctamente los cambios en algunos campos de las plantillas. El error se detectó al comparar los registros antes y después de una operación PUT, evidenciando que el documento en la base de datos no reflejaba los nuevos valores. La causa raíz fue un mapeo incorrecto de los campos del cuerpo de la solicitud hacia la estructura Go, lo que ocasionaba que se enviara un objeto parcialmente vacío a la operación de actualización. Una vez identificado, se corrigió el mapeo y se validó nuevamente con los mismos casos de prueba, confirmando el comportamiento esperado. Para el Prototipo 1, los ajustes más relevantes surgieron de la revisión del jefe de área, quien solicitó el control del límite de goroutines concurrentes, funcionalidad que no estaba presente en la versión inicial y que se implementó satisfactoriamente.

6.3.5. Ajustes a proyecto Angular/Node.js

Durante la última semana de la pasantía, del 5 al 17 de diciembre de 2025, se asignó una actividad complementaria consistente en revisar y ajustar un proyecto existente desarrollado por un compañero del área de innovación. Este proyecto estaba construido

con Angular en el frontend y Node.js en el backend, tecnologías distintas al stack principal utilizado durante la pasantía (Go).

Para poder ejecutar y revisar el proyecto fue necesario solicitar la instalación de Node.js en el equipo de trabajo, proceso que tomó parte del tiempo disponible. Una vez configurado el entorno, se realizó un estudio autónomo de los fundamentos de Angular para comprender la arquitectura del proyecto antes de intervenir el código. Tras asimilar la estructura del proyecto, se aplicaron los ajustes funcionales solicitados por el jefe de área, se ejecutaron pruebas de funcionamiento y se realizó la entrega formal del trabajo corregido antes del cierre de la pasantía.

Esta actividad representó un reto adicional al implicar el aprendizaje acelerado de un framework nuevo bajo presión de tiempo, lo que fortaleció la capacidad de adaptación tecnológica y el trabajo con código base ajeno, habilidades fundamentales en el contexto del desarrollo de software profesional.

6.3.6. Documentación técnica de los prototipos

A lo largo de toda la pasantía se elaboró documentación técnica de los desarrollos realizados, siguiendo los estándares del Sistema de Gestión Integrado (SGI) de Sistemas GyG S.A. Esta documentación incluyó el registro formal de los requerimientos de cada prototipo, los análisis previos al desarrollo, los resultados de las pruebas funcionales y la bitácora semanal de actividades.

La Ficha de Objetivos fue elaborada y ajustada en múltiples iteraciones junto con la jefe de área, Linna Sarmiento, para reflejar con precisión el alcance real de cada proyecto. Los formatos institucionales utilizados incluyeron el F-212 (Formato Solicitud Inicial de Teletrabajo) y los documentos de registro de avance propios del SGI. La bitácora de pasantía fue actualizada semana a semana como instrumento de trazabilidad del trabajo realizado.

Toda la documentación fue compartida con el equipo a través de las herramientas colaborativas institucionales (Microsoft Teams y GitLab), garantizando que el conocimiento generado durante la pasantía quedara disponible para el área de innovación más allá de la duración del proyecto.

6.4. Manual de usuario

En esta sección se presenta la documentación de uso de los dos prototipos de software desarrollados durante la pasantía. El manual está dirigido a los usuarios técnicos del área de Gerencia de Innovación de Sistemas GyG S.A. que necesiten ejecutar, configurar o mantener cualquiera de los dos servicios. Por razones de confidencialidad, las rutas y credenciales específicas del entorno de producción deben ser consultadas con el equipo de infraestructura de la empresa.

6.3.7. Manual de usuario Procesador automatizado de archivos Excel.

Descripción general

El Prototipo 1 es una herramienta de línea de comandos desarrollada en Go que automatiza el procesamiento de archivos Excel. Recibe un archivo de entrada con datos tabulares, aplica validaciones fila por fila utilizando goroutines concurrentes y genera un archivo Excel de salida con los resultados consolidados.

Requisitos del sistema

- Go 1.22 o superior instalado en el sistema.
- Acceso de lectura al archivo Excel de entrada.
- Acceso de escritura en la carpeta de destino del archivo de salida.
- Sistema operativo: Windows 10/11, Linux (Ubuntu 20.04+) o macOS 12+.

6.3.8. Configuración inicial

Paso	Instrucción
1	Clonar el repositorio del proyecto desde GitLab: <code>git clone <url-repositorio-prototipo1></code>
2	Acceder al directorio del proyecto: <code>cd prototipo1-excel</code>
3	Descargar las dependencias del proyecto: <code>go mod tidy</code>
4	Verificar que el entorno esté correctamente configurado ejecutando: <code>go build ./...</code> (no debe arrojar errores)

6.3.9. Preparar el archivo Excel de entrada

Paso	Instrucción
------	-------------

Paso	Instrucción
1	El archivo Excel de entrada debe tener una hoja con los datos a procesar a partir de la primera fila.
2	Las columnas deben estar en el orden definido por el equipo de innovación (ver documentación de requerimientos interna).
3	Guardar el archivo en formato .xlsx en una ruta de fácil acceso (por ejemplo: C:/datos/entrada.xlsx en Windows o /home/usuario/datos/entrada.xlsx en Linux).

6.3.10. Ejecutar el programa

Paso	Instrucción
1	Abrir una terminal o símbolo del sistema en la carpeta raíz del proyecto.
2	Ejecutar el programa con los parámetros requeridos: <code>go run main.go -input /ruta/archivo_entrada.xlsx -output /ruta/archivo_salida.xlsx</code>
3	El programa mostrará en consola el progreso del procesamiento: número de filas leídas, goroutines activas y estado de validación.
4	Al finalizar, el archivo de salida estará disponible en la ruta indicada con el parámetro -output.

6.3.11. Interpretar los resultados

Paso	Instrucción
1	Abrir el archivo Excel de salida generado por el programa.
2	Cada fila del archivo de salida corresponde a una fila del archivo de entrada procesada.
3	Las filas con errores de validación estarán marcadas con un indicador en la columna

Paso	Instrucción
	de estado (por ejemplo: 'ERROR' o 'INVÁLIDO').
4	Las filas procesadas correctamente tendrán el indicador 'OK' y los campos calculados o transformados con sus valores finales.

6.3.12. Solución de problemas frecuentes

Paso	Instrucción
P1	Error: 'archivo no encontrado' → Verificar que la ruta especificada en -input sea correcta y que el archivo exista.
P2	Error: 'permiso denegado' → Verificar que el usuario tenga permisos de lectura sobre el archivo de entrada y de escritura sobre la carpeta de destino.
P3	El archivo de salida está vacío → Verificar que el archivo de entrada tenga datos a partir de la primera fila y que el formato de columnas sea el esperado.
P4	El programa termina abruptamente (panic) → Revisar la consola para el mensaje de error detallado y verificar que la versión de Go instalada sea 1.22 o superior.

7. CONCLUSIONES

La pasantía realizada en Sistemas GyG S.A. entre agosto y diciembre de 2025 permitió desarrollar dos prototipos funcionales en Go que respondieron a necesidades reales del área de Gerencia de Innovación, consolidando competencias técnicas y profesionales difíciles de adquirir exclusivamente en el entorno académico.

Los tres objetivos específicos se cumplieron satisfactoriamente. El Prototipo 1 automatizó el procesamiento de archivos Excel mediante programación concurrente con goroutines, reduciendo la intervención manual del equipo. El Prototipo 2 estableció un servicio de envío automatizado de correos integrado con RabbitMQ, MongoDB y AS400, dotando a la empresa de una capa de comunicación desacoplada y extensible sin modificar los sistemas centrales. Ambos desarrollos fueron revisados y aprobados por el jefe de área. Las pruebas funcionales, ejecutadas de forma iterativa a lo largo del proceso, permitieron detectar y corregir errores tempranos, validando el comportamiento esperado en quince casos de prueba formales. La documentación técnica elaborada bajo los estándares del SGI quedó disponible como insumo para el equipo más allá de la duración de la pasantía. Entre los principales retos enfrentados se destacan el aprendizaje acelerado de Go como lenguaje nuevo, los retrasos derivados de dependencias externas como la configuración de RabbitMQ, y la adaptación a restricciones de confidencialidad propias del sector financiero. La actividad complementaria con Angular y Node.js en la última semana reforzó la capacidad de intervenir código ajeno bajo presión de tiempo, habilidad fundamental en entornos de desarrollo profesional.

En términos de aprendizaje, la experiencia confirmó que el desarrollo de software en un contexto empresarial real exige disciplina de documentación, pensamiento concurrente y comprensión de arquitecturas de integración. El contacto con tecnologías como Go, RabbitMQ y MongoDB aplicadas a un problema de negocio concreto en el sector financiero delineó con claridad el perfil técnico hacia el cual orientar el crecimiento profesional.

8. TABLA DE ILUSTRACIONES

Figura 1 <i>Certificado curso Inducción a la Compañía.</i>	17
Figura 2 <i>Certificado curso Inducción SGC.</i>	18
Figura 3 <i>Certificado curso Inducción SGSI y Ciberseguridad.</i>	19
Figura 4 <i>Certificado curso Inducción SGC.</i>	20
Figura 5 <i>Certificado curso Desarrollo Gráfico.</i>	21
Figura 6 <i>Investigación Técnicas de Programación.</i>	22
Figura 7 <i>Investigación IBM.</i>	23
Figura 8 <i>Certificado de finalización del curso Go – Udemy.</i>	24
Figura 9 <i>Diagrama de flujo – Prototipo 1: Procesador de Excel con goroutines.</i> ... 26	
Figura 10 <i>Captura de pantalla – Ejecución del Prototipo 1</i>	27
Figura 11 <i>Captura de pantalla – Ejecución del Prototipo 1</i>	28
Figura 12 <i>Captura de pantalla – Ejecución del Prototipo 1</i>	29
Figura 13 <i>Diagrama Arquitectura general del microservicio de envío automatizado de correos electrónicos.</i>	31
Figura 14 <i>Diagrama Flujos del Prototipo 2</i>	32
Figura 15 <i>Diagrama Interacciones MongoDB.</i>	33
Figura 16 <i>Ejecución Inicial del prototipo 2</i>	34
Figura 17 <i>Prueba del Prototipo 2 - Thunder Client</i>	35
Figura 18 <i>Estado primera ejecución del Prototipo 2</i>	36
Figura 19 <i>Verificación de la ejecución 1 del Prototipo 2</i>	36
Figura 20 <i>Evidencia primera ejecución del Prototipo 2</i>	37
Figura 21 <i>Verificación de la ejecución 2 del Prototipo 2</i>	38
Figura 22 <i>Evidencia primera ejecución del Prototipo 2</i>	38

9. Figuras

Figura 1 <i>Certificado curso Inducción a la Compañía.</i>	17
Figura 2 <i>Certificado curso Inducción SGC.</i>	18
Figura 3 <i>Certificado curso Inducción SGSI y Ciberseguridad.</i>	19
Figura 4 <i>Certificado curso Inducción SGC.</i>	20

Figura 5 <i>Certificado curso Desarrollo Gráfico.</i>	21
Figura 6 <i>Investigación Técnicas de Programación.</i>	22
Figura 7 <i>Investigación IBM.</i>	23
Figura 8 <i>Certificado de finalización del curso Go – Udemy.</i>	24
Figura 9 <i>Diagrama de flujo – Prototipo 1: Procesador de Excel con goroutines.</i> ... 26	
Figura 10 <i>Captura de pantalla – Ejecución del Prototipo 1</i>	27
Figura 11 <i>Captura de pantalla – Ejecución del Prototipo 1</i>	28
Figura 12 <i>Captura de pantalla – Ejecución del Prototipo 1</i>	29
Figura 13 <i>Diagrama Arquitectura general del microservicio de envío automatizado de correos electrónicos.</i>	31
Figura 14 <i>Diagrama Flujos del Prototipo 2</i>	32
Figura 15 <i>Diagrama Interacciones MongoDB</i>	33
Figura 16 <i>Ejecución Inicial del prototipo 2</i>	34
Figura 17 <i>Prueba del Prototipo 2 - Thunder Client</i>	35
Figura 18 <i>Estado primera ejecución del Prototipo 2</i>	36
Figura 19 <i>Verificación de la ejecución 1 del Prototipo 2</i>	36
Figura 20 <i>Evidencia primera ejecución del Prototipo 2</i>	37
Figura 21 <i>Verificación de la ejecución 2 del Prototipo 2</i>	38
Figura 22 <i>Evidencia primera ejecución del Prototipo 2</i>	38

10. Tablas

Tabla 1. Objetivos específicos.....	11
Tabla 2. Listado de actividades realizadas en el proyecto.	12
Tabla 3. Herramientas utilizadas en el desarrollo del proyecto.	14
Tabla 4. Casos de prueba – Prototipo 1: Procesador de Excel con goroutines. ...	39
Tabla 5. Casos de prueba – Prototipo 2: API de envío de correos.....	40

11. Referencias

- Cutajar, J. (2023). *Learn Concurrent Programming with Go*. Shelter Island: Manning Publications.
- Felderer, M., & Ramler, R. (2020). *The Future of Software Quality Assurance*. Cham: Springer Nature.
- GitLab. (2026). *GitLab*. Obtenido de <https://about.gitlab.com/>
- Go. (2026). *Go*. Obtenido de <https://go.dev/>
- GO. (2026). *Go concurrency patterns*. Obtenido de <https://go.dev/blog/pipelines>
- IBM. (04 de 2026). *Modernization tools for IBM i*. IBM Corporation. Obtenido de <https://www.ibm.com/mx-es>
- Kernighan, B. W. (2015). *The Go Programming Language*. Upper Saddle River: Addison-Wesley Professional.
- MongoDB. (11 de 11 de 2025). *MongoDB*. Obtenido de <https://www.mongodb.com/>
- Newman, S. (2021). *Building Microservices: Designing Fine-Grained Systems*. Sebastopol: O'Reilly Media.
- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. USA: McGraw-Hill.
- RabbitMQ. (2026). *RabbitMQ*. Obtenido de <https://www.rabbitmq.com/>
- Sistemas GyG S.A. (29 de 01 de 2026). *sistemasgyg*. Obtenido de Sistemas GyG S.A.: <https://www.sistemasgyg.com.co/inicio.html>
- Sommerville, I. (2011). *Ingeniería del software*. Madrid: Pearson Educación.
- Udemy. (2025). *Udemy*. Obtenido de <https://www.udemy.com/>
- Visual Studio Code. (11 de 2025). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com/>

12. ANEXOS

En esta sección se incluyen los documentos y evidencias complementarias que soportan los desarrollos realizados durante la pasantía. Los anexos están organizados en orden de referencia a lo largo del documento.

Anexo 1. Certificado de finalización del curso Go – Udemey.

Se adjunta el certificado de finalización del curso de programación en Go realizado en la plataforma Udemey entre el 1 de septiembre y el 1 de octubre de 2025. El curso comprendió 22 secciones que cubrieron desde los fundamentos del lenguaje hasta temas avanzados como concurrencia, APIs RESTful con Gin, persistencia con MySQL y GORM, y pruebas unitarias.



Certificado_Udemey_G
O.pdf

Anexo 2. Bitácora completa de la pasantía.

Se adjunta la bitácora semanal completa de la pasantía, que registra de forma detallada las actividades realizadas cada semana desde el 19 de agosto hasta el 17 de diciembre de 2025. Este documento sirvió como instrumento de seguimiento y trazabilidad del trabajo a lo largo de los cuatro meses de ejecución.



BITACORA
PASANTÍA.xlsx

Anexo3. Presentación Técnicas de programación.

Se adjunta la primera presentación que se realizó en la segunda semana la cual fue presentada para el grupo de Innovación de la empresa.



TECNICAS DE
PROGRAMACION.pdf

Anexo3. Presentación IBM i Merlin.

Se adjunta la segunda presentación que se realizó en la segunda semana la cual fue presentada para el grupo de Innovación de la empresa.



PRESENTACION IBM
iMERLIN.pdf