



**Optimización del proceso en la gestión de seguros: integración del aliado Cardif al
SIS**

Maria Camila Ramírez López

Ingeniera de Sistemas

Asesor

Jose Ignacio López Vélez, Ingeniero de Sistemas

Universidad de Antioquia

Facultad de ingeniería

Ingeniería de sistemas

Medellín

2025

Cita	(Ramírez López, 2025)
Referencia	Ramírez López, M. C. (2025). <i>Optimización del proceso en la gestión de seguros: integración del aliado Cardif al SIS</i> [Semestre de industria]. Universidad de Antioquia, Medellín.
Estilo APA 7 (2020)	



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

A quienes han sido mi ejemplo a seguir y mi más grande motivación. A Juan Carlos, que desde el cielo sigue guiando cada uno de mis pasos. Y a Sandra Milena, por creer en mí incluso cuando yo misma dudaba. Este logro es tan suyo como mío.

Agradecimientos

A la Universidad de Antioquia, por formarme no solo a nivel profesional, sino también personal. A Bancolombia, por abrirme las puertas para crecer profesionalmente y poner en práctica todo lo aprendido. A mis amigos, que sin su apoyo, este proceso no hubiera sido igual. Y a todas las personas que, de manera directa o indirecta, contribuyeron a este logro, gracias por ser parte de mi camino.

Tabla de contenido

Resumen	7
Abstract	8
Introducción	9
1 Objetivos	11
1.1 Objetivo general	11
1.2 Objetivos específicos	11
2 Marco teórico	13
2.1 Metodología ágil Scrum	13
2.2 Prácticas de CI/CD	15
2.3 Pruebas	16
2.4 Arquitectura	17
2.5 Negocio	18
3 Metodología	20
4 Resultados	22
4.1 Estudio de requisitos	22
4.2 Aplicación de la metodología	22
4.3 Análisis de resultados	24
5 Conclusiones	29
6 Recomendaciones	30
Referencias	31

Lista de figuras

Figura 1 Scrum framework	14
Figura 2 Diagrama Cardif en el SIS	18
Figura 3 Diagrama Cardif antes del SIS	25
Figura 4 Diagrama Cardif en el SIS	27

Siglas, acrónimos y abreviaturas

SIS	Sistema Integral de Seguros
API REST	Application Programming Interface Representational State Transfer
DoD	Definition of Done

Resumen

El presente informe detalla el desarrollo e implementación de un microservicio de sincronización en el proyecto "Optimización del proceso en la gestión de seguros: integración del aliado Cardif al SIS", realizado como parte de las prácticas académicas. Este proyecto tuvo como objetivo principal integrar a la aseguradora Cardif en el Sistema Integral de Seguros (SIS) de Bancolombia, permitiendo la expedición de pólizas de manera automática y eficiente. A través de la metodología ágil Scrum, se desarrolló un servicio capaz de procesar, actualizar y almacenar la información de transacciones de seguros en tiempo real, mejorando la experiencia de los usuarios, reduciendo errores y optimizando los tiempos de respuesta. El resultado fue una integración exitosa que contribuyó significativamente a la optimización de los procesos del banco, alineándose con los objetivos estratégicos de automatización y mejora continua. Este desarrollo no solo fortalece la capacidad operativa del banco, sino que también impulsa la digitalización de sus servicios, generando valor a los clientes y facilitando una gestión más eficiente de las pólizas.

Palabras clave: seguro, cotización, expedición, póliza, integración.

Abstract

This report details the development and implementation of a synchronization microservice in the project "Optimization of the Insurance Management Process: Integration of the Cardif Ally into the SIS," conducted as part of the academic practices. The main objective of this project was to integrate the insurer Cardif into the bank's Integral Insurance System (SIS), enabling the automatic and efficient issuance of policies. Using the agile Scrum methodology, a service was developed to process, update, and store insurance transaction information in real-time, improving the user experience, reducing errors, and optimizing response times. The result was a successful integration that significantly contributed to optimizing the bank's processes, aligning with strategic goals of automation and continuous improvement. This development not only strengthens the bank's operational capacity but also drives the digitization of its services, adding value to customers and facilitating more efficient policy management.

Keywords: insurance, quotation, issuance, policy, integration.

Introducción

En un entorno donde la agilidad es clave para mantener la competitividad, mejorar los procesos internos se vuelve una necesidad. Bancolombia ofrece una amplia variedad de seguros a sus clientes y cuenta con un grupo de asesores encargados de guiar el proceso de adquisición. Sin embargo, los seguros no son gestionados directamente por el banco, este se apoya en un conjunto de aliados aseguradores.

El presente informe documenta el desarrollo e implementación de un microservicio de sincronización como parte de un proyecto llevado a cabo en el marco de las prácticas académicas. Este proyecto tuvo como propósito principal integrar a Cardif, aliado asegurador del banco, en el Sistema Integral de Seguros (SIS). Previamente, los procesos de cotización y expedición de seguros con este aliado se realizaban de manera manual, lo que ocasionaba retrasos, errores y una experiencia poco eficiente para los usuarios y los asesores.

El microservicio desarrollado procesa en tiempo real todas las transacciones resultantes del flujo de venta de un seguro a través del aplicativo SIS. Por medio de dichas transacciones se almacenan las ventas exitosas, los intentos de compra y las ventas fallidas. Una vez se procesa una transacción, se sincroniza toda la información en la base de datos para guardar registro de ello.

Uno de los puntos clave en el desarrollo de este microservicio fue el asegurar la información proveniente por parte del aliado, para conseguirlo se utilizaron buenas prácticas de desarrollo siguiendo los últimos lineamientos de seguridad del banco.

Dentro de las metodologías para la gestión de proyectos, el equipo, basándose en la complejidad de la iniciativa, optó por elegir la metodología ágil Scrum, la cual permite un desarrollo iterativo, generando entregas de valor constantes en cortos periodos de tiempo y alineadas a las necesidades cambiantes del cliente.

En efecto, se logró automatizar el proceso de gestión de seguros, mejorando la eficiencia operativa del banco y optimizando los tiempos de respuesta para los asesores y clientes. Además, dicha integración apalancó la digitalidad en la venta de seguros jubilandos un proceso donde el

uso de papelería era fundamental. Este informe destaca los conceptos teóricos aplicados, los resultados obtenidos y el impacto que tuvo la solución implementada.

1 Objetivos

1.1 Objetivo general

Integrar a Cardif en el Sistema Integral de Seguros (SIS) de Bancolombia, mediante el desarrollo y despliegue de un servicio de sincronización que apalanque la automatización del proceso de expedición de seguros, mejorando la eficiencia operativa y reduciendo los tiempos de respuesta antes de enero de 2025.

1.2 Objetivos específicos

- Documentar el proceso actual de cotización y expedición de seguros mediante un diagrama de flujo, el cual permita identificar claramente las necesidades y problemáticas, facilitando la detección de los principales puntos de mejora para optimizar el flujo de trabajo.
- Planificar y organizar el desarrollo del proyecto utilizando la metodología Scrum, estableciendo un backlog de tareas y un cronograma de sprints de dos semanas, asegurando la claridad de los entregables y la alineación con el equipo.
- Diagramar el proceso de venta de seguros de Cardif por medio del SIS, asegurando que toda la información quede registrada en la biblioteca digital del banco, con el objetivo de que todo el equipo tenga acceso a los aspectos relevantes acerca de la implementación de esta iniciativa.
- Implementar la lógica de negocio en el servicio de sincronización que procese la información recibida para actualizar los registros de cotización e insertar los registros de expedición en la base de datos, asegurando que cumpla con los estándares de calidad y seguridad establecidos por el banco.
- Integrar y realizar pruebas del servicio en el entorno de pruebas del SIS, asegurando que el proceso de sincronización funcione de acuerdo a los requisitos en un 100% de los casos probados, y documentar cualquier incidencia para su corrección.
- Desplegar el servicio en el entorno de producción antes del 6 de diciembre y asegurar una transición sin interrupciones para los usuarios del SIS, alcanzando una tasa de éxito de implementación del 99%.

- Acompañar la etapa de estabilización durante las semanas posteriores al despliegue del servicio en el entorno de producción, realizando un monitoreo constante del estado de los servicios, implementando mejoras o correcciones necesarias para asegurar que el aplicativo funcione de manera óptima y esté alineado con las expectativas del cliente.
- Medir la eficiencia de la solución por medio de la comparación de los tiempos empleados por parte de los asesores en la experiencia de venta de seguros de Cardif por medio del SIS, esto con el objetivo de obtener al menos un 15% de reducción en los tiempos durante las 2 semanas posteriores al despliegue en producción.

2 Marco teórico

En el contexto de este proyecto, es importante entender algunos conceptos, para ello se tienen en cuenta las siguientes definiciones:

2.1 Metodología ágil Scrum

Scrum es un marco de trabajo ágil que permite gestionar proyectos de manera iterativa e incremental, diseñado para maximizar la adaptabilidad y responder de manera eficiente a las necesidades cambiantes del cliente.

El principio clave de Scrum es el enfoque en “inspeccionar y adaptar”, lo cual permite al equipo reflexionar sobre su progreso, identificar oportunidades de mejora y ajustar tanto el producto como los procesos en función de los aprendizajes adquiridos. Esta metodología promueve la innovación, la flexibilidad y la transparencia durante todo el desarrollo.

Uno de los conceptos base de esta metodología es el Sprint, el cual es un período de tiempo definido en el que el equipo se enfoca en completar un conjunto de tareas seleccionadas. Normalmente, los Sprints tienen una duración fija entre 1 y 4 semanas. Durante el Sprint, los elementos priorizados no se modifican, y el equipo trabaja para cumplir con los objetivos establecidos. Al final del Sprint, se revisa y evalúa el trabajo realizado para identificar mejoras futuras.

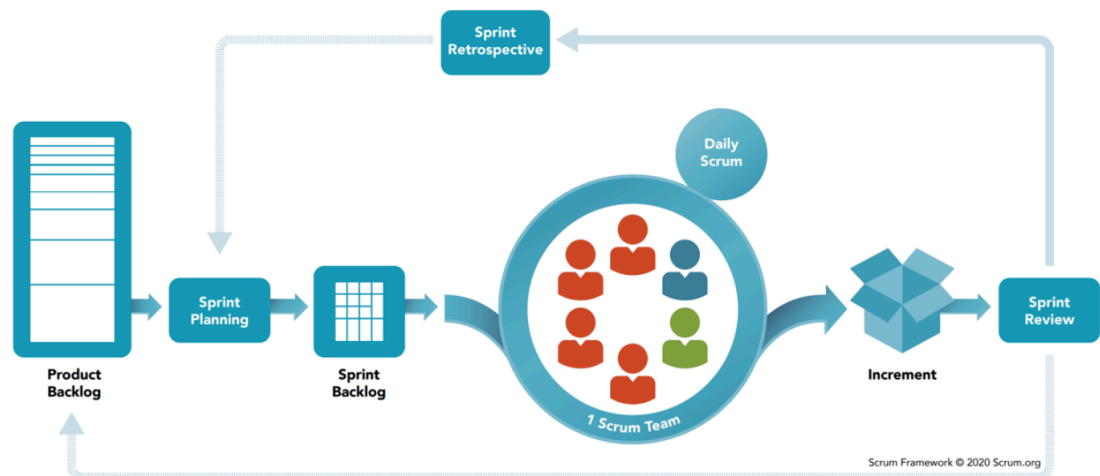
Además, el equipo cuenta con un Backlog, que es una lista priorizada de tareas o requisitos necesarios para el desarrollo de un producto. En Scrum, este se divide en Product Backlog, que incluye todas las necesidades del proyecto, y Sprint Backlog, que contiene los elementos seleccionados para completarse en un Sprint.

Dentro de Scrum, existen eventos o ceremonias principales que estructuran y organizan el trabajo del equipo:

- **Sprint Planning:** Al inicio de cada Sprint, se realiza una sesión de planificación en la que el equipo selecciona, desde el backlog del producto, los elementos que serán trabajados durante ese ciclo. Durante este evento, se definen los objetivos del Sprint y se establece un plan para cumplirlos, asegurándose de que todos los miembros del equipo comprendan claramente sus responsabilidades.

- Daily Scrum: Es una breve reunión de 15 minutos que se lleva a cabo todos los días del Sprint. Durante esta reunión, cada miembro del equipo responde a tres preguntas clave: ¿Qué hice ayer? ¿Qué haré hoy? ¿Hay algo que obstaculice mi progreso? Este evento fomenta la comunicación, la colaboración y permite identificar problemas de forma temprana.
- Sprint Review: Este evento se realiza al final del Sprint y tiene como objetivo presentar el trabajo completado a los interesados. Es una oportunidad para obtener retroalimentación valiosa que se utilizará para mejorar en los siguientes ciclos. Durante esta reunión, el equipo muestra el incremento del producto desarrollado y discute posibles adaptaciones al backlog en función de las necesidades detectadas.
- Sprint Retrospective: Al cierre de cada Sprint, el equipo reflexiona sobre su desempeño y analiza qué funcionó bien, qué no funcionó y qué podría mejorarse en los próximos ciclos. Este evento busca optimizar los procesos del equipo y reforzar la colaboración entre sus miembros.

Figura 1
Scrum framework



Nota. Fuente <https://www.scrum.org/resources/blog/que-es-scrum>

Por otra parte, dentro de la metodología existen 3 roles principales, cada uno con responsabilidades específicas que contribuyen al éxito del proyecto:

- **Product Owner:** Es responsable de identificar y priorizar las funcionalidades del producto, decidir qué elementos se incluirán en el próximo Sprint y ajustar continuamente las prioridades según las necesidades del negocio. El Product Owner asume la responsabilidad de las pérdidas y ganancias del producto, especialmente si este es comercial.
- **Equipo de desarrollo:** Es el encargado de construir el producto que se entregará al cliente, como una aplicación o un sitio web. El equipo es multifuncional, lo que significa que cuenta con todas las habilidades necesarias para entregar un producto funcional al final de cada Sprint, y es autoorganizado, con autonomía para gestionar su trabajo sin la dirección de un jefe de equipo o proyecto.
- **Scrum Master:** Actúa como guía y facilitador del equipo, asegurándose de que todos comprendan y apliquen el marco de trabajo Scrum de manera efectiva. Su objetivo principal es ayudar al equipo a maximizar el valor entregado al negocio y eliminar cualquier impedimento que pueda surgir durante el desarrollo.

Scrum pone un énfasis particular en la entrega constante de valor, priorizando la satisfacción del cliente y la adaptabilidad frente a cambios inesperados. Su filosofía iterativa y el enfoque en la mejora continua han convertido este marco de trabajo en uno de los más utilizados en la gestión de proyectos modernos.

2.2 Prácticas de CI/CD

CI/CD (Integración Continua y Despliegue Continuo) es una práctica de desarrollo que combina la automatización tanto de la integración como del despliegue del software, optimizando el ciclo de vida de desarrollo.

La Integración Continua (CI) consiste en integrar y fusionar los cambios de código de manera frecuente, lo que mejora la calidad del software y aumenta la productividad del equipo. Incluye la construcción y pruebas automáticas del software, garantizando que cada integración sea estable y funcional.

El Despliegue Continuo (CD) va un paso más allá al automatizar también el proceso de despliegue, enviando cada cambio directamente a los entornos de producción, sin intervención manual. Esto asegura una entrega constante y confiable de actualizaciones.

Los ambientes son los diferentes entornos en los que se desarrolla y opera un software. El ambiente de desarrollo es donde los desarrolladores crean, prueban y ajustan el código antes de integrarlo. Luego, el ambiente de certificación es el espacio intermedio donde el software se somete a pruebas exhaustivas en condiciones controladas, asegurando que cumpla con los requisitos antes de ser lanzado. Finalmente, el ambiente de producción es el entorno donde el software se despliega para ser utilizado por los usuarios finales, en condiciones reales de operación.

Por otra parte, un repositorio se encarga de centralizar el almacenamiento y gestión del código fuente, permitiendo el control de versiones y la colaboración entre desarrolladores. Los pipelines son procesos automatizados clave en el desarrollo continuo. El pipeline de build compila y construye el software, ejecutando pruebas unitarias y generando artefactos listos para la entrega. Y el pipeline de release gestiona el despliegue del software desde la compilación hasta su implementación final en los diferentes ambientes, asegurando la validación y revisión del sistema antes de su publicación.

2.3 Pruebas

Las pruebas son el proceso de evaluar y verificar que un sistema o aplicación cumpla con los requisitos definidos, funcione correctamente y esté libre de defectos. Este proceso busca garantizar la calidad y confiabilidad del software mediante la ejecución de distintos tipos de pruebas. Entre las pruebas realizadas a lo largo del desarrollo de este proyecto se encuentran:

- **Pruebas unitarias:** Son pruebas diseñadas para validar el funcionamiento de componentes individuales del software, como funciones o métodos, de forma aislada. Permiten identificar errores en las unidades más pequeñas del código antes de integrarlas al sistema.
- **Pruebas de aceptación:** Validan que el sistema completo cumpla con los requisitos del usuario y satisfaga las expectativas del cliente. Estas pruebas suelen realizarse al final del desarrollo para determinar si el software está listo para su implementación.

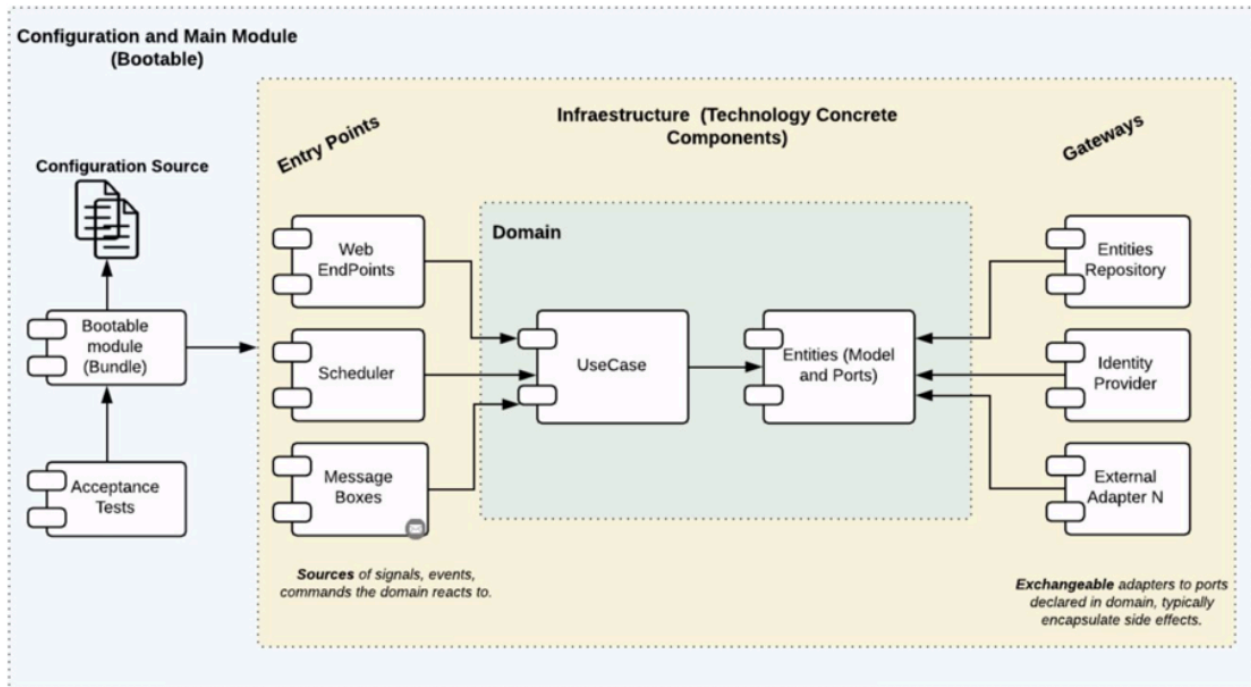
- Pruebas end-to-end (E2E): Evalúan el sistema completo desde el principio hasta el final, simulando el flujo real de uso de un usuario. Estas pruebas verifican la interacción entre diferentes componentes y sistemas para asegurar que trabajen correctamente de manera integrada.
- Pruebas de performance: Miden el rendimiento del software bajo diversas condiciones, evaluando aspectos como la velocidad, capacidad de respuesta y estabilidad cuando se enfrenta a diferentes cargas de trabajo.
- Mutation tests: Son pruebas diseñadas para evaluar la calidad del código y la efectividad de las pruebas existentes. Se introducen cambios deliberados en el código (mutaciones) para verificar si las pruebas son capaces de detectarlos.

2.4 Arquitectura

La solución planteada ante la situación problemática que se presentaba en el banco fue realizar el desarrollo de un microservicio de sincronización. Un microservicio es un estilo arquitectónico que consiste en desarrollar una aplicación como un conjunto de servicios independientes, escalables y colaborativos. Cada servicio es autónomo y está diseñado para adaptarse y evolucionar dentro de ecosistemas complejos.

Los microservicios surgieron para superar las limitaciones de los sistemas monolíticos, que concentran todos los servicios en una única unidad lógica y comparten los mismos recursos computacionales, dificultando su escalabilidad y evolución. Este enfoque moderno facilita la migración hacia infraestructuras en la nube y permite a las organizaciones construir sistemas más ágiles y adaptativos.

Hablando de la arquitectura empleada para el desarrollo de este microservicio, se utilizó Clean Architecture. Este es un patrón de diseño que organiza el desarrollo de software en capas bien definidas, donde la lógica del negocio, las reglas y las dependencias se mantienen separadas de las capas de entrada (frontend) y salida (salida externa como APIs). Su objetivo es mejorar la mantenibilidad, la escalabilidad y la independencia entre las diferentes partes del sistema, promoviendo un desarrollo limpio y flexible. En el siguiente diagrama se puede apreciar de mejor manera cada una de las capas que lo componen:

Figura 2*Diagrama Clean Architecture*

Nota. Fuente <https://bancolombia.github.io/scaffold-clean-architecture/docs/intro/>

Para la implementación de esta iniciativa, se requirió el desarrollo de un API REST (Application Programming Interface Representational State Transfer) quien se encarga de recibir las peticiones enviadas por parte del aliado y hacerlas llegar al microservicio de sincronización, esto haciendo uso del protocolo HTTP. Un API es un conjunto de reglas y herramientas que permite que diferentes programas se comuniquen entre sí. Facilita el acceso y la interacción con las funcionalidades y datos de un sistema, aplicación o servicio, permitiendo que las aplicaciones puedan compartir recursos y realizar operaciones de manera eficiente.

2.5 Negocio:

Hablando un poco más del negocio, es importante entender lo que es un seguro y cómo es el proceso de adquisición.

Un seguro es un contrato mediante el cual una persona (el asegurado) paga una suma de dinero (prima) a una compañía de seguros (el asegurador), a cambio de la promesa de esta última

de indemnizar al asegurado en caso de que ocurra un evento específico que cause pérdidas o daños (siniestro).

El seguro puede ser de diferentes tipos, como seguros de vida, de salud, de vehículos, de hogar, cada uno diseñado para cubrir riesgos específicos. Su finalidad principal es proporcionar tranquilidad a los asegurados al permitirles transferir el riesgo financiero de la ocurrencia de eventos imprevistos a la aseguradora.

Para adquirir un seguro, la persona interesada deberá primero identificar una necesidad de protección financiera frente a un riesgo. Una vez identificada dicha necesidad, la persona se pondrá en contacto con alguna entidad aseguradora para proceder a la cotización del seguro. La cotización es ese proceso mediante el cual la aseguradora calcula y ofrece un precio determinado al cliente, durante este proceso entran en juego diversos factores como el tipo de cobertura, el perfil del asegurado, el riesgo asociado al seguro. En esta se incluyen detalles como el valor de la prima, los beneficios y las condiciones, información que el cliente deberá tener en cuenta a la hora de evaluar las diferentes opciones.

Una vez aceptada la oferta, la aseguradora procede a la expedición de la póliza. La expedición se refiere al acto de emitir el contrato que detalla los derechos y deberes de ambas partes, con esta se marca el inicio formal de la cobertura, es decir que, a partir de ese momento, el asegurado está protegido bajo los términos y condiciones acordados en la póliza.

Posteriormente, el cliente realiza el pago de la prima, la cual puede darse una única vez o de manera periódica. En caso de ocurrir el siniestro, el asegurado deberá notificar a la aseguradora y seguir el procedimiento adecuado para la reclamación.

Existen múltiples empresas que se dedican a ofrecer seguros a sus clientes, entre ellas, Cardif, una compañía francesa que comercializa productos y servicios de ahorros y protección. Se encuentran presentes en Colombia desde el año 2009, durante estos años se han consolidado como una empresa líder en el sector de bancaseguros y actualmente prestan sus servicios a millones de clientes, siendo hoy en día uno de los principales aliados de Bancolombia para la venta de seguros.

Es importante, además, tener una noción básica de lo que es el SIS. El Sistema Integral de Seguros, es el aplicativo web por medio del cual los asesores del banco cotizan y expiden seguros para los clientes en tiempo real. Adicionalmente, se puede generar reportes de las pólizas cotizadas y expedidas por aliado en x periodo de tiempo.

3 Metodología

Para el desarrollo de este proyecto, se utilizó la metodología ágil Scrum, la cual facilita la ejecución de proyectos de software mediante ciclos iterativos e incrementales. El proyecto se organizó en sprints de dos semanas. Cada sprint comenzó con una reunión de planificación (Planning) en la que se definieron las tareas a realizar, basándose en los objetivos generales y específicos del proyecto. Además, se llevó a cabo una reunión diaria con el equipo (Daily) con el objetivo de socializar los avances e impedimentos en las tareas desarrolladas por cada integrante. Cada sprint finalizó con una reunión de revisión (Review), en la cual se validó el cumplimiento de los objetivos propuestos para este. Además de la retrospectiva (Retrospective), reunión en la cual se reflexiona acerca del desempeño propio y del equipo y se identifican oportunidades de mejora.

Con la finalidad de alcanzar los objetivos específicos, se plantearon una serie de pasos a seguir, los cuales permitieron llevar a cabo el proyecto de manera exitosa:

- **Análisis y diseño:** Durante los primeros sprints, se recopilaban las necesidades del cliente, definiendo los requerimientos funcionales y no funcionales. Además, se diseñó la arquitectura y los diagramas que representan el funcionamiento esperado del microservicio en el contexto del SIS.
- **Desarrollo iterativo:** El servicio se desarrolló por medio de entregas parciales en cada sprint. Esto permitió revisar y ajustar el código en función de la retroalimentación recibida asegurando que se cumpliera con los requisitos definidos.
- **Pruebas y validación:** A medida que se avanzó con los desarrollos, se realizaron pruebas, tanto automatizadas como manuales, para garantizar que se abarcaran las funcionalidades esperadas y se cumplieran los estándares de calidad del banco.
- **Integración continua:** Se implementó la práctica de integración continua, lo que significa que cada nueva funcionalidad fue integrada inmediatamente en el entorno de desarrollo y se realizaron pruebas automáticas para verificar su correcto funcionamiento. Dejando finalmente el proyecto desplegado en el entorno de producción.

- Documentación: A medida que el desarrollo fue avanzando, se generó la documentación técnica necesaria para garantizar que el código y su funcionamiento fueran comprensibles para cualquier miembro del equipo, actual o futuro.

4 Resultados

El banco cuenta con un aplicativo llamado “Sistema Integral de Seguros” (SIS), el cual permite a los asesores gestionar pólizas de manera automatizada. Anteriormente, esto sólo era posible para uno de sus aliados aseguradores. Sin embargo, el proceso para otros aliados, como Cardif, seguía siendo manual. Esto implicaba demoras significativas en la cotización y expedición, además de una alta probabilidad de errores en la gestión de la información. Esta situación generaba insatisfacción en los clientes y limitaba la capacidad del banco para ofrecer un servicio competitivo. Se identificó la necesidad de integrar al aliado Cardif en el SIS para unificar los procesos, mejorar la experiencia del usuario y disminuir significativamente los errores operativos.

Para ello, hubo que realizar modificaciones en el backend y frontend de varios microservicios ya existentes, además de desarrollar un nuevo microservicio de sincronización encargado de actualizar el detalle de las cotizaciones y de insertar o actualizar los registros de las nuevas expediciones en la base de datos.

El backend es la parte de una aplicación que maneja la lógica del servidor, la gestión de bases de datos, la seguridad y las operaciones del lado del servidor, procesando los datos y aplicando las reglas de negocio. Por otro lado, el frontend es la parte visible y de interacción con el usuario, que incluye la interfaz gráfica, los formularios, los botones y los elementos visuales, encargándose de recibir las solicitudes del usuario, enviarlas al backend y mostrar los resultados en la pantalla.

4.1 Estudio de requisitos

Entre los requisitos funcionales del proyecto, se encontraba la gestión de peticiones enviadas por parte del aliado, la actualización de la información en tiempo real (información personal del cliente, del medio de pago, del producto) y el manejo de posibles errores como la recepción de información incompleta o incorrecta. Por otra parte, los principales requisitos no funcionales a tener en cuenta fueron la seguridad de la información, la escalabilidad y el tiempo de respuesta del microservicio.

4.2 Aplicación de la metodología

El proyecto tuvo una duración total de 14 sprints, a lo largo de los cuales se fueron cumpliendo diversas tareas que permitieron llevar a cabo lo propuesto.

- Sprint 1 (3 julio - 16 julio): A pesar de que este sprint tuvo como fecha de inicio el 3 de julio, las prácticas comenzaron el día 8 de julio, por ende se dio inicio a la planeación del proyecto a mitad del sprint. Se desglosaron todas las actividades a realizar estimando su

fecha de entrega y detallando las dependencias con otras áreas con el fin de evitar retrasos en medio de la ejecución.

- Sprint 2 (17 julio - 30 julio): Se llevó a cabo la exploración, documentación y diagramación del proceso antiguo, además de la reunión con la persona encargada de definir la arquitectura de toda la iniciativa.
- Sprint 3 (31 julio - 13 agosto): Se crearon los nuevos recursos (namespace, repositorio, pipeline de build). Se implementó la estructura de clean architecture en el proyecto. Se realizó la maquetación inicial, creando carpetas, clases e interfaces.
- Sprint 4 (14 agosto - 27 agosto): Se iniciaron los desarrollos de las funcionalidades propuestas. Se realizó una pequeña modificación en una tabla previamente existente a nivel de base de datos, esto con el fin de homologar cierta información entre el aliado y el banco, llevando el cambio por los diferentes ambientes (desarrollo, certificación y producción).
- Sprint 5 (28 agosto - 10 septiembre): Se continuó con los desarrollos necesarios. Se creó el pipeline de release para ambiente de desarrollo, dejándolo configurado con las respectivas tareas y variables. Se llevó a cabo una reunión entre el equipo y el analista de ciberseguridad con el fin de solicitar los lineamientos que permitieran asegurar el manejo de la información de los clientes en todo momento (cifrado de la información que viaja entre el aliado y el banco, conexión segura a la base de datos, etc).
- Sprint 6 (11 septiembre - 24 septiembre): Se recibieron los lineamientos necesarios por parte del área de ciberseguridad, se implementaron en el microservicio y se hizo entrega de la evidencia al analista. Se finalizaron los desarrollos, realizando las respectivas pruebas manuales y corrigiendo los errores encontrados. Se desarrollaron las pruebas unitarias, ejecutando a su vez mutation tests con el fin de garantizar la calidad de las pruebas.
- Sprint 7 (25 septiembre - 8 octubre): Se realizaron y ejecutaron de manera exitosa las pruebas de aceptación, permitiendo así el despliegue en ambiente de desarrollo y probando nuevamente de manera manual el correcto funcionamiento del servicio.
- Sprint 8 (9 octubre - 22 octubre): se diligenció la matriz de riesgos y el *test plan*. También se desarrollaron las pruebas de performance y pruebas e2e, se creó el pipeline de release para ambiente de certificación y posteriormente se realizó el despliegue.
- Sprint 9 (23 octubre - 5 noviembre): Se realizaron las pruebas de integración con la API por parte del equipo y las pruebas exploratorias por parte de la Product Owner y el equipo funcional, quienes son los encargados de verificar que todo el flujo tanto de la cotización como de la expedición se realice de manera correcta. Para ello se cotizaron varios seguros, para algunos se finalizó el proceso hasta la expedición y para otros se desistió de la venta, posteriormente se verificó a nivel de base de datos la correcta actualización y almacenamiento de los datos. Se simularon dichos escenarios con el fin de poner a prueba los diferentes casos que se presentan en un entorno real y de verificar el correcto funcionamiento del microservicio.
- Sprint 10 (6 noviembre - 19 noviembre): Se llevaron a cabo las pruebas en conjunto con el aliado de manera exitosa. Dichas pruebas comprendieron el envío de varias peticiones con diferentes características por parte del aliado (ventas exitosas, desistidas, diferentes

medios de pago, diferentes tipos de seguro), esto con el fin de abarcar la mayor cantidad posible de escenarios. Además, se realizó una modificación a nivel de código por solicitud de la Product Owner y se ajustaron las pruebas unitarias.

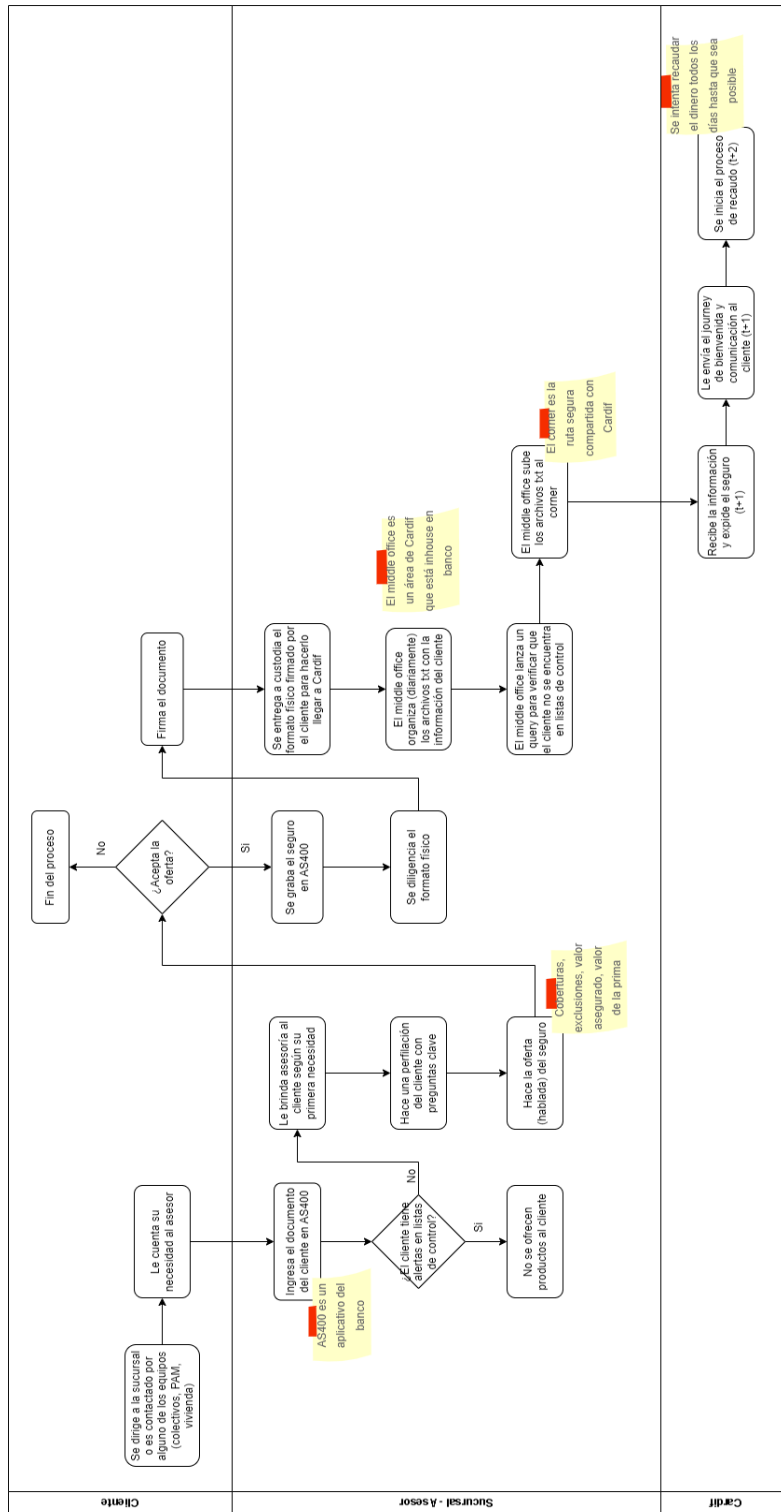
- Sprint 11 (20 noviembre - 3 diciembre): Se llevaron los cambios mencionados a ambientes de desarrollo y certificación. Se creó el DoD (definition of done, conjunto de criterios que determinan cuando una tarea está completamente terminada) y el pipeline de release para producción llevando a cabo también el despliegue en dicho ambiente y realizando pruebas manuales en conjunto con el equipo técnico, funcional y Product Owner.
- Sprint 12 (4 diciembre - 17 diciembre): Etapa de estabilización del servicio (comprende la resolución de los posibles incidentes que se puedan reportar por parte de los asesores con el uso del servicio). Se realizó de manera detallada toda la documentación técnica necesaria, dejándola almacenada en la biblioteca digital del banco.
- Sprint 13 (18 diciembre - 31 diciembre): Etapa de estabilización del servicio. Se realizó el diagrama de flujo del nuevo proceso.
- Sprint 14 (1 enero - 14 enero): Se llevó a cabo la entrega de los desarrollos realizados al equipo de soporte, quienes serán los encargados de continuar dando solución a los posibles incidentes que se presenten con el servicio. Los posibles incidentes serán registrados por parte de los asesores en una herramienta del banco y asignados al equipo de soporte de seguros, posteriormente algún integrante de dicho equipo le dará solución al incidente reportado o de ser necesario se contactará con el asesor para revisar el caso. Finalmente, en conjunto con la Product Owner, se sacaron las eficiencias generadas por la iniciativa en términos de tiempo.

4.3 Análisis de resultados

El siguiente diagrama representa el flujo que se seguía anteriormente para la venta de un seguro con el aliado Cardif. El proceso requería del uso de un formato físico que el asesor debía diligenciar con todos los datos del cliente y era allí donde se presentaba una gran cantidad de errores, lo que ocasionaba el rechazo de la expedición de pólizas por parte del aliado debido a datos erróneos. La validación del cliente en listas de control se debía realizar dos veces, una por parte del asesor y otra por parte del middle office (área de Cardif que se encuentra inhouse en el banco). Adicionalmente, el mensaje de bienvenida y comunicación se enviaba al cliente un día después de realizado el proceso de la expedición.

Figura 3

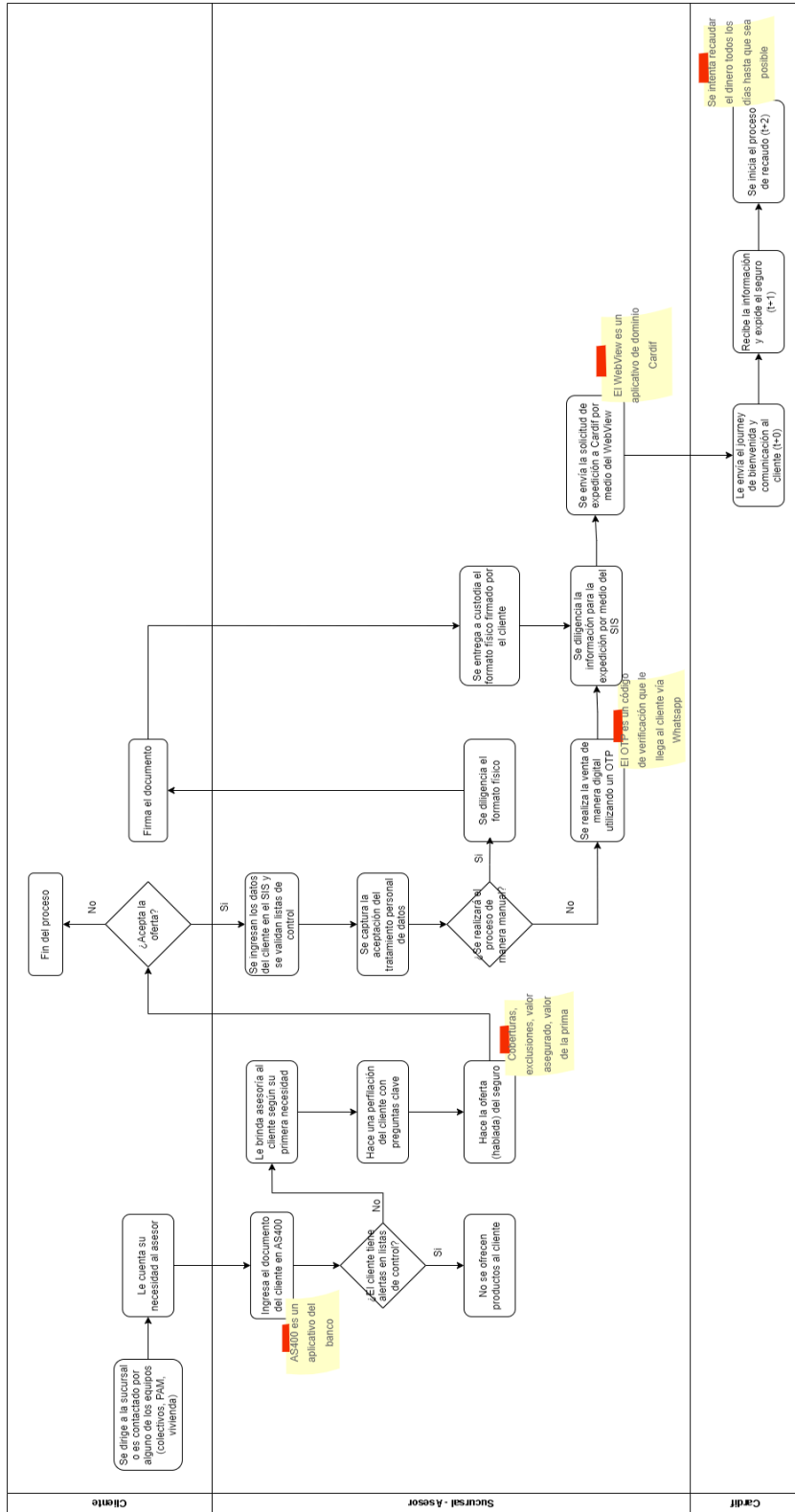
Diagrama Cardif antes del SIS



En el siguiente diagrama de flujo se puede observar el paso a paso del proceso actual de la venta de un seguro con Cardif, después de haber integrado a dicho aliado en el SIS. Actualmente, el asesor puede decidir si desea realizar el proceso directamente de manera digital o si desea registrar la información de manera manual y posteriormente ingresarla de manera digital, teniendo así una alternativa en caso de presentarse intermitencias en el sistema integral de seguros. Sin embargo, independientemente de la manera en la que el asesor decida hacerlo, el tiempo se ve reducido respecto al flujo anterior ya que con la integración de Cardif al SIS, se optimizaron algunos pasos en el proceso que anteriormente se realizaban por medio de otro aplicativo del banco que es un poco más tardado. El proceso digital ahora es mucho más sencillo y elimina la posibilidad de errores en el diligenciamiento de la información, reduciendo así en gran medida la cantidad de pólizas rechazadas por parte del aliado. La validación del cliente en listas de control se realiza una primera vez por parte del asesor y posteriormente se hace de manera automática en el flujo de venta del SIS. Ahora, el mensaje de bienvenida y comunicación se envía al cliente de manera inmediata.

Figura 4

Diagrama Cardif en el SIS



Entre los principales resultados obtenidos gracias a esta integración, se destaca la reducción del tiempo que se toma el asesor en la venta de una póliza (cotización y expedición). Anteriormente, un asesor se tardaba en promedio un tiempo de 17.94 minutos en este proceso. Actualmente, si el asesor desea realizar el proceso de manera manual, se tarda en promedio 14.44 minutos, y si lo realiza de manera digital le toma 12.44 minutos en promedio. Obteniendo así una reducción en el tiempo de venta de un seguro de entre 19.5% y 30.7% aproximadamente.

5 Conclusiones

La integración de Cardif en el Sistema Integral de Seguros (SIS) de Bancolombia ha logrado cumplir los objetivos planteados de manera efectiva, generando mejoras significativas en la eficiencia operativa del proceso de cotización y expedición de seguros. A través de la documentación detallada del flujo inicial y la identificación de los puntos críticos del proceso, se estableció una base sólida para diseñar una solución que abordara las principales problemáticas, como la alta tasa de errores en el diligenciamiento manual y los tiempos prolongados de respuesta. La planificación bajo la metodología Scrum permitió un desarrollo organizado, asegurando que cada sprint estuviera alineado con los requerimientos del proyecto y las expectativas del cliente.

La implementación del servicio de sincronización logró automatizar el flujo de datos entre Cardif y el SIS, asegurando que la información fuera procesada y registrada de manera eficiente y confiable. Esto no solo redujo los errores en la expedición de pólizas, sino que también optimizó los tiempos en el proceso de venta. Los resultados obtenidos demuestran una mejora notable: el tiempo promedio para completar una venta disminuyó de 17.94 minutos a 14.44 minutos en el proceso manual y a 12.44 minutos en el digital, representando una reducción de entre 19.5% y 30.7%. Estas mejoras reflejan el impacto positivo de la solución tanto para los asesores como para los clientes, quienes ahora disfrutan de un proceso más ágil y confiable.

En términos generales, este proyecto no solo alcanzó su objetivo principal antes del plazo establecido, sino que también demostró ser un ejemplo de cómo la integración tecnológica y la automatización pueden transformar procesos manuales, complejos y propensos a errores en flujos eficientes y de alto valor agregado. La solución entregada no solo responde a las necesidades actuales del banco y de sus aliados, sino que también sienta las bases para futuras expansiones e innovaciones en el sistema.

6 Recomendaciones

En el futuro, se plantea la posibilidad de expandir la integración del SIS para incluir nuevos aliados aseguradores, lo que permitiría aumentar la eficiencia operativa al centralizar aún más los procesos de cotización y expedición. Esto contribuiría a fortalecer la posición del banco como un intermediario clave en la oferta de seguros, proporcionando a los clientes una experiencia más completa y ágil. Asimismo, es importante considerar la mejora continua del monitoreo del microservicio de sincronización, implementando herramientas más avanzadas para la supervisión en tiempo real, la detección temprana de fallos y la generación de alertas automáticas. Estas mejoras no solo garantizarían un funcionamiento más confiable del servicio, sino que también reducirían los tiempos de respuesta ante posibles incidencias, asegurando la calidad y continuidad del sistema a largo plazo.

Referencias

- Nosotros. (s.f). BNP PARIBAS CARDIF. <https://bnpparibascardif.com.co/nosotros>
- Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices.* (2017). IEEE Journals & Magazine | IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/7884954>
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). *The Scrum primer.* https://goodagile.com/scrumprimer/scrumprimer_es.pdf
- Laura Mamani, C. A. (2023). *Pruebas de Software para Microservicios.* Redalyc.org. <https://www.redalyc.org/articulo.oa?id=673874721011>
- Mamani Rodríguez, Z., Del Pino Rodríguez, L., & Gonzales Suarez, J. C. (2020). *Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua.* <https://www.redalyc.org/journal/816/81665362014/81665362014.pdf>
- Pérez Torres, J.L. (1986). *Teoría general del seguro.* https://www.researchgate.net/profile/Jose-Luis-Perez-Torres/publication/40942409_Conociendo_el_seguro_teoría_general_del_seguro/links/56dc897e08aeb4638c0324a/Conociendo-el-seguro-teoria-general-del-seguro.pdf
- ¿Qué es Scrum?* (s. f.). Scrum.org. <https://www.scrum.org/resources/blog/que-es-scrum>
- Scaffold Clean Architecture.* (s. f.). <https://bancolombia.github.io/scaffold-clean-architecture/docs/intro/>