

Pontificia Universidad
Católica del Ecuador

FACULTAD DE INGENIERÍA
COORDINACIÓN DE POSGRADO



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA

Trabajo de Titulación como requisito previo para la obtención del título de
Magíster en Tecnologías de Información mención Gestión y Administración
de TI

**GESTIÓN DE UN PROYECTO ÁGIL PARA LA INTEGRACIÓN DE
ASISTENTE VIRTUAL CON MICROSERVICIOS**

Autor: Marco Vinicio Luna Aizaga

Director: Msc Leonardo Napoleón Arévalo Rivera

Quito, Abril 2024.

ÍNDICE DE TABLAS

Tabla 1 Ventajas y Desventajas de la Metodología en Cascada	19
Tabla 2 Principales metodologías ágiles	23
Tabla 3 Principales características de SCRUM	25
Tabla 4 Tecnología de Machine Learning y Cognitive Services	31
Tabla 5 Asistentes virtuales más utilizados	31
Tabla 6 Componentes de DialogFlow.....	34
Tabla 7 Evolución de las arquitecturas de sistemas.....	38
Tabla 8 Customer Journey Map de los entrevistados	46
Tabla 9 Operacionalización de Variables	50
Tabla 10 Puntos más destacados de las entrevistas	52
Tabla 11 Principales ideas seleccionadas	54
Tabla 12 Historias de usuario identificadas	56
Tabla 13 Planificación de los Sprints.....	59
Tabla 14 Estructura del Sprint	59
Tabla 15 Información relevante de Banco General Rumiñahui.....	60
Tabla 16 Resumen ejecución historia de usuario.....	63
Tabla 17 Resumen ejecución historia de usuario.....	65
Tabla 18 Resumen ejecución historia de usuario.....	69
Tabla 19 Resumen ejecución historia de usuario.....	71
Tabla 20 Resumen ejecución historia de usuario.....	73
Tabla 21 Resumen ejecución historia de usuario.....	78

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Impactos en la Banca Digital debido al COVID 19.....	14
Ilustración 2 Madurez de la Banca Digital	15
Ilustración 3 Etapas de la Gestión de Proyectos	17
Ilustración 4 Grupo de procesos PMBOK	20
Ilustración 5 Los 4 valores del Manifiesto Ágil.....	22
Ilustración 6 Los 12 principios del Manifiesto Ágil	23
Ilustración 7 Modelo de historia de usuario.....	26
Ilustración 8 Ejemplo de Burndown Chart	29
Ilustración 9 Ciclo de un Sprint de SCRUM	29
Ilustración 10 Arquitectura de alto nivel propuesta para un Asistente Virtual.....	32
Ilustración 11 Arquitectura utilizando DialogFlow y CloudFuntions de Google.....	33
Ilustración 12 Interacción de los componentes de DialogFlow	34
Ilustración 13 Arquitectura de integración de DialogFlow.....	35
Ilustración 14 Integración DialogFlow y APIGEE.....	36
Ilustración 15 SOA vs Microservicios.....	40
Ilustración 16 SOAP vs REST	41
Ilustración 17 Ejemplo de Customer Journey Map.....	44
Ilustración 18 Repositorio de información para consulta de Controles de Cambio	62
Ilustración 19 Comparativo de ejecución de la HU002	64
Ilustración 20 Comparativo de ejecución de la HU003	66
Ilustración 21 Burn Down Chart Sprint 1	66
Ilustración 22 API REST para la consulta de Controles de Cambio autorizados	67
Ilustración 23 Ejecución exitosa del API REST de consulta de Controles de Cambios Autorizados.....	67
Ilustración 24 Ejecución fallida del API REST de consulta de Controles de Cambios Autorizados.....	68
Ilustración 25 Ejecución exitosa del API REST de consulta de Socializaciones	68
Ilustración 26 Ejecución fallida del API REST de consulta de Socializaciones	68
Ilustración 27 Comparativo de ejecución de la HU004	70
Ilustración 28 Comparativo de ejecución de la HU005	72

Ilustración 29	Comparativo de ejecución de la HU006	74
Ilustración 30	Burn Down Chart Sprint 2	75
Ilustración 31	API REST para la consulta de la planificación de un Control de Cambio	75
Ilustración 32	Ejecución exitosa del API REST de consulta de la planificación de un control de cambio	76
Ilustración 33	Ejecución fallida del API REST de consulta de la planificación de un control de cambio	76
Ilustración 34	Ejecución exitosa del API REST de consulta del resultado de un control de cambio	76
Ilustración 35	Ejecución exitosa del API REST de consulta del resultado de un control de cambio	77
Ilustración 36	Ejecución exitosa del API REST de consulta del estado de un control de cambio	77
Ilustración 37	Ejecución fallida del API REST de consulta del estado de un control de cambio	77
Ilustración 38	Comparativo de ejecución de la HU001	79
Ilustración 39	Burn Down Chart Sprint 3	80
Ilustración 40	Creación del Chatbot en DialogFlow Essential	81
Ilustración 41	Extracción de la clave para la integración del Chatbot con la plataforma Whatsapp	82
Ilustración 42	Modificación del Intent de bienvenida por defecto del Chatbot	82
Ilustración 43	Creación del Intent para consulta de cambios autorizados	83
Ilustración 44	Configuración de la frase por defecto si el parámetro está en blanco	84
Ilustración 45	Configuración de Ngrok para crear la URL pública que exponga los microservicios	85
Ilustración 46	Configuración de Webhook para el intento	85
Ilustración 47	Prueba unitaria del Intent dentro de la plataforma de DialogFlow	86
Ilustración 48	Creación del autoresponder para integración con Whastapp en el equipo celular	87
Ilustración 49	Inicio de la conversación con el Chatbot en la plataforma de Whatsapp	87
Ilustración 50	Consulta de la información de cambios autorizados en Whatsapp mediante el Chatbot	88
Ilustración 51	Comprobación de consultas al microservicio expuesto por Ngrok	89
Ilustración 52	Creación del Intent para consulta de fecha de socialización de los cambios... ..	89

Ilustración 53 Prueba del Intent de consulta de socialización desde la plataforma de Whatsapp	90
Ilustración 54 Creación del Intent para consulta de fecha de planificación para ejecución de los cambios.....	90
Ilustración 55 Comprobación de la consulta del microservicio de planificación mediante Ngrok	91
Ilustración 56 Comprobación del Intent de la consulta de planificación en Whatsapp	91
Ilustración 57 Creación del Intent para consulta el estado de los cambios.....	92
Ilustración 58 Consulta directa del estado de dos cambios en la BDD	92
Ilustración 59 Consulta del estado del cambio desde Whastapp	93
Ilustración 60 Prueba del Intent de consulta del resultado de los cambios.....	93
Ilustración 61 Comprobación de la consulta a nivel del microservicio expuesto con Ngrok. 94	
Ilustración 62 Comprobación de la consulta a nivel de la BDD.....	94
Ilustración 63 Consulta del resultado del cambio a nivel del Intent en DialogFlow.	95
Ilustración 64 Consulta del resultado del cambio a nivel de Whatsapp.	95

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
MAESTRIA EN TECNOLOGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN
Y ADMINISTRACIÓN DE TI

**GESTIÓN DE UN PROYECTO ÁGIL PARA LA INTEGRACIÓN DE ASISTENTE
VIRTUAL CON MICROSERVICIOS**

Autor: Marco Vinicio Luna Aizaga

Director -Tutor: Leonardo Napoleón Arévalo Rivera

Fecha: Quito, abril 2024

RESUMEN

El tema "Gestión de un Proyecto Ágil para la Integración de Asistente Virtual con Microservicios" aborda la implementación de un asistente virtual integrado a microservicios utilizando metodologías ágiles para la gestión de proyectos.

En este proyecto, se aplican principios ágiles, el Marco de Referencia Scrum y herramientas de gestión como Kanban, para gestionar el desarrollo del asistente virtual y su integración con la arquitectura de microservicios. Esto implica un enfoque iterativo e incremental, donde se priorizan las funcionalidades más importantes y se entregan en ciclos cortos de desarrollo, permitiendo una rápida adaptación a los cambios y retroalimentación del cliente.

La integración del asistente virtual con microservicios implica dividir la funcionalidad en componentes independientes y escalables, lo que facilita la mantenibilidad y la evolución del sistema a largo plazo. Además, se pueden aprovechar las ventajas de los microservicios, como la flexibilidad, la escalabilidad y la facilidad de integración para implementar nuevas características.

La gestión de este proyecto implica la coordinación de un equipo multidisciplinario, que puede incluir desarrolladores de software, diseñadores de experiencia de usuario, especialistas en inteligencia artificial, entre otros. Se fomenta la colaboración y la comunicación constante entre los miembros del equipo, así como la adaptación a los cambios y la mejora continua del proceso de desarrollo.

En resumen, la gestión de un proyecto ágil para la integración de un asistente virtual con microservicios combina las metodologías ágiles con los principios de arquitectura de microservicios para desarrollar un sistema flexible, escalable y adaptable a las necesidades del cliente.

Palabras clave: SRUM, METODOLOGÍAS ÁGILES, KANBAN, MICROSERVICIOS.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
MAESTRIA EN TECNOLIGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN
Y ADMINISTRACIÓN DE TI

**MANAGEMENT OF AN AGILE PROJECT FOR THE INTEGRATION OF
VIRTUAL ASSISTANT WITH MICROSERVICES**

Autor: Marco Vinicio Luna Aizaga

Director -Tutor: Leonardo Napoleón Arévalo Rivera

Date: Quito, April 2024

ABSTRACT

The topic "Management of an Agile Project for the Integration of Virtual Assistant with Microservices" addresses the implementation of a virtual assistant integrated with microservices using agile methodologies for project management.

In this project, agile principles, the Scrum Reference Framework and management tools such as Kanban are applied to manage the development of the virtual assistant and its integration with the microservices architecture. This implies an iterative and incremental approach, where the most important functionalities are prioritized and delivered in short development cycles, allowing rapid adaptation to changes and customer feedback.

Integrating the virtual assistant with microservices involves dividing the functionality into independent and scalable components, which facilitates long-term maintainability and evolution of the system. Additionally, the advantages of microservices such as flexibility, scalability, and ease of integration can be leveraged to implement new features.

The management of this project involves the coordination of a multidisciplinary team, which may include software developers, user experience designers, artificial intelligence specialists, among others. Collaboration and constant communication between team members are encouraged, as well as adaptation to changes and continuous improvement of the development process.

In summary, agile project management for the integration of a virtual assistant with microservices combines agile methodologies with the principles of microservices architecture to develop a flexible, scalable and adaptable system to the client's needs.

Keywords: SCRUM, AGILE METHODOLOGIES, KANBAN,
MICROSERVICES

INTRODUCCIÓN

Hoy en día los asistentes virtuales en las instituciones bancarias son de mucha utilidad como medios de consulta e inicializadores de procesos que por su naturaleza son tareas repetitivas y operativas, la puesta en producción de los asistentes virtuales se lo realiza mediante la ejecución de proyectos que permiten establecer un alcance, una línea de tiempo y sobre todo definir como estos asistentes apoyan en la consecución de los objetivos de estas instituciones.

Según el portal Statista (Fernandez, 2021), para el año 2019 ya se tenía un uso de 3,25 miles de millones de asistentes virtuales alrededor del mundo y se prevé un uso de 8,4 miles de millones de asistentes virtuales para el año 2024.

El uso de los asistentes virtuales sin duda se presenta como una alternativa disruptiva para innovar servicios y la forma en que se interactúa con los clientes y/o usuarios; de hecho, existen ya asistentes que forman parte de la cultura popular como Alexa, Siri, Cortana, entre otros.

En la actualidad la arquitectura casi de facto para la integración entre plataformas y dispositivos es mediante la exposición y consumo de Microservicios. La mayoría de los lenguajes de programación y plataformas de desarrollo brindan la posibilidad de construir microservicios arquitecturas y protocolos estándar como, por ejemplo, SOAP y REST.

El presente trabajo de investigación fue elaborado mediante la utilización del marco de referencia SCRUM para la formulación del proyecto y la ejecución del sprint y la plataforma DialogFlow para la creación del asistente virtual.

CAPÍTULO I:

PLANTEAMIENTO DEL PROBLEMA

1.1 Formulación del problema

En el Plan Estratégico de Banco General Rumiñahui (BGR) para el periodo 2018 – 2021 estipula, como uno de sus objetivos estratégicos, el “Potencial el modelo de negocio centrado en la experiencia del cliente con enfoque ágil, innovador y digital”; sin embargo, para la gestión de proyectos, BGR utiliza una metodología en cascada que ha significado la entrega de soluciones hacia los clientes de manera extemporánea y en muchos casos perdiendo el “time to market” con relación a otras instituciones financieras que trabajan sobre sus mismo nichos de mercado (militar y civil).

Las tareas que se ejecutan dentro de los proyectos en BGR son burocráticos y, además, implican solicitar una serie de permisos para la puesta en producción de las soluciones desarrolladas, lo cual no ha permitido que el banco tenga un poder de reacción ágil ante los cambios del mercado. Mientras en otras instituciones financieras se ofrecen la mayoría de los productos bancarios de forma digital, en BGR aún no se ha podido entregar algunas de esas facilidades a sus clientes, lo cual también ha ido en contra de la experiencia del usuario.

En la actualidad en BGR se ha ido implementando una serie de mejoras a la metodología de gestión de proyectos incluyendo una serie de herramientas del mundo del agilismo pero que no han podido atacar el problema de raíz que es el poder entregar soluciones de forma ágil.

La estadística para el año 2020¹ indicaba que un proyecto pequeño como, por ejemplo, para la creación de un chatbot, tomaba un promedio de 4 a 6 meses cuando en la competencia se ofrece este tipo de soluciones ya y cualquier incremento en su

¹ Fuente: KPIs Gestión de Proyectos BGR año 2020

funcionalidad lo realizan en cuestión de semanas.

En BGR un cambio o incremento de funcionalidad, como por ejemplo la integración de un chatbot con microservicios, bajo la metodología actual, significaría la formulación de otro proyecto pequeño por lo que, el cliente, debería esperar a otros 4 o 6 meses para poder hacer uso de ese cambio o mejora.

Esto ha puesto de manifiesto la necesidad de contar con una nueva metodología que permita cumplir con los objetivos estratégicos del banco y que, además, permita darle al cliente una experiencia de usuario única con efecto WOW².

1.2 Objetivos de la Investigación

1.2.1 Objetivo General

Gestionar un proyecto ágil para la integración de un asistente virtual con microservicios en el Banco General Rumiñahui.

1.2.3 Objetivos Específicos

- Fundamentar teóricamente el marco de referencia ágil para la ejecución de proyectos.
- Elaborar el Backlog con los requerimientos funcionales para la creación del asistente virtual
- Crear un Chatbot para que se integra con microservicios
- Comprobar la entrega de valor en el menor tiempo posible mediante la ejecución de los Sprints.

² WOW: capacidad que tiene la marca de generar sensaciones únicas en el cliente.

1.3 Justificación de la Investigación

Con la ejecución de proyectos con el uso de metodologías ágiles, tanto los clientes como los usuarios internos del Banco General Rumiñahui se verán beneficiados por la entrega de soluciones, que agreguen valor al negocio, en periodos más cortos.

Además, aumentará el poder de resiliencia ante los cambios del mercado haciendo que el agilismo se convierta en una ventaja competitiva.

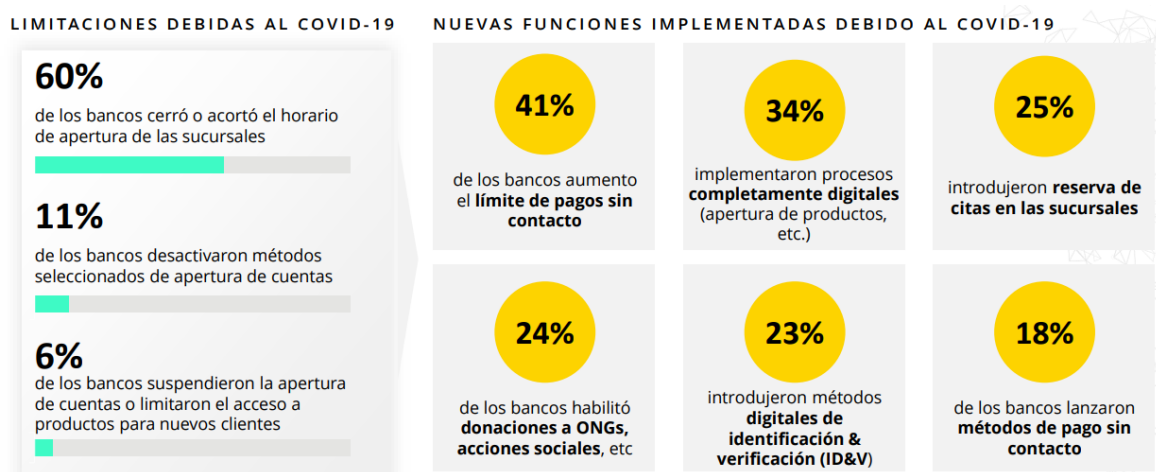
El uso de metodologías ágiles para la ejecución de proyectos que conlleven la creación de aplicaciones con un grado de complejidad medio o alto, representarán también un beneficio para BGR toda vez que se podrá poner en producción aplicaciones con cierto grado de complejidad en periodos más cortos y ofrecer productos digitales a los clientes que aún no los tienen en otras instituciones financieras.

El enfoque interactivo que tienen las metodologías ágiles permitirá que el Banco General Rumiñahui pueda colocar también de en periodos cortos, nuevas funcionalidades, cambios, mejoras o correcciones a los productos digitales actuales y futuros. Con esto, la experiencia del usuario se verá beneficiada porque podrá tener productos que cumplan sus necesidades y que tengan el efecto deseado.

A continuación, el estudio de Madurez de la Banca Digital del año 2020 expuesto por Deloitte³.

³ Fuente: <https://www2.deloitte.com/content/dam/Deloitte/cl/Documents/financial-services/cl-madurez-de-la-banca-Digital-2020.pdf>

Ilustración 1 Impactos en la Banca Digital debido al COVID 19



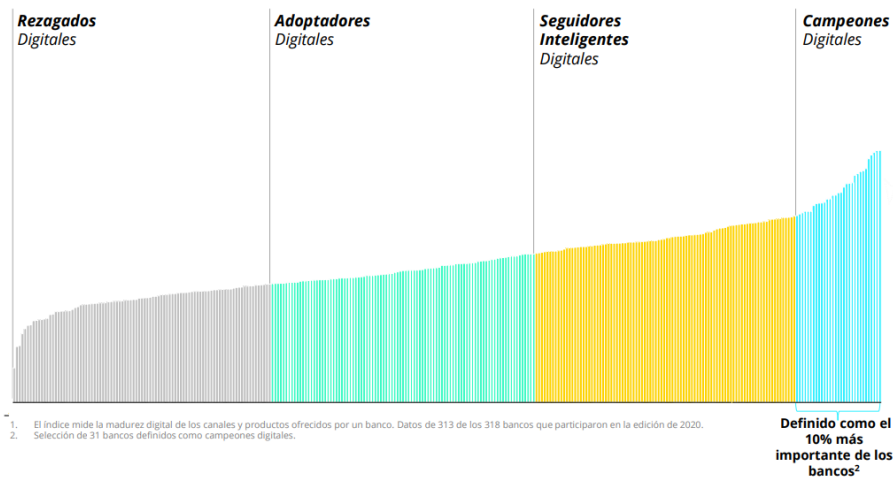
Fuente: (Deloitte, 2020)

Con base a esta estadística, se puede apreciar que la realidad de la Banca en el mundo ha cambiado, es decir, la Banca Digital ha acelerado su uso y se ha convertido en el canal más importante para los bancos a raíz de la pandemia lo cual justifica el hecho de que BGR debe fortalecer y mejorar su “time to market” frente a esta nueva realidad a nivel mundial y en la industria financiera y bancaria.

Si bien la gestión de proyectos no trae con sí la mejora de los productos, sí es un factor crítico a la hora de poner en producción productos digitales en el menor tiempo posible y poder lograr el mencionado “time to market” o poder de resiliencia. Esta necesidad se puede justificar con base a la estadística presentada en el mismo estudio de Deloitte como se muestra en la siguiente gráfica:

Ilustración 2 Madurez de la Banca Digital

Índice de Madurez de la Banca Digital ¹



Qué define a los campeones Digitales?

Los campeones digitales ofrecen una amplia gama de funcionalidades relevantes para los clientes y una experiencia de usuario atractiva.

Los Campeones digitales establecen tendencias digitales claves y tienen prácticas líderes en el mercado, lo que los convierte en ejemplos de los cuales aprender

Fuente: (Deloitte, 2020)

Como se puede apreciar, el banco que ofrezca a sus clientes una amplia gama de productos digitales será el que marque el ritmo del mercado en los próximos años y debido a esto, BGR debe poder completar su portafolio de productos digitales en el menor tiempo posible y poder adelantarse a sus competidores.

CAPÍTULO II:

FUNDAMENTACIÓN TEÓRICA

2.1. Antecedentes de la Investigación

En investigaciones realizadas respecto a la aplicación de metodologías de proyectos tradicionales versus ágiles tenemos el trabajo de (Gómez, 2015) quien en su investigación realizada en Proyectos de IT en el entorno Industrial en España, entre otros, buscó determinar cuáles son las causas del fracaso de los proyectos de IT, para lo cual cita el estudio realizado por The CHAOS Report publicado por Standish Group, en donde se muestra que entre el 2011 y 2015, de los proyectos analizados, solo el 29% son exitosos y existe un 52% de proyectos son discutidos; es decir, proyectos que no han sido aceptados en su totalidad debido a problemas en la gestión y sus resultados. Se concluyó que la clave a la hora de gestionar un proyecto es determinar la metodología o marco de referencia que más se ajuste a la naturaleza del proyecto en sí.

Por otro lado, tenemos que (IEBS Bussiness School, 2023) quienes, en su artículo científico sobre metodologías ágiles de proyectos, entre otros, presentan los principales conceptos de agilidad para la gestión de proyectos bajo este tipo de metodologías y su apego al Manifiesto Ágil. En este trabajo se concluye que “el uso de metodologías ágiles busca la entrega de valor al negocio de manera iterativa lo cual puede redundar en periodos más cortos de entrega de versiones funcionales y usables a la producción” (pág. 4).

Adicional, tenemos que (Manrique, 2022) en su investigación sobre soluciones tecnológicas para mejorar el proceso de atención a clientes en entidades bancarias utilizando asistentes virtuales, buscaron mostrar, entre otros, el grado de mejoría de la satisfacción del cliente utilizando bots. En este trabajo se muestra el resultado de la encuesta realizada a los bancos Seylan Bank y Sampath Bank donde se puede apreciar que “los índices sobre: respuesta esperada por el usuario, usabilidad, facilidad de uso y precisión; se incrementan cuando se utiliza un bot con una tecnología adecuada y orientada a las características particulares de los clientes de dichos bancos” (pág. 135).

2.2. Bases Teóricas.

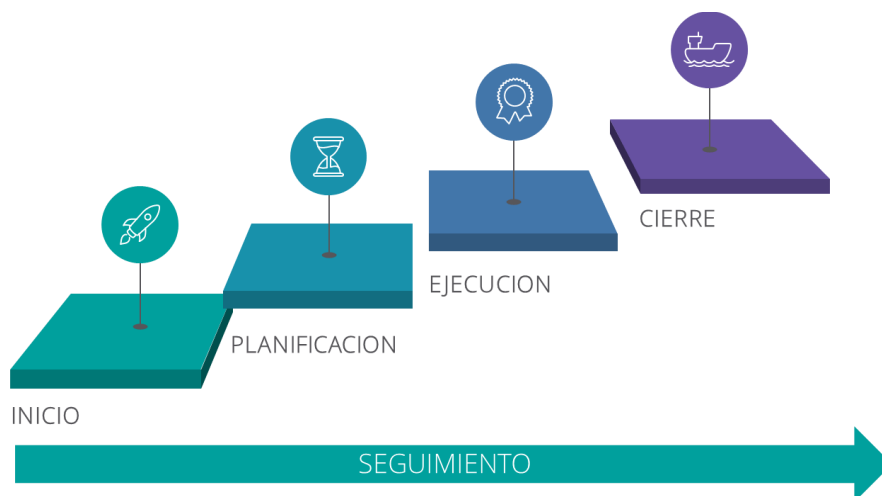
A continuación, se describe una serie de conceptos que servirán de base para la comprensión de presente trabajo de investigación.

2.2.1 Gestión de Proyectos

Según (PMPBOOK, 2022) La gestión de proyectos se define como la disciplina que busca el manejo adecuado y coordinado de los esfuerzos, recursos, materiales, personas, alcances, tiempos y costos para la entrega de un producto o servicio que satisfaga los requerimientos iniciales

Con mayor o menor énfasis, la gestión de proyectos contempla cinco etapas principales:

Ilustración 3 Etapas de la Gestión de Proyectos



Elaborado por: Marco Vinicio Luna Aizaga

- Inicio: Es la etapa en la cual generalmente se realizan los estudios de factibilidad del proyecto, es decir, en esta etapa se determina si el proyecto es financiera, operativamente viable. En la actualidad, es recomendable que también en esta etapa se haga un análisis de factibilidad con base en el ecosistema tecnológico de la organización dado que puede existir problemas de obsolescencia o de prerequisites tecnológicos que pueden poner en riesgo el éxito de un proyecto.

Adicional, es en esta etapa se realiza la recolección de los requerimientos funcionales y no funcionales que debe cubrir el proyecto

- **Planificación:** En esta etapa, se realiza la estimación del alcance, costo y tiempo que tomará el proyecto para su ejecución con base a los análisis de factibilidad y los requerimientos levantados en la etapa inicial. Generalmente el producto de esta etapa es un plan donde se muestre el manejo de los recursos y un cronograma donde se indique las tareas que se ejecutarán, en qué tiempo y a qué costo.
- **Ejecución:** En esta etapa se llevan a cabo las tareas descritas en la planificación con el fin de crear el producto o el servicio objeto del proyecto. Generalmente en esta etapa se utilizan marcos de referencia especiales según la naturaleza del producto o servicio sea software, una obra física, un proceso, entre otros. En esta etapa también se elabora la documentación y entregables del proyecto, se realizan las contrataciones, se realizan los pagos y se realiza el control del alcance, tiempo y costo.
- **Seguimiento:** Se trata de una etapa especial dado que se ejecuta a lo largo del ciclo de vida del proyecto y sirve como complemento de las otras etapas dado que se realiza el monitoreo del desarrollo del proyecto de principio a fin con el afán de controlar su normal desarrollo y alertar sobre impactos que redunden en riesgos, pérdidas o retrasos.
- **Cierre:** Finalmente, cuando se pone en funcionamiento o producción el producto o servicio objeto de un proyecto, se deben ejecutar una serie de tareas de cierre como, por ejemplo, el archivo de documentos, la firma de contratos de mantenimiento en producción, la entrega de informes de auditoría o fiscalización, entre otros. Se trata del cierre administrativo del proyecto.

La gestión de proyectos se lleva a cabo mediante la adopción y ejecución de metodologías y/o marcos de referencia que, según su naturaleza, ejecutan las cinco etapas mencionadas con un mayor o menor grado de especialización y con un orden

establecido. Puede ser que, según la metodología o marco de referencia, se presenten más etapas; sin embargo, solo se trata de una especialización de alguna de las etapas descritas o en su defecto, un cambio en el orden y periodicidad de ejecución de las etapas antes descritas.

2.2.2 Metodología en Cascada

Según (PMPBOOK, 2022), La concepción tradicional de la metodología de proyectos tiene una orientación a realizar las actividades de forma secuencial y con dependencia de una tarea con otra; si bien existe la posibilidad de ejecutar tareas de forma paralela, las macro tareas o fases siempre serán secuenciales; es decir, no puede iniciar una etapa sin haberse finalizado la otra.

Esto supone ciertas ventajas y desventajas que se pueden resumir a continuación:

Tabla 1 Ventajas y Desventajas de la Metodología en Cascada

VENTAJAS	DESVENTAJAS
Se tiene todos los requerimientos funcionales y no funcionales bien definidos al inicio de la ejecución	Se debe ejecutar un proceso de control de cambios para poder modificar el alcance del proyecto lo cual puede conllevar trámites administrativos y carga burocrática
Se tiene un presupuesto delimitado	Si se debe realizar ajustes al presupuesto de forma incremental, esto debe ser aprobado generalmente por comités y dicha aprobación está sujeta a la disponibilidad.
Se conoce de antemano cuánto va a durar el proyecto	Cualquier cambio en la fecha de fin del proyecto puede constituirse en retraso y en indicadores negativos.

Fuente: (PMPBOOK, 2022)

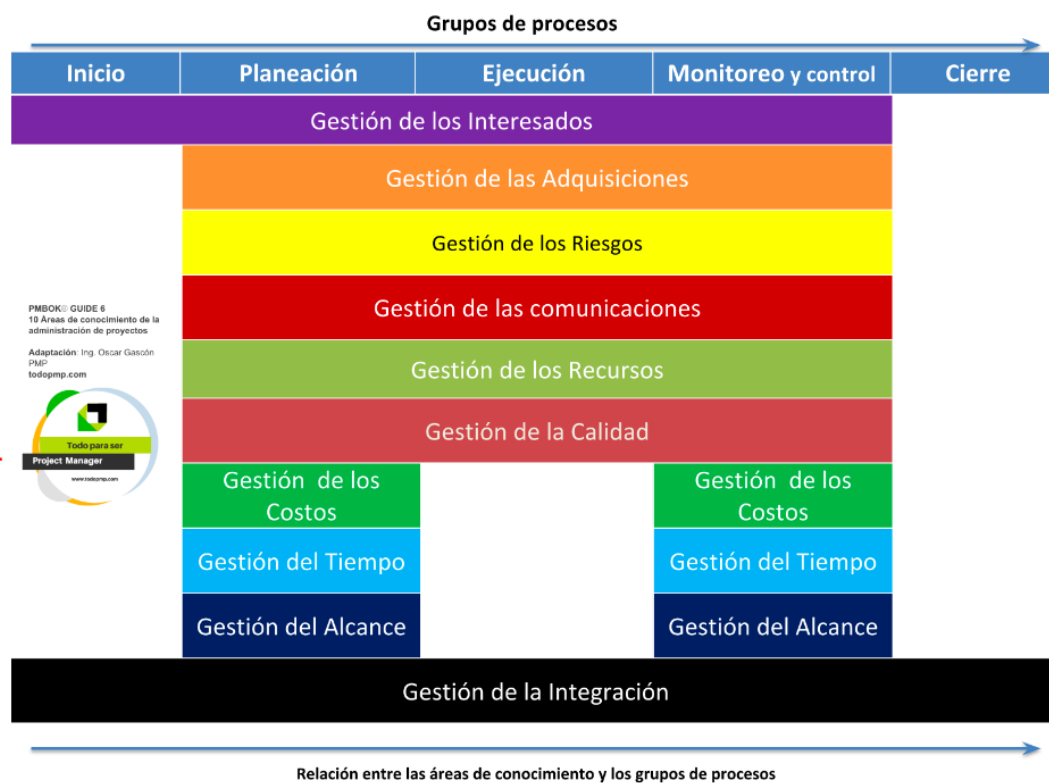
Elaborado por: Marco Vinicio Luna Aizaga

Existen varios ejemplos de metodologías en cascada, con mayor o menor grado de detalle en ciertas fases o la inclusión de sub-fases, pero siempre se cumple con la premisa de que las fases no son paralelas sino secuenciales e interdependientes.

La metodología en cascada mas reconocida a nivel mundial es la emitida por el Instituto de Project Management (Project Management Institute, 2024) misma que ofrece una certificación que ha ganado prestigio a nivel profesional a lo largo de los años como es el PMP (Project Management Professional) la cual avala a su poseedor como experto en el dominio de los cuerpos de conocimiento de la mencionada metodología y que no son otra cosa que una serie de procesos que van describiendo cada una de las diciplinas que, según el PMI, intervienen en la gerencia de un proyecto.

Sin embargo, esta metodología no deja de ser una metodología en cascada con un fuerte componente secuencial.

Ilustración 4 Grupo de procesos PMBOK



Fuente: (PMPBOOK, 2022)

Toda la descripción de los procesos y cuerpos de conocimientos, el PMI los recopila en el PMBOK el cual es la guía principal de la metodología.

Es importante mencionar que, si bien esta metodología está muy bien elaborada y es de uso común, no deja de ser solo una herramienta de trabajo. Es necesario poner todo ese conocimiento en la práctica y contraponerlos ante las situaciones cotidianas de un proyecto.

2.2.3 Manifiesto Ágil

La gestión de proyectos es una disciplina que se puede ejecutar en varias industrias independientemente del giro de negocio; sin embargo, tiene su mayor grado de uso en la industria del desarrollo de software.

Muchas compañías han utilizado durante décadas varias metodologías estructuradas y en cascada para la gestión de proyectos que les han permitido poner en producción muchos de los avances tecnológicos que vemos hasta la fecha. Con el pasar de los años, las empresas han visto la necesidad de buscar aceleradores que les permita poner cosas en producción en períodos mas cortos sin sacrificar la calidad, pero buscando siempre tener el “time to market” como una ventaja competitiva.

La combinación del ciclo de vida de desarrollo de software con las metodologías de gestión de proyectos ha provocado el apareamiento de marcos de referencia para el desarrollo de software en tiempos mas cortos y es así como (Beck & Beedle, 2001) se reúnen con otros 13 expertos en programación y redactan lo que se conoce como el Manifiesto Ágil⁴ que supuso un cambio radical en la forma de desarrollar (pág. 1).

El Manifiesto Ágil está formado de 4 valores y 12 principios que buscan demostrar lo que realmente debe importar al momento de desarrollar software y sobre todo pone en la palestra el concepto de agilismo.

⁴ Fuente: <https://agilemanifesto.org/iso/es/manifesto.html>

Ilustración 5 Los 4 valores del Manifiesto Ágil



Fuente: (Ruiz, 2018)

Estos valores demuestran la intencionalidad de los expertos que los enunciaron y es la necesidad de desarrollar software, que genere valor, en periodos más cortos. Esto describe claramente lo que es agilidad, misma que no tiene que ver con rapidez.

Es el símil de un auto Ferrari compitiendo contra una camioneta 4X4 como un carro de la marca Jeep; si queremos definir solamente qué carro es más rápido, es evidente que la Ferrari ganará la partida, pero si adicionamos la variable del tipo de terreno y decimos que los dos deben competir en un terreno con grava e irregular, seguramente el Jeep será relativamente más rápido; sin embargo, realmente lo que está siendo es más ágil en ese tipo de terreno.

Las metodologías y/o marcos de referencia ágiles se basan en esos 4 valores, buscando entregas en periodos más cortos, pero sin sacrificar calidad, control y sobre todo el agregar valor al negocio. Todos los procesos que se mencionan en dichas metodologías ponen en práctica también los 12 principios del manifiesto ágil.

Ilustración 6 Los 12 principios del Manifiesto Ágil



Fuente: (Benavidez, 2019)

2.2.4 Metodologías y/o Marcos de Referencia Ágiles

Existen varias metodologías y/o marcos de referencia ágiles; sin embargo, según el (IEBS Bussiness School, 2023) las 3 metodologías ágiles más utilizadas, entre otras, son Extreme Programming (XP), SCRUM y Kanban. Para el presente trabajo se tomará en cuenta estas tres metodologías ágiles con fines explicativos.

Una vez definido lo que es agilismo, los valores y principios que lo gobiernan y, además de definir las metodologías ágiles más utilizadas, es necesario caracterizarlas a fin de decidir cuál metodología y/o marco de referencia se utilizará para el desarrollo del presente trabajo.

Tabla 2 Principales metodologías ágiles

Metodología	Creador y año de creación	Principales características
XP	Kent Beck (1999)	Se basa en pruebas unitarias, refactoring, pair programming, comunicación constante entre usuarios y desarrolladores, feedback constante.
Scrum	Jeff Sutherland y	Método iterativo e incremental en el

	Ken Schwaber (1995)	cual se define tres roles principales (Product Owner, Scrum Máster y Scrum Team), ejecución de sprint para el desarrollo y entrega de valor, mantiene los requerimientos funcionales bajo el concepto de product y sprints backlogs.
Kanban	Taiichi Ono (Toyota) (1963)	Nació como la herramienta principal de la metodología LEAN que se creó en Toyota para mejorar el rendimiento y productividad de la fabricación de automóviles. Kanban realmente no es una metodología, es solo una herramienta; sin embargo, es quizá la herramienta más utilizada en el agilismo.

Elaborado por: Marco Vinicio Luna Aizaga

Del cuadro anterior se desprenden dos metodologías formales y una herramienta. Para el presente trabajo se utilizará SCRUM para demostrar la gestión de proyectos ágiles.

2.2.5 Marco de Referencia SCRUM

SCRUM, más que una metodología que indique a modo de receta lo que se debe hacer y cómo hacerlo, es más bien, un marco de referencia principales características se resumen de la siguiente manera (Martins, 2024):

Tabla 3 Principales características de SCRUM

Característica	Descripción
Roles	Product Owner Scrum Máster Scrum Team
Fases	Creación del Product Backlog Priorización del Product Backlog Planificación de releases Creación del Sprint Backlog Ejecución del Sprint (Iteración) Entrega del resultado del Sprint
Principales Ceremonias	Release Planning Sprint Planning Reunión diaria (daily) Revisión del Sprint Retrospectiva del Sprint
Principales entregables y herramientas	User stories Product Backlog Sprint Backlog Burn down chart Definition of Done Definition of Ready

Fuente: (Martins, 2024)

Elaborado por: Marco Vinicio Luna Aizaga

El alma de SCRUM son las historias de usuario (user stories) las cuales representan los requerimientos funcionales y no funcionales que van a ser abordados durante el proyecto. Existe una forma estándar para la elaboración de las historias de usuario misma que se describe a continuación:

Ilustración 7 Modelo de historia de usuario

HISTORIA DE USUARIO: <<número/código>>		
COMO	<<role, persona>> Persona o rol de usuario que tiene la necesidad	
QUIERO	<<objeto, comportamiento>> Lo que se quiere obtener: funcionalidad, característica, entre otros.	
PARA	<<motivo, razón, valor>> Motivo por el que se necesita, valor que se obtiene como resultado, entre otros.	
VALIDACIONES:	VALOR:	
	PRIORIDAD:	
	ESTIMACIÓN:	

Fuente: propia

Elaborado por: Marco Vinicio Luna Aizaga

Como se aprecia en la imagen, la historia de usuario describe con claridad lo que se quiere, quien lo requiere y por qué lo requiere; pero adicional, también tiene la versatilidad para mostrar elementos adicionales que ayudan a la descripción de la historia de usuario y facilitan su inclusión dentro del Backlog del proyecto aportando información relevante como, por ejemplo, los criterios de aceptación, la estimación de esfuerzo, entre otros.

Una vez establecidas las historias de usuario del proyecto, se realiza una de las principales ceremonias en SCRUM como lo es el Release Planning en el cual se priorizan las historias de usuario para determinar el orden en el cual serán atendidas y sobre todo para determinar los releases o entregas a la producción. Como resultado de esta ceremonia, se tiene el Product Backlog priorizado; es decir, un tablero en el cual se tienen todas las historias de usuario del proyecto ordenadas y priorizadas.

Otro concepto importante dentro de SCRUM es el sprint que se refiere a un ciclo o iteración en la cual se ejecutan las tareas necesarias para satisfacer las necesidades descritas en las historias de usuario, pero solo en aquellas seleccionadas para el mencionado sprint. En este punto es necesario definir que, antes de la ejecución del sprint, se lleva a cabo el Sprint Planning la cual es otra de las ceremonias de SCRUM; en dicha reunión se selecciona las historias de usuario que se desarrollarán, se establece

la duración del sprint, los actores que formarán parte de este, se establece las tareas a detalle que se ejecutarán para cumplir con la historia de usuario, entre otras.

Si bien existen solo tres roles principales en SCRUM, también se pueden incluir roles adicionales que aporten valor a la ejecución del sprint como arquitectos, ingenieros de QA, ingenieros de infraestructura, entre otros.

Otro elemento importante dentro del desarrollo de SCRUM es las Historias Épicas que no son más que historias de usuario que han sido redactadas a muy alto nivel y que requieren ser disgregadas a más de una historia de usuario con mayor detalle y que permita medir su cumplimiento a cabalidad.

El resultado del Sprint Planning es el Sprint Backlog que contendrá las historias de usuario seleccionadas y, sobre todo, el incremento o producto generado luego del Sprint.

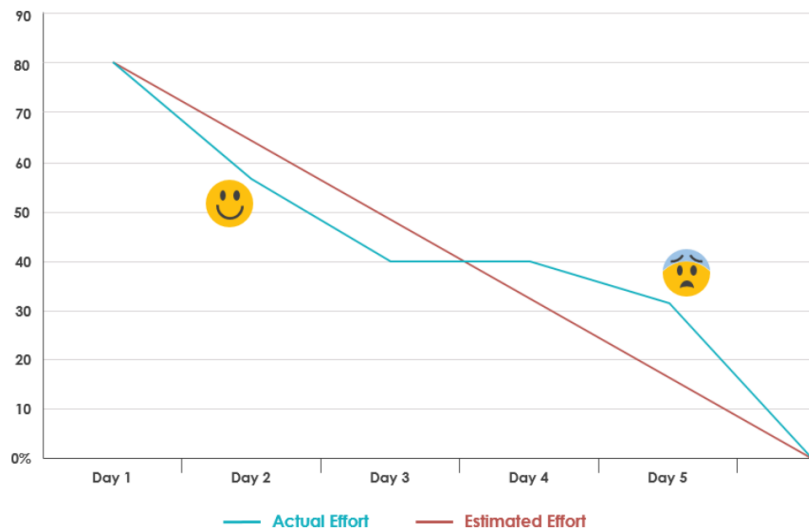
Dentro del desarrollo del Sprint se tiene varias herramientas útiles para su ejecución:

- Daily: la cual se trata de una reunión diaria con el fin de conocer, por parte de cada uno de los miembros del equipo, qué hicieron el día de ayer, qué van a ejecutar el día de hoy y qué impedimentos han tenido. El Scrum Máster tendrá la responsabilidad de quitar los impedimentos que se mencionen en dicha ceremonia.
- Kanban del Sprint: es un tablero compuesto por secciones verticales que van describiendo el avance de las tareas. Este tablero está compuesto generalmente por tres secciones: To Do (tareas por hacer), Doing (tareas en ejecución) y Done (tareas ya realizadas); en dichas secciones se van colocando y moviendo las tareas que se crearon para cumplir con cada una de las historias de usuario que forman parte del sprint.
- Sprint Review: es una de las ceremonias de SCRUM en la cual el equipo se

reúne para realizar una demostración de lo generado durante el sprint. De esta reunión se puede desprender nuevas historias de usuario que pueden llevar a re priorizar el Product Backlog e incluso generar nuevas tareas.

- **Sprint Retrospective:** es una ceremonia de SCRUM que se lleva a cabo al finalizar cada Sprint y cuyo objetivo es analizar cómo se ha desarrollado el sprint, establecer puntos de mejora, ver qué se puede mejorar, en otras palabras, asegurar la mejora continua del proyecto.
- **Burndown Chart:** es un diagrama en el cual se muestra el trabajo completado y su performance respecto a un ritmo ideal de ejecución de tareas. Es una herramienta útil para medir la velocidad y buscar la forma de ejecutar las tareas de forma rápida. Si la línea del esfuerzo actual está por debajo de la línea de esfuerzo estimado, significa que el trabajo está adelantado o se está realizando en menor tiempo al estimado. Si la línea de esfuerzo estimado está por encima de la línea de esfuerzo estimado, significa que el trabajo está retrasado y hay que establecer las acciones necesarias para regresar o acercarse a la línea de esfuerzo estimado.
- **Definición de hecho (Definition of done - DoD):** es un acuerdo entre el equipo en el cual se demuestra un entendimiento compartido, entre todos los actores, sobre las expectativas que el incremento (resultado del Sprint) debe cumplir para poder ser liberado a producción.
- **Definición de listo (Definition of ready - DoR):** Es un acuerdo entre el equipo para determinar cuándo una historia de usuario está lista para ser tomada en cuenta dentro del Sprint, es decir, que está lista en fondo y forma y aceptada entre las partes.

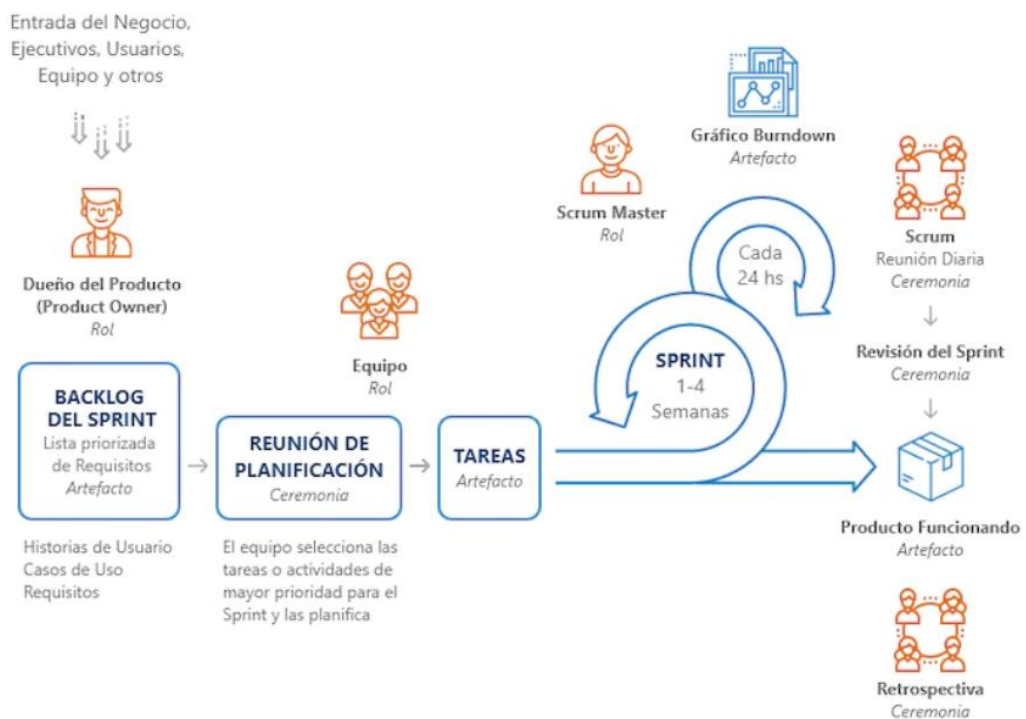
Ilustración 8 Ejemplo de Burndown Chart



Fuente: (Visual-Paradigm, 2024)

En la siguiente imagen se resume un Sprint tomando en cuenta todos los puntos mencionados anteriormente:

Ilustración 9 Ciclo de un Sprint de SCRUM



Fuente: (González, 2021)

2.2.6 Tecnología de Asistentes Virtuales

Un asistente virtual es un software desarrollado para la interacción con humanos, mediante sistema de voz. Se trata de un tipo de tecnología capaz de reconocer preguntas y contestarlas, de manera mucho más fluida y natural que un chatbot.

Al tratarse de un robot, este asistente está disponible de forma continua, 24/7, para ayudar a los clientes en sus posibles dudas. Además, de las interacciones aprenden para futuras ocasiones (Lauria, 2021)

Existen desde asistentes virtuales que permiten la interacción autónoma con dispositivos hasta asistentes virtuales que ejecutan tareas autónomas sin la necesidad de la interacción hombre-máquina.

Los asistentes virtuales forman parte de un espectro amplio conocido como Inteligencia Artificial y se crean sobre la base de dos componentes principales:

- **Machine Learning:** es una tecnología que habilita la capacidad de aprendizaje a partir de las interacciones del usuario.
- **Servicios Cognitivos:** es una tecnología que se basa en una serie de APIs que habilitan capacidades para el aprendizaje y comportamiento predictivo de los asistentes virtuales (servicios de reconocimiento de voz, imágenes, motores de búsqueda, bases de conocimiento, motores de lenguaje, entre otros).

A continuación, un resumen de las principales tecnologías utilizadas en estos dos componentes sobre los cuales se construye los Asistentes Virtuales:

Tabla 4 Tecnología de Machine Learning y Cognitive Services

Componente	Fabricante	Tecnología
Machine Learning	IBM	IBM Cognos Analytics
	Oracle	Oracle Analytics Services
	SAP	SAP Business Objects BI
	SAS	SAS Platform
	Microsoft	SSRS
Servicios Cognitivos	Microsoft	Azure Cognitive Services
	Oracle	Servicios de OCI
	IBM	Watson
	Amazon	Amazon IA

Elaborado por: Marco Vinicio Luna Aizaga

Es necesario mencionar que, si bien estas son las tecnologías sobre las cuales se basan los asistentes virtuales, finalmente éstos son ofertados en el mercado como soluciones completas que encapsulan dichas tecnologías.

Es así como se ofertan en el mercado varios asistentes virtuales muy populares como, por ejemplo:

Tabla 5 Asistentes virtuales más utilizados

Fabricante	Asistente Virtual
Amazon	Alexa, incluido en los altavoces Echo
Microsoft	Cortana, el cual está incluido en el sistema operativo Windows.
Apple	Siri, incluido en todos los dispositivos Apple junto al sistema operativo IOS
Google	Google Assistant incluido en la mayoría de los dispositivos con sistema operativo

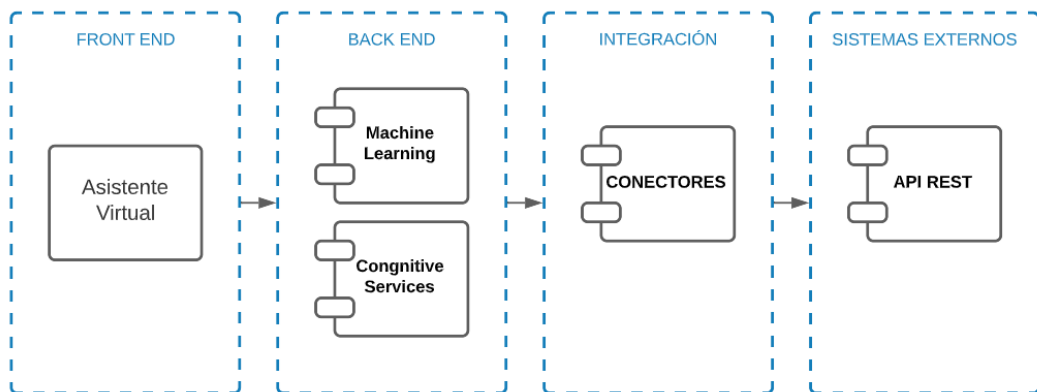
	Android.
Samsung	Bixby, incluido en la mayoría de sus dispositivos actualmente.

Elaborado: Marco Vinicio Luna Aizaga

Si bien los asistentes virtuales se presentan como soluciones cerradas, en la actualidad ofrecen una capa de integración a fin de ampliar y personalizar sus capacidades hacia una empresa y negocio en particular.

Esto nos lleva a establecer una arquitectura a alto nivel propuesta para la creación de asistentes virtuales en donde se incluya los componentes que hemos revisado hasta el momento:

Ilustración 10 Arquitectura de alto nivel propuesta para un Asistente Virtual

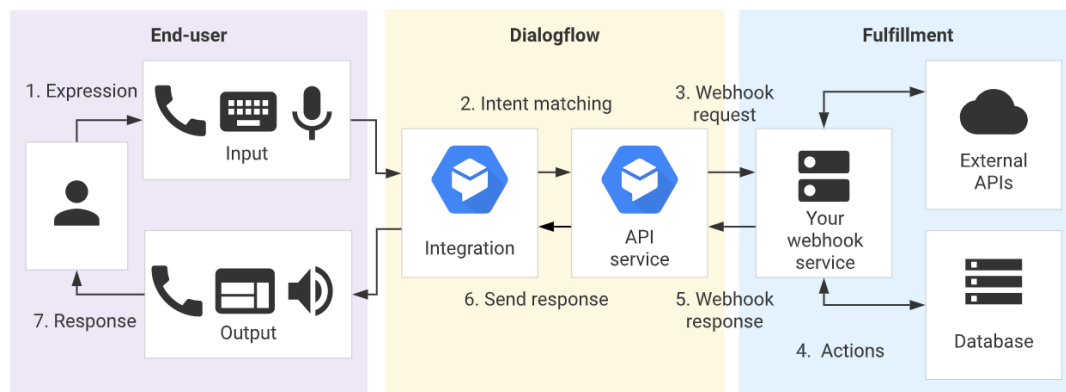


Elaborado por: Marco Vinicio Luna Aizaga

Con base a esta arquitectura, se define el uso de Google Assistant (DialogFlow) para la construcción del Asistente Virtual y su integración mediante microservicios a través de Google Cloud Functions.

A continuación, se muestra la arquitectura en particular que se utilizará para el presente trabajo.

Ilustración 11 Arquitectura utilizando DialogFlow y CloudFuntions de Google



Fuente: (Google, 2022)

2.2.7 DialogFlow

Según su fabricante (Google, 2022) “es una plataforma que brinda la capacidad para crear agentes virtuales con comprensión de lenguaje natural” (pág.1); se utiliza para la creación de soluciones con interfaz de usuario conversacional que puede adaptarse a dispositivos móviles, web, bots, sistemas de respuesta de voz interactiva, entre otros, DialogFlow ofrece dos sub-plataformas para la creación de agentes virtuales:

- DialogFlow CX: que se utiliza para la creación de agentes avanzados adecuado para agentes grandes y complejas.
- DialogFlow ES: que se utiliza para la creación de agentes estándar adecuado para agentes pequeños y simples.

Adicional, DialogFlow ofrece una API para la creación de chatbots incluida en la plataforma DialogFlow ES. Esta API lleva el nombre de Agent Assist.

DialogFlow consta de los siguientes componentes:

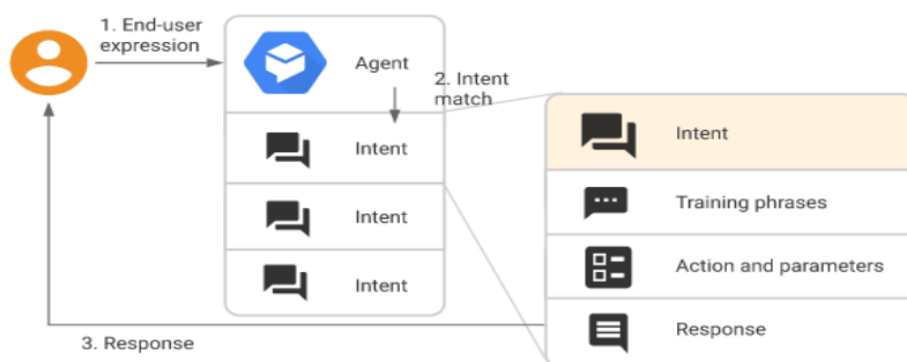
Tabla 6 Componentes de DialogFlow

Componente	Descripción
Agente	Es el agente virtual en sí y maneja la conversación con el usuario final
Intentos	Representan las acciones realizadas por el agente
Entidades	Sirven para extraer información de la consulta del usuario
Parámetros	Son objetos que guardan información útil para los intentos
Contextos	Son objetos que permiten almacenar información dentro de una misma sesión
Eventos	Se utilizan para lanzar intentos desde código
Respuesta	Es el objeto que se utiliza para enviar la respuesta al usuario

Elaborado por: Marco Vinicio Luna Aizaga

A continuación, se muestra la interacción de los componentes descritos según la documentación del fabricante:

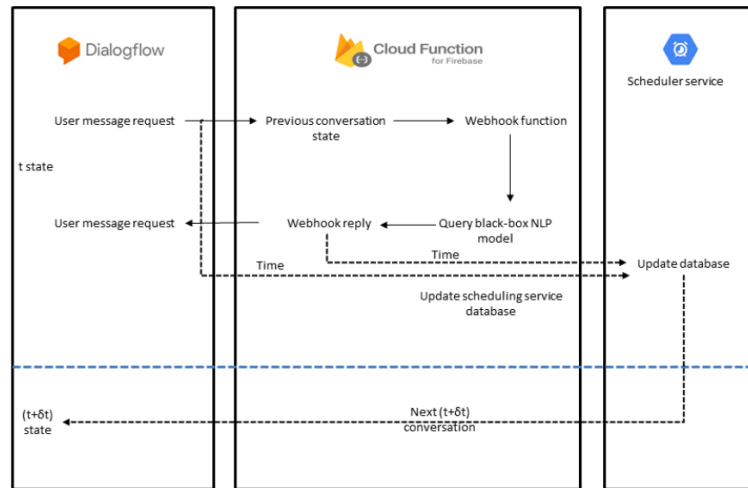
Ilustración 12 Interacción de los componentes de DialogFlow



Fuente: (Google, 2023)

La arquitectura propuesta por Google para la integración de DialogFlow con servicios web es mediante su componente de nube Cloud Functions el cual permite combinar el manejo del flujo de la conversación en el chatbot con los servicios que debe llamar, según sea el caso, en el paso respectivo de dicho flujo conversacional.

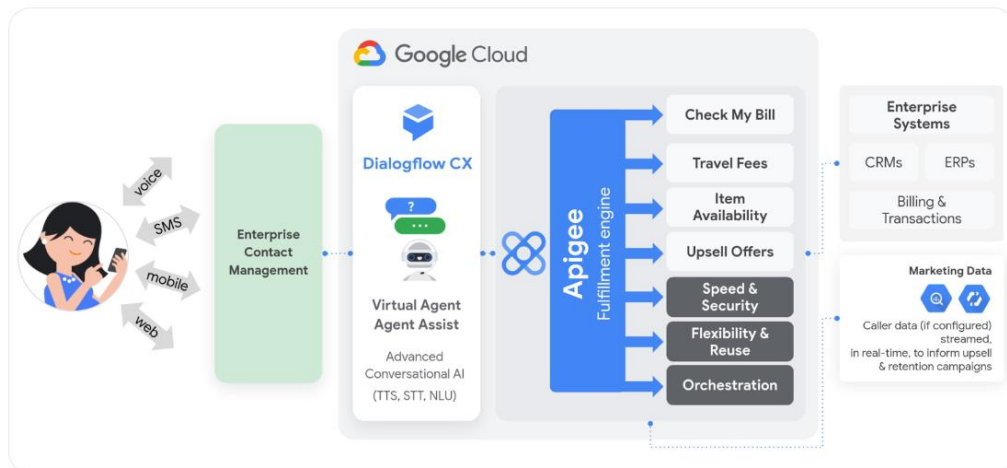
Ilustración 13 Arquitectura de integración de DialogFlow



Fuente: (Ganesch, Akopian, Rhud, & Inupakutika, 2022)

Cloud Functions es un componente que se encuentra en la arquitectura de la nube de Google y se presenta como una característica nativa de la plataforma APIGEE para exponer servicios que se integren o empaten hacia servicios web expuestos en plataformas externas en Internet. A continuación, se muestra, a alto nivel, cómo se relaciona el mundo de DialogFlow con APIGEE para la comunicación con servicios y/o plataformas externas.

Ilustración 14 Integración DialogFlow y APIGEE



Fuente: (Google cloud, 2021)

DialogFlow se presenta como una solución en la nube de tipo SaaS en el cual el cliente paga un costo por solicitud. Es importante definir este término ya que según (Google cloud, 2021) una solicitud se define como “cualquier llamada a la API al servicio Dialogflow, ya sea directa con la API o indirecta con la integración o la consola. Según la tarea y el diseño del agente, el número de solicitudes que se necesitan para que un usuario final complete una tarea con un agente de Dialogflow puede variar en gran medida” (pág. 1). En este caso, cada solicitud tiene un costo de 0,007 USD.

El esquema de costos para un proyecto de integración de DialogFlow con servicios web externos debe contemplar también el costo de la plataforma de APIGEE la cual

2.2 8 Microservicios

Según (Amazon, 2023) Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes.

Las arquitecturas de microservicios hacen que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Esto permite la innovación y acelera el tiempo de

comercialización de las nuevas características.

Arquitectura monolítica en comparación con la arquitectura de microservicio con las arquitecturas monolíticas, todos los procesos están estrechamente asociados y se ejecutan como un solo servicio. Esto significa que, si un proceso de una aplicación experimenta un pico de demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica se vuelve más complejo a medida que crece la base de código. Esta complejidad limita la experimentación y dificulta la implementación de nuevas ideas. Las arquitecturas monolíticas aumentan el riesgo de la disponibilidad de la aplicación porque muchos procesos dependientes y estrechamente vinculados aumentan el impacto del error de un proceso.

Con una arquitectura de microservicios, una aplicación se crea con componentes independientes que ejecutan cada proceso de la aplicación como un servicio. Estos servicios se comunican a través de una interfaz bien definida mediante API ligeras. Los servicios se crean para las capacidades empresariales y cada servicio desempeña una sola función. Debido a que se ejecutan de forma independiente, cada servicio se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación.

2.2.9 Arquitecturas Monolíticas vs Desacopladas

Según (Harrys, 2024) Las arquitecturas monolíticas la conforman los sistemas con baja cohesión y alto acoplamiento, es decir, son sistemas en donde toda su funcionalidad se encuentra desarrollada en un mismo componente, sin división de funciones y que crea un punto único de falla.

El reto de integrar este tipo de sistemas es grande debido a que significaría una parada total del aplicativo para el despliegue de esta nueva funcionalidad; adicional, que dicha integración puede significar la creación o refactorización completa del sistema en sí.

Por otro lado, las arquitecturas llamadas desacopladas han significado un avance en el proceso de integración de sistemas. Son aplicaciones cuya funcionalidad se

encuentra diseminada en varios componentes de la aplicación donde ya se ve división de funciones y en los cuales el desarrollo y despliegue de una integración puede significar la parada solo de una parte del sistema.

En este contexto, han aparecido durante el transcurrir de los años varias arquitecturas que han buscado cumplir con el paradigma de la alta cohesión y bajo acoplamiento como, por ejemplo:

Tabla 7 Evolución de las arquitecturas de sistemas

Arquitectura	Descripción
Cliente – Servidor	En esta arquitectura existe un componente cliente que interactúa con el usuario y envía peticiones hacia el componente servidor; sin embargo, este tipo de arquitectura continúa siendo monolítica
3 capas	En esta arquitectura existe un componente cliente, una capa intermedia con componentes que desacoplan la lógica de integración de la capa de backend o capa servidor. Esta arquitectura ya no es monolítica y significó el aparecimiento del concepto de Servicios Web que no son otra cosa que aplicaciones web sin interfaz de usuario que reciben peticiones y las mapean hacia la capa servidora y luego procesa la respuesta para entregar la misma hacia la capa cliente
N capas	Es una evolución de la arquitectura en 3 capas en la cual se crea un componente o capa por cada responsabilidad que se identifica dentro de un sistema; es decir, se tiene componentes que manejan la interfaz

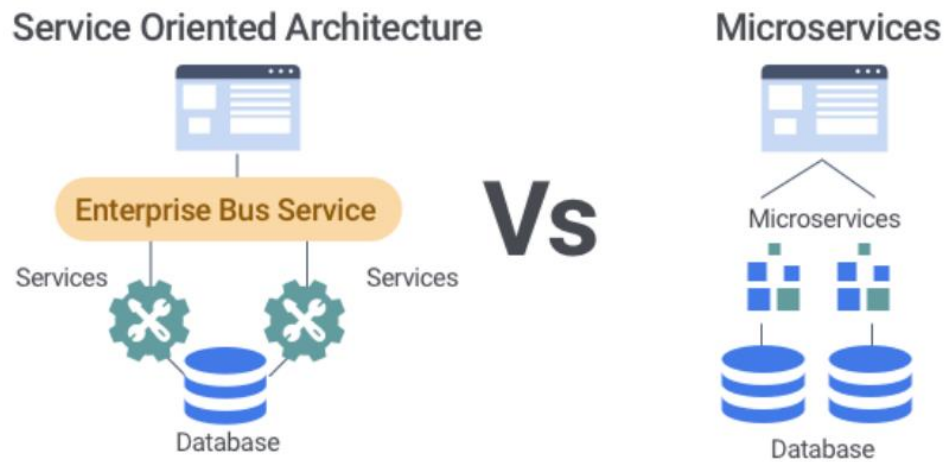
	de usuario, otros la interacción con la base de datos, otros ofrecen servicios especializados, otros manejan los logs, entre otros. En esta Arquitectura se fortaleció el concepto de Servicios Web y aparece el concepto de Microservicios
--	---

Elaborado por: Marco Vinicio Luna Aizaga

Con el apareamiento de servicios web, también se pone de manifiesto el concepto de SOA (Service Oriented Architecture) la cual representa una arquitectura de acoplamiento flexible que permite la integración de sistemas mediante la exposición de interfaces y contratos. Esta arquitectura significó también un reto en cuanto a los protocolos y tecnología creada por los fabricantes para la implementación de SOA. Adicional, este tipo de arquitectura también puso sobre la mesa el uso de componentes orquestadores o intermediarios entre las aplicaciones clientes y los servicios web como, por ejemplo, los Bus de Servicios Empresariales.

Los microservicios son en sí una evolución de los servicios web que cumplen el patrón de diseño de responsabilidad única, es decir, implementan una funcionalidad específica con un ámbito limitado; por ejemplo, mientras anteriormente un servicio web podía encargarse de las transacciones CRUD de un cliente y los productos de este dentro de la empresa; con los microservicios de tendría un microservicio para las transacciones CRUD del cliente y otro microservicio para el manejo de productos de la empresa. En la arquitectura de microservicios la responsabilidad de intermediación desaparece y la orquestación se la puede realizar mediante el uso de otras tecnologías como los API Gateway e incluso los mismos servicios web.

Ilustración 15 SOA vs Microservicios



Fuente: (Hiberus, 2019)

El advenimiento del concepto de microservicios significó el relanzamiento del concepto de API (Application Programming Interface – Interfaz de Programación de aplicaciones) que se refiere a aplicaciones que permiten conectar sistemas.

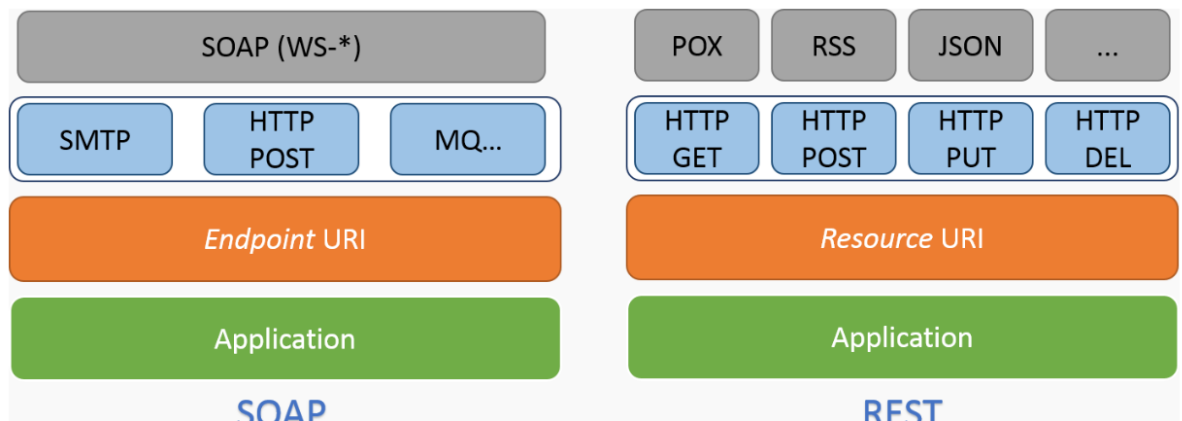
Es muy común que, al referirse a un microservicio, la gente y los fabricantes los nombren como APIs. En este punto es necesario mencionar que, como parte de su herencia de los servicios web, estas APIs se pueden construir de dos formas:

- SOAP (Simple Object Access Protocol): representa un protocolo estándar que define que, para que dos objetos distintos puedan interactuar, lo hagan mediante el intercambio de datos en formato XML⁵.
- REST (Representational State Transfer): representa un estilo arquitectónico en el cual dos objetos pueden interactuar mediante un intercambio de información mediante el uso del protocolo HTTP. En este estilo se puede intercambiar datos mediante mensajes en XML o JSON⁶

⁵ XML: Extensible Markup Language.

⁶ JSON: Javascript Object Notation.

Ilustración 16 SOAP vs REST



Fuente: (Hiberus, 2019)

Para el desarrollo del presente trabajo se utilizará los conceptos y tecnología REST para la creación de los microservicios de integración del Chatbot.

CAPÍTULO III: METODOLOGÍA

3.1. Tipo de Investigación

En la presente investigación se utilizó la técnica cualitativa de la entrevista a un grupo focal de usuarios que manejan el servicio de Gestión de Cambios de Banco General Rumiñahui a fin de conocer el grado de automatización y adopción de tecnologías que den soporte a dicho proceso y determinar cuál es la más idónea de cara a fortalecer la comunicación dentro del proceso de controles de cambio al ambiente productivo del banco. La investigación cualitativa será de tipo exploratoria; es decir, en un formato no estructurado en el que se espera que los entrevistados expresen sus opiniones con libertad.

3.2. Diseño de Investigación

Para el efecto, se ejecutó las siguientes acciones con el fin de obtener la información requerida:

- Definición del grupo focal de los intervinientes en el proceso de Gestión de Cambios de BGR con base a sus responsabilidades dentro del mismo.
- Determinación de las fases más importantes dentro del proceso.
- Mediante el uso de la técnica de Customer Journey Map, determinar los puntos de dolor de los entrevistados.
- Seleccionar de entre los puntos más relevantes, las ideas que pueden ser automatizables mediante el uso de un chatbot.
- Definición de entradas y salidas que podrá ejecutar el chatbot según cada punto seleccionado.
- Determinación de los flujos conversacionales que tendrá el chatbot que incluyan las entradas y salidas identificadas.
- Definición del backlog del proyecto para implementación del chatbot que contenga las historias de usuario tanto para la creación del flujo conversacional como de los distintos microservicios que intervienen en el mismo.
- Definición de los sprints que tendrá el proyecto

- Ejecución de los sprints conforme al marco de referencia SCRUM.

3.3. Población y muestra

En este punto se determinó como única unidad de estudio el área de Gestión de Servicios del Departamento de TI de Banco General Rumiñahui que tiene bajo su cargo la gobernanza del proceso de Gestión de Cambios.

El área tiene un grupo de personas propias del banco encargadas de tareas de gobernanza como medición de los acuerdos de niveles del servicio, sesiones de seguimiento periódicas con los ejecutores del proceso, entre otros.

También tiene una parte tercerizada con un proveedor externo que se encarga de la ejecución de las actividades que indica el proceso de Gestión de Cambios en sí; como, por ejemplo, agendamiento de Comités de Cambio, registro de decisiones del Comité, planificación de los cambios a los ambientes de test y producción, seguimiento del resultado de los cambios según el ambiente, entre otros.

El universo de actores que intervienen dentro del proceso de Gestión de Cambios lo completan los solicitantes de cambio y los ejecutores de los cambios en los ambientes de test y producción del banco.

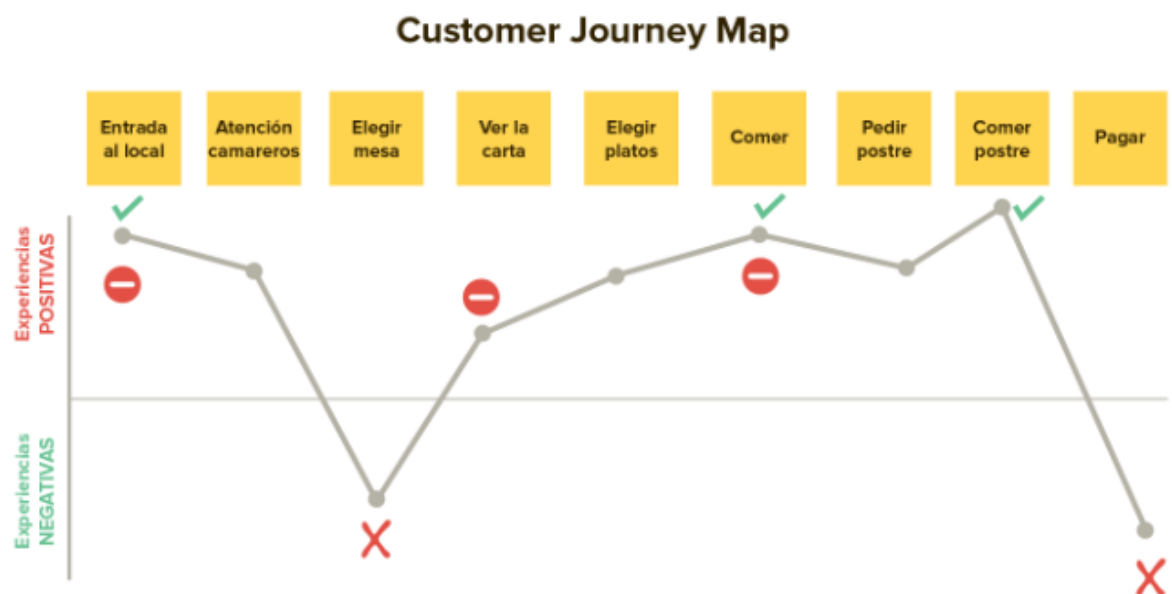
Para el presente trabajo se determinó que las entrevistas se llevarán a cabo al grupo focal constituido por la Subgerencia de Infraestructura de TI del banco, el Jefe de Servicios Transversales del proveedor, un Líder de Proyectos del banco a modo de solicitante de un cambio y uno de los miembros del equipo de Versionamientos y Despliegues del Banco. Es decir, la investigación se aplicará a una muestra.

A todos los entrevistados se les indicará que la entrevista es confidencial y anónima con el fin de no comprometer las respuestas recibidas.

3.4. Técnicas e instrumentos de recolección de datos

En esta fase se procede con la ejecución de las entrevistas de manera exploratoria como se indicó anteriormente; sin embargo, el resultado de las mismas quedarán plasmadas en un Customer Journey Map el cual consiste de un diagrama de dos dimensiones donde en el eje de las X se colocan las fases más importantes del proceso de Gestión de Cambios (momentos de verdad) y en el eje de las Y se colocan el sentir de cada actor respecto a la fase definiéndolo en términos de experiencias positivas y negativas mismas que han sido identificadas durante las entrevistas:

Ilustración 17 Ejemplo de Customer Journey Map



Fuente: (Mazza, 2022)

Vale la pena enfatizar que se creará un Customer Journey Map por cada actor del grupo focal entrevistado y se adicionará en el eje de las X las ideas más relevantes expuestas por cada entrevistado para mejorar su experiencia de acuerdo con cada fase del proceso sea que intervienen o no.

3.5. Técnica de Análisis de Datos

Una vez que se tiene el Customer Journey Map de cada entrevistado, se selecciona las experiencias negativas y se analiza las ideas de mejora expuestas por el entrevistado en dichas fases.

Para cada una de las ideas seleccionadas se determina si es automatizable o no mediante el uso de un Chatbot y con las que se identifiquen se procede a crear el backlog del proyecto.

Tabla 8 Customer Journey Map de los entrevistados

ENTREVISTADO:	Alejando Fernandez						
EMPRESA:	Banco General Rumiñahui						
ROL	Subgerente de Servicios de Infraestructura y Redes						
RESPONSABILIDAD:	Responsable de la producción del banco, es miembro del CAB (Comité Asesor de Cambios) y aprueba los versionamientos a producción						
	Ingreso Control de Cambios	Revisión de Control de Cambios	Socialización del Cambio	Autorización del Cambio	Planificación del Cambio	Ejecución del Cambio	Cierre del Cambio
Experiencia Positiva	X	X		X			
Experiencia Negativa			X		X	X	X
	Principales ideas						
	- He ingresado algunos cambios y la herramienta utilizada es amigable, el flujo es fácil de entender pero existen campos que no son usados	- No realizo la revisión de los cambios individuales, pero no he tenido quejas de los solicitantes o de sus líneas de supervisión	- No existe forma de conocer un calendario de socializaciones de cambios a fin de que la autorización se agilice; es decir, debe existir forma de conocer en qué fecha y hora se tiene planificado socializar un cambio a fin de reservar la agenda	- Generalmente la autorización la hago en la herramienta de controles de cambio comprobando que esté subida el acta del CAB	- Solo tenemos el chat de WhatsApp para conocer los cambios aprobados y planificados, pero es difícil buscar información sobre aprobaciones de CAB pasados.	- De la misma manera, es difícil buscar en el WhatsApp la información sobre el resultado de la ejecución de los cambios	- Debo conocer de primera mano si un control de cambio ya está cerrado o no. El ingresar a la herramienta y consultarlo me toma mucho tiempo
	¿Es automatizable?						
			SI		SI	SI	SI

ENTREVISTADO:	Luis Rodriguez
EMPRESA:	Tata Consultancy Services
ROL	Jefe de Servicios Transversales
RESPONSABILIDAD:	Responsable de la ejecución de las distintas fases del proceso de Gestión de Cambios, elaboración de los reportes de cumplimiento de los Acuerdos de Niveles del Servicio, presentación de resultados, ejecución de planes de acción para la mejora del servicio

	Ingreso Control de Cambios	Revisión de Control de Cambios	Socialización del Cambio	Autorización del Cambio	Planificación del Cambio	Ejecución del Cambio	Cierre del Cambio
Experiencia Positiva	X				X	-	X
Experiencia Negativa		X	X	X		X	
Principales ideas							
	- No realizo ingresos de controles de cambio, mi rol no me lo permite	- Los revisores de los controles de cambio deber tener un poco más de finura en la revisión, no analizan bien los riesgos de los cambios	- Se debería tener donde consultar las fechas y horarios en que se han pactado las socializaciones	- El acta de los CAB se suben a la herramienta de controles de cambio, pero se debería tener una forma de consulta de la autorización de primera mano	- Actualmente se informa vía WhatsApp de manera general los cambios planificados	- Se debe complementar lo que se reporta vía WhatsApp sobre el resultado de la ejecución, con una consulta específica por cambio para conocer incluso la bitácora de la ejecución y saber si tuvieron dificultades o no	- No hemos tenido problemas con esta parte del flujo
¿Es automatizable?							
		NO	SI	SI		SI	

ENTREVISTADO:	Paul García
EMPRESA:	Banco General Rumiñahui
ROL	Scrum Master
RESPONSABILIDAD:	Responsable de la ejecución de los proyectos de BGR asignados

	Ingreso Control de Cambios	Revisión de Control de Cambios	Socialización del Cambio	Autorización del Cambio	Planificación del Cambio	Ejecución del Cambio	Cierre del Cambio
Experiencia Positiva	X	X				X	X
Experiencia Negativa			X	X	X		
Principales ideas							
	- El ingreso de los controles de cambio es fácil de entender y utilizar	- Mi línea de supervisión no ha tenido dificultades en colocar en la herramienta la firma de revisión	- Es difícil cuadrar la agenda de todos los autorizadores para realizar la socialización del cambio, quizá si se determina fechas y horarios tentativos para poder seleccionar uno y planificar de acuerdo con dicha agenda	- Hay autorizadores que no llegan a la socialización y recién se enteran del cambio al momento de la autorización	- Existe dos herramientas donde se puede consultar cómo fue calendarizado un cambio, pero no se tiene certeza cuál de las dos es la oficial, se debería tener un método de consulta independiente de la herramienta	- No he tenido inconvenientes	- No he tenido inconvenientes
¿Es automatizable?							
			NO	NO	SI		

ENTREVISTADO:	Xavier Albán						
EMPRESA:	Banco General Rumiñahui						
ROL	Oficial de Librarian						
RESPONSABILIDAD:	Responsable del versionamiento y despliegue de los controles de cambio a los ambientes de test y producción asignados						
	Ingreso Control de Cambios	Revisión de Control de Cambios	Socialización del Cambio	Autorización del Cambio	Planificación del Cambio	Ejecución del Cambio	Cierre del Cambio
Experiencia Positiva	X	X	X	X		-	X
Experiencia Negativa					X	X	
	Principales ideas						
	- No realizo ingresos de controles de cambio	- No intervengo en esta parte del flujo	- Intervengo en las socializaciones a manera de oyente, pero no tengo ni voz ni voto	- Participo en los CAB, pero no intervengo, solo hago ciertas preguntas	- Actualmente tenemos dos herramientas, deberíamos tener solo una y de la cual se pueda extraer la información para mostrar a los usuarios	- Los resultados de la ejecución solo los conocen los grupos resolutorios y no se registra en ningún lado para una consulta posterior.	- No participo en el cierre
	¿Es automatizable?						
					SI	SI	

Elaborado por: Marco Vinicio Luna Aizaga

3.6. Operacionalización de Variables

Tabla 9 Operacionalización de Variables

VARIABLE	DEFINICIÓN	DIMENSIONES	INDICADORES	TÉCNICA	INSTRUMENTO
GESTIÓN DE UN PROYECTO ÁGIL PARA LA INTEGRACIÓN DE ASISTENTE VIRTUAL CON MICROSERVICIOS	El marco de referencia utilizado para la gestión del proyecto es SCRUM la cual se ejecuta mediante la creación del backlog del proyecto, la definición de los Sprints del mismo y la ejecución de estos con sus respectivas ceremonias	BACKLOG EL PROYECTO	Historias de usuario no épicas con definición de terminado	<p>Estructura de historia de usuario basada en las tres partes:</p> <ul style="list-style-type: none"> Yo (nombre de la persona o rol) Requiero (especificación funcional) Para (justificación del requerimiento) <p>Además, la historia de usuario deberá contener los puntos de historia estimados</p>	Customer Journey Map
		SPRINTS DEL PROYECTO	Número de historias de usuario planificadas para el sprint ejecutadas en su totalidad	Priorización de las historias de usuario y organización dentro de cada sprint	
			Baja o ninguna deuda técnica de sprint a sprint	Conteo de historias de usuario no ejecutadas en su totalidad y que requieran	

				pasarse para el siguiente sprint	
--	--	--	--	----------------------------------	--

Elaborado por: Marco Vinicio Luna Aizaga

CAPÍTULO IV: PRESENTACIÓN Y ANÁLISIS DE DATOS

4.1. Selección de ideas más destacadas

Luego de las entrevistas realizadas y su resultado aplicado a cada Customer Journey Map de los entrevistados, se tiene los siguientes detalles:

Tabla 10 Puntos más destacados de las entrevistas

Entrevistado	A) Número de experiencias negativas en momentos de verdad	B) Número de actividades automatizables en de acuerdo con el punto A)	Puntos más destacados durante la entrevista	Punto de mayor dolor identificado en la entrevista
Alejandro Fernandez	4	4	<ul style="list-style-type: none"> Proceso de búsqueda de información histórica debe ser más ágil El mecanismo de autorización podría mostrar información relevante en resumen para poder analizar de mejora manera el cambio 	<ul style="list-style-type: none"> Proceso de búsqueda de información histórica
Luis Rodriguez	4	3	<ul style="list-style-type: none"> Los autorizadores en las socializaciones envían a delegados que 	<ul style="list-style-type: none"> Los autorizadores deben revisar mejor los cambios en la

			<p>luego no les informan adecuadamente de los cambios y eso provoca reprocesos.</p> <ul style="list-style-type: none"> • El análisis del cambio se lo debe hacer con mayor finura desde el ámbito de acción de cada autorizador 	<p>socialización ya que llegan al CAB recién a enterarse de lo que se busca con el cambio y se han dado casos en los que los solicitantes les niegan el cambio por desconocimiento</p>
Paúl García	3	1	<ul style="list-style-type: none"> • Se debe tener predefinida fechas y horas para las socializaciones. • Quienes van a la socialización deben distribuir el conocimiento a la interna de su equipo para que, quien aprueba en el CAB, lo haga con conocimiento de causa y no provoquen pérdidas de tiempo 	<ul style="list-style-type: none"> • Los retrabajos por no tener una socialización efectiva por varios factores como el cuadro de agendas, el nivel de las personas que asisten a la socialización, entre otros.
Xavier Alban	2	2	<ul style="list-style-type: none"> • Que existan dos herramientas de planificación del cambio • Que todavía hay solicitantes de 	<ul style="list-style-type: none"> • Que no se tenga de primera mano y al instante el cierre de un Cambio luego de la comprobación

			<p>cambios que envían información por correo.</p> <ul style="list-style-type: none"> • Que los habilitantes del cambio se deben copiar tanto en un SFTP como en la herramienta de Controles de Cambio lo cual provoca duplicidad 	en producción.
--	--	--	---	----------------

Elaborado Por: Marco Vinicio Luna Aizaga

Complementando lo identificado en las entrevistas y que se muestra en la tabla anterior, se presenta, a continuación, las ideas más destacadas de acuerdo con cada fase del proceso catalogadas como momentos de verdad dentro del flujo:

Tabla 11 Principales ideas seleccionadas

Ingreso Control de Cambios	Revisión de Control de Cambios	Socialización del Cambio	Autorización del Cambio	Planificación del Cambio	Ejecución del Cambio	Cierre del Cambio
No se propusieron ideas para este paso del proceso.	La única propuesta en este paso no es susceptible de automatización	Se debería tener una agenda donde consultar las fechas y horarios en que se han pactado las socializaciones	Se debería tener una forma de consulta de la autorización de primera mano	Se debería tener un mecanismo de consulta de la planificación de los cambios aprobados en el CAB	Se debería tener un mecanismo de consulta del resultado de la ejecución de los cambios; incluso si	Se debería tener un mecanismo de consulta sobre el estado del cambio para conocer si el mismo ya fue cerrado.

					fueron emergentes en las madrugadas	
--	--	--	--	--	--	--

Elaborado por: Marco Vinicio Luna Aizaga

4.2. Backlog del proyecto con base a las ideas seleccionadas

A continuación, se exponen las distintas historias de usuario que formarán parte del backlog del proyecto, los valores de estimación están expresados en puntos de historia (PH) con una equivalencia de 1 punto de historia igual a 1 hora hombre.

Adicional, se contempla la reutilización del repositorio actual de controles de cambios para obtener la información respectiva.

Tabla 12 Historias de usuario identificadas

HISTORIA DE USUARIO: US001		
COMO	Solicitante de un control de cambio	
QUIERO	Quiero un menú conversacional con un chatbot	
PARA	Realizar las consultas de fechas de socialización de cambios, controles de cambios autorizados o no, planificación de un control de cambio, resultado de la ejecución de un control de cambios y estado de un control de cambio	
VALIDACIONES: La historia de usuario estará completa cuando se pueda visualizar el menú conversacional en el chat bot y que se pueda acceder a cada una de las consultas descritas. Se considera como caso positivo de certificación el que aparezca la información consultada al microservicio; en caso de que el microservicio se encuentra fuera de servicio, se debe mostrar un mensaje genérico.	VALOR:	No aplica, desarrollo interno
	PRIORIDAD:	1
	ESTIMACIÓN:	80 PH
HISTORIA DE USUARIO: US002		
COMO	Solicitante de un control de cambio	
QUIERO	Requiero la construcción de un microservicio	
PARA	Realizar la consulta de fechas planificadas de socialización de cambios	
VALIDACIONES: La historia de usuario estará completa cuando se pueda ejecutar la consulta al microservicio	VALOR:	No aplica, desarrollo interno
	PRIORIDAD:	1

desde la opción del menú conversacional del chatbot. Se debe considerar como caso negativo de certificación, el simular una caída del micro servicio a fin de que se muestre un mensaje genérico al cliente		ESTIMACIÓN:	40 PH
HISTORIA DE USUARIO: US003			
COMO	Solicitante de un control de cambio		
QUIERO	Requiero la construcción de un microservicio		
PARA	Realizar la consulta de controles de cambios autorizados o no		
VALIDACIONES: La historia de usuario estará completa cuando se pueda ejecutar la consulta al microservicio desde la opción del menú conversacional del chatbot. Se debe considerar como caso negativo de certificación, el simular una caída del microservicio a fin de que se muestre un mensaje genérico al cliente		VALOR:	No aplica, desarrollo interno
		PRIORIDAD:	1
		ESTIMACIÓN:	40 PH
HISTORIA DE USUARIO: US004			
COMO	Solicitante de un control de cambio		
QUIERO	Requiero la construcción de un microservicio		
PARA	Realizar la consulta de la planificación de un control de cambio		
VALIDACIONES: La historia de usuario estará completa cuando se pueda ejecutar la consulta al microservicio desde la opción del menú conversacional del chatbot. Se debe considerar como caso negativo de certificación, el simular una caída del microservicio a fin de que se muestre un mensaje genérico al cliente		VALOR:	No aplica, desarrollo interno
		PRIORIDAD:	<u>1</u>
		ESTIMACIÓN:	30 PH
HISTORIA DE USUARIO: US005			
COMO	Solicitante de un control de cambio		

QUIERO	Requiero la construcción de un microservicio	
PARA	Realizar la consulta de el resultado de la ejecución de un control de cambios	
VALIDACIONES: La historia de usuario estará completa cuando se pueda ejecutar la consulta al microservicio desde la opción del menú conversacional del chatbot. Se debe considerar como caso negativo de certificación, el simular una caída del microservicio a fin de que se muestre un mensaje genérico al cliente	VALOR:	No aplica, desarrollo interno
	PRIORIDAD:	<u>1</u>
	ESTIMACIÓN:	30 PH
HISTORIA DE USUARIO: US006		
COMO	Solicitante de un control de cambio	
QUIERO	Requiero la construcción de un microservicio	
PARA	Realizar la consulta de el estado de un control de cambio	
VALIDACIONES: La historia de usuario estará completa cuando se pueda ejecutar la consulta al microservicio desde la opción del menú conversacional del chatbot. Se debe considerar como caso negativo de certificación, el simular una caída del microservicio a fin de que se muestre un mensaje genérico al cliente	VALOR:	No aplica, desarrollo interno
	PRIORIDAD:	<u>1</u>
	ESTIMACIÓN:	20 PH

Elaborado por: Marco Vinicio Luna Aizaga

4.3. Planificación de Sprints

Una vez que tenemos las historias de usuario, se procede con la planificación de los Sprints colocando las mencionadas historias dentro de cada Sprint.

Tabla 13 Planificación de los Sprints

SPRINT	DURACION (semanas)	SPRINT BACKLOG	SEM 1	SEM 2	SEM 3	SEM 4	SEM 5	SEM 6
SPRINT 1	2	HU002, HU003						
SPRINT 2	2	HU004, HU005, HU006						
SPRINT 3	2	HU001						

Elaborado por: Marco Vinicio Luna Aizaga

La estructura del sprint en general es la siguiente:

Tabla 14 Estructura del Sprint

ACTIVIDAD	NUMERO RECURSOS	DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7	DIA 8	DIA 9	DIA 10
Presentación del diseño	3										
Desarrollo	2										
Despliegue a test y Certificación	1										
Despliegue a producción	1										

Elaborado por: Marco Vinicio Luna Aizaga

CAPÍTULO V: PRESENTACIÓN DE LA PROPUESTA

En este capítulo se presenta las actividades de gestión y resultados obtenidos como fruto de la ejecución de los Sprints definidos para el proyecto de integración del chatbot y el microservicio de consulta de información de controles de cambio.

En principio se describe brevemente la historia del Banco General Rumiñahui, su misión, visión y valores culturales sobre los cuales se enmarca el desarrollo del proyecto:

BGR nació en 1988 como respuesta a las necesidades financieras de las Fuerzas Armadas del Ecuador. Desde entonces evolucionó de ser un Banco Pequeño a ser el "Mejor Banco Mediano del país". El desarrollo alcanzado por BGR refleja una entidad financiera seria y responsable que brinda seguridad y confianza, capaz de administrar los recursos del público con transparencia y honestidad. Ese mismo año, BGR contó con el apoyo de socios fundadores como la Dirección de Industrias del Ejército (DINE), Dirección General de Aviación Civil (DAC), Astinave y militares en servicio activo que depositaron su confianza en un proyecto nuevo, sólido y enfocado en la generación de resultados. (Banco General Rumiñahui, 2024).

Tabla 15 Información relevante de Banco General Rumiñahui

DATO	INFORMACIÓN
MISIÓN	Proporcionar bienestar y progreso a nuestros clientes, colaboradores, accionistas y contribuir al desarrollo del país
VISIÓN	Ser un banco de excelencia, al servicio de nuestros clientes, en la construcción de sus sueños
VALORES	
Pasión por trascender	Somos mentores de cambio Cuidamos de nuestra gente, acompañamos e impulsamos el cambio

Nos esforzamos y creamos valor	Somos ágiles e innovadores: Nos aventuramos a probar nuevos caminos para superar las expectativas de nuestros clientes.
Responsables y coherentes	Asumimos nuestras responsabilidades: Tomamos decisiones con responsabilidad asumiendo riesgos controlados.
Pasión por trascender	Nos enfocamos en nuestros clientes para generar experiencias únicas: Generamos procesos sencillos, ágiles y eficientes centrados en el cliente.
El equipo es nuestra fortaleza	Somos un solo equipo: Nos comunicamos asertivamente; confiamos en el otro, y juntos encontramos una solución.

Fuente: (Banco General Rumiñahui, 2024)

Elaborado por: Marco Vinicio Luna Aizaga

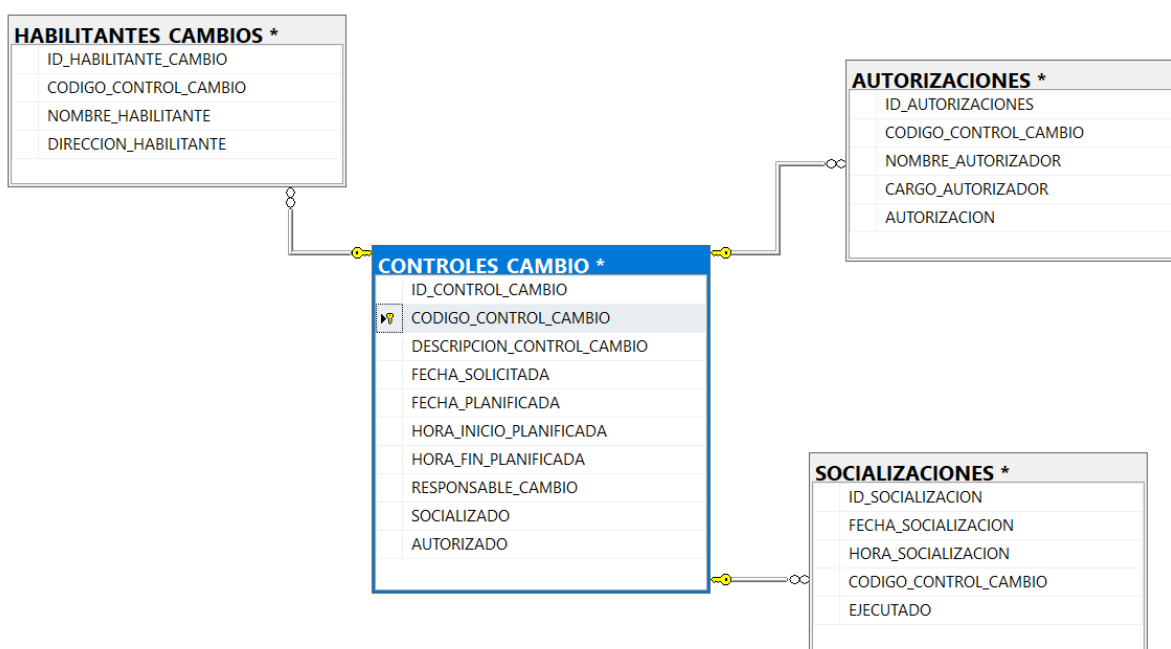
En Banco General Rumiñahui la herramienta para el manejo del proceso de Gestión de Cambios se llama Proactivanet ofrecida como un SaaS; en este sentido, existe dos formas para la consulta de información: la primera forma de acceso a la misma es desde fuera de su FrontEnd mediante una capa de APIs del tipo REST con un conjunto de métodos de consulta predefinidos; la segunda es a través de la descarga en archivos de MS Excel de las consultas predefinidas en el FrontEnd.

Dicha herramienta está construida sobre los conceptos de ITIL V4 y por tanto no contiene información en sí de socializaciones; además, no permite la integración directa con la tecnología mencionada anteriormente para la construcción de Chatbots.

Es por estas circunstancias que se ha determinado la necesidad de volcar la información desde la plataforma Proactivanet hacia una base de datos local para que la misma sea el repositorio desde donde se consulte la información requerida en los HU anteriormente descritas.

De esta manera la descarga y carga de esta información se la hace de manera manual y no forma parte del alcance del presente trabajo; sin embargo, se describe a continuación el esquema de la base de datos en mención:

Ilustración 18 Repositorio de información para consulta de Controles de Cambio



Elaborado por: Marco Vinicio Luna Aizaga

5.1. Ejecución de Sprints

A continuación, se describe las actividades ejecutadas como parte de los distintos Sprints definidos para el proyecto.

5.5.1. Primer Sprint

En las siguientes tablas se muestra el resultado de la ejecución de cada historia de usuario del sprint donde se ve el número de puntos de historia (PH) planificados versus los reales ejecutados y un indicador de velocidad que se muestra semaforizado mostrando en verde si los PH planificados fueron menores que los PH planificados, amarillo si fueron iguales y rojo si lo ejecutado fue mayor que lo planificado.

Ejecución Historia de Usuario 2 (HU002):

Como se puede apreciar en la tabla y gráfico del resultado de la ejecución de la HU, la planificación de esta no fue eficiente dado que la diferencia entre los PH

planificados y reales difieren en un PH; esto significa que hay oportunidades de mejora en la estimación.

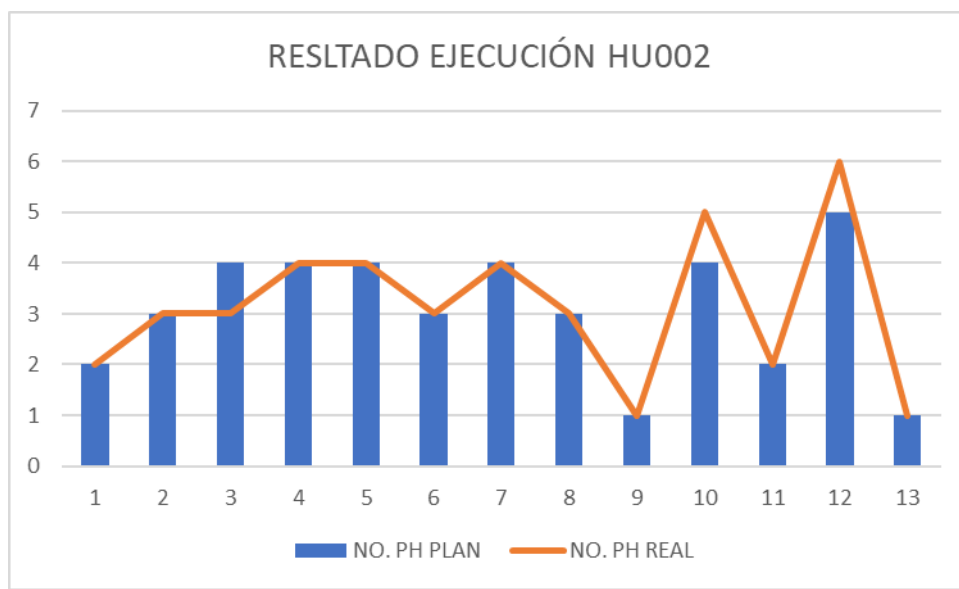
En cuanto al gráfico comparativo de la ejecución de la HU, se ratifica las oportunidades de mejora en la estimación dada la diferencia entre lo planificado y lo real tarea por tarea.

Tabla 16 Resumen ejecución historia de usuario

CODGO HU	HU002						
DESCRIPCION	Microservicio para realizar la consulta de fechas planificadas de socialización de cambios						
CODIGO TAREA	DETALLE TAREA	DIA	NO. PH PLAN	NO. PH REAL	VELOCIDAD	RESPONSABLE	ESTADO
TSK001	Reunión para establecer tareas técnicas	1	2	2		0 Product Owner Scrum Master Scrum Team	COMPLETADA
TSK002	Reunión de explicación del diseño	1	3	3		0 Scrum Master Scrum Team Arquitecto	COMPLETADA
TSK003	Creación de Store Procedures para consulta de socializacion	1	4	3		1 Scrum Team	COMPLETADA
TSK004	Creación de la capa de AD del microservicio	2	4	4		0 Scrum Team	COMPLETADA
TSK005	Creación capa de negocio microservicio	2	4	4		0 Scrum Team	COMPLETADA
TSK006	Creación de endpoints del microservicio	3	3	3		0 Scrum Team	COMPLETADA
TSK007	Pruebas de integracion capas y bdd	3	4	4		0 Scrum Team	COMPLETADA
TSK008	Revisión de código	4	3	3		0 Scrum Team Arquitecto	COMPLETADA
TSK009	Ejecución de pipeline de devops test	4	1	1		0 Scrum Team Librarian	COMPLETADA
TSK010	Elaboración de guión de pruebas	4	4	5		-1 Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK011	Ejecución pruebas sanity	5	2	2		0 Ingeniero QA	COMPLETADA
TSK012	Ejecución pruebas BUAT	5	5	6		-1 Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK013	Ejecución pipeline devops producción	5	1	1		0 Scrum Master Scrum Team Librarian	COMPLETADA

Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 19 Comparativo de ejecución de la HU002



Elaborado por: Marco Vinicio Luna Aizaga

Ejecución Historia de Usuario 3 (HU003):

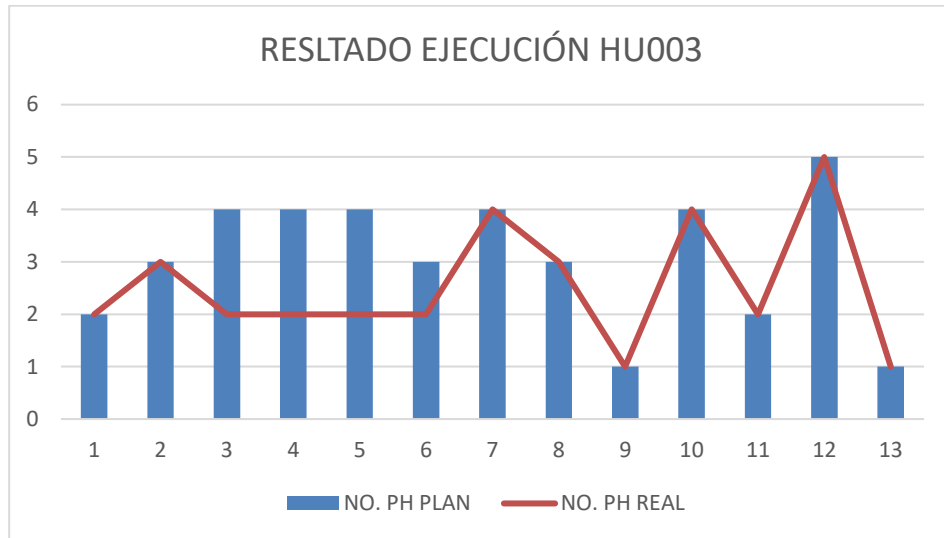
En esta HU se denota una aceleración en la ejecución de las tareas de desarrollo en donde las PH ejecutadas son menores a los planificados; sin embargo, se mantienen aún errores en la estimación lo cual denota la intención del equipo SCRUM de tratar de recuperarse de lo ocurrido en el HU anterior.

Tabla 17 Resumen ejecución historia de usuario

CODGO HU	HU003						
DESCRIPCION	Microservicio para realizarla consulta de controles de cambios autorizados o no						
CODIGO TAREA	DETALLE TAREA	DIA	NO. PH PLAN	NO. PH REAL	VELOCIDAD	RESPONSABLE	ESTADO
TSK001	Reunión para establecer tareas técnicas	6	2	2		Product Owner Scrum Master Scrum Team	COMPLETADA
TSK002	Reunión de explicación del diseño	6	3	3		Scrum Master Scrum Team Arquitecto	COMPLETADA
TSK003	Creación de Store Procedures para consulta de autorizaciones de controles de cambio	6	4	2		Scrum Team	COMPLETADA
TSK004	Creación de la capa y clase de AD del microservicio	7	4	2		Scrum Team	COMPLETADA
TSK005	Creación capa y clase de negocio microservicio	7	4	2		Scrum Team	COMPLETADA
TSK006	Creación de endpoint del microservicio	7	3	2		Scrum Team	COMPLETADA
TSK007	Pruebas de integracion capas y bdd	8	4	4		Scrum Team	COMPLETADA
TSK008	Revisión de código	8	3	3		Scrum Team Arquitecto	COMPLETADA
TSK009	Ejecución de pipeline de devops test	9	1	1		Scrum Team Librarian	COMPLETADA
TSK010	Elaboración de guión de pruebas	9	4	4		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK011	Ejecución pruebas sanity	9	2	2		Ingeniero QA	COMPLETADA
TSK012	Ejecución pruebas BUAT	10	5	5		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK013	Ejecución pipeline devops producción	10	1	1		Scrum Master Scrum Team Librarian	COMPLETADA

Elaborado por: Marco Vinicio Luna Aizaga

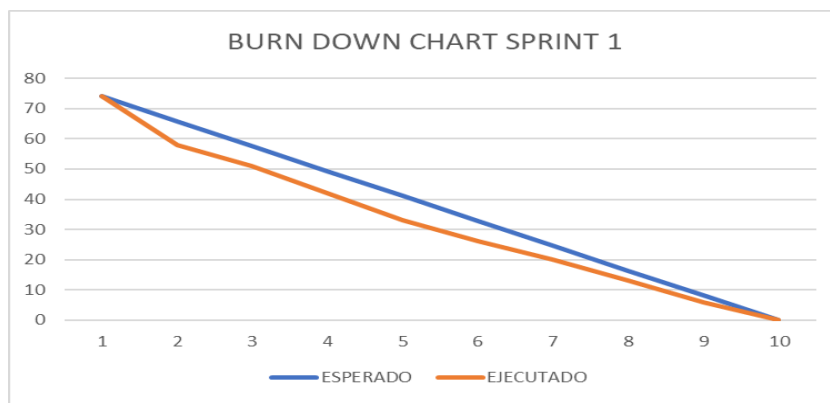
Ilustración 20 Comparativo de ejecución de la HU003



Elaborado por: Marco Vinicio Luna Aizaga

Analizando el desarrollo general del sprint, se nota una alta velocidad en la ejecución de este como se muestra en el diagrama Burn Down Chart del sprint; donde casi todo el sprint siempre se estuvo por debajo de la línea mostrando que se buscó completar el trabajo en el menor tiempo posible.

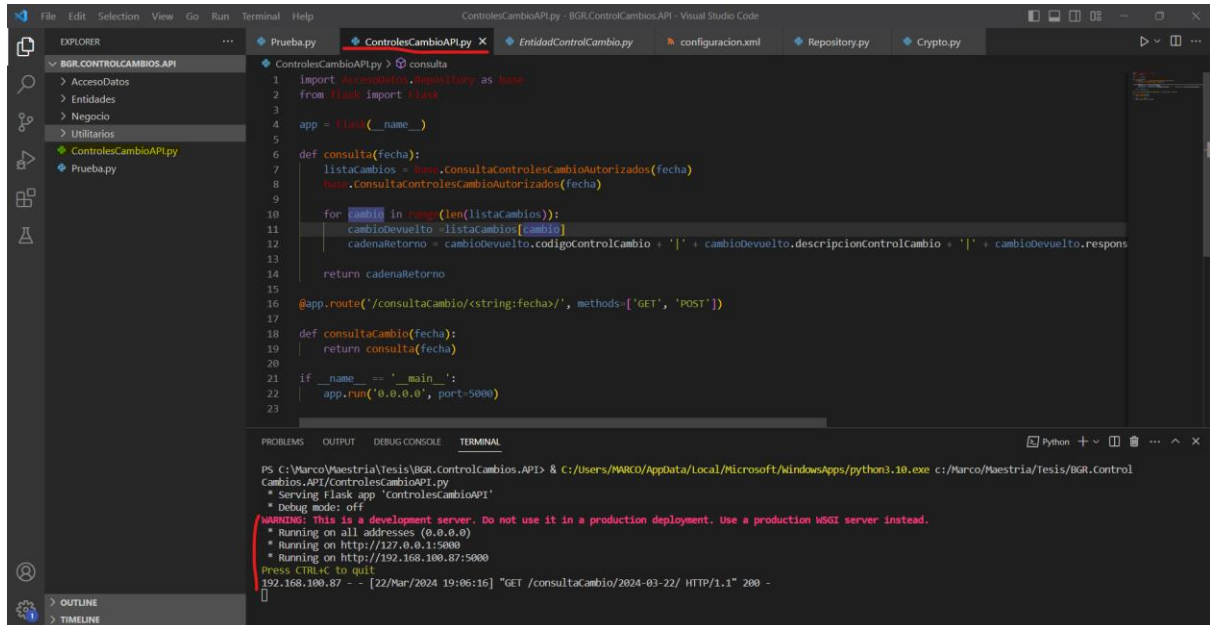
Ilustración 21 Burn Down Chart Sprint 1



Elaborado por: Marco Vinicio Luna Aizaga

Como entregable de la ejecución del sprint se tiene el API REST para consulta de controles de cambio autorizados como se puede ver en la siguiente imagen:

Ilustración 22 API REST para la consulta de Controles de Cambio autorizados



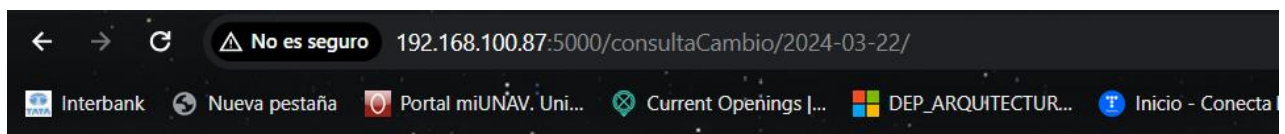
```
1 import sys, os, flask, repository as base
2 from flask import Flask
3
4 app = Flask(__name__)
5
6 def consulta(fecha):
7     listaCambios = base.consultaControlesCambioAutorizados(fecha)
8     base.consultaControlesCambioAutorizados(fecha)
9
10    for cambio in range(len(listaCambios)):
11        cambioDevuelto = listaCambios[cambio]
12        cadenaRetorno = cambioDevuelto.codigoControlCambio + '|' + cambioDevuelto.descripcionControlCambio + '|' + cambioDevuelto.respons
13
14    return cadenaRetorno
15
16 @app.route('/consultaCambio/<string:fecha>', methods=['GET', 'POST'])
17
18 def consultaCambio(fecha):
19     return consulta(fecha)
20
21 if __name__ == '__main__':
22     app.run('0.0.0.0', port=5000)
23
```

```
PS C:\Marco\Maestria\Tesis\BGR-ControlCambios-API> & C:/Users/MARCO/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Marco/Maestria/Tesis/BGR.ControlCambios-API/controlCambioAPI.py
* Serving Flask app 'controlCambioAPI'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.100.87:5000
Press CTRL+C to quit
192.168.100.87 - - [22/Mar/2024 19:06:16] "GET /consultaCambio/2024-03-22/ HTTP/1.1" 200 -
```

Elaborado por: Marco Vinicio Luna Aizaga

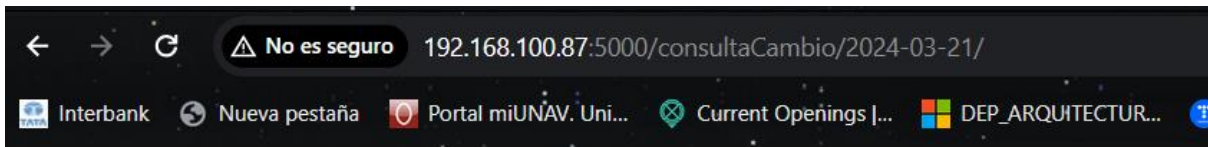
A continuación, se muestra el API REST en ejecución:

Ilustración 23 Ejecución exitosa del API REST de consulta de Controles de Cambios Autorizados



Elaborado por: Marco Vinicio Luna Aizaga

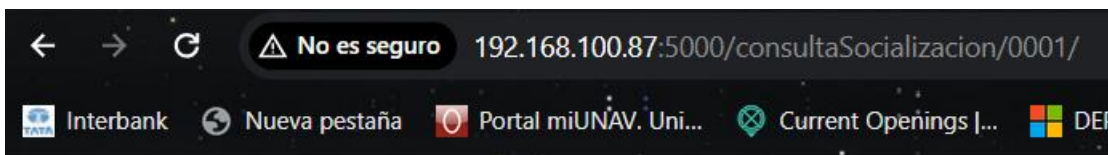
Ilustración 24 Ejecución fallida del API REST de consulta de Controles de Cambios Autorizados



No existe controles de cambio autorizados en la fecha enviada

Elaborado por: Marco Vinicio Luna Aizaga

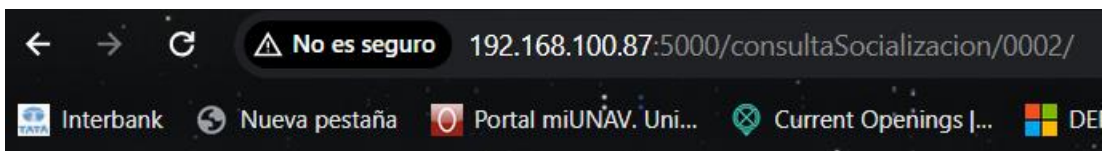
Ilustración 25 Ejecución exitosa del API REST de consulta de Socializaciones



RFC-2023-000001|2024-03-20|16:00:00.0000000

Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 26 Ejecución fallida del API REST de consulta de Socializaciones



No existe socializaciones para el código de cambio enviado

Elaborado por: Marco Vinicio Luna Aizaga

5.5.2. Segundo Sprint





Al igual que en el sprint anterior, a continuación, se tiene las tablas donde se muestra el resultado de la ejecución de cada historia de usuario y en donde se aprecia el número de puntos de historia (PH) planificados versus los reales ejecutados y un indicador de velocidad que se muestra semaforizado mostrando en verde si los PH planificados fueron menores que los PH planificados, amarillo si fueron iguales y rojo

si lo ejecutado fue mayor que lo planificado.

Ejecución Historia de Usuario 4 (HU004):

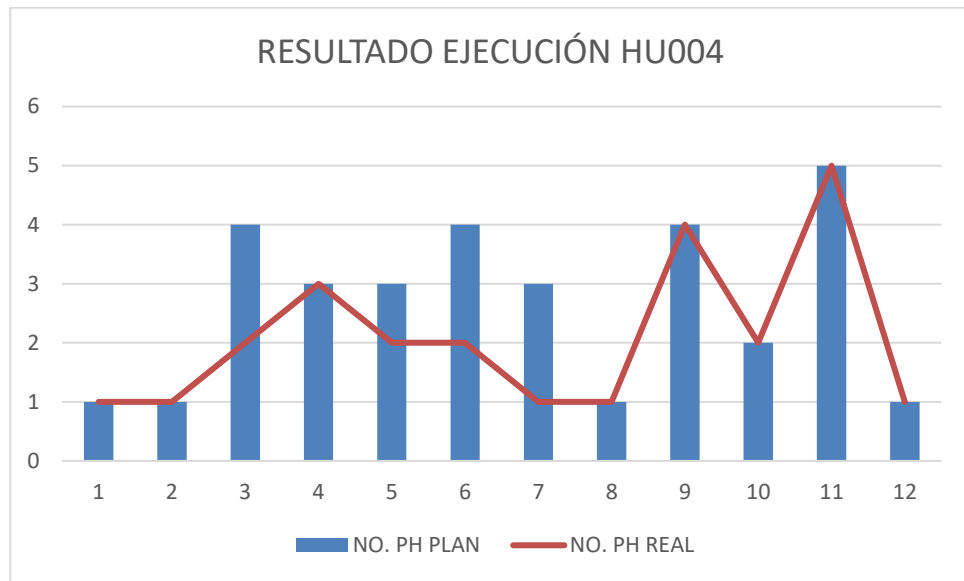
En ésta ya no se aprecia retrasos; de hecho, el equipo SCRUM ha acelerado en las tareas de desarrollo pero que denota una sobre estimación de historias de usuario lo cual denota aún oportunidades de mejora en la estimación.

Tabla 18 Resumen ejecución historia de usuario

CODGO HU	HU004					
DESCRIPCION	Realizar la consulta de la planificación de un control de cambio					
CODIGO TAREA	DETALLE TAREA	NO. PH PLAN	NO. PH REAL	VELOCIDAD	RESPONSABLE	ESTADO
TSK001	Reunión para establecer tareas técnicas	1	1		Product Owner Scrum Master Scrum Team	COMPLETADA
TSK002	Reunión de explicación del diseño	1	1		Scrum Master Scrum Team Arquitecto	COMPLETADA
TSK003	Creación de Store Procedures para consulta de los campos de planificación del cambio	4	2		Scrum Team	COMPLETADA
TSK004	Creación de método en la capa de AD del microservicio	3	3		Scrum Team	COMPLETADA
TSK005	Creación de endpoints del microservicio	3	2		Scrum Team	COMPLETADA
TSK006	Pruebas de integracion capas y bdd	4	2		Scrum Team	COMPLETADA
TSK007	Revisión de código	3	1		Scrum Team Arquitecto	COMPLETADA
TSK008	Ejecución de pipeline de devops test	1	1		Scrum Team Librarian	COMPLETADA
TSK009	Elaboración de guión de pruebas	4	4		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK010	Ejecución pruebas sanity	2	2		Ingeniero QA	COMPLETADA
TSK011	Ejecución pruebas BUAT	5	5		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK012	Ejecución pipeline devops producción	1	1		Scrum Master Scrum Team Librarian	COMPLETADA

Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 27 Comparativo de ejecución de la HU004



Elaborado por: Marco Vinicio Luna Aizaga

Ejecución Historia de Usuario 5 (HU005):

En ésta HU se aprecia una ejecución más eficiente; es decir, los PH planificados coinciden en su mayoría con los PH panificados, lo cual demuestra que el equipo ganó en finura en la estimación.

Tabla 19 Resumen ejecución historia de usuario

CODGO HU	HU005						
DESCRIPCION	Realizar la consulta de el resultado de la ejecución de un control de cambios						
CODIGO TAREA	DETALLE TAREA	DIA	NO. PH PLAN	NO. PH REAL	VELOCIDAD	RESPONSABLE	ESTADO
TSK001	Reunión para establecer tareas técnicas	4	1	1		Product Owner Scrum Master Scrum Team	COMPLETADA
TSK002	Reunión de explicación del diseño	4	1	1		Scrum Master Scrum Team Arquitecto	COMPLETADA
TSK003	Creación de Store Procedures para consulta del resultado del cambio	4	3	3		Scrum Team	COMPLETADA
TSK004	Creación de método en la capa de AD del microservicio	4	3	3		Scrum Team	COMPLETADA
TSK005	Creación de endpoints del microservicio	5	2	2		Scrum Team	COMPLETADA
TSK006	Pruebas de integracion capas y bdd	5	2	2		Scrum Team	COMPLETADA
TSK007	Revisión de código	5	2	1		Scrum Team Arquitecto	COMPLETADA
TSK008	Ejecución de pipeline de devops test	5	1	1		Scrum Team Librarian	COMPLETADA
TSK009	Elaboración de guión de pruebas	6	4	4		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK010	Ejecución pruebas sanity	6	2	2		Ingeniero QA	COMPLETADA
TSK011	Ejecución pruebas BUAT	7	5	5		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK012	Ejecución pipeline devops producción	7	1	1		Scrum Master Scrum Team Librarian	COMPLETADA

Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 28 Comparativo de ejecución de la HU005














Elaborado por: Marco Vinicio Luna Aizaga

Ejecución Historia de Usuario 6 (HU006):

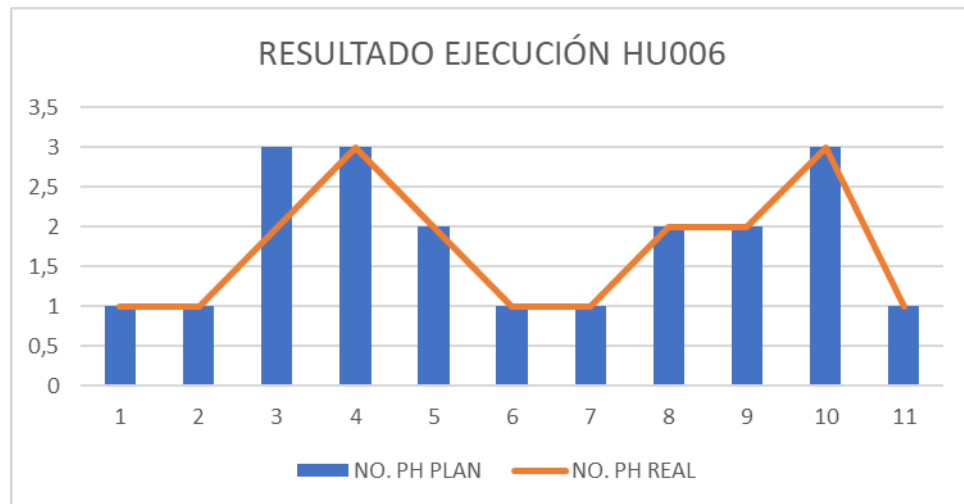
En ésta HU se puede ratificar el buen resultado de la estrategia utilizada por el equipo SCRUM al reutilizar lo construido en la HU005, eliminar una tarea que era redundante y por ende acelerar el proceso de desarrollo ejecutando menos tareas; adicional, se demuestra la ganancia en la finura de la estimación al no tener ya retrasos sino que lo planificado se acerca en casi todo a lo real.

Tabla 20 Resumen ejecución historia de usuario

CODGO HU	HU006						
DESCRIPCION	Realizar la consulta de el estado de un control de cambio						
CODIGO TAREA	DETALLE TAREA	DIA	NO. PH PLAN	NO. PH REAL	VELOCIDAD	RESPONSABLE	ESTADO
TSK001	Reunión para establecer tareas técnicas	8	1	1		Product Owner Scrum Master Scrum Team	COMPLETADA
TSK002	Reunión de explicación del diseño	8	1	1		Scrum Master Scrum Team Arquitecto	COMPLETADA
TSK003	Modificación de Store Procedures para consulta de los campos de planificación del cambio	8	3	2		Scrum Team	COMPLETADA
TSK004	Modificación de método en la capa de AD del microservicio	8	3	3		Scrum Team	COMPLETADA
TSK006	Pruebas de integracion capas y bdd	9	2	2		Scrum Team	COMPLETADA
TSK007	Revisión de código	9	1	1		Scrum Team Arquitecto	EN PROGRESO
TSK008	Ejecución de pipeline de devops test	9	1	1		Scrum Team Librarian	COMPLETADA
TSK009	Elaboración de guión de pruebas	9	2	2		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK010	Ejecución pruebas sanity	10	2	2		Ingeniero QA	COMPLETADA
TSK011	Ejecución pruebas BUAT	10	3	3		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK012	Ejecución pipeline devops producción	10	1	1		Scrum Master Scrum Team Librarian	COMPLETADA

Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 29 Comparativo de ejecución de la HU006

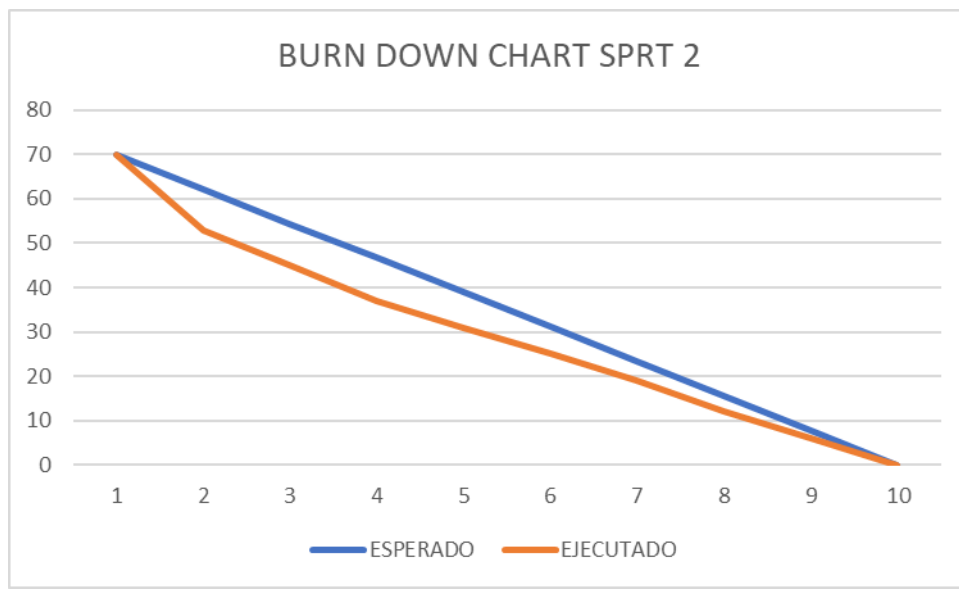


Elaborado por: Marco Vinicio Luna Aizaga

Analizando el resultado de la ejecución del Sprint se denota el impacto que ha tenido la creación de aceleradores en el desarrollo y adicional, el impacto en la mejora en la estimación haciendo que la ejecución ya no se tenga retrasos y, por el contrario, lo planificado se acerque lo más posible a lo real. Sin embargo, el tener tareas adelantadas, aún denota puntos de mejora en la estimación.

En cuanto a la velocidad, al igual que el sprint anterior, el equipo buscó terminar el trabajo lo antes posible con una marcada diferencia entre lo esperado y ejecutado en el día dos del sprint donde el equipo ejecutó mas PH de los planificados

Ilustración 30 Burn Down Chart Sprint 2



Elaborado por: Marco Vinicio Luna Aizaga

El entregable del sprint es el mismo microservicio del sprint anterior, pero con los métodos nuevos desarrollados durante las tres HU abordadas en el mismo.

Ilustración 31 API REST para la consulta de la planificación de un Control de Cambio

```

BGR.CONTROLCAMBIOS.API
├── AccesoDatos
├── __pycache__
├── Repository.py
├── Entidades
│   ├── __pycache__
│   ├── EntidadControlCambio.py
│   ├── EntidadPlanificacion.py
│   ├── EntidadSocializacion.py
│   ├── Negocio
│   └── Utilitarios
├── ControlesCambioAPI.py
└── Prueba.py

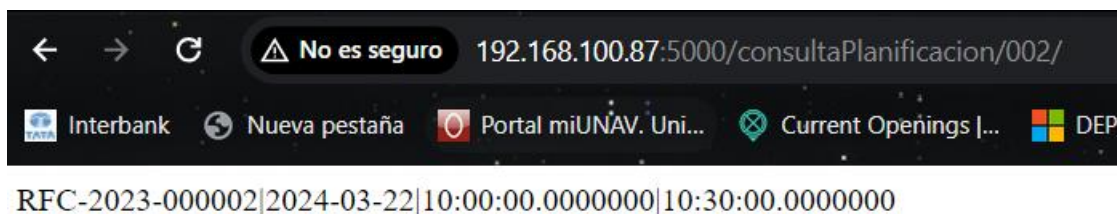
ControlesCambioAPI.py > consultaP
29 |     return cadenaRetorno
30 |
31 | def consultar(codigoCambio):
32 |     listaCambios = base.consultaPlanificacionControlesCambios(codigoCambio)
33 |     cadenaRetorno = ""
34 |     for cambio in range(len(listaCambios)):
35 |         cambioDevuelto = listaCambios[cambio]
36 |         cadenaRetorno = cambioDevuelto.codigoControlCambio + '|' + cambioDevuelto.horaInicio + '|' + cambioDevuelto.horaFin
37 |
38 |     if(cadenaRetorno == None):
39 |         cadenaRetorno = 'No existe planificación para el código de cambio enviado'
40 |
41 |     return cadenaRetorno
42 |
43 | @app.route('/consultaCambio/<string:fecha/>', methods=['GET', 'POST'])
44 |
45 | def consultaCambio(fecha):
46 |     return consultaA(fecha)
47 |
48 | @app.route('/consultaSocializacion/<string:codigoCambio/>', methods=['GET', 'POST'])
49 |
50 | def consultaSocializacion(codigoCambio):
51 |     return consultaS(codigoCambio)
52 |
53 | @app.route('/consultaPlanificacion/<string:codigoCambio/>', methods=['GET', 'POST'])
54 |
55 | def consultaPlanificacion(codigoCambio):
56 |     return consultaP(codigoCambio)
57 |
58 | if __name__ == '__main__':
59 |     app.run('0.0.0.0', port=5000)
60 |

```

Elaborado por: Marco Vinicio Luna Aizaga

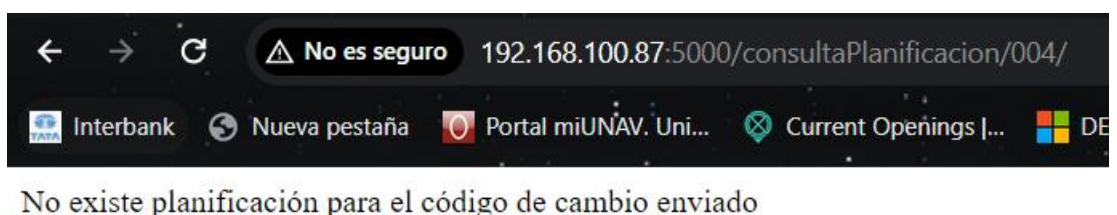
A continuación, se muestra el API REST en ejecución:

Ilustración 32 Ejecución exitosa del API REST de consulta de la planificación de un control de cambio



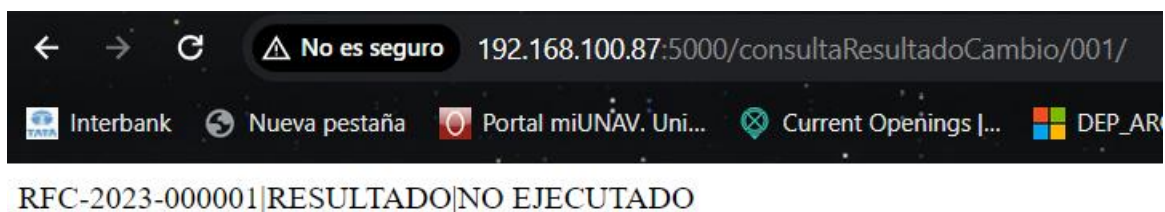
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 33 Ejecución fallida del API REST de consulta de la planificación de un control de cambio



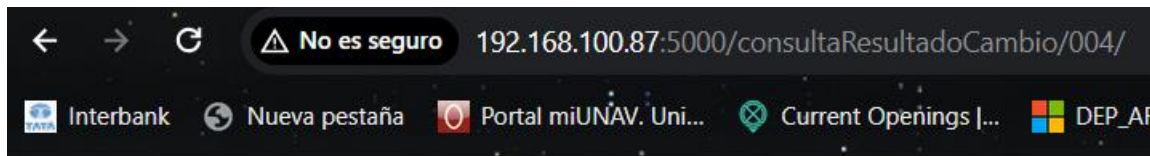
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 34 Ejecución exitosa del API REST de consulta del resultado de un control de cambio



Elaborado por: Marco Vinicio Luna Aizaga

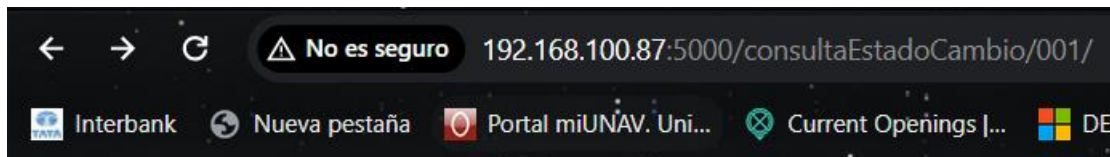
Ilustración 35 Ejecución exitosa del API REST de consulta del resultado de un control de cambio



No existe Resultado para el código de cambio enviado

Elaborado por: Marco Vinicio Luna Aizaga

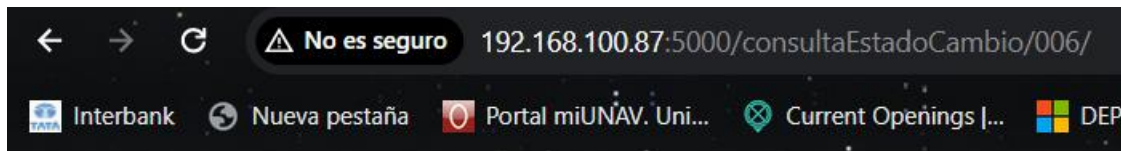
Ilustración 36 Ejecución exitosa del API REST de consulta del estado de un control de cambio



RFC-2023-000001|ESTADO|PLANIFICADO

Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 37 Ejecución fallida del API REST de consulta del estado de un control de cambio



No existe Estado para el código de cambio enviado

Elaborado por: Marco Vinicio Luna Aizaga

5.5.3. Tercer Sprint

Finalmente, el resultado de la ejecución de la última historia de usuario se aprecia que el número de puntos de historia (PH) planificados versus los reales ejecutados son en su mayoría iguales excepto en una tarea donde se sub dimensionó el esfuerzo; sin embargo, se puede apreciar que el equipo mantuvo su velocidad buscando culminar el trabajo en el menor tiempo posible.

Ejecución Historia de Usuario 1 (HU001):

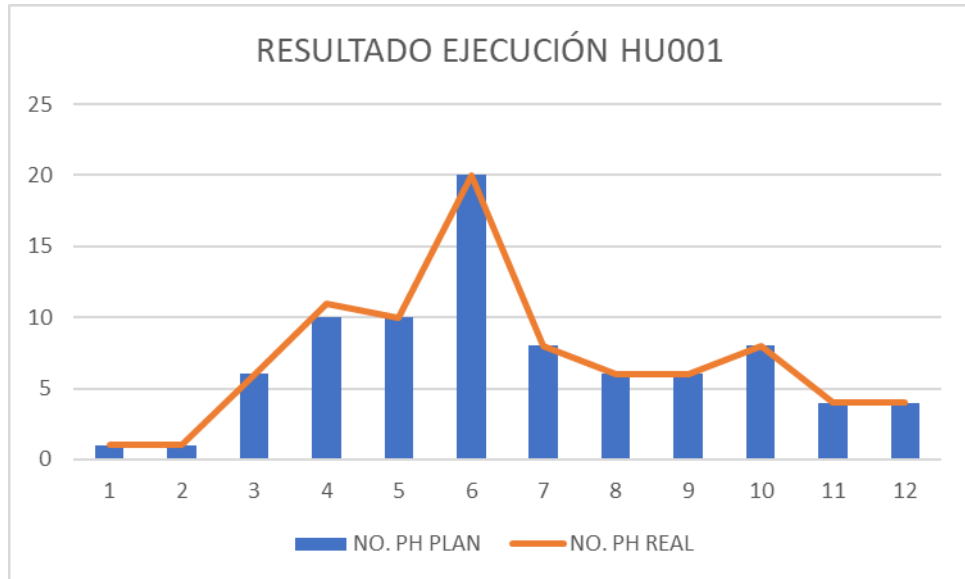
En ésta HU se aprecia un error de estimación en la tarea de despliegue del servidor pero que a la larga no afectó en el resultado final ya que se cumplió con el compromiso hacia el sprint.

Tabla 21 Resumen ejecución historia de usuario

CODIGO HU	HU001						
DESCRIPCION	Crear un menú conversacional con un chatbot para realizar las consultas de fechas de socialización de cambios, controles de cambios autorizados o no, planificación de un control de cambio, resultado de la ejecución de un control de cambios y estado de un control de cambio						
CODIGO TAREA	DETALLE TAREA	DIA	NO. PH PLAN	NO. PH REAL	VELOCIDAD	RESPONSABLE	ESTADO
TSK001	Reunión para establecer tareas técnicas	1	1	1		Product Owner Scrum Master Scrum Team	COMPLETADA
TSK002	Reunión de explicación del diseño	1	1	1		Scrum Master Scrum Team Arquitecto	COMPLETADA
TSK003	Configurar servidor para despliegue de los microservicios en la nube	1	6	6		Scrum Team Infraestructura	COMPLETADA
TSK004	Desplegar los microservicios en el servidor en la nube	2	10	11		Scrum Team	COMPLETADA
TSK005	Crear un Webhook en Dialogflow	3	10	10		Scrum Team	COMPLETADA
TSK006	Implementar código en el Webhook	4	20	20		Scrum Team	COMPLETADA
TSK007	Revisión de diseño e integración	5	8	8		Scrum Team Arquitecto	COMPLETADA
TSK008	Elaboración de guión de pruebas	6	6	6		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK009	Ejecución pruebas sanity	7	6	6		Ingeniero QA	COMPLETADA
TSK010	Ejecución pruebas BUAT	8	8	8		Product Owner Scrum Master Scrum Team Ingeniero QA	COMPLETADA
TSK011	Habilitar el chatbot en modo Friends & Family	9	4	4		Product Owner Scrum Master Scrum Team Infraestructura	
TSK012	Habilitar el chatbot a los usuarios en general	10	4	4		Product Owner Scrum Master Scrum Team Infraestructura	COMPLETADA

Elaborado por: Marco Vinicio Luna Aizaga

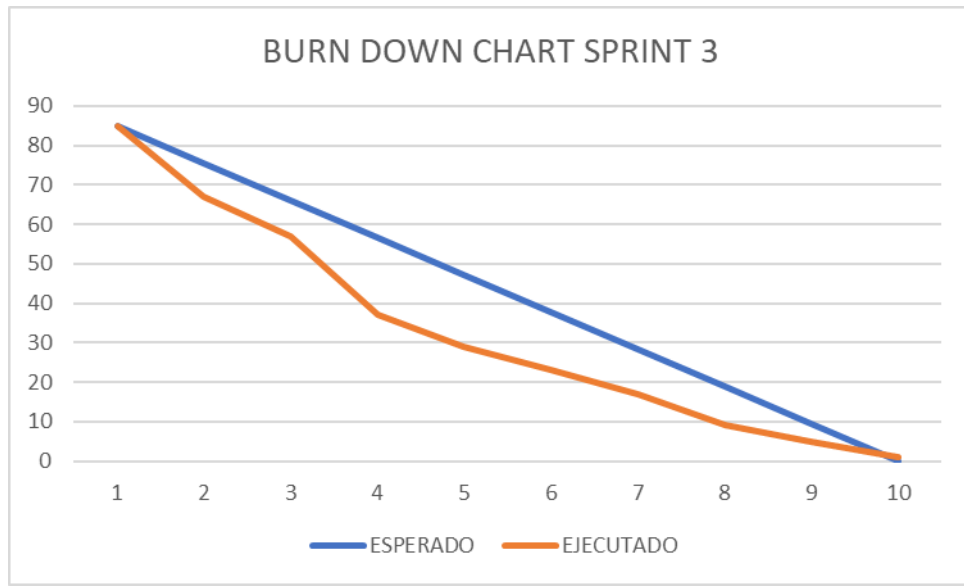
Ilustración 38 Comparativo de ejecución de la HU001



Elaborado por: Marco Vinicio Luna Aizaga

Analizando el resultado de la ejecución del Sprint se denota una reducción en los problemas de estimación que se tuvo al inicio del proyecto por parte del Equipo SCRUM pero sin desaparecer del todo dado que se tuvo una tarea aún con un sub dimensionamiento; sin embargo, en cuanto a velocidad se mantiene el mismo comportamiento de los Sprints anteriores; es decir, el equipo estuvo siempre debajo de la línea del trabajo esperado dada la orientación a terminar el trabajo lo antes posible.

Ilustración 39 Burn Down Chart Sprint 3



Elaborado por: Marco Vinicio Luna Aizaga

El entregable del sprint es el chatbot funcionando en producción respondiendo a los requerimientos de consulta e integrado con los microservicios antes mencionados.

Como se indicó anteriormente, la plataforma seleccionada para la construcción del Chatbot es DialogFow de Google en su versión Essential la cual nos permite la creación de este tipo de tecnología con fines educativos y de investigación.


A continuación, se mostrará los pasos, tanto para la construcción del Chatbot, la configuración de las intenciones del mismo, y la integración, mediante Webhook, con los microservicios anteriormente desarrollados.

Ilustración 40 Creación del Chatbot en DialogFlow Essential

BotControlCambios

SAVE

General Languages ML Settings Export and Import Environments Speech Share Advanced



DESCRIPTION
Describe your agent (will be used in Web Demo integration)

DEFAULT TIME ZONE
(GMT-5:00) America/New_York
Date and time requests are resolved using this timezone if not provided in the API requests.

AGENT AVATAR URI
Define URI to agent avatar that will be used in [Web Demo](#) and [Google Chat](#) integrations.

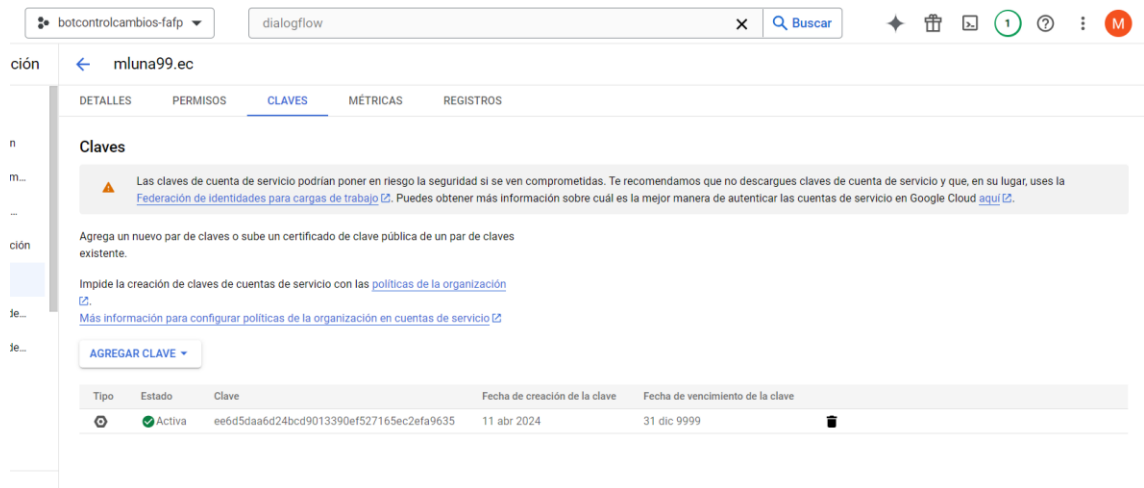
GOOGLE PROJECT

Project ID	botcontrolcambios-fafp
------------	------------------------

Elaborado por: Marco Vinicio Luna Aizaga

Denótese que el ID de Proyecto de nuestro Chatbot en la plataforma DialogFlow dado que mediante éste se podrá integrar con la plataforma Whatsapp para su puesta en producción.

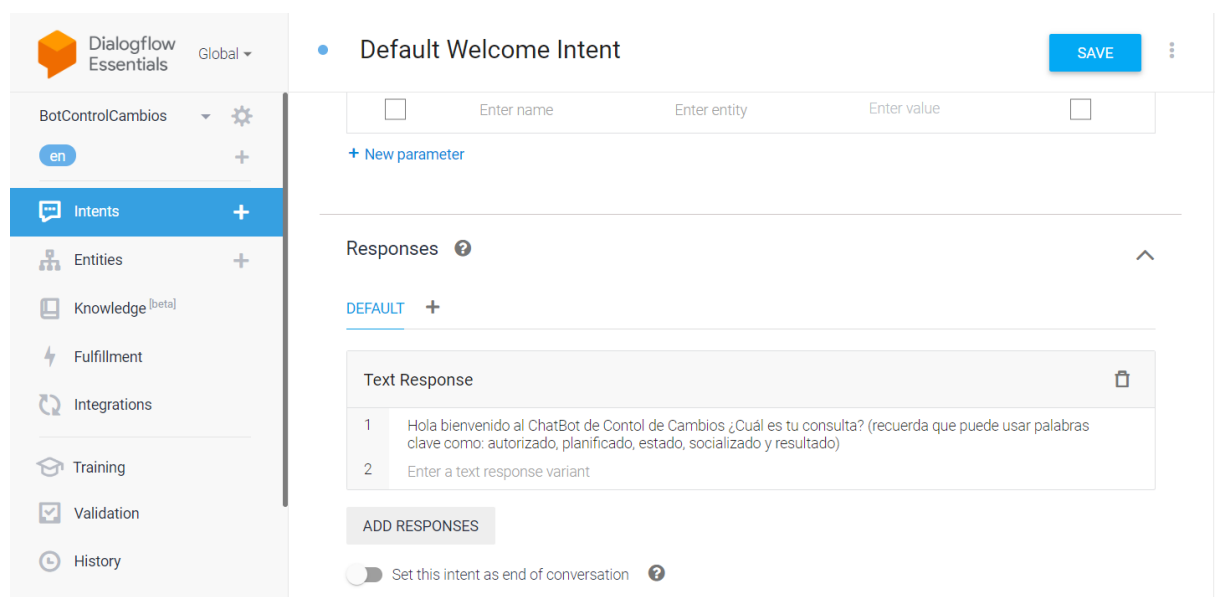
Ilustración 41 Extracción de la clave para la integración del Chatbot con la plataforma Whatsapp



Elaborado por: Marco Vinicio Luna Aizaga

Como se indicó anteriormente, el Chatbot se comunica con el usuario final mediante la ejecución de intenciones (Intents) los cuales determinan el contexto conversacional del Chatbot.

Ilustración 42 Modificación del Intent de bienvenida por defecto del Chatbot



Elaborado por: Marco Vinicio Luna Aizaga

Como se puede apreciar, es necesario cambiar el mensaje de respuesta del Intent de bienvenida del Chatbot a fin de que pueda responder ante cualquier inicio conversacional con un mensaje que guíe al usuario sobre las opciones de consulta que se tiene y sobre todo ordene dicha conversación. Esto con miras a mejorar la Experiencia de Usuario (UX).

Ilustración 43 Creación del Intent para consulta de cambios autorizados

The screenshot displays the Dialogflow Essentials interface for configuring an intent. The sidebar on the left shows the navigation menu with 'Intents' selected. The main content area is titled 'ControlCambiosAutorizados' and features a 'SAVE' button in the top right corner. Under the 'Training phrases' section, there is a yellow warning banner stating: 'Template phrases are deprecated and will be ignored in training time. More details here.' Below the warning, a text input field contains the phrase 'autorizados'. The 'Action and parameters' section is expanded, showing a table for defining parameters.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	fecha	@sys.date	\$date	<input type="checkbox"/>	la pregunta es...

Elaborado por: Marco Vinicio Luna Aizaga

Como se observa en la imagen anterior, el Intent responde ante la palabra “autorizados” y espera un parámetro esencial de nombre “fecha” y que tiene un tipo de dato particular.

Con base a esto, si el usuario durante la conversación con el Intent no indica el valor del parámetro de entrada, el Intent mostrará un mensaje guía al usuario para que entregue la información de éste.

Ilustración 44 Configuración de la frase por defecto si el parámetro está en blanco



Elaborado por: Marco Vinicio Luna Aizaga

A continuación, se muestra la configuración del “fullfillmen” en Webhook de la plataforma de DialogFlow para la integración con los microservicios anteriormente desarrollados; no sin antes indicar que, para que los mismos estén expuestos de forma pública y con la seguridad respectiva, se procedió con la utilización de la plataforma Ngrok la cual genera una URL pública que sirve como una especie de NAT para exponer el microservicio local hacia el público mediante una URL expuesta en Internet.

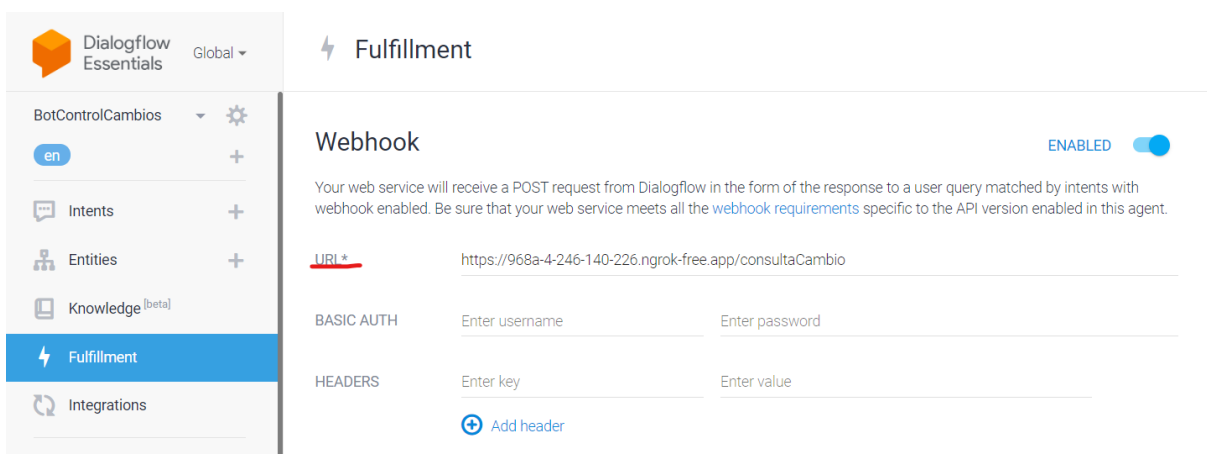
Ilustración 45 Configuración de Ngrok para crear la URL pública que exponga los microservicios



Elaborado por: Marco Vinicio Luna Aizaga

Una vez que se tiene la URL pública generada mediante Ngrok, se procede con la configuración del Fullfillment del Intent mediante la opción de Webhook en DialogFlow.

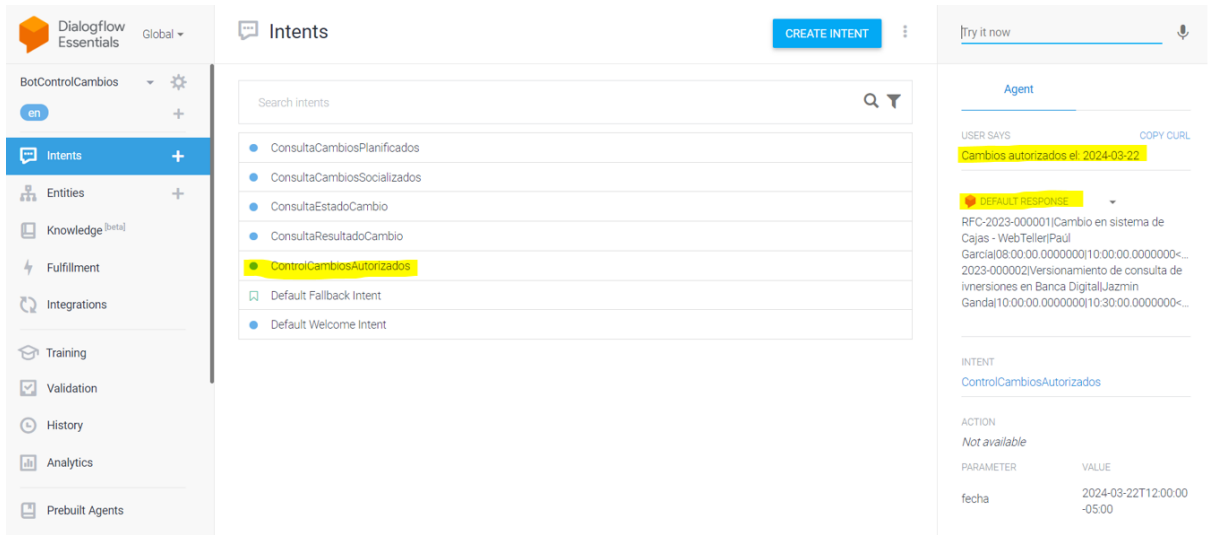
Ilustración 46 Configuración de Webhook para el intento



Elaborado por: Marco Vinicio Luna Aizaga

Finalmente, se procede con la prueba unitaria a fin de comprobar las configuraciones previo a su integración con Whatsapp.

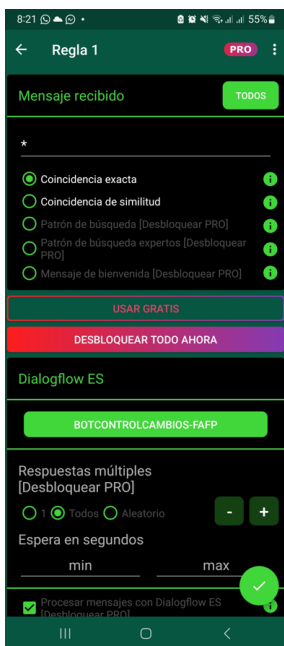
Ilustración 47 Prueba unitaria del Intent dentro de la plataforma de DialogFlow



Elaborado por: Marco Vinicio Luna Aizaga

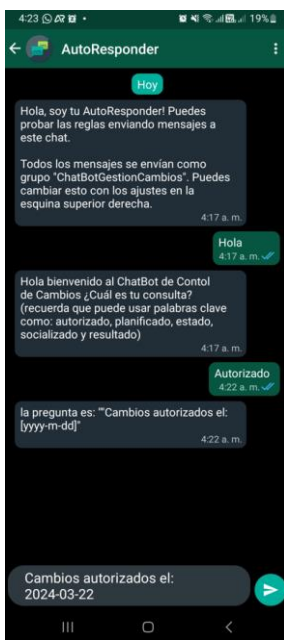
A continuación, se muestra la conversación con el Chatbot dentro del contexto de Intent desde la plataforma de Whatsapp; para el efecto, se configura un autoresponder utilizando el archivo de claves generado con el ID del Chatbot en la plataforma de Dialogflow.

Ilustración 48 Creación del autoresponder para integración con Whastapp en el equipo celular



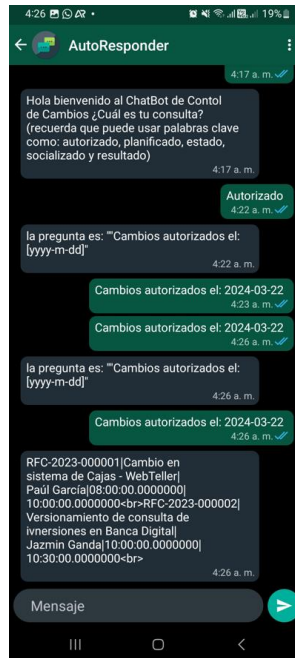
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 49 Inicio de la conversación con el Chatbot en la plataforma de Whatsapp



Elaborado por: Marco Vinicio Luna Aizaga

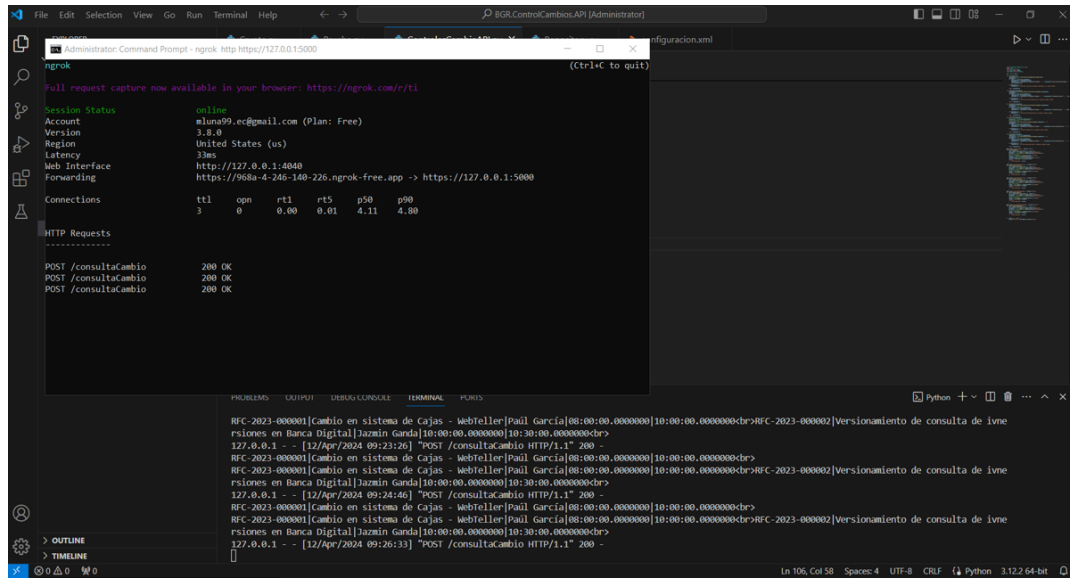
Ilustración 50 Consulta de la información de cambios autorizados en Whatsapp mediante el Chatbot



Elaborado por: Marco Vinicio Luna Aizaga

Como se puede apreciar, el entregable se lo creó con éxito y con la planificación y resultados antes indicados. A continuación, se muestra la interacción del chatbot y su comprobación contra la plataforma donde se encuentra desplegado los microservicios y expuestos mediante Ngrok.

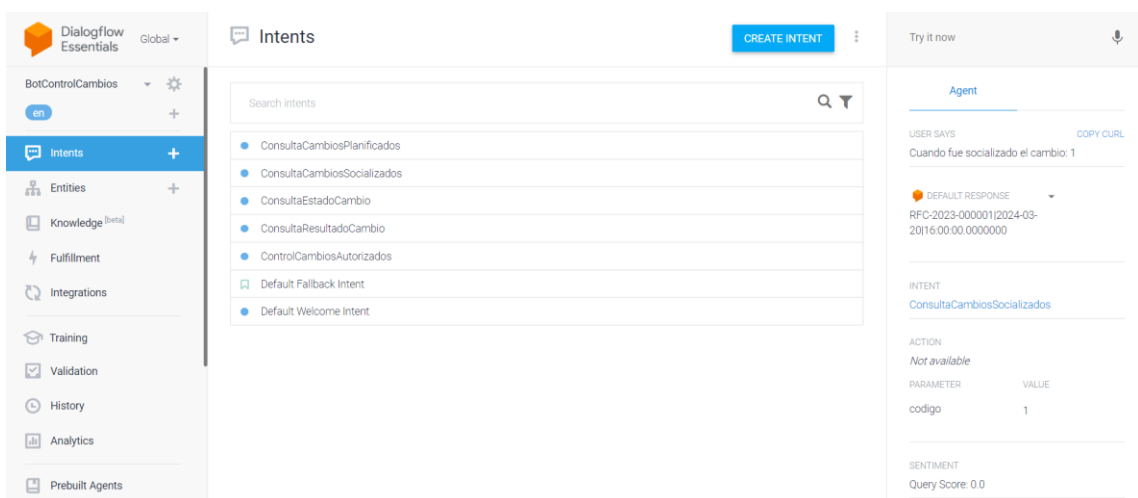
Ilustración 51 Comprobación de consultas al microservicio expuesto por Ngrok



Elaborado por: Marco Vinicio Luna Aizaga

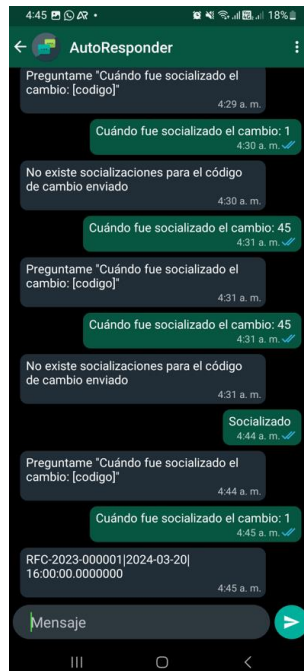
Las siguientes imágenes muestran la ejecución de las comprobaciones en producción para asegurar el éxito del versionamiento del proyecto.

Ilustración 52 Creación del Intent para consulta de fecha de socialización de los cambios



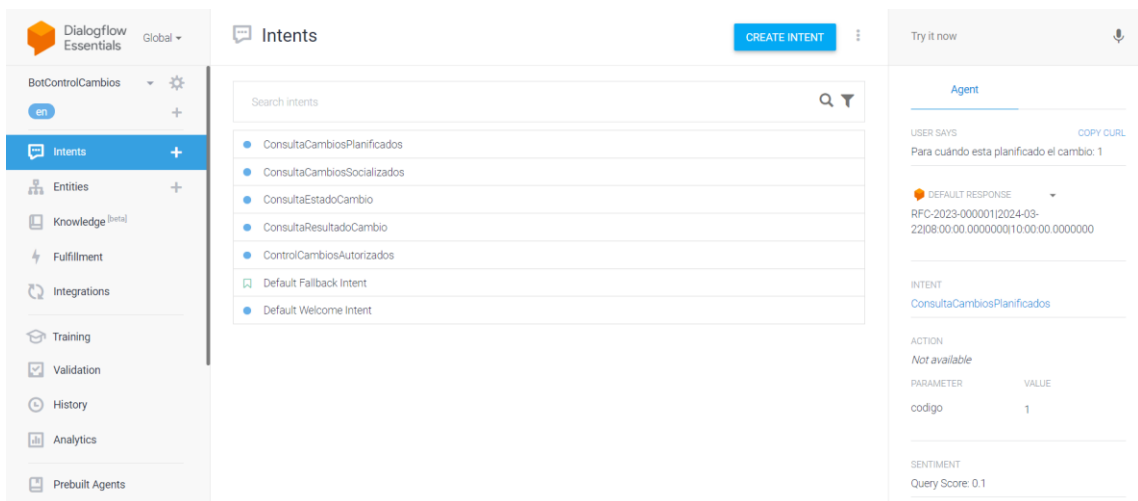
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 53 Prueba del Intent de consulta de socialización desde la plataforma de Whatsapp



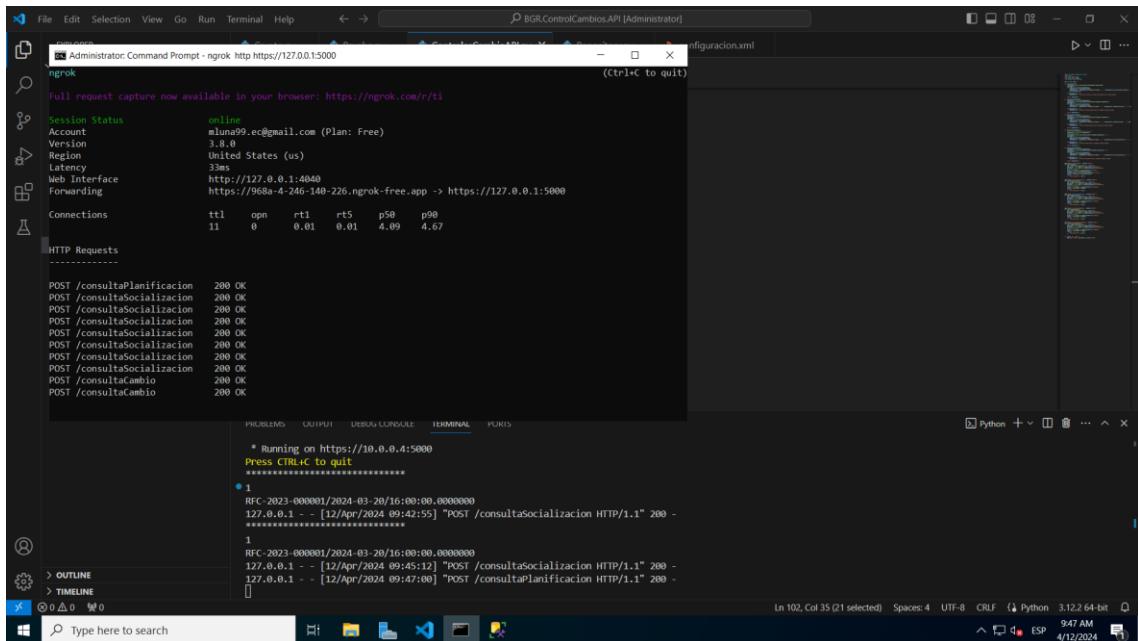
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 54 Creación del Intent para consulta de fecha de planificación para ejecución de los cambios



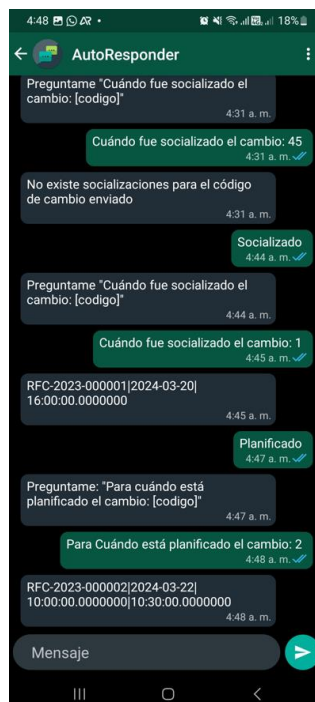
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 55 Comprobación de la consulta del microservicio de planificación mediante Ngrok



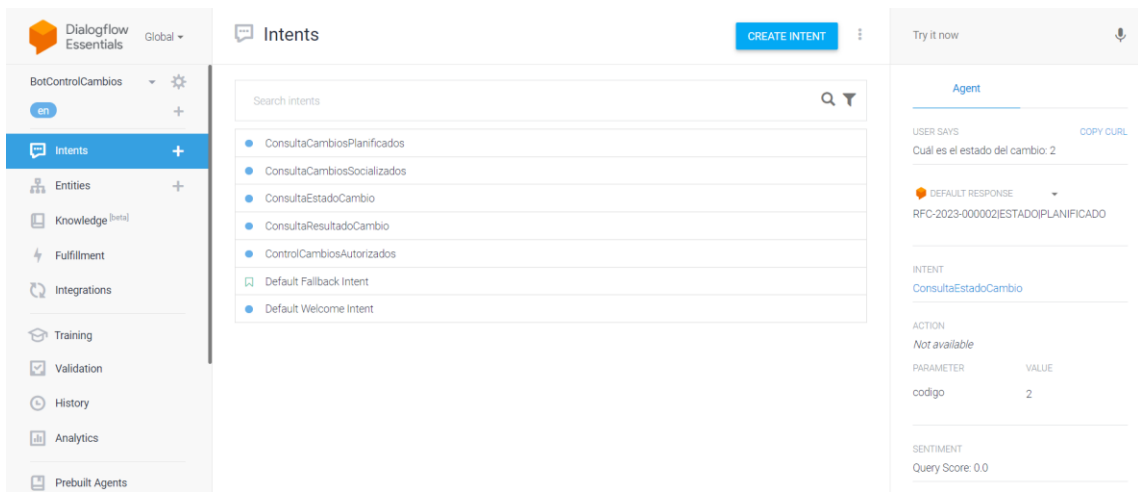
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 56 Comprobación del Intent de la consulta de planificación en Whatsapp



Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 57 Creación del Intent para consulta el estado de los cambios



Elaborado por: Marco Vinicio Luna Aizaga

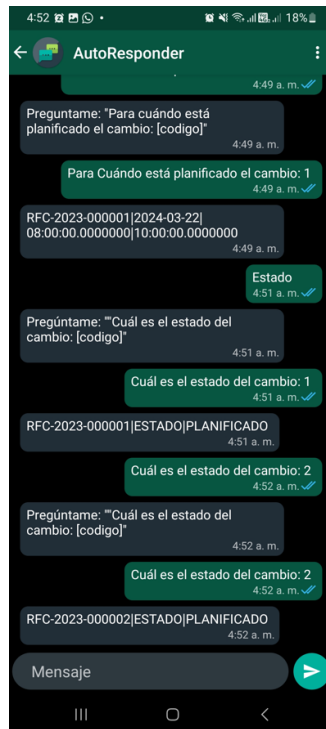
Con fines de comprobación, se procede a realizar la consulta directa a la BDD a fin de mostrar que, lo que aparece en la prueba llamando directo al Intent en DialogFlow, muestra la información en tiempo real

Ilustración 58 Consulta directa del estado de dos cambios en la BDD

	ID_CONTROL_CAMBIO	ESTADO	CODIGO_CONTROL_CAMBIO	DESCRIPCION_CONTROL_CAMBIO	FECHA_SOLICITADA	FECHA_PLANIFICADA	HORA_INICIO_PLANIFICADA
1	1	PLANIFICADO	RFC-2023-000001	Cambio en sistema de Cajas - WebTeller	2024-03-22	2024-03-22	08:00:00.0000000
2	2	PLANIFICADO	RFC-2023-000002	Versionamiento de consulta de inversiones en Ban...	2024-03-24	2024-03-22	10:00:00.0000000

Elaborado por: Marco Vinicio Luna Aizaga

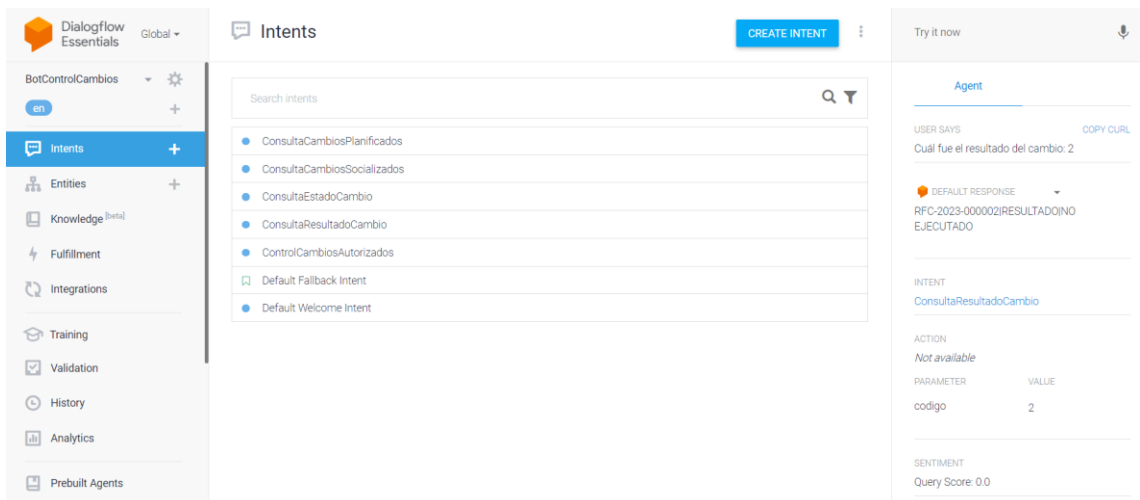
Ilustración 59 Consulta del estado del cambio desde Whastapp



Elaborado por: Marco Vinicio Luna Aizaga

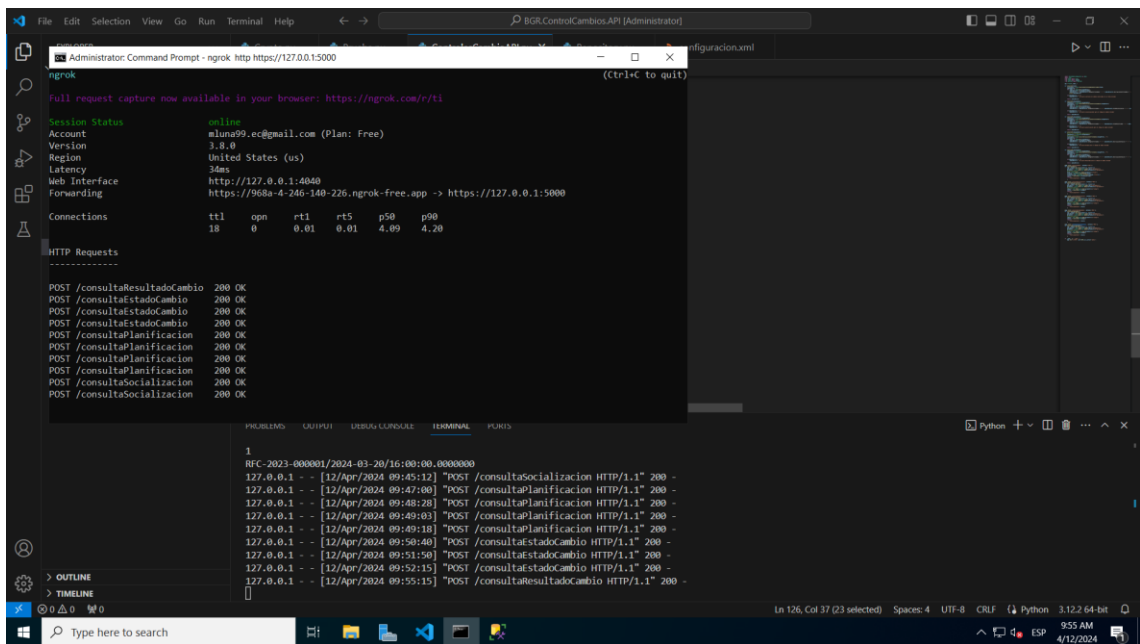
En las siguientes imágenes se muestra la prueba del Intent de consulta del resultado de los cambios, realizando una prueba de actualización del resultado para que se compruebe la consulta en tiempo real.

Ilustración 60 Prueba del Intent de consulta del resultado de los cambios.



Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 61 Comprobación de la consulta a nivel del microservicio expuesto con Ngrok.



Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 62 Comprobación de la consulta a nivel de la BDD.

ID_CONTROL_CAMBIO	RESULTADO	ESTADO	CODIGO_CONTROL_CAMBIO	DESCRIPCION_CONTROL_CAMBIO	FECHA_SOLICITADA	FECHA_PLANIFICADA
1	NO EJECUTADO	PLANIFICADO	RFC-2023-000001	Cambio en sistema de Cajas - WebTeller	2024-03-22	2024-03-22
2	NO EJECUTADO	PLANIFICADO	RFC-2023-000002	Versionamiento de consulta de inversiones en Ban...	2024-03-24	2024-03-22

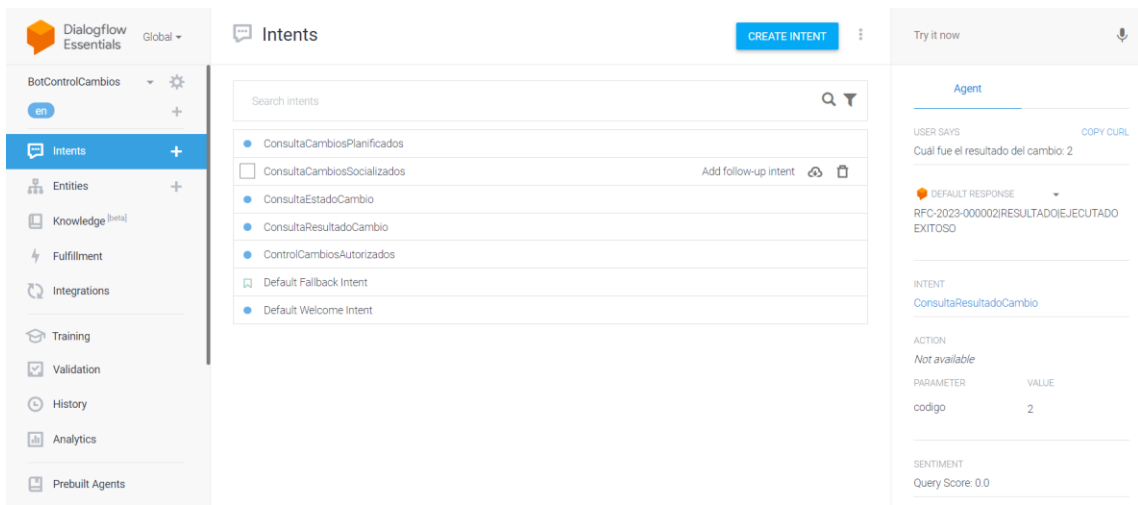
Elaborado por: Marco Vinicio Luna Aizaga

Gráfico 62. Modificación del resultado de uno de los cambios a nivel de BDD.

ID_CONTROL_CAMBIO	RESULTADO	ESTADO	CODIGO_CONTROL_CAMBIO	DESCRIPCION_CONTROL_CAMBIO	FECHA_SOLICITADA	FECHA_PLANIFICADA
1	NO EJECUTADO	PLANIFICADO	RFC-2023-000001	Cambio en sistema de Cajas - WebTeller	2024-03-22	2024-03-22
2	EJECUTADO EXITOSO	PLANIFICADO	RFC-2023-000002	Versionamiento de consulta de inversiones en Ban...	2024-03-24	2024-03-22

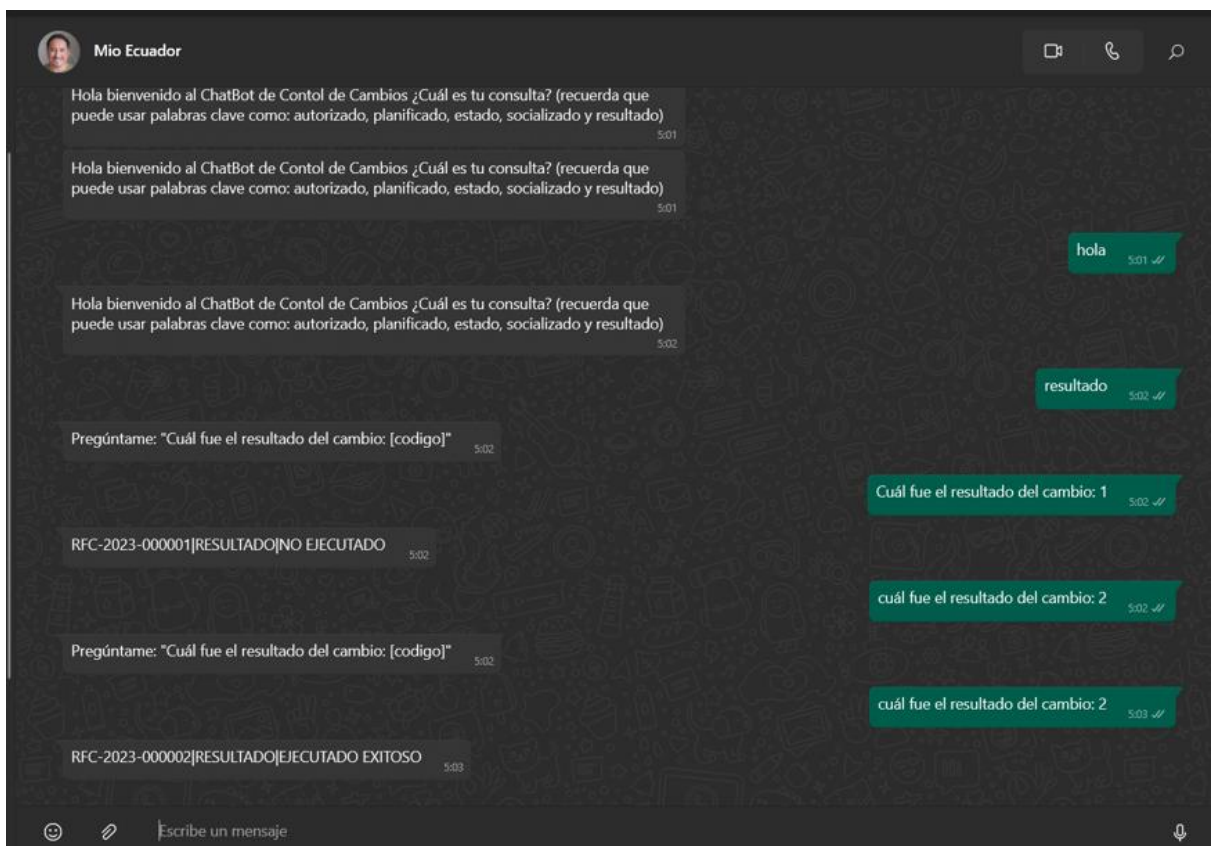
Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 63 Consulta del resultado del cambio a nivel del Intent en DialogFlow.



Elaborado por: Marco Vinicio Luna Aizaga

Ilustración 64 Consulta del resultado del cambio a nivel de Whatsapp.



Elaborado por: Marco Vinicio Luna Aizaga

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Se concluye que los marcos de referencia ágiles, de manera general, contribuyen de manera efectiva a la ejecución de proyectos y de manera particular el marco de referencia utilizado en el presente trabajo es una herramienta eficaz para el desarrollo de proyectos tecnológicos.
- Se concluye que el utilizar la herramienta de Backlog dentro del Marco de Referencia de SCRUM, es fundamental para la organización del trabajo y delinear la estrategia de solución a la necesidad que se ataca con un proyecto tecnológico de manera particular.
- Se concluye que la integración entre las tecnologías de chatbots y de microservicios no solo que es posible, sino que es necesaria para que la creación de Asistentes Virtuales agregue valor a las empresas y se pueda diseminar su uso distintas líneas de negocio, servicio e investigación.
- Se concluye que, a pesar de tener dificultades en la estimación de las tareas, el proceso de agregación de valor en el menor tiempo posible mediante el uso del Marco de Referencia de SCRUM es real y efectivo; todos los Sprints planificados se ejecutaron manteniendo a la disciplina y las ceremonias como las herramientas eficaces para la consecución de dicho objetivo.

6.2 RECOMENDACIONES

- Es imperativo que, durante la recolección de requerimientos funcionales para la creación del Backlog, las HU y los Sprints, también se contemplen los requerimientos no funcionales y el diseño de Arquitectura.
- Se recomienda que los Requerimientos No Funcionales y el Diseño de Arquitectura sean revalidados durante cada Sprint a fin de asegurar su cumplimiento. Si se requiere algún ajuste a dichos ítems, es necesario que se los aborde con otro grupo distinto al SCRUM Team.
- Es necesario de crear una cultura de mejora continua en la estimación de esfuerzos con el SCRUM Team a fin de ganar en finura y que los esfuerzos no se sobren o sub dimensionen. La mejor opción será revalidar esto en el Sprint Review.
- Es importante que, a la par de la utilización de las herramientas de un Marco de Referencia Ágil, también se evalúa la opción de adoptar alguna otra herramienta que ayude a la consecución de los objetivos; por ejemplo, LEAN. KAMBN, entre otros.

Referencias

- Amazon. (20 de 05 de 2023). *Qué son los microservicios*. Obtenido de aws.amazon: <https://aws.amazon.com/es/microservices/>
- Banco General Rumiñahui. (25 de 01 de 2024). *bgr.com*. Obtenido de Conoce BGR: <https://www.bgr.com.ec/informacion-corporativa?t=conoce-bgr>
- Beck, & Beedle. (2001). *Manifiesto por el Desarrollo Ágil de Software*. Obtenido de agilemanifiesto: <https://agilemanifiesto.org/iso/es/manifiesto.html>
- Benavidez. (12 de 09 de 2019). *muysegura.com*. Obtenido de Transformación digital o transformación personal: <https://www.muysegura.com/transformacion-digital-o-transformacion-personal/>
- Deloitte. (2020). *deloitte.com*. Obtenido de Madurez de la banca digital: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/<https://www2.deloitte.com/content/dam/Deloitte/cl/Documents/financial-services/cl-madurez-de-la-banca-Digital-2020.pdf>
- Fernandez, R. (7 de 12 de 2021). *es.statista.com*. Obtenido de <https://es.statista.com/estadisticas/972995/asistentes-virtuales-en-uso-en-el-mundo/>
- Ganesch, R., Akopian, D., Rhud, M., & Inupakutika, D. (01 de 15 de 2022). *Chatbot integrated with machine learning deployed in the cloud and performance evaluation*. Obtenido de Researchgate.net: https://www.researchgate.net/publication/359820770_Chatbot_integrated_with_machine_learning_deployed_in_the_cloud_and_performance_evaluation
- Gómez, J. (2015). <https://www.laboratorioti.com>. Obtenido de Informe del Caos 2015 (Chaos Report 2015) o Cómo de bien o mal fueron los proyectos en el año 2015: <https://www.laboratorioti.com/2016/05/16/informe-del-caos-2015-chaos-report-2015-bien-mal-fueron-los-proyectos-ano-2015/>
- González, O. (10 de 12 de 2021). *Sprint retrospective: ganar perspectiva ¡con retrospectiva!* Obtenido de appvizer.es: <https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/sprint-retrospective>
- Google. (05 de 07 de 2022). *cloud.google.com*. Obtenido de Fulfillment: <https://cloud.google.com/dialogflow/es/docs/fulfillment-overview?hl=es-419>
- Google. (27 de 10 de 2023). *cloudskillsboost.google*. Obtenido de Cómo diseñar flujos de conversación para tu agente: <https://www.cloudskillsboost.google/focuses/12347?locale=es&parent=catalog>

- Google cloud. (25 de 02 de 2021). *cloud.google*. Obtenido de Conversational AI with Apigee API Management for enhancing customer experiences: <https://cloud.google.com/blog/products/api-management/conversational-ai-with-dialogflow-and-apigee-api-management>
- Harrys, C. (2024). *atlassian.com*. Obtenido de Comparación entre la arquitectura monolítica y la arquitectura de microservicios: <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>
- Hiberus. (2019). *Arquitectura de Software*. Obtenido de Universidad de Mexico: <https://www.uv.mx/personal/ermeneses/files/2020/09/Clase1-Introduccion.pdf>
- IEBS Bussiness School. (14 de 11 de 2023). *iebschool.com*. Obtenido de Las metodologías ágiles más utilizadas en la empresa: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- Lauria, G. (30 de 11 de 2021). *debmedia.com*. Obtenido de Qué un asistente virtual y cómo funciona: <https://debmedia.com/blog/asistente-virtual>
- Manrique, F. (11 de 06 de 2022). *Universidad Peruana de Ciencias Aplicadas*. Obtenido de Solución tecnológica para mejorar el proceso de atención a clientes en una entidad bancaria utilizando un asistente virtual conversacional: <https://repositorioacademico.upc.edu.pe/handle/10757/660715>
- Martins, J. (15 de 02 de 2024). *SCRUM: Conceptos claves y como se aplica en la gestión de proyectos*. Obtenido de asana.com: <https://asana.com/es/resources/what-is-scrum>
- Mazza, F. (26 de 05 de 2022). *Customer Journey Map: Qué es y cómo crear uno*. Obtenido de fromdoppler.com: <https://blog.fromdoppler.com/customer-journey-map-como-crear-uno/>
- PMPBOOK. (22 de 04 de 2022). *todopm.com*. Obtenido de Guia de los fundamentos para dirección de proyectos: <https://todopmp.com/pmbok-6-resumen/>
- Project Management Institute. (15 de 02 de 2024). *pmi.org*. Obtenido de <https://www.pmi.org/>
- Ruiz, J. (04 de 08 de 2018). *Los 4 Valores de la Agilidad: El Manifiesto Ágil*. Obtenido de jorgeruizgile.com: <https://jorgeruizagile.com/2018/08/04/los-4-valores-de-la-agilidad-el-manifiesto-agil/>
- Visual-Paradigm. (28 de 02 de 2024). *What is Burndown Chart in Scrum?* Obtenido de visual-paradigm.com/scrum: <https://www.visual-paradigm.com/scrum/scrum-burndown-chart/>

ANEXOS