

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN CHATBOT PARA AGENDAMIENTO DE CITAS POR SERVICIOS PROFESIONALES USANDO INTERFACES CONVERSACIONALES.

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

ALISSON MONSERRATTE LÓPEZ MEJÍA

alisson.lopez01@epn.edu.ec

DIRECTOR: MSC. ADRIÁN VICENTE EGUEZ SARZOSA.

adrian.eguez@epn.edu.ec

**CODIRECTOR: MSC. CARLOS ESTALESMIT WILLIAM MONTENEGRO
ARMAS.**

carlos.montenegro@epn.edu.ec

Quito, julio 2023

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **ALISSON MONSERRATTE LÓPEZ MEJÍA**, bajo mi supervisión.

Ing. Adrián Vicente Eguez Sarzosa, MSc.
DIRECTOR DE PROYECTO

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **ALISSON MONSERRATTE LÓPEZ MEJÍA**, bajo mi supervisión.

Ing. Carlos Estalesmit William Montenegro Armas, MSc.
CODIRECTOR DE PROYECTO

DECLARACIÓN

Yo, **ALISSON MONSERRATTE LÓPEZ MEJÍA**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

ALISSON MONSERRATTE LÓPEZ MEJÍA

DEDICATORIA

A mi madre Nelly, que no importaba la hora, siempre estuvo lista para apoyarme por siempre ser la nobleza y fortaleza que necesitaba en todo momento, por ser mi norte y guiarme siempre por amarme infinitamente.

A mi padre Edwin por su apoyo incondicional, por darme un camino, por ser el ejemplo y trabajar incansablemente por mi.

A mi hermana Alejandra por ser el mejor regalo que la vida pudo darme, por creer siempre en mi.

Este trabajo es gracias a ustedes, gracias por todo querido equipo, ¡Lo logramos!

AGRADECIMIENTOS

Agradezco a Dios por acompañarme siempre y ayudarme a terminar esta etapa. Sin él, nada sería posible.

A mi madre, quien fue mi fortaleza, mi lugar seguro y, sobretodo, mi rayito de luz. Tu amor inquebrantable ha sido la fuerza que me ha impulsado a alcanzar este logro.

A mi padre, quien me ha acompañado desde pequeña a descubrir este mundo, sin importar la hora. Siempre estuviste presente, listo para ayudar y nos diste lo mejor.

A Leslie, quien me daba esa fuerza para seguir, abrazaba mis derrotas y me impulsaba a mejorar. Siempre confiaste en que lo lograría.

A Delia y Pedro, quienes ante toda adversidad supieron guiarme. Son símbolo de lucha y demuestran que todo se puede lograr con determinación y esfuerzo.

Agradezco a Sonia por su sacrificio, apoyo incondicional y por sacarnos adelante.

Agradezco a Eve y Nico por acompañarme en esta travesía, hacer este logro parte de ustedes y por siempre estar pendientes de mí.

A mis amigos, Andrés, Néstor, Ángeles, Danny y Mary, gracias por hacer este camino más ameno. Cada bello momento compartido juntos es invaluable.

A Kevin, agradezco haberte encontrado. Gracias por tu amor y apoyo incondicional. Qué bonita es la vida sosteniendo tu mano.

A mi familia Manticore, Cristian, Alexander, Andrés, Oscar y Edwin, gracias por compartir su conocimiento y hacer este camino más agradable.

A mi director MSc. Adrian Eguez, quien confió en mí desde el principio. Gracias por respaldarme y brindarme tantas oportunidades para crecer profesionalmente. Por ser un gran maestro y guía en el proceso de este proyecto

A mi co-director MSc. Carlos Montenegro, gracias por brindarme la ayuda necesaria para llevar a cabo este trabajo.

CONTENIDO

Índice de figuras	IX
Índice de cuadros	XII
Resumen	1
Abstract	2
1 INTRODUCCIÓN	3
1.1 Descripción del problema	3
1.2 Objetivos	4
1.2.1 Objetivo General	4
1.2.2 Objetivos Específicos	4
1.3 Alcance	4
1.4 Marco teórico	5
1.4.1 Procesamiento del lenguaje natural (PNL)	5
1.4.2 Interfaz conversacional	5
1.4.3 Webhook	6
1.4.4 Modelo vista controlador	6
1.4.5 Aplicación web	7
1.5 Herramientas de Desarrollo	7
1.5.1 Visual Studio Code	7
1.5.2 Draw io	8
1.5.3 Git	8
1.5.4 Github	8
1.5.5 Gitkraken	8
1.5.6 MySQL	8
1.5.7 Typescript	9
1.5.8 NestJS	9
1.5.9 HTML5	9
1.5.10 SCSS	9
1.5.11 Angular	10

1.5.12 PrimeNG	10
1.5.13 Angular material	10
1.5.14 Google Dialogflow	10
1.5.15 Google Calendar API	10
1.6 Metodología ágil	11
1.6.1 Extreme Programming XP	11
1.6.2 SCRUM	12
2 METODOLOGÍA	15
2.1 Por qué SCRUM	15
2.2 Equipo SCRUM	15
2.3 Ingeniería de requerimientos	16
2.3.1 Recopilación y análisis de información	17
2.4 Sprint	18
2.4.1 Sprint 0	18
2.4.2 Sprint 1	28
2.4.3 Sprint 2	34
2.4.4 Sprint 3	42
2.4.5 Sprint 4	48
2.4.6 Sprint 5	55
2.4.7 Sprint 6	64
2.4.8 Sprint 7	69
3 RESULTADOS Y DISCUSIÓN	76
3.1 Descripción del sistema	76
3.1.1 Descripción de los artefactos	76
3.2 Resultados de evaluación	83
3.2.1 Resultados Prueba de caja negra	83
3.3 Evaluación de usabilidad del sistema	88
3.3.1 Descripción de la metodología MARS	88
3.3.2 Resultados MARS	89
3.4 Discusión	93
4 CONCLUSIONES	95
4.1 Conclusiones	95
4.2 Recomendaciones	96
5 REFERENCIAS BIBLIOGRÁFICAS	97

Anexos	I
Anexo I Aplicación web	I
I.1 Repositorio front-end	I
I.2 Repositorio back-end	I
I.3 Diseño de la aplicación	I
I.4 Esquema de flujos conversacionales	I
I.5 Historias de usuario del proyecto	I
I.6 Diagrama de base de datos	II
I.7 Sitemap de la aplicación de gestión	II
I.8 Encuestas de usabilidad del sistema	II
I.9 Imágenes extras del desarrollo de los sprints	II

ÍNDICE DE FIGURAS

1.1	Marco general para la síntesis de programas. Elaborado por autor.	5
1.2	Asistentes inteligentes. Elaborado por autor.	6
1.3	Arquitectura aplicación web. Elaborado por autor.	7
1.4	Metodología ágil. Elaborado por autor.	11
2.1	Inicio de flujo. Elaborado por autor.	20
2.2	Primer de flujo. Elaborado por autor.	20
2.3	Segundo de flujo. Elaborado por autor.	21
2.4	Arquitectura de la aplicación. Elaborado por autor.	21
2.5	Nomenclatura de base de datos. Elaborado por autor.	22
2.6	Base de datos. Elaborado por autor.	22
2.7	Navegación de aplicación administrativa. Elaborado por autor.	23
2.8	Repositorio del aplicativo	26
2.9	Ramas del repositorio	27
2.10	Herramienta de docker para gestión de base de datos	27
2.11	Prototipo de media fidelidad.	30
2.12	Login.	30
2.13	Menú.	31
2.14	Burndown chart sprint 1. Elaborado por autor.	34
2.15	Modulo de empresa	36
2.16	Modulo de servicios	36
2.17	Horario por servicio	37
2.18	Burndown chart sprint 2. Elaborado por autor.	41
2.19	Entidades de dialoflow	44
2.20	Intents de dialoflow	44
2.21	Texto de respuesta de dialoflow	45
2.22	Url webhook.	45
2.23	Servicio ngrok	46
2.24	Intent dialogflow	46
2.25	Burndown chart sprint 3. Elaborado por autor.	48
2.26	Intent welcome. Elaborado por autor.	50
2.27	Saludo inicial. Elaborado por autor.	51
2.28	Activar intent. Elaborado por autor.	51

2.29 Intent información de la empresa. Elaborado por autor.	52
2.30 Activar intent de empresa. Elaborado por autor.	52
2.31 Burndown chart sprint 4. Elaborado por autor.	55
2.32 Intent ubicación. Elaborado por autor.	58
2.33 Intent servicio. Elaborado por autor.	59
2.34 Configurar intent agendar. Elaborado por autor.	59
2.35 Parámetros de intent agendar. Elaborado por autor.	60
2.36 Parámetros de intent agendar. Elaborado por autor.	60
2.37 Burndown chart sprint 5. Elaborado por autor.	64
2.38 Agendar cita validación fecha. Elaborado por autor.	66
2.39 Respuesta cita. Elaborado por autor.	66
2.40 Burndown chart sprint 6. Elaborado por autor.	68
2.41 API google calendar. Elaborado por autor.	71
2.42 Configuración de Facebook Dialoflow. Elaborado por autor.	71
2.43 Messenger bot. Elaborado por autor.	72
2.44 Chatbot Whatsapp. Elaborado por autor.	72
2.45 Burndown chart sprint 7. Elaborado por autor.	75
3.1 Pantalla de login	76
3.2 Pantalla menú de opciones	77
3.3 Pantalla información de la empresa	77
3.4 Pantalla gestión de servicios	78
3.5 Pantalla creación de servicio	78
3.6 Pantalla actualización de servicio	79
3.7 Pantalla horarios por servicio	79
3.8 Pantalla gestión de horarios	80
3.9 Pantalla creación de horario	80
3.10 Pantalla actualización de horario	81
3.11 Plataforma Dialogflow con intents	81
3.12 Plataforma Dialogflow con intents	82
3.13 Plataforma WhatsApp con chatbot activo	82
3.14 Grupo evaluado. Elaborado por autor.	90
3.15 Evaluación para las empresas de servicios y desarrolladores. Elaborado por autor.	90
3.16 Evaluación para las personas naturales y desarrolladores. Elaborado por autor.	91
3.17 Resumen de evaluación de chatbot por las empresas. Elaborado por autor.	91

3.18 Resumen de evaluación de chatbot por los desarrolladores. Elaborado por autor.	92
3.19 Resumen de evaluación de chatbot por las personas naturales. Elaborado por autor.	93

ÍNDICE DE CUADROS

2.1 Fases del proyecto. Elaborado por autor.	16
2.2 Backlog de proyecto. Elaborado por autor.	23
2.3 Sprint 1 backlog. Elaborado por autor.	29
2.4 Criterio de aceptación del sprint 1. Elaborado por autor.	31
2.5 Cumplimiento del sprint 1. Elaborado por autor.	32
2.6 Sprint 2 backlog. Elaborado por autor.	34
2.7 Criterio de aceptación del sprint 2. Elaborado por autor.	38
2.8 Cumplimiento del sprint 2. Elaborado por autor.	39
2.9 Sprint 3 backlog. Elaborado por autor.	42
2.10 Criterio de aceptación del sprint 3. Elaborado por autor.	46
2.11 Cumplimiento del sprint 3. Elaborado por autor.	47
2.12 Sprint 4 backlog. Elaborado por autor.	49
2.13 Criterio de aceptación del sprint 4. Elaborado por autor.	52
2.14 Cumplimiento del sprint 4. Elaborado por autor.	53
2.15 Sprint 5 backlog. Elaborado por autor.	55
2.16 Criterio de aceptación del sprint 5. Elaborado por autor.	61
2.17 Cumplimiento del sprint 5. Elaborado por autor.	62
2.18 Sprint 6 backlog. Elaborado por autor.	65
2.19 Criterio de aceptación del sprint 6. Elaborado por autor.	67
2.20 Cumplimiento del sprint 6. Elaborado por autor.	67
2.21 Sprint 7 backlog. Elaborado por autores.	69
2.22 Criterio de aceptación del sprint 7. Elaborado por autor.	73
2.23 Cumplimiento del sprint 7. Elaborado por autor.	74
3.1 Prueba de caja negra 01: Ingreso al sistema de gestión de servicios	83
3.2 Prueba de caja negra 02: Actualizar datos de la empresa	83
3.3 Prueba de caja negra 03: Creación de un servicio	84
3.4 Prueba de caja negra 04: Actualización de un servicio	85
3.5 Prueba de caja negra 05: Creación de horario sobre un servicio	85
3.6 Prueba de caja negra 06: Actualizar de horario sobre un servicio	86
3.7 Prueba de caja negra 07: Agendar cita	87
3.8 Categoría y subcategoría MARS.	88

RESUMEN

El proceso de agendamiento de citas depende mucho de consultar tanto la disponibilidad del cliente como la disponibilidad del servicio, normalmente un operador humano, inicia el proceso verificando las vacantes existentes para registrar un servicio, mientras que el usuario determina qué vacante es mejor, el tiempo que tarda en realizar el proceso de agendamiento de citas es considerable.

El presente trabajo, presenta una solución que consiste en utilizar un chatbot el cual puede ser accedido mediante varias plataformas como WhatsApp y Facebook Messenger teniendo como principal característica la disponibilidad durante las 24 horas del día, los 365 días del año, mejorando la atención al cliente además el chatbot puede ser customizado dependiendo del giro de negocio de la empresa. La solución propuesta presenta un backend desarrollado con el framework NestJS que permite agendar las citas directamente al calendario mediante la API de Google Calendar, el backend responde según el intent que se requiera, para esto Dialogflow realizar toda la lógica de coincidencia con la palabra clave y solicita al backend la respuesta de un intent específico a través del webhook. Finalmente, para customizar la información que presenta el chatbot se desarrolló una aplicación web administrativa, desarrollada con el framework Angular para modificar la información que presenta el chatbot.

El marco de trabajo utilizado en este proyecto es SCRUM permitiendo obtener una medición y distribución del trabajo basado en peso y prioridad de cada historia de usuario.

Para verificar la calidad del desarrollo, se utilizó la metodología MARS, obteniendo 4.35/5 en el apartado de funcionalidad.

Palabras clave: Chatbot, Agendamiento de citas, Angular, NestJs, Aplicación web, SCRUM

ABSTRACT

The main process of scheduling an appointment involves verifying availability on the calendar. The process begins when the operator checks the available slots, and then the customer selects a slot and confirms the day and time for the appointment. This process takes too long.

The proposed solution involves using a chatbot available on Facebook Messenger and WhatsApp platforms, which is available 24/7, 365 days a year, to enhance customer service. The chatbot can be customized according to the company's needs. The backend is developed using the NestJS framework and allows appointments to be directly scheduled on the calendar through the Google Calendar API. Dialogflow handles all the matching logic with keywords and requests the backend to respond to specific intents through the webhook. The web application enables customization of the chatbot's information, and the frontend is built using the Angular framework.

The project utilized the SCRUM framework, enabling measurement and distribution of work based on the weight and priority of each user story.

To verify the quality of development, the MARS methodology was employed, resulting in a functionality score of 4.35/5.

Keywords: Chatbot, Appointment scheduling, Angular, NestJs, Web application, SCRUM

1 INTRODUCCIÓN

1.1 DESCRIPCIÓN DEL PROBLEMA

La era digital ha transformado nuestra cultura de comunicación y la forma en la cual interactuamos con los demás, estos cambios se dan mediante una evolución constante de la tecnología. Es así, como esta era permite hacer tareas de manera óptima utilizando al máximo todos los recursos tecnológicos [1]. Aquí es donde las empresas tienen un papel de suma importancia ya que deben aprovechar estos nuevos recursos para brindar servicios de mejor calidad al cliente, dado el crecimiento de los mercados y el incremento constante de la competencia ya no es suficiente ofrecer un producto de calidad a precio razonable que satisfaga una necesidad, si no, es primordial que estén acompañados de un servicio diferenciador. [2]

Según el INEC en su informe sobre calidad de servicios, 45.2% piensan que las empresas públicas y privadas ofrecen igual calidad de servicio, mientras que el 37.4% creen que las públicas son peores y solo un 12% consideran que las privadas son mejores. El tiempo y la calidad de respuesta influyen en la calidad de servicio y en ambos casos, ni las públicas ni las privadas superan el 50% de satisfacción por parte del usuario final. [3]

Las empresas a menudo no comprenden que los usuarios buscan velocidad, eficiencia y seguridad al momento de consultar un servicio o producto de su interés, razón por la cual sus servicios de atención al cliente son de muy baja calidad, ya que no cuentan con los recursos ni el personal suficiente para poder ofrecer disponibilidad 24/7. Actualmente, empresas que basan su servicio en atención al cliente o en generación de reservas han optado por implementar chatbots para ahorrar tiempo y recursos en la gestión de citas y recordatorios para sus clientes.

Los chatbots están diseñados para simular una conversación en lenguaje natural con el usuario. Al recibir una entrada en forma de texto o voz, el chatbot consulta su base de

conocimientos y da una respuesta lo más parecida a la de un humano. [4] Esto permite a las interfaces conversacionales, crear una experiencia rápida y eficiente para el usuario, logrando mejorar la satisfacción y lealtad de los clientes. [5]

1.2 OBJETIVOS

1.2.1 Objetivo General

Desarrollar un chatbot para el agendamiento de citas por servicios profesionales usando interfaces conversacionales, mediante la red social Facebook y la aplicación de mensajería instantánea WhatsApp para ser agendadas mediante la API de Google calendar.

1.2.2 Objetivos Específicos

- Diseñar la arquitectura y el flujo conversacional
- Implementar los flujos conversacionales mediante un webhook para la comunicación entre Facebook, WhatsApp y Dialogflow.
- Conectar las citas con el API de Google calendar
- Probar la usabilidad del sistema con la metodología MARS.

1.3 ALCANCE

El presente trabajo de titulación tiene como objetivo principal diseñar un chatbot con el fin de mejorar la experiencia de atención al cliente de una empresa. Se utilizará Dialogflow para crear interfaces conversacionales disponibles en Facebook Messenger y WhatsApp.

El chatbot permitirá a los clientes realizar agendamientos de citas, mejorando los tiempos de respuesta y satisfacción. Serán agendados en Google Calendar y atendidos de forma personalizada por un representante. La implementación del chatbot ahorrará tiempo y recursos en la gestión de citas y permitirá al personal enfocarse en otras tareas.

1.4 MARCO TEÓRICO

1.4.1 Procesamiento del lenguaje natural (PNL)

El procesamiento del lenguaje natural (PLN) se centra en las interacciones entre el lenguaje humano y las computadoras. Para lograr esto, se implementa la inteligencia artificial, la cual analiza el lenguaje humano sintácticamente, semánticamente y contextualmente, lo que permite comprender la intención que quiere expresar el usuario [6], como se muestra en la figura.

1.1



Figura 1.1: Marco general para la síntesis de programas. Elaborado por autor.

El PLN analiza la estructura gramatical y la sintaxis del lenguaje humano para identificar la intención del usuario, incluyendo la identificación de palabras clave, la comprensión de la relación entre palabras y frases y la identificación de patrones sintácticos para determinar el contexto de la conversación.[7] De esta manera, el PLN puede identificar la intención del usuario a partir de los sinónimos o abreviaciones utilizadas en la conversación.[8]

1.4.2 Interfaz conversacional

Una interfaz conversacional permite simular una conversación humana a través de un dispositivo. [9] Se utiliza en aplicaciones o servicios que requieren interacción directa con el usuario a través de mensajes de texto o voz. [10].

Los gigantes tecnológicos han adoptado los asistentes inteligentes. Estos permiten realizar búsquedas, agendar eventos, generar recordatorios y demás funcionalidades, lo que brinda una experiencia inmersiva tanto en pantallas como en dispositivos.[11] La interacción se produce de forma natural a través del lenguaje humano con dispositivos y asistentes como se muestra en la figura. 1.2

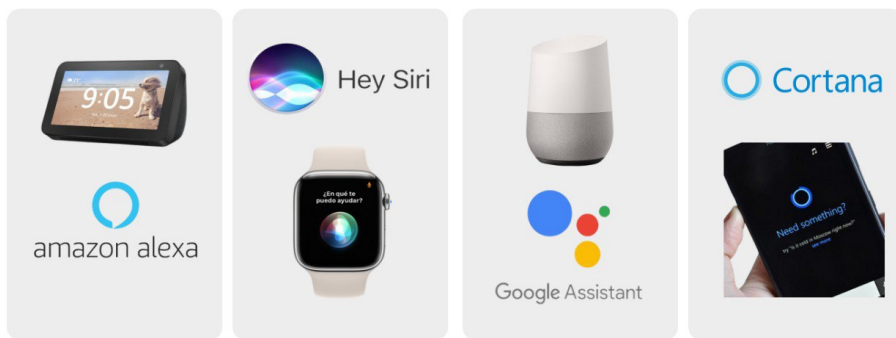


Figura 1.2: Asistentes inteligentes. Elaborado por autor.

1.4.3 Webhook

Los Webhooks son una herramienta útil para la integración de servicios en línea. Se invocan con una llamada HTTP con método POST cuando se produce un evento en una API externa. Esto permite a los desarrolladores complementar acciones y crear nuevas conexiones, incluyendo desde conexiones a bases de datos hasta la comunicación con otras APIs [12].

Dado que los webhooks no siguen un estándar, el desarrollador puede diseñar webhooks con autonomía y libertad dependiendo de la API, a la cual se desea generar una respuesta a un evento [13].

1.4.4 Modelo vista controlador

Es un patrón de diseño de aplicaciones de software en donde se separa la lógica de negocio (backend) de la interfaz de usuario (frontend) y de la representación de datos. El usuario realiza la acción en la interfaz, en donde se activará el controlador emitiendo un evento que notificará al modelo de esta acción del usuario, este responderá con información o cambiará la vista [14].

Existen tres principales componentes:

- ❑ **Modelo:** Tiene acceso a los datos y las principales funcionalidades que definen la lógica de negociación.

- ❑ **Vista:** Muestra los datos que recibió del modelo para ser mostrados en la interfaz.

- ❑ **Controlador:** Contiene las peticiones y emite un evento cada vez que el usuario intenta acceder a este controlador.

1.4.5 Aplicación web

Es una aplicación de software que no necesita ser instalada en el sistema operativo, permite al usuario final acceder a un servidor Web a través de un navegador con acceso a la red como internet o intranet. Se codifica en lenguajes como (HTML, JavaScript, etc) [15].

Existen distintos tipos, Las páginas estáticas, muestran documentos escritos en HTML, que han obtenido desde un servidor web, en donde estos HTML solo presentan la información sin la posibilidad de interactuar con ellos, mientras que para generar páginas dinámicas se utiliza un servlets con tecnologías tipo JSP para realizar las funciones o peticiones a la base de datos, mientras que del lado del cliente se muestra el HTML que interactuara con el usuario para enviar las peticiones al servidor [16], como se muestra en la figura 1.3.

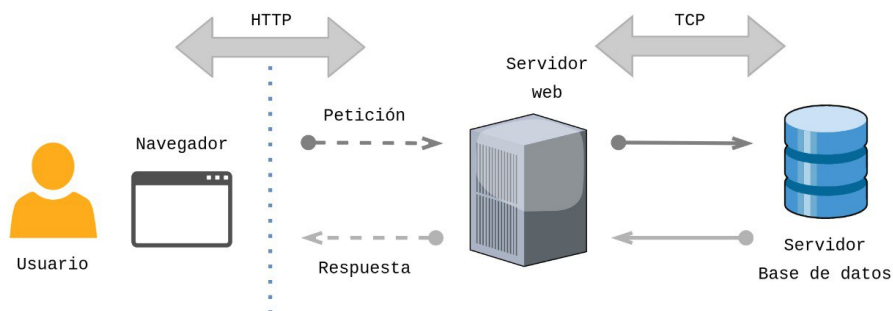


Figura 1.3: Arquitectura aplicación web. Elaborado por autor.

1.5 HERRAMIENTAS DE DESARROLLO

Para la implementación del sistema de recomendación se usaron las siguientes herramientas:

1.5.1 Visual Studio Code

Es un editor de código fuente, disponible para Windows, macOS y Linux, para escritorio que incluye soporte para JavaScript, TypeScript y Node.js adicional permite la integración

de paquetes que brindan temas, idiomas, depuradores y demás comandos que facilitan la implementación de código [17].

1.5.2 Draw io

Es un editor de diagramas gratuito en línea desarrollado por JGraph Ltd, que permite realizar diagramas de diferentes tipos como UML, modelos de base de datos, una de las principales características es que permite el trabajo colaborativo [18].

1.5.3 Git

Git es el sistema de control de versiones gratuito para multiplataformas, creado por Linus Torvalds, el cual permite la gestión de código de manera local o en la nube a través de ramas que no necesariamente dependen una de la otra. Tiene la licencia GNU 2.0 (General public license) [19].

1.5.4 Github

Es una plataforma de desarrollo colaborativo de software de código el cual permite almacenar código, con el control de versiones Git, también maneja diferentes tipos de repositorios privados como públicos [20].

1.5.5 Gitkraken

Es una plataforma que funciona como cliente de git con una interfaz amigable, permitiendo gestionar uno o varios repositorios de forma local utilizando comandos de git para almacenarlos en su repositorio en la nube [21].

1.5.6 MySQL

Es un SGBD (sistema gestor de bases de datos) de código abierto, que proporciona un servidor de base de datos. robusto, multiusuario y multi hilo que puede ser integrado a cualquier sistema utilizando una conexión correcta a la base de datos [22].

1.5.7 Typescript

También conocido como ECMAScript es un lenguaje de programación orientado hacia navegadores, una de las características principales es el tipado estático que define el tipo de dato de las variables permitiendo tener control sobre el código y el flujo del mismo, si bien fue diseñado inicialmente para fragmentos cortos de código incrustados en una página web, ha evolucionado hasta convertirse en una herramienta completa para escribir cualquier tipo de aplicaciones web [23].

1.5.8 NestJS

Es un marco que permite construir aplicaciones del lado del servidor, escalables y eficientes utilizando TypeScript permite utilizar peticiones de HTTP lo cual lo hace accesible para cualquier API [24].

1.5.9 HTML5

Hypertext Markup Language brinda tres características importantes: estructura, estilo y funcionalidad, es la combinación entre HTML, CSS y Javascript, permitiendo organizar tanto contenido dinámico como estático, los documentos HTML están estrictamente organizados, ya que cada parte del documento está organizada y declarada por etiquetas específicas [25].

1.5.10 SCSS

Sassy Cascading Style Sheets es un preprocesador de CSS (lenguaje de hojas de estilo en cascada) este usa dos sintaxis, la primera los anidamientos se especifican con llaves y cada regla termina en punto y coma, mientras que la sintaxis indentada es mediante tabulaciones en lugar de llaves, ambos sirven para simplificar el diseño y la presentación de la página [26].

1.5.11 Angular

Es un marco de diseño de aplicaciones, esta plataforma permite crear aplicaciones web escalables e integrar diferentes colecciones de bibliotecas de componentes o funcionalidad que permitirán la interacción con el cliente y el servidor [27].

1.5.12 PrimeNG

Es una librería de componentes de código abierto, gratuito y disponible para varios frameworks entre los más populares Angular, fue desarrollado por PrimeTek Informatics, bajo la licencia MIT, tiene alrededor de 90 componentes y documentación detallada que permite manejar sus eventos y modificarlos de ser necesario [28].

1.5.13 Angular material

Es una librería de componentes diseñada especialmente para ser utilizado con el framework angular, bajo la licencia MIT, posee componentes internacionalizados y permite su personalización dentro de los límites de la especificación de Material Design [29].

1.5.14 Google Dialogflow

Es una plataforma que comprende el lenguaje natural y permite diseñar interfaces conversacionales, permitiendo integrarlos sobre diferentes aplicaciones móviles, web o sistemas de respuesta de voz interactiva. Interactuando con el cliente de una manera fácil ya sea a través de texto o con voz sintética [30].

1.5.15 Google Calendar API

Es una API RESTful de Google Calendar para acceder a través de peticiones de HTTP a la mayoría de las funciones disponibles en la interfaz web [31].

1.6 METODOLOGÍA ÁGIL

Las metodologías ágiles se enfocan en tener procesos altamente definidos para obtener software en tiempo y costo con la calidad requerida, ya que permiten adaptar los proyectos de forma flexible, autónoma y eficazmente dependiendo de los condicionantes del proyecto permitiendo retroalimentar y mejorar el mismo en cada iteración [32]. Dentro de los principales valores de la metodología ágil se encuentran: individuos e interacciones sobre los procesos y herramientas, priorizar un software funcional sobre documentación extensa, colaborar con el cliente sobre la negociación de contratos y ser flexibles al cambio sobre seguir un plan o cronograma [33].

Esta metodología, se implementa en varios procesos ágiles como lo son: XP (Programación Extrema), KANBAN, Agile inception, SCRUM.

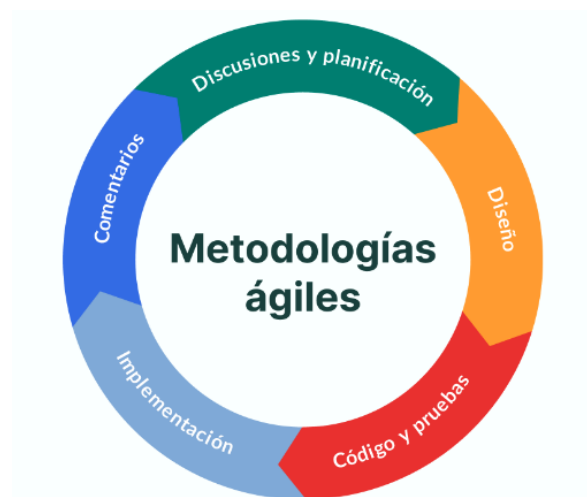


Figura 1.4: Metodología ágil. Elaborado por autor.

1.6.1 Extreme Programming XP

Es una metodología de desarrollo de software Agile que se enfoca en mejorar la calidad del software a través de la colaboración continua y la comunicación constante entre el equipo, utilizando prácticas como programación en parejas, diseño y planificación del proyecto con el cliente. Además, XP promueve la simplicidad y la flexibilidad en el desarrollo de software.

1.6.2 SCRUM

Se define como un proceso empírico, iterativo e incremental de desarrollo, donde se busca maximizar la retroalimentación sobre el desarrollo, permitiendo corregir problemas y mitigar riesgos, en cada iteración. Se define el Product Backlog, que contiene tanto los requerimientos funcionales como los no funcionales, estos pueden estar especificados según requiera la organización en features, casos de uso, diagramas de flujos [33].

1.6.2.1 Equipo de Scrum

- ❑ **Scrum master:** Es el responsable de establecer Scrum, deberá capacitar, apoyar y promover la teoría y la práctica de Scrum durante el proyecto.

- ❑ **Product owner:** Gestiona el product backlog, además deberá maximizar el valor como tal del producto.

- ❑ **Desarrolladores:** Están a cargo de cada incremento y se responsabilizan por desarrollar el producto según el objetivo planteado.

1.6.2.2 Eventos de Scrum

Un sprint es la iteración que normalmente dura entre 2 a 4 semanas, durante este tiempo se realizan las tareas necesarias para cumplir con los criterios de aceptación de cada historia de usuario seleccionada a ser realizada en el sprint, los cambios no son aceptados durante este tiempo [34].

1.6.2.3 Planificación de Sprint

Este evento inicia el Sprint, se establecen las historias de usuario a realizarse, se determina cómo se realizará y los encargados de cada tarea. Estas reuniones son conducidas por el Product owner, deben asistir los miembros del equipo y el scrum master.

1.6.2.4 Daily Scrum

Este evento se realiza cada día hasta culminar con el sprint, deberá durar aproximadamente 15 minutos, el objetivo de esta reunión es que el equipo conozca el estado del proyecto diariamente, también se realizan las siguientes preguntas ¿Qué hicieron ayer?, ¿Tiene algún bloqueo en las tareas realizadas?, ¿Qué tienen planeado hacer el día de hoy?, ¿Qué dificultad puede tener para cumplir sus tareas asignadas?.

1.6.2.5 Sprint review

Este evento es el resultado del sprint donde se presentan los entregables al product owner, se revisa que se cumplan los criterios de aceptación, también las tareas que podrían tener ciertas adaptaciones. De tener trabajo pendiente, este deberá ser ajustado al próximo sprint.

1.6.2.6 Sprint retrospective

Este evento permite discutir sobre posibles mejoras, acontecimientos, que se tuvo durante el sprint, lo que se podría mejorar y lo que definitivamente no se debe hacer. En esta reunión deben participar desde scrum Master hasta los desarrolladores, generalmente esta reunión dura entre 15 a 30 minutos.

1.6.2.7 Artefactos

Product backlog

Es una lista del trabajo, estos están listos para ser seleccionados para cualquier evento de planificación de Sprint, los desarrolladores a cargo son los que realizar el trabajo y determinarán el tamaño de esta actividad.

Sprint backlog

Es la lista de trabajo que se han seleccionado para ser realizados en el Sprint a realizarse, los ítems seleccionados durante el Sprint no pueden cambiar hasta terminar la ejecución del Sprint.

Burndown chart

Es un gráfico que representa el trabajo pendiente en el backlog del Sprint en curso vs el progreso del sprint, permitiendo visualizar rápidamente el trabajo que se está realizando rápidamente.

2 METODOLOGÍA

2.1 POR QUÉ SCRUM

Para la implementación del presente proyecto se ha decidido combinar SCRUM y XP son metodologías de desarrollo de software que se complementan y se pueden combinar. SCRUM se enfoca en la planificación y gestión del proyecto, mientras que XP se enfoca en la mejora continua y la calidad del código. La combinación de ambos permite obtener una planificación eficiente y una mejora constante en la calidad del software, lo que se traduce en un desarrollo ágil y una mayor satisfacción del cliente. Además, la incorporación de historias de usuario en SCRUM y la práctica de refactorización de código en XP permiten involucrar al usuario y al equipo de desarrollo en el proceso de construcción del software.

Entre las principales definiciones se tiene:

- Duración de sprint:** 15 días
- Validación de historias de usuario:** Se realizarán historias de usuario sencillas con criterios de aceptación claros y mandatorios
- Peso:** Cada historia deberá tener un peso utilizando FirePoker.
- Diseño:** Tanto el flujo como las pantallas de usuario deberán ser simples y funcionales

2.2 EQUIPO SCRUM

El equipo de trabajo de SCRUM trabaja de manera horizontal por lo cual no tiene jerarquías, ya que todos se encuentran al mismo nivel, enfocados en un objetivo a la vez. Los roles que se cumplirán en este proyecto son:

- ❑ **Scrum Master:** Msc. Adrian Eguez; Su principal responsabilidad será ayudar a los miembros del equipo a comprender los principios y como implementar SCRUM durante todo el proyecto.
- ❑ **Product Owner:** Ing. Cristian Lara; Su principal responsabilidad será maximizar el valor del producto, teniendo como enfoque las principales necesidades del cliente final.
- ❑ **Desarrolladora:** Alisson López; Su principal responsabilidad es diseñar y desarrollar el producto.

2.3 INGENIERÍA DE REQUERIMIENTOS

Es un proceso esencial para el desarrollo de software ya que permite establecer las características, servicios y restricciones que deberá proveer el sistema este genera cierta documentación para poder darle seguimiento de cumplimiento al desarrollo. El presente documento utilizara como documentación historias de usuario, que permiten describir el flujo de cada sección y contienen el peso y los criterios de aceptación a ser validados en la entrega del aplicativo [35].

Según la reunión mantenida con el Product Owner, se decidió agrupar el presente proyecto en fases.

Tabla 2.1: Fases del proyecto. Elaborado por autor.

Fases	Requerimiento	Sección
Flujo conversacional	<ul style="list-style-type: none"> ❑ Flujo de preguntas frecuentes sobre empresa ❑ Flujo de agendamiento de citas ❑ Recopilación información posible cliente 	Diseño

Implementación en Dialogflow	<ul style="list-style-type: none"> <input type="checkbox"/> Generación de entidades relacionadas a cada intent <input type="checkbox"/> Generación de intents según el flujo 	Implementación Dialogflow
Webhook	<ul style="list-style-type: none"> <input type="checkbox"/> Implementación de intents en webhook <input type="checkbox"/> Consulta de tablas, información a ser mostrada en cada intent 	Backend
Información empresa	<ul style="list-style-type: none"> <input type="checkbox"/> Gestionar información empresa <input type="checkbox"/> Gestionar información de ubicación <input type="checkbox"/> Gestionar servicios. 	Backend
Conexión con API Google calendar	<ul style="list-style-type: none"> <input type="checkbox"/> Establecer conexión con API Google Calendar <input type="checkbox"/> Consumir eventos que permitan generar las citas en la API 	Backend
Gestionar empresa a través de interfaz	<ul style="list-style-type: none"> <input type="checkbox"/> Diseño de aplicativo interfaz media fidelidad <input type="checkbox"/> Desarrollo de aplicativo para gestionar empresa 	Frontend

2.3.1 Recopilación y análisis de información

Se realizó un benchmarking de aplicativos para la gestión de citas, tanto virtuales como físicas. Los siguientes métodos fueron encontrados:

- Manejo físico:** Son barreras o guías en donde se crea una fila para ser atendido, normalmente se utilizan en parques de diversión y se accede de manera presencial.
- Manejo basado en tickets:** Se utilizan tickets numerados que se adquieren cuando

el cliente llega y espera a ser llamado. Se puede monitorear el flujo de citas en tiempo real y generalmente se usan en servicios estatales, y se accede de manera presencial.

- ❑ **Manejo humano:** Servicios que pueden ser agendados por interacción humana, por ejemplo, a través de un call center. El servicio se puede realizar de manera virtual o física.
- ❑ **Manejo digital:** los usuarios generan citas a través de sistemas digitales y pueden monitorear el flujo en tiempo real. Sin embargo, el desarrollo de estas aplicaciones puede ser costoso para la organización. El servicio se puede realizar de manera virtual.

La importancia de un sistema efectivo de agendamiento de citas optimizará el tiempo de espera de los clientes y la productividad de los colaboradores. Según Debmedia, se pierden aproximadamente 6 meses de vida en espera en filas y una persona promedio espera hasta 43 minutos en línea para hacer una cita [36].

Implementar una aplicación accesible en todo momento, como en Facebook Messenger y WhatsApp con la integración de Google Calendar, reducirá costos, mejorará la experiencia del usuario y mantendrá informados a los colaboradores sobre las citas agendadas.

2.4 SPRINT

En esta sección se detalla el trabajo resultante de cada Sprint. En el (Anexo I.9) se adjunta un archivo Excel con la medición de cada sprint y la distribución de puntos.

2.4.1 Sprint 0

Sprint planning

Objetivos del sprint

- ❑ Diseñar el flujo conversacional del chatbot.

- ❑ Definir la arquitectura de la aplicación.
- ❑ Realizar el modelo de bases de datos.
- ❑ Definir el diagrama de navegación de la aplicación web.
- ❑ Realizar historias de usuario.
- ❑ Configurar el ambiente de desarrollo.

Ejecución del sprint

2.4.1.1 Diseño del flujo conversacional

El flujo conversacional de los chatbots es una estructura que se utiliza para guiar la conversación con el usuario. Su importancia radica en que permite al usuario llegar al objetivo deseado de manera más eficiente, evitando distracciones o conversaciones no relevantes. Además, ayuda a mejorar la experiencia del usuario, ya que puede ser más fácil y rápido encontrar la información o realizar la tarea deseada.[37].

2.4.1.2 Flujos conversacionales

El proyecto cuenta con dos flujos conversacionales principales para atender a los usuarios. El primer flujo permite brindar información sobre la empresa, su ubicación y los servicios que ofrece. Por otro lado, el segundo flujo está enfocado en el agendamiento de citas, permitiendo al usuario seleccionar el servicio de su preferencia y proporcionar los detalles necesarios para la cita. La estructura de los flujos conversacionales se detalla en el (Anexo I.4). Para iniciar la conversación con el bot, se utiliza cualquier saludo como: hola, buenas tardes, ya que dialogflow tiene este intent como default. En la figura 2.1 se muestra el diagrama de inicio del flujo.

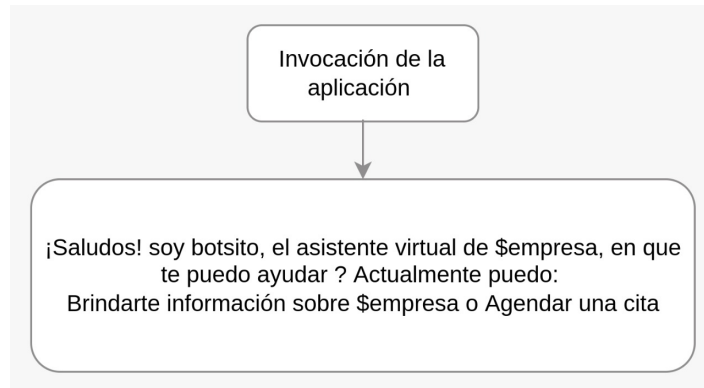


Figura 2.1: Inicio de flujo. Elaborado por autor.

Primer flujo

El objetivo del primer flujo es contestar preguntas frecuentes como los servicios que ofrece la empresa, ubicación en la que se encuentra. En la figura 2.2 se puede observar el flujo diseñado para esta sección.

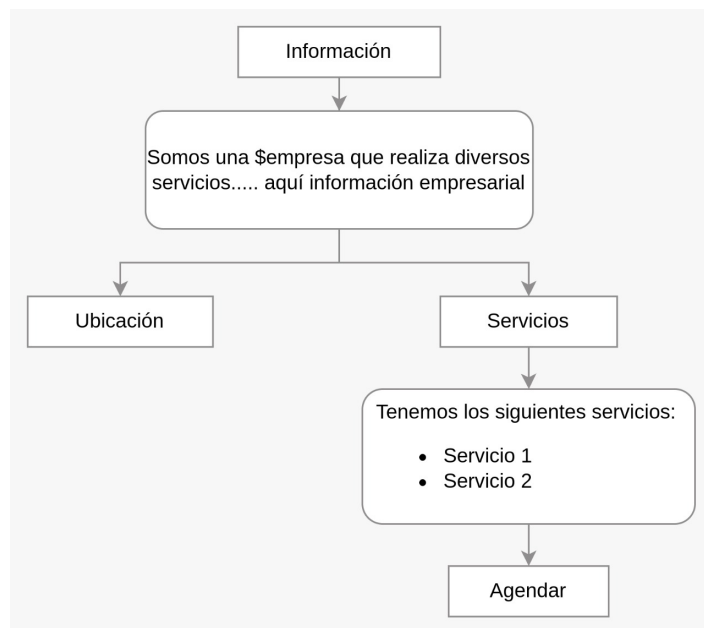


Figura 2.2: Primer de flujo. Elaborado por autor.

Segundo flujo

El objetivo del segundo flujo es agendar la cita como tal, para esto no es necesario que el usuario navegue por el primer flujo simplemente sino puede agendar una cita directamente consultando los servicios disponibles de la empresa. En la figura 2.3 se muestra el flujo diseñado para esta sección.

El objetivo del segundo flujo es agendar citas, no es necesario que el usuario navegue

por el primer flujo. El usuario puede consultar directamente los servicios disponibles de la empresa y agendar una cita. La figura 2.3 se observa el flujo diseñado para esta tarea.

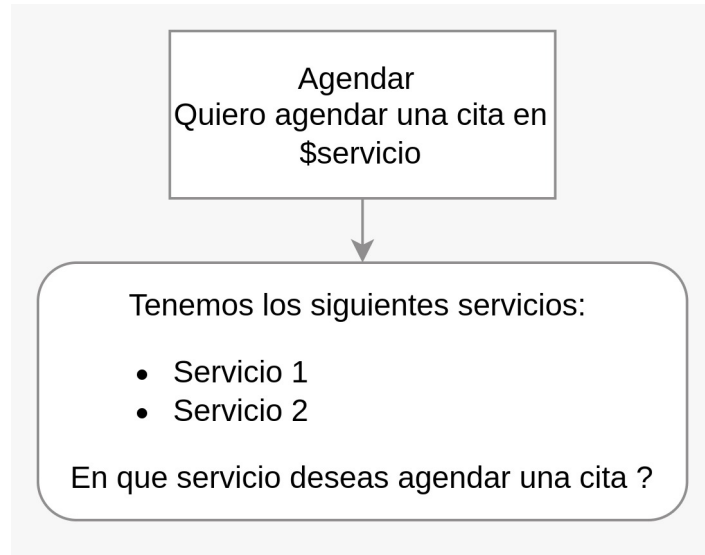


Figura 2.3: Segundo de flujo. Elaborado por autor.

2.4.1.3 Arquitectura de la aplicación

El proyecto implementará la arquitectura Modelo-Vista-Controlador (MVC) para una mejor organización y mantenibilidad. La aplicación estará dividida en tres componentes: modelos, vistas y controladores. Cada intent tendrá su propio controlador principal que gestionará las funcionalidades a través de diferentes servicios. Se usará la API de Google Calendar para agendar citas y se almacenará toda la información en una base de datos relacional MySQL.

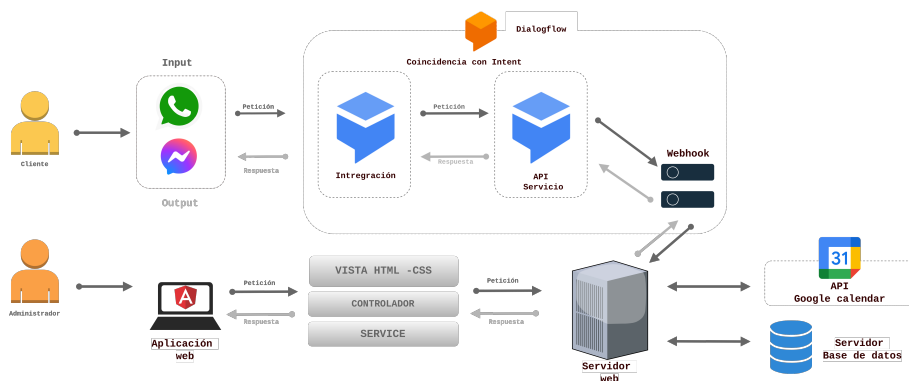


Figura 2.4: Arquitectura de la aplicación. Elaborado por autor.

2.4.1.4 Modelo de bases de datos

El modelo de base de datos se construyó basándose en un enfoque de base de datos relacional. La figura 2.5 muestra las nomenclaturas que utiliza la empresa para identificar las características de los campos.

Obligatorios, Indices, Posibles valores
*C Obligatorio Crear
C Puede Crearse
U Actualizarse
I Indice(Puede usarse para búsqueda)
(Valor1, valor2, valor3, valor4) Valores posibles (1,0)
A Letras
1 Números
\$ Símbolos especiales (./*-'?!")

Figura 2.5: Nomenclatura de base de datos. Elaborado por autor.

El modelo consta de 3 entidades, encargadas de almacenar la información de la empresa, ubicación, servicios y citas agendadas.

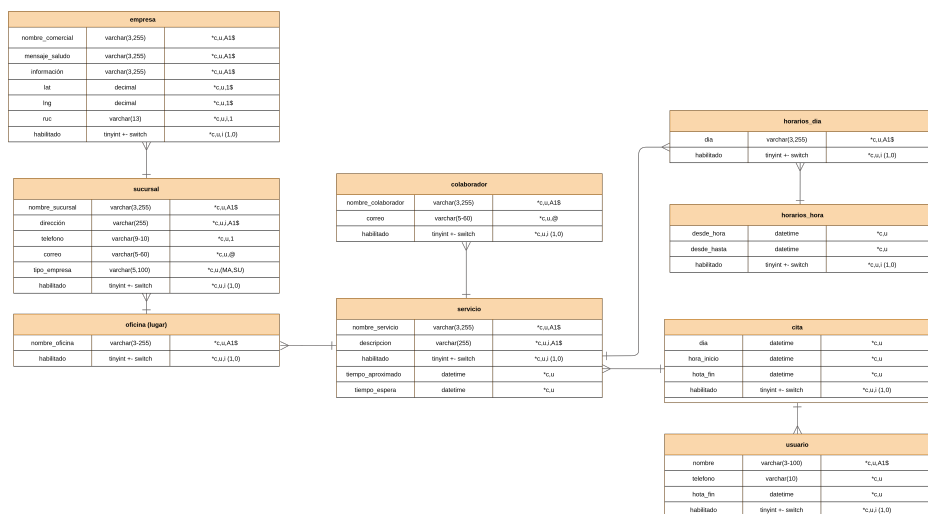


Figura 2.6: Base de datos. Elaborado por autor.

2.4.1.5 Diagrama de navegación

Si bien el sistema está enfocado en el diseño y desarrollo de un chatbot, también se han realizado pantallas para la gestión de la información que va a mostrar el chatbot en las distintas plataformas.

En la figura 2.7 se muestran las rutas que tendrá la aplicación administrativa.

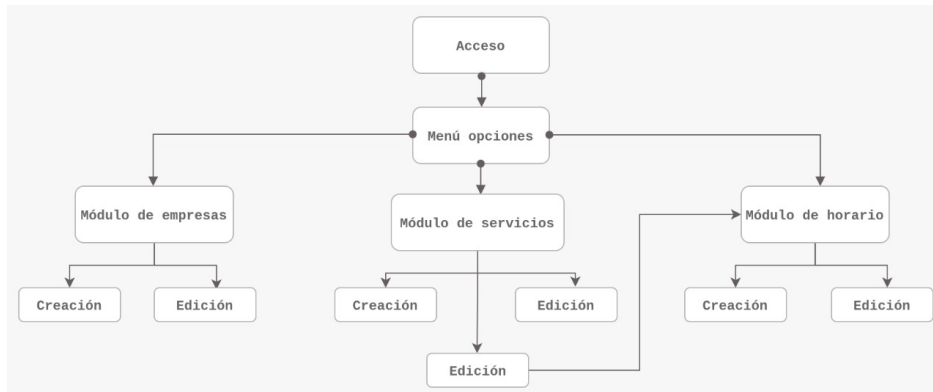


Figura 2.7: Navegación de aplicación administrativa. Elaborado por autor.

2.4.1.6 Definición de historias de usuario

Una vez que se han expresado las necesidades, por parte del Product owner, se generan las historias de usuario, donde se describen las necesidades enfocadas en el flujo que tendría el usuario con sus respectivos criterios de aceptación.

2.4.1.7 Product backlog

El product backlog que se ha generado para el presente proyecto es en base a las historias de usuario obtenidas. Sin embargo esta lista puede cambiar a medida que se desarrolle el proyecto, ya que se pueden obtener nuevas características o corrección de bugs. El peso y la prioridad de las historias de usuario han sido estimadas utilizando Planning poker por el Scrum master, product owner y la desarrolladora.

Tabla 2.2: Backlog de proyecto. Elaborado por autor.

HU	Descripción	Peso	Prioridad	Sprint	Total Sprint
----	-------------	------	-----------	--------	--------------

1	Como usuario administrador, quiero visualizar una demo de la aplicación web administrativa para gestionar la información que se mostrará en el chatbot.	13	Alta	1	31
2	Como usuario administrador deseo poder iniciar sesión en el sistema de gestión de información del chatbot para poder realizar cambios y actualizaciones en la información que mostrará el chatbot.	13	Alta		
3	Como usuario administrador, deseo tener un menú claro y fácil de usar para navegar por las opciones disponibles en el sistema.	5	Baja		
4	Como usuario administrador, deseo crear y editar la información relacionada a la empresa que mostrará el chatbot.	13	Alta	2	39
5	Como usuario administrador quiero gestionar los servicios de la empresa, de manera eficiente y sencilla.	13	Alta		
6	Como usuario administrador, quiero ser capaz de gestionar los horarios disponibles para cada servicio de mi empresa para poder ofrecer horarios disponibles a los clientes.	13	Alta		
7	Como usuario administrador, quiero que mi charbot utilice la plataforma Dialogflow para mejorar su funcionamiento y proporcionar una experiencia más fluida.	13	Alta	3	29

8	Como usuario administrador, quiero que el chatbot pueda obtener información de la aplicación web para ofrecer una experiencia mas completa y personalizada a los usuarios finales.	8	Media		
9	Como usuario administrador deseo configurar el intent de default para cuando el chatbot no encuentra coincidencias en el flujo de conversación.	8	Baja		
10	Como usuario final, deseo ver un saludo personalizado en el chatbot para poder ver la información que se ha configurado previamente en la aplicación web.	21	Alta	4	34
11	Como usuario final, deseo que el chatbot muestre información sobre la empresa, la cual se ha configurado previamente en la aplicación web.	13	Baja		
12	Como usuario final, quiero que el chatbot muestre la ubicación de la empresa configurada en la aplicación web.	8	Media	5	47
13	Como usuario final, quiero poder visualizar los servicios que se han configurado en la aplicación web a través del chatbot, para tener acceso a esta información de manera rápida y eficiente.	13	Alta		
14	Como usuario final, quiero ser capaz de agendar una cita en un servicio específico a través del chatbot para tener una experiencia de usuario más eficiente y cómoda.	13	Alta		

15	Como usuario final, deseo poder agendar una cita en un servicio a una hora y fecha específica.	13	Media		
16	Como usuario administrador quiero que el chatbot solo permita agendar citas hasta con 5 días de anticipación.	8	Baja	6	29
17	Como usuario final, deseo aceptar o rechazar la cita que previamente estoy seleccionado.	13	Alta		
18	Como usuario administrador deseo visualizar mis citas en google calendar.	13	Media	7	39
19	Como usuario administrador deseo que el chatbot aparezca en Facebook Messenger.	13	Media		
20	Como usuario administrador deseo que el chatbot este disponible en WhatsApp.	13	Media		

2.4.1.8 Configuración ambiente de desarrollo

Se creó el repositorio donde se almacenará el código como se muestra en la figura 2.8, se procede a generar los proyectos y ramas base para empezar el desarrollo.

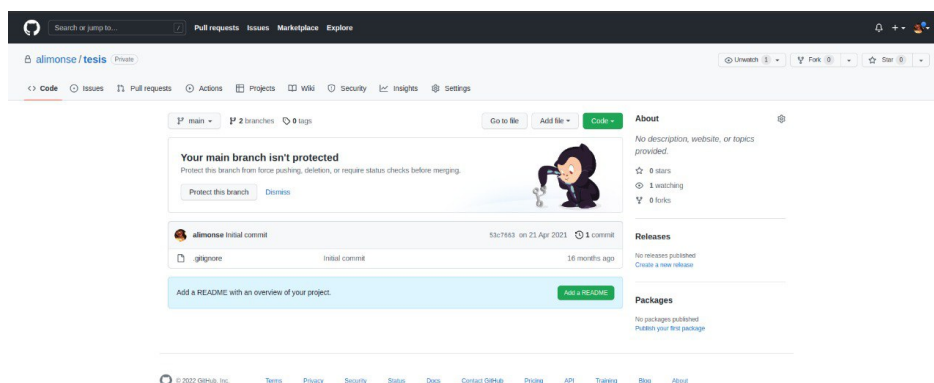


Figura 2.8: Repositorio del aplicativo

Una vez clonado el proyecto se procedió a crear las ramas a ser utilizadas en el proyecto.

En la figura 2.9 se muestran las ramas creadas “develop” en donde se realizará los distintos avances y “main” en donde se encontrará todo el código estable.

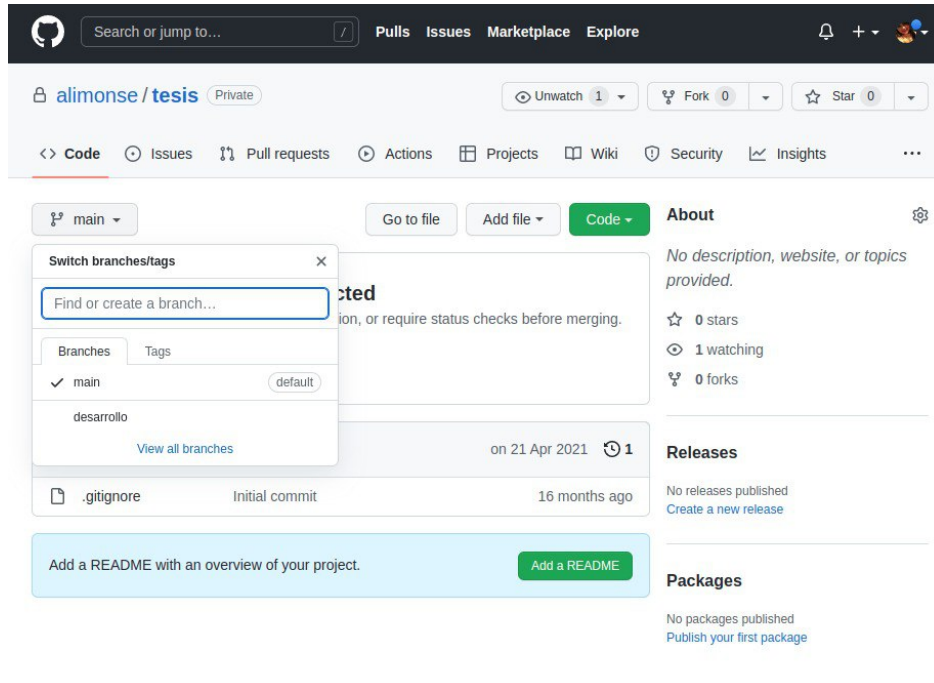


Figura 2.9: Ramas del repositorio

También se procedió a crear la base de datos a utilizarse, usando una instancia de docker para mysql. En la figura 2.10 se muestra el contenedor creado en el dashboard de docker.

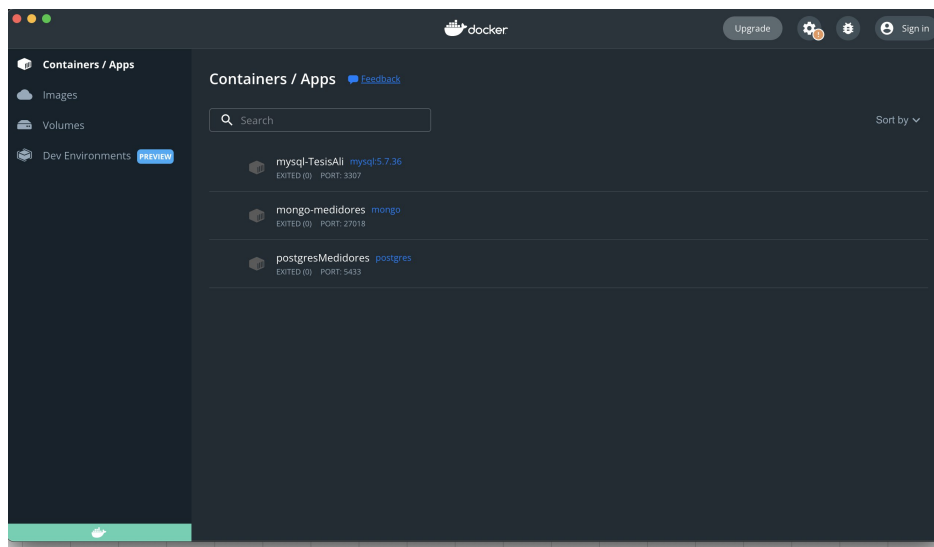


Figura 2.10: Herramienta de docker para gestión de base de datos

Sprint retrospective

¿Qué salió bien en la iteración?

- Se estableció la arquitectura de la aplicación que incluye conexiones entre la aplicación administrativa, la base de datos y los servicios de Dialogflow, Facebook Messenger y WhatsApp, así como la integración con Google Calendar a través de su API.
- Se diseñó dos flujos conversacionales uno con información relevante para la empresa y otro netamente con el proceso de agendar una cita.
- Se establecieron las rutas de navegación del aplicativo web de administración.
- Se definió las historias de usuario a realizarse durante el proyecto.
- Se crearon los repositorios necesarios para backend como frontend.

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué se puede mejorar?

Se puede mejorar el desarrollo al utilizar imágenes de bases de datos en Docker, en particular la imagen de MySQL.

2.4.2 Sprint 1

Sprint planning

Objetivos del sprint

- Crear un prototipo de media fidelidad para gestionar la información de empresas, localización, servicios, horarios por servicio.
- Crear el login para autenticación del usuario.
- Crear menú del aplicativo para navegación del sistema.

Sprint backlog

En esta ocasión el backlog del sprint 1 se seleccionó las historias 01,02,03 cuyas tareas se listan en la tabla 2.3

Tabla 2.3: Sprint 1 backlog. Elaborado por autor.

Código	Historia de usuario	Tareas de sprint
HU01	Como usuario administrador, quiero visualizar una demo de la aplicación web administrativa para gestionar la información que se mostrará en el chatbot.	<ul style="list-style-type: none"><input type="checkbox"/> Recolección de requerimientos.<input type="checkbox"/> Maquetación.<input type="checkbox"/> Animación maqueta.
HU02	Como usuario administrador deseo poder iniciar sesión en el sistema de administración de información del chatbot para poder realizar cambios y actualizaciones en la información que mostrará el chatbot.	<ul style="list-style-type: none"><input type="checkbox"/> Crear servicio REST para autenticar usuario.<input type="checkbox"/> Crear pantalla para ingresar credenciales de ingreso.<input type="checkbox"/> Crear ruta "Login".
HU03	Como usuario administrador, deseo tener un menú claro y fácil de usar para navegar por las opciones disponibles en el sistema.	<ul style="list-style-type: none"><input type="checkbox"/> El sistema mostrará las opciones del aplicativo.<input type="checkbox"/> El menú aparecerá después de loguearse.<input type="checkbox"/> El menú tendrá las siguientes opciones:<ul style="list-style-type: none">✧ Gestión empresa.✧ Gestión servicios.✧ Gestión horarios.

Ejecución del sprint

1. Se realizó un prototipo de media fidelidad en el aplicativo Figma, como se muestra en la figura 2.11. Estas pantallas servirán para gestionar la información que el bot va a consumir a través del webhook consultando a la base de datos.

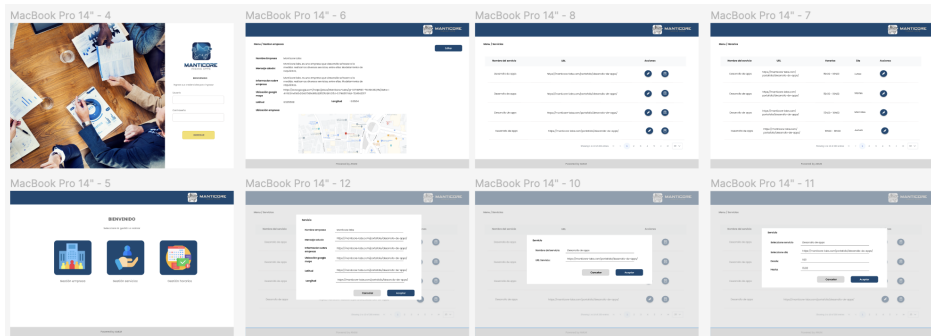


Figura 2.11: Prototipo de media fidelidad.

2. Se generó la sección tanto en backend como en frontend. En la figura 2.12 se muestra la sección de autenticación de usuario que se ha codificado en el sprint 1.

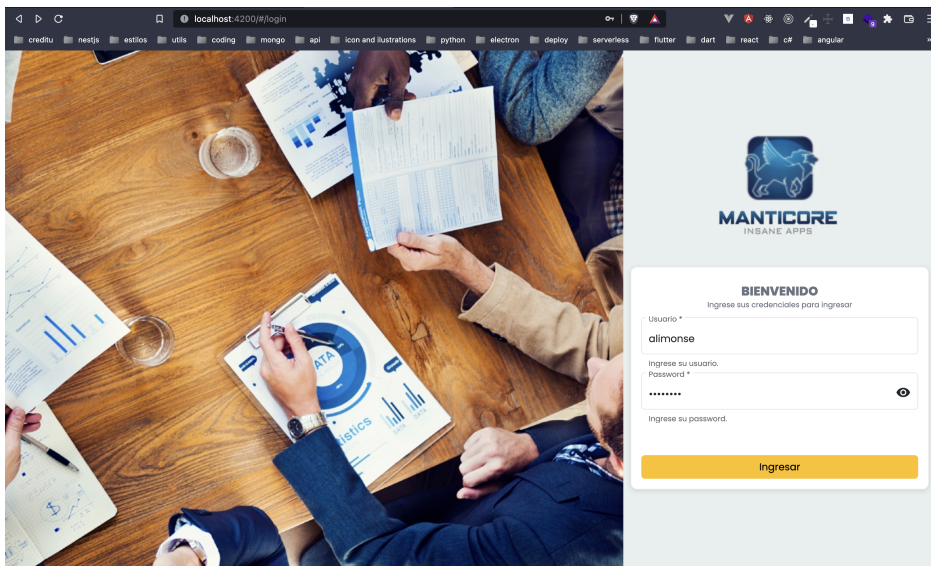


Figura 2.12: Login.

- a) Se creó la sección de autenticación, en backend.
- b) Se crearon los dtos y base de datos en la tabla login.
- c) Se creó los campos en el módulo login.
- d) Se creó la función de autenticación.

3. Se creó el menú para navegar por las opciones del aplicativo como se muestra en la figura 2.13



Figura 2.13: Menú.

- a) Se crearon las opciones en el frontend .
- b) Se creó la ruta para acceder al menú.
- c) Se creó las rutas de las opciones:
 - Gestión empresa.
 - Gestión servicios.
 - Gestión horarios.

Sprint review

Dado que se terminó el sprint 1, se realizó el sprint review, en el cual se evidencia el cumplimiento de las historias de usuario desarrolladas, el cual se encuentra en la tabla 2.4

Tabla 2.4: Criterio de aceptación del sprint 1. Elaborado por autor.

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

HU01	<ul style="list-style-type: none"> <input type="checkbox"/> La maqueta debe permitir la visualización de la información que se mostrará en el chatbot. <input type="checkbox"/> La maqueta debe ser interactiva y mostrar cómo se realizarán las acciones de gestión de información. 	100%
HU02	<ul style="list-style-type: none"> <input type="checkbox"/> El sistema permitirá el acceso al aplicativo, mediante un usuario y contraseña. <input type="checkbox"/> Si el usuario accede al sistema se auto redireccionará al menú principal del aplicativo. <input type="checkbox"/> El usuario y contraseña serán almacenados directamente en la base de datos. 	100%
HU03	<ul style="list-style-type: none"> <input type="checkbox"/> El menú debe ser fácilmente accesible <input type="checkbox"/> Las opciones del menú deben ser claras y explicativas. <input type="checkbox"/> El menú debe ser intuitivo y fácil de usar. <input type="checkbox"/> El menú aparecerá después de iniciar sesión. <input type="checkbox"/> El menú tendrá las siguientes opciones: <ul style="list-style-type: none"> ✧ Gestión empresa. ✧ Gestión servicios. ✧ Gestión horarios. 	100%

En la tabla 2.5 se observa el cumplimiento por tareas del sprint 1

Tabla 2.5: Cumplimiento del sprint 1. Elaborado por autor.

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo (No/Parcial)
HU01	Maquetación login	Si	N/A
	Maquetación gestión empresarial	Si	N/A
	Maquetación gestión servicios	Si	N/A
	Maquetación gestión horarios por servicio	Si	N/A
	Animación de interfaces	Si	N/A

HU02	Crear servicio REST de "login"	Si	N/A
	Crear tabla "login" en base de datos	Si	N/A
	Crear ruta "login" en frontend	Si	N/A
	Crear formulario para "login" en frontend	Si	N/A
	Implementar función "autenticar"	Si	N/A
HU03	Creación de opciones para menú	Si	N/A
	Creación de ruta menú	Si	N/A
	Redireccionamiento de opción Gestión empresa	Si	N/A
	Redireccionamiento de opción Gestión servicios	Si	N/A
	Redireccionamiento de opción Gestión horarios	Si	N/A

Sprint retrospective

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué salió bien en la iteración?

- Se cumplieron todos los objetivos principales del sprint
- Se creó un prototipo con animación de media fidelidad del aplicativo completo para definir todos los flujos.
- Se creó la sección para autenticar al usuario con la redirección al inicio del sistema.
- Se creó el menú con las opciones solicitadas por el cliente.

¿Qué se puede mejorar?

Se podría agregar más historias de usuario para el sprint 1, ya que aunque la gráfica muestra que se han cumplido todas las historias, es posible que haya una situación no deseable que deba ser considerada para el siguiente sprint, ya que puede que no se esté asumiendo suficiente trabajo.

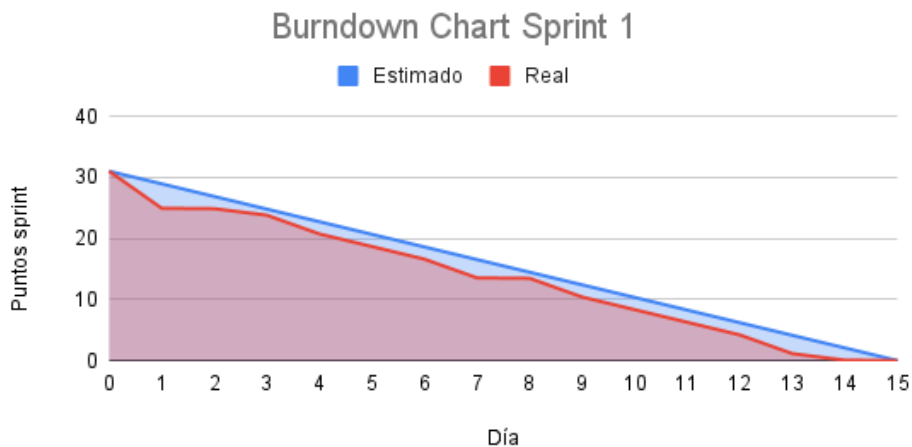


Figura 2.14: Burndown chart sprint 1. Elaborado por autor.

2.4.3 Sprint 2

Sprint planning

Objetivos del sprint

- Crear el módulo de gestión de empresa en el proyecto de administración tanto para backend como frontend.
- Crear el módulo de gestión de servicios en el proyecto de administración tanto para backend como frontend.
- Crear el módulo de gestión de horarios para el proyecto de administración tanto para backend como frontend.

Sprint backlog

Para el backlog del sprint 2 se estima trabajar las historias de usuario 04,05,06 cuyas tareas se encuentran listadas en la tabla 2.6

Tabla 2.6: Sprint 2 backlog. Elaborado por autor.

Código	Historia de usuario	Tareas de sprint
--------	---------------------	------------------

HU04	<p>Como usuario administrador, deseo poder crear y editar la información relacionada a la empresa que se mostrará en el chatbot.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Crear servicio REST “empresa” en backend. <input type="checkbox"/> Crear tabla de “empresa” a nivel de base de datos. <input type="checkbox"/> Crear ruta en frontend <input type="checkbox"/> Crear formulario de edición de “empresa”
HU05	<p>Como usuario administrador, quiero poder gestionar los servicios de mi empresa de manera eficiente y sencilla.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Crear servicio REST “servicio por empresa” en backend. <input type="checkbox"/> Crear tabla de “servicios por empresa” a nivel de base de datos. <input type="checkbox"/> Crear ruta en frontend <input type="checkbox"/> Crear formulario de creación y edición de “servicios por empresa” <input type="checkbox"/> Implementar campos de búsqueda
HU06	<p>Como usuario administrador, quiero ser capaz de gestionar los horarios disponibles para cada servicio de mi empresa para poder ofrecer horarios disponibles a los clientes.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Crear servicio REST “horarios por servicio”. <input type="checkbox"/> Crear tabla para “horario por servicio” <input type="checkbox"/> Crear ruta para “horario por servicio” <input type="checkbox"/> Crear formulario de creación y edición de “horarios por servicios” en frontend <input type="checkbox"/> Implementar campos de búsqueda

Ejecución del sprint

1. Se creó el módulo “empresa” tanto para frontend como en backend. En la figura 2.15 se muestra el módulo “empresa” en frontend.

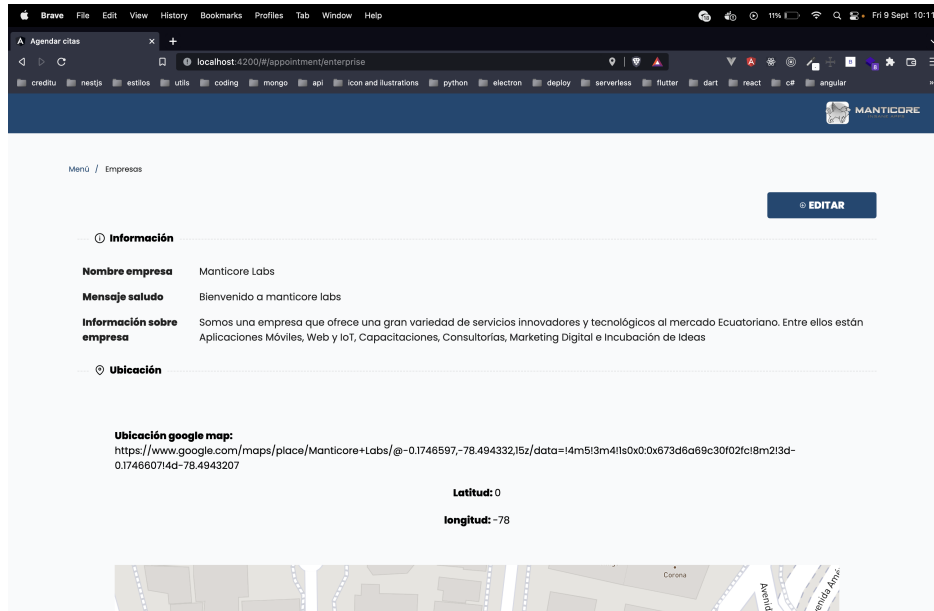


Figura 2.15: Modulo de empresa

2. Se creó el módulo “servicios por empresa” tanto para frontend como en backend. En la figura 2.16 se muestra el módulo “servicio por empresa” en frontend.

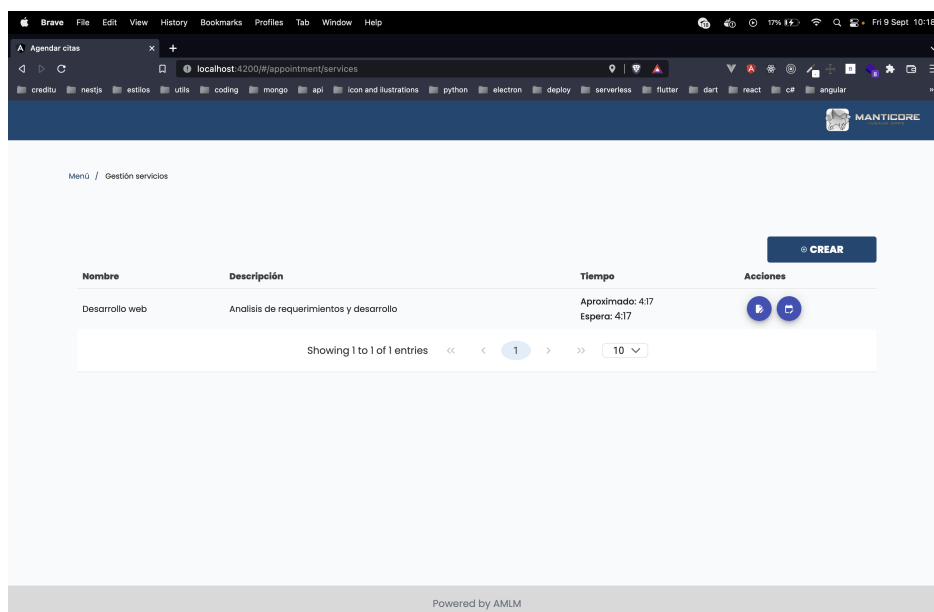


Figura 2.16: Modulo de servicios

- a) Se creó el módulo “servicios por empresa” en backend.
 - b) Se creó la base de datos del módulo “servicio por empresa” en el motor de base de datos.
 - c) Se crean los campos para “servicios por empresa”
 - d) Se crearon los campos de formulario para el módulo “servicio por empresa”
3. Se creó el módulo “horarios por servicio” tanto en el proyecto de backend como frontend. En la figura 2.17 se muestra el módulo “horarios” en frontend.

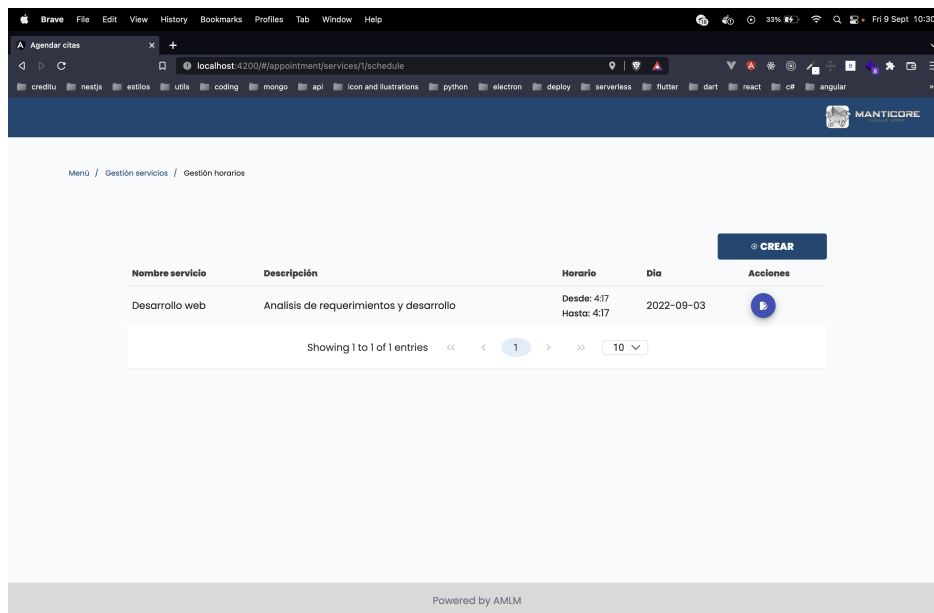


Figura 2.17: Horario por servicio

- a) Se creó el módulo “horarios” de un “servicio” en backend.
- b) Se creó la base de datos del módulo “horario” en el motor de base de datos.
- c) Se crearon los dtos, entity y validaciones para crear horarios.
- d) Se creó la opción para acceder a horarios por servicio.
- e) Se creó el formulario horarios por servicio.

Sprint review

Al final del sprint 2, se lleva a cabo el sprint review para evaluar el progreso en cuanto al cumplimiento de las historias de usuario desarrolladas. La evaluación incluye la revisión de la tabla de resultados del sprint review para asegurarse de que todas las historias están

completas y cumplen con los criterios de aceptación establecidos. Además, se discuten las lecciones aprendidas y se planifican las acciones necesarias para mejorar en el próximo sprint, el resultado del sprint review se encuentra en la tabla 2.7

Tabla 2.7: Criterio de aceptación del sprint 2. Elaborado por autor.

Historia	Criterio de aceptación	Cumplimiento
HU04	<ul style="list-style-type: none"> <input type="checkbox"/> El sistema permitirá crear, editar la información de la empresa <input type="checkbox"/> El sistema permitirá cargar el saludo del bot, <input type="checkbox"/> El sistema permitirá cargar la ubicación geográfica de la empresa. <input type="checkbox"/> Los campos que se mostrarán en el formulario son: <ul style="list-style-type: none"> ✧ Nombre empresa (Puede ingresar letras y números) (Editable) (Obligatorio) ✧ Mensaje saludo (Puede ingresar letras y números) (Editable) (Obligatorio) ✧ Información sobre empresa (Puede ingresar letras y números) (Editable) (Obligatorio) ✧ Ubicación google maps: (Puede ingresar letras y números) (Editable) (Obligatorio) ✧ Latitud(Puede ingresar números) (Editable) (Obligatorio) ✧ Longitud (Puede ingresar números) (Editable) (Obligatorio) 	100 %

HU05	<ul style="list-style-type: none"> <input type="checkbox"/> El sistema permitirá crear, listar, editar un “servicio” de un “empresa” <input type="checkbox"/> Se permite crear varios “servicios” para una “empresa” previamente generada. <input type="checkbox"/> Los servicios se pueden buscar por “nombre de servicio” <input type="checkbox"/> Tendrá la opción “horarios por servicio” que redirigirá a esta opción. <input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> ✧ Nombre del servicio (Puede ingresar letras y números) (Editable) (Obligatorio). ✧ URL (Puede ingresar letras y números) (Editable) (Obligatorio) 	100%
HU06	<ul style="list-style-type: none"> <input type="checkbox"/> El sistema permitirá crear listar un “horario” por servicio. <input type="checkbox"/> Solo se puede crear “horarios” para un “servicio” existente. <input type="checkbox"/> Los “horarios” se pueden buscar por día. <input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> ✧ Servicio (Puede seleccionar servicios ya creados) ✧ Día (Puede seleccionar un día) ✧ Desde (Puede seleccionar una hora en hh:mm) ✧ Hasta (Puede seleccionar una hora en hh:mm) 	100%

En la tabla 2.8 se puede ver el cumplimiento del sprint 2 por tareas.

Tabla 2.8: Cumplimiento del sprint 2. Elaborado por autor.

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo (No/Parcial)
--------	-------	------------------------------	---------------------

HU04	Crear servicio REST de “empresa”	Si	N/A
	Crear tabla de “empresa” a nivel de base de datos	Si	N/A
	Crear ruta en frontend	Si	N/A
	Crear formulario en frontend de edición de “empresa”	Si	N/A
HU05	Crear servicio REST de “servicios por empresa”	Si	N/A
	Crear tabla de “servicios por empresa” en la base de datos	Si	N/A
	Crear dto de “servicios por empresa”	Si	N/A
	Crear ruta “servicios por empresa”	Si	N/A
	Crear formulario en frontend de creación y edición de “servicios por empresa”	Si	N/A
	Funcionalidad para realizar búsquedas por campo.	Si	N/A
HU05	Crear servicio REST de “horarios por servicio”	Si	N/A
	Crear tabla de “horarios por servicio” en la base de datos	Si	N/A
	Crear dto de “horarios por servicio”	Si	N/A
	Crear ruta “horarios por servicio”	Si	N/A
	Crear formulario en frontend de creación y edición de “horarios por servicio”	Si	N/A
	Funcionalidad para realizar búsquedas por campo “día”	Si	N/A

Sprint retrospective

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué salió bien en la iteración?

- Se cumplió con todos los objetivos del sprint

- Se creó el formulario de editar para el módulo “empresa”

- Se creó el CRUD del módulo “servicios por empresa”

- Se creó el CRUD del módulo “horarios por servicio”

- Se realizó la navegación del sistema.

¿Qué se puede mejorar?

La solución de “horarios por servicio” puede no ser la más adecuada, ya que depende del usuario administrador los horarios disponibles por servicios.

En la Figura 2.18 se puede visualizar mediante un burndown el esfuerzo realizado durante los días del sprint, la gráfica indica que se completaron los puntos planificados para el presente sprint. Se completaron alrededor de 2 o 3 puntos diarios permitiendo completar el sprint a tiempo.

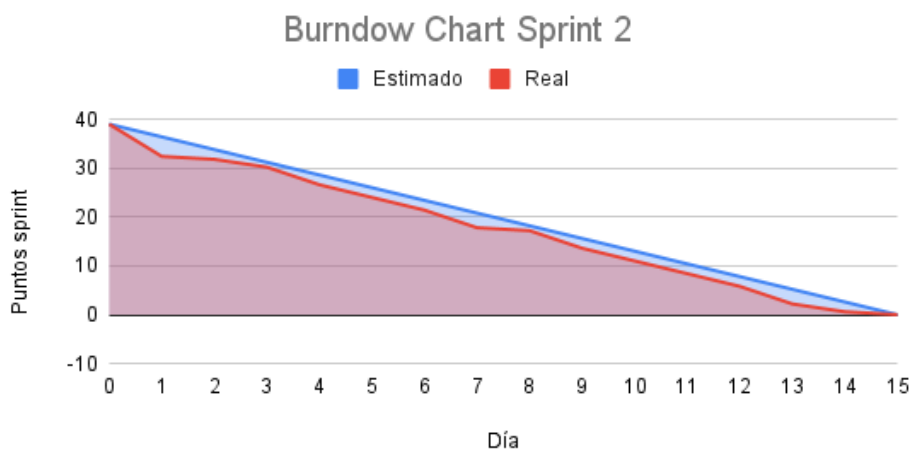


Figura 2.18: Burndown chart sprint 2. Elaborado por autor.

2.4.4 Sprint 3

Sprint planning

Objetivos del sprint

- Creación y configuración de Dialogflow para que pueda recibir información de la aplicación web.
- Crear intents de inicio y activación del chatbot para la activación del bot.
- Crear intents de default para cuando el bot pierda el hilo de la conversación.

Sprint backlog

En esta ocasión el backlog del sprint 3 se seleccionó las historias 07,08,09 cuyas tareas se listan en la tabla 2.9

Tabla 2.9: Sprint 3 backlog. Elaborado por autor.

Código	Historia de usuario	Tareas de sprint
HU07	Como usuario administrador, quiero que mi chatbot utilice la plataforma Dialogflow para mejorar su funcionamiento y proporcionar una experiencia de usuario más fluida.	<ul style="list-style-type: none"><input type="checkbox"/> Crear cuenta en google Dialogflow.<input type="checkbox"/> Crear agente.<input type="checkbox"/> Configuración de idiomas.

HU08	Como usuario administrador, quiero que el chatbot pueda obtener información de la aplicación web para ofrecer una experiencia más completa y personalizada a los usuarios finales.	<ul style="list-style-type: none"> <input type="checkbox"/> Creación de una URL pública temporal para poder ser accedida por Dialogflow. <input type="checkbox"/> Configuración de cores para que permita hacer peticiones desde el Dialogflow. <input type="checkbox"/> Configuración de Dialogflow con webhook.
HU09	Como usuario administrador deseo configurar el intent de default para cuando el chatbot no encuentra coincidencias en el flujo de conversación.	<ul style="list-style-type: none"> <input type="checkbox"/> Creación de intent de activación. <input type="checkbox"/> Configuración de intents de default.

Ejecución del sprint

1. Se creó la cuenta de Google Dialogflow para el bot y se creo un agente.

Para crear un nuevo bot en Dialogflow, se debe ingresar su nombre único y seleccionar el idioma principal. Este será también parte del nombre del proyecto en la nube de Google. Después de crear el agente, se puede configurar entidades que reconocerá, agregar sinónimos a estas entidades y mapearlas en cada intent para interactuar con ellas. La figura 2.19 se muestra la configuración de la entidad "Servicios".

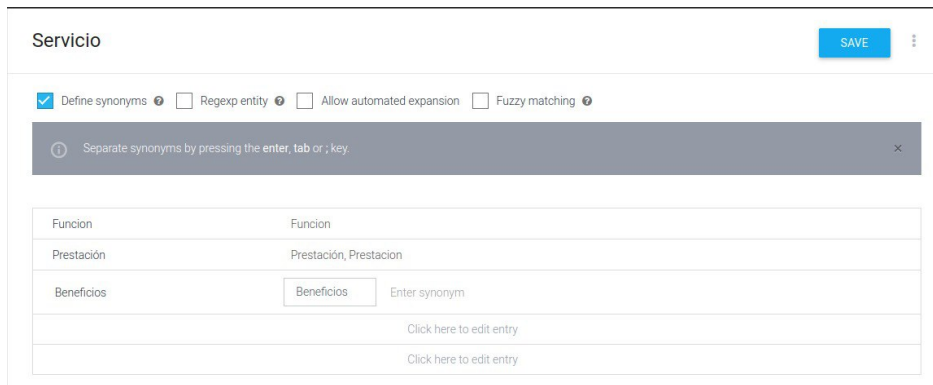


Figura 2.19: Entidades de dialoflow

Intents: Los intents deben ser configurados en base al flujo previsto. La característica principal de los intents es que Dialogflow puede detectar ciertas entidades para activar el intent utilizando las frases de entrenamiento. El intent se activa con una palabra clave, independientemente de si se encuentra al principio, en el medio o al final de la oración. En la sección de frases de entrenamiento, se deben agregar todas las posibles frases que pueden activar el intent. La 2.20 muestra las diferentes etapas de entrenamiento y cómo Dialogflow detecta las entidades dentro de estas frases, resaltándolas con diferentes colores dependiendo de si hay una o varias entidades.

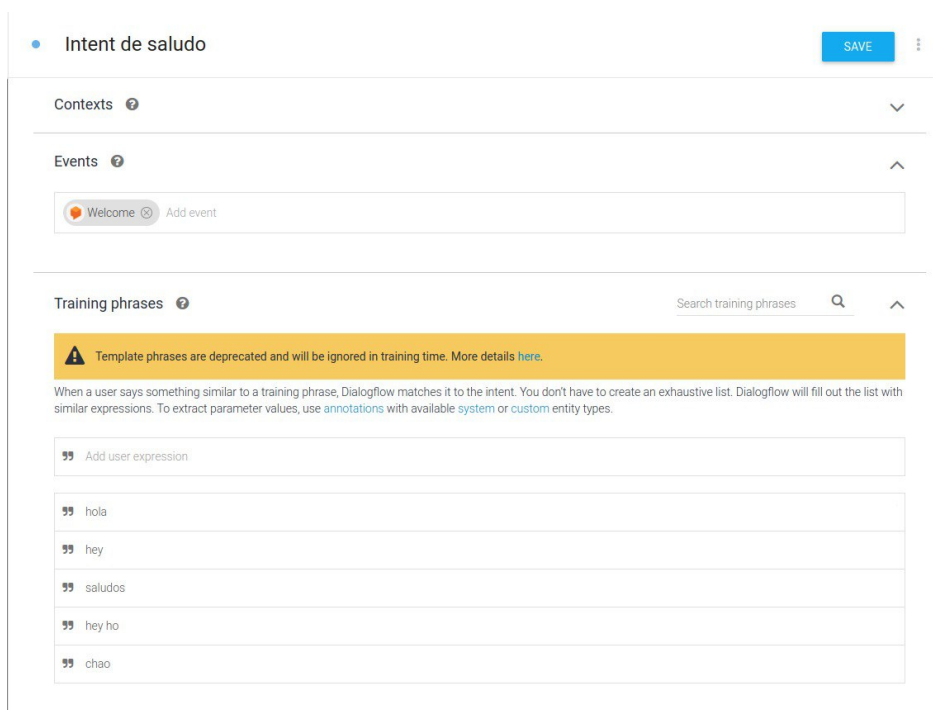


Figura 2.20: Intents de dialoflow

Texto de respuesta: Una vez activado el intent, accederá a la información agregada

en la sección de texto de respuesta para emitírsela al usuario final. Sin embargo, en este caso, utilizaremos la sección de Fulfillment para activar la llamada a nuestro webhook. En la figura 2.21 se puede observar la activación del webhook para este intent.

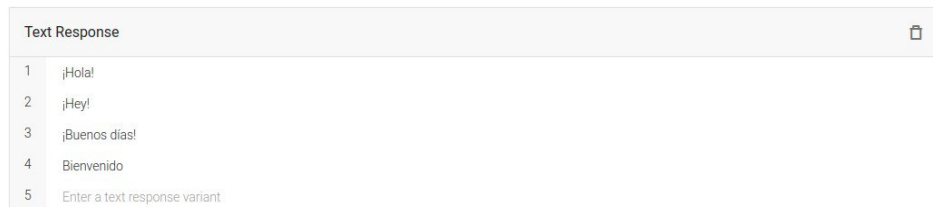


Figura 2.21: Texto de respuesta de dialogflow

Url webhook: Para poder acceder al webhook, se debe configurar una URL, el requisito principal es que esta utilice el protocolo HTTPS. Se puede utilizar el endpoint utilizando un método POST. En la figura 2.22 se puede observar la configuración de la URL.

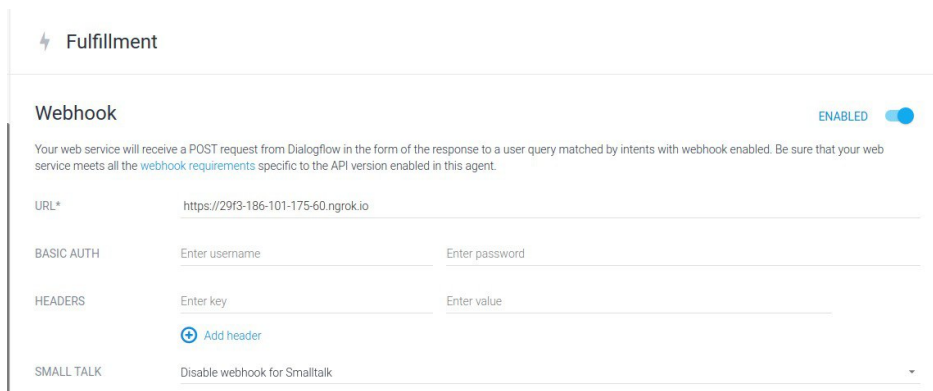


Figura 2.22: Url webhook.

2. Se configuró ngrok para exponer la aplicación a internet, apuntando a nuestro servicio web en la máquina local. En la figura 2.23 se muestra el servicio iniciado.

```

ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             alimonse (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding          http://422a-186-101-175-60.ngrok.io -> http://localhost:3000
                   https://422a-186-101-175-60.ngrok.io -> http://localhost:3000

Connections
ttl    opn    rt1    rt5    p50    p90
0      0      0.00  0.00  0.00  0.00

```

Figura 2.23: Servicio ngrok

3. Se creó el intent default en dialogflow como se muestra en la figura 2.24.

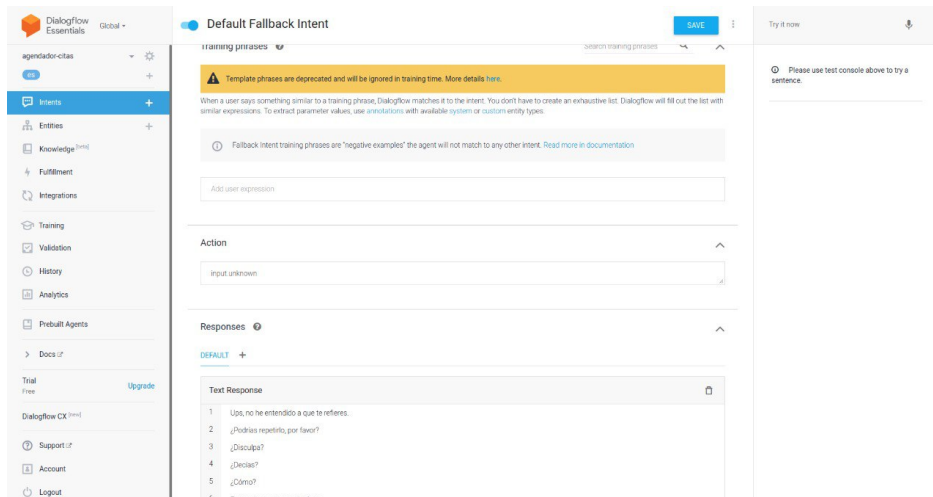


Figura 2.24: Intent dialogflow

Sprint review

Al culminar el sprint 3, se realiza el sprint review, en el cual se logra validar el flujo de cumplimiento con respecto a las historias de usuario desarrolladas, el resultado del sprint review se encuentra en la tabla 2.10

Tabla 2.10: Criterio de aceptación del sprint 3. Elaborado por autor.

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

HU07	<input type="checkbox"/> El usuario administrador podrá hacer uso de la cuenta de Dialogflow <input type="checkbox"/> Agente estará disponible en Dialogflow <input type="checkbox"/> Bot deberá estar configurado en idioma español	100 %
HU08	<input type="checkbox"/> El chatbot debe ser capaz de conectarse a la aplicación web y obtener información en tiempo real. <input type="checkbox"/> La información obtenida debe ser precisa y actualizada. <input type="checkbox"/> El usuario administrador debe tener la capacidad de controlar qué información se comparte con el chatbot.	100 %
HU09	<input type="checkbox"/> Creación de intent de default.	100 %

En la tabla 2.11 se puede ver el cumplimiento del sprint 3 por tareas.

Tabla 2.11: Cumplimiento del sprint 3. Elaborado por autor.

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo (No/Parcial)
HU07	Creación de cuenta Dialoflow	Si	N/A
	Creación de agente	Si	N/A
	Configuración de idioma y huso horario	Si	N/A
HU08	Conexión de entre aplicativo web y Dialgflow	Si	N/A
	Habilitación de cores entre backend, front-end y Dialogflow	Si	N/A
HU09	Creación de intent "Default"	Si	N/A
	Configuración de intent default.	Si	N/A

Sprint retrospective

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué salió bien en la iteración?

- Se cumplieron todos los objetivos principales del sprint.
- Se pudo configurar tanto el agente como los intents de welcome como de default.

¿Qué se puede mejorar?

Google está optimizando constantemente Dialogflow y ha habido varios cambios recientes, incluyendo la exigencia de una URL pública con certificado SSL. La documentación disponible puede estar desactualizada debido a la velocidad en la que la herramienta cambia. A pesar de haber tardado más tiempo del planificado en algunos días, se logró cumplir con el sprint 3 en su totalidad, según se muestra en la grafica burndown Sprint 3."

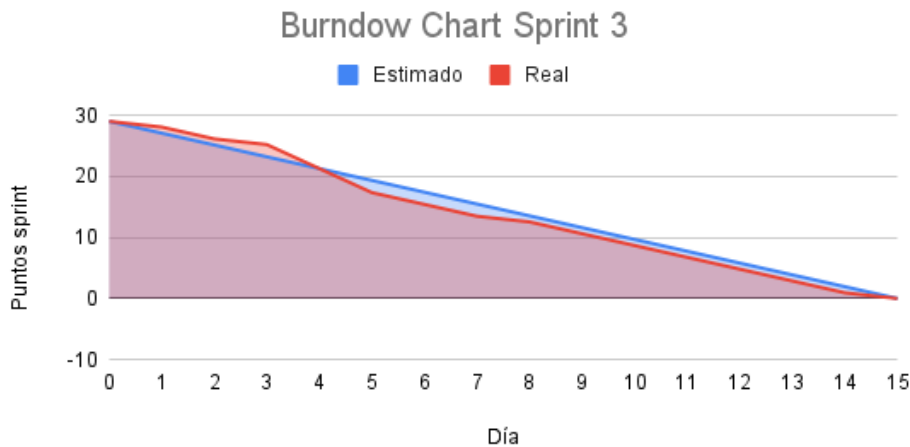


Figura 2.25: Burndown chart sprint 3. Elaborado por autor.

2.4.5 Sprint 4

Sprint planning

Objetivos del sprint

- Obtener la información necesaria desde el backend para mostrar el saludo en la plataforma Dialogflow.
- Obtener información sobre la empresa para ser mostrada en la plataforma Dialogflow.
- Configurar intents para realizar peticiones a backend.

Sprint backlog

En esta ocasión el backlog del sprint 4 se seleccionó las historias 10,11 cuyas tareas se listan en la tabla 2.12

Tabla 2.12: Sprint 4 backlog. Elaborado por autor.

Código	Historia de usuario	Tareas de sprint
HU10	Como usuario final, deseo ver un saludo personalizado en el chatbot, para poder ver la información que se ha configurado previamente en la aplicación web.	<ul style="list-style-type: none"><input type="checkbox"/> Crear intent “welcome” para iniciar conversación.<input type="checkbox"/> Realizar consulta para encontrar el servicio inicial.<input type="checkbox"/> Implementar servicio para encontrar saludo inicial.<input type="checkbox"/> Implementar controlador para encontrar saludo inicial.<input type="checkbox"/> Conectar intent a ser disparados desde chat con servicio del Backend.<input type="checkbox"/> Mostrar el saludo consumiendo información desde backend.

<p>HU11</p>	<p>Como usuario final, deseo que el chatbot muestre información sobre la empresa, la cual se ha configurado previamente en la aplicación web.</p>	<ul style="list-style-type: none"> ❑ Crear intent “informaciónEmpresa” para iniciar conversación. ❑ Realizar consulta para encontrar el servicio sobre información empresa. ❑ Implementar servicio para encontrar información sobre la empresa. ❑ Implementar controlador para acceder a servicio de información sobre la empresa. ❑ Conectar intent a ser disparados desde chat con servicio del Backend. ❑ Mostrar saludo consumiendo información de la empresa desde backend.
-------------	---	--

Ejecución del sprint

1. Se creó intent welcome en plataforma Dialogflow.

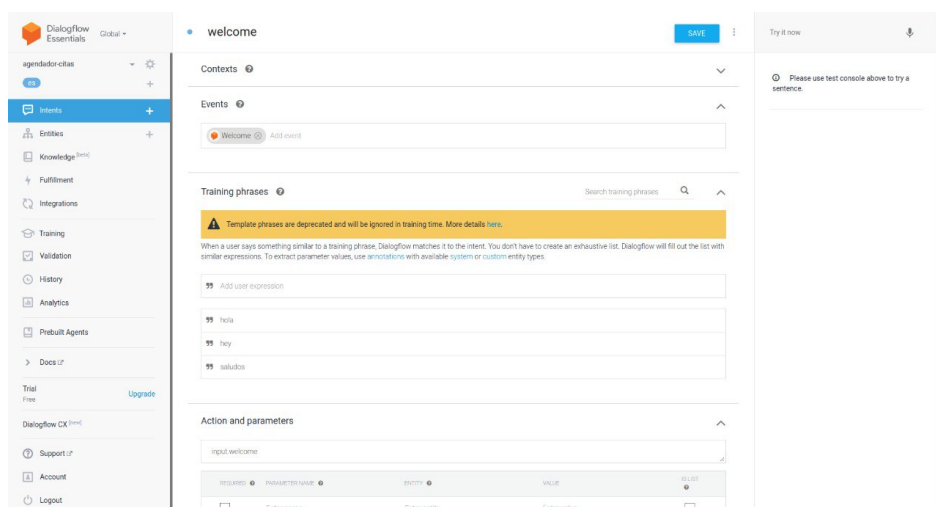


Figura 2.26: Intent welcome. Elaborado por autor.

2. Se realizó la consulta para mostrar un saludo inicial desde el aplicativo web y activar

el intent de saludo consumiendo información desde el backend, como se visualiza en la figura. 2.27 y 2.28

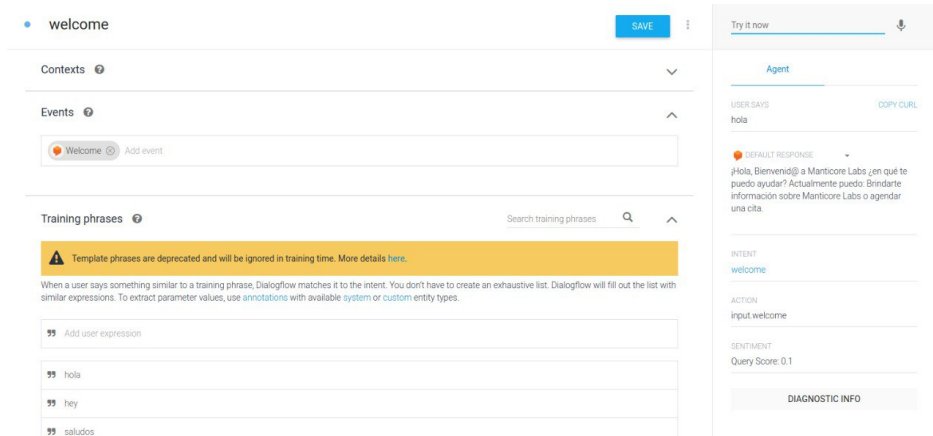


Figura 2.27: Saludo inicial. Elaborado por autor.

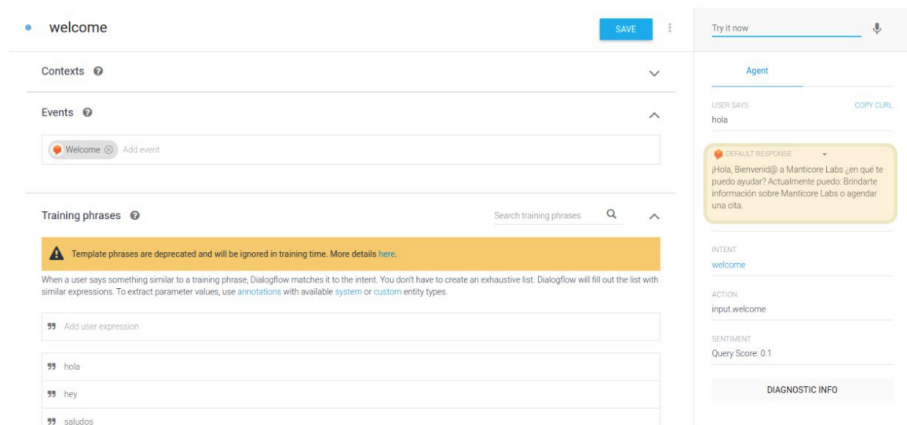


Figura 2.28: Activar intent. Elaborado por autor.

3. Se creó intent informacion-empresa el cual se activará con: "informacion", "sobre empresa", " a que se dedican".

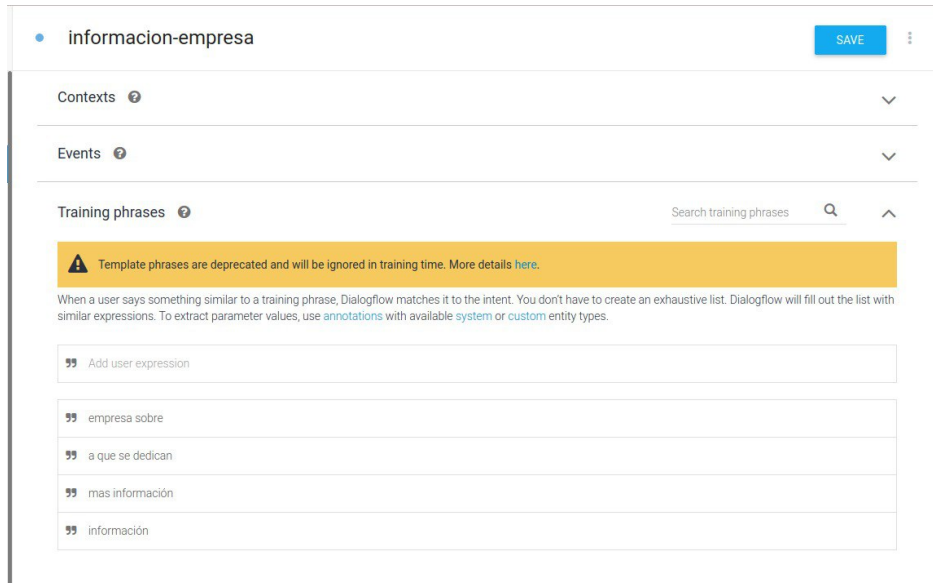


Figura 2.29: Intent información de la empresa. Elaborado por autor.

- Al activar el intent, muestra la información sobre la empresa que se ha configurado previamente en el aplicativo.

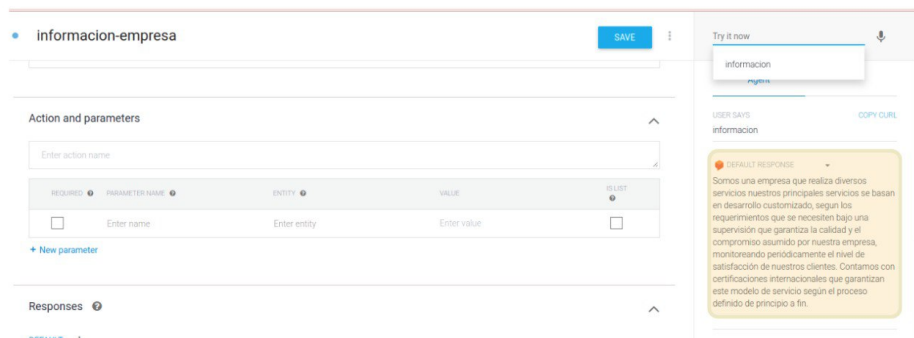


Figura 2.30: Activar intent de empresa. Elaborado por autor.

Sprint review

El sprint review se realizó al culminar el sprint 4, en donde el cumplimiento del trabajo con respecto a las historias de usuario se describe a continuación, el resultado del sprint se encuentra detallado en la tabla 2.13

Tabla 2.13: Criterio de aceptación del sprint 4. Elaborado por autor.

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

HU10	<ul style="list-style-type: none"> <input type="checkbox"/> El chatbot mostrará el saludo inicial configurado previamente. <input type="checkbox"/> El chatbot debe estar conectado a la aplicación web. <input type="checkbox"/> El usuario debe tener la capacidad de configurar un saludo personalizado en la aplicación web <input type="checkbox"/> El chatbot debe mostrar el saludo personalizado configurado por el usuario cada vez que el usuario final salude. 	100%
HU11	<ul style="list-style-type: none"> <input type="checkbox"/> La información configurada en la aplicación web debe ser accesible desde el chatbot. <input type="checkbox"/> La respuesta del chatbot debe ser proporcionada en un tiempo razonable. <input type="checkbox"/> La información mostrada en el chatbot debe ser coherente y actualizada con la información configurada en la aplicación web. 	100%

En la tabla 2.14 se puede ver el cumplimiento del sprint 4 por tareas.

Tabla 2.14: Cumplimiento del sprint 4. Elaborado por autor.

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo (No/Parcial)
HU10	Creación de entity para almacenar saludo.	Si	N/A
	Creación de dtos para validar creación	Si	N/A
	Creación de intent	Si	N/A
	Configurar frases de activación de intent	Si	N/A
	Creación de servicio para consultar información	Si	N/A
	Configurar fullfilment	Si	N/A
	Creación de controlador para conectar servicio con webhook	Si	N/A

HU11	Entity para almacenar información sobre la empresa.	Si	N/A
	Creación de dto de información empresa	Si	N/A
	Creación intent "información-empresa"	Si	N/A
	Agregar frases de activación de intent	Si	N/A
	Creación de servicio para obtener información de la base de datos	Si	N/A
	Conectar webhook con controlador para activación fullfilment	Si	N/A

Sprint retrospective

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué salió bien en la iteración?

Al empezar con la configuración del webhook desde este sprint facilitará la implementación de nuevos intents, logrando conectar con facilidad los intents desde la plataforma Dialogflow con el webhook

¿Qué se puede mejorar?

Como se muestra en la figura 2.31 se puede visualizar mediante el burndown que se han completado con normalidad los puntos de el sprint, se recomienda continuar con el flujo y la distribución de puntos para el resto de sprints.

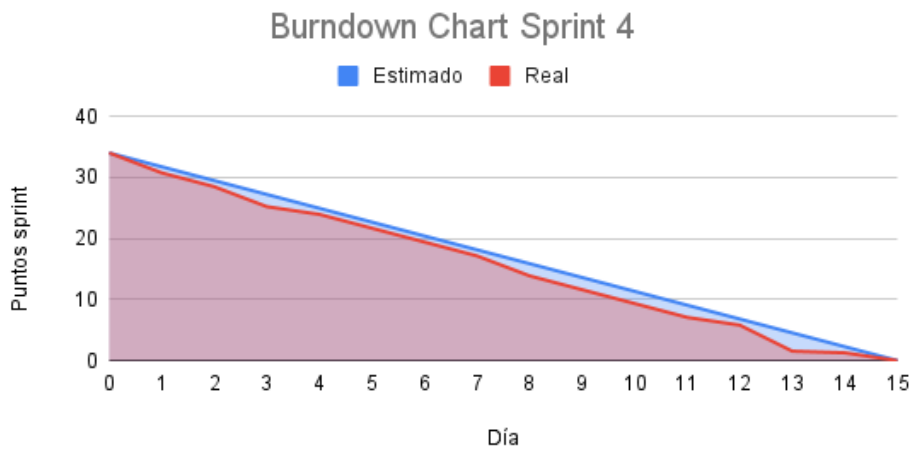


Figura 2.31: Burndown chart sprint 4. Elaborado por autor.

2.4.6 Sprint 5

Sprint planning

Objetivos del sprint

- Consumir la información almacenada en la base de datos para mostrar los servicios que han sido creados en el aplicativo.
- Mostrar las horas disponibles para agendamiento de citas, según servicio seleccionado.
- Agendar una cita en una hora y servicio seleccionado.

Sprint backlog

Para el backlog del sprint 5 se agrupó las historias de usuario 12,13,14,15 las cuales contienen las tareas que se describen en la tabla 2.15

Tabla 2.15: Sprint 5 backlog. Elaborado por autor.

Código	Historia de usuario	Tareas de sprint
--------	---------------------	------------------

<p>HU12</p>	<p>Como usuario final, quiero que el chatbot muestre la ubicación de la empresa configurada en la aplicación web.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Crear intent “ubicacion” para mostrar la ubicación de la empresa. <input type="checkbox"/> Realizar consulta obteniendo latitud, longitud de la empresa desde la base de datos <input type="checkbox"/> Crear servicio y controlador para encontrar ubicación de la empresa. <input type="checkbox"/> Conectar intent a webhook para ser activado cuando se requiera obtener la ubicación de la empresa. <input type="checkbox"/> Mostrar ubicación en Google Maps en plataforma Dialogflow.
<p>HU13</p>	<p>Como usuario final, quiero poder visualizar los servicios que se han configurado en la aplicación web a través del chatbot, para tener acceso a esta información de manera rápida y eficiente.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Crear intent “servicios” para mostrar servicios. <input type="checkbox"/> Realizar consulta obteniendo servicios de la base de datos <input type="checkbox"/> Crear servicio para encontrar servicios de la empresa. <input type="checkbox"/> Implementar controlador para encontrar servicios de la empresa. <input type="checkbox"/> Conectar intent a webhook para ser activado cuando se requiera. <input type="checkbox"/> Mostrar servicios en plataforma Dialogflow.

<p>HU14</p>	<p>Como usuario final, quiero ser capaz de agendar una cita en un servicio específico a través del chatbot, para tener una experiencia de usuario más eficiente y cómoda.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Configurar intent “agendar” para que obtenga un servicio seleccionado <input type="checkbox"/> Función para crear cita en bases de datos. <input type="checkbox"/> Función para agregar cita a google calendar. <input type="checkbox"/> Implementar servicio para crear citas en bases de datos. <input type="checkbox"/> Implementar controlador para generar citas en bases de datos como en Google Calendar. <input type="checkbox"/> Conectar intent de respuesta con plataforma Dialogflow <input type="checkbox"/> Mostrar respuesta de confirmación cita desde backend.
<p>HU15</p>	<p>Como usuario final deseo poder agendar una cita en un servicio seleccionado a una hora y fecha específica.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Generar respuesta a intent con las horas disponibles. <input type="checkbox"/> Mostrar las horas configuradas en el aplicativo. <input type="checkbox"/> Consultar horas para agendar cita. <input type="checkbox"/> Implementar servicio para encontrar horas disponibles. <input type="checkbox"/> Implementar controlador para acceder a servicio de obtención de horas. <input type="checkbox"/> Agregar webhook para mostrar horas disponibles.

Ejecución del sprint

1. Se creó intent “ubicación” en plataforma Dialogflow, para ser activado con diferentes palabras de entrenamiento, entre las principales son: ubicación, donde se encuentran, tienen oficinas. Como respuesta el agente consulta a través del webhook, realizando la petición al servidor para obtener la respuesta, finalmente esta se muestra como en la figura 2.32 la ubicación de la empresa

The screenshot displays the Dialogflow console for an intent named "ubicacion".

Left Pane (Configuration):

- Contexts:** None listed.
- Events:** None listed.
- Training phrases:** A list of phrases including "ubicación", "tienen oficinas", "ubicacion", and "donde queda?". A warning message states: "Template phrases are deprecated and will be ignored in training time. More details here."
- Action and parameters:** A table for defining parameters for the action.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

Right Pane (Agent Simulation):

- User Says:** "ubicacion"
- Default Response:** "Ubicación empresa Gregorio Bobadilla y razones undas" with a link to a Google Maps location.
- Intent:** "ubicacion"
- Action:** "Not available"
- Sentiment:** "Query Score: 0.6"

Figura 2.32: Intent ubicación. Elaborado por autor.

2. Se hizo el intent “servicios” para consultar los servicios desde el aplicativo web, y se realizó la función para crear citas en Google calendar, adicional se agregó la conexión entre el intent de servicios con el intent de “agendar”, este se activará, cuando el usuario de clic en el botón “Agendar Cita” como se visualiza en la figura 2.33.

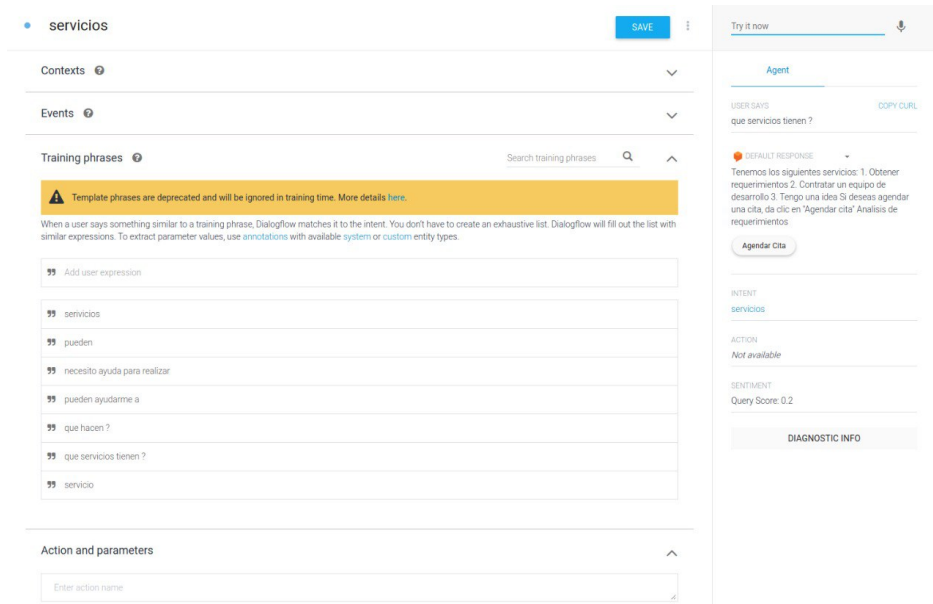


Figura 2.33: Intent servicio. Elaborado por autor.

3. Se realizó el intent “agendar” este intent es especial ya que según el contexto los siguientes mensajes que se envíen, el agente intentará obtener los parámetros servicio día y hora.

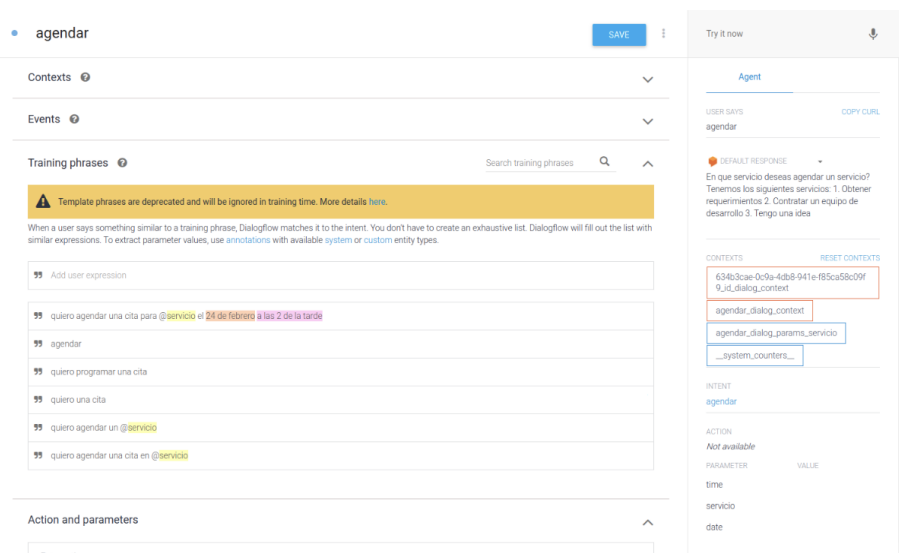



Figura 2.34: Configurar intent agendar. Elaborado por autor.

4. Se realizó la configuración de los parámetros que el intent “agendar” podrá recibir, como se muestra en la figura 2.35.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	servicio	@servicios	\$servicio	<input type="checkbox"/>	Define prompts...
<input checked="" type="checkbox"/>	time	@sys.time	\$time	<input type="checkbox"/>	Define prompts...
<input checked="" type="checkbox"/>	date	@sys.date	\$date	<input type="checkbox"/>	Define prompts...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	-

Figura 2.35: Parámetros de intent agendar. Elaborado por autor.

- Se configuro el intent para esperar el parámetro de día y hora, sin embargo, una vez que el usuario seleccione el servicio el intent “agendar” se mantendrá activo esperando los parámetros “día” y “hora”.


DEFAULT RESPONSE

¿ Que día deseas agendar el servicio seleccionado ?

CONTEXTS
RESET CONTEXTS

634b3cae-0c9a-4db8-941e-f85ca58c09f9_id_dialog_context

agendar_dialog_context

agendar_dialog_params_date

__system_counters__

INTENT

agendar

ACTION

Not available

PARAMETER	VALUE
time	
date	
servicio	cotizar

Figura 2.36: Parámetros de intent agendar. Elaborado por autor.

Sprint review

Al terminar el sprint 5, se evalúa según los criterios de aceptación y cumplimiento cuánto del sprint ha sido terminado y se realiza el respectivo feedback. En la tabla 2.16 se describe el resultado del sprint 5.

Tabla 2.16: Criterio de aceptación del sprint 5. Elaborado por autor.

Historia	Criterio de aceptación	Cumplimiento
HU12	<ul style="list-style-type: none"><input type="checkbox"/> El chatbot debe mostrar la dirección física de la empresa.<input type="checkbox"/> El chatbot debe mostrar un link para redireccionar a Google Maps la ubicación marcada.<input type="checkbox"/> La información de la ubicación de la empresa debe ser consultada en tiempo real desde la aplicación web.<input type="checkbox"/> La respuesta del chatbot debe ser fácil de entender y estar disponible para cualquier usuario que realice la pregunta.	100%
HU13	<ul style="list-style-type: none"><input type="checkbox"/> El chatbot debe ser capaz de recuperar la información de los servicios configurados en la aplicación web.<input type="checkbox"/> La información de los servicios debe ser mostrada en un formato claro y conciso para el usuario.<input type="checkbox"/> La respuesta del chatbot debe ser inmediata y precisa.<input type="checkbox"/> El chatbot debe validar que la información de los servicios esté disponible en la aplicación web antes de mostrarla al usuario.	100%

HU14	<ul style="list-style-type: none"> <input type="checkbox"/> El usuario debe ser capaz de seleccionar un servicio de una lista presentada por el chatbot. <input type="checkbox"/> El usuario debe ser capaz de elegir una fecha y hora disponible para la cita. <input type="checkbox"/> El chatbot debe confirmar la cita con el usuario. <input type="checkbox"/> La información sobre la cita agendada debe ser almacenada en la aplicación web. 	100 %
HU15	<ul style="list-style-type: none"> <input type="checkbox"/> El usuario debe ser capaz de seleccionar una fecha y hora específica para la cita. <input type="checkbox"/> La aplicación debe mostrar una confirmación de la cita <input type="checkbox"/> La información de la cita agendada debe ser registrada y visible para el usuario administrador en Google Calendar. 	100 %

En la tabla 2.17 se puede ver el cumplimiento del sprint 5 por tareas.

Tabla 2.17: Cumplimiento del sprint 5. Elaborado por autor.

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo (No/Parcial)
HU12	Creación de intent "ubicacion"	Si	N/A
	Crear una consulta para obtener información sobre latitud y longitud de la empresa.	Si	N/A
	Crear controlador para acceder a servicio y consultar información sobre ubicación	Si	N/A
	Crear servicio para realizar la coincidencia con el intent de Dialogflow con la función de mostrar ubicación empresa.	Si	N/A
	Creación de tarjeta para mostrar en una tarjeta la ubicación de la empresa.	Si	N/A
	Configurar webhook para mostrar la ubicación.	Si	N/A

HU13	Creación de intent "servicios"	Si	N/A
	Consulta para obtener servicios sobre la empresa.	Si	N/A
	Controlador para obtener el servicio y consultar servicios cargados previamente sobre la empresa.	Si	N/A
	Crear servicio para encontrar intent de Dialogflow con la función de listar servicios.	Si	N/A
	Creación de arreglo de servicios para mostrar en lista los servicios de la empresa.	Si	N/A
HU14	Creación de intent "agendar"	Si	N/A
	Creación para crear cita en la base de datos.	Si	N/A
	Crear cita en google calendar	Si	N/A
	Implementación de servicio para consumir servicios de la empresa	Si	N/A
	Creación de controlador para encontrar función de mostrar servicio.	Si	N/A
HU15	Configurar intent "agendar" para que reciba parámetro "dia"	Si	N/A
	Configurar intent "agendar" para seguir activo y obtener parámetro "hora"	Si	N/A
	Configurar intent para mantenerse activo hasta obtener parámetros.	Si	N/A
	Agregar cita, en servicio, día y hora seleccionado.	Si	N/A
	Agregar webhook para mostrar horas disponibles.	Si	N/A

Sprint retrospective

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué salió bien en la iteración?

El crear un solo intent “agendar” para obtener parámetros de día y hora, fue de gran utilidad ya que no se tuvo que crear diferentes intents. La plataforma de Dialogflow permite realizar esto utilizando un “FollowEvent” para esperar que se obtengan todos los parámetros.

¿Qué se puede mejorar?

Como se muestra en la figura 2.37 se puede visualizar mediante el burndown que se han completado con normalidad los puntos de el sprint 5, sin embargo, fueron 47 puntos hasta el momento el sprint con más puntos importantes por culminar además se tuvo varias historias de alta prioridad. Por lo tanto, se recomienda distribuir el peso de las HU en varios sprints para que el sprint no sea tan pesado para desarrollar.

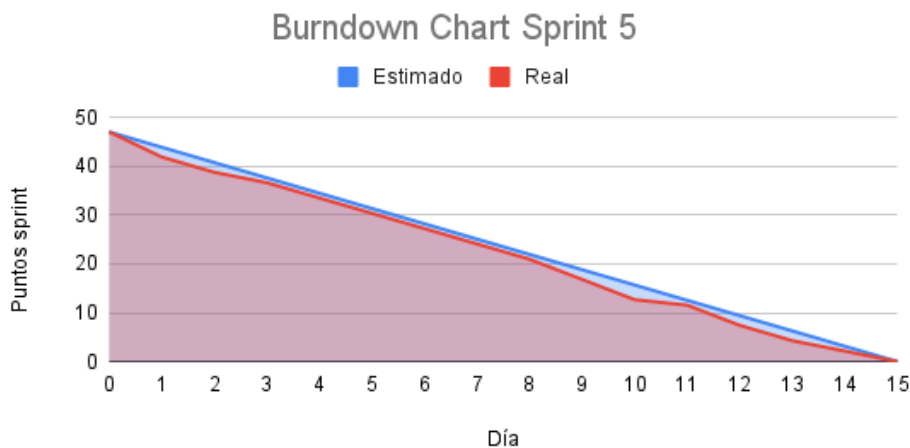


Figura 2.37: Burndown chart sprint 5. Elaborado por autor.

2.4.7 Sprint 6

Sprint planning

Objetivos del sprint

- Configurar intent para recopilar la información sobre la cita y esperar que el usuario acepte o rechace la misma.
- Validar fecha ingresada por usuario para agendar cita, hasta con 5 días de anticipación.

Sprint backlog

Las historias de usuario a realizarse en el sprint backlog 6 son 16 y 17. Las tareas se detallan en la tabla 2.18

Tabla 2.18: Sprint 6 backlog. Elaborado por autor.

Código	Historia de usuario	Tareas de sprint
HU16	Como usuario administrador quiero que el chatbot solo permita agendar citas hasta con 5 días de anticipación.	<ul style="list-style-type: none"><input type="checkbox"/> Crear validación para fecha ingresada por usuario no supere los 5 días permitidos<input type="checkbox"/> Configurar respuesta de chatbot recopilando información sobre la cita previamente solicitada<input type="checkbox"/> Configurar respuesta de chatbot para reajustar fecha y hora del usuario.<input type="checkbox"/> Crear función para modificar fecha y hora de cita previamente almacenada.<input type="checkbox"/> Configurar agente para activar nuevamente el intent "agendar" para mostrar prestación.
HU17	Como usuario final deseo aceptar o rechazar la cita que previamente había seleccionado.	<ul style="list-style-type: none"><input type="checkbox"/> Configurar intent "agendar" para validar parámetros de fecha y hora se hayan almacenado correctamente.<input type="checkbox"/> Configurar intent "agendar" para aceptar o rechazar la cita<input type="checkbox"/> Crear controlador para activar "agendar" para mostrar volver al flujo "agendar".

Ejecución del sprint

1. Creación de función que valida la fecha ingresada por usuario para agendar cita, no supere los 5 días permitidos previamente configurados.

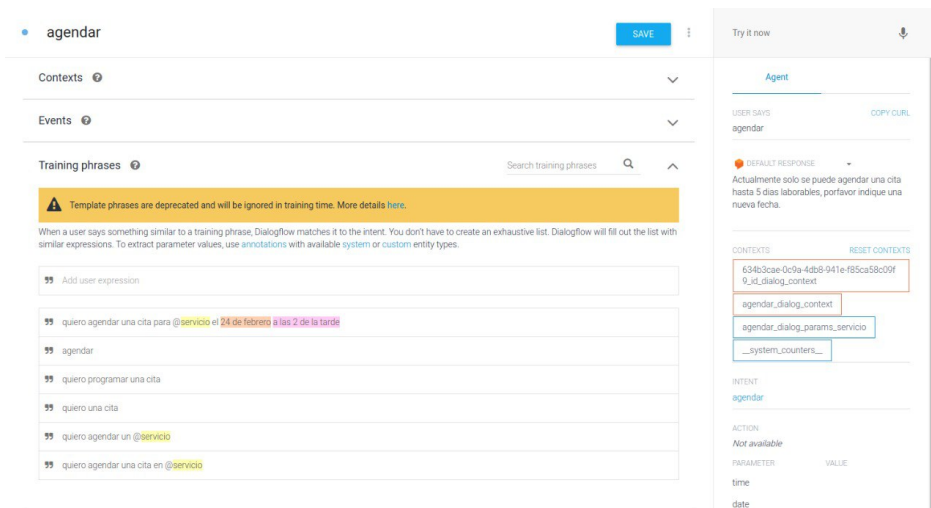


Figura 2.38: Agendar cita validación fecha. Elaborado por autor.

2. Como respuesta el agente receptorá toda información sobre la cita, y la presentará para la aceptación del usuario.

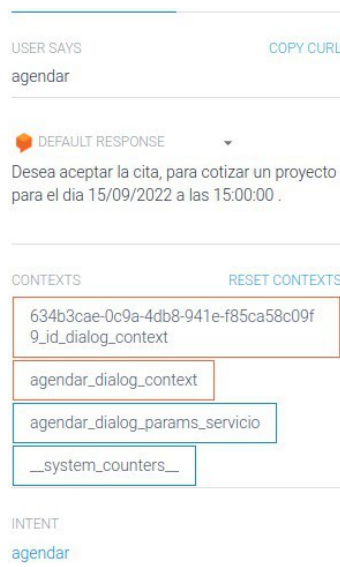


Figura 2.39: Respuesta cita. Elaborado por autor.

Sprint review

Al terminar el sprint 6, se realiza el sprint review, para visualizar el cumplimiento de las historias de usuario desarrolladas, el resultado del sprint review se encuentra en la tabla 2.19

Tabla 2.19: Criterio de aceptación del sprint 6. Elaborado por autor.

Historia	Criterio de aceptación	Cumplimiento
HU16	<ul style="list-style-type: none"><input type="checkbox"/> Validar que la fecha ingresada por el usuario no supere los 5 días permitidos.<input type="checkbox"/> Si supera los 5 días, el chatbot deberá, solicitar reajustar la fecha y hora al usuario.	100%
HU17	<ul style="list-style-type: none"><input type="checkbox"/> Se debe mostrar un resumen de los parámetros que el usuario ha seleccionado.<input type="checkbox"/> Se mostrará un mensaje "Cita agendada correctamente".	100%

En la tabla 2.20 se puede ver el cumplimiento del sprint 6 por tareas.

Tabla 2.20: Cumplimiento del sprint 6. Elaborado por autor.

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo (No/Parcial)
HU16	Función que valida que fecha ingresada, no supere cinco días previamente configurados.	Si	N/A
	Reorganizar intent para solicitar al usuario reajuste hora y fecha de cita.	Si	N/A
	Validar que la fecha agendada no se sobreponga con otras fechas.	Si	N/A
	Crear controlador para activar nuevamente intent "agendar".	Si	N/A

HU17	Función que valida que se hayan receptado todos los parámetros requeridos.	Si	N/A
	Receptar información de toda la cita, para ser mostrada en el prompt final.	Si	N/A
	Función para aceptar o rechazar cita.	Si	N/A

Sprint retrospective

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué salió bien en la iteración?

Las historias de usuario de este sprint 6 mayormente fueron validaciones y recolectar información durante todo el flujo, por lo cual se pudo terminar correctamente el sprint 6

¿Qué se puede mejorar?

En el burndown adjunto, se muestra el esfuerzo realizado durante los 15 días del sprint, en donde se verifica que los puntos planificados han sido culminados en su totalidad. Culminando alrededor de 2 o 3 puntos diarios.

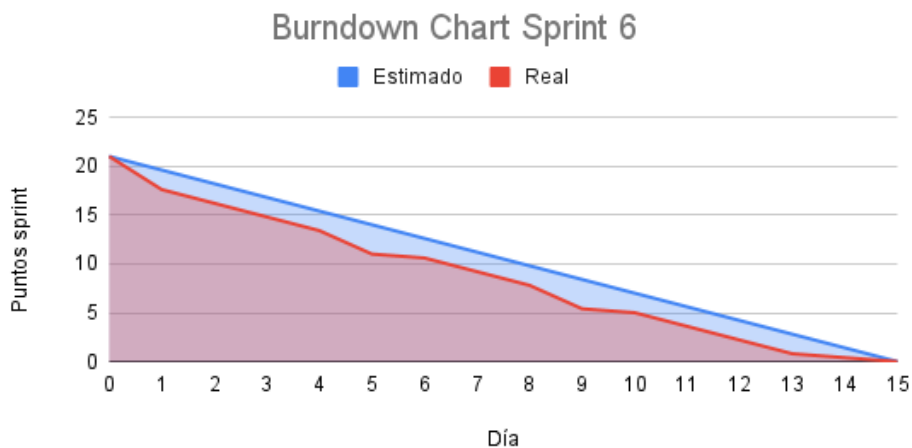


Figura 2.40: Burndown chart sprint 6. Elaborado por autor.

2.4.8 Sprint 7

Sprint planning

Objetivos del sprint

- Visualizar citas agendadas en calendario en google calendar
- Acceder al chatbot a través de la plataforma “Facebook messenger”
- Acceder al chatbot a través de la plataforma de mensajería “Whatsapp”

Sprint backlog

Para el backlog del sprint 7 se seleccionó las historias 18,19,20 cuyas tareas se listan en la tabla 2.21

Tabla 2.21: Sprint 7 backlog. Elaborado por autores.

Código	Historia de usuario	Tareas de sprint
HU18	Como usuario administrador deseo visualizar mis citas en google calendar	<ul style="list-style-type: none"><input type="checkbox"/> Configurar cuenta google calendar<input type="checkbox"/> Instalación de librerías google calendar<input type="checkbox"/> Creación de calendario “Agendamiento citas”<input type="checkbox"/> Conexión de API para creación de citas.<input type="checkbox"/> Función para crear eventos.

<p>HU19</p>	<p>Como usuario administrador deseo que el chatbot aparezca en Facebook Messenger.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Configurar página de facebook corporativa. <input type="checkbox"/> Conexión entre página de facebook con chatbot. <input type="checkbox"/> Habilitar permisos para realizar peticiones.
<p>HU20</p>	<p>Como usuario administrador deseo que el chatbot esté disponible en WhatsApp</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Configurar Twilio para realizar peticiones a través de WhatsApp y Dialogflow. <input type="checkbox"/> Conexión entre WhatsApp y Dialogflow. <input type="checkbox"/> El chatbot permitirá realizar peticiones a través de WhatsApp

Ejecución del sprint

1. Configuración de API google calendar a ser agendados, en donde se visualizan las citas agregadas al calendario, a través del agente.

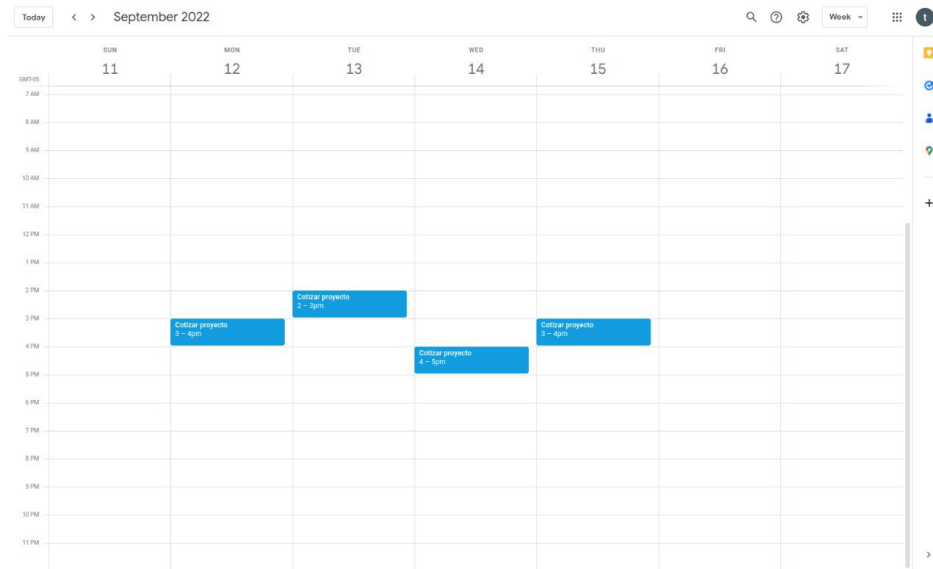


Figura 2.41: API google calendar. Elaborado por autor.

2. Configuración de página de Facebook para conectar con agente de Dialogflow.

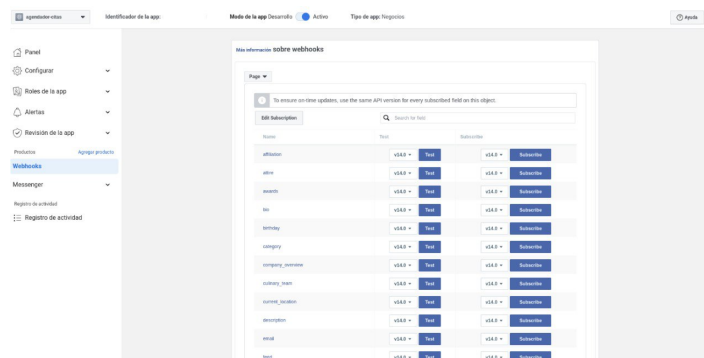


Figura 2.42: Configuración de Facebook Dialoflow. Elaborado por autor.

3. Acceder a través de la plataforma Facebook Messenger al chatbot, para agendar una cita.

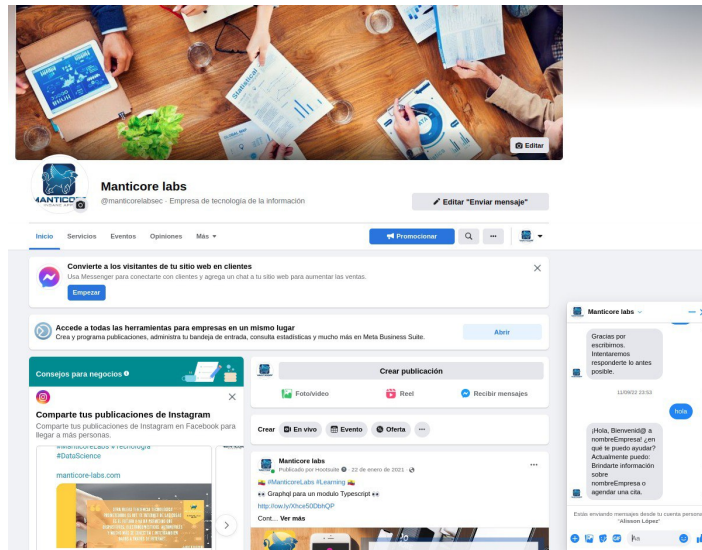


Figura 2.43: Messenger bot. Elaborado por autor.

4. Inicializar el chatbot a través de WhatsApp a través del chatbot, para agendar una cita.

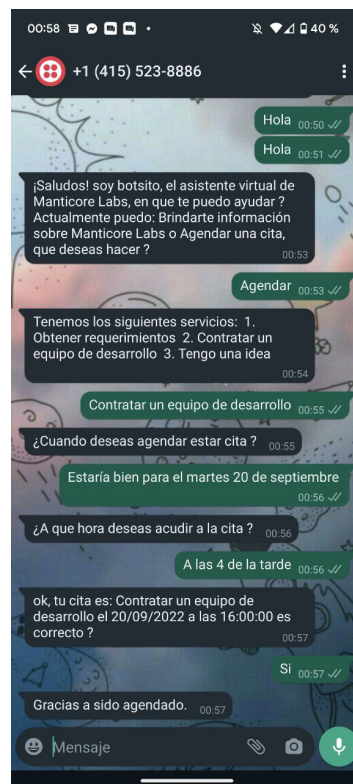


Figura 2.44: Chatbot Whatsapp. Elaborado por autor.

Sprint review

Al culminar el sprint 7, se realiza el sprint review, para verificar el cumplimiento de las historias de usuario 18, 19 y 20 para este sprint 7, el resultado del sprint review se encuentra en la Tabla 2.22

Tabla 2.22: Criterio de aceptación del sprint 7. Elaborado por autor.

Historia	Criterio de aceptación	Cumplimiento
HU18	<ul style="list-style-type: none"><input type="checkbox"/> Las citas deberán ser almacenadas en el calendario “Agendamiento citas”<input type="checkbox"/> Tanto las citas receptadas por Facebook Messenger y WhatsApp deben verse reflejadas en el calendario “Agendamiento citas”.	100%
HU19	<ul style="list-style-type: none"><input type="checkbox"/> El chatbot deberá estar disponible en la página oficial de Facebook Messenger de la empresa.<input type="checkbox"/> El chatbot deberá consumir la información previamente configurada en el aplicativo web.<input type="checkbox"/> La respuesta del chatbot debe ser proporcionada en un tiempo razonable.<input type="checkbox"/> La información sobre la cita agendada debe ser almacenada en la aplicación web	100%
HU20	<ul style="list-style-type: none"><input type="checkbox"/> El chatbot deberá estar disponible en WhatsApp de la empresa.<input type="checkbox"/> El chatbot deberá consumir la información previamente configurada en el aplicativo web.<input type="checkbox"/> La respuesta del chatbot debe ser proporcionada en un tiempo razonable.<input type="checkbox"/> La información sobre la cita agendada debe ser almacenada en la aplicación web	100%

En la tabla 2.23 se puede ver el cumplimiento del sprint 7 por tareas.

Tabla 2.23: Cumplimiento del sprint 7. Elaborado por autor.

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo (No/Parcial)
HU18	Configurar cuenta de Google Calendar.	Si	N/A
	Agregar librerías de Google Calendar	Si	N/A
	Crear calendario "Agendamiento de citas"	Si	N/A
	Crear función para crear eventos en "Google Calendar"	Si	N/A
HU19	Configurar página de Facebook	Si	N/A
	Configuración webhook Facebook con Dialogflow	Si	N/A
	Habilitar permisos para realizar peticiones	Si	N/A
	Configurar Dialogflow, para peticiones de Facebook Messenger.	Si	N/A
HU18	Configurar Twilio para obtener una cuenta con un número de teléfono temporal	Si	N/A
	Agregar API keys a google cloud de Dialogflow.	Si	N/A
	Conectar WhatsApp y Dialogflow.	Si	N/A

Sprint retrospective

¿Se alcanzaron los objetivos propuestos?

Si

¿Qué salió bien en la iteración?

Aunque la documentación ha cambiado y actualmente todo tiene un costo, se logró la integración con ambas plataformas y se verificó la creación de las citas en el calendario de Google Calendar

¿Qué se puede mejorar?

Se puede optar por crear un propio servidor para utilizar un número de teléfono local ya que el número proporcionado por Twilio es internacional.

En el Burndown adjunto se puede visualizar el esfuerzo realizado durante los días del sprint,

la figura 2.45 indica que se han completado los 39 puntos completamente.

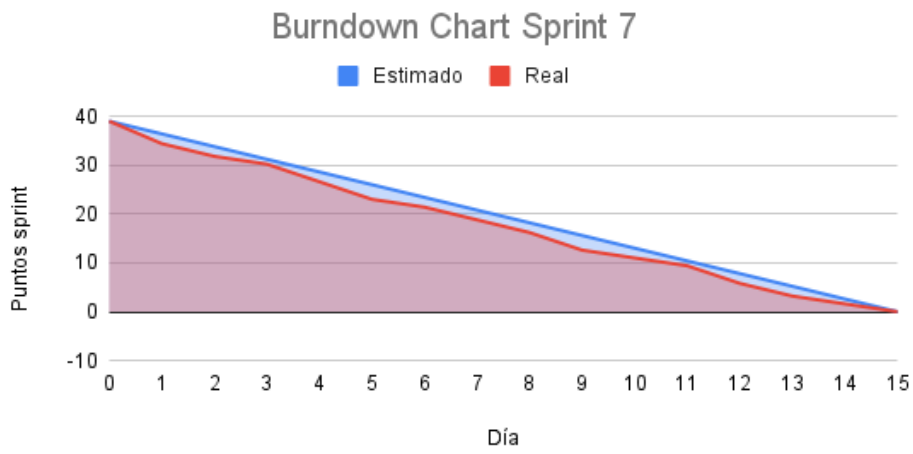


Figura 2.45: Burndown chart sprint 7. Elaborado por autor.

3 RESULTADOS Y DISCUSIÓN

3.1 DESCRIPCIÓN DEL SISTEMA

3.1.1 Descripción de los artefactos

3.1.1.1 Sistema de gestión de servicios

Pantalla login de ingreso al sistema: En esta pantalla el usuario podrá ingresar sus credenciales para gestionar la información de la empresa y sus servicios.



Figura 3.1: Pantalla de login

Pantalla opciones para el usuario: En esta pantalla el usuario podrá ver tres opciones las cuales puede modificar e ingresar datos.

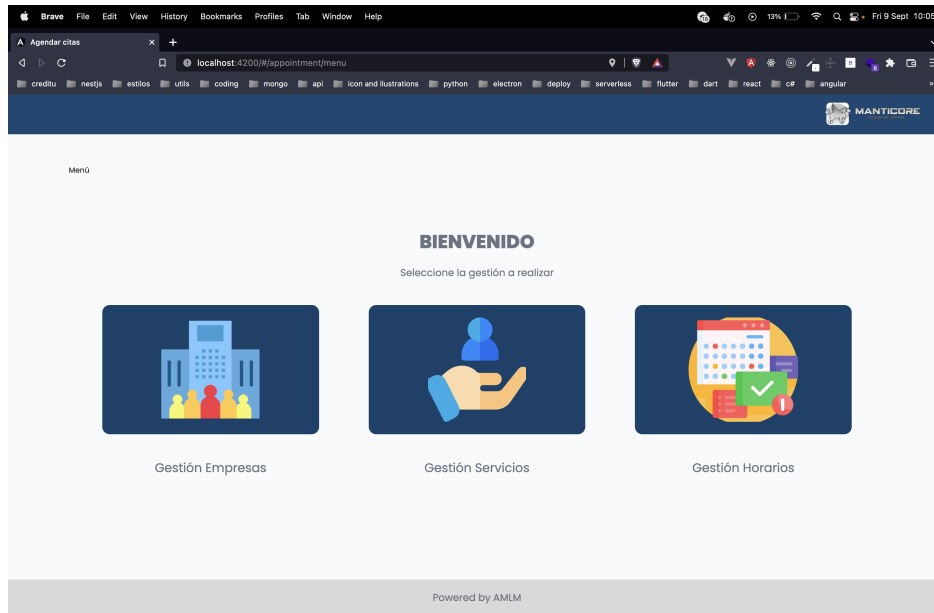


Figura 3.2: Pantalla menú de opciones

Pantalla información de la empresa: En esta pantalla el usuario podrá modificar la información de la empresa tales como: Nombre de la empresa, Saludo, Información, Latitud y Longitud (Ubicación empresa).

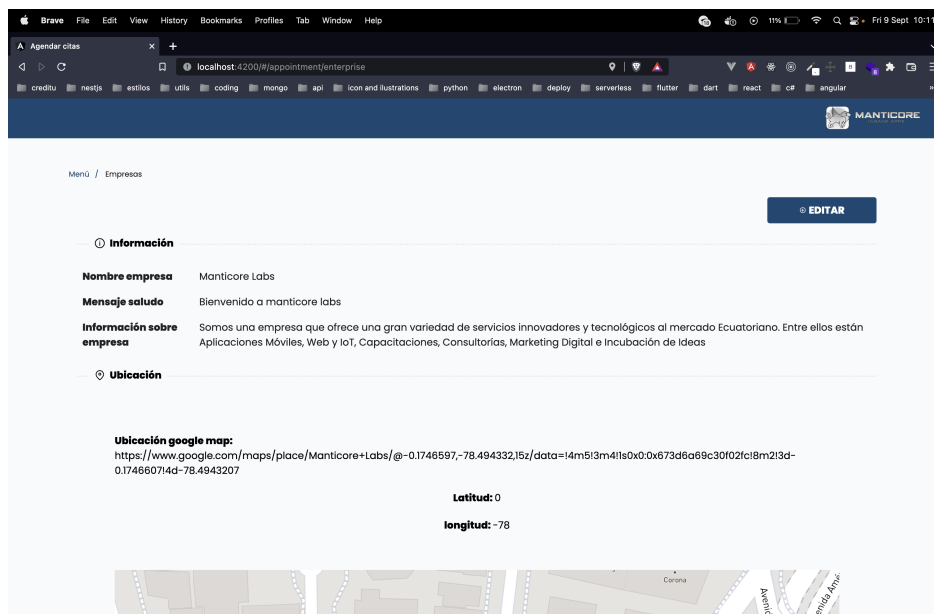


Figura 3.3: Pantalla información de la empresa

Pantalla gestión de servicio: En esta pantalla el usuario podrá crear o modificar la información del servicio.

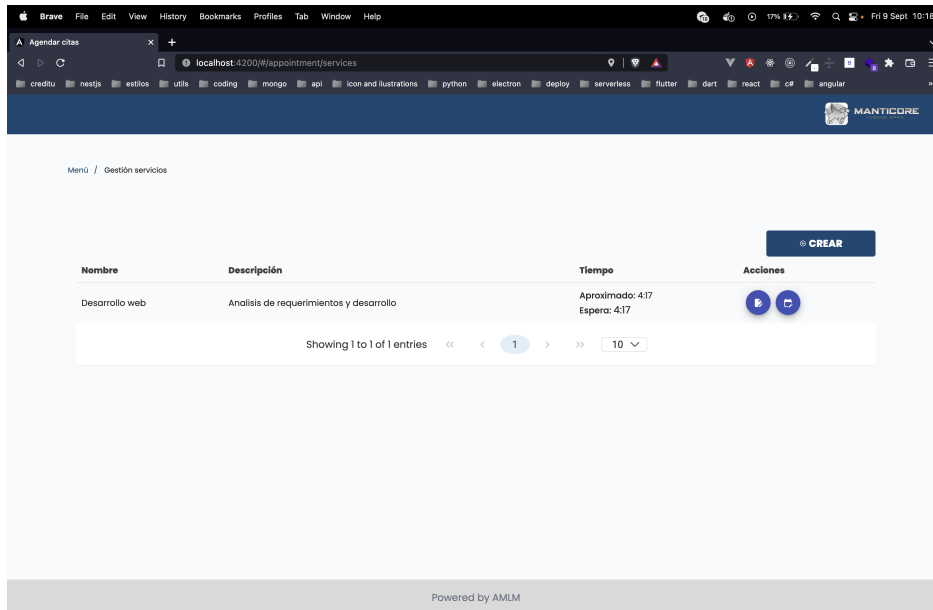


Figura 3.4: Pantalla gestión de servicios

Para crear un nuevo servicio se deberá dar clic en en crear y a continuación ingresar la información requerida para poder guardar

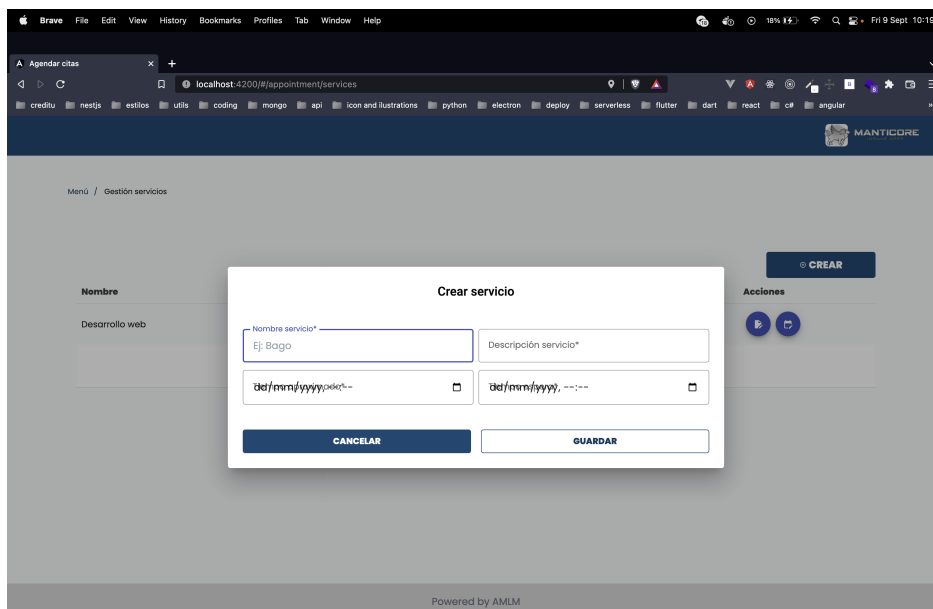


Figura 3.5: Pantalla creación de servicio

De igual forma para modificar solo se debe ingresar los datos que se desea modificar

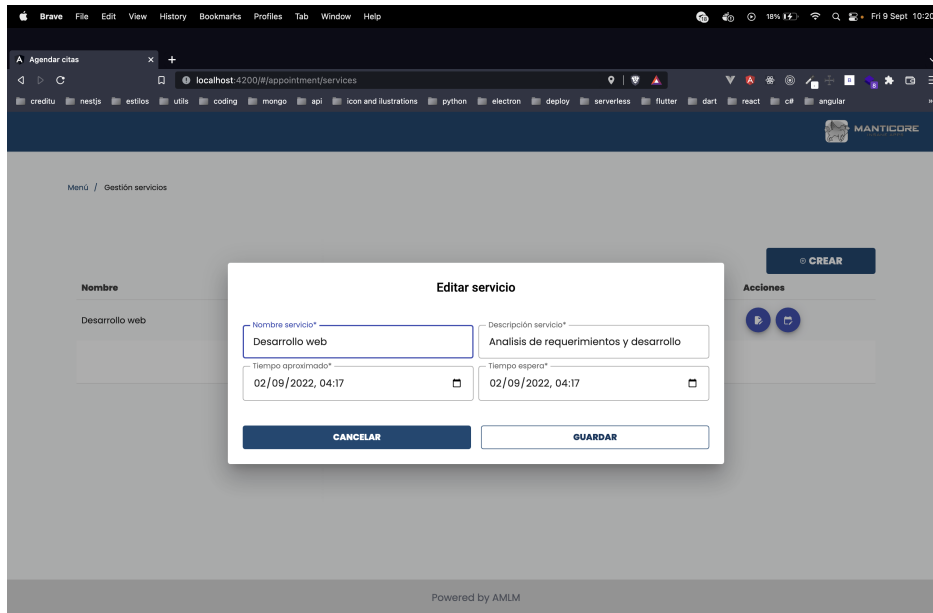


Figura 3.6: Pantalla actualización de servicio

Además, se puede ver los horarios del servicio que se crearon.

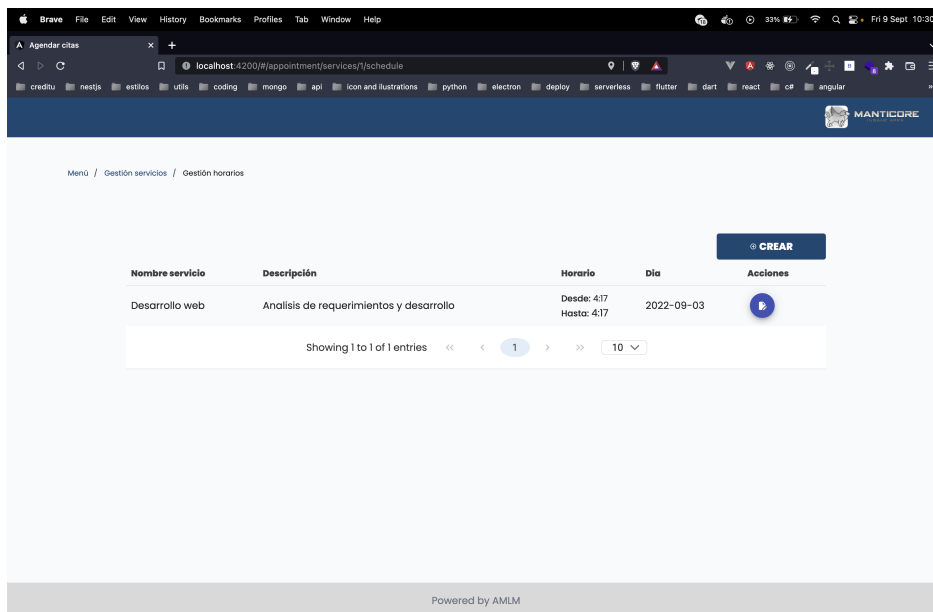


Figura 3.7: Pantalla horarios por servicio

Pantalla gestión de horarios: En esta pantalla el usuario podrá crear o modificar la información del horario.

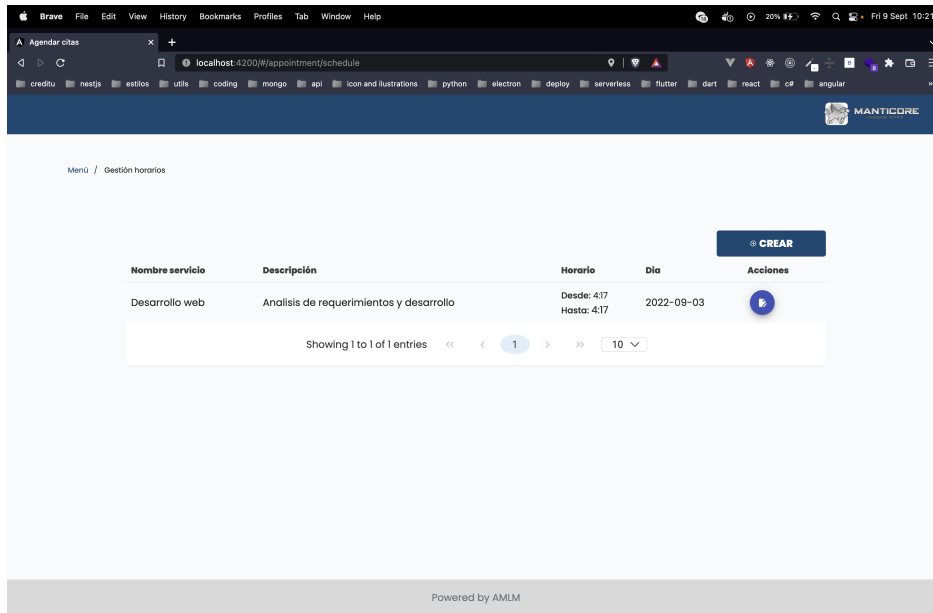


Figura 3.8: Pantalla gestión de horarios

para crear se deberá dar clic en en crear y a continuación ingresar la información requerida para poder guardar

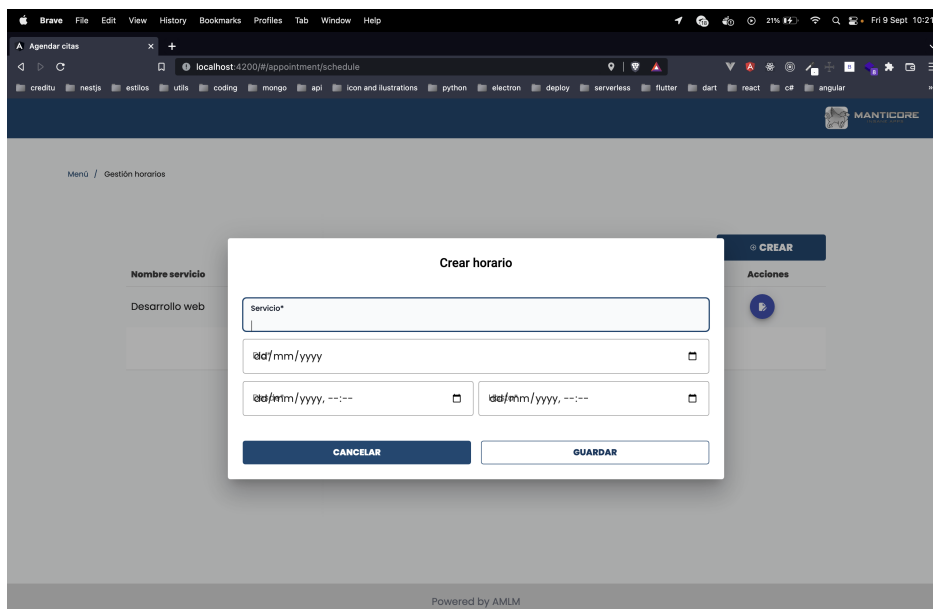


Figura 3.9: Pantalla creación de horario

de igual forma para modificar solo se debe ingresar los datos que se desea modificar.

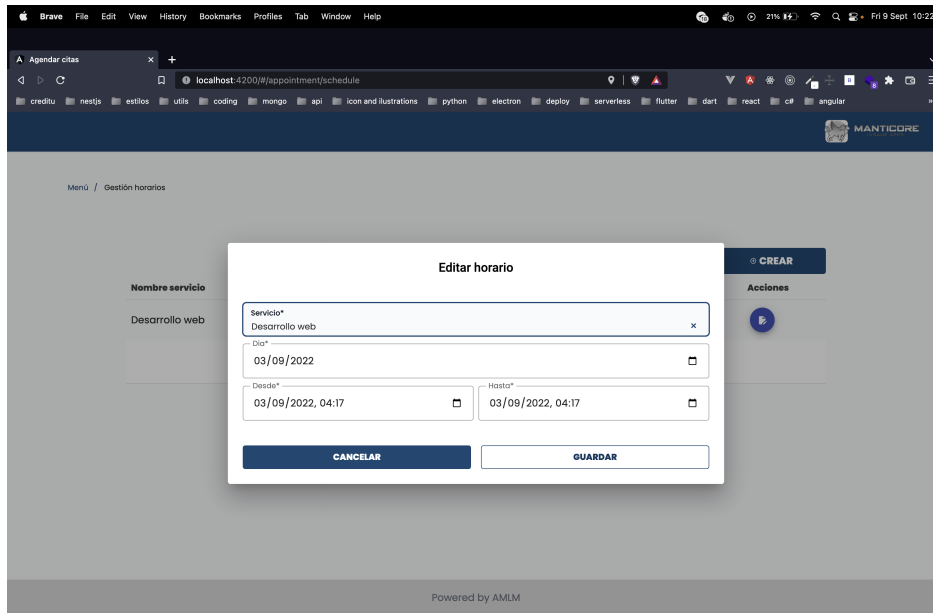


Figura 3.10: Pantalla actualización de horario

3.1.1.2 Agendador de citas

El agente se encuentra disponible en la plataforma Dialogflow. En esta pantalla el usuario puede visualizar los intents previamente configurados que hacen match con los controladores que a su vez realizan la petición a los diferentes servicios del servidor a través del webhook.



Figura 3.11: Plataforma Dialogflow con intents

El agente también puede ser accedido a través de la plataforma de Facebook Messenger

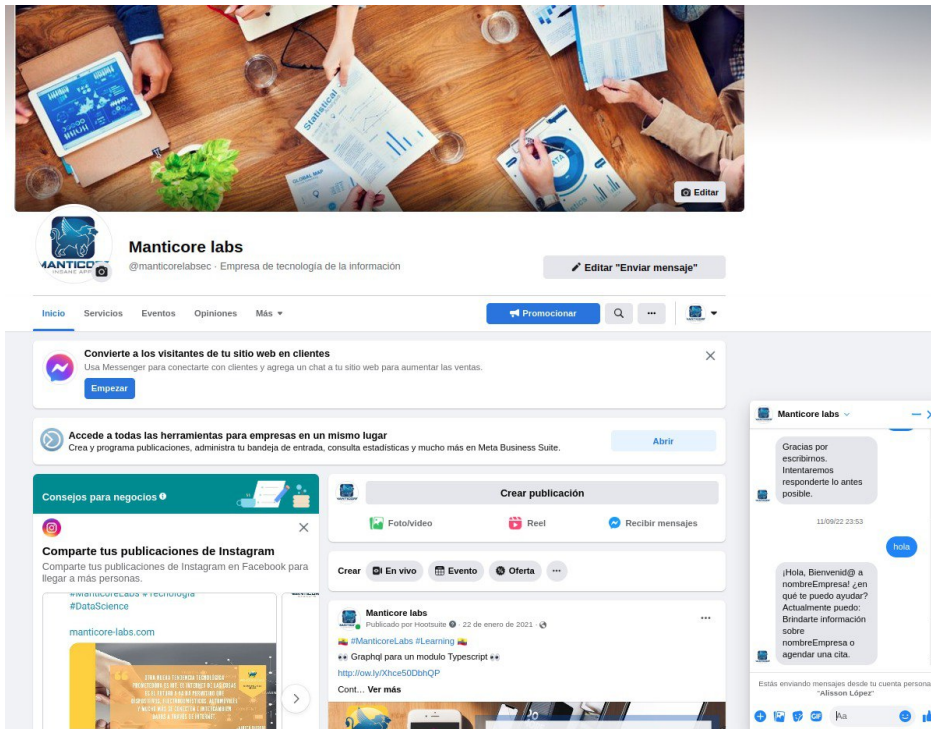


Figura 3.12: Plataforma Dialogflow con intents

Para acceder al agente se puede utilizar la plataforma de WhatsApp

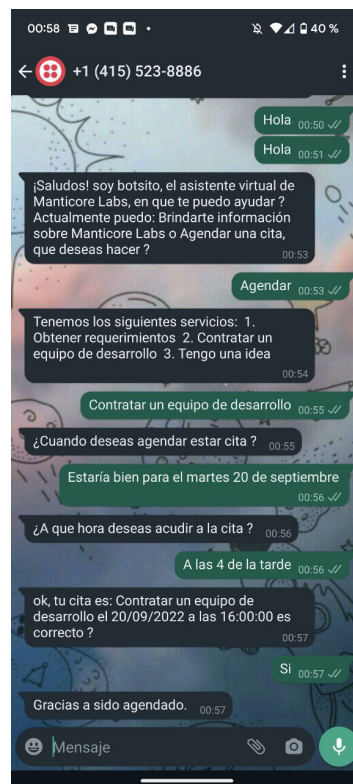


Figura 3.13: Plataforma WhatsApp con chatbot activo

3.2 RESULTADOS DE EVALUACIÓN

3.2.1 Resultados Prueba de caja negra

El motivo de realizar las pruebas de caja negra, es para validar el correcto funcionamiento del sistema sin tomar en cuenta el código fuente, a continuación se presenta el diseño de dichas pruebas.

Tabla 3.1: Prueba de caja negra 01: Ingreso al sistema de gestión de servicios

PCN-01	Ingreso al sistema de gestión de servicios
Propósito	Ingresar al sistema para poder modificar datos de la empresa, crear o modificar servicios y horarios de atención.
Prerrequisito	Usuario debe estar registrado.
Datos de entrada	Credenciales.
Pasos	<ol style="list-style-type: none">1. El usuario debe ingresar su usuario.2. El usuario debe ingresar su contraseña.
Resultado Esperado	Visualizar el menú de opciones.
Resultado Obtenido	Se visualiza el menú con sus diferentes opciones.
Resultado prueba	Correcto

Tabla 3.2: Prueba de caja negra 02: Actualizar datos de la empresa

PCN-02	Actualizar datos de la empresa
Propósito	Cambiar los datos principales de la empresa.
Prerrequisito	Registrar la empresa que va a brindar sus servicios.

Datos de entrada	Datos del formulario de empresa.
Pasos	<ol style="list-style-type: none"> 1. El usuario puede ingresar Nombre Comercial. 2. El usuario puede ingresar Información sobre la empresa. 3. El usuario puede ingresar Mensaje de bienvenida. 4. El usuario puede ingresar Latitud. 5. El usuario puede ingresar Longitud.
Resultado Esperado	Visualizar los cambios sobre la empresa.
Resultado Obtenido	Se visualiza los cambios en los campos actualizados.
Resultado prueba	Correcto

Tabla 3.3: Prueba de caja negra 03: Creación de un servicio

PCN-03	Creación de un servicio
Propósito	Creación de un servicio que brinda la empresa.
Prerrequisito	Tener registrado la empresa.
Datos de entrada	Datos del formulario del servicio.
Pasos	<ol style="list-style-type: none"> 1. El usuario debe ingresar Nombre del servicio 2. El usuario debe ingresar Descripción 3. El usuario debe ingresar Tiempo aproximado del servicio 4. El usuario debe ingresar Tiempo de espera del servicio

Resultado Esperado	Visualizar el servicio creado.
Resultado Obtenido	Se visualiza el servicio que se creó en el primer puesto.
Resultado prueba	Correcto

Tabla 3.4: Prueba de caja negra 04: Actualización de un servicio

PCN-04	Actualización de un servicio
Propósito	Modificar los campos del servicio.
Prerrequisito	El servicio debe estar registrado.
Datos de entrada	Datos del formulario del servicio.
Pasos	<ol style="list-style-type: none"> 1. El usuario puede ingresar Nombre del servicio. 2. El usuario puede ingresar Descripción. 3. El usuario puede ingresar Tiempo aproximado del servicio. 4. El usuario puede ingresar Tiempo de espera del servicio.
Resultado Esperado	Visualizar los cambios del servicio.
Resultado Obtenido	Se visualiza los cambios en los campos actualizados.
Resultado prueba	Correcto

Tabla 3.5: Prueba de caja negra 05: Creación de horario sobre un servicio

PCN-05	Creación de horario sobre un servicio
---------------	--

Propósito	Creación de un horario sobre el servicio que brinda la empresa.
Prerrequisito	El servicio debe estar registrado.
Datos de entrada	Datos del formulario de horario.
Pasos	<ol style="list-style-type: none"> 1. El usuario debe seleccionar el servicio. 2. El usuario debe ingresar el Día. 3. El usuario debe ingresar la Hora inicio. 4. El usuario debe ingresar la Hora fin.
Resultado Esperado	Visualizar el horario creado.
Resultado Obtenido	Se visualiza el servicio que se creó en el primer puesto.
Resultado prueba	Correcto

Tabla 3.6: Prueba de caja negra 06: Actualizar de horario sobre un servicio

PCN-06	Actualizar de horario sobre un servicio
Propósito	Modificar los campos del horario.
Prerrequisito	El horario debe estar registrado.
Datos de entrada	Datos del formulario de horario.
Pasos	<ol style="list-style-type: none"> 1. El usuario puede seleccionar el servicio. 2. El usuario puede ingresar el Dia. 3. El usuario puede ingresar la Hora inicio. 4. El usuario puede ingresar la Hora fin.

Resultado Esperado	Visualizar el horario actualizado.
Resultado Obtenido	Se visualiza los cambios en los campos actualizados.
Resultado prueba	Correcto

Tabla 3.7: Prueba de caja negra 07: Agendar cita

PCN-07	Agendar cita
Propósito	Agendar una cita mediante Facebook Messenger o WhatsApp.
Prerrequisito	La empresa debe estar registrada con sus respectivos servicios y horarios disponibles.
Datos de entrada	Respuestas del usuario.
Pasos	<ol style="list-style-type: none"> 1. El usuario debe tipar "Hola". 2. El usuario debe seleccionar un servicio. 3. El usuario debe fijar una fecha de atención del servicio.
Resultado Esperado	Obtener una cita agendada.
Resultado Obtenido	Se agendó una cita para el usuario.
Resultado prueba	Correcto

3.3 EVALUACIÓN DE USABILIDAD DEL SISTEMA

3.3.1 Descripción de la metodología MARS

MARS es una herramienta de evaluación de calidad la cual nos brinda una medida multidimensional con indicadores de calidad de compromiso, funcionalidad, estética y la calidad de la información, además, es fiable y flexible para investigadores, desarrolladores y profesionales en la salud [38]. MARS está conformada por 5 categorías, cada una de ellas contiene subcategorías [39].

Tabla 3.8: Categoría y subcategoría MARS.

Categoría	Subcategoría
Atractivo	<ol style="list-style-type: none">1. Entretenimiento.2. Interés.3. Personalización.4. Interactividad.5. Grupo Objetivo.
Funcionalidad	<ol style="list-style-type: none">1. Rendimiento.2. Facilidad de uso.3. Navegación.4. Diseño gestual.
Estética	<ol style="list-style-type: none">1. Diseño.2. Gráficos.3. Atractivo visual.

Información	<ol style="list-style-type: none"> 1. Exactitud descripción. 2. Objetivos. 3. Calidad de la información. 4. Cantidad de información. 5. Información visual. 6. Credibilidad. 7. Evidencias de uso.
-------------	---

3.3.2 Resultados MARS

Tanto la aplicación administrativa del chatbot como chatbot y su configuración fueron evaluados por tres grupos:

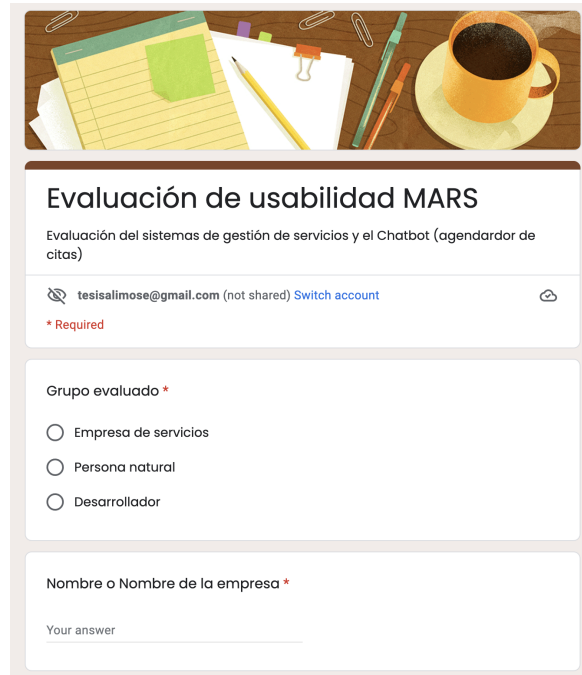
- Empresa de servicios.
- Personas naturales.
- Desarrolladores.

A cada grupo se le entregó un solo producto para su evaluación, a las empresas se les entregó un link, en el cual se les dio acceso a la gestión de servicios que van a brindar, mientras que a las personas naturales se les entregó acceso anticipado únicamente el chatbot para que pueda realizar citas, finalmente a los desarrolladores se le entregó el sistema de gestión de servicios y el chatbot.

Las condiciones para poder realizar la evaluación fueron las siguientes:

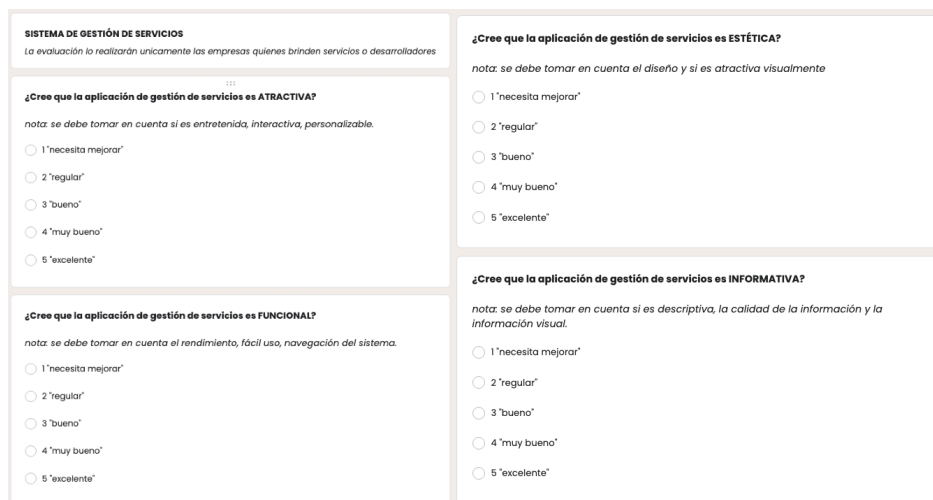
- Para las empresas de servicios la demostración del aplicativo web se realizó de manera presencial, además, tuvieron un demo del sistema.
- Para las personas naturales la demostración del chatbot se realizó de manera presencial, en donde se les brindó un demo.
- Para los desarrolladores la demostración del sistema se realizó de manera híbrida, además, tuvieron un demo tanto del chatbot como de la aplicación web.

Se realizó una encuesta con las categorías mencionadas anteriormente por la metodología MARS. como se muestra en las figuras 3.14, 3.15, 3.16



The image shows a survey form titled "Evaluación de usabilidad MARS". At the top, there is a decorative header with a coffee cup, pens, and sticky notes. Below the title, the subtitle reads "Evaluación del sistemas de gestión de servicios y el Chatbot (agendador de citas)". The form includes a user email field "tesisalimose@gmail.com (not shared) Switch account" and a red asterisk indicating a required field. The main question is "Grupo evaluado *" with three radio button options: "Empresa de servicios", "Persona natural", and "Desarrollador". Below this is a text input field for "Nombre o Nombre de la empresa *" with a placeholder "Your answer".

Figura 3.14: Grupo evaluado. Elaborado por autor.



The image shows a survey form with three evaluation questions. The first question is "¿Cree que la aplicación de gestión de servicios es ATRACTIVA?" with a note "nota: se debe tomar en cuenta si es entretenida, interactiva, personalizable." and five radio button options from "1 'necesita mejorar'" to "5 'excelente'". The second question is "¿Cree que la aplicación de gestión de servicios es ESTÉTICA?" with a note "nota: se debe tomar en cuenta el diseño y si es atractiva visualmente" and five radio button options. The third question is "¿Cree que la aplicación de gestión de servicios es FUNCIONAL?" with a note "nota: se debe tomar en cuenta el rendimiento, fácil uso, navegación del sistema." and five radio button options. The form is titled "SISTEMA DE GESTIÓN DE SERVICIOS" and includes a subtitle "La evaluación lo realizarán únicamente las empresas quienes brinden servicios o desarrolladores".

Figura 3.15: Evaluación para las empresas de servicios y desarrolladores. Elaborado por autor.

CHATBOT (Agendador de citas)
La evaluación la realizarán únicamente las personas naturales o desarrolladores

...
¿Cree que la aplicación de gestión de servicios es ATRACTIVA?
nota: se debe tomar en cuenta si es entretenida, interactiva, personalizable.

1 "necesita mejorar"
 2 "regular"
 3 "bueno"
 4 "muy bueno"
 5 "excelente"

...
¿Cree que la aplicación de gestión de servicios es ESTÉTICA?
nota: se debe tomar en cuenta el diseño y si es atractiva visualmente.

1 "necesita mejorar"
 2 "regular"
 3 "bueno"
 4 "muy bueno"
 5 "excelente"

...
¿Cree que la aplicación de gestión de servicios es FUNCIONAL?
nota: se debe tomar en cuenta el rendimiento, fácil uso, navegación del sistema.

1 "necesita mejorar"
 2 "regular"
 3 "bueno"
 4 "muy bueno"
 5 "excelente"

...
¿Cree que la aplicación de gestión de servicios es INFORMATIVA?
nota: se debe tomar en cuenta si es descriptiva, la calidad de la información y la información visual.

1 "necesita mejorar"
 2 "regular"
 3 "bueno"
 4 "muy bueno"
 5 "excelente"

Figura 3.16: Evaluación para las personas naturales y desarrolladores. Elaborado por autor.

El chatbot fue evaluado por tres grupos:

☐ Empresa de servicios:

- ✦ Manticore
- ✦ MTE
- ✦ JR SUPPLY
- ✦ Creditu
- ✦ DEAS CORE

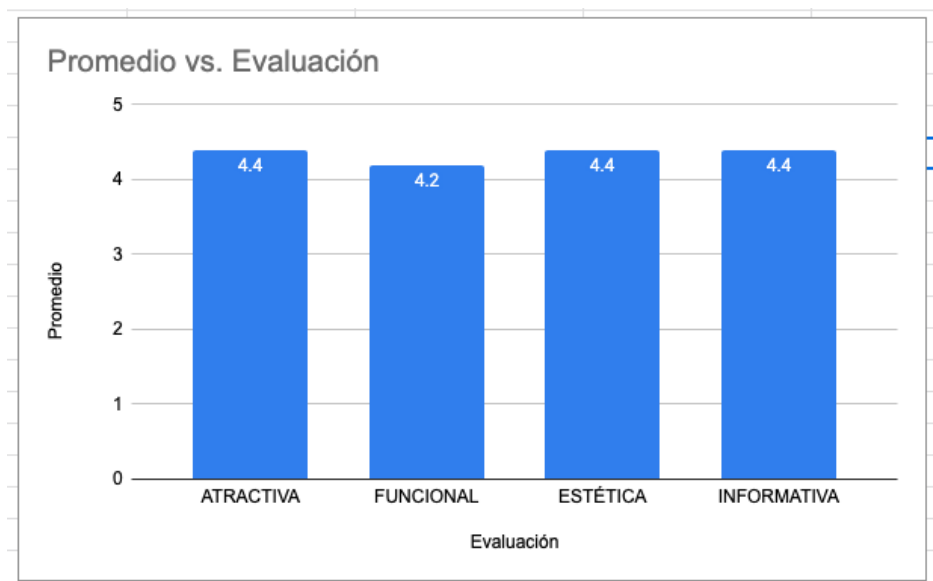


Figura 3.17: Resumen de evaluación de chatbot por las empresas. Elaborado por autor.

Los resultados fueron excelentes se obtuvo una calificación de 4.35 sobre 5.

❑ Desarrolladores:

- ❖ Kevin Jimenez
- ❖ Andrés Pantoja
- ❖ Josue Diaz
- ❖ Cristian Lara
- ❖ Andrés Torres
- ❖ Edwin Guamushig

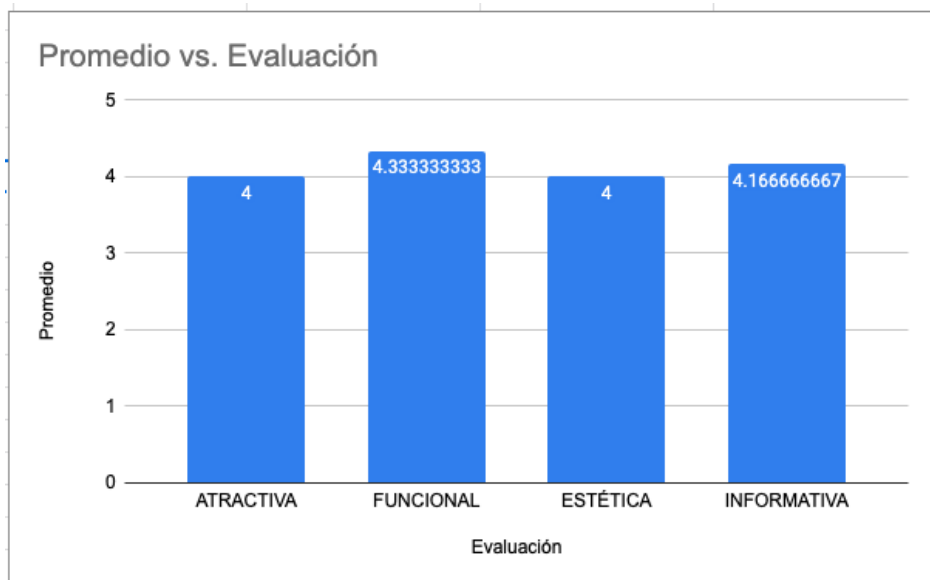


Figura 3.18: Resumen de evaluación de chatbot por los desarrolladores. Elaborado por autor.

Los resultados fueron excelentes se obtuvo una calificación de 4.12 sobre 5.

❑ Personas naturales:

- ❖ Evelyn Mejía
- ❖ Marilyn Guano
- ❖ Edwin López
- ❖ Camila Jaramillo
- ❖ Leslie López

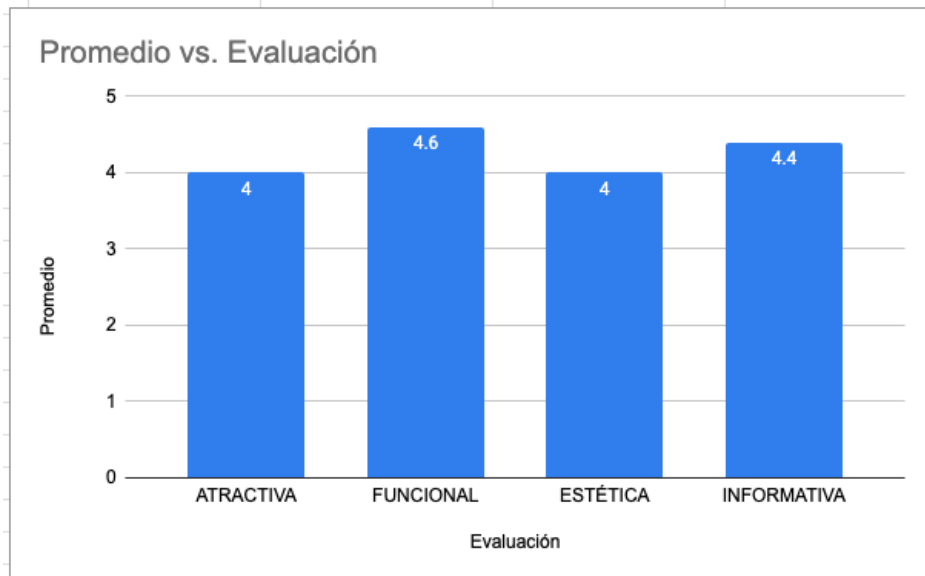


Figura 3.19: Resumen de evaluación de chatbot por las personas naturales. Elaborado por autor.

Los resultados fueron excelentes se obtuvo una calificación de 4.25 sobre 5.

3.4 DISCUSIÓN

Los resultados de las evaluaciones realizadas a los tres grupos permiten determinar un buen desempeño. Pequeñas variaciones se deben a las condiciones iniciales de presentación del sistema y del chatbot. Las empresas accedieron al sistema a través de un link temporal y el chatbot funcionó rápido y sin errores. La usabilidad del aplicativo web administrativo y del chatbot fue evaluada con calificaciones elevadas según los parámetros de MARS.

Después de la evaluación, se realizó una reunión con los desarrolladores para obtener su feedback sobre la experiencia de uso del chatbot. Se identificaron errores como el chatbot entrar en un bucle hasta obtener todos los parámetros requeridos para la cita, pero también se detectaron oportunidades para mejorar el flujo en el futuro. Los desarrolladores encontraron al chatbot fácil de usar y rápido, con un diseño que funciona en las principales redes sociales.

A continuación se mencionan algunos comentarios de los desarrolladores:

- ❑ El chatbot entiende muy bien cada palabra o pregunta que se le realiza, además, es muy fácil de usar y veloz. - .Andrés Torres (Desarrollador Front-end)"

- ❑ Tanto el sistema como el chatbot en su uso es bastante sencillo y fácil, su diseño se adapta a cualquier dispositivo. - "Josué Díaz (Desarrollador full stack)"
- ❑ El responsive del sistema es excelente, debido a su diseño y la ubicación de cada componente. - "Kevin Jiménez (Consultor técnico)"

Los usuarios finales solo pudieron acceder al chatbot a través de la página de Facebook de la empresa porque requería una cuenta de marketing que la empresa aún no tiene. Los comentarios de los usuarios fueron positivos, pudieron agendar citas fácilmente y se redujo el tiempo que normalmente se tiene al agendarlo de manera manual.

En general, tanto el aplicativo web administrativo como el chatbot fueron muy útiles, especialmente por la facilidad de configurar el chatbot a través del aplicativo web sin tener que programar. Los errores y bugs serán corregidos para mejorar el agente y su entrenamiento. En algunas ocasiones, el chatbot perdía el flujo y no respondía correctamente debido a que no encontraba la palabra clave para activar el intent adecuado.

4 CONCLUSIONES

4.1 CONCLUSIONES

- ❑ Se logró agregar las citas con un flujo conversacional simple para ser agendadas por el chatbot a un calendario de Google Calendar.
- ❑ Se logró desarrollar el chatbot el cual es customizable ya que permite modificar la información que se presenta a través de una aplicación web.
- ❑ Se diseñó una solución eficiente en cuanto a arquitectura, con tres componentes clave: un backend que maneja todas las solicitudes que llegan desde plataformas como WhatsApp y Facebook Messenger y se integra con el API de Google Calendar. La segunda sección se basa en Dialogflow, que se integra con un webhook y hace coincidir los diferentes intents y controladores del backend para responder adecuadamente al intent seleccionado. Por último, una aplicación cliente hecha con Angular personaliza la información que se presenta en el flujo del chatbot.
- ❑ Se logró realizar satisfactoriamente las pruebas de usabilidad con la metodología MARS, los participantes lograron agendar una cita con facilidad cuando lograban comprender el flujo, prediseñado mientras que los administradores del chatbot pudieron customizar la información de manera fácil e intuitiva.
- ❑ La documentación de Dialogflow cambia constantemente inicialmente era un servicio de Google, en versiones anteriores no contenía muchas restricciones de pago, sin embargo, en la versión actual, cada integración y sección está administrada por Google Cloud la cual es monetizada.
- ❑ La metodología SCRUM permite distribuir el trabajo a realizarse en secciones iterables, priorizando las historias de usuario relevantes para que generaban avances

importantes en el proyecto, lo cual permitió que no hubieran retrasos y para el sprint final solo se realicen las integraciones respectivas con las diferentes plataformas.

- ❑ Mientras se buscaba información sobre los posibles sistemas de agendamiento de citas populares, se encontró información sobre el tiempo que tarda una persona común en agendar un tipo de cita con dos parámetros como lo son día y hora. El proceso de agendamiento comienza con la solicitud de servicio, el operador consulta la disponibilidad y finalmente realiza el agendamiento, el tiempo que dura este proceso es bastante considerable sin tomar en cuenta el recurso humano puede atender solo una petición a la vez, versus el chatbot el cual logra atender varias peticiones con origen en diferentes plataformas y se encuentra disponible a cualquier hora, optimizando el tiempo de respuesta y casi anulando por completo el tiempo de espera.

4.2 RECOMENDACIONES

- ❑ Se recomienda entrenar el chatbot, para poder reaccionar a diferentes jergas que se manejan en el español alimentando la base de datos con las frases más utilizadas por el resto de los clientes permitiendo que la palabra clave coincida con mayor facilidad con el intent.
- ❑ Dialogflow actualmente es un recurso con bastantes limitaciones de pago, inicialmente la mayoría de las funcionalidades eran gratuitas y dependían netamente de la utilización de código, con la actual tendencia tecnológica de inteligencia artificial cada integración tiene un valor definido y este se descuenta desde la cuenta principal de Google Cloud lo cual la hace poco accesible a futuros desarrollos.
- ❑ Dialogflow permite integrarse a diferentes asistentes de voz con el mismo flujo se podría agendar una cita netamente desde cualquier dispositivo de voz que cuente con el asistente virtual.
- ❑ Dialogflow cuenta con su propia integración con Nodejs, esta librería permite utilizar respuestas con mayores características como son cards, galerías y botones, que podrían mejorar la presentación de las respuestas del chatbot al usuario final.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] BBVA. «Reinventar la empresa.» (Agosto de 2015), dirección: <https://www.bbvaopenmind.com/wp-content/uploads/2015/01/BBVA-OpenMind-libro-Reinventar-la-Empresa-en-la-Era-Digital-empresa-innovacion1-1.pdf>. (Último acceso: 20 Julio 2022).
- [2] A. C. C. Triviño. «Perspectiva para la mejora del servicio al cliente en las empresas publicas de la ciudad de Guayaquil.» (sep. de 2019), dirección: <https://dspace.ups.edu.ec/bitstream/123456789/17454/1/UPS-GT002690.pdf>. (Último acceso: 20 Julio 2022).
- [3] INEC. «CALIDAD DE LOS SERVICIOS EN ECUADOR.» (jun. de 2008), dirección: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Bibliotecas/Estudios/Estudios_Economicos/Calidad_Servicio_2008.pdf. (Último acceso: 20 Julio 2022).
- [4] D. A. S. R. D. Dharani. «NTEGRATION OF AIML CHATTER BOT FOR NEWS APPLICATION ON.» (mayo de 2016), dirección: <https://www.irjet.net/archives/V3/i5/IRJET-V3I5439.pdf>. (Último acceso: 20 Julio 2022).
- [5] A. S. a. A. Madheswari. «The Role of Smart Personal Assistant for improving personal Healthcare.» (mayo de 2015), dirección: ljaems.com. (Último acceso: 20 Julio 2022).
- [6] J. Lloret Egea, «Procesamiento del Lenguaje Natural (PLN) | Natural Language Processing (NLP),» ago. de 2019. DOI: 10.13140/RG.2.2.30635.39205.
- [7] X. Liu y D. Wu, «From Natural Language to Programming Language,» en mayo de 2018, págs. 110-130, ISBN: 9781522559702. DOI: 10.4018/978-1-5225-5969-6.ch004.
- [8] R. Celi, E. Varela Tapia, I. Acosta y N. Montaña-Pulzara, «Técnicas de procesamiento de lenguaje natural en la inteligencia artificial conversacional textual,» *AlfaPublicaciones*, vol. 3, págs. 40-52, nov. de 2021. DOI: 10.33262/ap.v3i4.1.123.

- [9] H. Candello y C. Pinhanez, «Designing Conversational Interfaces,» en oct. de 2016, <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=25402>.
- [10] M. Morales-Rodríguez. «Agentes Conversacionales como un Sistema de Diálogo.» (2020), dirección: https://www.academia.edu/1187395/Agentes_Conversacionales_como_un_Sistema_de_Di%C3%A1logo. (Ultimo acceso: Agosto 2022).
- [11] C. M. López, «Interfaz conversacional en Dialogflow para recomendación de películas,» en sep. de 2019.
- [12] GitLab. «Webhooks all tiers.» (2020), dirección: <https://docs.gitlab.com/ee/user/project/integrations/webhooks.html>. (Ultimo acceso: Agosto 2022).
- [13] M. Biehl, «Webhooks: Events for RESTful APIs,» en dic. de 2017, pág. 65.
- [14] J. P. Mestras, «El patrón Modelo-Vista-Controlador (MVC),» en dic. de 2017.
- [15] G. M. y. V. S. Valarezo Pardo Honores Tapia, «COMPARACIÓN DE TENDENCIAS TECNOLÓGICAS EN APLICACIONES WEB,» en oct. de 2018, <https://www.3ciencias.com/wp-content/uploads/2018/09/Art2.pdf>.
- [16] Unknown, «Aplicación Web,» en oct. de 2018, <http://meteo.ieec.uned.es/wwwsumeteo2/Memoria/Capitulo3.pdf>.
- [17] V. S. Code. «Visual Studio Code Documentation.» (2020), dirección: <https://code.visualstudio.com/docs>. (Ultimo acceso: Septiembre 2021).
- [18] draw.io. «draw.io.» (2020), dirección: <https://drawio-app.com/>. (Ultimo acceso: Septiembre 2021).
- [19] Git. «Git Documentation.» (2020), dirección: <https://git-scm.com/doc>. (Ultimo acceso: Septiembre 2021).
- [20] GitHub. «GitHub Guides.» (2020), dirección: <https://guides.github.com/features/pages/>. (Ultimo acceso: Septiembre 2021).
- [21] GitKraken. «GitKraken Documentation.» (2020), dirección: <https://support.gitkraken.com/>. (Ultimo acceso: Septiembre 2021).
- [22] MySQL. «MySQL Documentation.» (2020), dirección: <https://dev.mysql.com/doc/>. (Ultimo acceso: Septiembre 2021).
- [23] Typescript. «Typescript Documentation.» (2020), dirección: <https://support.gitkraken.com/>. (Ultimo acceso: Septiembre 2021).

- [24] NestJS. «NestJS Documentation.» (2020), dirección: <https://support.gitkraken.com/>. (Ultimo acceso: Septiembre 2021).
- [25] HTML5. «HTML5.» (2020), dirección: <https://developer.mozilla.org/es/docs/Glossary/HTML5>. (Ultimo acceso: Septiembre 2021).
- [26] SCSS. «SCSS Documentation.» (2020), dirección: <https://sass-lang.com/documentation/>. (Ultimo acceso: Septiembre 2021).
- [27] Angular. «Angular Documentation.» (2020), dirección: <https://angular.io/docs>. (Ultimo acceso: Septiembre 2021).
- [28] PrimeNG. «PrimeNG.» (2020), dirección: <https://www.primefaces.org/primeng/>. (Ultimo acceso: Septiembre 2021).
- [29] Angular. «Angular material.» (2020), dirección: <https://material.angular.io/guide/getting-started>. (Ultimo acceso: Septiembre 2021).
- [30] Google. «Dialogflow Documentation.» (2020), dirección: <https://cloud.google.com/dialogflow/docs>. (Ultimo acceso: Septiembre 2021).
- [31] Google. «Google Calendar API.» (2020), dirección: <https://developers.google.com/calendar/api>. (Ultimo acceso: Septiembre 2021).
- [32] Unknown, «MANIFIESTO AGILE. VALORES Y PRINCIPIOS SCRUM,» en oct. de 2018, <https://academy.lareddelcoach.com/wp-content/uploads/2021/05/MODULO-I-TEMA-2.pdf>.
- [33] V. R. J. C. Amaro Calderon Sarah Dámaris, «Metodologías Ágiles,» en oct. de 2018, <https://www.scribd.com/document/575878555/Metodologias-Agiles-with-cover-page-v2>.
- [34] K. S. J. Sutherland, «La Guía Scrum,» en oct. de 2020, <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>.
- [35] M. Cristiá, «Introducción a la Ingeniería de Requerimientos,» en oct. de 2018, <https://www.fceia.unr.edu.ar/mcristia/publicaciones/ingreq-a.pdf>.
- [36] DBMedia, «ABC del manejo de filas,» en oct. de 2018, https://www.debmedia.com/ABC_Filas.pdf.
- [37] Google. «Conceptos básicos de Dialogflow CX.» (2020), dirección: <https://cloud.google.com/dialogflow/cx/docs/basics?hl=es-419>. (Ultimo acceso: Septiembre 2021).

- [38] S. Stoyanov, L. Hides, D. Kavanagh, O. Zelenko, D. Tjondronegoro y M. Mani, «Mobile App Rating Scale: A New Tool for Assessing the Quality of Health Mobile Apps,» *JMIR mHealth uHealth*, vol. 3, e27, mar. de 2015. DOI: 10.2196/mhealth.3422.
- [39] F. F. E. PÉREZ, «REVISIÓN Y EVALUACIÓN DE APLICACIONES EDUCATIVAS PARA PERSONAS CON DISCAPACIDAD INTELECTUAL,» en ago. de 2018, <https://bibdigital.epn.edu.ec/bitstream/15000/19703/1/CD-9111.pdf>.

I APLICACIÓN WEB

I.1 REPOSITORIO FRONT-END

<https://github.com/alimonse/tesis-front.git>

I.2 REPOSITORIO BACK-END

<https://github.com/alimonse/tesis.git>

I.3 DISEÑO DE LA APLICACIÓN

<https://www.figma.com/file/bq0uO6aGG9QObzCIVDgv2R/Mockups-tesis?node-id=0%3A1&t=4BWuODbOTYav3nuZ-1>

I.4 ESQUEMA DE FLUJOS CONVERSACIONALES

https://docs.google.com/document/d/1-i815A_FNko14y_y3KEbxxQjrPLoEw42mfXRhOxz8Sk/edit?usp=sharing

I.5 HISTORIAS DE USUARIO DEL PROYECTO

https://docs.google.com/document/d/132Zox_CtjytGwdNGxdkdhwcQwPr_AbnKhmq0j_vEqy0/edit?usp=sharing

I.6 DIAGRAMA DE BASE DE DATOS

https://docs.google.com/document/d/1plyJWhEMLfVaNSYXIVr1vJal6D-rbReoWG7c42eYsDM/edit?usp=share_link

I.7 SITEMAP DE LA APLICACIÓN DE GESTIÓN

<https://docs.google.com/document/d/1BnV3doMQdOTF7LxLPmlcltrBkB5W01DV-ACCOklhaE/edit?usp=sharing>

I.8 ENCUESTAS DE USABILIDAD DEL SISTEMA

https://docs.google.com/spreadsheets/d/1Bpa_0RKgkTXsnO2a0s2KZ2ptxKCJUMVygcV-pPdttdQ/edit?usp=sharing

I.9 IMÁGENES EXTRAS DEL DESARROLLO DE LOS SPRINTS

https://docs.google.com/spreadsheets/d/18TUnORn_kli6E8t4juyll6rllTDVliFG/edit?usp=sharing&oid=117261994216011663508&rtpof=true&sd=true