

UNIVERSIDAD TECNOLÓGICA NACIONAL



INTELIGENCIA ARTIFICIAL

TP II : Chatbot

2024 – Grupo 3
Regional Santa Fe

IZAGUIRRE, Ezequiel Matías
ezequielmizaguirre@gmail.com

RESUMEN

Este informe presenta el desarrollo de un agente inteligente capaz de relacionarse con el mundo en donde se desenvuelve, utilizando técnicas de procesamiento del lenguaje natural.

Se propone el desarrollo de un agente tipo **Chatbot** que pertenece a una empresa de internet y telefonía (ISP) y cuyo objetivo es asistir, resolver y gestionar problemas y reclamos de los clientes. El agente será capaz de mantener conversaciones naturales con clientes, entender sus solicitudes, reclamos, consultas, etc; y responder acordemente.

Para ello, **la primer sección** se encarga de esbozar la idea conceptual de qué tipo de información se debe entender (vocabulario), cómo se podría idear una solución para procesarlas y cuáles son las respuestas.

La **segunda sección** muestra la implementación junto con toda el vocabulario completo, las herramientas utilizadas y cómo algunos de los conceptos explicados antes se aplican.

La **tercer sección** muestra los resultados obtenidos con conversaciones reales en una aplicación de mensajería de uso común.

ÍNDICE

Contenido

VOCABULARIO.....	4
1.1 • Introducción.....	4
1.2 • Dominio.....	4
1.3 • Base de conocimiento.....	8
1.3.1 • Árbol ilustrativo	9
1.3.2 • Algunas preguntas.....	10
IMPLEMENTACIÓN	12
2.1 • Introducción	12
2.2 • Diseño	13
2.2.1 • Intents.....	13
2.2.2 • Contexts y prioridades.....	14
2.2.3 • Parameters.....	15
2.2.4 • Events.....	16
2.2.5 • Fulfillments.....	16
2.2.6 • Integrations	16
2.3 • Comentario final.....	17
RESULTADOS.....	18
3.1 • Introducción	18
3.2 • Conversaciones.....	18
CONCLUSIÓN.....	27

1.1 • Introducción

Para poder comenzar el desarrollo del agente, la primera cuestión a considerar es qué tipo de vocabulario debe ser capaz de reconocer, interpretar y responder. Como existen muchos tipos de preguntas y múltiples maneras de expresarlas, se necesita idear un sistema para captar las palabras claves y procesarlas con mecanismos de inferencia.

Para conocer las respuestas, se debe cargar al agente con una base de conocimiento inicial. De forma genérica se divide el dominio en 3 categorías; para cada una de ellas se define una lista de preguntas o problemas frecuentes, junto con las respuestas que se deben dar. Algunas preguntas requerirán introspección adicional por parte del agente para conocer más detalles, por lo que se debe contar con herramientas para establecer flujos de decisión y almacenar la información que se va obteniendo.

1.2 • Dominio

En general se deberán responder las siguientes cuestiones:

1. Dar de baja (cancelar servicio, motivo cancelación)
2. Dar de alta, contratar (información planes, proceso contratación, instalación del servicio?)
3. Pagar (Métodos de pago, confirmación de pago, facturación)
4. Problemas con el Internet y modem (conexión a internet, configuración modem)
5. Consultas sobre planes, cobertura (info planes, comparación planes, áreas de cobertura, mejora de plan)
6. Denuncias, quejas (registrar queja, seguimiento de queja)

A continuación se mostrarán las categorías y preguntas que el agente será capaz de responder.

Asistencia al usuario

Agrupar preguntas que requieren respuestas informativas sobre los servicios que ofrece la empresa.

Preguntas	Palabras claves
Quiero contratar / dar de alta / baja un plan	Contratar, plan, baja, alta
Quiero cambiar un plan	Cambiar, plan
¿Hay cobertura en mi zona?	Cobertura, zona

¿Cuál es mi plan ideal?	Plan, ideal
¿Qué significa ... X? (Mb/s, Fibra, etc)	X, significado

Soporte técnico

Agrupar preguntas o reclamos sobre el funcionamiento del servicio.

Preguntas	Palabras claves
No tengo Wi-Fi / Internet / Televisión	Wi-Fi, no, Internet, Televisión
El Internet está muy lento	Internet, lento
Tengo poca cobertura	Cobertura, poca
¿Cómo cambio la contraseña de mi modem?	Modem contraseña
¿Cómo pago mi factura?	Factura, pago

Registro de incidentes

Este es un grupo especial que contempla la posibilidad de que los usuarios guarden reclamos en la base de datos que serán atendidos por un empleado de la empresa. Los clientes pueden luego consultar el estado de sus reclamos.

Preguntas	Palabras claves
No me llegó mi factura	Factura
No hay internet, televisión (infraestructura)	Internet, Televisión
No llega la fibra a mi zona	Fibra, Zona
Me están cobrando demasiado	Cobro, demasiado, factura

El agente deberá utilizar las palabras claves y realizarle preguntar al cliente para inferir cuál es la cuestión a la que se está haciendo referencia. Este proceso requiere de algún sistema que permita medir el grado de similitud entre las entradas recibidas y la información que caracteriza a una pregunta particular. Cuando se encuentra una coincidencia con un alto grado de similitud (por ejemplo, mayor al 80%) el agente puede confiar en que su respuesta será la correcta. Si existen varias respuestas con niveles similares, se deberá idear algún mecanismo de resolución con reglas que asignen prioridades.

Fuente de datos

Las respuestas del agente se redactarán usando como base los centros de ayuda de varios proveedores de Internet de la zona (*Telecom, Fibertel, Gigared*, etc) obtenido de sus sitios web.

A continuación se ejemplifica un flujo de decisión para la cuestión: *Quiero contratar un plan*

Cliente: Quiero contratar un plan

Chatbot: ¡Excelente! Te haré algunas preguntas para determinar el plan más óptimo según tus necesidades.

Chatbot: ¿Dónde resides?

Cliente: Santa Fe

Chatbot: ¿Es para uso doméstico o empresarial?

Cliente: Doméstico

Chatbot: De acuerdo, ¿cuántas personas se conectarán a la red?

Cliente: Unas cuantas...

Chatbot: Disculpa, no te entendí, ¿podrías explicarlo de otra manera?

Cliente: 5 personas

Chatbot: Anotado. ¿Normalmente usas internet desde una PC o celular?

Cliente: Me equivoqué en la otra pregunta.

Chatbot: Volvamos hacia atrás, ¿cuántas personas se conectarán a la red?

Cliente: 6 personas

Chatbot: Corregido. ¿Normalmente entrás a Internet desde una PC o celular?

Cliente: Desde el celular

Chatbot: Casi estamos. Última pregunta, ¿te gustaría agregar el servicio de televisión?

Cliente: No

Chatbot: ¡Perfecto! Dados los datos que me has provisto, he encontrado el plan ideal para vos:

Plan conectividad #1ACF

- Internet Wi-Fi por fibra óptica
- Bajada: 100 MB/s
- Subida: 20 MB/s
- +
- Línea móvil 10 GB
- \$ 25.000

¿Estás de acuerdo?

Cliente: No necesito la línea móvil.

Chatbot: Lo siento, ¿qué te parece esta opción?

Plan conectividad #1AC

- Internet Wi-Fi por fibra óptica
- Bajada: 100 MB/s
- Subida: 20 MB/s
- \$ 18.000

Cliente: Suena bien.

Chatbot: ¡Listo! Visitá este [link] para ponerte en contacto con nosotros y tener tu internet funcionando lo más rápido posible. ¿Alguna otra cosa?

Cliente: Nada más por ahora.

Chatbot: ¡Adios! Cualquier otra cosa, acá estoy.

En este ejemplo, el agente comprendió la categoría (**asistencia al usuario**) y pregunta abstracta (**contratar un plan**) del cliente, luego comenzó un proceso de búsqueda para tratar de encontrar la mejor respuesta.

1.3 • Base de conocimiento

El ejemplo de la sección anterior muestra que para la cuestión **contratar un plan**, el agente tiene un conjunto de respuestas y debe hilar más fino para encontrar la adecuada.

Entonces, la base de conocimiento está organizada de manera jerárquica según el árbol:

Base de conocimiento (Nodo raíz)

|----- Categoría (Nodo interno)

|----- Pregunta (Nodo interno)

|----- Características (Nodo interno)

|----- Respuesta final (Nodo hoja)

Cada nodo interno puede tener más de un hijo, o sea, una categoría varias preguntas y cada pregunta varias características. Las características son recursivas, pueden estar compuestas de otras.

El agente posee **preguntas** en cada nodo interno y **respuestas** en los nodos hojas.

Cada nodo está compuesto de un conjunto de palabras claves; los hijos tienen las palabras de sus padres más las propias. Para que el agente baje un nivel (nodo) en el árbol, se debe haber encontrado un nivel de similitud (entre un nodo y la información de entrada) superior a algún límite.

Sin embargo, existe la posibilidad de que el cliente entregue información específica de un nodo hijo **antes** de que el agente haya inferido el nodo superior.

En el ejemplo anterior, el cliente comienza la conversación diciendo que quiere contratar un plan. Esto genera las palabras clave [contratar, plan] que pertenecen al nodo *contratar plan* dentro de *asistencia al usuario*.

Asistencia al usuario – Claves: [ayuda] - Similitud: (0%)

|----- **Quiero contratar un plan** – Claves: [ayuda, contratar, plan] - Similitud: (66%)

Vemos que el nodo hijo tiene una mayor similitud que el padre, porque la conversación empezó al revés. Esto indica que, a la hora de calcular similitudes, el valor en el nodo padre debe ser el máximo entre la similitud de sus claves y la de sus hijos. En este caso, se tiene:

$$\text{MAX}(0, 66) = 66$$

Por lo que la similitud del padre deberá tomar el valor 66%. Con este sistema, si un nodo en el nivel **k** tiene una similitud mayor a todos sus ascendentes, su valor se propagará hasta el nodo raíz.

Si se decidiera que 66% es una similitud aceptable, se continuaría desde el nodo **Quiero contratar un plan** y se asumen las claves superiores como verdaderas (ayuda).

1.3.1 • Árbol ilustrativo

La base de conocimientos del agente tendría la siguiente forma.

Asistencia al usuario [ayuda]

|----- Contratar un plan [ayuda, contratar, plan]

|----- Doméstico [ayuda, contratar, plan, doméstico]

|----- Muchas personas [..., muchas, personas]

|----- Celular [..., celular]

...

|----- ¡Perfecto! Tengo el Plan #4FCD...

|----- Sin celular [..., sin celular]

...

|----- ¡Perfecto! Tengo el Plan #4FCH...

|----- Empresarial [ayuda, contratar, plan, empresarial]

...

|----- ¡Perfecto! Tengo el Plan #5TRCD

|----- Dar de baja un plan [ayuda, baja, plan]

...

Solo se muestran las respuestas finales, pero el Chatbot también cuenta con **preguntas** para cada nodo interno

El agente se posiciona en el nodo más similar, realiza preguntas para incrementar su nivel de similitud y, si este alcanza un límite, se considera al nodo confiable y se ingresa a él. El sistema debe contar con algún sistema de prioridades si existen varios nodos a elegir.

Volviendo al ejemplo, supongamos que estamos en el siguiente punto:

Chatbot: De acuerdo, ¿cuántas personas se conectarán a la red?

El agente se encuentra en el nodo Asistencia al usuario → Contratar un plan → **Doméstico** y sus hijos son:

- Muchas personas [..., muchas, personas, > 5]
- Algunas personas [..., par, personas, 5 < y > 2]

- Pocas personas [..., pocas, personas, solo, $5 < y > 2$]

Entonces sabe que ahora debe preguntar por alguna cantidad de personas para poder seguir avanzando, lo que lo lleva a la pregunta *¿cuántas personas se conectarán a la red?*

Otros ejemplos

Soporte técnico → Mi velocidad es muy lenta

¿Podrías verificar si hay alguno realizando descargas?

Mi velocidad es muy lenta → Nadie está descargando

Mi velocidad es muy lenta → Alguien está descargando

Soporte técnico → No tengo Wi-Fi

¿Has intentado reiniciar tu router?

No tengo Wi-Fi → No reinicié el router

No tengo Wi-Fi → Reinicié el router

1.3.2 • Algunas preguntas

A continuación se ejemplifican algunas preguntas o frases que el agente podría hacerle al usuario.

Asistencia al usuario

- ¿Necesitas ayuda?
- ¿En qué puedo servirte?
- ¿Tienes una consulta?

Soporte técnico

- ¿Algo anda mal?
- ¿Tienes un problema con tu Internet?
- ¿Las cosas no andan bien?

Registro de incidencias

- ¿Te gustaría realizar una queja?
- ¿Tienes una queja?

General

- Lamento escuchar eso...
- No te he entendido, ¿podrías repetir?
- Volvamos a comenzar...
- ¡Buen día! ¿En qué te puedo ayudar?

- ¡Nos vemos!
- Si tenés otra duda, no dudés en preguntarme

2.1 • Introducción

Para la implementación se optó por utilizar la herramienta **DialogFlow** de Google.

DialogFlow es un entorno de desarrollo en la nube que permite la creación de agentes inteligentes orientados a la asistencia de usuarios. Está pensando, y tiene integración directa, con plataformas como Google Assistant, Facebook Messenger, Telegram y más.

DialogFlow es una herramienta madura, con una buena documentación y una creciente comunidad detrás. Debido a su propósito específico, resulta ser una opción ideal para el desarrollo del trabajo práctico.

Existen tres versiones del software. Sus principales diferencias se listan a continuación.

- **DialogFlow Trial:** No posee costos de uso pero tiene funcionalidad limitada. Pensada como entorno de experimentación y aprendizaje.
- **DialogFlow Essentials (ES) :** Es la edición utilizada para este trabajo. También es gratuita pero posee un sistema de *pay-as-you-go* cuyo coste es proporcional al uso.
- **DialogFlow CX:** Posee la misma política que la versión Essentials. Su principal diferencia con las dos anteriores es que rediseña los conceptos y herramientas para el desarrollo de agentes. Se requiere la creación de proyectos que implementen *CX Agent*.

Como se explicó antes, las versiones de pago poseen un coste proporcional a la cantidad de uso, medido en *Quotas*. Abajo se ejemplifican los costos para la versión Essentials.

CX Agent		ES Agent
Feature	Trial Edition	Essentials Edition
Text (includes all DetectIntent and StreamingDetectIntent requests that do not contain audio)	Production-ready quota	Production-ready quota
	No charge	\$0.002 per request
Audio input (also known as speech recognition, speech-to-text, STT)	Limited quota	Production-ready quota
	No charge	\$0.0065 per 15 seconds of audio
Audio output (also known as speech synthesis, text-to-speech, TTS)	Limited quota	Production-ready quota
	No charge	\$4 to \$16 per million characters
Knowledge connectors (Beta)	Limited quota	Production-ready quota
	No charge	No charge
Sentiment analysis	Not available	Unlimited quota
		\$0.25 to \$1.00 per 1000 requests

Por otro lado, nuestro agente hace uso de una base de datos donde guarda la información de los planes, los usuarios y los reclamos. Ésta se desarrolló con **MongoDB** y se desplegó en la nube usando **Atlas**.

Para el manejo de la base de datos y de otras cuestiones de implementación, se requirió un servidor backend. El servidor está hecho con: **NextJS**, **TypeScript**, **Prisma** y desplegado en la nube con **Vercel** como un *Serverless*.

La necesidad de este servidor se explica en la subsección **Fulfillments**.

2.2 • Diseño

DialogFlow posee conceptos propios pero similares a muchas otras herramientas de creación de agentes. Se verá como éstos solucionan los problemas descriptos en la sección anterior. Una importante aclaración es que el reconocimiento de texto se hace con el uso de tanto similitud de gramática (o Grammar Matching) y **Machine Learning**.

El desarrollo de agentes se hace con una consola directamente desde el navegador.

Todo el flujo del agente se baja en **conversaciones**, que son intercambios de información entre el mismo y un usuario final (o simplemente usuario) desde un inicio hasta un fin (el fin lo puede generar cualquiera de las 2 partes). Estos intercambios pueden ser en texto, audio o videollamada, aunque para este trabajo se prefirió centrarse en conversaciones textuales.

2.2.1 • Intents

En primer lugar, los agentes se basan en **Intents**. Estos representan las “intenciones” que un agente puede comprender de un usuario final durante una conversación.

Cada Intent asocia un conjunto de palabras u oraciones claves a un concepto al que se es capaz de dar servicio. La intención es “matcheada” o asociada cuando las entradas del usuario son lo suficientemente cercanas. Las entradas se llaman “Training phrases”. Para cada Intent, el agente tendrá un conjunto de respuestas.

Por ejemplo, un Intent del tipo “Pronóstico” podría reconocer y responder preguntas relacionadas con el clima para un agente meteorológico. Esto va en correspondencia con lo explicado en la sección anterior, donde se consideraba que el agente debía recibir las entradas y calcular un grado de precisión con algún nodo del árbol.

A continuación se muestran los Intents implementados y sus usos.

Presentación

Se asocia al inicio de la conversación y cuando el usuario le pide al agente que se presente.

Agradecimiento

Se asocia cuando el usuario le da las gracias o un cumplido al agente. El último responde con otro agradecimiento.

Calumnia

Opuesto al anterior. Si el usuario le escribe insultos, el agente también responde de forma apocada.

Despedida

Se asocia cuando el usuario se despide. Se considera el fin de la conversación.

Por defecto o fallback

Sólo se asocia si no existe ningún otro Intent que mejor represente las entradas. Significa que el agente no entendió lo que se le dijo o no está capacitado para resolver esa cuestión.

Planes

Este es un amplio conjunto de Intents que agrupan los intercambios que permiten al agente generar planes de servicio (Internet, Televisión y Telefonía) que se adapten a las necesidades del usuario. Estos Intents también le permiten al agente conocer la **cobertura** de la empresa, o sea, todas las provincias donde se ofrecen servicios. Se permite contratar, modificar, conocer y dar de baja un plan.

Ayudas

Conjunto de Intents que le ofrecen asistencia técnica al usuario. Están orientados a responder preguntas y solucionar problemas sobre el Internet, la televisión y telefonía. Es lo más cercano a un *preguntas frecuentes*.

Reclamos

Si los usuarios consideran que la asistencia del agente es insuficiente, se les permite generar órdenes de reclamos (o simplemente reclamos) detallando el problema, para un usuario determinado. Estos reclamos son pasados a la base de datos de la empresa para que sean tratados directamente por un empleado de soporte. Los usuarios pueden en todo momento consultar el estado de sus reclamos, para ver si estos han sido solucionados.

Terminologías

Es un conjunto similar al de ayudas, pero que no ofrece soporte técnico sino explicaciones sobre conceptos importantes de las redes, televisión y telefonía. Está pensando para resolver dudas sobre el significado de términos como *datos móviles*, *streaming*, *Mb/s*, etc.

2.2.2 • Contexts y prioridades

Como se mostrará luego en los resultados, se posee una gran flexibilidad durante la comunicación con el agente, lo que le permite al usuario escribir cosas como “3”, “Sí/No” y que esas entradas sean interpretadas bajo algún Intent en particular, como respuestas a preguntas anteriores.

Además, es posible iniciar conversaciones dando información total o parcial. Por ejemplo, el usuario puede escribir “tengo un problema” o “no me anda el Internet” y que ambos sean asociados con el mismo Intent (Ayudas). Esto se corresponde con el problema de los nodos hijos teniendo más exactitud que sus padres.

Todo esto es posible gracias al concepto de **Contexts**. Un intent puede recibir o generar contextos, los cuales se mantienen activos durante un tiempo. El tiempo se mide como un número de *rondas* (pares de preguntas/respuestas) de conversación. Mientras un

Contexts está activo, aquellos Intents que lo posee como entrada tienen muchas más posibilidades de ser asociados. Además, un Intent con un contexto de entrada, **no** puede ser asociado si ese contexto no está activo.

Esto permite que dos Intents diferentes con mismos Training Phrases resuelvan sus colisiones en base a los contextos que están activos. Esto también resuelve, parcialmente, el **encadenamiento hacia adelante**.

Un tipo especial de Context son los **Followup Contexts**. Estos son de corta duración, son generados por un Intent padre y se colocan como entrada de un Intent hijo (que también se conoce como **Followup Intent**). El objetivo es permitir a un usuario responder a una pregunta inmediatamente anterior y continuar con la respuesta. En el caso del “Sí/No”, este contexto le permite a un subsiguiente Intent hijo seguir el hilo de un Intent más general.

En nuestro agente existen muchos contextos que permiten una conversación fluida y natural con el usuario.

Como nota aparte, DialogFlow permite asignar una prioridad arbitraria a cada Intent, de manera que si existen colisiones, y todas las demás condiciones son iguales, aún así exista un modo de elegir uno sobre el otro.

2.2.3 • Parameters

Los **parámetros** son datos que se extraen de una entrada de usuario y se convierten a un tipo cómodo (estándar) para tratarlos programáticamente. Son similares a un Intent en el sentido de que también poseen un conjunto de palabras claves que los pueden matchear, pero su intención es usarlos como una variable para alguna respuesta.

Por ejemplo, debajo se muestra un parámetro *arg-province*. A la izquierda están los valores que puede tomar y a la derecha las palabras claves que matchean cada valor.

arg-province SAVE

☒ Define synonyms ☐ Regexp entity ☐ Allow automated expansion ☐ Fuzzy matching

Capital Federal	CABA, Capital, Capital Federal, Porteño, Ciudad Autónoma de Buenos Aires
Buenos Aires	Buenos Aires, BA
Catamarca	Catamarca
Chaco	Chaco
Chubut	Chubut
Córdoba	Córdoba, Cordoba
Corrientes	Corrientes
Entre Ríos	Entre Ríos, Entre Rios
Formosa	Formosa
Jujuy	Jujuy
La Pampa	La Pampa, Pampa

Si un usuario escribe “CABA”, “Porteño” o “Capital” todas esas entradas se asociarán con el valor “Capital Federal” lo que permite su uso cómodo.

Un Intent puede tener cero o más parámetros y estos pueden ser opcionales u obligatorios. Cuando un Intent se asocia, se busca si sus parámetros obligatorios fueron

también asociados. Si este no es el caso, se genera una respuesta en la que el agente le pide el usuario que escriba la información que le permita obtenerlo(s).

Los parámetros se asocian no solo a Intent, sino también a los contextos donde éstos se matchean. Un Intent posterior, dentro de ese contexto, puede acceder a ellos.

2.2.4 • Events

Existe un tipo especial de contexto, llamados **eventos**, que cuando están activos automáticamente matchean y ejecutan un Intent que los tenga como entrada. Los eventos se usan para cambiar el curso de la conversación de manera programática y también pueden tener parámetros. Se explicarán más adelante.

2.2.5 • Fulfillments

Una última pregunta que queda resolver es ¿qué tipo de respuesta debe entregar el agente?

Dialogflow ofrece la generación de respuestas *estáticas* o *customizadas*. Las estáticas estan ya precargadas desde la consola y pueden o no usar parámetros.

Las respuestas customizadas involucran aquellas que no se pueden saber de antemano usando solo las herramientas del programa. Son más complejas, pero muy útiles si se debe buscar información en una base de datos externa antes de continuar.

Para esto, el desarrollador debe crear un **servidor webhook** que funcione como API y acepte petición JSON provenientes de DialogFlow (*WebhookRequest*), donde se detalla el tipo de Intent matcheado junto con muchos datos más. El servidor procesa la petición y, en respuesta, envía un JSON (*WebhookResponse*) detallando la respuesta que se debe entregar al usuario, junto con otros datos más (por ejemplo, qué contexto o evento se debe activar).

Para que las respuestas de un Intent se obtengan por medio de un servidor externo, se debe activar la opción **Fulfillments** y escribir la dirección del servidor hacia donde enviar las peticiones POST.

Nuestro agente hace uso de una base de datos donde guarda la información de los planes, los usuarios y los reclamos. En consecuencia, también se tuvo que desarrollar el servidor y activarlo para algunos Intents. Los que no tienen esta opción activada usan respuestas estáticas.

2.2.6 • Integrations

Como se explicó al inicio de la sección, DialogFlow ofrece integración directa con varios programas de mensajería. Para el desarrollo de nuestro agente se decidió utilizar la integración con **Telegram**. Para ello, primero se debe crear un bot de Telegram siguiendo los pasos detallados en su página web (el bot aparecerá como un contacto con el que se puede chatear). Luego, Telegram genera un código o *token* que le permite a un programa externo acceder a una API para poder comunicarse con el bot creado.

Al entregarle este token a DialogFlow, este puede consultar la API para obtener la conversación con el usuario y devolverle respuestas. Todo el proceso es transparente para el desarrollador del agente. Las respuestas pueden ser texto plano, audio, video, cartas u otros tipos especiales soportados por el programa de mensajería (se conocen como *Rich Responses*).

2.3 • Comentario final

Debido a la complejidad del agente y las herramientas involucradas, y para no alargar demasiado el informe, se hizo un repaso rápido de las cuestiones más importantes. Para ver en detalle la implementación se puede importar el proyecto de DialogFlow y ver el repositorio en Github del servidor.

Tener en cuenta que el servidor **debe estar corriendo**, su dirección https pública establecida desde la consola, y los correspondientes Intents con la opción Fulfillment activada.

3.1 • Introducción

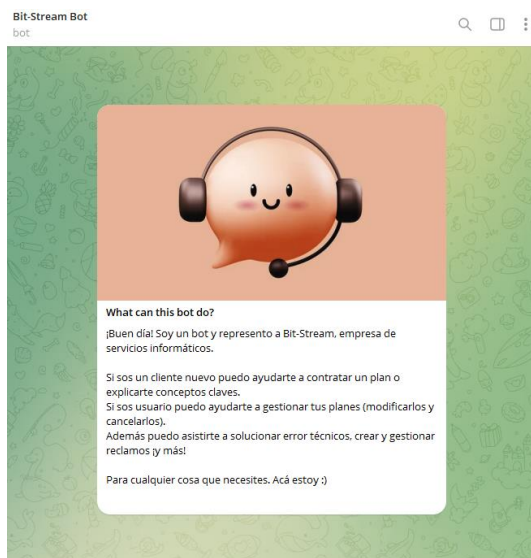
A continuación se muestran ejemplos de conversaciones reales con un usuario para ver cómo el agente reconoce las entradas, las asocia con sus Intents disponibles (descritos en la sección anterior) y genera respuestas.

Vale aclarar de que existen muchas formas diferentes de desencadenar cada acción, o sea, muchos tipos de entradas que el usuario puede escribir. Solo se muestran algunos casos, el resto se dejan para observar en la implementación.

3.2 • Conversaciones

Nota: Observará que a veces el agente no es capaz de entender los parámetros, incluso aunque el usuario escriba la palabra exacta que lo matchea. Esto es un bug de DialogFlow y se soluciona volviendo a escribir la palabra.

Presentación



Agradecimiento



Calumnias



Despedida

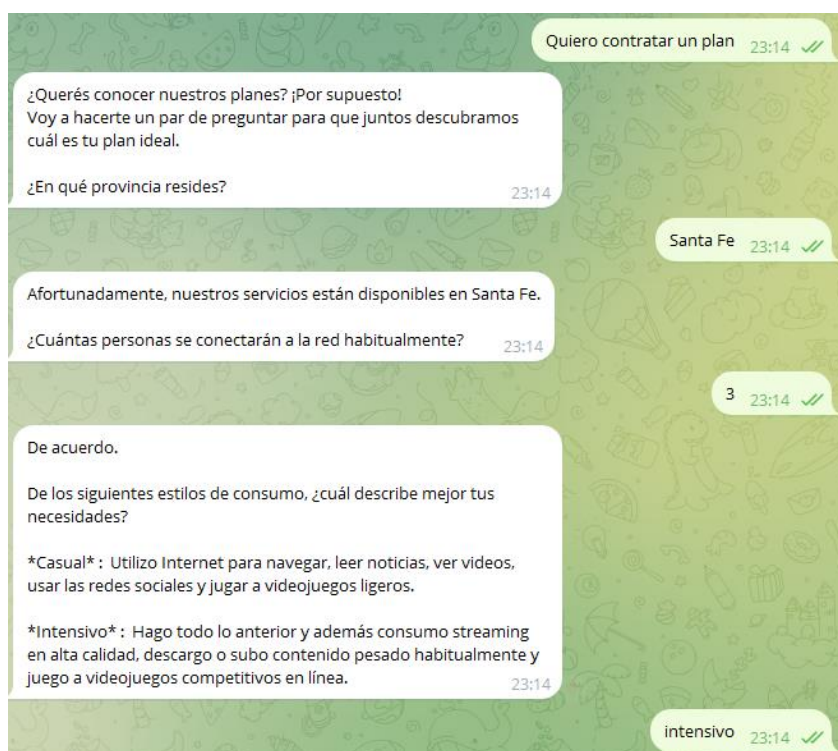


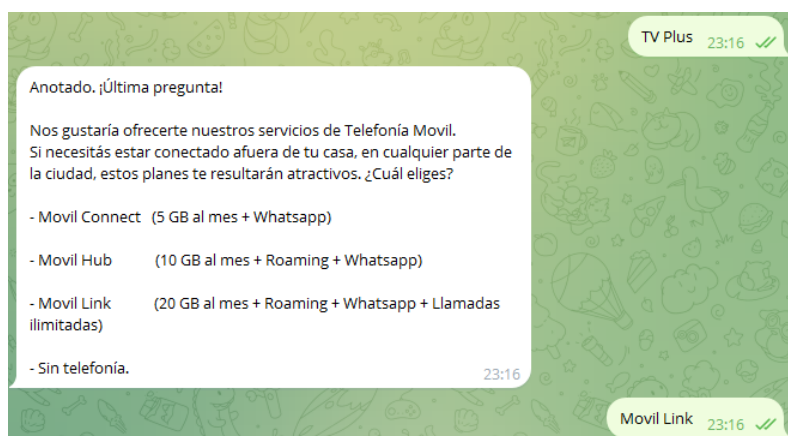
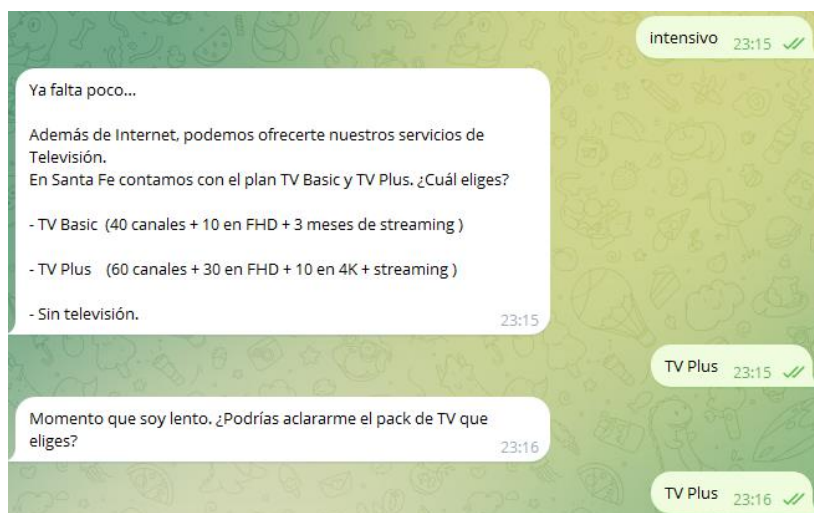
Por defecto o fallback

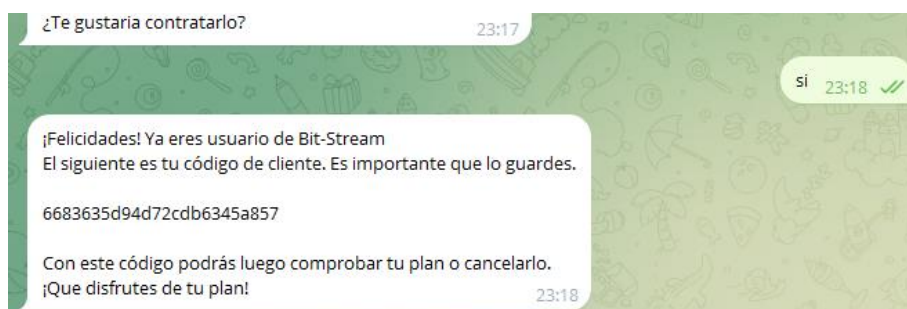
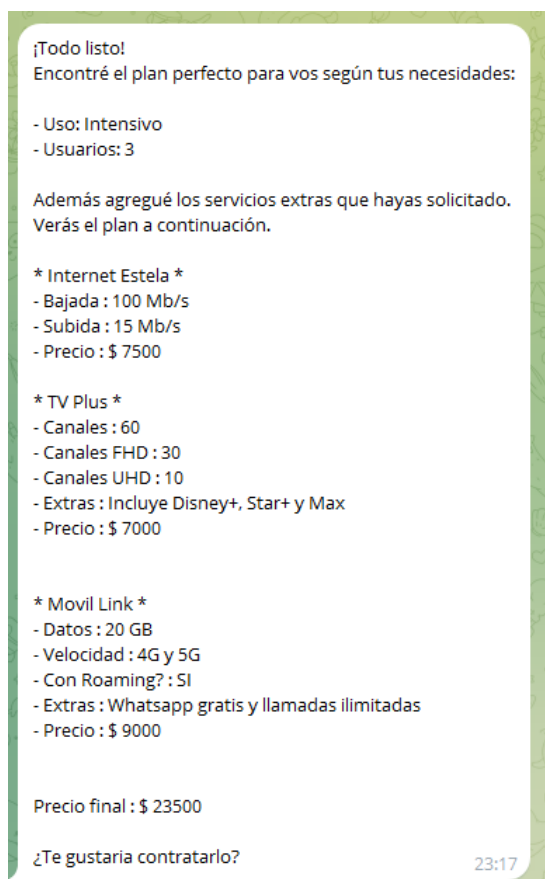


Planes

▪ Contratando un plan

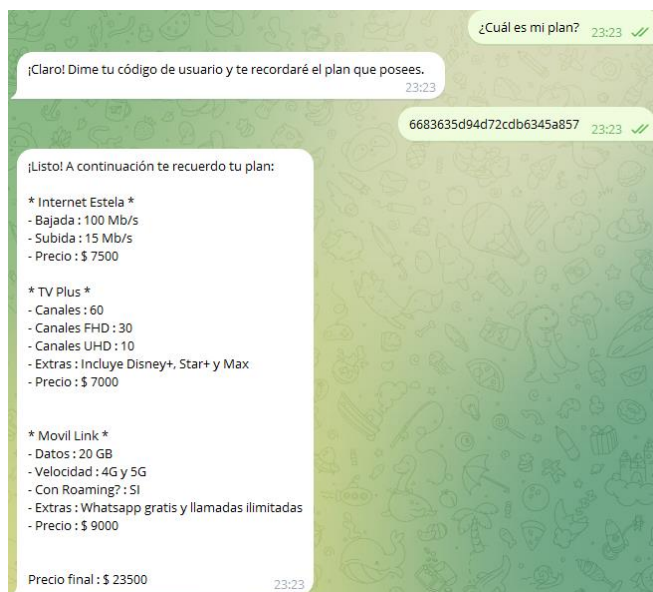






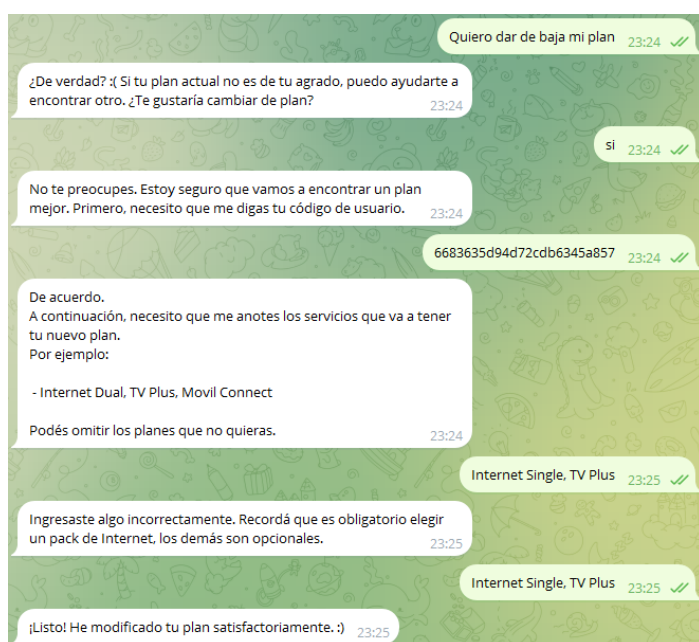
El agente guarda el plan generado en su base de datos. El cual puede luego ser consultado con el código: 6683635d94d72cdb6345a857.

- Ver mi plan



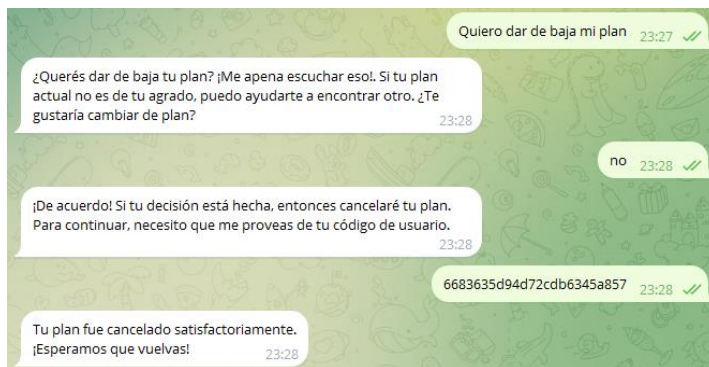
■ Cambiar de plan

Se procede amenazando con dar de baja el plan y el bot, como buen empleado de empresa de telefonía, te propone darte un mejor plan a cambio.



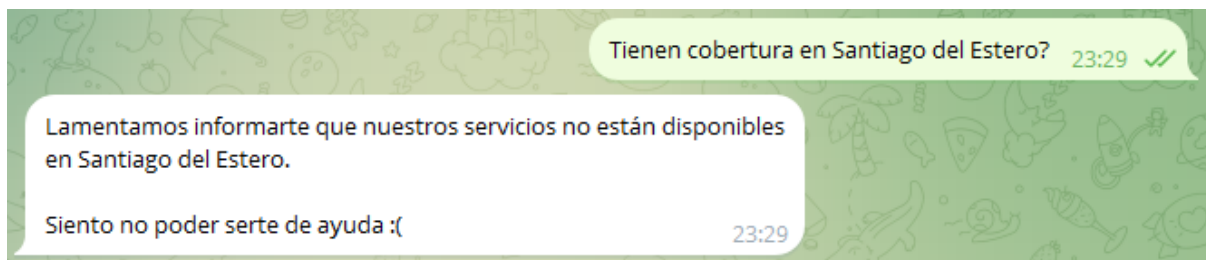
■ Dar de baja

Si en la conversación anterior elegimos decirle que “no”, el agente cancelará nuestro plan, quedando el código inutilizable.

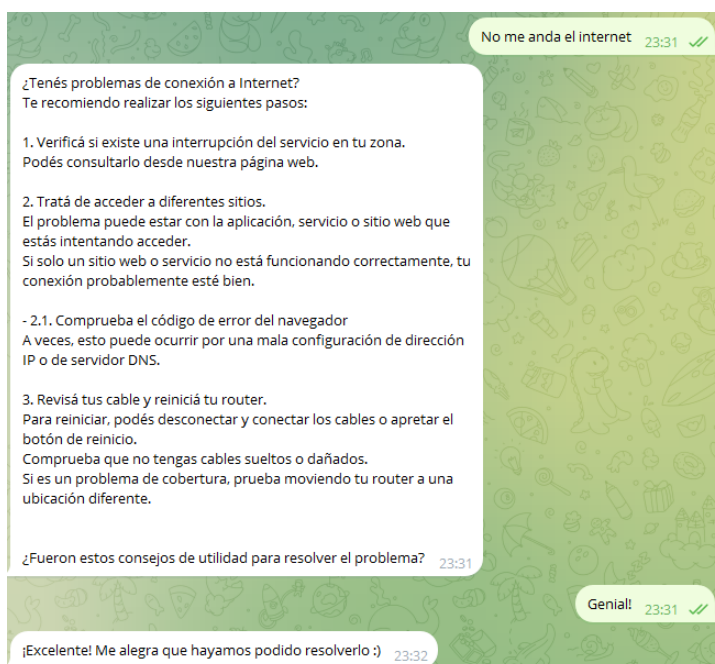


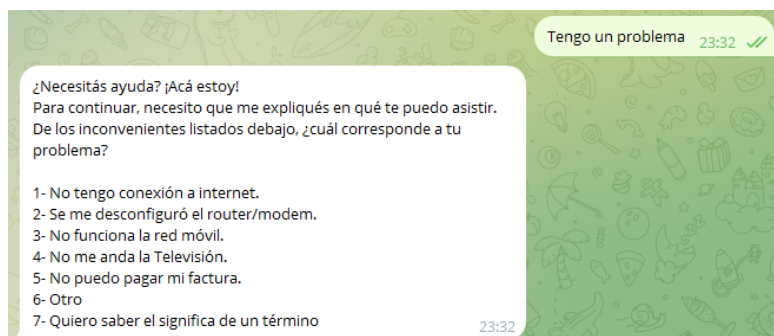
■ Cobertura

Como extra, se puede consultar la cobertura del servicio independientemente.



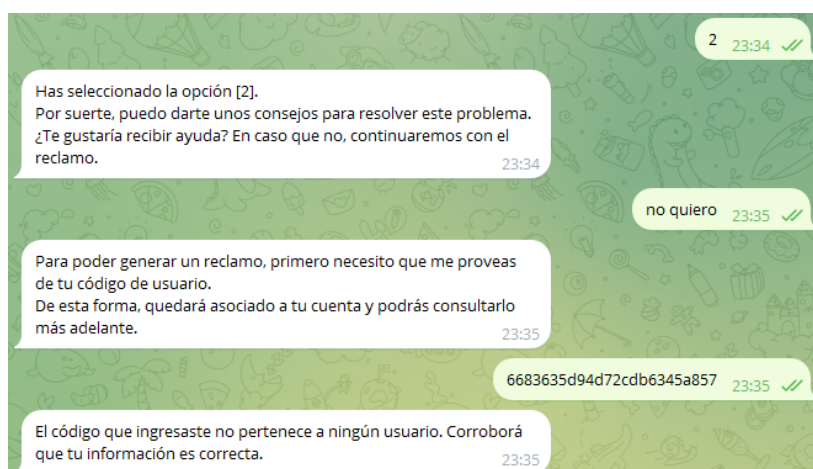
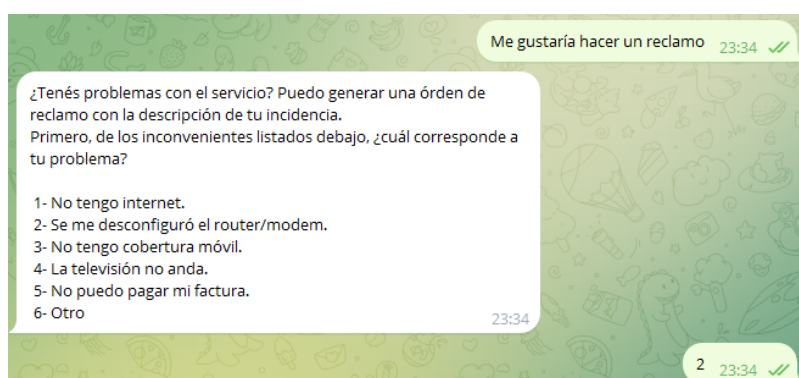
Ayudas



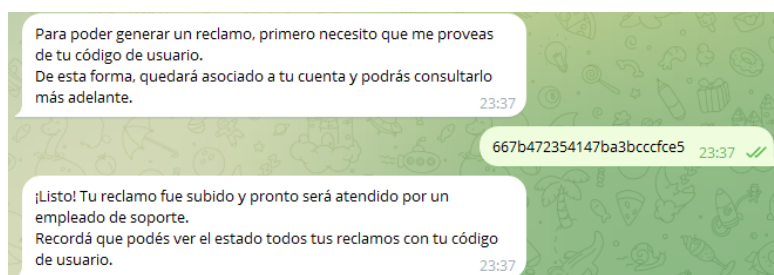


Reclamos

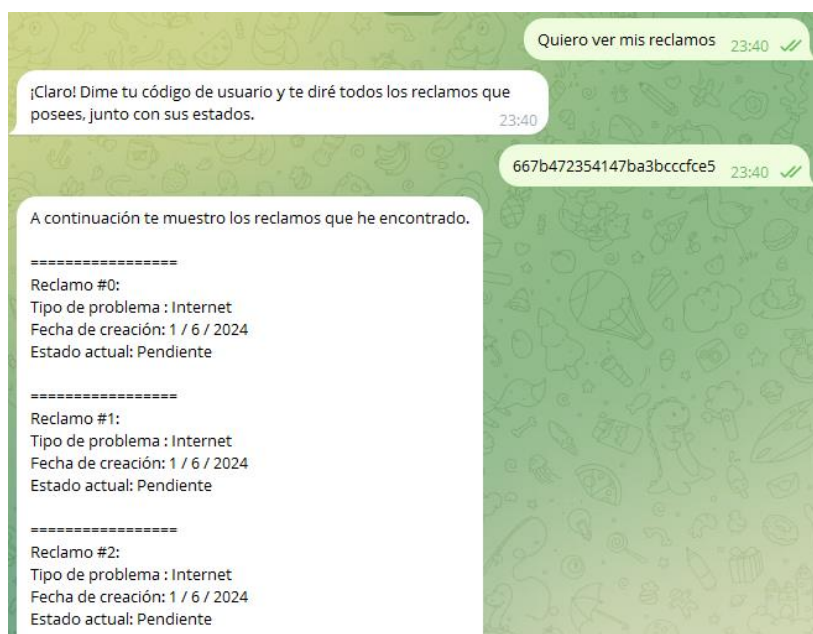
- Generar reclamos.



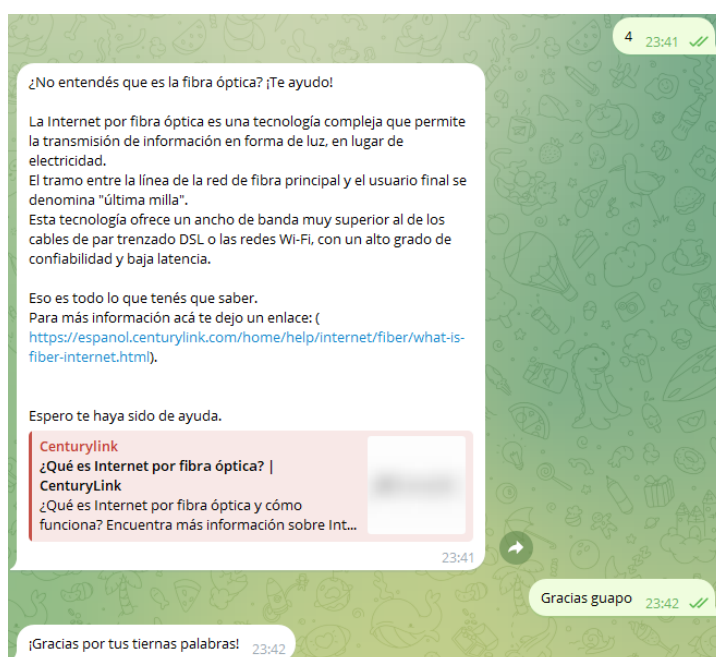
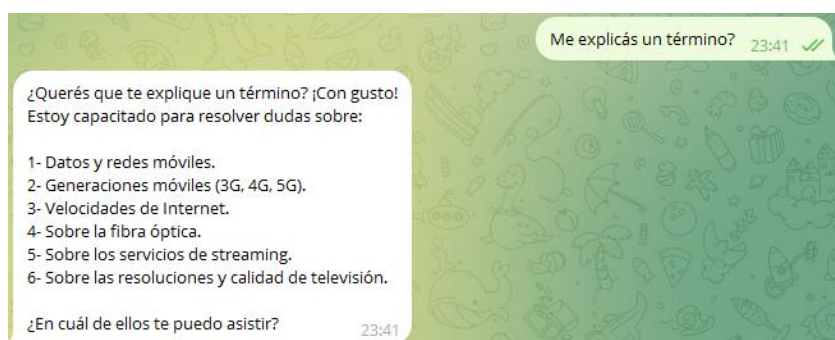
Observe que se trató de usar el código que se había dado de baja.



- Consultar reclamos

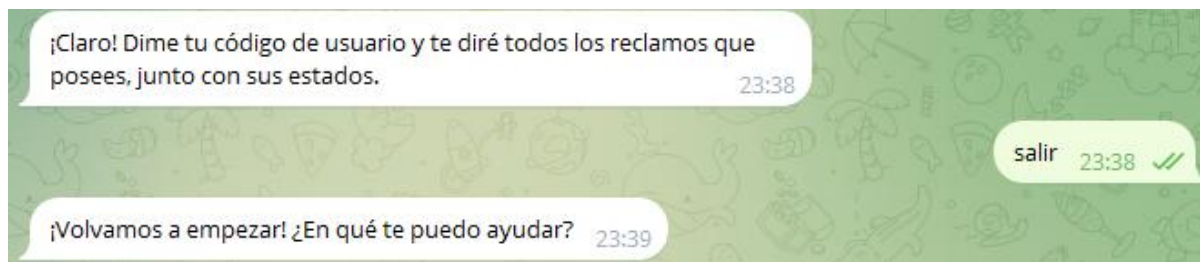


Terminologías



Abortar una acción

Si se quiere “reiniciar el bot”, se le puede pedir que aborte o termine la operación actual. Esto termina cualquier intercambio de parámetros que haya ocurrido en la conversación.



Usando la herramienta de DialogFlow, se pudo construir un agente robusto, con un lenguaje natural y que tiene una gran flexibilidad a la hora de manejar los parámetros que obtiene de la conversación. Se lograron aplicar los conceptos necesarios para el desarrollo del trabajo, y solucionar los problemas que se habían descubierto en la etapa temprana o conceptual.

Las mayores dificultades se encontraron a la hora de establecer el servidor y de pensar en el “flujo” del agente, asegurando que se mantenga el hilo de la conversación para distintos escenarios. Puede parecer que las conversaciones son simplemente lineales, pero el entender cómo manejar los contextos, eventos y trasladar los parámetros necesarios para que el agente los “recuerde” durante una cantidad arbitraria de Intents no es sencillo.

La conclusión final es que DialogFlow es una herramienta más que adecuada para el desarrollo de este tipo de agentes.