



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Chatbot accesible para el Area de Asuntos Estudiantiles (UNLP)

AUTORES: Navas Gago, Maximiliano - Navajas, Leandro

DIRECTOR: Lic. Diaz, Javier

CODIRECTOR: Mg. Harari, Ivana

ASESOR PROFESIONAL:

CARRERA: Licenciatura en Sistemas

Resumen

Esta tesina desarrolla un chat dotado con Inteligencia Artificial, destinado a aportar un nuevo mecanismo de comunicación. Entendiendo el escenario de la UNLP, optamos por una solución inclusiva, generando así un "chatbot accesible". Es importante entender que la solución es genérica y para esto ideamos una arquitectura de aplicación escalable, basada en módulos con sus respectivas responsabilidades. Una vez finalizada realizamos validaciones de accesibilidad y el testing de usuarios finales.

Palabras Clave

- Chatbot
- Inteligencia artificial
- Accesibilidad
- Natural Language Processing
- Wit
- Arquitectura de aplicaciones
- Lector de pantalla
- Facebook Messenger

Conclusiones

Con la implementación del chatbot buscamos aportar un nuevo mecanismo de comunicación a la UNLP, el cual pueda resolver la interacción de cualquier area y temática deseada en forma ágil y habilitando el acceso a todo usuario mas allá de las limitaciones funcionales, perceptivas o derivadas del contexto que posea.

Trabajos Realizados

*Análisis de mecanismos de comunicación de la UNLP.
Análisis de situación actual de chatbots.
Análisis y selección de las tecnologías apropiadas para la implementación del sistema.
Diseño y desarrollo de una arquitectura orientada a componentes.
Test de accesibilidad.
Test de usuarios finales.*

Trabajos Futuros

Comprendemos de utilidad los siguientes puntos:

- Manejo de usuarios del Backoffice: roles y permisos.
- Automatizar, mediante un moderador, el incremento de conocimiento del chatbot.

Fecha de la presentación: Octubre 2019

Agradecimientos

Esta tesis es el resultado de largos años de esfuerzo, trabajo y aprendizaje. Quiero expresar mi eterno agradecimiento hacia mis padres y mi hermana por haberme ayudado y apoyado incondicionalmente durante todos los años de la carrera, dándome fuerzas en los momentos más difíciles para aprender de los errores y seguir adelante.

Y un párrafo aparte para mi madre, quien a pesar de no poder estar físicamente en este momento tan importante, ha sido mi principal fuente de inspiración para nunca rendirme y cumplir con el objetivo de obtener el título.

Maximiliano Navas Gago

Agradezco a mis padres quienes me apoyaron durante toda la carrera incondicionalmente, ayudándome con sus consejos ante las adversidades. Agradezco a mis abuelos, quienes día a día me recordaban el verdadero valor de un título universitario. Agradezco a mi hermano, quien a pesar de las diferentes profesiones, siempre fue un gran compañero de largos días de estudio. A mi pareja, quien supo brindarme un espacio, y por último a mis amigos y compañeros los cuales fueron una compañía fundamental durante el transcurso de la carrera.

Leandro Navajas

Prólogo

A la hora de seleccionar un tema para la Tesina de grado, analizamos varios conceptos, pero el foco principal siempre fue realizar un APORTE. Basándonos en esto analizamos diversos temas, debatiendo por varias semanas hasta dar con el indicado (desde nuestro punto de vista).

Ya en nuestros últimos pasos en la carrera, decidimos pensar como ingresantes, y analizamos los faltantes y limitantes recordados de aquella instancia. Algunos de estos ya se habían solucionado, pero a su vez, dimos con otros los cuales desde nuestro punto de vista aún no se han abordado. Así llegamos al tema de la Comunicación, sin recaer en la falta de información, ya que esto no lo vemos como un problema. Identificamos como problemática el acceso a la información.

Realizamos varios análisis y nos topamos con otro gran problema: la descentralización de los datos. Este, si bien era un buen tema a tratar, no reflejaba nuestro interés, el cual radica en realizar un aporte que tenga como beneficiario directo al estudiante.

Decidimos enfocarnos en el acceso a la información, debido a que nuestra facultad recibe cientos de ingresantes del interior. Nuestra idea principal era que puedan quitarse las dudas sin tener que venir hasta La Plata.

Aquí nació el proyecto de implementar un Chatbot, el cual responda todas las dudas de los futuros ingresantes a través de un chat como el de Facebook, en el que hoy en día, una gran cantidad de personas poseen una cuenta, dándonos así un mayor alcance. Además el servicio es gratuito a diferencia de otras herramientas de mensajería masivas como Whatsapp.

Independientemente de esto, nuestra idea no es atarlo al chat de dicha plataforma, brindando la posibilidad de en un futuro poder integrarlo con otras aplicaciones de mensajería.

Otra gran ventaja que nos brinda Facebook Messenger es la Accesibilidad. La empresa promueve el acceso a la información para todas las personas, independientemente de las discapacidades de cada una. Basándose en el lema:

“La misión de Facebook es unir al Mundo, y esto significa a todos. Nuestro objetivo es hacer posible que cualquier persona, independientemente de su capacidad, tenga acceso a la información y las conexiones que ocurren en Facebook.”

Con este panorama, ya teníamos aspectos cerrados como:

- El aporte
- El alcance
- La tecnología/herramienta

Esto fue lo que nos motivó a acercar nuestra propuesta a distintas autoridades de la facultad. Nuestra idea generó más interés del que augurábamos en un principio, lo que derivó en las siguientes preguntas:

¿por qué acotarnos sólo a la Facultad de Informática?, ¿por qué no hacer más extenso el alcance?

En base a esto, situamos nuestra propuesta en un alcance aún mayor: la Universidad Nacional de La Plata. Aquí apareció otro aspecto muy importante, el cual nos terminó dando un cambio de rumbo: el Área de Bienestar Universitario de la UNLP.

En este punto queremos aclarar que nuestra idea no es acotar el Chatbot a un único área. Nuestro principal objetivo es que sea genérico. Amén de esto, por un tema de tiempos y contenidos, lo instanciamos¹ sobre el área de Asuntos Estudiantiles para mostrar un ejemplo de su funcionamiento y las ventajas que este provee.

¹ Una instancia (en inglés, instance) es la particularización, realización específica u ocurrencia de una determinada entidad.

Índice

Introducción	9
Definición	10
Historia	10
Inteligencia Artificial	11
Accesibilidad	13
Accesibilidad web	14
WCAG 2.1	15
Facebook	15
Lectores de pantalla	15
Talk Back (Android)	16
Voice over (iOS)	16
NVDA (Windows)	16
JAWS (Windows)	17
Herramienta para evaluar la accesibilidad web del chatbot	17
Wave	17
Conclusión	18
Chatbots aplicados en ámbitos universitarios	18
Chatbot Lola	19
Chatbot Beacon	19
Chatbot Lucy	20
Chatbot Diario Clarín	20
Asuntos estudiantiles	21
Situación Actual	21
Solución Planteada	23
Aspectos Técnicos	24
Arquitectura	24
Paso a paso	25
Estructura de la API	36
Responsabilidades de cada nivel	37
Controller	37
Facade	37
Operation	37
Service	38
Client	38
Ventajas	38
Tecnologías	38
Frontend	38
Angular	38
Backend	40

Java	40
Maven	40
Spring	40
Jetty	41
Api externa	41
NLP (Natural Language Processing)	41
Arquitectura de un sistema de NLP	42
Wit	42
Base de datos	43
Cassandra	43
Aplicación	43
Funcionalidad	43
Interfaz del Usuario	43
Diseño del Back office	44
Diseño del frontend del chatbot	51
Testing	60
Pruebas de usabilidad	62
Resultados	67
Encuesta: Tomás Falco	76
Setup	82
Conclusiones y Trabajo Futuro	84
Referencias	86

Introducción

Partimos de la idea que cuando hablamos de chatbots, indirectamente estamos haciendo mención a la Inteligencia Artificial, la cual puede definirse como:

"La Inteligencia Artificial es el estudio de los agentes que reciben percepciones del entorno y llevan a cabo las acciones". (Russell y Norvig. 2004)

"A su vez Inteligencia Artificial es el estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor". (Rich y Knight. 1991)

Para el desarrollo de nuestra tesina decidimos basarnos en estas definiciones, considerando que si las llevamos al ámbito de la comunicación que se establece hoy en día entre la Universidad Nacional de La Plata con diferentes usuarios, se percibe fácilmente que dicha institución no cuenta con una herramienta que pueda brindar respuestas automáticas y rápidas a las consultas que le llegan.

El objetivo de esta tesina fue desarrollar un chat dotado con Inteligencia Artificial, destinado a solucionar la problemática anteriormente planteada.

Esta herramienta cuenta con una interfaz que permite cargar consultas y respuestas al chatbot, aumentando de esta manera su inteligencia y su capacidad de evacuar las dudas del usuario sin que éste se frustre. A su vez, se brinda accesibilidad para usuarios con discapacidad.

Para la aplicación nos pareció interesante el área de Asuntos Estudiantiles de la UNLP ya que abarca a toda la comunidad estudiantil de la Universidad. Esto es un desafío muy importante por los distintos perfiles a tener en cuenta, considerando diferentes niveles de formación que va desde ingresantes hasta alumnos avanzados, conocimientos disciplinares disímiles que van desde ciencias duras a sociales, como también por la diversidad de la población estudiantil en cuanto a sus condiciones socio económicas, culturales, formación previa como por limitaciones o discapacidad que puedan poseer.

Definición

Según el Oxford English Dictionary, el concepto de chatbot se define de la siguiente manera:

"Programa de computadora diseñado para simular la conversación con usuarios humanos, especialmente a través de internet."

En la literatura científica, los chatbots se conocen más formalmente como agentes conversacionales.

El principio subyacente de cada chatbot es interactuar con un usuario humano (en la mayoría de los casos) a través de mensajes y comportarse como si fueran capaces de comprender la conversación, respondiendo al usuario de manera adecuada.

El origen de la conversación entre humanos y computadoras es tan antiguo como el campo de la ciencia informática. En efecto, Alan Turing definió una prueba simple llamada ahora "Test de Turing" en 1950, donde una persona tendría que predecir si la entidad que está respondiendo el comunicado es una persona o un programa informático. Sin embargo, este test es mucho más ambicioso que el uso habitual de los chatbots, la principal diferencia es que el conocimiento del dominio de un chatbot es limitado, mientras que el Test de Turing asume que se puede hablar sobre cualquier tema con el agente. Esto ayuda durante el diseño de conversación del chatbot, ya que no está obligado a tener un conocimiento de dominio (potencialmente) infinito y, como tal, puede centrarse en ciertos temas muy específicos como, por ejemplo, ayudar a los usuarios a reservar una mesa en un restaurante.

Además, otro supuesto general en el diseño de los chatbots es tener en cuenta que los usuarios suelen tener un objetivo que desean alcanzar al final de la conversación, cuando inician una interacción con un agente. Esto influye entonces en el flujo y los temas de la conversación para lograr el objetivo elegido. Esto puede ser explotado por los desarrolladores ya que, como resultado, tienden a surgir ciertos patrones de comportamiento.

Historia

La primera instancia de un agente de conversación nació en 1966: ELIZA. Se trataba de un programa de computadora que parodiaba al psicólogo Carl Rogers y reformulaba las opiniones de los usuarios utilizando técnicas básicas de procesamiento de lenguaje natural (según los estándares de hoy). A pesar de ser relativamente simple, el programa logró dar la ilusión de comprender los problemas del usuario y engañó con éxito a muchas personas. Su creador, Joseph Weizenbaum, mencionó que hasta su secretaria le ha pedido que dejara la habitación para poder tener una conversación privada con ELIZA, confiando más en el programa que en el propio Weizenbaum.

Luego de varias décadas, los chatbots siguieron, en gran medida, el enfoque de ELIZA, aunque con menores adiciones introducidas, como la síntesis del habla y el manejo de las emociones. En 2001 apareció SmarterChild, un agente de conversación desarrollado por ActiveBuddy, Inc. (ahora Colloquis) que operaba en AOL Instant Messenger y MSN Messenger. Inspirado por el auge de las plataformas de mensajería instantánea como SMS, SmarterChild se creó para proporcionar un acceso rápido a noticias, pronósticos meteorológicos, resultados deportivos, entre otros. La principal innovación fue que SmarterChild se conectaba a una base de conocimientos y mantenía información útil para sus usuarios. Desafortunadamente, las limitaciones técnicas del procesamiento del lenguaje natural se encontraron con los robots en esas plataformas y fueron olvidadas por la historia.

El siguiente avance para los agentes de conversación lo realizó un equipo de IBM, a través del proyecto Watson AI, que ha estado en desarrollo desde 2006. El agente fue diseñado

con el único propósito de ganar el programa de televisión estadounidense Jeopardy!, objetivo que consiguió en 2011, cuando competía contra dos de los ex campeones del programa.

Jeopardy! es interesante desde el punto de vista del Procesamiento del Lenguaje Natural (sus siglas en inglés, NLP), ya que las preguntas involucran mucho juego de palabras y requieren una rápida recuperación de la información en vastas bases del conocimiento. Desafortunadamente, la Inteligencia Artificial en su forma anterior solo pudo responder a preguntas de una sola línea y no pudo mantener una conversación adecuada con otra persona.

Finalmente, a principios de 2010 surgió el aumento de asistentes virtuales como Siri de Apple, Cortana de Microsoft, el asistente de Google, Alexa de Amazon y otros. Estos agentes introdujeron los conceptos de conversación y diálogo orientado a objetivos. Otro evento importante en el campo de los chatbots fue el lanzamiento de la Plataforma Messenger para Facebook Messenger en 2016, que permitió la creación de agentes conversacionales para empresas no relacionadas con la Inteligencia Artificial.

Inteligencia Artificial

Dada la dificultad para precisar el propio concepto de "inteligencia", existe prácticamente una definición de Inteligencia Artificial por cada autor que escribe sobre el tema. En cualquier caso, desde distintas perspectivas, en el seno de la Inteligencia Artificial como ciencia y tecnología, se han ido acumulando conocimientos sobre cómo emular las diversas capacidades del ser humano para exhibir comportamientos inteligentes y se han desarrollado sistemas cada vez más perfeccionados que reproducen parcialmente dichas capacidades.

“La Inteligencia Artificial también llamada inteligencia computacional, es la inteligencia exhibida por máquinas. En ciencias de la computación, una máquina “inteligente” ideal es un agente racional flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea.” (Pool, David. 2018)

“Coloquialmente, el término inteligencia artificial se aplica cuando una máquina imita las funciones “cognitivas” que los humanos asocian con otras mentes humanas, como por ejemplo: “aprender” y “resolver problemas”.” (Russell, Stuart y Norvig, Peter. 2009)

“A medida que las máquinas se vuelven cada vez más capaces, la tecnología que alguna vez se pensó que requería de inteligencia, se elimina de la definición. Por ejemplo, el reconocimiento óptico de caracteres ya no se percibe como un ejemplo de la “inteligencia artificial” habiéndose convertido en una tecnología común.” (Schank, Roger. 1991)

Avances tecnológicos todavía clasificados como inteligencia artificial son los sistemas de conducción autónomos o los capaces de jugar al ajedrez o al Go.

Según Takeyas (2007) *“la IA es una rama de las ciencias computacionales encargada de estudiar modelos de cómputo capaces de realizar actividades propias de los seres humanos en base a dos de sus características primordiales: el razonamiento y la conducta.”*

En 1956, John McCarthy acuñó la expresión “Inteligencia Artificial”, y la definió como *“la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes”*.

Para Nils John Nilsson son cuatro los pilares básicos en los que se apoya la inteligencia artificial:

- Búsqueda del estado requerido en el conjunto de los estados producidos por las acciones posibles.
- Algoritmos genéticos (análogo al proceso de evolución de las cadenas de ADN).
- Redes neuronales artificiales (análogo al funcionamiento físico del cerebro de animales y humanos).
- Razonamiento mediante una lógica formal análogo al pensamiento abstracto humano.

También existen distintos tipos de percepciones y acciones, que pueden ser obtenidas y producidas, respectivamente, por sensores físicos y sensores mecánicos en máquinas, pulsos eléctricos u ópticos en computadoras, tanto como por entradas y salidas de bits de un software y su entorno software.

Varios ejemplos se encuentran en el área de control de sistemas, planificación automática, la habilidad de responder a diagnósticos y a consultas de los consumidores, reconocimiento de escritura, reconocimiento del habla y reconocimiento de patrones. Los sistemas de Inteligencia Artificial actualmente son parte de la rutina en campos como la economía, la medicina, la ingeniería y la milicia, y se ha usado en gran variedad de aplicaciones de software, juegos de estrategia (como ajedrez de computador) y otros videojuegos.

Accesibilidad

Según los comentarios de Comes (2006), autor de *“El concepto de accesibilidad: La perspectiva relacional entre población y servicios”*, indica que *“El concepto de accesibilidad ha sido muy utilizado desde los años 60. Fue en el momento en que se acordó la necesidad de concretar el desarrollo social de los países latinoamericanos. A partir de ahí se planteó el crecimiento por áreas, siendo la salud una de las más relevantes para concretar esta meta. Esta preocupación por el “desarrollo social”, partía de la base de que sería necesario crédito externo y tiempo para poder igualar a todos los países del mundo. La preocupación por el avance del socialismo fue uno de los motivos que precipitaron estas políticas.*

La accesibilidad fue definida como la forma en que los servicios de salud se acercarían a la población. Este concepto fue entendido como un problema de la oferta y sería necesario, desde esta óptica, eliminar las barreras que se pudieran interponer. Las barreras fueron caracterizadas como:

- *Geográficas: se entendía que la imposibilidad de acercar los servicios a la población era por un problema geográfico, ya sea a un accidente geográfico, o una barrera construida por el hombre (por ejemplo, una autopista).*

- *Económicas: aludían a la imposibilidad de acceso por motivos de falta de dinero, tanto para el transporte como, por ejemplo, para comprar medicamentos.*
- *Administrativas: expresaban la dificultad que impone la organización misma de los servicios, por ejemplo los horarios de atención o los turnos.*
- *Cultural: se centraban las diferentes culturas que existían entre el personal de los servicios y la población, que podían ser una barrera de acceso. “ (Comes, Y. 2006)*

El profesor Samaja la define como *“una barrera simbólica en tanto consideremos que el imaginario social y las representaciones también pueden constituirse en barreras. El problema de definir a la accesibilidad desde la oferta es que queda invisibilizado el hecho de que los sujetos también son constructores de accesibilidad”.*

Definir la accesibilidad como el encuentro entre los sujetos y los servicios incorpora una dimensión que obliga a pensar este concepto desde otra perspectiva. Si se considera que se trata de un problema de oferta, las representaciones, discursos y prácticas de la población no son necesariamente considerados por los servicios a la hora de dar respuestas para mejorar la accesibilidad.

Al pensarlo como un problema de encuentro/desencuentro entre la población y los servicios de salud, es ineludible la inclusión de las prácticas de vida y salud de la población en cada estrategia para mejorar el acceso. En este sentido, se puede definir a la accesibilidad como una relación entre los servicios y los sujetos en la que, tanto unos como otros, contendrían en sí mismos la posibilidad o imposibilidad de encontrarse.” (Almeida, C. 2000)

Esto nos lleva a considerar el concepto en su complejidad, incluyendo en su análisis su dimensión relacional. Se hace necesario pensar en la población, en los servicios de salud y en la relación entre los servicios y la población en tanto los servicios son productores de discursos que se entran en representaciones y prácticas de la población, dando como resultado distintos modos de acercamiento y utilización. Dicho esto, definimos accesibilidad como el vínculo que se construye entre los sujetos y los servicios.

“Este vínculo se construye a partir de la combinatoria entre las condiciones y discursos de los servicios y las condiciones y representaciones de los sujetos y se manifiesta en la modalidad particular que adquiere la utilización de los servicios”. (Stolkiner y Otros. 2000)

Esta definición parte de la consideración de Bourdieu que plantea que las ciencias sociales son ciencias relacionales. (Bourdieu P. y Wacquant L. 1995)

Accesibilidad web

Podemos definir la accesibilidad Web como la posibilidad de que un producto o servicio web esté a disponibilidad y pueda ser usado por el mayor número de personas, indiferentemente de las limitaciones propias del individuo o de las derivadas del contexto de uso.

En la definición, “las limitaciones propias del individuo” no solo engloban aquellas representadas por discapacidades, sino también otras como pueden ser el idioma, conocimientos o experiencia.

Además, la accesibilidad no sólo implica la necesidad de facilitar acceso, sino también la de facilitar el uso. La distinción entre usabilidad, facilidad de uso, y accesibilidad, como indica Henry no solo es difícil, sino en muchos casos innecesaria.

“Un diseño será accesible cuando sea usable para más personas en más situaciones o contextos de uso” (Henry, Shawn Lawton. 2003)

WCAG 2.1

Las Pautas de Accesibilidad al Contenido Web (WCAG) 2.1 cubren una amplia gama de recomendaciones para hacer que el contenido web sea más accesible. Seguir estas pautas hará que el contenido sea más accesible para una gama más amplia de personas con discapacidades, incluidas las adaptaciones para ceguera y baja visión, sordera y pérdida de audición, movimientos limitados, discapacidades del habla, fotosensibilidad y combinaciones de estas, y algunas adaptaciones para discapacidades de aprendizaje y limitaciones cognitivas; pero no abordará las necesidades de cada usuario para las personas con estas discapacidades. Estas pautas abordan la accesibilidad del contenido web en computadoras de escritorio, computadoras portátiles, tabletas y dispositivos móviles. Seguir estas pautas también hará que el contenido web sea más usable para los usuarios en general.

Facebook

Como hemos comentado, Facebook se rige bajo el lema: *“La misión de Facebook es unir al Mundo, y esto significa a todos. Nuestro objetivo es hacer posible que cualquier persona, independientemente de su capacidad, tenga acceso a la información y las conexiones que ocurren en Facebook.”*, mediante el cual expresa su interés por la accesibilidad en todas sus plataformas. A su vez posee cuentas oficiales para tratar los temas relacionados a accesibilidad, en donde se brinda información y actualizaciones que Facebook va realizando para mejorar la experiencia de la gente con alguna discapacidad en cualquiera de sus plataformas.

Además son espacios abiertos y de fácil acceso para que cualquier usuario pueda reclamar por alguna funcionalidad que todavía no existe o mejorar algo que ya se haya implementado.

Los canales de comunicación oficiales específicamente de accesibilidad que brinda Facebook son:

- Pagina de Facebook: <https://www.facebook.com/accessibility>
- Cuenta oficial de Twitter: <https://twitter.com/fbaccess>

Además Facebook posee documentación con un apartado de Accesibilidad (https://es-la.facebook.com/help/273947702950567/?helpref=hc_fnav), en el cual uno de los incisos está relacionado con Messenger y se brindan respuestas a preguntas frecuentes sobre accesibilidad en dicha plataforma de mensajería.

Más adelante en este capítulo abordaremos evaluaciones cuyos resultados respaldarán el compromiso que Facebook afirma tener con la accesibilidad en la plataforma que nos interesa para nuestra tesina, Facebook Messenger.

Lectores de pantalla

Con el objeto de proveer accesibilidad como una característica fundamental del chatbot, es necesario analizar las herramientas de apoyo que utilizan las personas con discapacidad.

Los lectores de pantalla (“*screen readers*” en inglés) son un software que permite la utilización del sistema operativo y las distintas aplicaciones mediante el empleo de un sintetizador de voz que “lee y explica” lo que se visualiza en la pantalla, lo que supone una ayuda para las personas con graves problemas de visión o completamente ciegas.

Talk Back (Android)

Es el lector de pantalla de Google incluido en los dispositivos Android. Esta función emite comentarios por voz para que puedas usar el dispositivo sin mirar la pantalla.

Elegimos mencionar este lector de pantalla desarrollado por Google para el sistema operativo Android, porque durante la coordinación de las pruebas de usabilidad con personas ciegas, ellas nos comentaron que usaban este lector de pantalla en sus dispositivos con sistema operativo Android.

Otro de los motivos de la elección es que Facebook, en su documentación de accesibilidad, lo recomienda en caso de utilizar Android.

Voice over (iOS)

VoiceOver es un lector de pantalla basado en gestos que te permite disfrutar del iPhone aunque no puedas ver la pantalla. Con VoiceOver activado, sólo tienes que hacer clic tres veces en el botón de inicio para acceder a esta funcionalidad desde cualquier lugar de iOS. Escucharás una descripción de todo lo que hay en tu pantalla, desde el nivel de la batería hasta quién te está llamando o el nombre de la app sobre la que tienes puesto el dedo. Puedes ajustar la velocidad de habla e incluso elegir el tono de voz.

El lector de pantalla por excelencia de iOS tiene muchos problemas para leer las respuestas rápidas de Facebook Messenger, como se puede ver en el siguiente video subido en Youtube https://www.youtube.com/watch?time_continue=67&v=x2v4S7MoqRs. Con lo cual no recomendamos a las personas ciegas o con problemas severos de visión utilizar la aplicación Facebook Messenger con un Iphone.

NVDA (Windows)

NVDA (Non Visual Desktop Access) es un lector de pantalla libre y gratuito desarrollado por NVAccess que permite a las personas ciegas y con discapacidad visual usar ordenadores. Para ello lee el texto que se muestra en pantalla mediante una voz sintética. Se puede controlar lo que NVDA lee moviendo el cursor al área relevante que contiene el texto, tanto poniendo el ratón encima como usando las flechas del teclado.

Luego de realizar las pruebas de usabilidad con personas ciegas, llegamos a la conclusión que este lector de pantalla fue el más utilizado por ellas quienes usaban en su computadora de escritorio el sistema operativo Windows.

Además, en la documentación sobre accesibilidad de Facebook también lo recomiendan, en caso de tener un sistema operativo Windows.

JAWS (Windows)

JAWS, Job Access With Speech, es el lector de pantalla más popular del mundo, desarrollado para usuarios de computadoras cuya pérdida de visión les impide ver el contenido de la pantalla o navegar con un mouse. JAWS proporciona salida de voz y Braille para las aplicaciones de computadora más populares en su PC. Podrá navegar por Internet, escribir un documento, leer un correo electrónico y crear presentaciones desde su oficina, escritorio remoto o desde su casa.

La elección de incorporar este lector de pantalla como ejemplo de nuestra tesis de grado, se debió a la popularidad que tiene dentro de los regularmente utilizados para el sistema operativo Windows. A su vez en la documentación sobre accesibilidad de Facebook también lo recomienda junto a NVDA.

Herramienta para evaluar la accesibilidad web del chatbot

Wave

WAVE es una herramienta para ayudar a los desarrolladores web a hacer más accesible su contenido web. WAVE no puede decirle si su contenido web es accesible. Sólo un humano puede determinar la verdadera accesibilidad. Pero, WAVE puede ayudarlo a evaluar la accesibilidad de su contenido web.

Como especifica la documentación de esta herramienta, no nos puede dar la certeza de que la aplicación Facebook Messenger es 100% accesible. Lo que sí pudimos determinar es que la evaluación fue positiva y, por consiguiente, puede ser analizada perfectamente bien por lectores de pantalla.

A continuación, mostraremos una captura de pantalla en donde ejecutamos una evaluación sobre la versión de escritorio de Facebook Messenger en donde estaba abierto un chat producto de una conversación con nuestro chatbot (Fig. 1.1). En el mismo Wave analizó tanto el texto plano como las respuestas rápidas.

Como muestra la figura los resultados de la evaluación arrojados en el panel lateral izquierdo, sólo aparecieron 5 errores y no están sobre el flujo de la conversación, en donde todos los tags pasaron la evaluación satisfactoriamente (incluso las respuestas rápidas).

Los errores se encuentran sobre el historial de chats abiertos en Facebook (columna izquierda) y sobre los detalles del chat que se encuentra abierto (columna derecha).

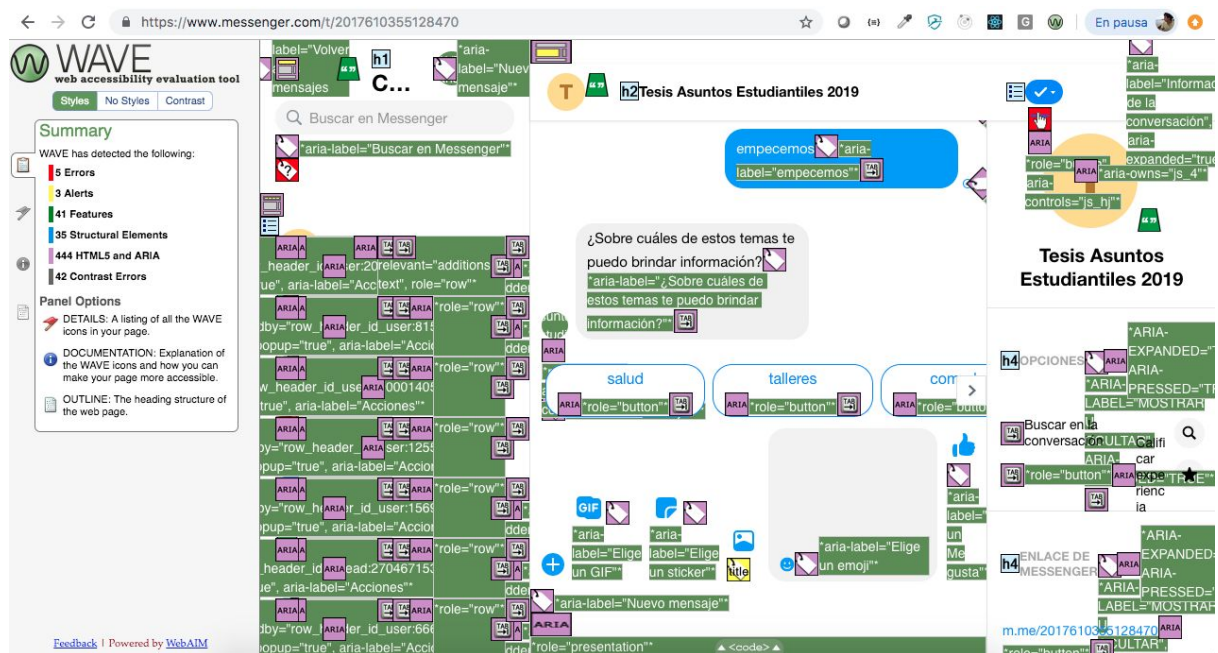


Figura 1.1

Luego de leer la documentación sobre la accesibilidad de las aplicaciones Facebook y Facebook Messenger, sumado al resultado positivo que obtuvimos ejecutando la herramienta de evaluación de accesibilidad web (Wave) sobre nuestro chatbot, entendemos que se cumplen con creces las WCAG (Web Content Accessibility Guidelines) más importantes para que personas con diferentes discapacidades puedan interactuar sin inconvenientes.

Más tarde, durante el desarrollo de la presente tesis de grado, se mostrarán las pruebas realizadas para afirmar o refutar dicha hipótesis.

Respecto de la accesibilidad de Facebook Messenger con los diferentes lectores de pantalla utilizados en diversos sistemas operativos, llegamos a la conclusión que para dispositivos móviles lo ideal es contar con Android, ya que Talk Back funciona perfectamente. Y, en caso de interactuar usando la versión de escritorio, lo mejor es tener un sistema operativo Windows porque tanto JAWS como NVDA también hacen muy bien su trabajo.

Por el contrario, no recomendamos hacer uso de la aplicación tanto móvil como de escritorio en un sistema operativo iOS, ya que el lector de pantalla VoiceOver no cumple con las expectativas suficientes de los usuarios ciegos o con problema severos de visión.

Chatbots aplicados en ámbitos universitarios

En la siguiente sección describiremos casos actuales de chatbots en ambientes universitarios.

Chatbot Lola

En Julio de 2018, la Universidad de Murcia y la Universidad Politécnica de Cartagena presentaron a Lola, un chatbot creado para guiar y ayudar a los futuros estudiantes con

todos sus trámites de preinscripción y matrícula. De esta manera, se convirtieron en las primeras universidades en aplicar esta tecnología en España.

Lola resuelve dudas como: cuándo empieza el proceso de matriculación o cuándo se podrán consultar las admisiones en la carrera deseada. Utiliza la tecnología Dialogflow de Google, que es la que le permite ir aprendiendo de sus interacciones y la mejora continua. El desarrollo estuvo a cargo de la empresa 1 Million Bot y se financió a través del banco Santander.

En sólo unas semanas, el chatbot ya había atendido a un total de 4609 alumnos y llevó a cabo 13184 conversaciones. En los días de mayor afluencia, resolvió unas 38708 dudas. El porcentaje final de respuestas acertadas fue del 91,67%.

Una característica muy importante y particular de Lola es que interactúa y comprende el lenguaje que utilizan los estudiantes. Además, explican las autoridades de dichas universidades, el nombre del chatbot no es casual, si no que es un homenaje a Lola Ferrer, la primera persona que empezó a trabajar en el Servicio de Información Universidad de Murcia, contestando las dudas de los estudiantes.

Chatbot Beacon

Staffordshire University presentó en Enero de 2019 a Beacon, un chatbot creado para brindar ayuda las 24 horas del día a sus estudiantes. Para el desarrollo la universidad se asoció con un proveedor de servicios en la nube llamado ANS Group, y se convirtió en la primera universidad de Reino Unido utilizar esta tecnología.

En particular brinda información personalizada relacionada con los horarios de cursadas de las materias en las que están anotados los alumnos que interactúan, establece contacto entre los estudiantes y sus tutores personales, y además provee respuestas a 400 preguntas frecuentes acerca de las instalaciones del campus, servicios de apoyo entre otros.

Beacon reconoce a los estudiantes cuando se loguean en la aplicación y estos pueden elegir si interactuar a través de mensajes de texto o de voz. Cuanto más conversan, el chatbot se vuelven más inteligente y las autoridades de la universidad afirman que tienen pensado desarrollar notificaciones para recordar a los alumnos sobre conferencias y ofrecer consejos sobre libros y ayudas de estudio útiles.

Chatbot Lucy

En Febrero de 2018, la Universidad de Canberra dio a conocer a Lucy, un chatbot para brindar información importante a sus estudiantes.

Para poder tener las respuestas a las consultas de sus usuarios, Lucy escanea el contenido de sitios de soporte de la universidad como por ejemplo AskUC. De esta manera puede ayudar a los estudiantes con consultas que van desde la inscripción y los horarios de clase hasta el estacionamiento de automóviles y los servicios para estudiantes.

Algo que distingue a Lucy es que, si no puede responder una pregunta, los estudiantes pueden pedirle al sistema que levante un ticket con el equipo del Centro de Estudiantes de la Universidad para recibir una respuesta.

Chatbot Diario Clarín

Si bien este chatbot no corresponde al ámbito universitario, lo citamos porque para la implementación del nuestro nos basamos en el flujo de conversación que desarrolló el Diario Clarín, brindando a los usuarios botones (respuestas rápidas de Facebook Messenger) con los temas que comprende, para ayudarlos con la interacción y evitando la frustración ante una consulta que no tenga respuesta. De esta manera tanto los lectores de Clarín, como los estudiantes de la UNLP podrán notar que ambos chatbots son muy similares al momento establecerse la conversación.

Clarín hizo público su chatbot en Octubre de 2016, al cual se puede acceder a través de la plataforma Facebook Messenger. Ofrece una serie de opciones a las que el usuario puede suscribirse y permite marcar sus preferencias por secciones y mirar las notas.

El servicio brinda la posibilidad de recibir resúmenes diarios, alertas con noticias de último momento e información actualizada sobre el clima y los transportes. Además, se puede consultar por la tapa del día y buscar noticias por temas. Un ejemplo: si uno escribe "Messi", el robot le enviará en forma instantánea un carrusel con noticias sobre el futbolista argentino.

Este chatbot tiene una particularidad: no está 100% robotizado. Esto significa que el usuario puede suscribirse a resúmenes informativos armados manualmente por los periodistas del diario. Los mismos se envían de lunes a viernes y tienen dos horarios: el de la mañana (a las 10) y el de la tarde (a las 17). En la tarjeta de "Resumen de Noticias" pueden elegir a cual de ellos desean suscribirse.

Asuntos estudiantiles

"El área de Asuntos Estudiantiles de la UNLP tiene como objetivo contribuir al proceso de formación de los estudiantes en las distintas etapas, promover la participación activa en programas y proyectos, fomentar políticas de bienestar para garantizar el ingreso, la permanencia y el egreso, a través de programas de apoyo económico, becas, prácticas profesionales, pasantías, actividades que complementen la actividad académica como charlas y visitas, entre otras actividades que fortalezcan la vinculación de la facultad con el entorno." (Sitio Web, Prosecretaría de Asuntos Estudiantiles. 2019)

Situación Actual

Actualmente, el área de Asuntos Estudiantiles de la UNLP posee los siguientes canales de comunicación:

- Página web: https://unlp.edu.ar/la_palabra/bienestar-universitario-10700
- Facebook: <https://www.facebook.com/BienestarUNLP/>
- Twitter: <https://twitter.com/BienestarUNLP>
- Instagram: <https://www.instagram.com/bienestarunlp/?hl=es-la>

- Mail: pbu.presidencia@presi.unlp.edu.ar
- Folletos

Hernán J. Figueredo, el encargado del área, especificó que el alcance actual está brindado por dichos canales. Si bien es el esperado, el problema que poseen es que los clientes de los medios de comunicación “no leen el contenido” y realizan preguntas sobre lo ya indicado en los mismos.

Por otro lado, y al indicarle acerca de nuestra tecnología y las ventajas que esta posee, surgió la problemática de las faltas ortográficas de los clientes. Más adelante brindaron la solución planteada a dicho problema.

El abanico de temas con los que cuenta el área son:

Becas estudiantiles:

El amplio sistema de Becas que ofrece la UNLP está contemplado en el Programa Igualdad de Oportunidades para Estudiar, destinado a estudiantes de todos los años.

Las becas actuales son:

- **Beca de ayuda económica:** consiste en un beneficio económico que se propone como mecanismo de contención para facilitar la permanencia en los estudios de grado a sectores estudiantiles de bajos recursos.
- **Beca para estudiantes inquilinos:** tiene como objeto facilitar la permanencia en los estudios de grado a los estudiantes que sean inquilinos y provengan de alguna localidad del interior del país.
- **Beca de Comedor Universitario:** consiste en una beca completa para acceder al menú de forma totalmente gratuita.
- **Beca para estudiantes con discapacidad:** busca facilitar la permanencia en los estudios de grado a estudiantes que poseen algún tipo de discapacidad. El trabajo de selección y seguimiento de los/as beneficiarios/as es realizado en conjunto con el Departamento de Atención Social.
- **Beca para estudiantes con hijos:** para estudiantes que tienen hijos de entre 45 días y 5 años de edad. La UNLP busca colaborar en la cobertura de los costos destinados a su cuidado en establecimientos especializados, o a cargo de una persona debidamente capacitada.
- **Beca de Bicicleta universitaria:** consiste en el préstamo de una bicicleta que el beneficiario podrá utilizar durante el período que duren sus estudios de grado.
- **Beca de Albergue Universitario:** destinada a alumnos provenientes del interior, brinda alojamiento, alimentación y transporte en forma gratuita.

Comedor:

Este espacio brinda un menú completo, nutritivo y saludable, con un costo subsidiado.

Además, es un punto de encuentro e intercambio con docentes, no docentes y estudiantes de todas las carreras. Posee varias sedes. El valor de cada almuerzo está subvencionado en un 60% por la UNLP.

Salud:

La UNLP brinda una cobertura integral de la salud, a través de su promoción, prevención, educación y atención. Los servicios gratuitos que ofrece son los siguientes:

- Atención Médica Clínica
- Centro de vacunación
- Consultorios ginecológicos/odontológicos/de obstetricia
- Seguro Público de Salud
- Atención médica domiciliaria
- Entrega de medicamentos
- Análisis clínicos
- Curso de cuidado de la voz

Talleres:

Espacios gratuitos de aprendizaje, integración y creación, que permiten acceder a la práctica y el aprendizaje de una disciplina.

Albergue:

El Albergue Universitario es parte de las políticas de igualdad de oportunidades para estudiar que desarrolla la Universidad Nacional de La Plata. La Beca del Albergue consiste en el acompañamiento integral de los estudiantes, garantizando las condiciones mínimas para poder transitar una carrera universitaria. El albergue universitario es una apuesta que considera a la educación universitaria como un bien social, a través de acompañar el trayecto universitario de los becarios, facilitar su proceso de aprendizaje individual y grupal, en la búsqueda de su desarrollo integral.

Solución Planteada

Desde nuestro lugar, y tras escuchar atentamente las inquietudes del responsable del área, especificamos las soluciones que la tecnología puede brindar:

- Para las faltas de ortografía, el chatbot posee 2 caminos de respuesta:
 - Respuestas rápidas: el cliente solo tiene que clicar en ellas, bajando considerablemente el ingreso desde teclado del mismo.
 - Escritura desde teclado: la solución al problema antes mencionado es la técnica de Procesamiento de Lenguaje Natural, el cual transforma una frase ingresada por el usuario en entidades reconocidas por la aplicación. Este reconocimiento de entidades tiene una tolerancia denominada por Wit como "umbral de confianza", para los casos donde el cliente no utilice una ortografía correcta, la confianza bajará pero igualmente puede quedar dentro del umbral. Además vamos a proveer una herramienta para cargar sinónimos, buscando con esta solución sumar mayor vocabulario al chatbot.
- Foco del cliente: para mantener el foco del cliente, el chatbot se basa en preguntas cerradas, y respuestas concisas a sus preguntas. A esto se le suman las respuestas rápidas, que proveen solamente la información necesaria, a la cual el cliente está buscando acceder. Evitando así amplios textos y lecturas enfocadas en el interés del cliente.

Aspectos Técnicos

En el siguiente capítulo pasaremos a comentar la arquitectura desarrollada y las tecnologías utilizadas.

Arquitectura

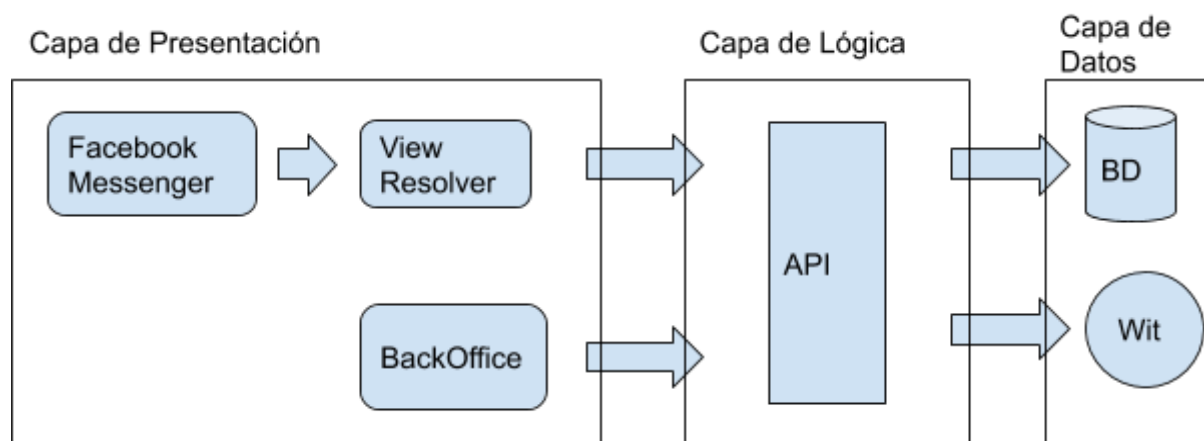


Figura 1.2

La arquitectura de la aplicación (Fig 1.2) está compuesta por:

- **Facebook Messenger:** Realizamos interacciones con la API de Facebook mediante los cuales enviamos y recibimos los mensajes a la plataforma Facebook Messenger
- **View Resolver:** Esta capa es la encargada de la comunicación entre nuestra API y la API de Facebook.
- **BackOffice:** Desarrollado con el framework frontend Angular 4, este componente nos brinda la visual mediante la cual el Operador podrá cargar información y, de esta manera, potenciar la inteligencia del Chatbot.
- **API:** Contamos con una API rest, desarrollada en Java, la cual es responsable de las interacciones con la base de datos y la API de Wit.
- **BD:** Utilizamos una base de datos Cassandra, mediante la cual almacenaremos información relacionada a entidades conocidas (ej: Becas) las cuales el chatbot comprende para realizar sus respuestas.
- **Wit:** Disponemos de una conexión con la API de Wit. A dicha API le enviaremos los mensajes enviados por el usuario y esta se encargará de resolver las entidades y la

intención del mensaje. Esta conversión es realizada mediante el mecanismo NLP brindado por la herramienta.

Paso a paso

Para lograr una arquitectura eficiente, decidimos que el primer estudio que debíamos realizar eran las interacciones de la aplicación, observando cuán necesarias serían determinadas implementaciones y considerando delegar lógica en aplicaciones externas.

En una primera instancia todo nace como una aplicación monolítica² (Fig. 1.3), centralizando toda la información en una misma aplicación.

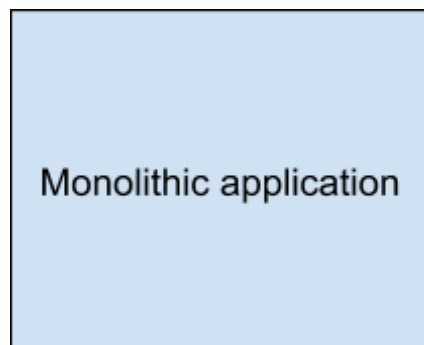


Figura 1.3

Ante este panorama, empezamos a analizar posibles subsistemas en donde delegar responsabilidades.

Comenzamos estudiando implementaciones actuales. Observamos varios casos, como por ejemplo el chatbot de Grupo Clarín S.A. y Farmacity S.A., que son reconocidos como casos de éxito, debido a las consultas diarias que reciben. A partir de esto, llegamos a la conclusión de que la plataforma más utilizada en la instanciación de chatbots es Facebook Messenger.

Una vez establecidas las alternativas, analizamos ventajas y desventajas de delegar en Facebook Messenger la interacción con los usuarios o desarrollar nosotros un módulo para resolverla.

Para esto clasificamos las principales características de la siguiente manera:

- **Hosting:** En caso de aceptar nosotros la responsabilidad de desarrollar la solución, deberíamos buscar un sitio donde alojar dicho componente. Caso contrario, Facebook Messenger nos brindaría el hosting necesario.
- **Arquitectura:** El desarrollo propio de un componente trae aparejada la obligación de colocarlo en un sitio en el que desconocemos la arquitectura utilizada. En caso de que la arquitectura sea la esperada, tendremos que analizar las versiones mobile,

² Una arquitectura monolítica describe una aplicación en la que toda la funcionalidad del sistema (ej. acceso a datos, interfaz de usuario, lógica, etcétera) está implementada y mezclada en una sola capa.

por ejemplo Android y iOS. A su vez, contaremos con el desarrollo de soluciones para browsers accedidos desde dispositivos móviles. Con la utilización de Facebook Messenger se omitirá esta problemática.

- **Alcance:** Si bien desde el punto de vista de aplicaciones desktop el alcance sería similar, actualmente hay que considerar el alcance para usuarios de aplicaciones móviles. En este caso, Facebook Messenger cuenta con ventaja debido a la gran cantidad de usuarios que utilizan sus aplicaciones.
- **Accesibilidad:** El compromiso de desarrollar una aplicación accesible demanda respetar varios protocolos y estándares. En caso de optar por Facebook Messenger tendríamos esto resuelto, debido a que la aplicación dispone dicha característica.
- **Compatibilidad:** Al desarrollar el módulo por nuestra cuenta, deberíamos contemplar la problemática conocida como compatibilidad de browsers. De utilizar Facebook Messenger no contaríamos con este tipo de problema.
- **Propiedad:** Entendemos como una ventaja contar con el ownership del módulo desarrollado, debido a que podríamos responder ante situaciones adversas. En caso de utilizar Facebook Messenger seríamos dependientes de los problemas de la plataforma.

Basados en el análisis antes mencionado, optamos por la integración con la API de Facebook Messenger (Fig. 1.4).

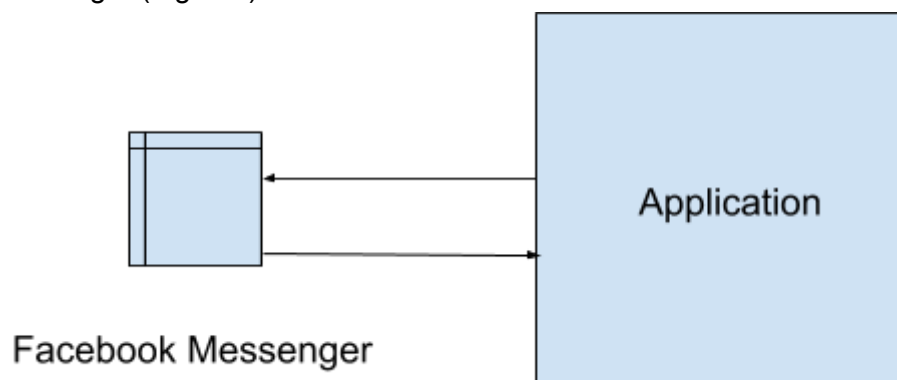


Figura 1.4

Una vez resuelto el componente de mensajería, debíamos analizar qué es lo que haríamos con la información introducida por los usuarios. ¿Cómo la procesaríamos? Investigando el mercado, dimos con el concepto: Natural Language Processing (NLP), el cual es un subcampo de Artificial Intelligence (AI) que se enfoca en permitir que las computadoras entiendan y procesen lenguajes humanos, para acercar a las computadoras a una comprensión del lenguaje a nivel humano.

En este punto nos vimos inmersos en una problemática similar a la del módulo de mensajería: Implementar la solución o delegar en un subsistema o API externa.

Realizamos un análisis de los componentes utilizados hoy en día y optamos como mejor alternativa: Wit.ai.

Wit es una interfaz de lenguaje natural para aplicaciones capaces de convertir oraciones en datos estructurados. Esto significa que podés crear bots, que permiten interactuar con personas en plataformas de mensajería. Con Wit.ai es posible crear aplicaciones con las

que podés hablar o enviar mensajes. Los bots siguen aprendiendo a medida que se les habla y se vuelven más inteligentes con cada interacción.

Si bien dicha herramienta soluciona la comprensión de los mensajes enviados por el usuario, esta solución era parcial, pues necesitábamos un lugar donde alojar los datos y la posibilidad de guardarlos en memoria no era una solución viable, debido a que necesitábamos que dichos datos persistan en el tiempo.

Descartada rápidamente la posibilidad de alojar los datos en memoria, la problemática era clara, necesitábamos una Base de Datos para alojar el conocimiento de nuestra aplicación, el estudio debía apuntar a cuál era la indicada para nuestro caso.

Cuando se trata de elegir una base de datos, una de las decisiones más importantes que se debe tomar es la elección entre una estructura de datos relacional (SQL), o no relacional (NoSQL). Si bien ambas son buenas opciones, cada una tiene sus ventajas y desventajas. Antes de realizar la comparación entre ambas, describamos cada una:

- **SQL** (*Structured Query Language*): es un lenguaje de programación que se utiliza para administrar datos en bases de datos relacionales. Las bases de datos relacionales usan relaciones (generalmente llamadas tablas) para almacenar datos y luego hacer coincidir esos datos mediante el uso de características comunes dentro del dataset.
- **NoSQL** (Not only SQL): una base de datos NoSQL, por otro lado, se autodescribe, por lo que no requiere un esquema. Tampoco impone relaciones entre tablas en todos los casos. Su nombre se refiere a bases de datos no relacionales de alto rendimiento, que utilizan una amplia variedad de modelos de datos. Estas bases de datos son altamente reconocidas por su facilidad de uso, rendimiento escalable, gran capacidad de recuperación y amplia disponibilidad.

Existen muchas diferencias entre SQL y NoSQL, las mismas se dividen en:

- Lenguaje
- Escalabilidad
- Comunidad
- Estructura

A continuación detallaremos cada una.

Lenguaje: Una de las principales diferencias entre las bases SQL y NoSQL es el lenguaje. Como se mencionó, las bases de datos SQL utilizan un lenguaje de consulta estructurado para definir y manipular datos. Esto permite que el SQL sea extremadamente versátil y ampliamente utilizado. Sin embargo, también lo hace más restrictivo. SQL requiere que se usen esquemas predefinidos para determinar la estructura de sus datos, antes de comenzar a trabajar con ellos. Sus datos también deben seguir la misma estructura, lo que puede implicar una preparación inicial significativa junto con una ejecución cuidadosa.

Una base de datos NoSQL presenta un esquema dinámico para datos no estructurados y los datos se pueden almacenar de maneras diferentes: en forma de gráficos, orientados a documentos, orientados a columnas u organizados como un almacén key-value. Esta flexibilidad extrema le permite crear documentos sin tener que planificar y definir cuidadosamente su estructura, agregar campos a medida que avanza y variar la sintaxis de

una base de datos a otra. También le permite dar a cada documento su propia estructura única, brindándole más libertad en general.

Escalabilidad: Otra gran diferencia entre SQL y NoSQL es su escalabilidad. En la mayoría de las bases de datos SQL, son escalables verticalmente, lo que significa que puede aumentar la carga en un solo servidor aumentando los componentes como RAM, SSD o CPU. En contraste, las bases de datos NoSQL son escalables horizontalmente, lo que significa que pueden manejar un aumento del tráfico simplemente agregando más servidores a la base de datos. Las bases de datos NoSQL tienen la capacidad de ser más grandes y mucho más potentes, lo que las convierte en la opción preferida para conjuntos de datos grandes o en constante evolución.

Comunidad: Las bases de datos SQL son más maduras, lo que se traduce en una comunidad mucho más fuerte y más desarrollada en comparación con las NoSQL. Hay miles de chats y foros disponibles donde los expertos pueden compartir conocimientos y discutir las mejores prácticas de SQL, mejorando continuamente las habilidades. Esto no quiere decir que la comunidad NoSQL sea un factor negativo, pero si hay que tener en cuenta que a la hora de comparar dichas comunidades, la SQL tiene ventaja.

Estructura: Finalmente, el último ítem a considerar es la estructura. Las bases de datos SQL están basadas en tablas, lo que las convierte en una mejor opción para las aplicaciones que requieren transacciones de varias filas. Un ejemplo de esto pueden ser sistemas contables o incluso sistemas heredados que se construyeron originalmente para una estructura relacional. Las bases de datos NoSQL pueden ser pares key-value, almacenes wide-column, bases de datos de gráficos o basadas en documentos.

Basados en estas diferencias y en la lógica de nuestra aplicación, optamos por la selección de una Base de Datos NoSQL. Los motivos son:

- El dinamismo que nos proporciona el lenguaje
- La escalabilidad horizontal, ya que teniendo en cuenta que nuestra aplicación en un primer momento será pequeña, queremos ofrecer la posibilidad de un crecimiento exponencial.
- Creemos que la estructura indicada es pares key-value.

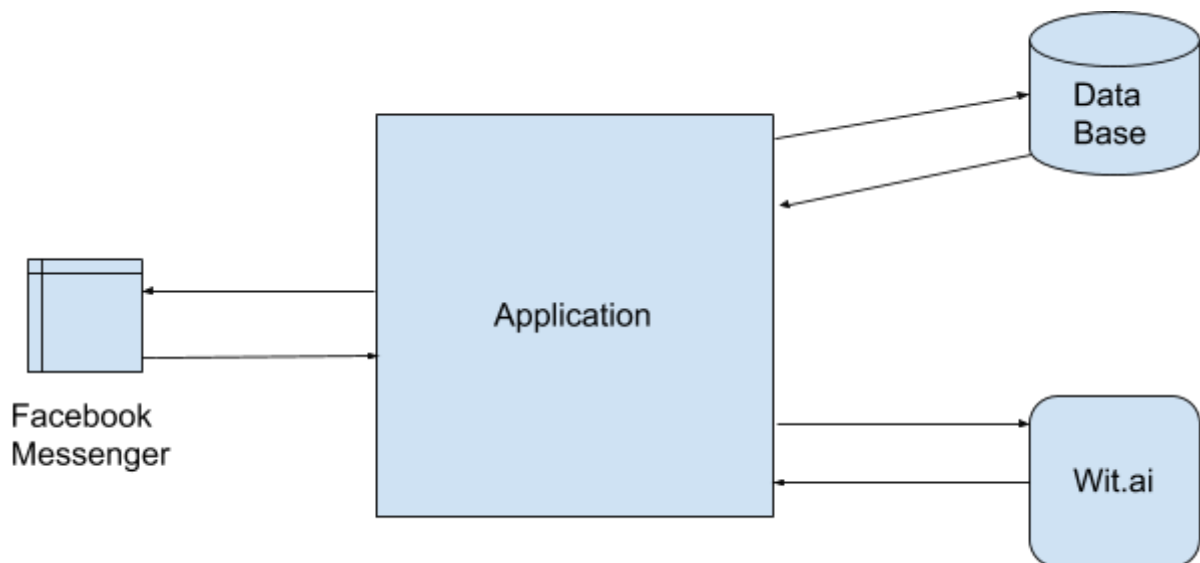


Figura 1.5

En este punto habíamos resuelto la interacción con el usuario, a través de Facebook Messenger, y la comprensión de sus mensajes, a través de Wit.ai (Fig. 1.5). Pero nos faltaba un componente el cual alimente de conocimiento a nuestro chatbot.

Para esto acordamos crear un Backoffice (Fig. 1.6), donde un usuario del sistema puede loguearse y mediante determinadas acciones cargar contenido al chatbot.

El desarrollo del Backoffice fue realizado con el framework frontend Angular 2+. A continuación listaremos algunas ventajas y desventajas del mismo, y finalmente explicaremos por qué lo hemos elegido:

Pros:

- Arquitectura basada en componentes que permite una mayor calidad de código: Los componentes pueden ser vistos como pequeñas piezas de una interfaz y que son independientes entre ellas. Este tipo de arquitectura da muchas ventajas como:
 - Reusabilidad: Los componentes que tienen una misma naturaleza pueden ser encapsulados. Los desarrolladores pueden reusarlos en diferentes partes de la aplicación.
 - Legibilidad: El encapsulamiento también permite a los desarrolladores que se acaban de sumar a un proyecto leer mejor el código.
 - Tests unitarios: La naturaleza independientes de los componentes simplifica los tests unitarios.
 - Mantenimiento: Los componentes que son fácilmente desacoplados entre sí puede ser fácilmente reemplazados con mejores implementaciones.
- Está escrito usando el lenguaje TypeScript, que elimina errores muy comunes en la escritura de código JavaScript, esto da como resultado un código mucho más limpio. Además lo hace mucho más escalable.
- Posee un soporte a largo plazo (LTS - Long Term Support) de Google, esto significa que por mucho tiempo Angular 2+ será mantenido y mejorado por un gigante como Google.

Contras:

- La curva de aprendizaje es algo mayor en relación a React o Vue, ya que hay varios tópicos y aspectos a ser cubiertos como módulos, inyección de dependencia, componentes, servicios y templates, etc.
Además, la librería de programación reactiva (RxJs) para la programación asíncrona -aunque sea en los niveles iniciales de Angular- debe ser usada, lo cual añade aún más complejidad.
- La falta de documentación oficial para el CLI (Interfaz de Línea de Comandos) es algo que preocupa mucho a la comunidad de desarrolladores que usan Angular. Y nosotros no fuimos la excepción a lo largo de la tesis.

La elección de Angular como framework para el desarrollo frontend del backoffice para esta tesis, se dio por el conocimiento que teníamos acerca de esta tecnología y que hemos adquirido en el ámbito laboral, con diferentes proyectos en los que participamos. Con lo cual las dos contras que podíamos encontrar sobre este framework estaban solucionadas al momento de comenzar con la implementación.

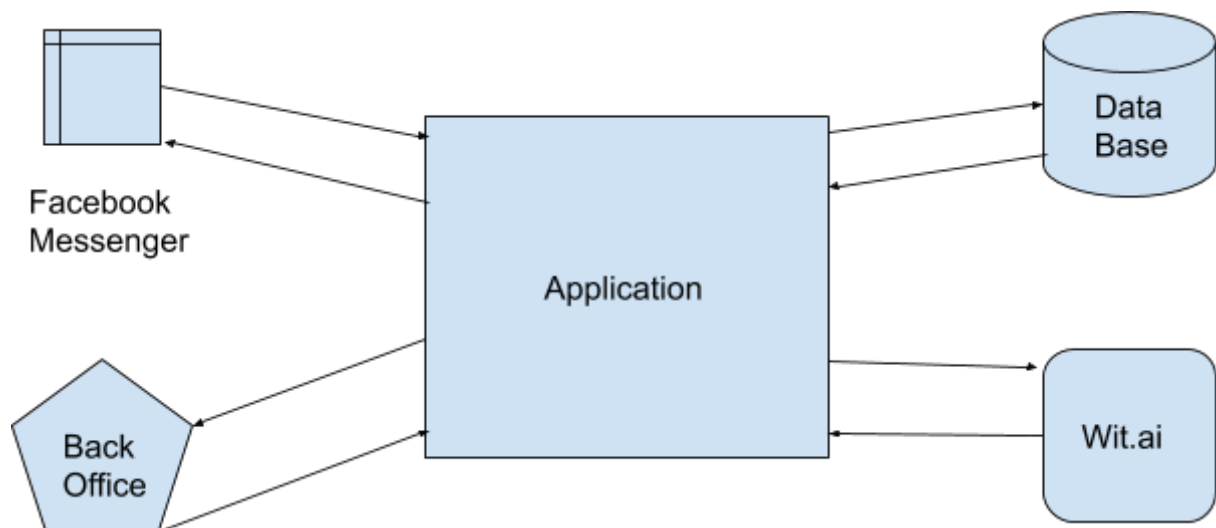


Figura 1.6

Con la idea del Backoffice, solo nos quedaba analizar qué debía responder el sistema. Partimos de la base que el chatbot sabe comprender preguntas y dar respuestas. Entendimos que estas preguntas deberían ser cerradas, debido a que si le damos al usuario la posibilidad de explayarse en sus respuestas, la comprensión del mensaje sería compleja, y Wit no entendería el tema ni la intención del usuario. Para el correcto funcionamiento, necesitábamos crear un flujo de conversación, así íbamos a ayudar al usuario a armar sus preguntas y a su vez nos daría el control de la conversación. Para esto identificamos los principales temas a tratar y sus categorías, quedando la siguiente resultante:

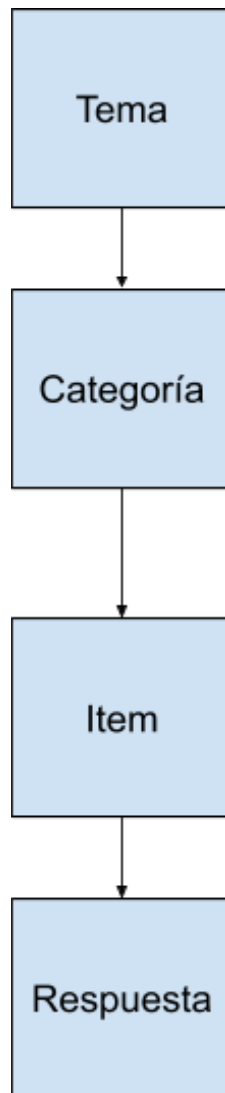


Figura 1.7

Mediante dicho flujo (Fig. 1.7), pudimos crear las interacciones necesarias para poder responder varias preguntas. El caso práctico sería: el usuario elige el tema de su incumbencia, luego selecciona la categoría y por último el ítem, aquí es donde nuestro chatbot brinda la información.

Todo esto es posible a través de las quick replies provistas por Facebook Messenger (Fig 1.8). Las Quick Replies (o respuestas rápidas) nos proporcionan una forma de presentar un conjunto de hasta 11 botones de conversación que contienen un título y una imagen opcional, y aparecen sobre el input de texto. También puede usar respuestas rápidas para solicitar la ubicación, la dirección de correo electrónico y el número de teléfono de una persona.

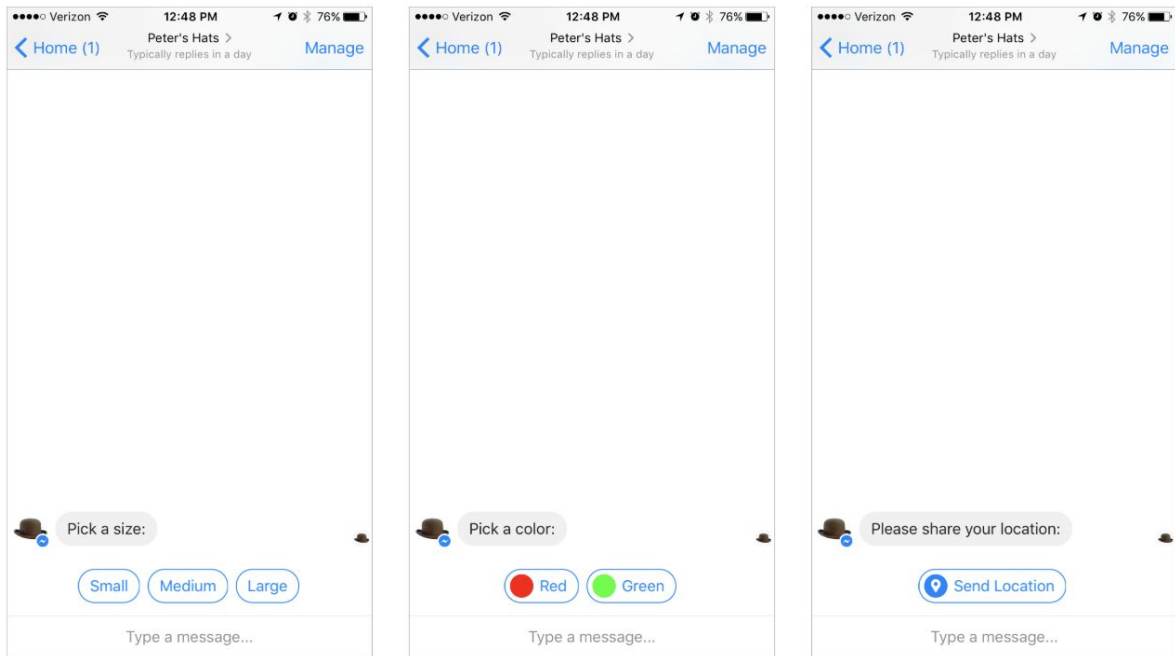


Figura 1.8

Paralelamente a esto, el usuario puede optar por la comunicación mediante entrada de texto, aquí es Wit quien se encarga de traducir el tema/categoría/item y la intención del usuario.

Para la implementación del flujo conversacional mencionado anteriormente, necesitábamos una estructura de datos eficiente, la cual en pocos accesos pueda recuperar los datos requeridos. Para esto creamos una estructura de árbol, donde cada tema pueda poseer varias categorías y cada categoría contenga varios ítems (Fig. 1.9).

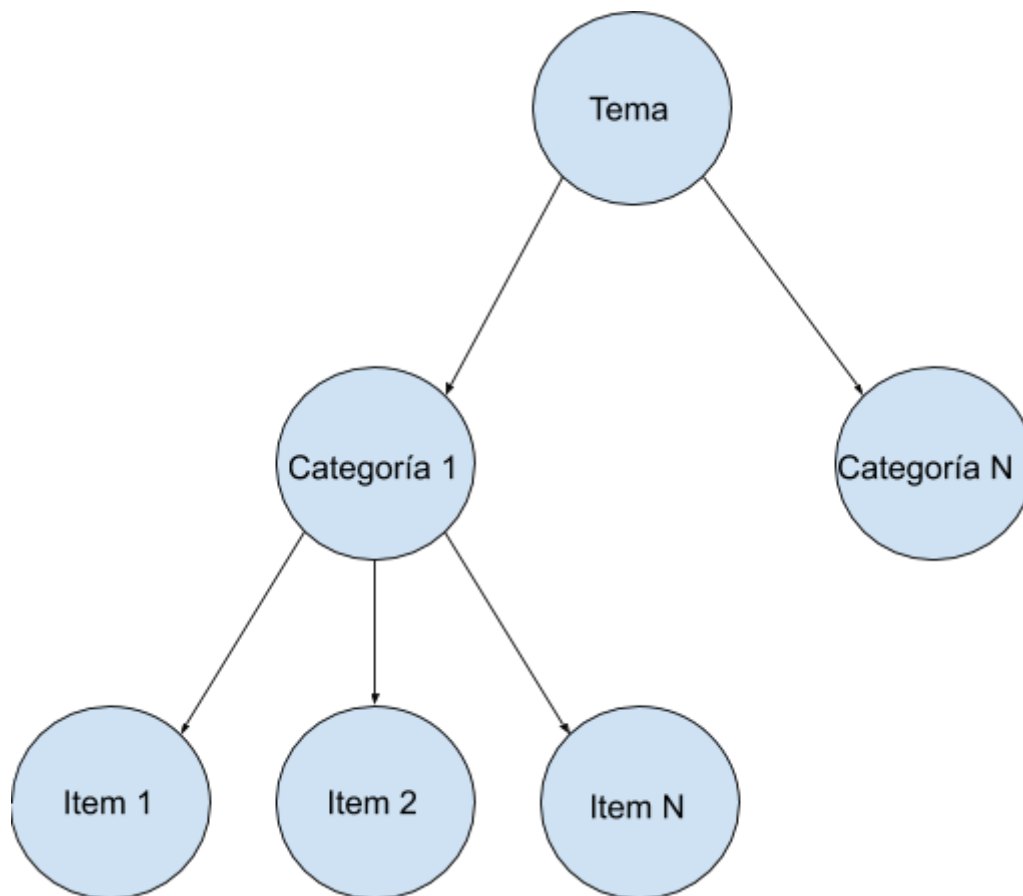


Figura 1.9

Esta lógica es complementada con la estructura de nuestros objetos guardados en la base de datos, los cuales indexamos por el id de la entidad.

Por lo tanto, el usuario selecciona el quick reply o mediante entrada de texto nos comunica que es lo que desea consultar. Wit nos traduce el mensaje recibido en entidades que nosotros comprendemos y a partir de esto vamos a la base de datos a recuperar el objeto y, en caso de la entidad no ser un nodo hoja, consultamos por sus hijos, transformándolos luego en los nuevos quick replies.

Elegimos esta arquitectura porque entendíamos que era óptima y una manera sencilla de recuperar las entidades necesarias.

Ya con la lógica de la aplicación diagramada, procedimos a la selección de la base de datos. En ese momento contábamos con las siguientes premisas:

- La Base de datos debía ser NoSQL
- Nuestro sistema contaría con muchas lecturas a la base de datos (consultas de usuarios mediante Facebook Messenger) y pocas escrituras (carga de conocimiento a través del Backoffice)

Una vez definida la lógica de la aplicación y las funciones de la misma, estábamos en condiciones de definir la base de datos a utilizar.

Las premisas con las que contábamos eran las siguientes:

Escalabilidad: si bien el sistema era pequeño y contaba sólo con una instancia en el cluster de la base de datos, queríamos brindar una solución escalable a futuro.

- En este punto entendíamos que el sistema iba a tener más peticiones de lectura que de escritura. Estimamos que menos del 5% de los request a la aplicación serían de escrituras, mediante el Backoffice, y más de un 95% de los request iban a ser de lecturas, mediante las consultas de los usuarios a través de la plataforma de Facebook Messenger.

Analizando posibilidades de selección, optamos por Cassandra, reconocida como un sistema muy robusto, de alto rendimiento y descentralizado. No es la única solución, pero al estar bien documentado y con una comunidad sólida, entendíamos que era una de las mejores opciones.

Ventajas que brinda Cassandra:

- Open Source: Cassandra es un proyecto de Apache de código abierto. Su naturaleza de código abierto ha dado a luz a una gran comunidad donde personas de ideas afines comparten sus opiniones, consultas y sugerencias relacionadas con Big Data.
- Arquitectura Peer-to-Peer: Posee una arquitectura de peer-to-peer, en lugar de la arquitectura master-slave. Por lo tanto, no hay un punto único de falla. Además, se puede agregar cualquier número de instancias a un clúster. Como todas las máquinas están en el mismo nivel, cualquier servidor puede recibir solicitudes de cualquier cliente.
- Escalabilidad elástica: Las instancias en un cluster pueden ser fácilmente agregadas o eliminadas. No es necesario reiniciar el clúster ni cambiar configuraciones. Esta es la razón por la cual Cassandra es popular: por tener un rendimiento muy alto para el mayor número de nodos. A medida que se incrementa el número de instancias en el cluster, el rendimiento de lectura y escritura aumenta simultáneamente sin tiempo de inactividad o cualquier pausa en las aplicaciones.
- Alta disponibilidad y tolerancia a fallas: La replicación significa que cada dato se almacena en más de una ubicación. Esto se debe a que, incluso si un nodo falla, el usuario debería poder recuperar los datos con facilidad desde otra ubicación. En un clúster de Cassandra, cada fila se replica según la clave de la fila. Puede establecer el número de réplicas que desea crear. Al igual que la escalabilidad, la replicación de datos puede ocurrir en múltiples clusters. Esto conduce aún más a las competencias de respaldo y recuperación de alto nivel en Cassandra.
- Alta performance: Demuestra un rendimiento brillante bajo grandes volúmenes de datos.
- Orientado a columnas: Cassandra tiene un modelo de datos de muy alto nivel. Esto significa que almacena columnas basadas en los nombres de las mismas.
- Consistencia Tunable: En Cassandra, la consistencia puede ser de dos tipos: consistencia eventual y consistencia fuerte. Se puede adoptar cualquiera en función de sus necesidades. La consistencia eventual garantiza que el cliente es aprobado

tan pronto como el clúster acepte la escritura. Por otro lado, la consistencia fuerte significa que cualquier actualización se transmite a todos los nodos donde se encuentra la información en particular. También tiene la libertad de combinar consistencia eventual y fuerte. Por ejemplo, puede obtener una consistencia eventual en el caso de datos remotos donde la latencia es bastante alta y obtener una consistencia fuerte para los datos locales donde la latencia es baja.

- Esquema free: Las columnas se pueden crear a tu voluntad dentro de las filas. El modelo de datos de Cassandra también es conocido como un modelo de datos de esquema opcional. A diferencia de una base de datos tradicional, en Cassandra no es necesario mostrar todas las columnas que necesita su aplicación en la superficie, ya que no se espera que cada fila tenga el mismo conjunto de columnas.

Una vez definida la base de datos y los componentes necesarios de la aplicación, llegamos a un punto donde la aplicación se encontraba lo suficientemente sólida. Aquí fue donde decidimos integrar el último módulo: View Resolver.

La necesidad era clara: queríamos abstraer nuestra API de la comunicación con las plataformas de mensajerías. Este componente brindaría así la escalabilidad necesaria para no generar una dependencia con la plataforma de Facebook Messenger, y el día de mañana poder delegar en éste las integraciones con cualquier plataforma de mensajería, respetando siempre la firma propuesta por la API.

Continuamos con la modularización y creamos el ViewResolver. Dicho módulo realizaría la comunicación entre nuestra API y la API de Facebook.

Este módulo brindó un gran valor de escalabilidad a nuestra arquitectura, ya que si el día de mañana se requiere instanciar el chatbot sobre otra aplicación de mensajería (ejemplo Instagram, Twitter, Whatsapp, etc), bastaría con realizar la integración en este componente, y las traducciones para respetar la firma de nuestra API, sin modificar comportamiento en los otros componentes.

Este módulo lo realizamos basados en siguiente arquitectura:

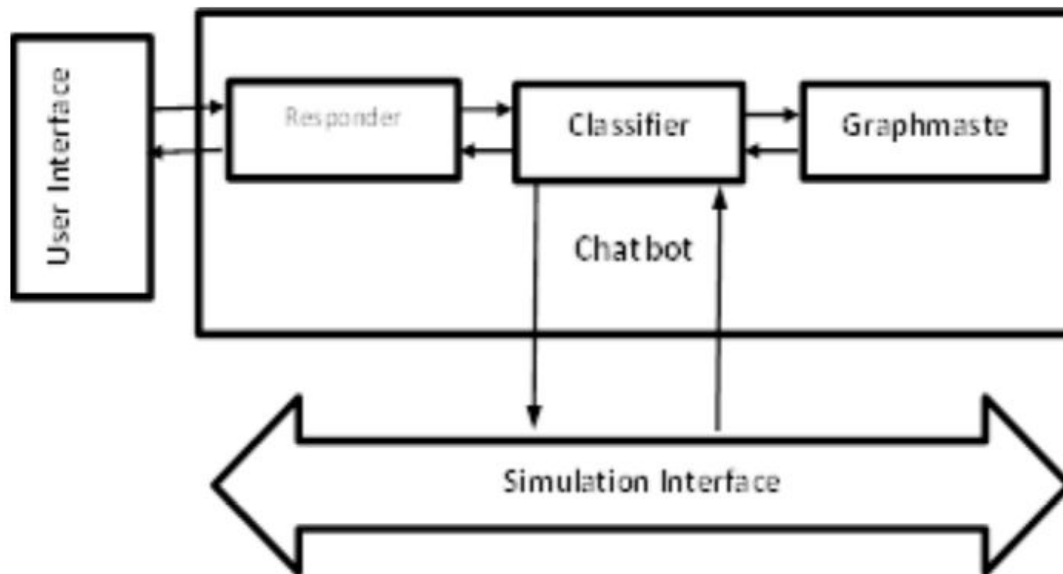


Figura 2.0. Components of Chatbot

El diseño de un Chatbot requiere la identificación de las partes constituyentes. El mismo puede ser dividido en tres partes:

- Responder: es la parte que desempeña el papel de interfaz entre las rutinas principales del bot y el usuario. Las tareas del respondedor son: transferir los datos del usuario al Classifier y control de la entrada y salida.
- Classifier: es la parte entre el Respondedor y el Graphmaster. Las funciones de esta capa son: filtrado y normalizando la entrada, segmentando la entrada introducida por el Usuario en componentes lógicos, transfiriendo la frase normalizada al Graphmaster, procesar la salida del Graphmaster, y comunicación con la base de datos.
- Graphmaster: es el cerebro de nuestro chatbot, encargado de realizar los algoritmos de coincidencia de patrones texto.

En nuestro caso, la entidad Responder la representamos con el View Resolver, la entidad Classifier será nuestra API y Wit hará las veces del Graphmaster.

Quedando la estructura de la siguiente manera:

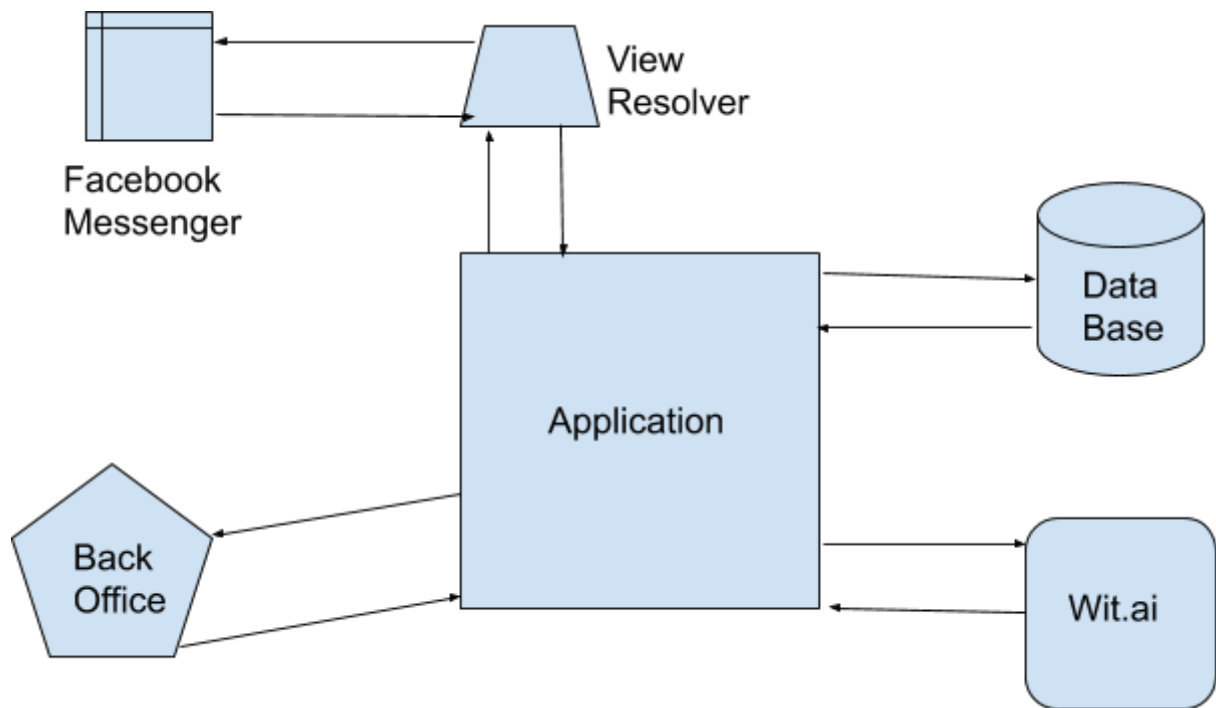


Figura 2.1

Estructura de la API

Para el diseño de nuestra API decidimos utilizar una arquitectura cliente-servidor N-tier, en la cual separamos en niveles las responsabilidades de la aplicación.

Comúnmente las arquitecturas N-tier respetan los siguientes niveles:

- Nivel de presentación
- Nivel de lógica
- Nivel de almacenamiento

Siguiendo la necesidad de nuestro sistema, decidimos segmentar el nivel de lógica y segmentar y modificar el nivel de almacenamiento, cambiándolo a nivel de comunicación, debido a que no solo nos abstrae de la interacción con la Base de Datos, sino que también asumirá dicha responsabilidad sobre la comunicación con clientes. Por lo tanto, la resultante será:

- Nivel de presentación
 - Controller
- Nivel de lógica
 - Facade
 - Operaciones
- Nivel de comunicación
 - Service
 - Client

De acuerdo a lo explicado, construimos el siguiente modelo:

Application

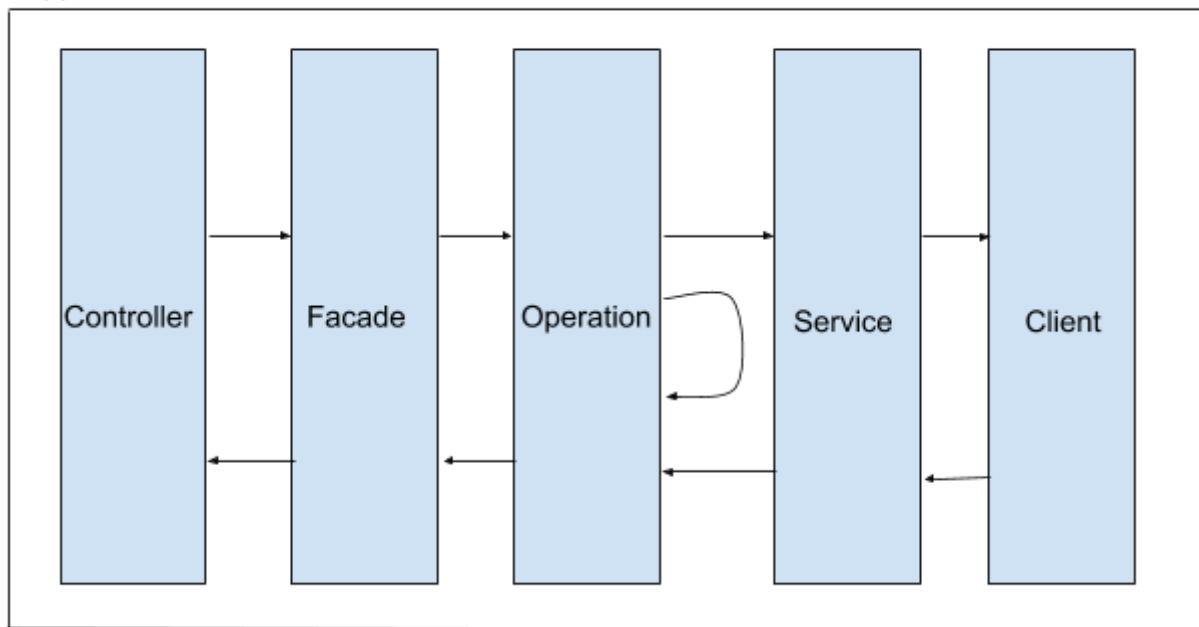


Figura 2.2

Responsabilidades de cada nivel

Cada nivel posee sus responsabilidades, que son encapsuladas ahí mismo para obtener un mayor grado de claridad en la arquitectura, haciendo sencillo agregar nuevo comportamiento.

Pasamos a detallar cada uno de estos.

Controller

Este es el nivel más alto de la aplicación. El nivel de presentación es el punto de entrada. Es quien brinda la firma de los servicios de la API, para que mediante ésta, el cliente externo pueda interactuar con cada uno. Luego de la recepción de la petición, se comunica con el siguiente nivel, solicitando la información al flujo indicado y espera la respuesta para enviarla al cliente.

Facade

La fachada de nuestra aplicación es el primer punto de abstracción. Ella conoce qué componentes del subsistema son los responsables de una determinada petición, y delega la resolución de las peticiones en el subsistema apropiado.

Utilizamos el patrón "*Facade by service*" debido a que la firma de la aplicación está organizada por servicios.

Operation

En el diseño de arquitectura seleccionado, las operations son los componentes más importantes de la aplicación. Juntas conforman el nivel encargado de controlar la funcionalidad de la aplicación, realizando un procesamiento detallado, abstrayéndonos de las comunicaciones con clientes.

Las operations representan una modularización de comportamiento, estas pueden tener interacciones con otras operations y resolver lógica en conjunto.

Service

Dicho nivel es el encargado de la abstracción de las interacciones con cada cliente, aglomerando todas las posibles interacciones de un cliente en particular en el mismo servicio.

Client

En el nivel de clientes contamos con las interacciones a otros subsistemas, APIs o bases de datos las cuales son demandadas por la lógica de la aplicación.

Ventajas

La arquitectura N-tier nos proporciona un modelo mediante el cual podemos crear una aplicación:

- Escalable
- Flexible
- Reusable
- Segregada
- Modularizada
- Legible
- Delegable
- Responsable

Tecnologías

En este apartado comentaremos las tecnologías seleccionadas.

Frontend

Angular

Es una plataforma que facilita la creación de aplicaciones con la web. Angular combina plantillas declarativas, inyección de dependencia, herramientas end to end y mejores prácticas integradas para resolver los desafíos de desarrollo. Angular permite a los desarrolladores crear aplicaciones web, móvil o de escritorio.

Para la implementación del backoffice se usó la versión 4, la cual está incluida dentro de familia de Angular 2+ en la cual se reescribió por completo el código de AngularJS.

Entre las herramientas que se usaron para complementar al framework elegido están:

- NgRx: es un framework para la construcción de aplicaciones reactivas en Angular. Provee el manejo de un estado global en la aplicación (ngrx/store), aislamiento de efectos secundarios a través de de la librería effects/ngrx, enlaces para conectar Angular Router con el estado global mencionado anteriormente. Permite la generación de código y el uso de herramientas como Store Devtools (se integra en

cualquier navegador para debuggear código) que mejoran mucho la experiencia de los desarrolladores.

- ngx-bootstrap: es una librería que nos permitió integrar fácilmente los componentes Bootstrap 3 con Angular.
- Angular CLI: Interfaz de línea de comandos (Command Line Interface) que permite crear proyectos Angular 2+, agregar archivos, llevar a cabo actualizaciones, realizar debugging testing y deployment.

Backend

Java

El lenguaje de programación Java® es un lenguaje de propósito general, concurrente, basado en clases y orientado a objetos. Está diseñado para ser lo suficientemente simple como para que muchos programadores puedan lograr fluidez en el lenguaje. Está relacionado con C y C++ pero se organiza de manera diferente, con una serie de aspectos que estos últimos omiten y algunas ideas de otros lenguajes. Está pensado para ser un lenguaje de producción, no un lenguaje de investigación.

Maven

Es una herramienta que se puede utilizar para construir y administrar cualquier proyecto basado en Java. Busca hacer más fácil el trabajo diario de los desarrolladores y ayudar con la comprensión de cualquier proyecto.

Del Yiddish, significa acumulador de conocimiento. Se inició originalmente como un intento de simplificar los procesos de construcción en el proyecto Yakarta Turbine. Se crearon varios proyectos, cada uno con sus propios archivos de compilación Ant, que fueron ligeramente diferentes y los JAR se registraron en CVS. Buscaban una forma estándar de construir los proyectos, una definición clara de en qué consistía el proyecto, una forma fácil de publicar información del proyecto y una forma de compartir JARs en varios proyectos.

Spring

Spring Framework proporciona un completo modelo de programación y configuración para aplicaciones empresariales modernas basadas en Java, en cualquier tipo de plataforma de implementación.

Un elemento clave de Spring es el soporte de infraestructura a nivel de aplicación: Spring se enfoca en la "*plomería*" (*en inglés, plumbing*) de las aplicaciones empresariales para que los equipos puedan enfocarse en la lógica de negocios a nivel de aplicación, sin vínculos innecesarios con entornos de implementación específicos.

Jetty

Eclipse Jetty Web Server proporciona un servidor HTTP y un contenedor Servlets capaz de servir contenido estático y dinámico desde una instancia independiente o embebida. A partir de jetty-7, el servidor web de jetty y otros componentes centrales están alojados por la Fundación Eclipse.

El proyecto proporciona:

- Servidor HTTP asíncrono

- Contenedor de Servlet basado en estándares
- Servidor websocket
- Servidor http/2
- Cliente asíncrono (http/1.1, http/2, websocket)
- Soporte para OSGI, JNDI, JMX, JASPI, AJP

Api externa

Para el procesamiento de la información enviada por el usuario, decidimos utilizar Wit.ai, para poder realizar una correcta interacción, delegando la lógica de transformación del texto a intenciones.

NLP (Natural Language Processing)

"Una de las tareas fundamentales de la Inteligencia Artificial (IA) es la manipulación de lenguajes naturales usando herramientas de computación. Aquí, los lenguajes de programación juegan un papel importante, ya que forman el enlace necesario entre los lenguajes naturales y su manipulación por una máquina."

El uso del lenguaje natural (LN) en la comunicación hombre-máquina presenta a la vez una ventaja y un obstáculo con respecto a otros medios de comunicación." (Mg. Augusto Cortez Vásquez, Mg. Hugo Vega Huerta y Lic. Jaime Pariona Quispe. 2009)

Arquitectura de un sistema de NLP

La arquitectura de un sistema de NLP se sustenta en una definición del Lenguaje Natural por niveles. Estos son: fonológico, morfológico, sintáctico, semántico, y pragmático.

- Nivel Fonológico: trata de cómo las palabras se relacionan con los sonidos que representan.
- Nivel Morfológico: trata de cómo las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas.
- Nivel Sintáctico: trata de cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega en la oración y que sintagmas son parte de otros sintagmas.
- Nivel Semántico: trata del significado de las palabras y de cómo los significados se unen para dar significado a una oración, también se refiere al significado independiente del contexto, es decir de la oración aislada.
- Nivel Pragmático: trata de cómo las oraciones se usan en distintas situaciones y de cómo el uso afecta al significado de las oraciones. Se reconoce un subnivel recursivo: discursivo, que trata de cómo el significado de una oración se ve afectado por las oraciones inmediatamente anteriores.

Wit

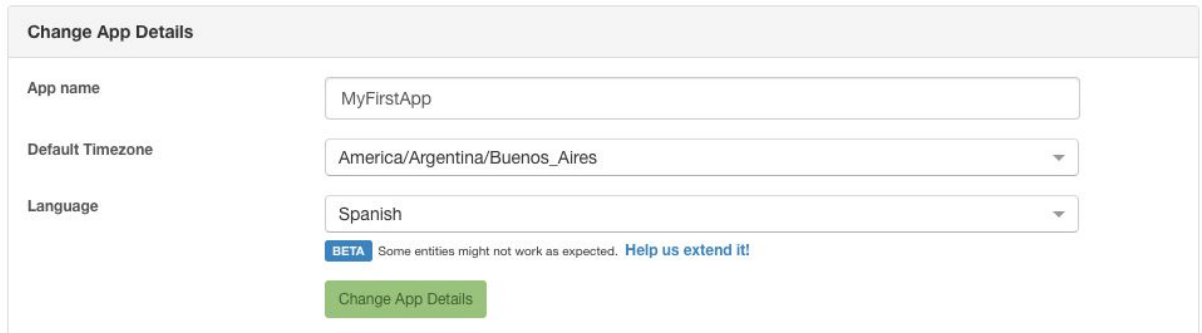
Wit es una interfaz de lenguaje natural para aplicaciones capaces de convertir oraciones en datos estructurados. Esto significa que puede crear bots que pueden interactuar con personas en plataformas de mensajería. Con Wit.ai los desarrolladores pueden crear aplicaciones con las que puedes hablar o enviar mensajes de texto.

Wit.ai ayuda a comprender qué dicen sus usuarios mediante extracciones de información relevante.

Wit utiliza la siguiente receta para ayudarte a resolver problemas comunes:

- Comprender: el procesamiento del lenguaje natural (NLP) permite comprender el significado de una entrada de usuario. Mediante éste, es posible transformar texto o voz en estructura de datos.
Junto con cada entidad (entity) extraída, Wit devuelve un nivel de confianza (confidence level), un valor que muestra cuán seguro está Wit de que extrajo la entidad correctamente. Cuando la confianza es demasiado baja, es posible que desee manejar la entidad extraída de manera diferente (o ignorarla por completo).
- Integrar Wit con nuestra aplicación: para llevar a cabo dicha acción usamos un Kit de Desarrollo de Software (SDK sus siglas en Inglés) de Node.js para Wit.ai, llamado node-wit y se encuentra en el repositorio de Github <https://github.com/wit-ai/node-wit>
- Administrar nuestra aplicación en Wit: creamos una nueva aplicación a través de la interfaz gráfica que nos brinda Wit.ai, la pusimos en modo privado para sólo tener acceso los desarrolladores de esta tesis.

En las las configuraciones (Fig. 2.3) elegimos el lenguaje Español para el entorno gráfico de Wit, la zona horaria de Buenos Aires (Argentina) y el Server Access Token lo tomamos para integrar nuestra aplicación de Wit con el SDK mencionado en el punto anterior.



Change App Details

App name: MyFirstApp

Default Timezone: America/Argentina/Buenos_Aires

Language: Spanish

BETA Some entities might not work as expected. Help us extend it!

Change App Details

Figura 2.3

Hemos decidido utilizar esta herramienta por las siguientes razones:

- Es open source
- Es sumamente sencillo implementarlo y usarlo sobre Facebook Messenger, que fue la plataforma de mensajería que elegimos para nuestro trabajo de tesis. Wit.ai fue adquirido en el año 2015 por Facebook.
- Tiene una gran comunidad de desarrolladores, lo cual da la posibilidad de acceder a bots públicos de otros usuarios para poder aprender y mejorar lo implementado.
- El motor de NLP es su beneficio más convincente, el modelo de Machine Learning que brinda Wit.ai está por sobre otras herramientas de grandes compañías en el mercado como Watson (IBM), Microsoft Bot Framework, Amazon Lex o Google DialogFlow.

Su mayor fortaleza está en su análisis de las expresiones escritas por los usuarios a través de su algoritmo NLP y en cómo devuelve respuestas valiosas y precisas.

A su vez también pudimos encontrar una contra, entrenar el motor NLP de Wit.ai se torna bastante laborioso. Este problema ocurre cuando Wit empieza a encolar los pedidos de entrenamiento que se hacen, y muchas veces los deja para empezar a hacerlos efectivos dentro de varias horas.

Este método fue el que encontró esta herramienta para evitar que usuarios malintencionados usen scripts para entrenar sus bots con muchísimos pedidos no permitiendo a otros usuarios bienintencionados entrenar su motor NLP.

Base de datos

Cassandra

La base de datos de Apache Cassandra es la opción correcta cuando se necesita escalabilidad y alta disponibilidad, sin comprometer el rendimiento. La escalabilidad lineal y la probada tolerancia a fallas en el hardware básico o la infraestructura en la nube, la convierten en la plataforma perfecta para datos de misión crítica. El soporte de Cassandra para la replicación en múltiples centros de datos es el mejor de su clase, porque brinda una menor latencia para sus usuarios y la tranquilidad de saber que puede sobrevivir a las interrupciones regionales.

Aplicación

Funcionalidad

En el siguiente apartado haremos foco en la funcionalidad de la aplicación desde el punto de vista de los usuarios del sistema, los cuales pueden ser:

- Usuario del Backoffice
- Usuario de Facebook Messenger

Interfaz del Usuario

En el presente capítulo se pretende explicar la importancia del back office para administrar la información y por lo tanto, la inteligencia del chatbot.

A su vez se detalla cada una de las funcionalidades que se implementaron, se describen algunas decisiones generales que se tomaron a lo largo de su desarrollo y las dificultades que se fueron presentando.

Además se mostrarán imágenes con su respectiva descripción del chatbot en funcionamiento en la plataforma Facebook Messenger. Con esto se pretende demostrar el impacto de la información cargada en el back office en las respuestas que da el chat a los usuarios que interactúan con él, en busca de respuestas precisas. Y, también, lo que pasa cuando alguna consulta no tiene una respuesta asociada.

Diseño del Back office

El Back office pretende ser el medio para administrar toda la información que brinda el Chatbot a través de un operador *data entry*³, que será el encargado de ingresar y remover todos los datos que considere necesarios para que las respuestas del Chabot sean óptimas.

³ Tarea de ingresar datos a una computadora para que estos sean procesados.

Con esta plataforma lo que se busca es que el Chatbot pueda manejar diferentes temas correspondientes al área de Asuntos Estudiantes de la Universidad Nacional de La Plata. Para desarrollar las diferentes funcionalidades que se implementaron se construyó e integró una *API*⁴, la cual es descrita en el capítulo Aspectos Técnicos.

Dentro de las acciones que se pueden llevar a cabo, se destacan: la posibilidad de loguearse y desloguearse. Poder crear y eliminar categorías, items, sinónimos y expresiones. Todas estas definiciones serán dadas más adelante dentro del marco de la plataforma.

La plataforma de back office cuenta con un formulario de logeo (Fig. 1.1), a la cual se ingresa colocando un nombre de usuario y su respectiva contraseña.

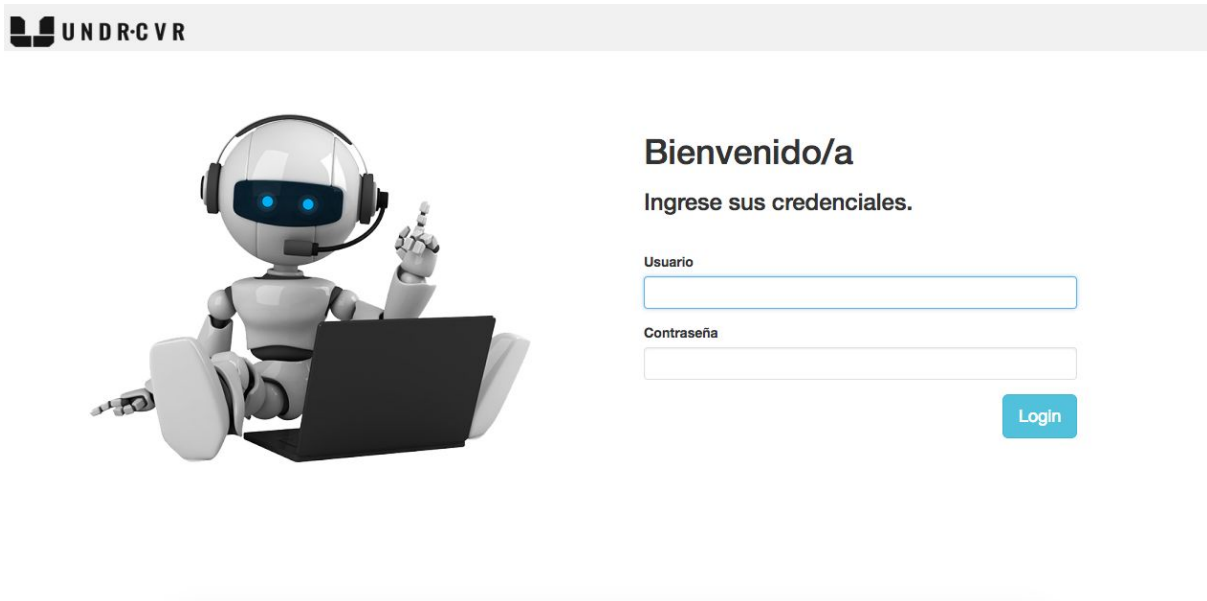


Figura 2.4

Dicho formulario cuenta con dos tipos de validaciones, la primera es completar los campos obligatorios, en este caso nombre de usuario y contraseña (Fig. 2.4)

⁴ Es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.



Bienvenido/a

Ingrese sus credenciales.

Usuario

El nombre de usuario es requerido.

Contraseña

La contraseña es requerida.

Login

Figura 2.5

La segunda validación refiere a que tanto el nombre de usuario como la contraseña existan en la base de datos y que se correspondan una con la otra, en caso de que esto no pase se mostrará un mensaje de error (Fig. 2.5)

[Consulta la información del sitio](#)



Bienvenido/a

Ingrese sus credenciales.

Usuario

Contraseña

Usuario y/o Contraseña inválido.

Login

Figura 2.6

Una vez que el usuario ingresa las credenciales y la validación fue correcta, se lo redirige a una pantalla que muestra la categorías⁵ listadas (Fig. 2.6).

⁵ Temas de conversación que el chatbot reconoce al interactuar con un usuario

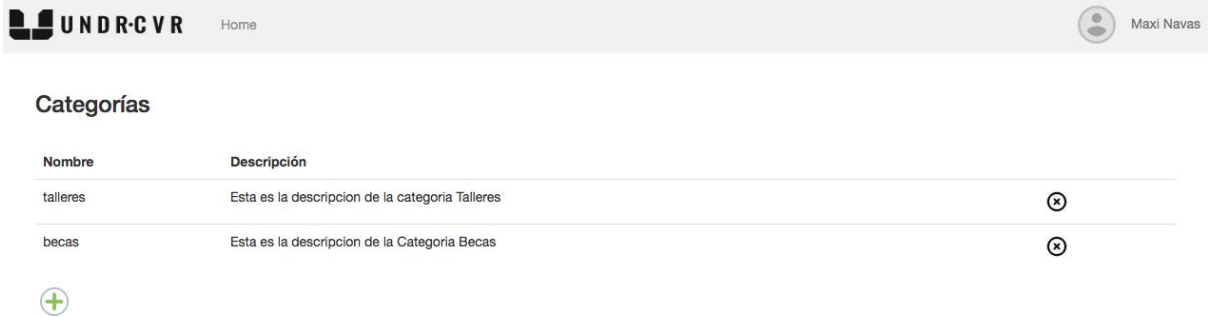


Figura 2.7

Una vez en esta pantalla, el usuario puede visualizar una tabla con las categorías cargadas, sus nombres y la descripción de cada una (Fig. 2.7). Dentro de las acciones que se pueden tomar están la de eliminar una categoría con el botón que se encuentra flotando a la derecha de la tabla y muestra un popover⁶ con dos opciones (Sí y No). Al seleccionarlo (Fig. 2.8) y agregar una nueva categoría a través del botón que se encuentra en la parte inferior izquierda de la pantalla.

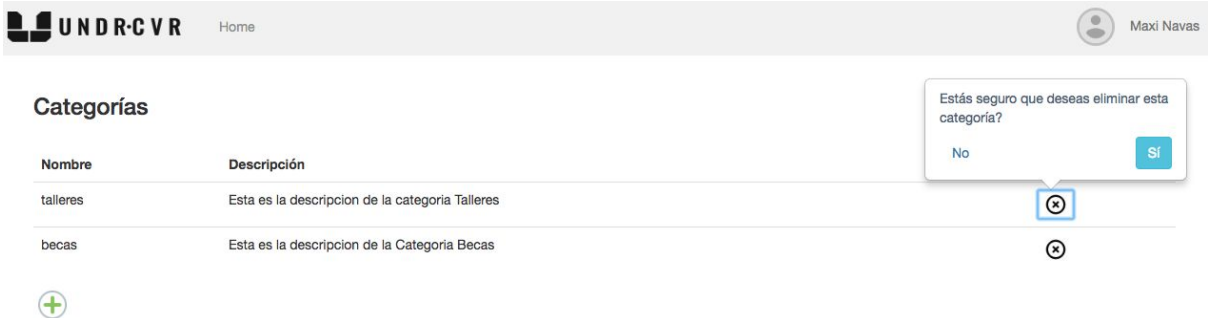


Figura 2.8

Una vez apretado se abre una ventana con un formulario (Fig. 2.9).

⁶ Ventana que se abre al presionar en otro elemento html



Figura 2.9

A su vez, el usuario también puede desloguearse presionando su nombre en la parte superior derecha, se abrirá un menú con la opción "Logout"⁷ (Fig 3.1) y seleccionando se cerrará la sesión, redirigiendo a la pantalla de login.

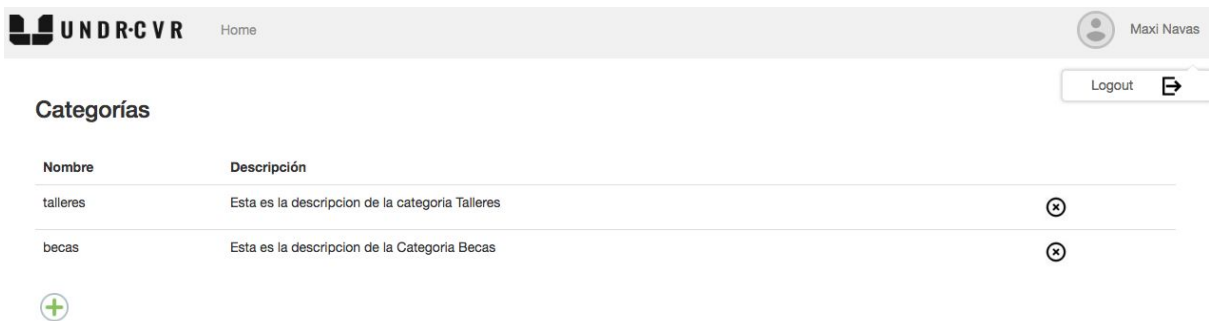


Figura 3.1

Si se presiona en cualquier de las categorías, se llevará al usuario a otra pantalla con que muestra el listado de ítems⁸ asociados (Fig. 3.2)

⁷ Significa desloguearse en Inglés

⁸ Palabras clave asociadas a una categoría determinada

Ítems de talleres

Nombre	Respuesta Asociada	Sinónimos
management	Podes encontrar información referida a talleres de management en el siguiente link: www.google.com.ar. Gracias por ponerte en contacto, espero haber sido de ayuda! Necesitás algo más?	administracion × gestion × Add a synonym...
programacion	Podes encontrar información referida a talleres de programación en el siguiente link: www.eldia.com.ar. Gracias por ponerte en contacto, espero haber sido de ayuda! Necesitás algo más?	programming × Add a synonym...



Figura 3.2

En dicha ventana se puede ingresar el nombre de la categoría (ej: Becas) y su respectiva descripción, que sirve para completar con más detalles, en caso de que el nombre ingresado no sea lo suficientemente informativo.

Si se selecciona el botón **"Cerrar"**, desaparecerá la ventana abierta y se mostrará nuevamente la tabla con las categorías. En cambio, si se completan los campos y se presiona **"Guardar"** se cerrará la ventana y se mostrará la tabla de categorías actualizada. A su vez, dicho formulario también cuenta con validaciones para sus campos, ambos son requeridos para ejecutar el guardado (Fig. 3.3)

Agregar una categoría

Nombre
Ej: Becas
El nombre de la categoría es requerido.

Descripción
Ingrese la descripción de la categoría aquí...
La descripción de la categoría es requerida.

Cerrar Guardar

Figura 3.3

En esta pantalla aparece una tabla con cada uno de los nombres de las palabras claves asociadas a la categoría que se seleccionó. Cada ítem cuenta con una respuesta asociada, y sinónimos para poder ampliar el vocabulario del chatbot.

La columna de sinónimos usa una librería para Angular 4 llamada ngx-chips⁹, que permite agregar nuevas palabras al diccionario de ítems, sin desplazarnos de dicha pantalla. A su vez, se puede eliminar cada uno de los ítems presionando en el ícono con forma de cruz (Fig. 3.4).



Figura 3.4

Para agregar un ítem nuevo se debe apretar en el botón que se encuentra en la parte inferior izquierda de la pantalla, el cual abrirá una ventana con un formulario dentro (Fig. 3.5)

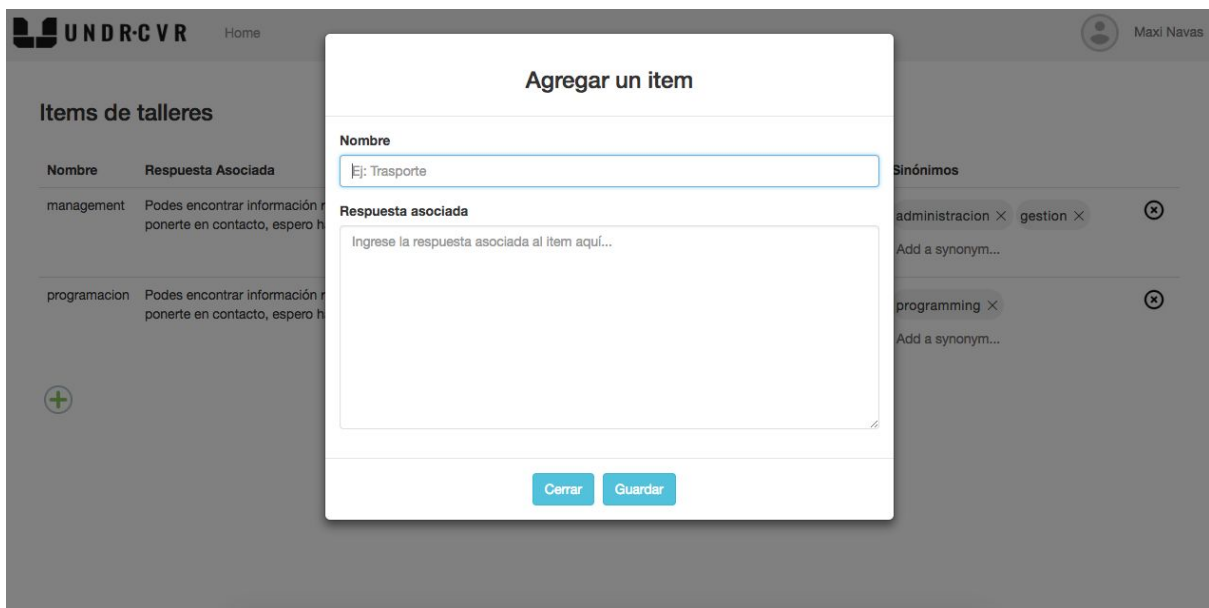


Figura 3.5

⁹ Repositorio de la librería utilizada: <https://github.com/Gbuomprisco/ngx-chips>

En esta ventana se puede ingresar el nombre del ítem (ej: Transporte) y su respectiva respuesta asociada, la cual será enviada por el chatbot cuando reconozca este ítem o algún sinónimo suyo que haya sido ingresado.

Si se selecciona el botón Cerrar, desaparecerá la ventana abierta y se mostrará nuevamente la tabla con los ítems. En cambio, si se completan los campos y se presiona guardar se cerrará la ventana y se mostrará la tabla de ítems actualizada con el nuevo ítem recientemente agregado.

A su vez, dicho formulario también cuenta con validaciones para sus campos, ambos son requeridos para ejecutar el guardado (Fig. 3.6)

The image shows a web application interface with a modal form titled "Agregar un ítem". The form is centered on the screen and has a white background. It contains two input fields. The first field is labeled "Nombre" and contains the text "Ej: Transporte". Below this field is a red error message: "El nombre del ítem es requerido.". The second field is labeled "Respuesta asociada" and contains the placeholder text "Ingrese la respuesta asociada al ítem aquí...". Below this field is another red error message: "La respuesta asociada al ítem es requerida.". At the bottom of the form, there are two buttons: "Cerrar" and "Guardar". The background of the application is dimmed, showing a table of workshop items and a list of synonyms.

Figura 3.6

Diseño del frontend del chatbot

Para que los usuarios puedan interactuar con el chatbot que desarrollamos, elegimos Facebook Messenger por factores como, la masividad que esta plataforma tiene a nivel mundial, ya que aproximadamente hay mil trescientos millones de usuarios activos mensualmente y Argentina no es la excepción en cuanto a su uso. A pesar de que hubo una caída de Facebook como red social, la mayoría de los usuarios siguen con su cuenta activa y la aplicación Messenger instalada.

Otra razón por la cual hicimos uso del chat de Facebook fue por la API que este tiene y su documentación, que es muy completa y cubría con todas nuestras necesidades para el desarrollo del chatbot.

A continuación se mostrarán algunas imágenes de una interacción entre un usuario con el chatbot dentro de la plataforma Facebook Messenger.

Se mostrarán diferentes casos de uso entre un usuario y el chatbot:

- Un usuario interactúa por primera vez y hace 1 o más consultas
- Un usuario vuelve a interactuar con el chatbot luego de haber interactuado por lo menos 1 vez.
- Un usuario hace una consulta y el chatbot no comprende su consulta

Cuando un usuario se conecta con el chatbot en la plataforma Facebook Messenger por primera vez, ve una pantalla en la cual hay un botón que dice "Empezar" (Fig. 3.7) para poder empezar a chatear con el bot de Asuntos Estudiantiles de la UNLP:

Tesis Becas 2018

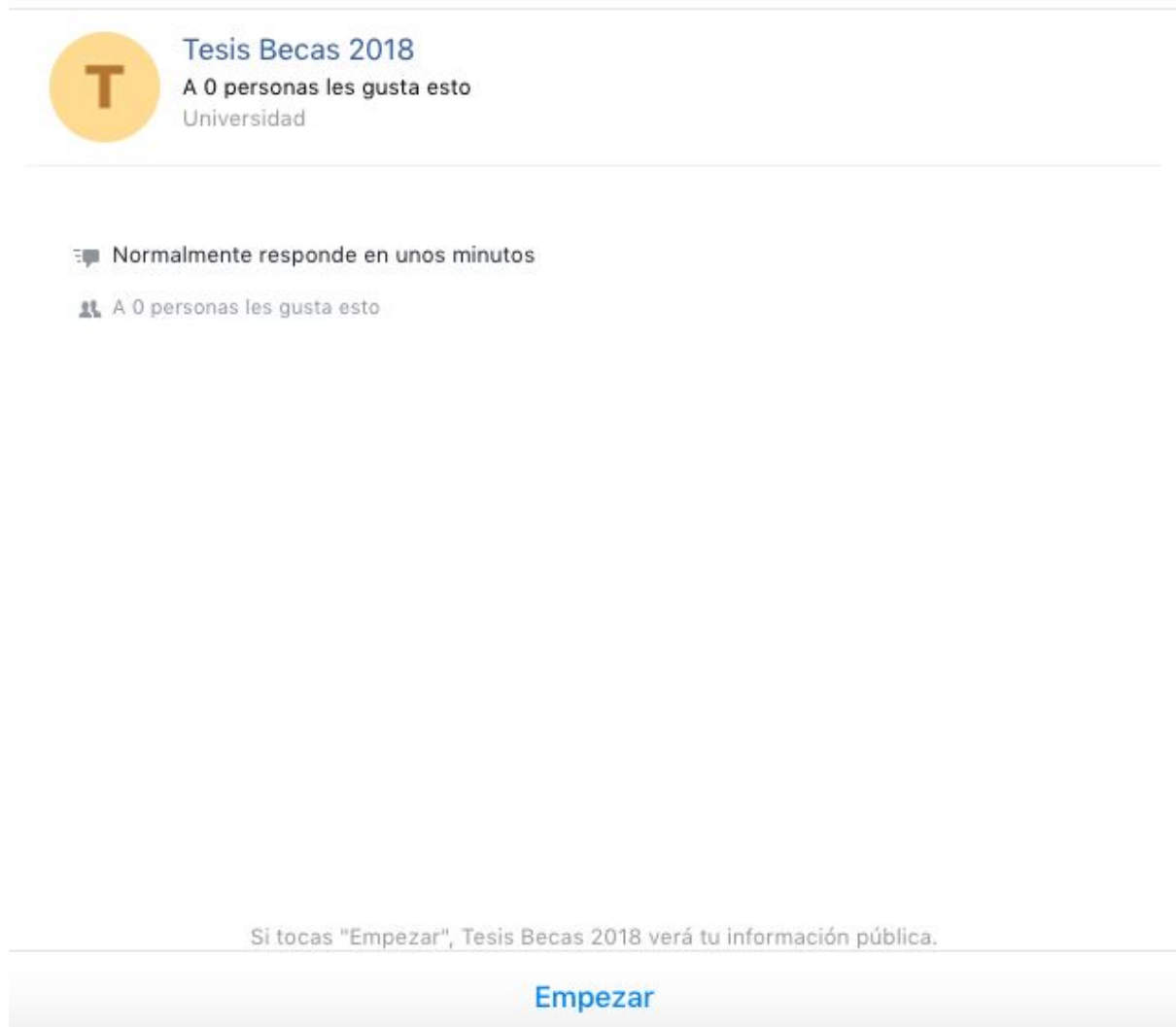


Figura 3.7

Cuando el usuario en cuestión presiona "Empezar", automáticamente el bot le responde con un mensaje de bienvenida personalizado con su nombre, y otros tres botones (Fig. 3.8):

1. "Empecemos": Si es presionado el bot le enviará una serie de temas en forma de respuestas rápidas para orientarlo al usuario sobre los tópicos que puede hacer consultas.

2. "Twitter Bienestar": Abre una nueva pestaña en el navegador con la cuenta de Twitter de Bienestar Estudiantil de la UNLP.
3. "Instagram Bienestar": Abre una nueva pestaña en el navegador con la cuenta de Instagram de Bienestar Estudiantil de la UNLP.

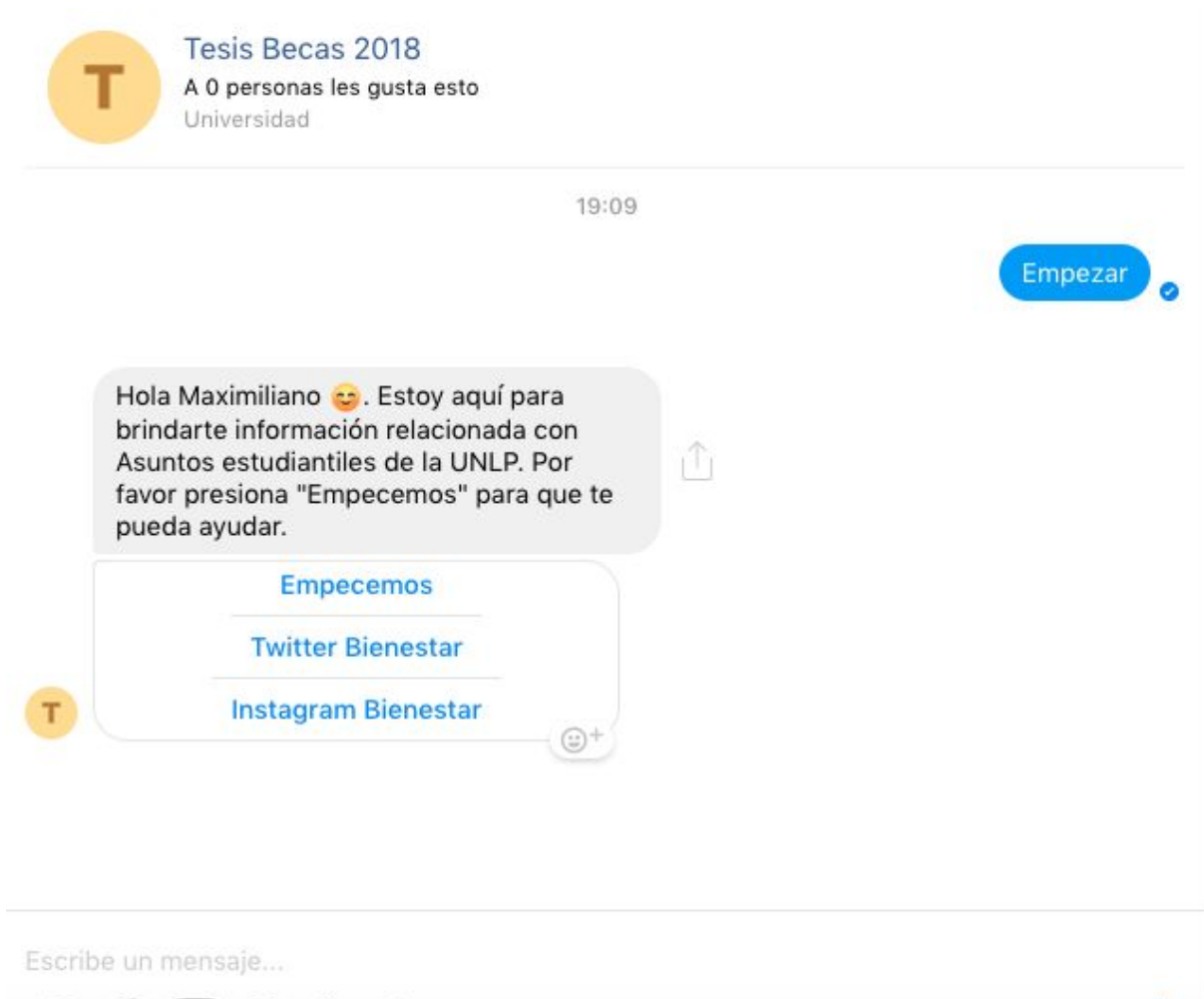


Figura 3.8

Si el usuario presiona "Empecemos" el bot le mostrará los temas que estén cargados en el BackOffice, los cuales, como se mencionó anteriormente, le sirven para orientar al usuarios sobre qué tipos de consultas podrán realizarle (Fig. 3.9):



Figura 3.9

En este caso de primer uso, vamos a suponer que elige y presiona la opción "becas":

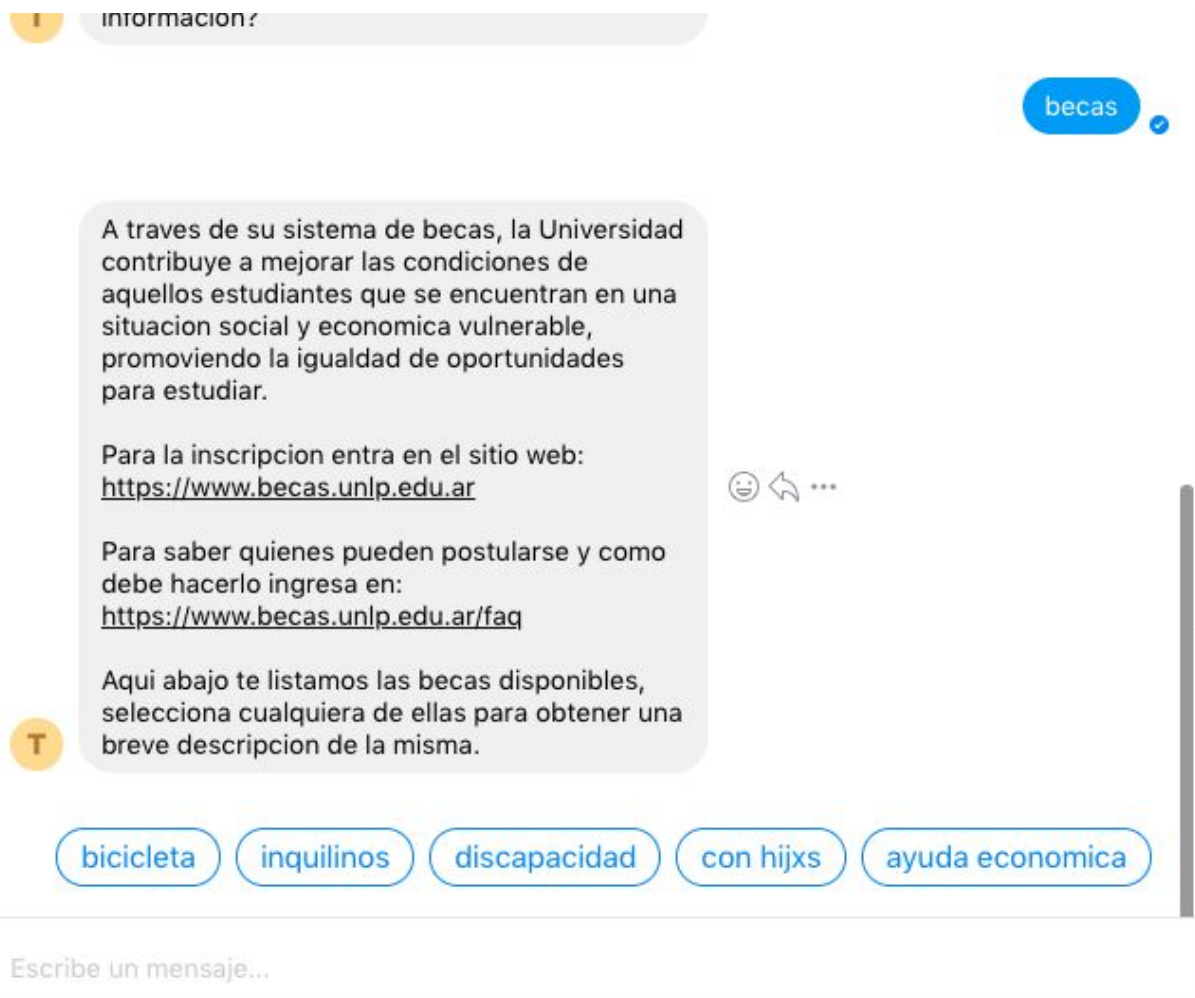


Figura 4.1

A continuación, el chatbot le envía al usuario un mensaje con información relacionada con la opción "becas" (Fig. 4.1), que posee links que, en caso de ser presionados, lo llevarán a sitios externos y una serie de opciones para guiarlo con becas específicas que puede seleccionar para obtener datos más específicos sobre lo que está buscando.

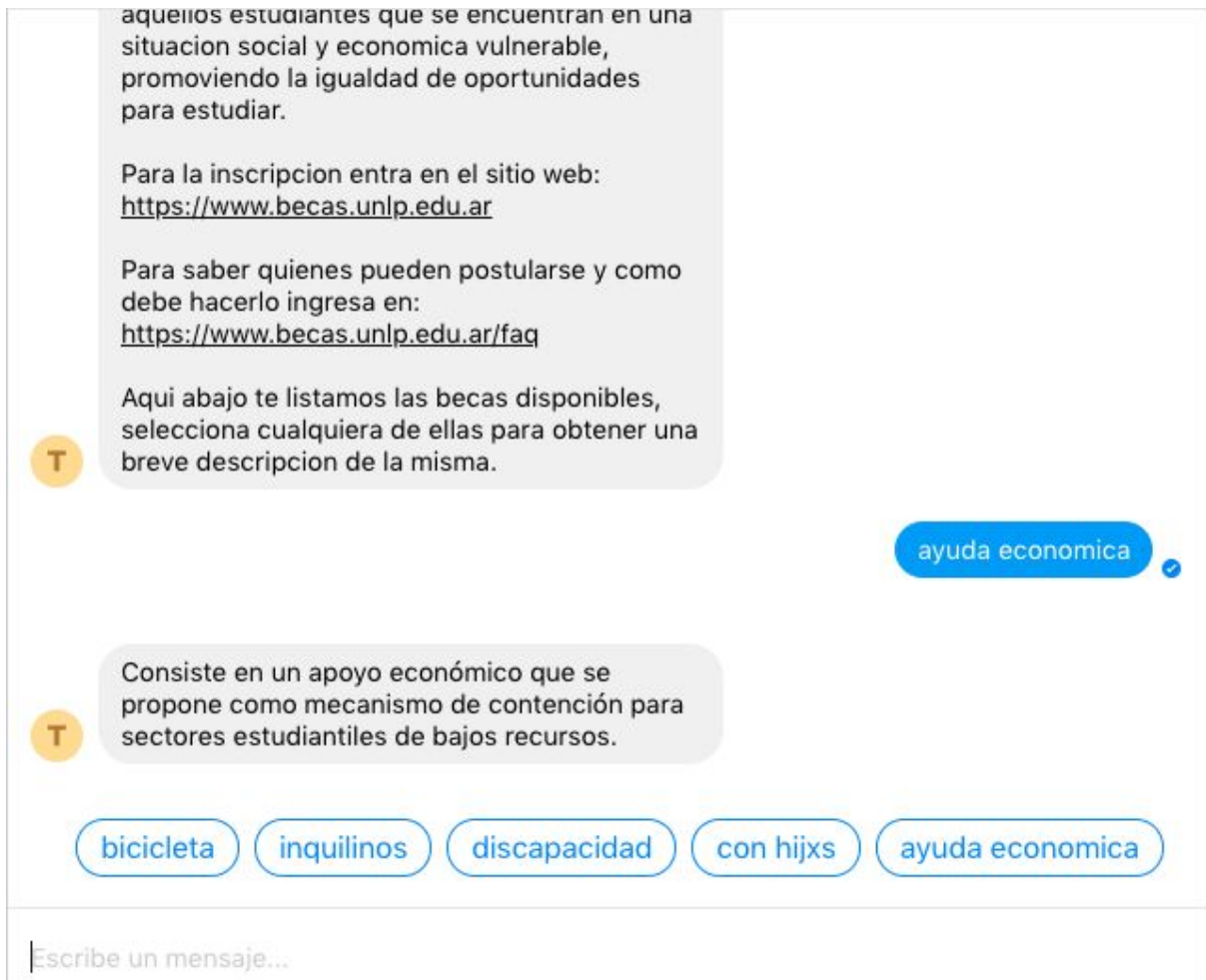


Figura 4.2

Si el usuario llegara a seleccionar la opción "ayuda económica", el chatbot le enviará información relacionada a dicha elección (Fig. 4.2). Es simplemente una pequeña explicación para orientar al usuario sobre de qué trata dicha beca, ya que lo importante son los links en donde podrá inscribirse a través de sitios externos.

En cuanto a la navegación del chat, es importante destacar que una vez que el usuario selecciona una opción del último nivel del árbol de respuestas, el chatbot le envía las opciones nuevamente que están en ese mismo nivel. De esta manera, nunca se pierde el flujo de la conversación y el usuario no se frustra.

Aquí en donde podemos introducirnos en el caso de uso número dos, en el cual el usuario ya ha interactuado por lo menos una vez con el chatbot. Dicho caso de uso se puede dar por dos motivos:

- El usuario ha llegado al último eslabón de respuestas del chatbot y necesita información sobre otro tema para seguir interactuando.
- El usuario ha llegado al último eslabón, abandona el chatbot y tiempo después vuelve a abrir la conversación para buscar información sobre otro tema en particular.

Ambos subcasos derivan en una misma acción que debe realizar el usuario, quien debe escribir una consulta para que el chatbot lo derive al nivel del árbol de respuestas que esté relacionado con sus necesidades.

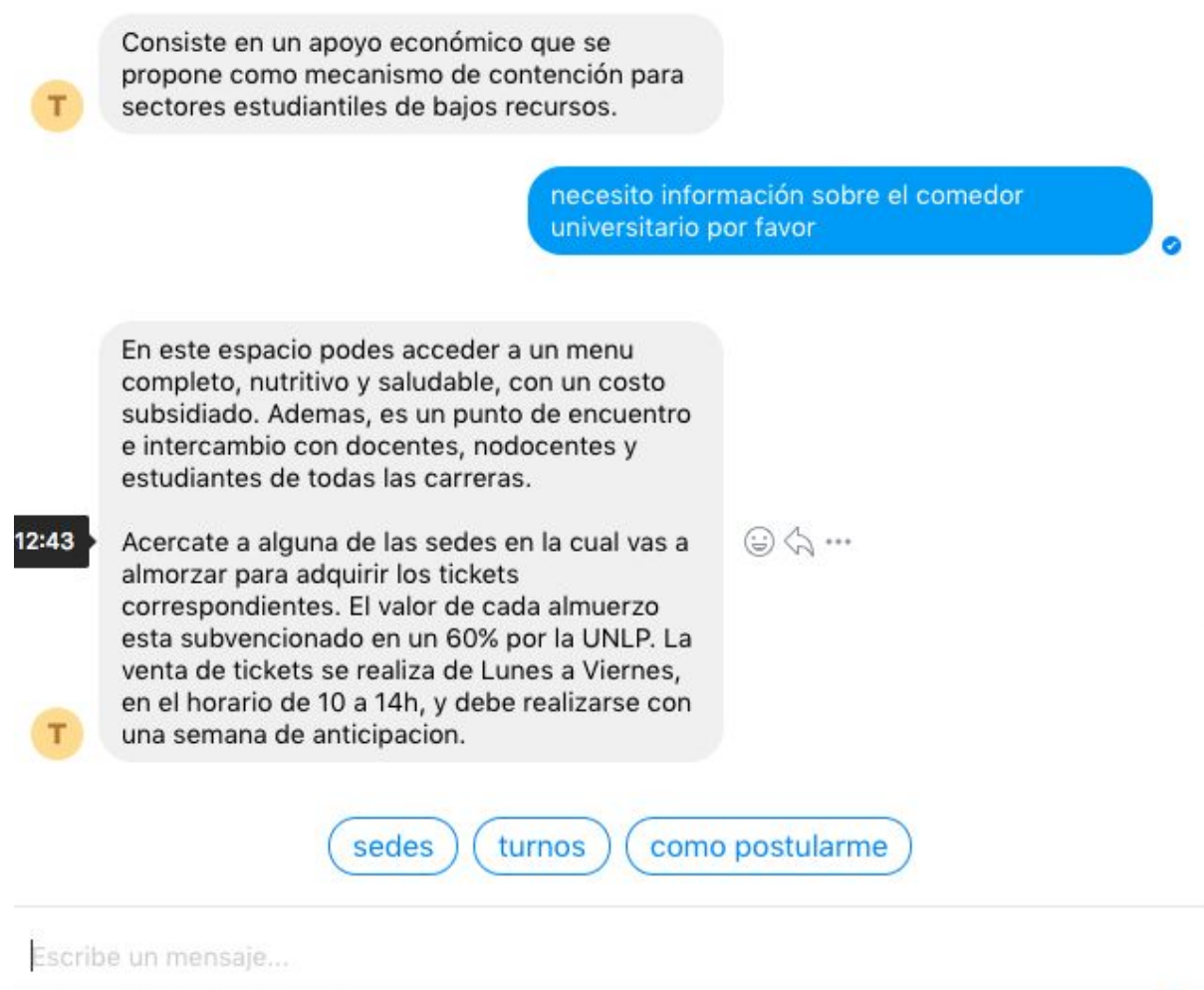


Figura 4.3

En este ejemplo en particular (Fig. 4.3), el usuario venía haciendo consultas sobre el temas becas, y como quería información sobre el comedor universitario escribió "necesito información sobre el comedor universitario por favor". El chatbot entendió su pedido y le respondió con algo de información y tres opciones específicas para ayudar al usuario a encontrar lo que necesita.

T una semana de anticipacion.

15:47

sedes

Para acceder, si sos ingresante, debés presentar una foto y tu certificado de inscripción a la UNLP. Si ya sos estudiante y vas por primera vez, tenés que llevar una foto carnet y un certificado de alumno regular.

El comedor cuenta con cuatro sedes, ubicadas en zonas de facultades:

- Sede I Bosque Oeste: calle 50 e/ 116 117.
- Sede II Bosque Este: en el boulevard 120 e/ 61 y 62 nro 1439.
- Sede III ATULP: av. 44 nro 733 e/ 9 y 10, Asociación de Trabajadores de la UNLP.
- Sede IV Club Everton: calle 14 e/ 63 y 64, salón planta baja.

T

sedes

turnos

como postularme

Escribe un mensaje...

Figura 4.4

El usuario apretó la opción "sedes" (Fig. 4.4), el chatbot lo respondió con información sobre todas las instalaciones con las que cuenta la UNLP para los comedores universitarios. Además, le volvió a mostrar las opciones en el caso que quiera seguir indagando sobre el mismo tema.

presentar una foto y tu certificado de inscripción a la UNLP. Si ya sos estudiante y vas por primera vez, tenés que llevar una foto carnet y un certificado de alumno regular.

El comedor cuenta con cuatro sedes, ubicadas en zonas de facultades:

- Sede I Bosque Oeste: calle 50 e/ 116 117.
- Sede II Bosque Este: en el boulevard 120 e/ 61 y 62 nro 1439.
- Sede III ATULP: av. 44 nro 733 e/ 9 y 10, Asociación de Trabajadores de la UNLP.
- Sede IV Club Everton: calle 14 e/ 63 y 64, salón planta baja.

T

me gustaría información sobre el curso de ingreso a la facultad de informática

T

Disculpa pero no te he entendido, intenta de nuevo por favor seleccionando algunas de la siguientes opciones:

salud

talleres

comedor

albergue

becas

Escribe un mensaje...

Figura 4.5

El último caso de uso planteado lo podemos ver en la última imagen (Fig. 4.5), en la cual el usuario hizo uso del texto libre del chatbot y escribió "me gustaría información sobre el curso de ingreso a la facultad de informática". Como actualmente el backoffice no está cargado con dicho tema, el chatbot no comprendió la consulta, le dio una respuesta acorde y además le entregó un listado de opciones sobre los temas que actualmente le puede brindar información, ya que todos están debidamente cargados en el backoffice.

Testing

En esta sección comentaremos el testing realizado y los resultados obtenidos.

Para llevar a cabo las pruebas antes mencionadas, creamos una única encuesta que entregamos a cada encuestado.

Por temas de organización, realizamos 2 conjuntos de encuestas:

- Por un lado encuestamos a Tomás, un estudiante ciego. Con él pudimos buscar problemas de interacción y nos dio un feedback sobre la accesibilidad de la herramienta.
- Por otro lado, realizamos encuestas a 13 personas, sin discapacidades, las cuales dividimos en 2 grupos, uno de 6 y otro de 7. Los encuestados son de diferentes edades, entre 19 y 57 años, de ambos sexos.

Para el número de encuestados nos basamos en la teoría de Nielsen, quien afirma que *“los mejores resultados de las pruebas de usabilidad provienen de ensayos de no más de 5 usuarios y la ejecución de tantas pruebas pequeñas como sea posible”* (Nielsen, Jakob, y Landauer, Thomas K. 1993).

En su investigación, Nielsen junto con Tom Landauer muestran que la cantidad de problemas de usabilidad encontrados en una prueba de usabilidad con n usuarios es:

$$N (1 - (1 - L)^n)$$

donde:

- N es el número total de problemas de usabilidad en el diseño
- L es la proporción de problemas de usabilidad descubiertos al probar a un solo usuario.

En dicho análisis resuelven que el valor típico de L es del 31%.

Al trazar la curva para $L = 31\%$ se obtiene el siguiente resultado:

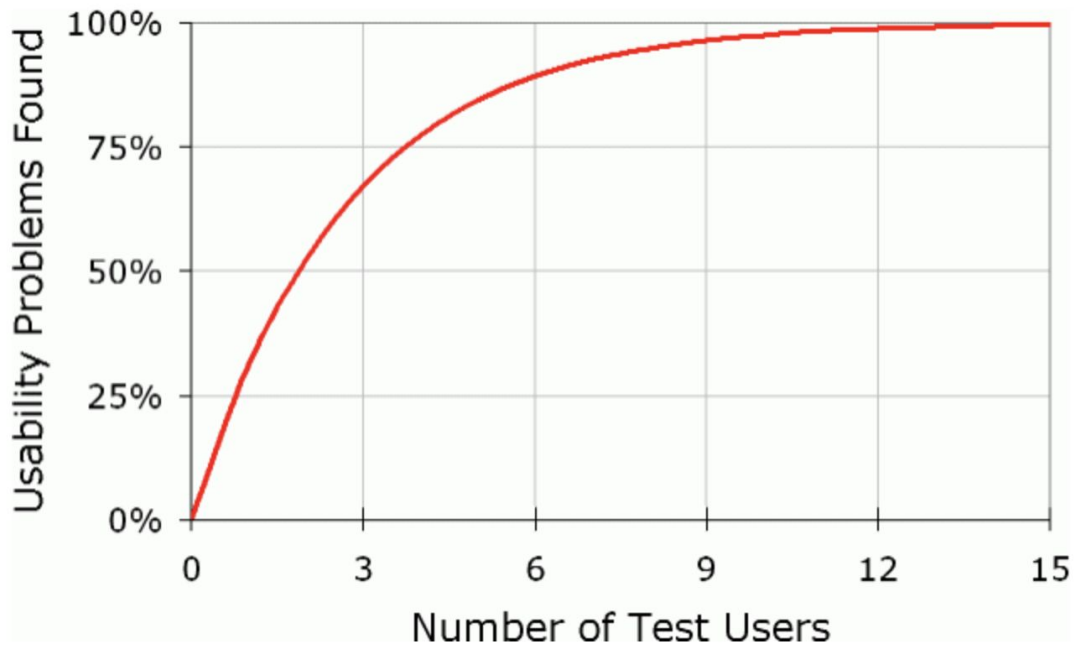


Figura 4.6. Number of test subjects or heuristic evaluators

Como se puede observar en la curva (Fig. 4.6) para cero usuarios, los problemas de usabilidad encontrados resultan ser cero por ciento .

Tan pronto como recopila datos de un solo usuario de prueba, la curva se dispara y se descubren un tercio de los problema de usabilidad. La diferencia entre cero y un usuario es asombrosa.

Al probar al segundo usuario, se repiten comportamientos del primero, por lo que hay algo que se superpone en lo que aprende el segundo. Las personas son definitivamente distintas, por lo que también habrá algo nuevo que el segundo usuario haga y que no haya observado con el primero. Así que el segundo usuario agrega cierta cantidad de información nueva, pero no tanto como lo hizo el primer usuario.

El tercer usuario hará muchas cosas que ya ha observado con el primero o con el segundo usuario e incluso algunas cosas que ya ha visto dos veces. Además, por supuesto, el tercero generará una pequeña cantidad de datos nuevos, aunque no tanto como el primero y el segundo usuario.

A medida que se agregan más y más usuarios, obtendremos cada vez menos información debido a que se seguirá viendo los mismos resultados una y otra vez. No hay una necesidad real de seguir observando lo mismo varias veces.

Después del quinto usuario, está perdiendo el tiempo observando los mismos hallazgos repetidamente sin aprender mucho de nuevo.

Por lo tanto, si bien entendemos que el número óptimo que Nielsen aconseja es de 5 personas, buscaremos realizar un testing de 14 usuarios totales, adquiriendo así un porcentaje superior a 95%.

Para dar fundamento a la accesibilidad de nuestra tesis, encuestamos a una persona no vidente.

Pruebas de usabilidad

Para las pruebas de usabilidad nos decidimos por Google Forms, el cual nos brindó una interfaz amigable y todas las herramientas necesarias para diagramar el formulario que queríamos realizar.

Dividimos la encuesta en 4 secciones:

1. Perfil del encuestado
2. Actividad
3. Preguntas sobre el uso del chatbot
4. Recomendaciones

Perfil del encuestado

Encuesta sobre uso del Chatbot

*Obligatorio

Nombre y Apellido *

Tu respuesta

Edad *

Tu respuesta

Perfil del encuestado

Sabe ud qué es un chatbot? *

Sí

No

Ha interactuado en alguna ocasión con un chatbot? *

SI

No

Conoce de alguna tecnología similar? *

Sí

No

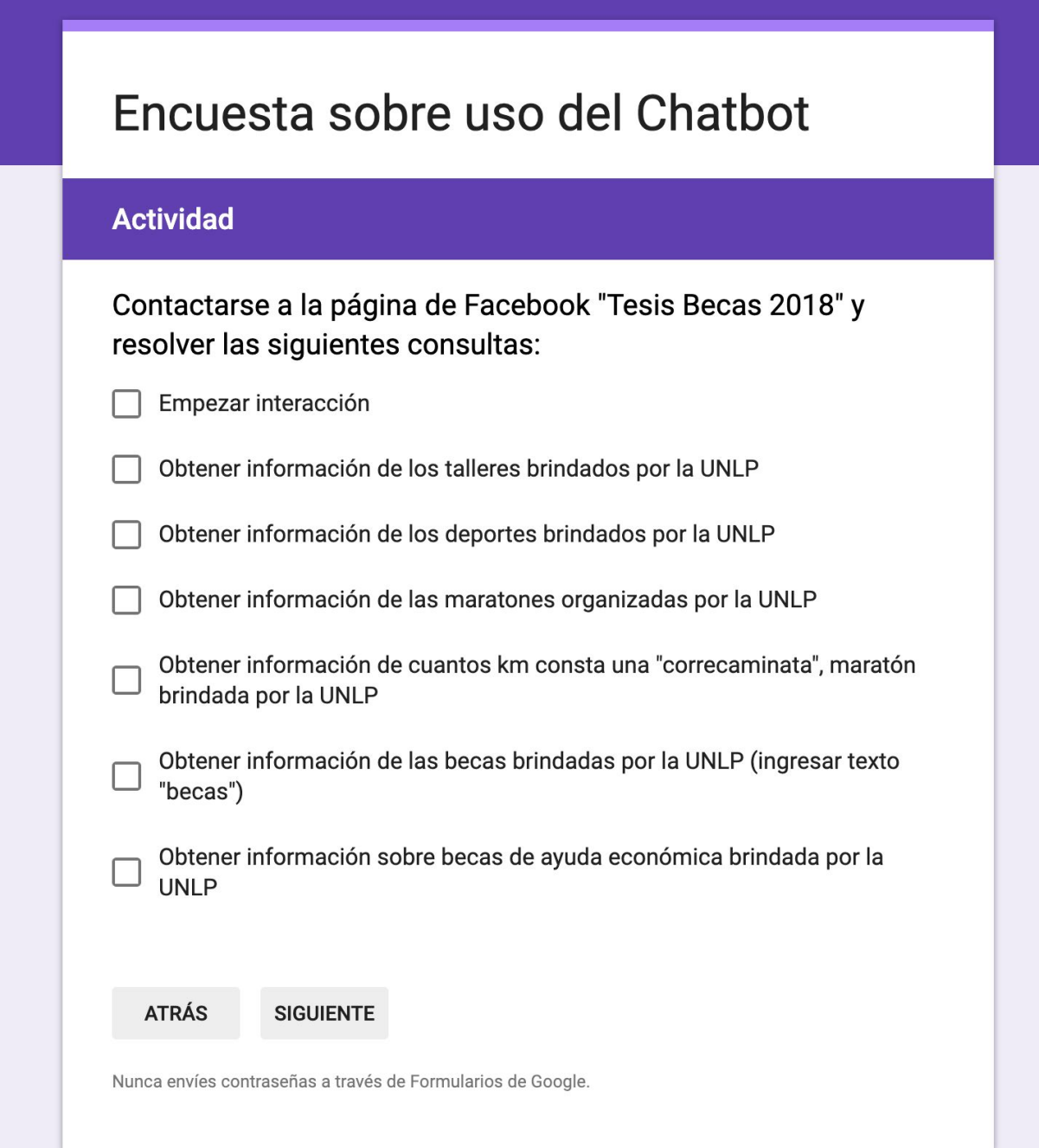
SIGUIENTE

Nunca envíes contraseñas a través de Formularios de Google.

Figura 4.7

En dicha sección (Fig. 4.7) solicitamos al encuestado sus datos personales y realizamos preguntas básicas sobre la tecnología, con el objetivo de saber si estamos ante una persona con o sin conocimiento del tema.

Actividad



Encuesta sobre uso del Chatbot

Actividad

Contactarse a la página de Facebook "Tesis Becas 2018" y resolver las siguientes consultas:

- Empezar interacción
- Obtener información de los talleres brindados por la UNLP
- Obtener información de los deportes brindados por la UNLP
- Obtener información de las maratones organizadas por la UNLP
- Obtener información de cuantos km consta una "correcaminata", maratón brindada por la UNLP
- Obtener información de las becas brindadas por la UNLP (ingresar texto "becas")
- Obtener información sobre becas de ayuda económica brindada por la UNLP

ATRÁS **SIGUIENTE**

Nunca envíes contraseñas a través de Formularios de Google.

Figura 4.8

En dicha sección (Fig. 4.8) pedimos al encuestado que realice breves actividades interactuando con el chatbot.

Colocamos un checkbox al lado de cada actividad, con el fin de que pueda ir marcando las actividades completadas.

Estimamos que el tiempo de realización de las actividades no supere los 10 minutos.

Preguntas sobre el uso del chatbot

Encuesta sobre uso del Chatbot

*Obligatorio

Preguntas sobre el uso del chatbot

Las actividades fueron claras? *

Sí

No

En caso negativo. Que inconvenientes tuvo?

Tu respuesta

Pudo resolver todas las consultas? *

Sí

No

En caso negativo. Que consulta no pudo resolver?Cuál fue el inconveniente?

Tu respuesta

En la interacción con el chatbot, respondió utilizando: *

Respuestas automáticas

Escribió sus respuestas

Ambas

Cuán satisfecho esta con las respuestas brindadas por el chatbot? *

Muy satisfecho

Satisfecho

Conforme

Defraudado

Muy defraudado

Cree que será util la implementación del chatbot? *

Sí

No

[ATRÁS](#) [SIGUIENTE](#)

Nunca envíes contraseñas a través de Formularios de Google.

Figura 4.9

A partir de la sección de actividades, creamos una nueva en la que realizamos consultas sobre la interacción con el chatbot (Fig. 4.9).

Buscamos captar la claridad en cuanto a las actividades y los enunciados, como también el grado de satisfacción que tuvo el encuestado.

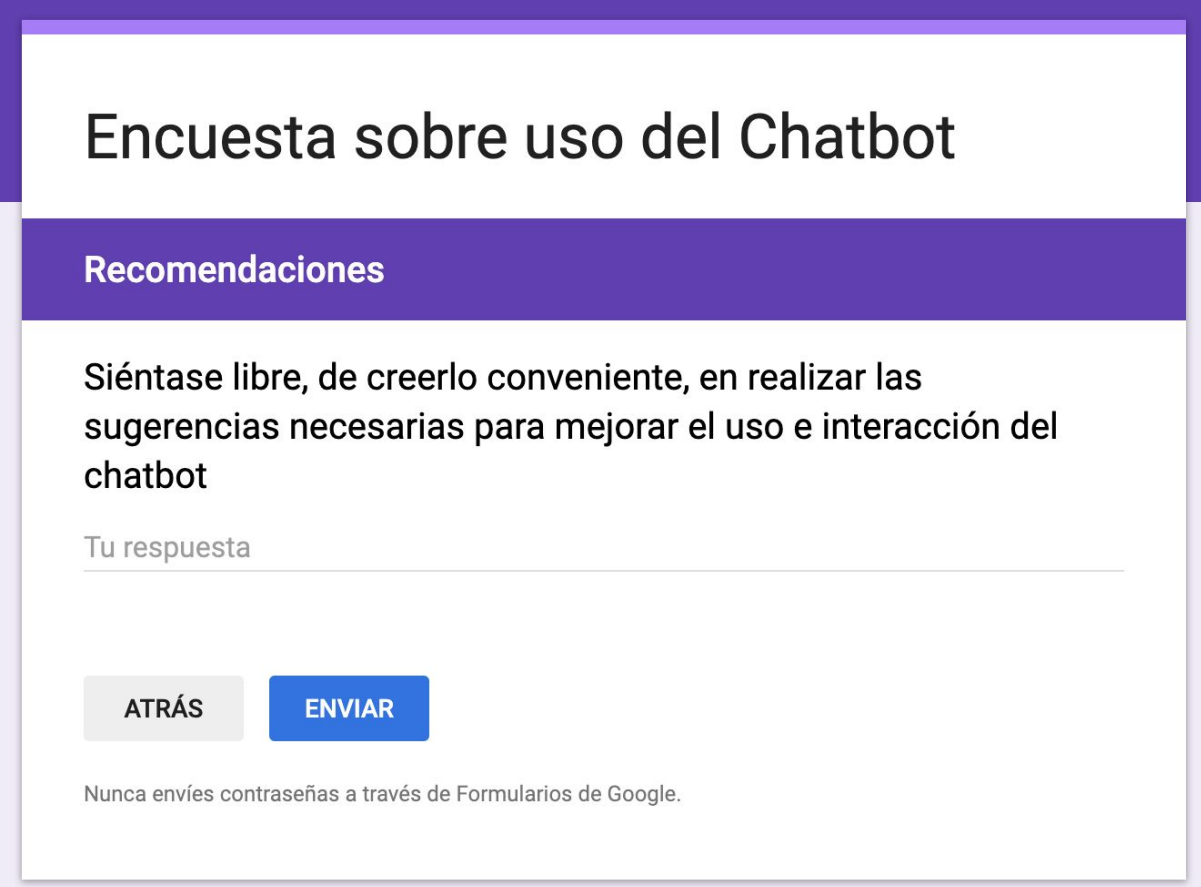
En dicha sección incluimos una pregunta la cual consideramos muy importante:

“En la interacción con el chatbot, respondió utilizando:”

A partir de esta consulta comprenderemos donde debemos situar el foco para la comprensión de la entrada de los futuros usuarios:

- Si la mayoría de los encuestados opta por "respuestas rápidas", eso no indicará que debemos poner énfasis en el armado del flujo de nuestro chatbot, para así brindarle todas las herramientas al cliente, con el objetivo de que llegue a la información final.
- En el caso de que los encuestados respondan -en su mayoría- “Escribió sus respuestas” debemos poner énfasis en el procesamiento de lenguaje natural.

Recomendaciones



The image shows a survey form with a purple header and a white body. The title is 'Encuesta sobre uso del Chatbot'. Below the title is a purple bar with the word 'Recomendaciones' in white. The main text asks the respondent to provide suggestions for improving chatbot use. There is a text input field labeled 'Tu respuesta' and two buttons: 'ATRÁS' (grey) and 'ENVIAR' (blue). At the bottom, there is a disclaimer: 'Nunca envíes contraseñas a través de Formularios de Google.'

Figura 5.1

Por último, creamos la sección de Recomendaciones (Fig. 5.1), la cual brinda al encuestado la posibilidad de darnos un pequeño feedback, sugiriendo posibles mejoras para el uso o interacción con el chatbot.

Resultados

Teniendo en cuenta que llegamos a una versión estable del chatbot, se acordó una sesión formal de pruebas, focalizadas en la usabilidad que brinda Facebook Messenger. Hemos encuestado a 14 personas, dentro de las cuales se encuentra Tomás Falco, un chico ciego.

Perfil del encuestado

Las edades variaron entre los 19 y 57 años.

Edad

14 respuestas

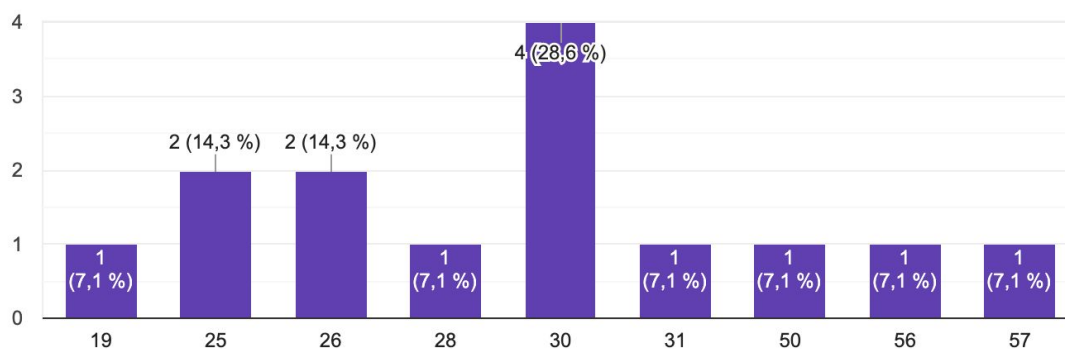


Figura 5.2

Buscamos diversidad en este punto, ya que creemos que es de suma importancia que cualquier persona pueda interactuar con el chatbot, dando una posibilidad a todos los usuarios de indicarnos un feedback, en caso de que crean correspondiente. Así ampliamos el universo de recomendaciones.

Las preguntas relacionadas con el conocimiento inicial de la tecnología arrojaron los siguientes resultados:

Sabe ud qué es un chatbot?

14 respuestas

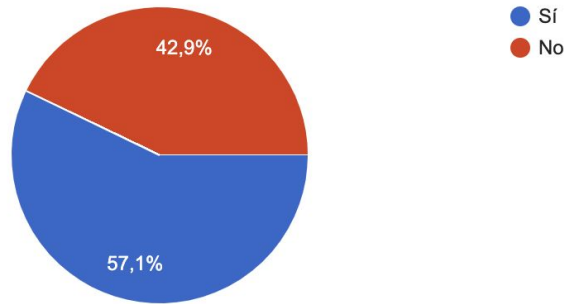


Figura 5.3

Ha interactuado en alguna ocasión con un chatbot?

14 respuestas

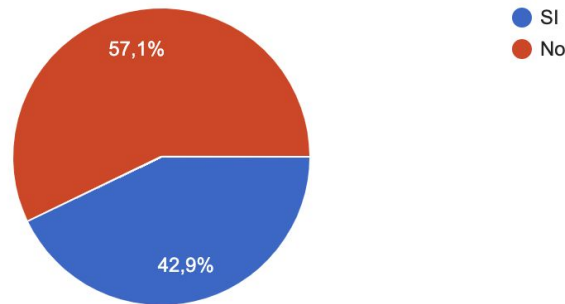


Figura 5.4

Basados en la resultante de las 2 preguntas indicadas (Fig. 5.3 y 5.4), entendemos que si bien hay encuestados que ya han interactuado conscientemente con un chatbot, existe un pequeño grupo (14,2%) que sabe lo que es un chatbot, pero aún no han interactuado con uno, denotando que si bien el concepto es relativamente conocido, es una tecnología nueva para muchas personas.

Conoce de alguna tecnología similar?

14 respuestas

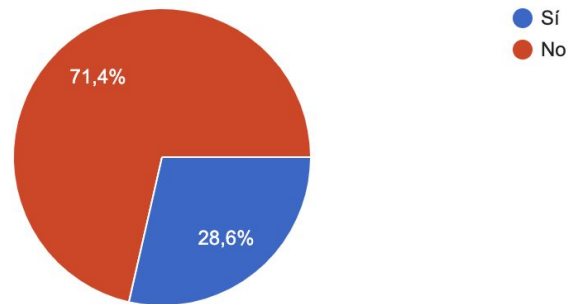


Figura 5.5

Actividad

Ingresando en la sección de la actividad podemos observar:

Contactarse a la página de Facebook "Tesis Becas 2018" y resolver las siguientes consultas:

14 respuestas

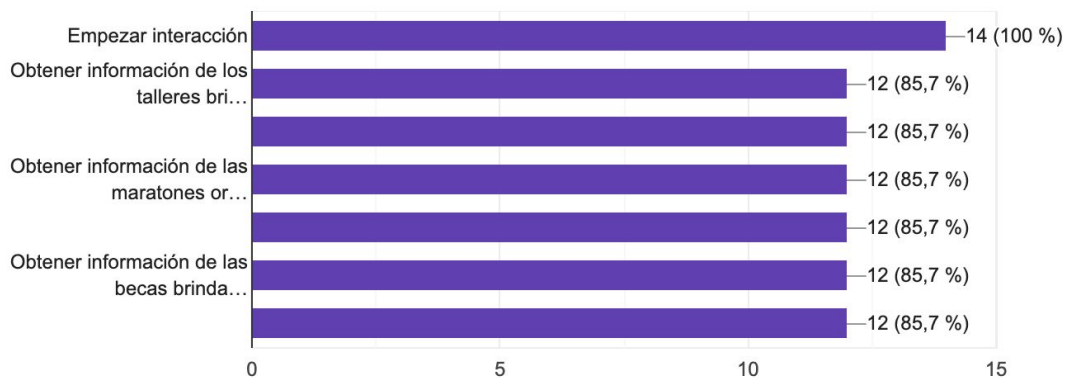


Figura 5.6

Ejercicios para el uso del chatbot

En referencia a la imagen anterior (Fig 5.6), si bien hay 2 personas que no clickearon en algunas de los ejercicios dados, en el gráfico de torta siguiente podrán observar que todos pudieron completar las actividades:

Pudo resolver todas las consultas?

14 respuestas

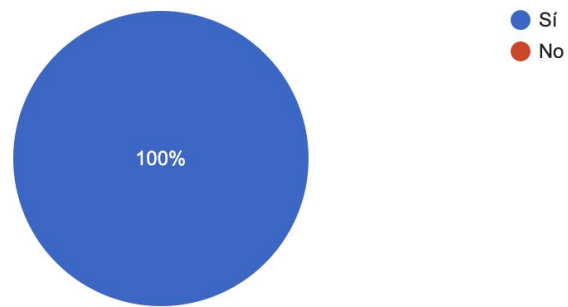
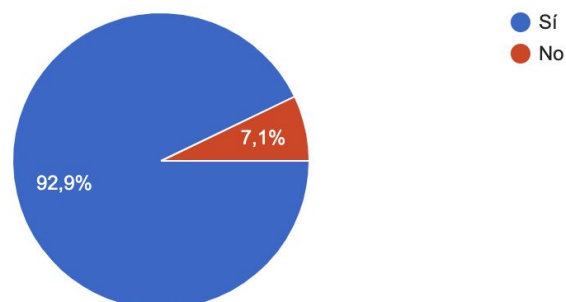


Figura 5.7

La idea de los ítems era solo un ayuda-memoria para que el encuestado pueda ir avanzando, sin tener que releer actividades.

Las actividades fueron claras?

14 respuestas



En caso negativo. Que inconvenientes tuvo?

2 respuestas

desde la página de facebook no pude visualizar las respuestas rápidas, pero las fui pulsando desde la app móvil

como usuario pienso que hace falta un acceso directo permanente a los temas principales, de manera tal de facilitar la navegación por toda la gama de herramientas. El motivo de ingresar EMPEZAR cada vez que se quiere ingresar una consulta distinta, dificulta su utilidad.

Figura 5.8

Como se puede observar (Fig 5.8), 2 personas nos indicaron que tuvieron problemas para realizar las actividades:

- En primer lugar se encuentra Tomás. En su comentario explicó que su lector de pantalla no lee las respuestas rápidas de Facebook Messenger en sus versiones desktop. Hizo el intento tanto en m.facebook.com como en messenger.com. Para solventar el problema utilizó la aplicación móvil.
- En segundo lugar nos aconsejaron agregar una referencia al inicio. En una primera etapa nos pareció que esto rompía con el hilo del chatbot, pero -basados en las recomendaciones recibidas-, optamos por realizar un ajuste, dando la posibilidad al usuario de volver al inicio en cualquier punto de la conversación. Para esto dimos la posibilidad al usuario de escribir las siguientes palabras claves en cualquier punto del flujo:
 - empezar
 - empecemos
 - comenzar
 - comencemos
 - comienzo
 - inicio

- iniciar
- iniciemos

Por ejemplo: Si escribimos la frase "Volver al inicio", el bot nos responderá con los temas sobre los cuales nos puede brindar información (Fig 5.9).



Figura 5.9

También agregamos el botón **“volver”** (Fig. 6.1), a nivel de ítem, para que los usuarios tengan la posibilidad de regresar al inicio simplemente con un click. De esta manera puede iniciar el flujo de la conversación nuevamente.

Ejemplificamos con las siguientes imágenes:

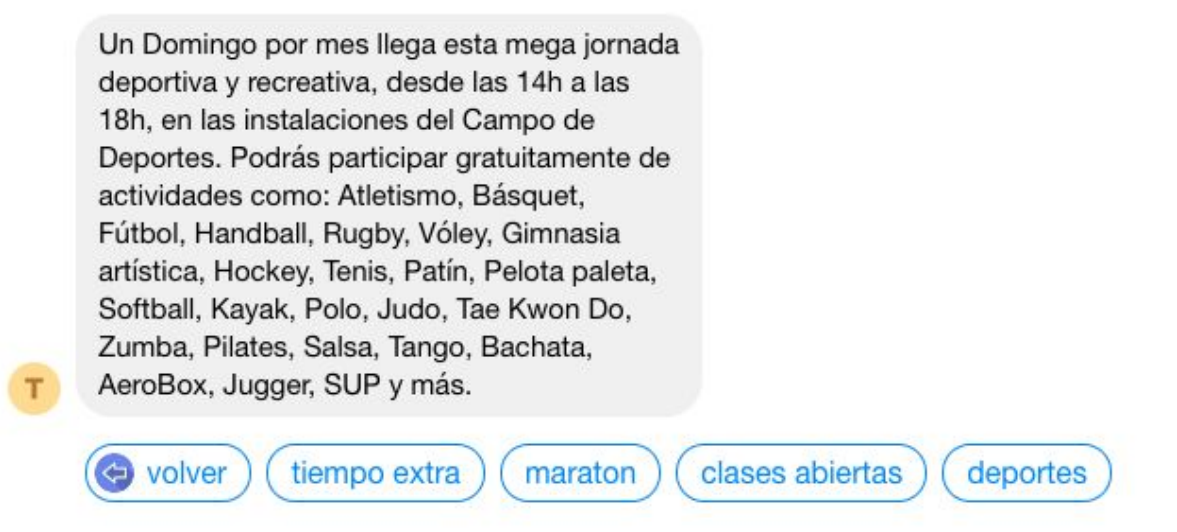


Figura 6.1

Al clicar en volver, el usuario verá el siguiente mensaje:



Figura 6.2

En la interacción con el chatbot, respondió utilizando:

13 respuestas

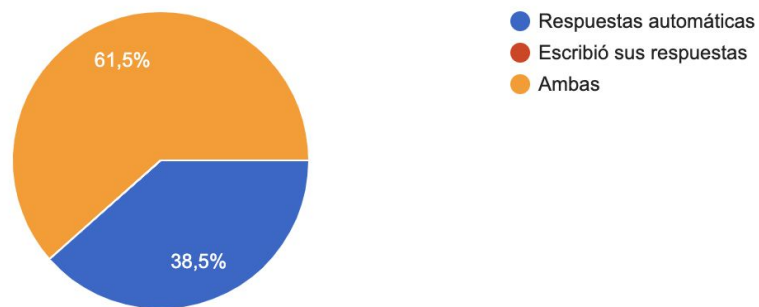


Figura 6.3

Si bien hay un 38.5% de los encuestados que respondieron que solo utilizaron "Respuestas automáticas", la actividad demandaba que en un punto ingrese por teclado una consulta:

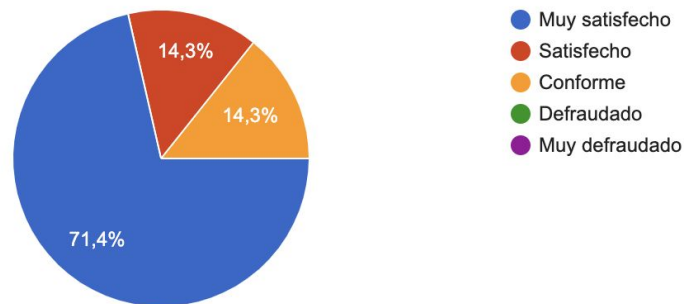
Obtener información de las becas brindadas por la UNLP (ingresar texto "becas")

Entendemos que esta pregunta no refleja la información correcta.

La siguiente imagen (Fig. 6.4) refleja el grado de satisfacción de los encuestados:

Cuán satisfecho esta con las respuestas brindadas por el chatbot?

14 respuestas



Cree que será útil la implementación del chatbot?

14 respuestas

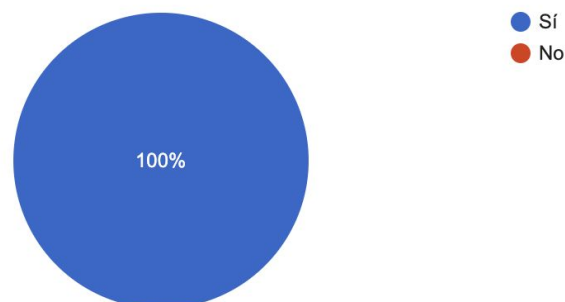


Figura 6.4

Recomendaciones

En la siguiente sección (Fig. 6.5 y 6.6), dimos libertad a los encuestados para expresar -de creer correspondiente- sus recomendaciones.

Nueve de los 14 encuestados decidieron indicar algunos comentarios:

Siéntase libre, de creerlo conveniente, en realizar las sugerencias necesarias para mejorar el uso e interacción del chatbot

9 respuestas

se puede agregar la información sobre los diferentes transportes de la unlp como micro, tren y demás, que si bien está en las páginas de bienestar, como toda la información cuesta encontrarla y a veces es confusa. La interacción con respuestas automáticas se trabaja mejor desde la aplicación móvil, al menos en Android, y me respaldé en la página móvil.

<https://m.facebook.com> para escribir las diferentes consultas ya que permite mayor flexibilidad, pero no lee las respuestas rápidas. Podría también tener algo de info de parciales y finales, o en todo caso algo mas fácil, que redirija a la web de cada cátedra, y ahí el alumno busque la información que esté colgada en la página.

Estaría bueno tener la posibilidad de ingresar a la etapa anterior sin utilizar una respuesta incorrecta como herramienta para llegar a tal objetivo. Sería conveniente tener una opción que de la posibilidad de volver.

La sugerencia fue ingresada en la respuesta 1.

Probe usando la palabra "becas" y "beca". Creo que en este caso debería mostrarme la misma info sobre las becas sin importar si la escribo en plural o singular. Debería entender que estoy buscando información sobre las becas.

Darle publicidad a la herramienta para que sea útil y todo el mundo la pueda usar

Es claro y sencillo de usar. Quizás puede resultar un poco difícil al principio si no se sabe bien hasta donde se puede preguntar

-falta una palabra clave que permita volver al principio de la conversación (pienso en los casos que las personas tengan mas de una duda), y no que suceda por poner algo que no reconoce, también es raro que en caso de escribir erróneamente una palabra, vuelva al comienzo y no desde el ultimo punto
-en algunas cuestiones creo que la información podría ser un poco mas detallada que solo un link, por ejemplo en deportes podría presentar la lista de los que se practican en la universidad (igual en la mayoría de los casos me parecen bien las respuestas)
-es medio raro depender todo el tiempo de palabras claves que vayan apareciendo a medida que las voy apretando, una vez que probas un poco empezas a aprendértelas y se hace mas fácil, pero me parece que estaría bueno que potenciales usuarios puedan hacer preguntas que si contienen esas palabras, les entregue la respuesta correcta (por ej: envié un mensaje que decía "quiero saber de los talleres" y no pudo responder, pero cuando pregunte "hay beneficio para inquilinos" si me dio la respuesta directamente)
-siguiendo con la línea de lo anterior, creo que funcionaría agregar palabras relacionadas que también cumplan función, un caso específico es en becas cuando quise escribir hijos o hijas no pudo responder
-dentro de las primeras palabras (salud, talleres, albergue,...) podría llegar a haber alguna que presente información general de la unlp, pudiendo ser dirección del edificio central, contacto, etc
-si la idea es que siempre se manejen con las palabras que van apareciendo, creo que deberían estar siempre "atrás" e "inicio".

Figura 6.5

excelente herramienta rápida y directa. La interacción chatbox-persona es clara, sólo desglosaría más información básica de cada tema y en todos los casos poner a donde me dirijo para más información (horarios de atención, dirección web, dirección física, teléfono, etc.) EJEMPLO: "talleres" como estudiante avanzada sólo busco saber qué talleres se están dictando, dónde y horarios, y así ahorrarle entrar a las páginas web engorrosas. Las preguntas que suelo hacerme y que podría resolver cómodamente con esta herramienta serían consultar números telefónicos, horarios de atención de algún sector, mail de área a contactar según tipo de consulta; todas referidas a mi facultad.

El sistema es intuitivo, sin saber que era un chatbot pude usarlo con facilidad.

Me parece de utilidad tener una respuesta inmediata a preguntas que los alumnos nos hacemos frecuentemente, sobre todo para los aspirantes e ingresantes a la universidad que tienen muchas dudas. Por eso creo que además de las áreas de interés podrían incluirse algunas preguntas frecuentes o links para este sector.

Puntualmente en relación a las áreas seleccionadas en la encuesta, en la sección talleres me hubiese gustado obtener un poco más de información sobre que talleres se brindan, sus horarios y sobretodo el enfoque, para saber si tomar el taller me capacita para trabajar o suma a mis estudios. Tal vez se deba a la falta de un link relacionado que en las otras respuestas si aparece.

Pienso que sería útil también si esta herramienta además de responder preguntas automáticamente, pudiese enviar mensajes o alertas a los usuarios que ya tuvieron conexión, por ejemplo de campañas de vacunación o cuando se realizan arreglos odontológicos gratuitos. Muchas veces no existe forma de realizar avisos masivamente que lleguen a las distintas facultades y esta sería una buena oportunidad. Este punto no se si es posible llevarlo a cabo, tal vez se desvie del objetivo.

Por otro lado me parece que estaría bueno contemplar que el sistema funcione con audio para los que no puedan leer.

Gracias!

Figura 6.6

De la lectura de los comentarios, obtuvimos los siguientes puntos:

- Información: observamos reiterados comentarios acerca de la información expuesta por el chatbot. Si bien comprendemos los comentarios, la finalidad de la encuesta apunta a la funcionalidad del chatbot. Todos los temas relacionados con la información expuesta por el chatbot se pueden resolver agregando la información correspondiente mediante el Backoffice.
- Respuestas rápidas: agregando al comentario ya mencionado por Tomás, vemos un comentario donde nos indican un cierto grado de ligadura con el flujo del chatbot. Entendemos que puedan sentir esto en las primeras interacciones, pero al ir adquiriendo experiencia con la herramienta, comenzarán a interactuar mediante el input desde teclado, lo que quitará esa sensación de sujeción al flujo.
- Botón "inicio" y botón "atrás": observamos que en varias ocasiones los encuestados solicitan este tipo de atajo. Por tal motivo decidimos agregar la lógica necesaria para que el chatbot comprenda un set de palabras claves y redirija al usuario al inicio. También hemos agregado el botón "**volver**", para darle una referencia visual y permitirle volver al inicio del flujo fácilmente.
- Problema de reconocimiento de entrada de teclado: uno de los encuestados nos comenta que la herramienta no reconoce palabras en singular (en su ejemplo marcó la palabra: beca). Esto se puede solucionar sumando dicha palabra desde el Backoffice al "Tema" => "Becas".
- Publicidad: lógicamente esto no corre por nuestra cuenta y escapa al alcance de nuestra tesis.

- Mensajes o alertas: si bien nos parece muy interesante este accionar, lo hemos analizado y acordamos dejarlo fuera del alcance de la primera etapa que desarrollamos.
- Lectores de pantalla: entendemos que el encuestado nos recomienda que la aplicación lea la información enviada, esta funcionalidad se realiza mediante un lector de pantalla.

Encuesta: Tomás Falco

En la siguiente sección haremos foco en las respuestas de Tomás, para puntualizar los temas de accesibilidad que brinda Facebook Messenger.

Como hemos comentado en secciones anteriores, Tomás es un estudiante de la Facultad de Informática no vidente, quien está cursando los primeros años de la carrera, y tiene mucha experiencia con aplicaciones accesibles.

La sesión se llevó a cabo el día 21/02/2019 en el aula 5 de Posgrado de la Facultad de Informática. El estudiante llevó su propia computadora, para trabajar en un entorno que le sea familiar.

La sesión de prueba se dividió en 3 etapas: presentación del chatbot, introducción a la sesión de prueba y preguntas para conocer su perfil (Fig. 6.8).

Sus respuestas fueron las siguientes:

Perfil del encuestado

Sabe ud qué es un chatbot? *

Sí

No

Ha interactuado en alguna ocasión con un chatbot? *

SI

No

Conoce de alguna tecnología similar? *

Sí

No

Figura 6.8

- Actividad.
Aquí se le brindaron unas actividades prácticas para que realice:

Actividad

Contactarse a la página de Facebook "Tesis Becas 2018" y resolver las siguientes consultas:

- Empezar interacción
- Obtener información de los talleres brindados por la UNLP
- Obtener información de los deportes brindados por la UNLP
- Obtener información de las maratones organizadas por la UNLP
- Obtener información de cuantos km consta una "correcaminata", maratón brindada por la UNLP
- Obtener información de las becas brindadas por la UNLP (ingresar texto "becas")
- Obtener información sobre becas de ayuda económica brindada por la UNLP

Figura 6.9

- Interacción con el chatbot y entrevista informal con el participante. Cabe destacar que el participante estuvo 30 minutos interactuando con el chatbot, eligiendo diferentes opciones que éste le brindaba y pudo lograr comunicarse con el mismo sin que hubiera ningún tipo de problema en el flujo de información (Fig. 6.9).
- Preguntas sobre el uso del chatbot. Luego de finalizar las actividades solicitadas, el participante comenzó a responder las preguntas del formulario.

Las actividades fueron claras? *

Sí

No

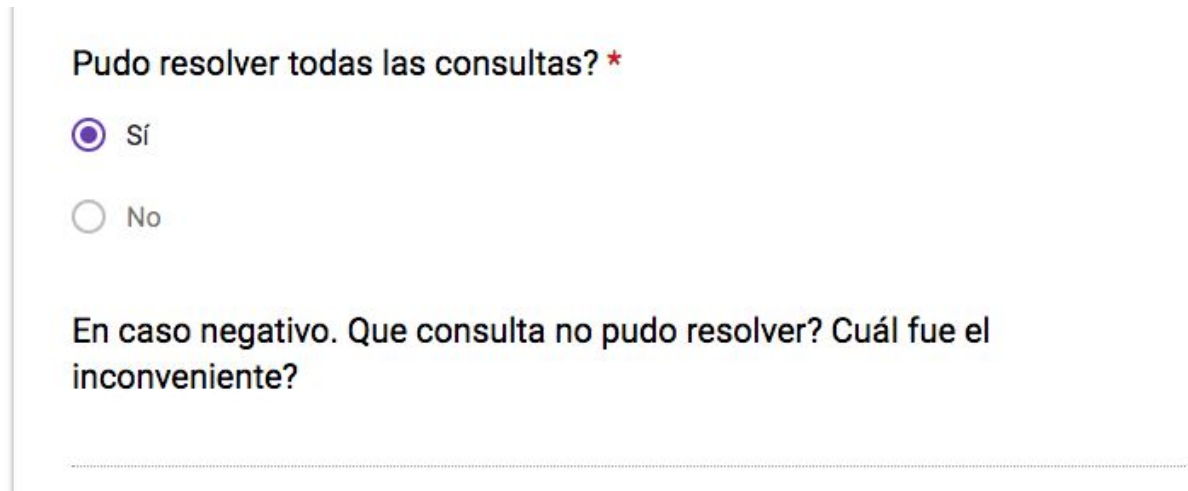
En caso negativo. Que inconvenientes tuvo?

desde la página de facebook no pude visualizar las respuestas rápidas, pero las fui pulsando desde la app móvil

Figura 7.1

Vale aclarar que el encuestado respondió que los enunciados de las actividades que realizó fueron claros, pero al momento de especificar los inconvenientes que tuvo al interactuar con el chatbot hizo mención a la dificultad que tuvo para entender las respuestas rápidas desde la página web de Facebook, y no así desde la aplicación móvil, donde pudo usarlas sin problemas (Fig 7.1).

Destacamos que el participante pudo resolver todas las actividades que realizó.



Pudo resolver todas las consultas? *

Sí

No

En caso negativo. Que consulta no pudo resolver? Cuál fue el inconveniente?

Figura 7.2

Como hemos comentado a lo largo del documento, para interactuar con el chatbot hay dos maneras: las respuestas rápidas y la escritura a mano por parte del usuario. Al encuestado se le consultó cuáles de las dos posibilidades de interacción utilizó y respondió lo siguiente:



En la interacción con el chatbot, respondió utilizando: *

Respuestas automáticas

Escribió sus respuestas

Ambas

Figura 7.3

La respuesta fue que usó ambos métodos de interacción (Fig. 7.3), con lo cual las siguientes preguntas tuvieron que ver con cuán conforme estuvo con la respuestas brindadas por el chatbot y si sentía que las mismas iban a ser de utilidad (Fig. 7.4). El encuestado respondió lo siguiente:

Cuán satisfecho esta con las respuestas brindadas por el chatbot? *

- Muy satisfecho
- Satisfecho
- Conforme
- Defraudado
- Muy defraudado

Cree que será util la implementación del chatbot? *

- Sí
- No

Figura 7.4

Al finalizar la encuesta se le dio la posibilidad al encuestado que pueda dar sugerencias para mejorar la interacción del chatbot.

Recomendaciones

Siéntase libre, de creerlo conveniente, en realizar las sugerencias necesarias para mejorar el uso e interacción del chatbot

se puede agregar la información sobre los diferentes transportes de la unlp como micro, tren y demás, que si bien está en las páginas de bienestar, como toda la información cuesta encontrarla y a veces es confusa.

La interacción con respuestas automáticas se trabaja mejor desde la aplicación móvil, al menos en Android, y me respaldé en la página móvil.

<https://m.facebook.com> para escribir las diferentes consultas ya que permite mayor flexibilidad, pero no lee las respuestas rápidas. Podría también tener algo de info de parciales y finales, o en todo caso algo mas fácil, que redirija a la web de cada cátedra, y ahí el alumno busque la información que esté colgada en la página.

Figura 7.5

Como se muestra en la figura 7.5, Tomás indicó que agregaría información al chatbot. En este punto, comprendemos que si bien el chatbot está en una versión estable, el contenido

es de prueba. Por lo tanto, para satisfacer sus indicaciones bastaría con que un administrador del backoffice cargue la información necesaria y las posibles preguntas que puede llegar a recibir el chatbot.

Finalmente con respecto a la interacción con el chatbot, el encuestado utilizó:

- Lector de pantalla NVDA¹⁰
- Navegador Google Chrome
- Versión móvil de Facebook Messenger (m.facebook.com), la cual al tener menos contenido tarda menos tiempo en cargar y le permite moverse con el lector de pantalla más rápidamente.

Como contra, usando la versión móvil utilizada en desktop, encontró que a veces no actualizaba los mensajes automáticamente por lo que debía hacerlo él manualmente. Además, no leía las respuestas rápidas que enviaba el chatbot a medida que se desarrollaba la charla.

Para solucionar estos problemas combinó la página móvil de Facebook Messenger con la aplicación nativa de Android y el lector Talk Back¹¹, para que le leyera las respuestas rápidas de manera satisfactoria.

La versión de escritorio de Facebook Messenger para el usuario tiene demasiado contenido, lo que lo hace más lento para cargarse y más complejo para seguir el hilo de la charla con el lector de pantalla. A su vez, aclaró que tampoco lee las respuestas rápidas.

Su problema fue resuelto cuando se le sugirió utilizar la aplicación de escritorio y móvil de Messenger (<https://www.messenger.com/t/2017610355128470>) la cual leyó perfectamente las respuestas rápidas con el NVDA y, a su vez, el contenido de la página web era mucho menor que al de la versión de escritorio de Facebook Messenger (<https://www.facebook.com/messages/t/2017610355128470>). Nótese que los número que siguen están en las URL's forman el identificador del chatbot que desarrollamos.

En cuanto a la accesibilidad de los navegadores, explicó que Facebook Messenger la mantiene tanto en Chrome, Firefox como Internet Explorer.

Podemos concluir que las respuestas rápidas en la versión de escritorio de Facebook Messenger no son una buena opción para los usuarios no videntes. Y sí lo son en la aplicación nativa de Android, con un lector de pantalla Talk Back y en la versión de escritorio <https://www.messenger.com> con un lector de pantalla NVDA.

¹⁰ Lector de pantalla para Microsoft Windows gratuito. Es un proyecto de software libre que se puede descargar de su página web <https://www.nvaccess.org/>

¹¹ TalkBack es el lector de pantalla de Google incluido en los dispositivos Android. Esta función emite comentarios por voz para que puedas usar el dispositivo sin mirar la pantalla

Setup

Para poder levantar el backoffice sobre un servidor local se debe hacer lo siguiente:

1. Abrir una consola de comandos
2. Posicionarse sobre la carpeta raíz "backoffice" donde está el archivo package.json el cual tiene el script start: "ng serve"
3. Una vez ahí, escribir el comando "ng serve" ó bien el mismo comando abreviado "ng s" y presionar la tecla Enter.
4. Abrir cualquier navegador, nosotros preferentemente usamos Google Chrome.
5. En la barra de navegación escribir <http://localhost:4200> y presionar la tecla Enter.
6. Finalmente el navegador redireccionará a la pantalla /login

Para poder implementar el chatbot haciendo uso de la aplicación de Facebook Messenger, tuvimos que cumplir con los siguientes requisitos:

1. Crearnos una cuenta de Facebook



The image shows the Facebook registration page titled "Abre una cuenta". It includes a sub-header "Es gratis y lo será siempre." and several input fields: "Nombre" and "Apellido" (two separate boxes), "Número de celular o correo electrónico", and "Contraseña nueva". Below these is a "Fecha de nacimiento" section with a date picker set to "6 jun 1994". There is also a "Sexo" section with radio buttons for "Mujer", "Hombre", and "Personalizado". At the bottom, there is a green "Registrarte" button and a small disclaimer about terms and conditions.

Figura 7.6

2. Fue necesario crear una aplicación dentro de Facebook Developers <https://developers.facebook.com>, la llamamos "Tesis Asuntos Estudiantiles 2019"

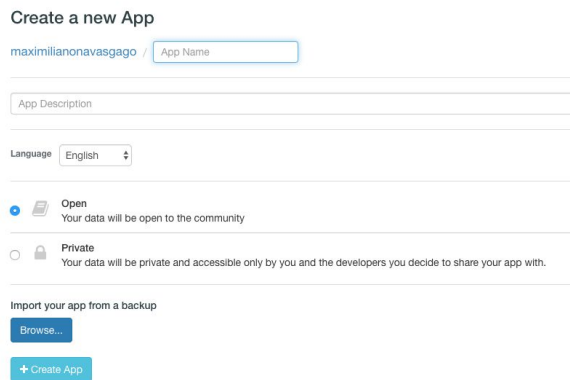


The image shows the Facebook Developers page for creating a new app identifier. The title is "Crear un nuevo identificador de la app". Below the title, it says "Empieza a integrar Facebook en tu app o sitio web". There are two input fields: "Nombre para mostrar" with the placeholder text "El nombre con el que quieres asociar este identificador de la app", and "Correo electrónico de contacto" with the value "maxinavasgago@hotmail.com". At the bottom, there is a "Cancelar" button and a blue "Crear identificador de la app" button. A small disclaimer at the bottom left reads "Al continuar, aceptas las Políticas de la plataforma de Facebook".

Figura 7.7

3. Crear una cuenta en <https://wit.ai/> y crear una aplicación.

wit.ai



Create a new App

maximilianonavasgago / App Name

App Description

Language English

Open
Your data will be open to the community

Private
Your data will be private and accessible only by you and the developers you decide to share your app with.

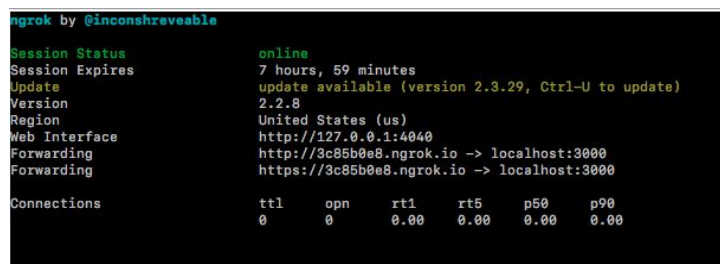
Import your app from a backup

Browse...

+ Create App

Figura 7.8

4. Abrimos una consola de comandos
5. Para poder levantar el chatbot y dejarlo corriendo en el puerto 3000, hay que posicionarse en la carpeta raíz "Tesis-Chatbot" en donde se encuentran los archivos package.json que tiene el script "dev" configurado para correrlo, e index.js el cual tiene el código y las variables de entorno necesarias.
6. Luego, seteando los valores de las variables de entorno y escribiendo el comando "dev" para levantarlo es suficiente: VERIFY_TOKEN="XXXX" PAGE_ACCESS_TOKEN="XXXX" WIT_TOKEN="XXXX" FB_APP_SECRET="XXXX" npm run dev
7. Abrimos otra consola de comandos
8. Con ngrok exponemos el servidor web local a internet con el comando "ngrok http 3000". Es importante aclarar que el puerto debe ser el mismo del que está corriendo el chatbot.
9. Copiamos la URL https que crea ngrok:



```
ngrok by @inconshreveable
Session Status      online
Session Expires    7 hours, 59 minutes
Update              update available (version 2.3.29, Ctrl-U to update)
Version             2.2.8
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://3c85b0e8.ngrok.io -> localhost:3000
                    https://3c85b0e8.ngrok.io -> localhost:3000
Connections
  ttl   opn   rt1   rt5   p50   p90
   0     0    0.00  0.00  0.00  0.00
```

Figura 7.9

10. En la página de Facebook Developers, dentro de la aplicación que creamos "Tesis Asuntos Estudiantiles 2019", en la opción "webhooks", en "Edit Subscription", al abrirse el popup con 2 campos para completar, tenemos que pegar la URL https que copiamos de ngrok en "URL de devolución de llamada" y en "Verificar Token" pegar el mismo valor que pusimos en la variable de entorno VERIFY_TOKEN del punto 6, luego presionar el botón "Verificar y guardar".

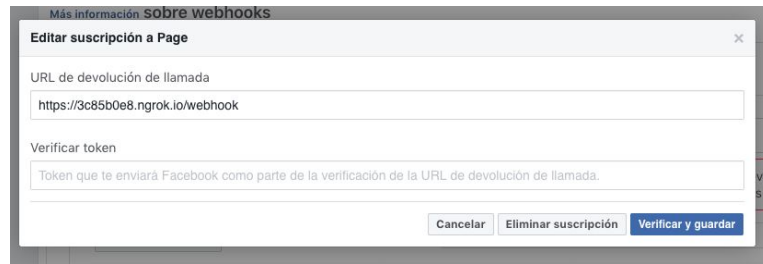


Figura 8.1

11. Para administrar los datos para la inteligencia del chatbot utilizamos una base de datos Apache Cassandra y para correr la API, desarrollada en Java, el IDE llamado IntelliJ IDEA.
12. Abrir otra consola de comandos
13. Para correr Cassandra es necesario escribir el comando "cassandra" y presionar la tecla Enter. Esto levantará una instancia de Cassandra en nuestro ambiente local.
14. Abrir la aplicación del IDE IntelliJ IDEA, elegir el proyecto con el código de la API y luego ejecutar el run del metodo main de la clase App.java (línea 27):

```
App.java x
1 package unlp.info.chatbot;
2
3 import ...
20
21 public class App {
22     private static final Logger LOGGER = LoggerFactory.getLogger(App.class);
23
24     /** Context path of the application. */
25     private static final String APP_PATH = "/*";
26
27     public static void main(String[] args) throws Exception {
28         LOGGER.info("[APP] Starting Chatbot - Environment Configuration: {}", "LOCAL");
29
30         Server server = new Server( port: 9290);
31
32         AnnotationConfigWebApplicationContext spring = new AnnotationConfigWebApplicationContext();
33         spring.register(ComponentConfiguration.class);
34
35         ServletContextHandler servletHandler =
36             new ServletContextHandler(ServletContextHandler.NO_SESSIONS);
37         servletHandler.setContextPath("/chatbot");
38
39         DispatcherServlet servlet = new DispatcherServlet(spring);
40         servlet.setDispatchOptionsRequest(true);
41         servletHandler.addServlet(new ServletHolder(servlet), APP_PATH);
42
43         addGZipHandler(servletHandler);
44
45         FilterHolder encodingFilter =
46             servletHandler.addFilter(
47                 CharacterEncodingFilter.class, APP_PATH, EnumSet.of(DispatcherType.REQUEST));
48         encodingFilter.setInitParameter( param: "encoding", value: "UTF-8");
49         encodingFilter.setInitParameter( param: "forceEncoding", value: "true");
50
51         server.setHandler(servletHandler);
52         server.setStopAtShutdown(true);
53
54         try {
55             server.start();
56             server.join();
57         } catch (Exception exception) {
58             LOGGER.error("[APP] Error starting the application " + exception.getMessage());
59             exit( status: 1);
60         }
61     }
62 }
```

Figura 8.2

Al levantar la aplicación verán el siguiente mensaje:

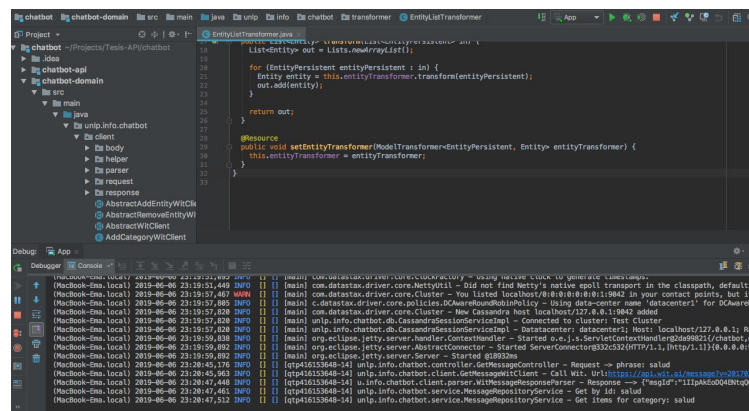


Figura 8.3

15. Finalmente dentro de Facebook Messenger hay que buscar "Tesis Asuntos Estudiantiles 2019" y empezar a interactuar con el chatbot.

Es muy importante destacar que todo el setup anteriormente especificado sólo se debe hacer en caso de querer correr el backoffice y el chatbot sobre un servidor local. Si se pudieran hostear ambas aplicaciones, el backoffice podría ser accedido desde cualquier navegador web y el chatbot estaría corriendo todo el tiempo, con lo cual podría ser accedido sin problemas por cualquier usuario, en cualquier momento, a través de sus versiones de escritorio o móvil.

El chatbot se puede configurar para cualquier página de Facebook. Para nuestra tesis cuyo objetivo es brindar información del área de Bienestar Estudiantil de la UNLP recomendamos que se coloque en la página oficial de Facebook de Bienestar de la UNLP: <https://www.facebook.com/BienestarUNLP/> Además la API de Facebook Messenger nos permite instalar un plugin de acceso directo al chatbot en cualquier sitio web. Cabe destacar que todas las herramientas de interfaz de usuario que usamos para el desarrollo de nuestro chatbot (respuestas rápidas, botones para empezar, etc) en la documentación de la API de Facebook Messenger aparecen como compatibles con cualquier navegador, tanto en versión de escritorio como móvil.

Conclusiones y Trabajo Futuro

Creemos que este trabajo de tesis puede ser el punto de partida para que el departamento de Asuntos Estudiantes de la UNLP pueda brindar información a través de una herramienta eficiente, accesible y conocida por muchísimas personas.

Con la implementación de este chat se ha buscado resolver la comunicación de muchos temas importantes, que actualmente son de difícil acceso, ya que las vías de comunicación son desconocidas y dispersas por diferentes canales.

La decisión de usar Facebook Messenger para alojar nuestro chatbot tuvo que ver con la importancia que le da Facebook a la accesibilidad en sus herramientas, con el conocimiento y alcance masivo que tiene entre las personas a nivel mundial. Desde el lado técnico, con la completa documentación que brinda para poder desarrollar nuestra solución.

Respecto del backoffice, definimos la realización de dicho entorno para darle flexibilidad al chatbot, y que este pueda dar información sobre diferentes temas a los usuarios con los que interactúa. Para eso desarrollamos una API muy completa, que le permite a un usuario (data entry) poder crear y eliminar temas de conversación de acuerdo a las necesidades que tenga en un momento determinado.

A pesar de que esta solución es viable, estamos convencidos que hay algunos puntos que se pueden investigar y mejorar para llevar a cabo una herramienta aún más completa.

Entre ellos están:

- El desarrollo de manejo de usuarios y permisos dentro del BackOffice: Actualmente sólo hay un usuario, el cual se creó a modo de prueba, y los permisos no fueron implementados. Si se quisiera tener varios usuarios con diferentes accesos y permisos, hoy no sería posible.
- Una investigación acerca de cómo mejorar la automatización de la inteligencia del chatbot. Hay muchas tecnologías de Machine Learning que se puedan utilizar para cumplir con este requerimiento, como por ejemplo TensorFlow, Keras, entre otras.

Referencias

APELLIDOS EN VERSALITA, Inicial del nombre (año). Título en cursiva. Lugar de publicación: editorial.

- Facebook. *Accesibilidad*: <<https://www.facebook.com/help/accessibility>> [Consulta: 28 de Agosto de 2019]
- Weizenbaum, J. (1966). *ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine*. Cambridge, Mass: Communications of the ACM
- Poole, D., Mackworth, A. y Goebel, R. (2018). *Computational Intelligence: A Logical Approach*. Nueva York: Oxford University Press.
- Norvig, P. y Russell, S.J. (2009). *Artificial intelligence: a modern approach (tercera edición)*. Upper Saddle River, N.J.: Prentice Hall.
- Schank, R.C. (1991). *Where's the AI*. AI Magazine. Volumen 12, número 4.
- López Takeyas, B, (2007). *Introducción a la Inteligencia Artificial*. Nuevo Laredo, Tamps. México:
- McCarthy, J. (2007). "Basic Questions" en *What Is Artificial Intelligence?*. Computer Science Department.
- Comes, Y., Solitario, R., Garbus, P., Mauro, M., Czerniecki, S., Vázquez, A., Sotelo, R. y Stolkiner, A. (2006). *El concepto de accesibilidad: La perspectiva relacional entre población y servicios*.
- Almeida C. (2000). *Investigación en Sistemas y Servicios de Salud. Resultados, evaluación de actividades y desarrollo futuro de la Red. Cuadernos para discusión. Número 1*. Rio de Janeiro, Brasil: Unipe.
- UIAccess. *Another -ability: Accessibility Primer for Usability Specialists*. UPA (Usability Professionals' Association) 2003 Conference. <<http://www.upassoc.org/conf2003/call/downloads/01-Another-Ability.pdf>> [Consulta: 28 de Agosto de 2019]
- Henry, S.L (2006). "Understanding Web Accessibility" en *Web Accessibility: Web Standards and Regulatory Compliance*. Glasshaus
- W3C. *Web Content Accessibility Guidelines (WCAG) 2.1*. <<https://www.w3.org/TR/WCAG21/>> [Consultado: 28 de Agosto de 2019]
- Ayuda de Google. *Comienza a usar Android con TalkBack*. <https://support.google.com/accessibility/android/answer/6283677?hl=es-419&ref_to_pic=3529932> [Consultado: 28 de Agosto de 2019]
- Apple. *VoiceOver*. <<https://www.apple.com/la/accessibility/iphone/vision/>> [Consultado: 28 de Agosto de 2019]
- NVDA. *¿Qué es NVDA?*. <<https://nvda.es/>> [Consultado: 28 de Agosto de 2019]
- Freedomscientific. *JAWS*. <<https://www.freedomscientific.com/products/software/jaws/>> [Consultado: 28 de Agosto de 2019]
- WAVE. *WAVE Help*. <<https://wave.webaim.org/help>> [Conusltado: 28 de Agosto de 2019]
- Universidad de Murcia. *La Universidad de Murcia presenta a LOLA, un asistente de inteligencia artificial para ayudar a los nuevos alumnos*. <<https://www.um.es/web/sala-prensa/-/la-universidad-de-murcia-presenta-a-lola-un-a>>

- [sistente-de-inteligencia-artificial-para-ayudar-a-los-nuevos-alumnos](#)> [Consultado: 28 de Agosto de 2019]
- 1millionbot. *Chatbots, dos casos de éxito: Lola (Universidad de Murcia) y Ana (prensa, Diario Información)*.
<<https://1millionbot.com/chatbots-dos-casos-de-exito-lola-universidad-y-ana-prensa/>> [Consultado: 28 de Agosto de 2019]
 - Staffordshire University. *Introducing Beacon – a digital friend to Staffordshire University students*.
<<http://www.staffs.ac.uk/news/2019/01/introducing-beacon-a-digital-friend-to-staffordshire-university-students>> [Consultado: 28 de Agosto de 2019]
 - University of Canberra. *Students make new friend in Lucy the chatbot*.
<<https://www.canberra.edu.au/about-uc/media/newsroom/2018/february/students-make-new-friend-in-lucy-the-chatbot>> [Consultado: 28 de Agosto de 2019]
 - Diario Clarín. *Periodismo por mensajería. Qué es y cómo funciona el robot de Clarín en Facebook Messenger*.
<https://www.clarin.com/tecnologia/funciona-robot-clarin-facebook-messenger_0_Bkjbv_ph.html> [Consultado: 28 de Agosto de 2019]
 - Prosecretaría de Asuntos Estudiantiles.
<http://www2.ing.unlp.edu.ar/bienestar/index_new.php> [Consultado: 28 de Agosto de 2019]
 - Wit.ai. *Recipes for apps you can talk to*. <<https://wit.ai/docs/recipes>> [Consultado: 28 de Agosto de 2019]
 - Facebook for developers. *Respuestas rápidas*.
<<https://developers.facebook.com/docs/messenger-platform/send-messages/quick-replies/>> [Consultado: 28 de Agosto de 2019]
 - Datastax. *Cassandra vs. MongoDB vs. Couchbase vs. HBase*.
<<https://www.datastax.com/nosql-databases/benchmarks-cassandra-vs-mongodb-vs-hbase>> [Consultado: 28 de Agosto de 2019]
 - Abdul-Kader, S.A. y Woods, J. “Survey on Chatbot Design Techniques in Speech Conversation Systems” en *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 7, 2015
 - Stoner, D. J., Ford, L., y Ricci, M. (2003). *Simulating Military Radio Communications Using Speech Recognition and Chat-Bot Technology*. [Figura]
<https://thesai.org/Downloads/Volume6No7/Paper_12-Survey_on_Chatbot_Design_Techniques_in_Speech_Conversation_Systems.pdf> [Consultado: 28 de Agosto de 2019]
 - Angular. *Introduction to the Angular Docs*. <<https://angular.io/docs>> [Consultado: 28 de Agosto de 2019]
 - NgRx. *What is NgRx?* <<https://ngrx.io/docs>> [Consultado: 28 de Agosto de 2019]
 - Ngx-bootstrap. *Angular Bootstrap*. <<https://valor-software.com/ngx-bootstrap/#/>> [Consultado: 28 de Agosto de 2019]
 - Oracle. *Java*. <<https://docs.oracle.com/javase/specs/jls/se8/html/jls-1.html>> [Consultado: 28 de Agosto de 2019]
 - Apache. *Apache Maven Project*. <<http://maven.apache.org/what-is-maven.html>> [Consultado: 28 de Agosto de 2019]
 - Spring. *Spring Framework*. <<https://spring.io/projects/spring-framework>> [Consultado: 28 de Agosto de 2019]
 - Eclipse. *Eclipse Jetty*. <<https://www.eclipse.org/jetty/about.html>> [Consultado: 28 de Agosto de 2019]
 - Cortez Vásquez, A., Vega Huerta, H. y Pariona Quispe, J. (2009). “Procesamiento de lenguaje natural (NLP)”.

<http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/risi/2009_n2/v6n2/a06v6n2.pdf
> [Consultado: 28 de Agosto de 2019]

- Apache. *Apache Cassandra*. <<http://cassandra.apache.org/>> [Consultado: 28 de Agosto de 2019]
- Nielsen, J. y Landauer, T.K. (1993). "A mathematical model of the finding of usability problems" en *Proceedings of ACM INTERCHI'93 Conference*. Amsterdam: Bellcore.
- Nielsen, J. (2000). "Why You Only Need to Test with 5 Users". [Figura]
<<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>>
[Consultado: 28 de Agosto de 2019]
- Facebook developers. *Plugin de chat con clientes*.
<<https://developers.facebook.com/docs/messenger-platform/discovery/customer-chat-plugin/>> [Consultado: 28 de Agosto de 2019]