

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**DESARROLLO DE UN CHATBOT PARA MODELADO
COLABORATIVO EN SKYPE**

**Ismael Fernández Arroyo
Tutora: Sara Pérez Soler
Ponente: Esther Guerra Sánchez**

JUNIO 2019

Desarrollo de un chatbot para modelado colaborativo en Skype

AUTOR: Ismael Fernández Arroyo

TUTORA: Sara Pérez Soler

PONENTE: Esther Guerra Sánchez

**Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2019**

Resumen

El desarrollo de software es una actividad colaborativa. A su vez, en el desarrollo dirigido por modelos, dichos modelos no son una excepción a esta regla, por lo que se crean mayoritariamente de forma colaborativa. Para facilitar la tarea, las herramientas de modelado deben proporcionar, además de entornos para la creación de modelos, mecanismos de comunicación y discusión que sean sencillos de utilizar y de fácil acceso para las personas que utilizan estas herramientas.

Para ello, las redes sociales han demostrado su gran eficacia para coordinar y comunicar a grandes grupos de personas, incluso en la distancia. Junto a ellas, el uso de los chatbots ha dado un giro a la manera en que interactuamos con los sistemas informáticos, facilitando así la interacción con el usuario y haciendo, de esta manera, más sencilla e intuitiva la funcionalidad que se ofrece. Redes sociales como Skype proporcionan mecanismos de comunicación más directos, como llamadas, videollamadas o videoconferencias. Además, integran el uso de los chatbots con estos mecanismos de comunicación favoreciendo la colaboración entre usuarios.

Usando los mecanismos de comunicación de las redes sociales, concretamente Skype, en este trabajo proponemos crear un chatbot que realice tareas de modelado. Para ello, se implementará un bot utilizando las tecnologías de Microsoft, concretamente Bot Framework para el desarrollo, el cual se realizará en lenguaje C# y .NET, y Azure Bot Service para desplegar dicho bot en Skype. Este chatbot utilizará una API REST que proporciona la funcionalidad para la creación y gestión de modelos de forma colaborativa.

Palabras clave

Chatbots, bots, modelado colaborativo, redes sociales, Skype.

Abstract

Software development is a collaborative activity. At the same time, in model-driven development such models are not an exception to this rule, for which they are mostly created collaboratively. In order to make the task easier, the modeling tools must provide in addition to environments for the creation of models, communication and discussion mechanisms that are simple to use and easily accessible for people who use these tools.

To do this, social networks have proved great effectiveness in coordinating and communicating large groups of people, even in the distance. Along with them, the use of chatbots has changed the way we interact with computer systems, thus facilitating interaction with the user and making the functionality offered easier and more intuitive. Social networks just as Skype provide more direct communication mechanisms such as calls, video calls or videoconferences. In addition, they integrate the use of chatbots with these communication mechanisms favoring collaboration among users.

In this project we propose to create a chatbot that performs modeling tasks by using the communication mechanisms of social networks, specifically Skype. For this, a bot will be implemented using Microsoft technologies, concretely Bot Framework for development which will be done in C# language and .NET, and Azure Bot Service to deploy the bot in Skype. This chatbot will use a REST API that provides the functionality for the creation and management of models in a collaborative way.

Keywords

Chatbots, bots, collaborative modeling, social networks, Skype.

Agradecimientos

En primer lugar me gustaría agradecer a mi familia, especialmente a mis padres, por haberme apoyado y dado ánimos en todo momento.

A Isa por haber estado ahí siempre que lo necesitaba desde que comencé esta carrera.

También agradecer a mis amigos, tanto de dentro como de fuera de la universidad, que me han ofrecido siempre su ayuda de una forma u otra.

Por último, quiero agradecer a mi tutora Sara por darme la oportunidad de realizar este trabajo y participar en esta gran idea.

ÍNDICE DE CONTENIDOS

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	1
1.3	Organización de la memoria	2
2	Estado del arte	3
2.1	Modelado colaborativo	3
2.1.1	SOCIO	4
2.2	Redes sociales de mensajería	4
2.2.1	Skype	5
2.3	Chatbots	6
2.3.1	Bots en el desarrollo de software	7
2.3.2	Bots en Skype	7
3	Diseño	9
3.1	Arquitectura	9
3.2	Azure Bot service	10
3.2.1	Bot Framework	10
3.2.2	Azure	10
3.3	SOCIO API REST	11
3.3.1	Funcionalidades	11
3.4	Lógica del bot	12
3.4.1	Principios de utilidad	12
3.4.2	Interfaz e interacción	13
4	Desarrollo	15
4.1	Comunicación	15
4.1.1	Contexto y actividad	15
4.1.2	Recepción y envío	15
4.2	Sugerencia de acciones	16
4.3	Elementos multimedia	16
4.3.1	Tarjetas	16
4.4	Almacenamiento de datos	17
4.4.1	Modelado	17
4.4.2	Conversación	17
4.5	Llamadas a SOCIO API REST	18
5	Herramienta	19
6	Conclusiones y trabajo futuro	25
6.1	Conclusiones	25
6.2	Trabajo futuro	26
	Referencias	27
	Glosario	29

ÍNDICE DE FIGURAS

FIGURA 2-1: GRÁFICA DE ACTIVIDAD DE LAS DISCUSIONES SOBRE CHATBOTS EN INTERNET	6
FIGURA 3-1: ARQUITECTURA DEL CHATBOT	9
FIGURA 3-2: CONEXIÓN A CANALES EN AZURE	11
FIGURA 4-1: SUGERENCIA DE ACCIONES	16
FIGURA 4-2: EJEMPLO DE TARJETA ENRIQUECIDA	17
FIGURA 5-1: COMANDO <i>START</i> EN CONVERSACIÓN INDIVIDUAL Y GRUPAL	19
FIGURA 5-2: COMANDOS <i>NEWPROJECT</i> Y <i>SETPROJECT</i>	20
FIGURA 5-3: EDICIÓN DE MODELOS MEDIANTE EL COMANDO <i>TALK</i>	21
FIGURA 5-4: COMANDOS <i>UNDO</i> Y <i>REDO</i>	21
FIGURA 5-5: COMANDO <i>UPDATES</i> Y EJEMPLO DE RAMA	22
FIGURA 5-6: COMANDO <i>STATISTICS</i> PARA ACCIONES Y PORCENTAJE DE AUTORÍA.....	23
FIGURA 5-7: AMPLIACIÓN DE LA FIGURA 5-6.....	24

ÍNDICE DE TABLAS

TABLA 2-1: COMPARATIVA DE APLICACIONES DE MENSAJERÍA CON SKYPE.....	5
---	---

1 Introducción

En este apartado se describen las ideas que han motivado a la realización de este trabajo, así como los objetivos a conseguir en el proyecto y la estructura de la que consta el documento.

1.1 Motivación

En la actualidad, el interés por el uso de modelos para el desarrollo de software está en auge. Esto es debido a que cada vez se busca más agilizar el proceso de creación de software y, para ello, el uso de modelos proporciona un mayor nivel de abstracción, obteniendo así un resultado de mayor calidad a un bajo coste [1].

Uno de los aspectos fundamentales del desarrollo de software dirigido por modelos es la colaboración. A la hora de realizar estas tareas es importante que cada miembro que forme parte de este desarrollo muestre su opinión y pueda aportar su parte al proyecto. Además estas aportaciones deben ser visibles a los demás y tienen que poder gestionarse de alguna forma para poder llegar a un acuerdo. Por otro lado, es importante destacar que se requiere la contribución de expertos en temas de modelado para conseguir un buen resultado en el proyecto en el que se está trabajando. También es necesario el uso de aplicaciones que simplifiquen estas tareas y, a pesar de ello, existe un escaso número de herramientas dedicadas al modelado colaborativo y que además se caractericen por su facilidad de uso.

A día de hoy, las redes sociales nos ayudan a comunicarnos de múltiples maneras e independientemente del lugar donde nos encontremos, por lo que es un avance de gran utilidad a la hora de hablar de colaboración. Muchas redes sociales, como Skype, ofrecen además distintas herramientas en este tipo de comunicación, y una de ellas son los bots. Los bots proporcionan una interacción sencilla con el usuario para lograr distintas funcionalidades, que siendo realizadas de otra forma serían mucho más complejas.

Debido a todas estas ideas, en este TFG se propone el desarrollo de un chatbot que facilite al usuario las tareas de modelado colaborativo, realizando y gestionando las mismas en la red social Skype.

1.2 Objetivos

El objetivo principal de este TFG es asistir a los usuarios a la hora de realizar tareas de modelado colaborativo a través de un chatbot de Skype. Para ello, dentro de este objetivo principal se dividen los siguientes objetivos más específicos:

- Creación de un bot en Skype haciendo uso de la herramienta Azure Bot Service que permita desarrollarlo, probarlo y administrarlo.

- Proporcionar facilidad de uso a los usuarios, tanto a la hora de gestionar el bot en Skype como en el momento de interacción con el mismo.
- Implementar funcionalidades para que los usuarios puedan realizar proyectos de modelado de forma sencilla e intuitiva. Esto incluye poder gestionar las tareas y proporcionar medios de discusión. Para dicha funcionalidad se utilizará la API REST de SOCIO [3].
- Permitir el desarrollo de modelos en Skype tanto de forma individual como en grupos.
- Dotar al bot de la capacidad para poder interactuar con otras redes sociales que también usen la API de SOCIO.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- En el **estado del arte** (Sección 2) se estudiará el estado actual de los temas fundamentales de los que consta este trabajo. Este apartado se encuentra dividido en tres secciones: el modelado colaborativo, las redes sociales de mensajería y los chatbots.
- En cuanto al apartado de **diseño** (Sección 3) se expondrá el planteamiento, estructura y herramientas que se han decidido utilizar para llevar a cabo el proyecto.
- La parte de **desarrollo** (Sección 4) dará a conocer más en detalle cómo se va a realizar la implementación de las distintas características planteadas para conseguir los requisitos deseados para el trabajo.
- En el apartado de **herramienta** (Sección 5) se expondrá el uso de la propia aplicación en Skype mostrando sus distintas funcionalidades.
- Por último, en las **conclusiones y trabajo futuro** (Sección 6) se llevará a cabo una valoración y se plantearán posibles formas de mejora del proyecto de cara al futuro.

2 Estado del arte

En este capítulo se realizará un estudio sobre las herramientas existentes en la actualidad sobre modelado colaborativo, redes sociales y chatbots, que supondrá la toma de decisiones sobre la que se basa el diseño y la implementación de este trabajo.

2.1 Modelado colaborativo

En el ámbito del desarrollo de software, el modelado colaborativo tiene un objetivo principal: el aumento de la productividad. Para ello se crean representaciones abstractas de los conocimientos y actividades que conforman un dominio [2], además se requiere de la comunicación y la colaboración de los integrantes del proyecto, incluyendo expertos en modelado. El uso de este tipo de planteamiento de desarrollo está aumentando, sobretodo en el sector empresarial, y es por ello por lo que ha ido surgiendo la necesidad de herramientas sencillas que permitan y faciliten el modelado colaborativo.

Este trabajo se centra en el tipo de modelado colaborativo online, ya que este permite la comunicación y actuación en tiempo real de los usuarios remotamente. Este tipo de modelado requiere de funcionalidades concretas en cuanto a conciencia del contexto, discusión y coordinación [3].

En el panorama actual de este tipo de herramientas de modelado colaborativo podemos encontrar algunos ejemplos como los siguientes:

- **Creately** [4]: se trata de una aplicación web que cuenta con una interfaz gráfica para crear y editar diagramas. Ofrece ciertas funcionalidades como colaboración en tiempo real con otros usuarios, historial de versiones o comentarios en línea.
- **Cacoo** [5]: similar a la herramienta anterior que cuenta con otras características como notificaciones instantáneas de actividad o realización de presentaciones de diagramas en vivo a audiencias, las cuales pueden presentar sus opiniones.
- **Gliffy** [6]: es un software que permite el modelado colaborativo de diagramas tanto offline como online mediante un editor gráfico y con integración en multitud de plataformas.
- **AToMPM** [7]: la aplicación cuenta con una página web y un entorno de modelado gráfico, con la peculiaridad de que ofrece la posibilidad de modelado a través de comandos.

Hay que destacar que casi todas las herramientas existentes se basan en editar un modelo gráficamente, por lo que se debe emplear más tiempo en la creación y podría resultar dificultoso en cuanto al modelado. Por otro lado, estas aplicaciones proporcionan medios de discusión que en ciertas ocasiones pueden resultar bastante limitados, ya que no cuentan con servicios como, por ejemplo, chats dedicados al desarrollo del modelo y su discusión, llamadas, videollamadas o conferencias en tiempo real.

2.1.1 SOCIO

En este proyecto se ha decidido optar por la utilización de SOCIO [3] como herramienta de modelado colaborativo. SOCIO consta de una API REST que facilita la realización de tareas de modelado, para ello interpreta las sentencias en lenguaje natural de los usuarios para la creación y edición de modelos. Además, ofrece distintas funcionalidades como mecanismos para la gestión de los modelos, visualización de estadísticas o realización de encuestas. Al ser una arquitectura extensible basada en servicios web, es fácilmente adaptable a distintas plataformas, como chatbots de Skype en este caso, simplificando su utilización y aumentando así los medios de discusión.

Debido a todas las características comentadas, en cuanto al modelado colaborativo ofrece grandes facilidades, tanto en uso como en accesibilidad, a la hora de realizar este tipo de tareas a usuarios no experimentados en modelado y simplifica tanto la creación como la manipulación de modelos que un editor gráfico de diagramas no llega a ofrecer.

2.2 Redes sociales de mensajería

Las redes sociales se han convertido en los últimos años en uno de los elementos más utilizados e imprescindibles en nuestro día a día. Concretamente, las redes sociales de mensajería instantánea han supuesto un gran avance en la comunicación debido a su rapidez de respuesta y a los diversos servicios que ofrece. Entre estos servicios encontramos, ya no únicamente el intercambio de mensajes, sino también de imágenes, videos, audio, archivos, realización de llamadas y videollamadas e incluso la incorporación de chatbots.

En este apartado se van a estudiar varias aplicaciones de mensajería instantánea que permiten la comunicación y colaboración entre personas y que ofrecen diversas características y servicios:

- **WhatsApp** [8]: a día de hoy es una de las aplicaciones de mensajería más utilizadas, por lo que una de sus principales características es la disponibilidad de usuarios. WhatsApp está diseñado para funcionar principalmente sobre móviles asociándose al número de teléfono. Por otro lado, no se cuenta con servicios como llamadas grupales, videoconferencias grupales ni con bots.
- **Facebook Messenger** [9]: es otra de las redes sociales de mensajería más demandadas, por lo que dispone de un gran número de usuarios activos. Cuenta con la mayoría de características fundamentales para la comunicación actual, ya sean llamadas, videoconferencias o bots, siendo también multiplataforma.
- **Telegram** [10]: la principal peculiaridad es que al ser un servicio en la nube cuenta con un gran número de plataformas sobre las que funciona, a diferencia de otras aplicaciones. También se caracteriza por el uso de canales para difundir mensajes públicos a grandes audiencias. No cuenta con servicio de videollamadas.

- **Hangouts** [11]: es un servicio de mensajería proporcionada por Google. A diferencia de otras aplicaciones no dispone de un gran número de usuarios y el número de plataformas es limitado ya que no funciona sobre Windows o Linux, aunque se han ido actualizando distintos servicios como llamadas, videollamadas y bots.

A modo de resumen, en la tabla 2-1 se muestra una comparativa entre las redes sociales de mensajería anteriormente estudiadas y Skype, la cual se utilizará en este trabajo y se expondrá a continuación:

	Whatsapp	Facebook Messenger	Telegram	Hangouts	Skype
Plataformas	Android, iOS, Web	Android, iOS, Web, Windows, Linux	Android, iOS, Web, Windows, Linux, extensión navegador	Android, iOS, Web	Android, iOS, Web, Windows, Linux
Envío de ficheros	Sí	Sí	Sí	Sí	Sí
Llamadas	Sí	Sí	Sí	Sí	Sí
Videoconferencia	No	Sí	No	Sí	Sí
Bots	No	Sí	Sí	Sí	Sí
Canales	No	No	Sí	No	No

Tabla 2-1: Comparativa de aplicaciones de mensajería con Skype

2.2.1 Skype

Una vez estudiadas las distintas tecnologías en cuanto a mensajería instantánea, explicaremos la aplicación Skype [12], que es la que se ha decidido utilizar en este TFG.

Skype es una de las redes sociales dedicadas a la mensajería y comunicaciones más utilizadas a día de hoy y cuenta con un gran número de usuarios. Pertenece a Microsoft y permite el intercambio de texto, voz y video sobre VoIP [13].

Dentro del mercado, esta aplicación es una de las que implementa, en mayor parte, los servicios más demandados en la actualidad en cuanto a comunicación: intercambio de mensajes, imágenes, audios, videos, ficheros, realización de llamadas, videollamadas, videoconferencias y permiten también el uso de bots tanto en conversaciones individuales como en grupos. A estas características también hay que añadir llamadas especiales entre ordenadores y red de telefonía fija o móvil. Además es multiplataforma y al estar formado

por un gran número de usuarios activos tiene una alta disponibilidad a la hora de contactar con las personas.

Debido a las funcionalidades ofertadas por Skype, nombradas anteriormente, se ha considerado que es una de las mejores herramientas para la realización de este trabajo, debido principalmente al amplio abanico de características que ofrece a nivel de comunicación que permiten la colaboración y discusión en la creación de modelos y, por otro lado, el servicio de bots sobre el que se basa la realización de este TFG.

2.3 Chatbots

El surgimiento y el avance de la inteligencia artificial en la informática ha supuesto la aparición de distintas herramientas en los últimos años, una de ellas son los chatbots. Los chatbots son programas que se encargan de mantener una conversación con una persona mediante el uso y procesamiento del lenguaje natural.

Las funcionalidades de este tipo de herramienta son diversas, pudiendo facilitar al usuario información o realización de ciertas tareas que resulten tediosas. Por otra parte, una de las ventajas de los chatbots es la rápida capacidad de respuesta y la sencillez que aporta a la interacción con el usuario. También hay que destacar el servicio que pueden dar este tipo de programas, ya que el desarrollo de un bot supone dar asistencia simultánea a un gran número de personas, lo que es muy provechoso para el creador de este tipo de software.

En la gráfica de la figura 2-1 se puede observar la repercusión actual de este tipo de herramientas, en ella se representa un estudio del crecimiento de la actividad de las discusiones sobre chatbots en Internet en los últimos años [14]. Se puede apreciar que el crecimiento por año se ha multiplicado por más de 5 veces su valor.

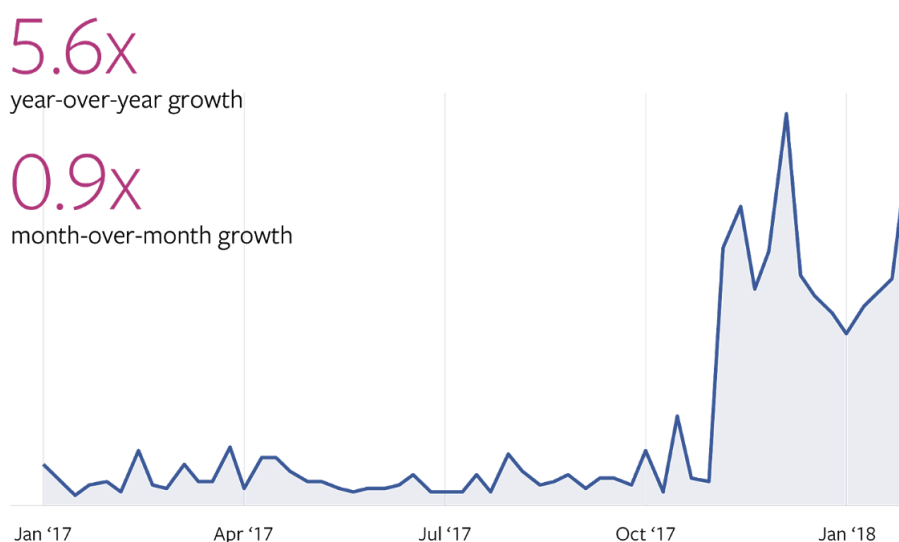


Figura 2-1: Gráfica de actividad de las discusiones sobre chatbots en Internet

2.3.1 Bots en el desarrollo de software

En cuanto a los chatbots relacionados con el desarrollo de software es un ámbito en el que cada vez se está profundizando más. El objetivo final sería conseguir precisamente esa programación automática, pero actualmente el campo en el que más se ha llegado a adaptar los chatbots es en la asistencia a programadores en las diferentes etapas del ciclo de vida del software [15].

Algunos ejemplos de estos bots son:

- **Prophet** [16]: se caracteriza por generar automáticamente sugerencias de correcciones de código sobre programas en C aprendiendo de los propios usuarios.
- **CodeBot** [17]: pertenece a la herramienta Slack y tiene la capacidad de ejecutar código JavaScript.
- **CodeClimate** [18]: este bot se encarga de ofrecer cobertura de tests en distintas herramientas de control de versiones, como GitHub, y también en otras plataformas como Slack.
- **Spotbot** [19]: su funcionalidad es analizar interfaces mediante capturas, ya sean de iOS, Android o un navegador, con el objetivo de encontrar errores de visualización y que sean notificados mediante el bot en Slack.

En cuanto a modelado colaborativo, actualmente no se han encontrado chatbots que cumplan con dicha finalidad.

2.3.2 Bots en Skype

Los bots de Skype comenzaron a funcionar en el año 2016 y en los años siguientes, poco a poco se fue permitiendo el acceso para que los desarrolladores trabajen en estas herramientas, por lo que a día de hoy se trata de una funcionalidad bastante nueva en esta aplicación y existe un escaso número de bots relacionados concretamente con el desarrollo de software. Algún ejemplo existente es el chatbot NLSQL [20] que permite transformar el lenguaje natural a SQL.

La necesidad y surgimiento de este tipo de bots en plataformas como Skype es una de las razones por la cual se ha decidido desarrollar este trabajo en el servicio de chatbots de esta red social. Además de esto, no existen bots relacionados con el modelado colaborativo, lo cual es algo de gran utilidad, ya no solo para esta aplicación sino para el desarrollo de software. También hay que resaltar que Skype, y concretamente Microsoft, han implementado los entornos Azure Bot Service y Bot Framework, que son unas novedosas herramientas que ofrecen diversas funcionalidades de gran potencial a la hora de desarrollar chatbots.

3 Diseño

En esta sección se describe el planteamiento sobre el que se basa el chatbot. Para ello, se dará a conocer la arquitectura del proyecto, las herramientas utilizadas en el diseño, explicando sus funcionalidades para este trabajo, y los conceptos e ideas planteadas que dan lugar al desarrollo del bot.

3.1 Arquitectura

El diseño de la arquitectura de este proyecto se basa principalmente en tres elementos diferenciados: Skype, Azure Bot Service y la API REST de SOCIO. Skype será la aplicación de mensajería instantánea a través de la cual los usuarios interactuarán con el bot, Azure Bot Service proporciona el soporte necesario para la creación, desarrollo y puesta en funcionamiento del mismo y la API de SOCIO facilitará la funcionalidad necesaria para el modelado colaborativo a dicho bot.

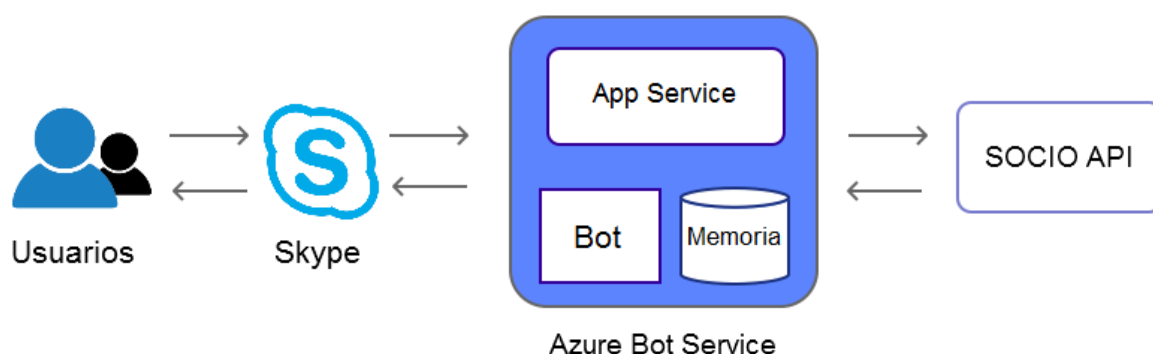


Figura 3-1: Arquitectura del chatbot

La figura 3-1 muestra tanto los componentes como la interacción entre los elementos de la que consta la arquitectura del chatbot diseñada. Los usuarios, a través de distintos dispositivos, acceden a la red social de mensajería Skype agregando el chatbot a sus conversaciones. Cuando se producen interacciones de los usuarios con el chatbot, Skype se comunica con Azure Bot Service, que es el servicio en la nube donde reside el código del bot alojado en una App Service y donde se encuentra el almacenamiento de los datos en memoria de la conversación concreta. Desde el bot procedemos a comunicarnos con la API de SOCIO mediante peticiones HTTP cuando se requieran las acciones de modelado oportunas. Socio hace uso de su propia base de datos para llevar a cabo las gestiones de modelado necesarias. Por último, una vez procesada la información y generada la respuesta en el bot de Azure Bot Service, este la envía a Skype y se muestra la información a los usuarios.

3.2 Azure Bot service

Azure Bot Service es un entorno integrado proporcionado por Microsoft para la creación, implementación, despliegue y gestión de bots en la nube [21].

En los siguientes apartados se expondrán los dos elementos fundamentales de Azure Bot Service a través de los cuales se dará la funcionalidad deseada al chatbot y forman parte del diseño: Bot Framework y Azure.

3.2.1 Bot Framework

Para el desarrollo del chatbot utilizaremos la herramienta Bot Framework [22]. Se trata de un entorno que hace uso de un kit de desarrollo llamado Bot Builder SDK. Este kit cuenta con una librería que facilitará la programación, ya que permite el uso de funciones que se encargan de gestionar distintas tareas del bot. Estas tareas pueden ser la comunicación con Skype o la utilización de funcionalidades concretas que ofrecen estos bots como actividades, estado, sugerencias de opciones, tarjetas o adjuntos, de los que se hablará más en detalle en el apartado de desarrollo (Sección 4).

Por otro lado, a la hora de trabajar de forma local en la implementación se hará uso de la aplicación Bot Framework Emulator, que permite ejecutar y probar el funcionamiento del bot. Una vez desarrollado, se publicará en Azure Bot Service para que su uso esté disponible en la nube.

3.2.2 Azure

Azure es una plataforma como servicio (PaaS) en la nube que se utilizará para publicar y alojar el bot, creando los recursos necesarios para su funcionamiento. Concretamente, a través del portal de Azure es posible llevar a cabo la administración del chatbot y sus especificaciones. Uno de los recursos que es necesario es un App Service, un servicio para hospedar el bot y ejecutarlo en la nube, el cual es posible crear y configurar en Azure.

Otra de las funciones que nos permite Azure, particularmente Azure Bot Service y por la cual se ha elegido este diseño, es la posibilidad de conectar con otras plataformas. Desde Azure existe la opción de añadir canales, es decir, la capacidad de extender el bot a distintos servicios y de manera muy sencilla. En este caso el canal que agregaremos será Skype. Esto es una potente funcionalidad debido a que con la implementación de un único chatbot se puede dar soporte a multitud de plataformas si así se desea.

Como se puede apreciar en la parte superior de la figura 3-2 el bot está extendido y funcionando en Skype. En la parte inferior se muestran los diferentes servicios, plataformas y redes sociales que se pueden añadir como canales a nuestro bot fácilmente desde el portal de Azure una vez que se ha implementado.

Connect to channels

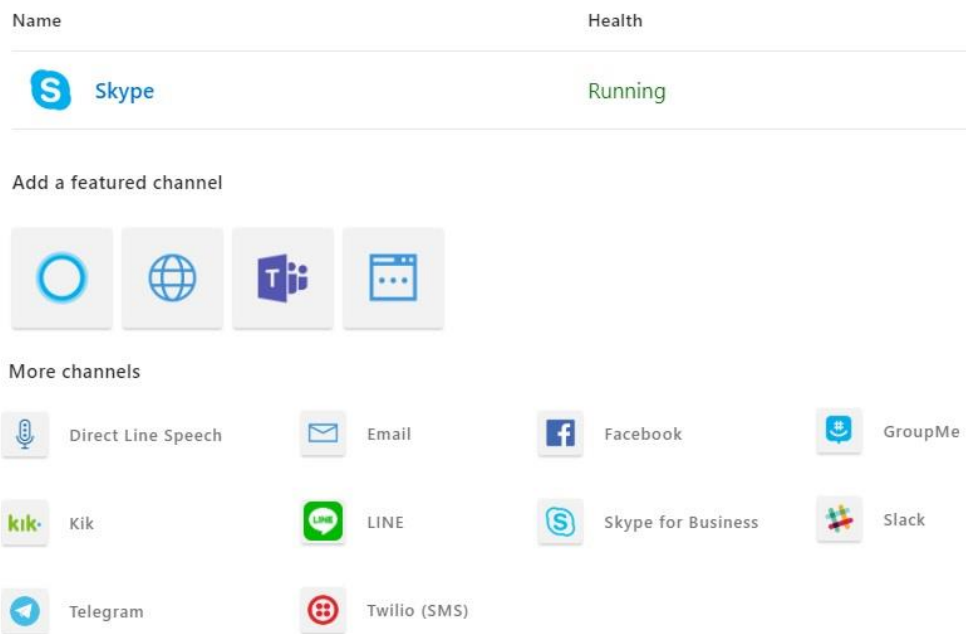


Figura 3-2: Conexión a canales en Azure

3.3 SOCIO API REST

Al usar la API de SOCIO en el diseño se da la capacidad al bot de realizar tareas de modelado colaborativo. Al ser una arquitectura REST que se basa en servicios web se permite el intercambio de información de una forma sencilla y de manera externa mediante llamadas HTTP, independientemente del lenguaje de programación que se esté usando. En el momento de ejecución en el que se requiera una acción de modelado, el bot se comunicará con la API enviando y recibiendo la información necesaria. Además de esto, SOCIO permite diferenciar la red social que se está usando, por lo que no existen problemas a la hora de añadir una red social nueva como Skype y de trabajar con distintas redes sociales a la vez en la API.

3.3.1 Funcionalidades

Las funciones que ofrece SOCIO sobre modelado son en las que se basará el chatbot para ofrecer el servicio deseado, por lo que de estas depende en mayor parte el diseño del bot y su interacción con el usuario. Por lo tanto, las principales funcionalidades que se proporcionan en SOCIO, y como consecuencia en el bot de Skype, son las siguientes:

- Creación, edición y visualización de modelos de dominio mediante interpretación del lenguaje natural del usuario.
- Creación y administración de proyectos que contendrán los modelos, cuyo administrador será el usuario creador del proyecto.

- Utilización de niveles de visibilidad en los proyectos y administración de los mismos a los usuarios: public (accesible a todos), protected (accesible a todos a nivel de lectura y con permisos a nivel de escritura) y private (privado a todos, en cuyo caso se necesitará permisos).
- Mecanismos para deshacer y rehacer acciones en la edición de un modelo.
- Validación del modelo para revisar errores de modelado.
- Obtención del historial de mensajes de edición del proyecto y posibilidad de filtrar los mismos por acción, elemento, fecha o usuario.
- Obtención del fichero donde se almacena el modelo.
- Visualización de estadísticas del proyecto sobre el número de acciones o mensajes, filtrando a su vez tanto por estadística distribuida en el tiempo o absoluta, como por usuarios totales o uno en particular. También se permite la visualización del porcentaje de autoría de cada usuario.
- Creación y edición de ramas (branches) a partir de un proyecto padre para su posterior elección por los usuarios.
- Capacidad para trabajar desde distintas redes sociales a la vez, así como la notificación de actualizaciones.

3.4 Lógica del bot

En este apartado se explicará el diseño concreto en el que se basa la lógica del chatbot, más allá de su arquitectura de funcionamiento. Para ello diferenciaremos entre su utilidad y su interfaz e interacción.

3.4.1 Principios de utilidad

El planteamiento de este bot se basa, independientemente de la funcionalidad de modelado, en el usuario, ya que es el consumidor del servicio. Para ser de utilidad al usuario el diseño se ha llevado a cabo asentándose en varios principios. Uno de ellos es buscar la solución al problema de modelado colaborativo, la cual debe realizarse de la forma más sencilla posible y en el menor número de pasos. También es importante que el usuario sepa en todo momento qué hacer y cómo hacerlo interactuando con el bot, es decir, que sea intuitivo. Por otro lado, debe ser accesible desde distintas plataformas y dispositivos, permitir la comunicación con otros usuarios y tener capacidad para interactuar con otras redes sociales. Por último, es fundamental ofrecer distintos elementos, como botones o imágenes, con los que se logrará una buena experiencia de usuario [23].

3.4.2 Interfaz e interacción

Una vez explicados los principios seguidos para hacer práctico el chatbot, se procede a exponer más en detalle los elementos específicos para el intercambio de información entre el usuario y el bot, obviando que la funcionalidad que se ofrece es la de modelado colaborativo.

Al ser un chatbot, la conversación se lleva a cabo mediante turnos de mensajes, alternando entre usuario y bot. Para proporcionar una buena experiencia de usuario el diseño se basa en el empleo de comandos o palabras concretas. La razón principal del uso de comandos es el fácil aprendizaje de la sintaxis y el resultado concreto y claro que se espera con su utilización. Combinando esta idea con el uso de turnos, la lógica y funcionalidades del bot se van desglosando en caminos de diálogo. Dependiendo de las opciones elegidas por el usuario, el bot va guiando la conversación, recopilando la información necesaria y ofreciendo distintas opciones hasta realizar la tarea deseada. Una vez hecho esto, se vuelve de nuevo al inicio del diálogo, es decir, al menú de opciones iniciales.

Para que sea intuitivo, se proporciona información concreta del uso de cada comando y se facilitará el empleo de botones que indicarán tanto los comandos como las opciones disponibles. El bot se encargará también de enviar al usuario los resultados en forma de imágenes si es necesario como, por ejemplo, el estado del modelo.

Por último, y con el objetivo de ser accesible, al implementarse en Skype este bot es multiplataforma y se puede utilizar desde distintos dispositivos. Además permite ser agregado de forma sencilla e interactuar en conversaciones tanto individuales como en grupo. En el caso de los grupos la interacción con el bot se realizará nombrando a este con “@” y, junto con la utilización del bot, Skype proporciona en estos grupos medios de comunicación y discusión entre usuarios: chats, llamadas, videollamadas y videoconferencias.

4 Desarrollo

En este capítulo se explicarán en detalle los conceptos y elementos empleados para la implementación de la funcionalidad del bot. Para este desarrollo se ha optado por la utilización del lenguaje C# y .NET.

4.1 Comunicación

La implementación de la comunicación es el aspecto fundamental a través del cual se realizará el intercambio de información entre el usuario y el bot. Para ello, se expondrá qué es lo que se intercambia para luego explicar cómo es el intercambio:

4.1.1 Contexto y actividad

En este bot, el contexto es la información que se intercambia. Se trata de un objeto que contiene toda información necesaria de la interacción del usuario con el bot. En esta información destacaremos principalmente la actividad.

La actividad es un elemento fundamental que nos indica el tipo de interacción que se ha realizado y sus atributos. Por ejemplo, nos sirve para saber si la actividad recibida es un mensaje y, a su vez, acceder al mismo y procesarlo. Esta es la actividad a la que se dará más uso en el bot, pero existen otras automáticas como la de un usuario nuevo en la conversación, a partir de la cual se realizan las acciones oportunas, como proceder a saludarle.

Tanto en el contexto como en la actividad podremos encontrar otra información necesaria como el usuario que realiza la acción, o funciones de respuesta, a las que se da uso en la implementación.

4.1.2 Recepción y envío

Para la recepción de información por parte del usuario, es decir, del contexto, se hará uso de la función `OnTurnAsync` del SDK. Dicha función es llamada cada vez que se realice una interacción con el bot y, por lo tanto, es en ella donde se procesará la información llegada por el usuario y dependiendo de esta se ejecuta la funcionalidad de los distintos comandos u opciones. Esta es la función principal sobre la que se basa el bot, ya que en ella realizamos la recepción y desde ella procedemos al envío de información al usuario, los dos elementos necesarios para el funcionamiento de un chatbot.

Por otro lado, para el envío de información se emplea la función `SendActivityAsync`, que también se encuentra en el SDK. Esta es una función que se encuentra en el contexto, con el objetivo de poder responder al usuario que ha realizado la interacción. En el bot se usa principalmente para enviar mensajes, aunque tiene otras utilidades que se comentarán a continuación.

4.2 Sugerencia de acciones

Como se ha comentado anteriormente, el envío de información no tiene que ser necesariamente un mensaje. Uno de los elementos que se pueden enviar son sugerencias de acciones. Estas sugerencias se desarrollan creando una respuesta para la actividad, añadiendo a este objeto la lista de acciones posibles y enviando dicha respuesta al usuario. En la figura 4-1 se muestra un ejemplo de visualización de las sugerencias. Cuando el usuario pulsa en alguna, se envía el valor de la acción correspondiente al bot.

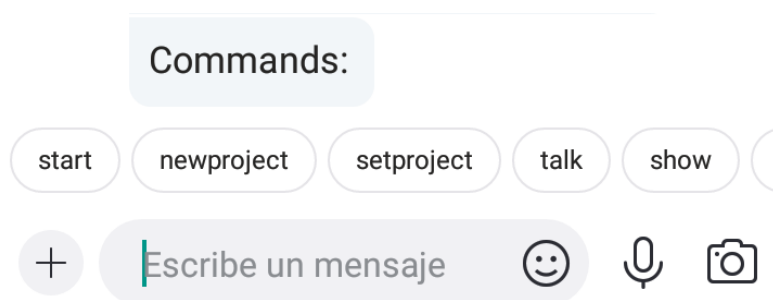


Figura 4-1: Sugerencia de acciones

Este tipo de elemento se proporciona en todo momento para dar información al usuario sobre los comandos u opciones disponibles para que sepa cómo interactuar con el bot en cualquier situación.

4.3 Elementos multimedia

Los elementos multimedia son otro componente que se puede enviar al usuario. En el desarrollo se dará el uso principalmente al envío de imágenes. Para ello, y como en los demás elementos, se crea una respuesta para la actividad, en ella se añaden los adjuntos y se procede a enviárselo al usuario.

4.3.1 Tarjetas

A parte de los adjuntos para el envío multimedia, se hará uso de tarjetas enriquecidas. En estas tarjetas se combinan elementos multimedia, acciones o texto, entre otras opciones y se muestra visualmente de una forma simple. En la figura 4-2 se puede observar un ejemplo de una tarjeta en la que se combina texto con un link a una página de ayuda.

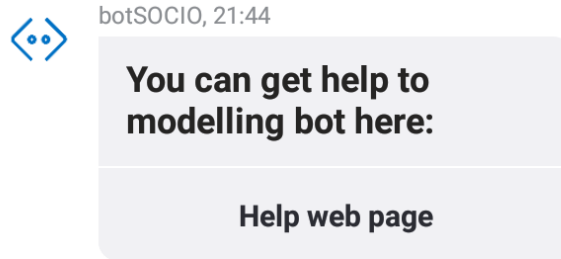


Figura 4-2: Ejemplo de tarjeta enriquecida

4.4 Almacenamiento de datos

Debido al tipo de bot que se ha desarrollado y su funcionalidad, se requiere de la implementación de medios de almacenamiento con el fin de conservar los datos. Para ello, se diferencia entre dos tipos de datos que conllevan dos tipos de almacenamiento distintos: los datos de modelado y los datos de la conversación.

4.4.1 Modelado

Los datos de modelado son todos aquellos relacionados con la utilización de la API de SOCIO. Estos datos contienen toda la información sobre la funcionalidad de modelado colaborativo que se proporciona en el bot como usuarios, canales, proyectos o modelos. Dicha información se encuentra almacenada en la base de datos de SOCIO, por lo que únicamente se implementarán las llamadas necesarias a la API tanto para la creación y administración de estos datos como para la obtención de los mismos.

4.4.2 Conversación

Para que las conversaciones con los usuarios tengan sentido, en ocasiones es necesario llevar a cabo un almacenamiento de los datos de la conversación. Con el fin de cubrir esta necesidad, ha sido necesaria la implementación de un estado. El estado es un mecanismo de almacenamiento en memoria y, al igual que la ejecución del bot, es individual para cada conversación que se mantenga con el mismo. Dicho estado se encuentra desarrollado en un módulo y este deberá ser cargado y guardado en cada interacción con el fin de preservar los datos.

La importancia de mantener esta información recae sobre todo a la hora de saber cuál es el punto de la conversación con el usuario en el que nos encontramos en cada interacción, con el fin de mantener una coherencia en el diálogo y saber qué preguntar o que mostrar en cada momento.

4.5 Llamadas a SOCIO API REST

Como ya se ha comentado anteriormente, para añadir al bot las funcionalidades de modelado colaborativo se hará uso de la API REST de SOCIO, la cual está basada en servicios web y, por lo tanto, se deben desarrollar las peticiones necesarias para su utilización.

Para llevar a cabo dichas llamadas se implementarán peticiones HTTP. Dependiendo de la funcionalidad requerida, estas llamadas se realizarán a la dirección URI correspondiente de la API. El intercambio de información en las peticiones se llevará a cabo principalmente mediante objetos JSON, del cual extraeremos o añadiremos los atributos necesarios dependiendo de la utilización. Para la obtención de datos más concretos como imágenes o ficheros, se utilizará el formato octet-stream (formato de archivos de datos binarios multimedia).

Según su objetivo, se diferencian dos tipos de peticiones HTTP en esta implementación:

- **GET:** llamadas realizadas principalmente para obtener una información concreta de la API. Se hace una llamada a la URI correspondiente y se reciben los datos.
- **POST:** peticiones llevadas a cabo para enviar datos a la API y recibir la información deseada. Dichos datos se envían como argumento en un objeto JSON.

La correcta implementación y utilización de las llamadas en el momento correspondiente, en el orden adecuado y con el uso y tratamiento de los datos necesarios, nos aportará a nuestro chatbot la funcionalidad de modelado. En el código del bot se ha desarrollado un módulo concreto para llevar a cabo estas peticiones y el intercambio de información.

5 Herramienta

En este apartado se expondrá cómo es el funcionamiento del chatbot en Skype y la forma en la que poder realizar las principales tareas que se pueden llevar a cabo. Para ello, se mostrará su interacción con los usuarios y la visualización de la información mediante la interfaz gráfica.

Como ya hemos visto anteriormente, la forma de interactuar con el bot se basará en comandos. Dentro de los distintos caminos que puede tener el diálogo para recopilar información, también se hará uso de la elección de distintas opciones en caso de que sea necesario para una tarea.

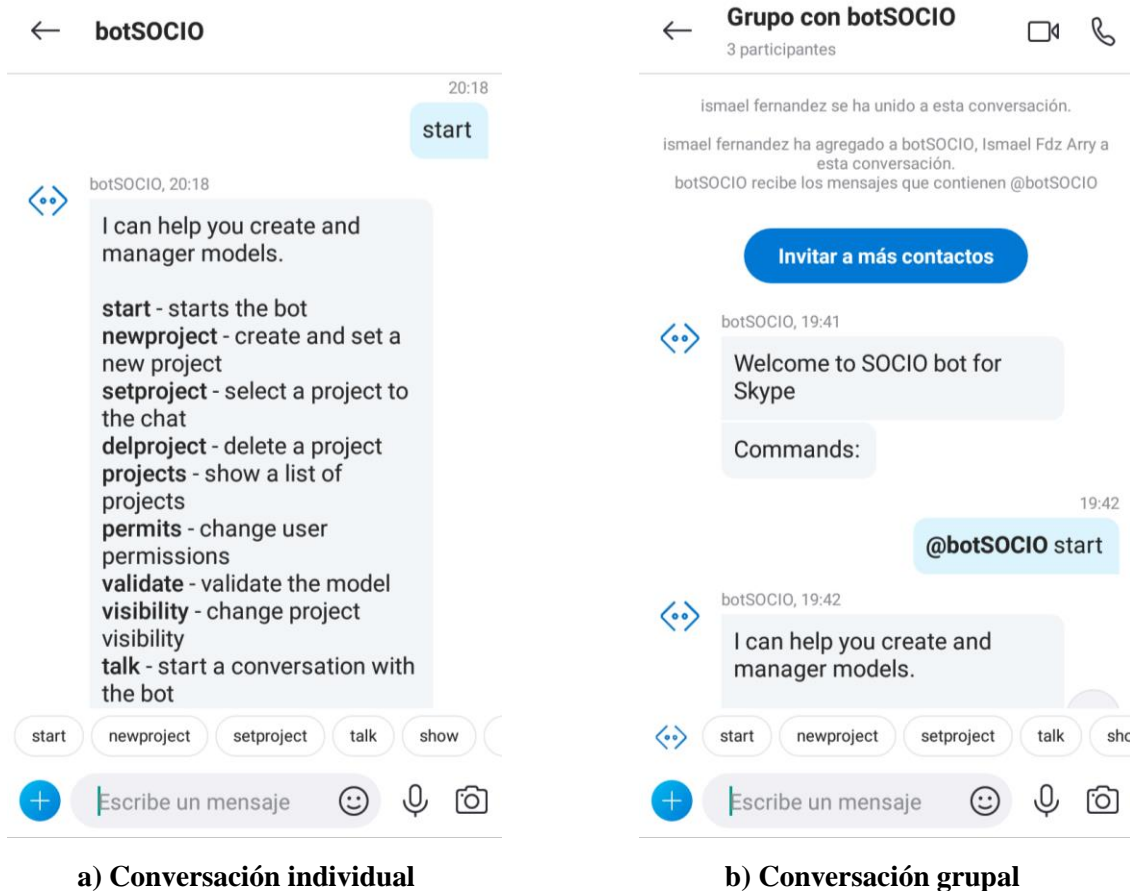
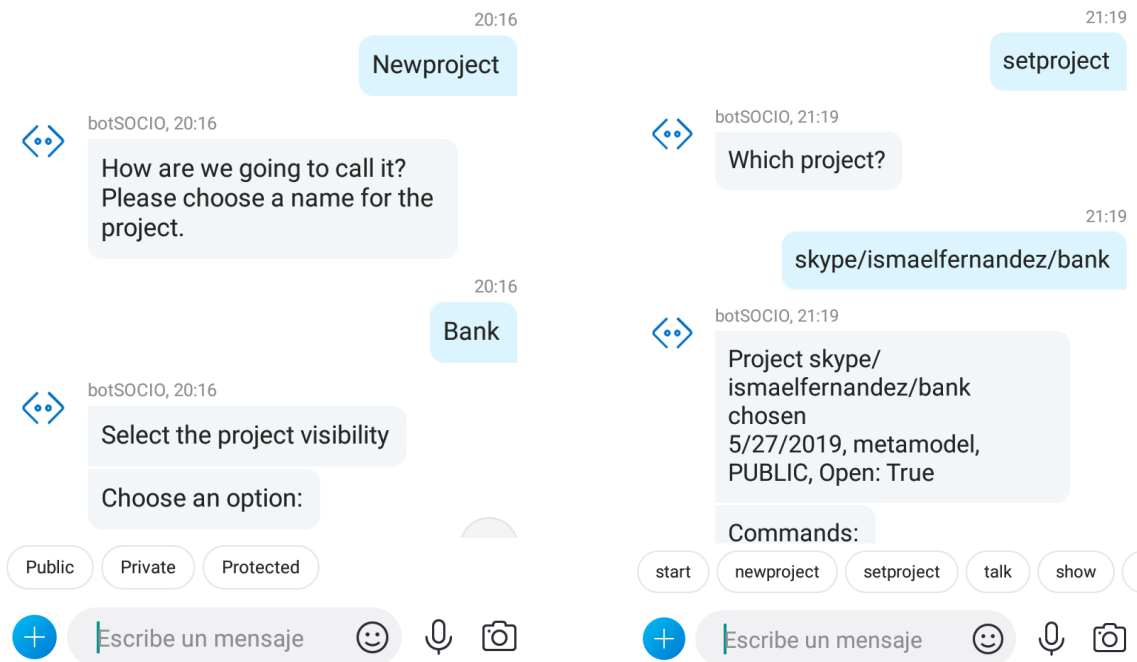


Figura 5-1: Comando *start* en conversación individual y grupal

En la figura 5-1 observamos el funcionamiento del bot en una conversación individual (a) y en otra grupal (b). En la conversación individual se conversa directamente con el bot, mientras que en la grupal podremos agregar a más usuarios para que interactúen con él o vean dicha conversación. Para enviar un mensaje el bot en el chat grupal, será necesario mencionarle con “@”. Esto permite que el resto de usuarios pueda dialogar y discutir sobre un modelo, e incluso realizar llamadas, videollamadas o conferencias, como se observa en los botones de la esquina superior derecha. En todo momento se encontrarán los botones de comandos u opciones sugeridas, en este caso concreto se ha hecho uso del comando *start*

que muestra la descripción de los distintos comandos. Adicionalmente, se podrá utilizar el comando *help*, que proporciona un link a una página de ayuda.

Con el fin de poder trabajar un modelo, primero se debe crear un proyecto, o bien seleccionar uno que ya se encuentre creado. Para ello se utilizará el comando *newproject* para crearlo o el comando *setproject* para seleccionar uno (Figura 5-2 a y b). Los proyectos se definen de la forma *Canal/Usuario/Proyecto*. Es importante destacar que dicho proyecto elegido no tiene por qué ser de Skype, ya que Socio puede estar implementado en distintas redes sociales y desde este bot de Skype poder trabajar en dichos proyectos y modelos.



a) Comando *newproject*

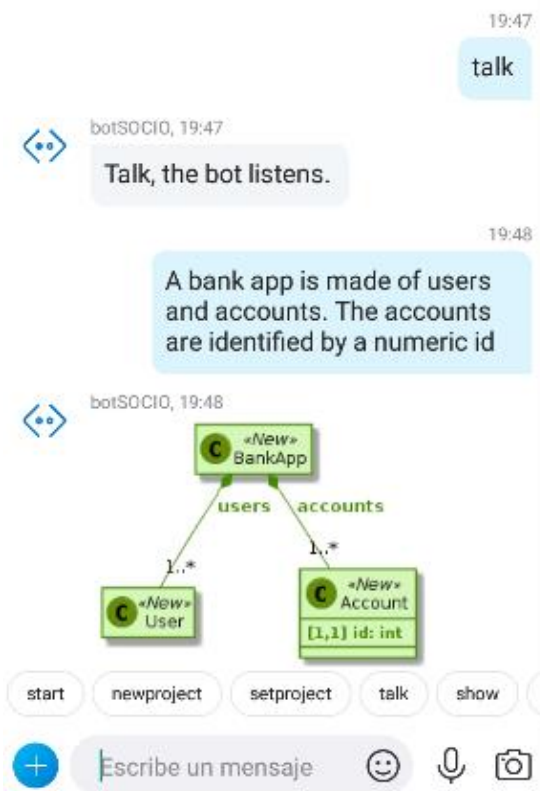
b) Comando *setproject*

Figura 5-2: Comandos *newproject* y *setproject*

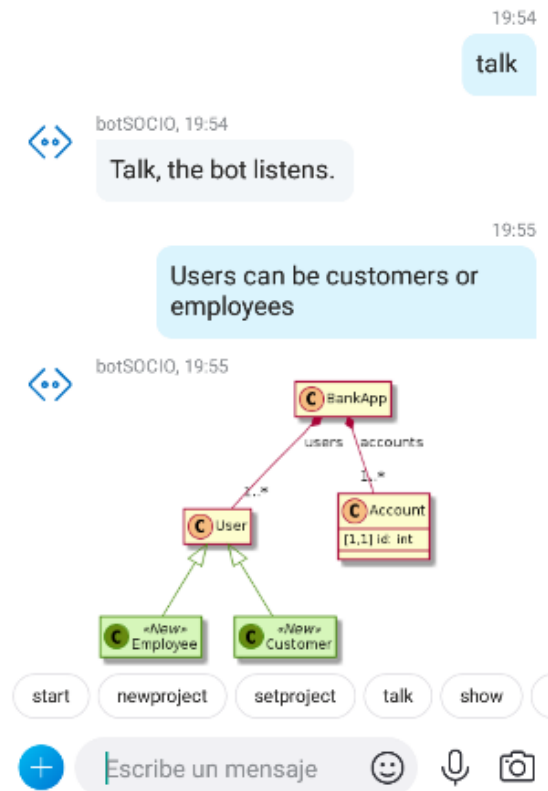
Para la administración de los proyectos existen los comandos *delproject*, para eliminar un proyecto, *projects*, para visualizar los proyectos disponibles, el comando *visibility*, para cambiar la visibilidad del proyecto a public, private o protected, y el comando *permits*, que sirve para dar permisos a los usuarios en los proyectos que lo requieran. También es posible obtener el historial del proyecto mediante el comando *history*.

Como se puede observar en figuras como la 5-2, el bot va preguntando, ofreciendo las opciones oportunas mediante botones y recopilando información necesaria para facilitar la interacción y las tareas al usuario.

Una vez creado o seleccionado el proyecto se puede trabajar en el modelo. Para crearlo y editarlo se hará uso del propio lenguaje natural del usuario, el cual procesará Socio y devolverá el modelo actualizado. Para ello se empleará el comando *talk* y se introducirá el mensaje de edición, como se puede ver en la Figura 5-3 a y b.



a) Comando *talk*



b) Continuación de uso de *talk*

Figura 5-3: Edición de modelos mediante el comando *talk*

Cada mensaje de modelado se puede deshacer y rehacer si el usuario lo considera necesario. Para ello se utilizarán los comandos *undo* y *redo*, respectivamente, con el objetivo de poder rectificar acciones. En la figura 5-4 a y b se muestra el uso de estos comandos a partir del modelo de la figura 5-3 b anterior.



a) Comando *undo*



b) Comando *redo*

Figura 5-4: Comandos *undo* y *redo*

Para mostrar la imagen del modelo en cualquier momento se puede hacer uso del comando *show*. Además existe la opción de ver el historial de mensajes de edición utilizando el comando *messages*, pudiendo filtrar por acción, elemento, fecha o usuario. Por otro lado, también es posible validar el modelo para encontrar errores de modelado mediante el comando *validate*, o generar un link para descargar el código del modelo en formato *ecore* empleando el comando *model*.

La posibilidad de trabajar desde distintas redes sociales hace necesario que el usuario sea informado si se han producido actualizaciones en un modelo. Para ello, se darán a conocer estas actualizaciones, en caso de que existan, antes de realizar una acción de modelado. También es posible consultar las actualizaciones manualmente mediante el comando *updates*.

Con el objetivo de ampliar las opciones en la toma de decisiones sobre un modelo, existe la posibilidad de crear ramas a partir de un proyecto padre en cualquier momento de la edición. Estas ramas se podrán editar por separado y elegir una para establecerla en el proyecto padre si se desea. Para crear una rama se usará el comando *branch* y para establecer una en el proyecto padre se utilizará el comando *setbranch*.

En la figura 5-5 se muestra el uso del comando *updates* (a), en el que se informa de una acción de modelado de un usuario de Telegram. También se puede observar un ejemplo de edición en una rama (b) sobre un proyecto padre, el cual destaca en color azul.

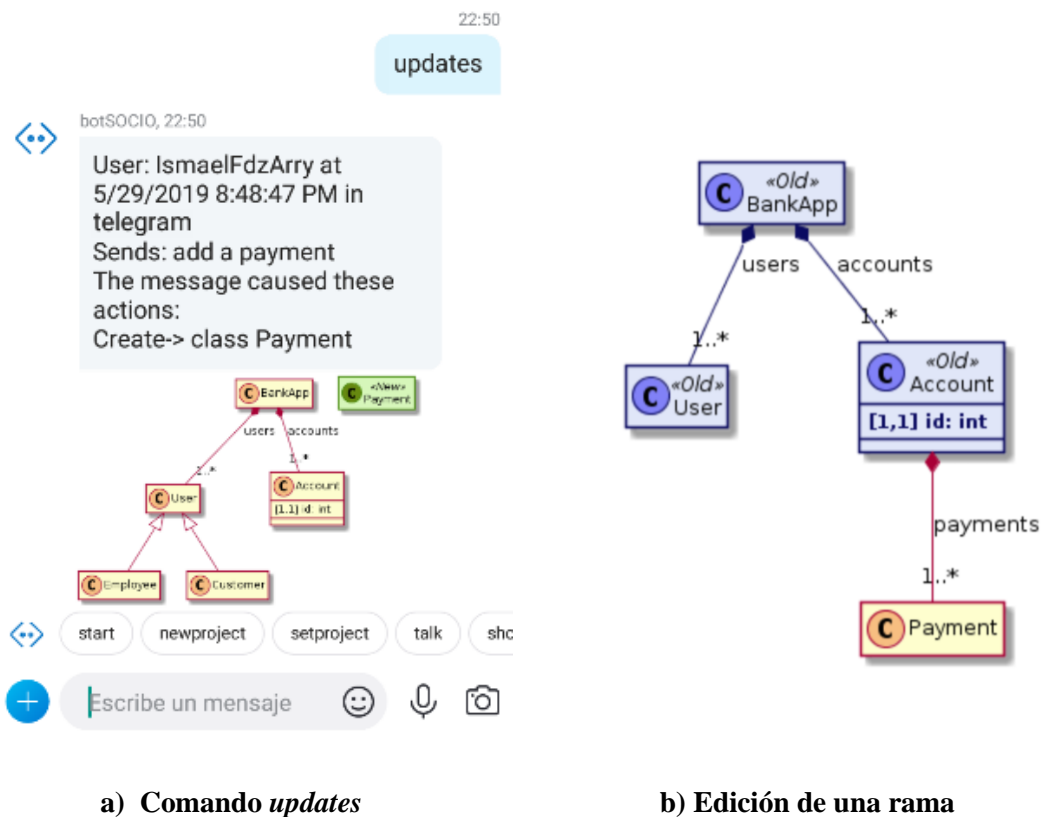
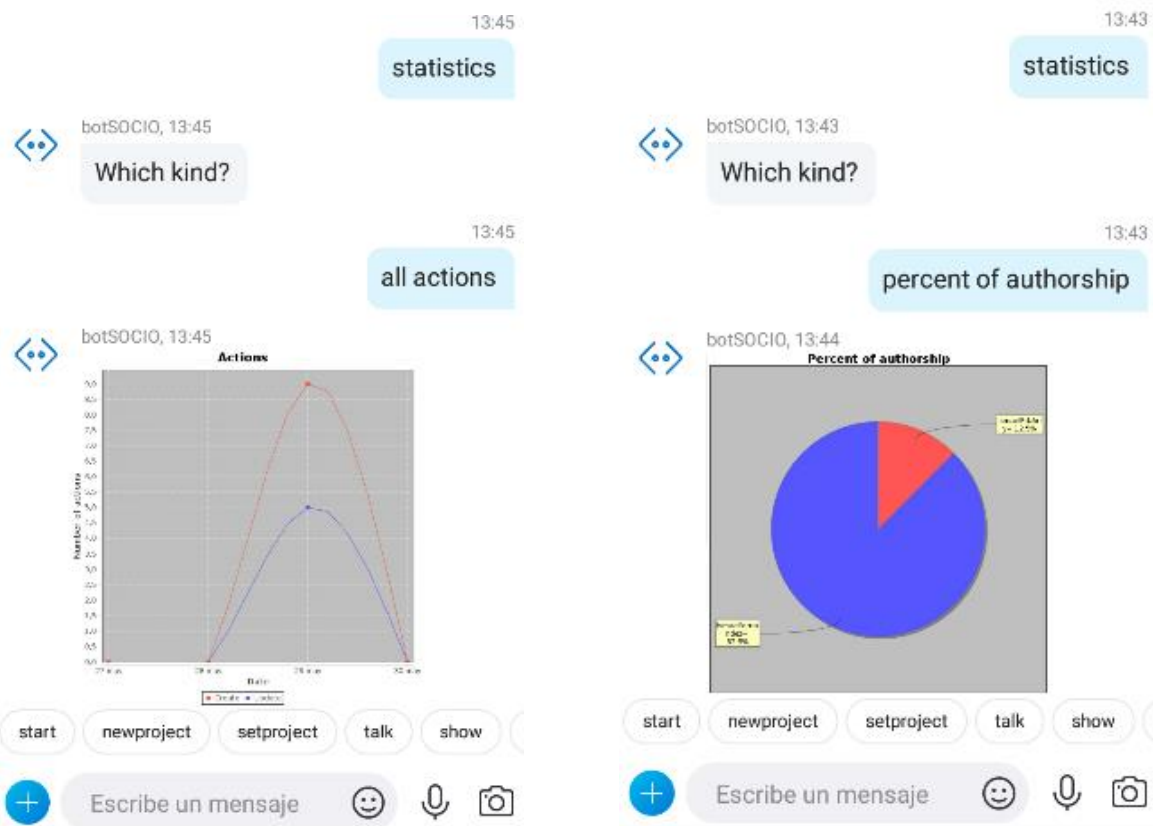


Figura 5-5: Comando *updates* y ejemplo de rama

Por último, para que sea posible realizar un seguimiento del modelado por parte de los usuarios, es posible generar y visualizar estadísticas. Estas estadísticas están basadas en el número de acciones o mensajes de todos los usuarios o de uno en concreto. Existen distintas formas de representación de estas, como distribuidas en el tiempo o de forma absoluta, y también es posible mostrar el porcentaje de autoría del modelo. Todas estas opciones de visualización de estadísticas se ofrecerán al emplear el comando *statistics*.

En la figura 5-6 se puede observar un ejemplo del uso de este comando para estadísticas de acciones y para mostrar el porcentaje de autoría.

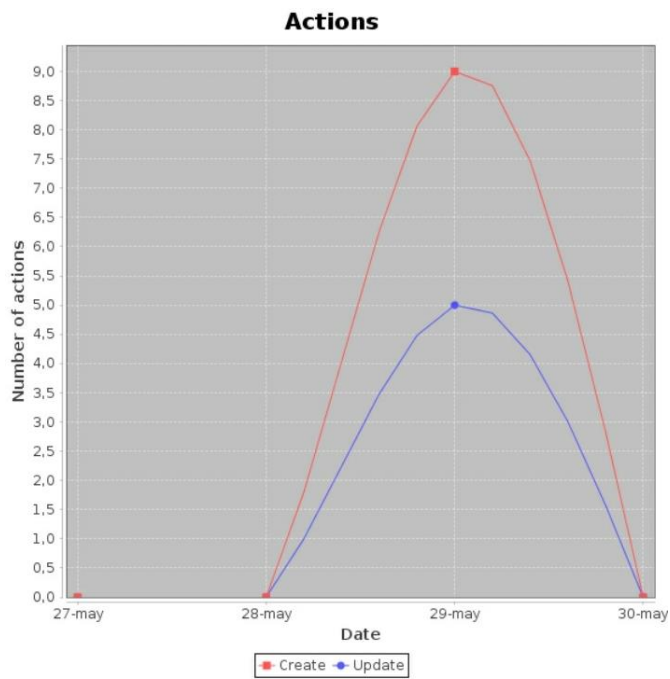


a) Comando *statistics* para acciones

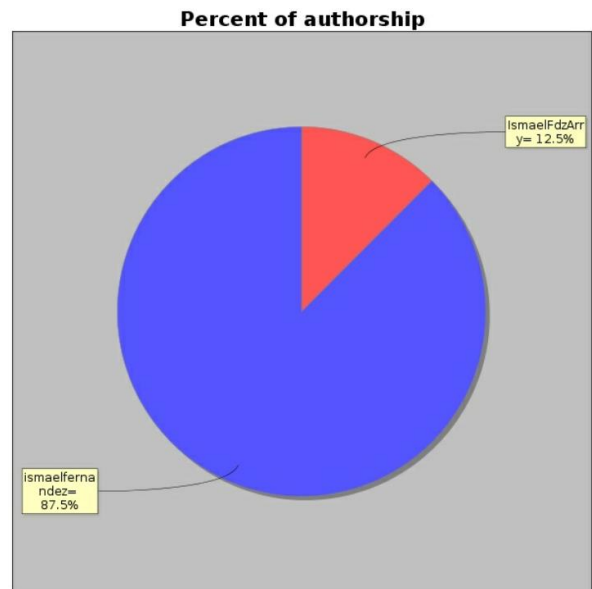
b) Comando *statistics* para porcentaje de autoría

Figura 5-6: Comando *statistics* para acciones y porcentaje de autoría

Para ver más en detalle las estadísticas recibidas en la figura 5-6 a y b, en la figura 5-7 a y b, respectivamente, se muestra una ampliación de las mismas.



a) Detalle de la figura 5-6 a



b) Detalle de la figura 5-6 b

Figura 5-7: Ampliación de la figura 5-6

6 Conclusiones y trabajo futuro

En esta sección se dan a conocer las conclusiones y resultados finales una vez realizado el trabajo. Por otro lado, se plantean una serie de trabajos futuros con el fin de mejorar este proyecto.

6.1 Conclusiones

En este trabajo se ha desarrollado un chatbot dedicado al modelado colaborativo en la red social de mensajería Skype. Esta herramienta proporciona asistencia mediante un bot a los usuarios para la elaboración de tareas de modelado teniendo en cuenta la colaboración y discusión entre personas mediante el uso de la API de SOCIO.

El desarrollo de modelos de forma conjunta se lleva a cabo ofreciendo la creación y la administración de proyectos, en los cuales se puede gestionar la visibilidad y los permisos para los usuarios. Se da la capacidad de visualizar y editar los modelos de los proyectos por medio del lenguaje natural, pudiendo deshacer o rehacer las acciones de edición que se consideren oportunas, e incluso pudiendo crear branches desde un modelo padre. Por otro lado, es posible visualizar y filtrar el historial de edición, así como generar distintas estadísticas del proyecto. Adicionalmente se permite la interacción con usuarios de otras redes sociales que también usen la API de SOCIO. Para todas estas operaciones, el bot se encarga de guiar al usuario, proporcionando y facilitando distintos comandos y opciones mediante botones.

Su diseño, implementación y administración mediante Azure Bot Service ofrece un servicio continuo en la nube para que los usuarios puedan hacer uso de este con total libertad en Skype, pudiendo extenderse incluso a otras redes sociales. Concretamente, mediante la utilización del bot en Skype, es posible agregarlo a conversaciones individuales y grupales para su utilización, siendo este bot individual para cada uno de los chats. Además, en los grupos se proporcionan en todo momento los medios de discusión necesarios, ya sea mediante mensajes, llamadas, videollamadas o conferencias.

La importancia de este trabajo recae principalmente en el desarrollo de software, ya que el fin es aumentar la productividad a un bajo coste mediante la creación de modelos y, a su vez, facilitar el modelado y la colaboración a través de un chatbot. En este ámbito, actualmente no se proporcionan herramientas que ofrezcan este tipo de servicios, por lo que el producto de este trabajo es una solución para el modelado colaborativo.

6.2 Trabajo futuro

Una vez desarrollado este trabajo que cumple con los requisitos y la funcionalidad deseada, sería posible realizar ciertas mejoras de esta versión de cara al futuro trabajando en los siguientes aspectos:

- Implementar llamadas de voz con el bot. Actualmente, Azure Bot Service se encuentra en proceso de desarrollo y mejora de este servicio, por lo que en un futuro cercano sería de utilidad para el proyecto. Concretamente, se realizaría una llamada al bot a la hora de realizar una edición en un modelo con el fin de procesar la voz y transformarla a texto.
- Extender el chatbot a distintas redes sociales. Aprovechando la propiedad de Azure Bot Service en la que es posible añadir canales al bot y la propiedad de la API de SOCIO de distinguir distintas redes sociales, es posible expandir el chatbot a distintas plataformas con una única implementación.
- Añadir reconocimiento de lenguaje natural al bot. Para hacer más realista la conversación, más allá de la utilización de comandos, es posible implementar este procesamiento para saber que quiere hacer el usuario conversacional y contextualmente mediante un lenguaje más natural, independientemente del procesamiento que realiza SOCIO en la edición de modelos.

Referencias

- [1] Vicente Pelechano, Antonio Vallecillo, Juan Manuel Vara, Cristina Vicente-Chicote, Jesús García Molina, Félix O. García Rubio. Desarrollo del software dirigido por modelos: Conceptos, métodos y herramientas. Ra-Ma Editorial, 2013.
- [2] Ingeniería dirigida por modelos:
https://es.wikipedia.org/wiki/Ingenier%C3%ADa_dirigida_por_modelos [Última fecha de acceso: 13/06/19]
- [3] Sara Pérez Soler. Modelado colaborativo en lenguaje natural a través de redes sociales. 2018.
- [4] Creately: <https://creately.com> [Última fecha de acceso: 13/06/19]
- [5] Cacao: <https://cacao.com> [Última fecha de acceso: 13/06/19]
- [6] Gliffy: <https://www.gliffy.com> [Última fecha de acceso: 13/06/19]
- [7] Eugene Syriani, Hans Vangheluwe, Raphael Mannadiar, Conner Hansen, Simon Van Mierlo and Huseyin Ergin. AToMPM: A Web-based Modeling Environment. 2013.
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.407.6965>
- [8] WhatsApp: <https://www.whatsapp.com> [Última fecha de acceso: 13/06/19]
- [9] Facebook Messenger: <https://www.messenger.com> [Última fecha de acceso: 13/06/19]
- [10] Telegram: <https://telegram.org> [Última fecha de acceso: 13/06/19]
- [11] Hangouts: <https://hangouts.google.com> [Última fecha de acceso: 13/06/19]
- [12] Skype: <https://www.skype.com> [Última fecha de acceso: 13/06/19]
- [13] Skype: <https://es.wikipedia.org/wiki/Skype> [Última fecha de acceso: 13/06/19]
- [14] Chatbot Report 2018. Global Trends an Analysis:
<https://chatbotsmagazine.com/chatbot-report-2018-global-trends-and-analysis-4d8bbe4d924b> [Última fecha de acceso: 13/06/19]
- [15] Jordi Cabot. Bots para optimizar el desarrollo de software. 2018.
<https://ingenieriadesoftware.es/bots-desarrollo-software-open-source/> [Última fecha de acceso: 13/06/19]
- [16] Prophet: <http://groups.csail.mit.edu/pac/patchgen> [Última fecha de acceso: 13/06/19]
- [17] CodeBot: <https://slack.com/apps/A2YG4UA3A-codebot> [Última fecha de acceso: 13/06/19]
- [18] CodeClimate: <https://codeclimate.com> [Última fecha de acceso: 13/06/19]
- [19] Spotbot: <https://spotbot.qa> [Última fecha de acceso: 13/06/19]

- [20] NLSQL: <https://www.nlsql.com> [Última fecha de acceso: 13/06/19]
- [21] Srikanth Machiraju, Ritesh Modi. *Developing Bots with Microsoft Bots Framework: Create Intelligent Bots using MS Bot Framework and Azure Cognitive Services*. Apress, 2018.
- [22] Bot Framework: <https://dev.botframework.com> [Última fecha de acceso: 13/06/19]
- [23] Principios de diseño de bots: <https://docs.microsoft.com/es-es/azure/bot-service/bot-service-design-principles?view=azure-bot-service-3.0> [Última fecha de acceso: 13/06/19]

Glosario

.NET	Framework de Microsoft dedicado a la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.
Framework	Entorno de trabajo para implementar y solucionar un tipo de problema.
C#	Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.
Azure	Plataforma de Microsoft que ofrece servicios en la nube.
Skype	Software que permite comunicaciones de texto, voz y video sobre Internet.
API	Application Programming Interface.
REST	Representational State Transfer.
HTTP	Hypertext Transfer Protocol.
JSON	JavaScript Object Notation.
URI	Uniform Resource Identifier.
SDK	Software Development Kit.
Octet-stream	Formato para archivos de datos binarios que pueden encontrarse en cualquier formato multimedia.
PaaS	Platform as a Service