Facultad de Ciencias

Implementación de un chatbot en Azure integrado con Teams y Dynamics

(Implementation of a chatbot in Azure integrated with Teams & Dynamics)

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Rubén Calleja Reigadas

Director: Diego García Saiz

Co-Director: Daniel Manuel Tarrío Quintela

Julio – 2021

Índice de Contenido

Índice de Figuras

Índice de Tablas

Resumen

En este trabajo he realizado el desarrollo e implementación de un chatbot en Azure integrado con Teams y con Dynamics. El objetivo principal de este chatbot es la interacción con los gerentes de CIC, que serán los usuarios de esta herramienta, para facilitarles la realización de tareas cotidianas. Algunas de estas tareas son la consulta o actualización de ciertos registros sobre la plataforma de Dynamics, los cuales servirán a los usuarios para obtener información comercial de las diferentes ventas de CIC. Este chatbot está diseñado con una interfaz sencilla ocultando la complejidad de las funciones al usuario, ya que les muestra todas las posibles opciones con las que les puede ayudar, guiándoles por ellas. Todo ello sin olvidarse de los usuarios más experimentados, que también podrán realizar estas acciones con oraciones en lenguaje natural de forma más rápida, permitiendo así el uso tanto a usuarios más avanzados como a principiantes. Además, el bot, les envía un mensaje de forma automática cada cierto tiempo con tareas en las que les puede ayudar. Asimismo, la integración con Teams facilita mucho su uso, ya que no es necesario instalar ni configurar ningún software especial, basta con comenzar un chat con dicho bot y así ya tendremos a nuestra disposición toda su funcionalidad.

Abstract

In this work I do the development and implementation of a chatbot in Azure integrated with Teams and Dynamics. The main objective of this chatbot is the interaction with the CIC managers, who will be the users of this tool, to facilitate the performance of daily tasks. Some of these tasks are the query or update of certain records on the Dynamics platform, which will be used by users to obtain commercial information on the different CIC sales. This chatbot is designed with a simple interface hiding the complexity of the functions from the user, since it shows them all the possible options with which it can help them, guiding them through them. All this without forgetting the more experienced users, who will also be able to perform these actions with sentences in natural language more quickly, thus allowing the use of both more advanced users and beginners. In addition, the bot automatically sends them a message from time to time with tasks in which it can help them. Likewise, the integration with Teams makes its use much easier, since it is not necessary to install or configure any special software, just start a chat with said bot and then we will have all its functionality at our disposal.

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia y a mis amigos por estar ahí siempre, apoyándome y confiando en mi en todo momento.

En segundo lugar, a mi tutor de este trabajo, Diego García Saiz, por ayudarme y aconsejarme a lo largo de la ejecución de este proyecto.

También me gustaría agradecer a CIC por la oportunidad de realizar este proyecto con ellos ya que es una empresa líder en el sector y todos los compañeros con los que he trabajado han hecho que la estancia y el desarrollo del proyecto sea más ameno.

Por último, pero no menos importante, agradecer a la Universidad de Cantabria por la formación impartida a lo largo de todo el grado, la cual ha hecho posible el desarrollo de este proyecto con éxito.

1. Introducción

En los últimos años se ha producido una gran expansión de las empresas hacia trabajos en la nube, debido a las grandes ventajas que esto supone, como pueden ser pagar solo por lo que usas, no necesitar equipos físicos en la empresa o no mantenerlos entre otras muchas. Asimismo, el uso de la inteligencia artificial está ampliamente extendido. Existen varios tipos de inteligencia artificial, débil, general y fuerte. La inteligencia artificial débil, consiste en que una máquina perciba su entorno y actúe en consecuencia para conseguir los mejores resultados posibles, establecidos un objetivo y un entorno concretos.[10]

En relación con todo lo anteriormente comentado, y teniendo en cuenta el gran crecimiento de los Customer Relationship Management (CRM) o gestión de la relación con el cliente en las empresas, se propone desarrollar un bot para interactuar con los CRM que estén desarrollados en Dynamics facilitando así la labor del usuario, en este caso, los gerentes de CIC. En concreto este trabajo se desarrollará en Consulting Informático (CIC), que dispone de un CRM interno sobre el cual trabaja este bot. Este CRM requiere de conocimientos previos sobre la plataforma y su funcionamiento, además de ser necesario acceder a ella, con todo lo que esto conlleva, como el uso de numerosos recursos y grandes tiempos de carga, entre otras.

Por todo esto se ha decidido desarrollar un chatbot en la nube de Azure para evitar el mantenimiento de sistemas y poder pagar exclusivamente por lo que se usa. Por otra parte, se ha empleado software de inteligencia artificial, ya que este bot va a interactuar con personas y es necesario que el contacto con él sea lo más natural posible, acortando así la distancia entre persona y computador.

Al estar extendida como herramienta de trabajo y de comunicación en la empresa Teams, se ha decidido que dicho bot esté integrado y funcione sobre esta aplicación, para comodidad de los usuarios, al no tener que instalar ningún software nuevo, siendo necesario solo empezar una conversación con dicho bot.

También este bot elimina la necesidad de conocer el entorno de Dynamics y tener que acceder a él. Este entorno consiste en una cartera de aplicaciones empresariales inteligentes. De esta forma, evitamos el acceso y el conocimiento de Dynamics al realizar todo a través de la aplicación de chat de Teams que está estandarizada en la empresa como se ha mencionado anteriormente.[9]

2. Objetivos

El objetivo principal de este trabajo es ayudar al usuario en sus tareas cotidianas realizadas a través del CRM de CIC. Los gerentes o usuarios de este bot son los responsables de cada departamento o gerencia en CIC y, por tanto, se encargan de dirigir y asegurar el correcto funcionamiento de este, encargándose principalmente de buscar nuevas ofertas con los clientes adquiridos, buscar nuevos clientes, realizar ventas y obtener el mayor beneficio posible de todo esto. Para llevar a cabo esta labor, realizan varias tareas relacionadas con el CRM implementado en la empresa para las cuales se requiere tiempo y conocer las diferentes interfaces de la plataforma de Dynamics. Gracias a este bot, no será necesario dicho conocimiento de la interfaz y se realizará de manera más rápida, ya que simplemente los usuarios tendrán que decirle al bot lo que quieren realizar y este responderá al instante con los datos solicitados o con los cambios necesarios realizados.

Las tareas que necesitamos que realice este bot son:

- Consultar los registros de una entidad concreta del CRM, con diferentes filtros. Por ejemplo, consultar las oportunidades que se encuentran abiertas.

- Cambiar la fecha de vencimiento de una oportunidad concreta. Esta es una tarea muy recurrente y pesada para el usuario ya que se le puede olvidar que va a vencer una oportunidad y que es necesario actualizar dicha fecha si no quiere perderla.

- Consultar las últimas oportunidades. Es una funcionalidad bastante frecuente, porqué normalmente al usuario no le interesa ver todas las oportunidades, sino las más recientes y de esta forma no se le satura con tanta información.

- Ver el detalle de una oportunidad concreta. Ya que al mostrarle al usuario una información resumida para mostrar la información más relevante, puede que el usuario en algún momento necesite más información sobre alguna oportunidad concreta.

- Obtener las oportunidades caducadas o cerca de caducar. Las oportunidades tienen una fecha de inicio prevista y está no puede ser nunca anterior al día actual. Por tanto, es necesario saber que oportunidades son necesarias actualizar.

- Consultar las oportunidades en función de su estado. Las oportunidades tienen un estado, abierta, perdida y ofertando entre otros. El usuario necesita poder obtener las oportunidades en un determinado estado para poder hacer así sus tareas más fácilmente.

- Notificar al usuario mensualmente de cuáles son sus oportunidades caducadas o cercanas a caducar, para que este pueda actualizarlas y no tener así ninguna oportunidad caducada.

3. Software utilizado

El software utilizado durante el desarrollo de este trabajo ha sido muy diverso y a continuación se va a citar el nombre de dicho software junto con una breve descripción.

En primer lugar, tenemos Visual Studio Enterprise 2019, que ha sido la principal plataforma para el desarrollo de todo el código fuente del bot. Es un software muy conocido de desarrollo de código en .NET que proporciona muchas facilidades en lo que a desarrollo de software se refiere. [1]

Por otro lado, tenemos la otra gran parte del proyecto, que es la que corresponde a la inteligencia artificial, para la cual se ha utilizado el entorno web que proporciona Microsoft para trabajar con el lenguaje LUIS, el cual permite a nuestro chatbot ser más humano y reconocer de manera sencilla el lenguaje natural, evitando tener que usar expresiones forzadas que no son naturales por parte del usuario. Este entorno nos da la posibilidad de crear los diferentes 'intents' que lanzará el bot y definir las diferentes entidades que va a reconocer. [2]

Además, se ha utilizado el portal de Azure para alojar diferentes funciones de Azure y gestionarlas de manera correcta. Dichas funciones nos permiten por ejemplo mandarle un mensaje al usuario cada cierto tiempo. El portal de Azure nos permite almacenar diferentes funciones en la nube y de forma que no es necesario tener un equipo disponible para su ejecución. Este portal también nos permite depurar las funciones que tenemos creadas, ejecutarlas y desactivarlas entre otras cosas. Cabe destacar también que desde este portal también se gestiona al bot, pudiendo realizar acciones similares a las ya mencionadas para las funciones. [3]

También ha sido utilizado Git para llevar un correcto control de versiones de todo el proyecto. Aunque Git como tal no es un software, sí que se ha utilizado software específico como Sourcetree o GitHub para llevar de manera más cómoda la gestión de la configuración. Este software permite principalmente llevar un control de versiones usando Git, sin necesidad de estar trabajando mediante línea de comandos, por ejemplo, ya que Sourcetree proporciona un entorno gráfico sobre el que trabajar. [4]

Por otra parte, se utilizó Postman para realizar el envío de peticiones HTTP de una forma cómoda y fácil, este programa nos permite generar cualquier tipo de mensaje HTTP y enviar al instante con un simple clic. Ha sido usado principalmente para realizar las llamadas a las funciones de Azure y comprobar el correcto funcionamiento de estas de maneras local y aislada del resto de funcionalidad. [5]

Tampoco nos podemos olvidar de Teams, que es la plataforma con la que se integra el chatbot y a través de la cual interaccionamos con él. Es una aplicación de mensajería a través de la cual se pueden organizar reuniones y llamadas de voz creada por Microsoft. [6]

4. Proceso de desarrollo software utilizado

Para la realización de este proyecto, se ha utilizado una metodología ágil. Mientras que, en las metodologías tradicionales, como el desarrollo en cascada, se ejecuta todo el proceso de forma secuencial y al terminar si hay algún problema es necesario volver al principio y ejecutar de nuevo desde ahí, en las metodologías ágiles, el desarrollo es incremental y muy interactivo con el cliente. Esto es debido a que, en un intervalo corto de tiempo, normalmente una o dos semanas, acordado con los stakeholders, se tiene una reunión con los mismos, en la cual se revisa que se está avanzando por buen camino, además de verificar si es necesaria alguna modificación y se definen los siguientes pasos a realizar. En este tipo de metodologías, el cambio siempre es bien recibido.

Más concretamente la metodología usada ha sido Scrum, que está muy extendida en el mundo del software y es muy popular. Esta metodología está dividida de la siguiente manera. Se comienza con una reunión entre el representante 'product owner' y el equipo de trabajo 'scrum team'. En ella se define el trabajo que se va a desarrollar y se crea un documento llamado 'product backlog'. Este documento contiene todas las historias de usuario del proyecto. A continuación, se realiza una nueva reunión con los mismos integrantes, llamada 'Sprint planning meeting' en la cual, se define el trabajo a realizar en el primer sprint o iteración. Con las historias a realizar definidas, el 'scrum team' se reúne para definir las tareas a realizar. El equipo se pone a trabajar en el 'sprint' el cual suele durar entre 1 y 4 semanas, durante el cual todos los días tiene lugar el 'daily scrum meeting' en el cual se comenta como va cada uno y si tiene algún bloqueo o problema. Una vez terminado el 'sprint', se reúnen el 'product owner' y el 'scrum team' nuevamente en la 'product review', en la cual se comprueba que se ha desarrollado lo acordado y de la forma acordada, además de revisar que funciona correctamente por parte del 'product owner' (pruebas de aceptación). Como última tarea del sprint el equipo de trabajo se reúne en la 'sprint retrospective' para ver cómo mejorar su manera de trabajar y analizar que ha ido bien y que es mejorable durante el desarrollo. Después de esta reunión se comenzaría de nuevo por el principio, iterando así hasta terminar.

En la figura 1 podemos ver cómo es una iteración en este tipo de metodología.

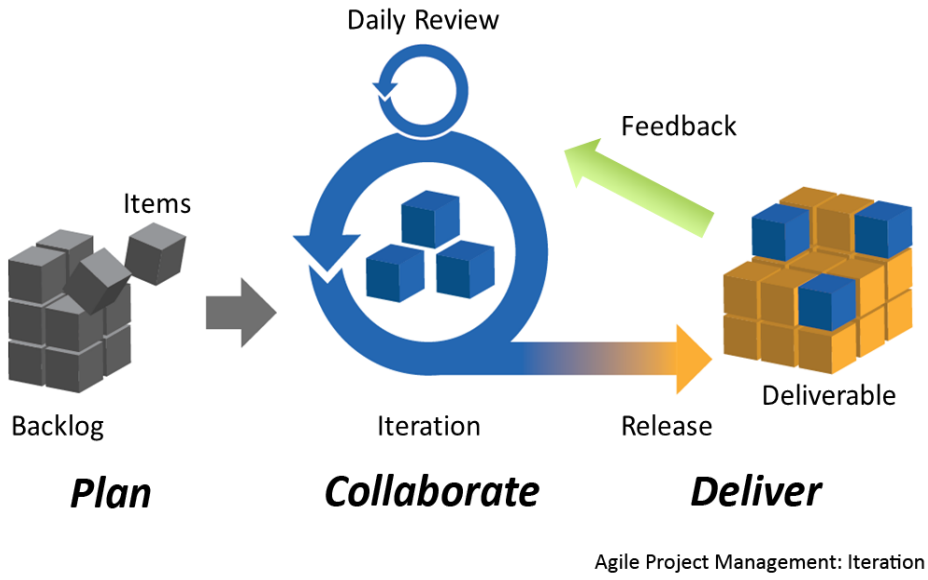

Figura 1 Metodología agile [11]

5. Desarrollo del proyecto

A continuación, en las figuras 2 y 3, podemos observar las diferentes fases en las que se ha dividido la programación del proyecto. Se ha realizado un proceso de formación dividido en dos partes, diferenciadas cada una por su tecnología. Posteriormente ha tenido lugar la especificación de requisitos del proyecto, en la cual se recogen todas las características y necesidades que tiene que cumplir. Terminada la especificación, comencé con las fases de diseño, tanto de la aplicación como del diseño arquitectónico. Finalizado el diseño, empecé a desarrollar la aplicación de LUIS y la aplicación del chatbot, las cuales ocuparon la mayor parte del tiempo del proyecto. Para acabar se realizaron y ejecutaron los diferentes tipos de pruebas, asegurándome así de que el desarrollo era correcto.

Modo de	Task Name	Duración
	Desarrollo del chatbot	**101 días**
	Formación chatbots	20 días
	Formación Azure	5 días
	Especificación de requisitos	1 día
	Diseño de la aplicación	3 días
	Diseño arquitectónico	4 días
	Desarrollo de la aplicación LUIS	20 días
	Desarrollo del chatbot	40 días
	Pruebas unitarias	4 días
	Pruebas de integración	3 días
	Pruebas de aceptación	1 día

Figura 2 Diagrama de Gantt Parte 1

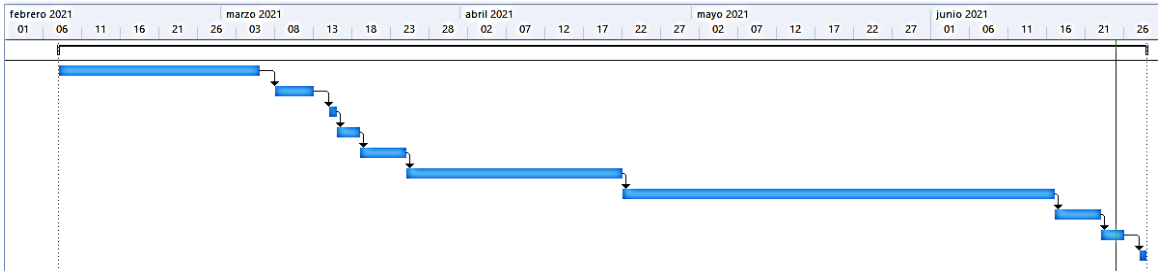

Figura 3 Diagrama de Gantt Parte 2

5.1. Proceso de formación

Al comenzar el proyecto se ha invertido una cantidad de tiempo importante en el análisis y formación en las tecnologías que se usarán durante el desarrollo del proyecto.
En primer lugar, comencé formándome en el entorno de los chatbots, realizando un curso en el que desarrollé un chatbot muy simple con la tecnología 'DialogFlow' de Google. Gracias a este curso conseguí entender mejor cómo funcionaban los chatbots, los problemas que podían tener y aspectos para tener en cuenta al desarrollar uno. Después comencé la formación sobre Azure incluyendo en esta tanto 'Azure functions' necesarias para poder realizar la integración y obtención de datos de la plataforma de

Dynamics, como en el 'Bot Service' de Azure ya que era el bot que iba a desarrollar finalmente.
Una vez entendí cómo funcionaban estas partes desarrollé inicialmente un bot sencillo para ir practicando y afianzar todos estos conceptos aprendidos.
Tras conocer todos estos aspectos sobre el bot, comencé a investigar sobre como dotar al bot de 'inteligencia', dotándole así la capacidad de entender el lenguaje natural y poder entender al usuario con el que interactúa. Para ello descubrí LUIS y me formé también en ello hasta que finalmente conseguí crear una aplicación con la cual el bot sería capaz de identificar y entender todo lo necesario.

5.2. Especificación de requisitos

La especificación de requisitos ha sido realizada en diferentes reuniones, en las cuales intervenían, el gerente del departamento que es el solicitante de la aplicación, con algún jefe de proyecto y programadores senior para realizar la captura de los diferentes requisitos a establecer en el proyecto de la mejor manera posible. Esta es una parte muy importante del proyecto, en la que es necesario entender y conocer bien lo que te piden para poder llevarlo a cabo correctamente.

5.2.1. Requisitos funcionales

El objetivo del chatbot es facilitar y agilizar el uso del CRM de CIC para todos sus usuarios. Para ello se han establecido los requisitos que se describen a continuación:

Identificador	Nombre	Descripción
RF01	Consultar las diferentes entidades	Es necesario que se puedan consultar los diferentes registros de las diferentes entidades en el CRM, como son las oportunidades.
RF02	Cambiar la fecha de una oportunidad	Las oportunidades tienen una fecha de comienzo la cual a menudo es necesario cambiar y para ello el bot nos lo facilita, ofreciéndonos una funcionalidad que dado el código de una oportunidad y la fecha a establecer nos realizará el cambio.
RF03	Consultar las últimas oportunidades	Dado que el CRM contiene muchas oportunidades y la gran mayoría serán antiguas y por tanto al usuario no le interesarán, tenemos una funcionalidad mediante la cual el bot nos mostrará las últimas oportunidades que se han creado, reduciendo así la información a analizar.
RF04	Ver más detalles de una oportunidad concreta	Mientras el usuario consulta información de las diferentes oportunidades, ya que no se muestra toda la información de la oportunidad puede requerir de información más detallada de una oportunidad concreta y para ello podemos preguntar al bot por una oportunidad concreta y nos mostrará más información sobre dicha oportunidad.
RF05	Filtrar las oportunidades por su estado	Las oportunidades tienen varios estados y puede resultar interesante filtrar por las oportunidades con un estado concreto, como puede ser el estado ofertando, por ejemplo. Para ello le solicitarás al bot las oportunidades con dicho estado y te las mostrará.

RF06	Dar la bienvenida al usuario	También es de esperar en un bot donde se pretende tener una experiencia de usuario satisfactoria y similar a la que se tendría en la interacción con un humano, si le saludas te devuelva el saludo y el bot no se puede quedar atrás en este aspecto. Ya que cuando le saludas te dará la bienvenida a CIC mostrándote el logo y el acceso al portal del empleado.
RF07	Consultar las oportunidades cercanas a caducar	Una de las funciones de los usuarios del CRM es actualizar la fecha de las oportunidades cuando estas cambien o están cercanas a expirar, para ello tenemos esta función que nos devuelve las que están caducadas o cercanas a caducar.
RF08	Informar mensualmente al usuario	Para que no se le olvide a nadie actualizar sus oportunidades y que haya algunas caducadas. El bot enviará un mensaje al mes a los usuarios con sus oportunidades caducadas o cercanas a caducar.

Tabla 1 Requisitos Funcionales

5.2.2. Requisitos no funcionales

Además, es necesario cumplir los siguientes requisitos no funcionales solicitados por la empresa:

Identificador	Nombre	Descripción
RNF01	Estar integrado con Teams	Ya que Teams es la plataforma de comunicación implantada en la empresa y a la cual todo empleado tiene acceso, es necesario que esté integrado en ella para que así todos los empleados puedan acceder al bot y usarle sin ningún tipo de problema ni instalación de software específico
RNF02	Seguridad a nivel usuario	Cada usuario, solo podrá acceder a los mismos datos a los cuales tendría acceso desde la plataforma, es decir, como si estuviese logueado en el CRM.
RNF03	Tiempo de respuesta adecuado	El tiempo de respuesta de la aplicación será lo más bajo posible, estando acorde con la acción a realizar.

Tabla 2 Requisitos no Funcionales

5.3. Casos de uso

En la siguiente imagen podemos observar el diagrama con los diferentes casos de uso del proyecto.

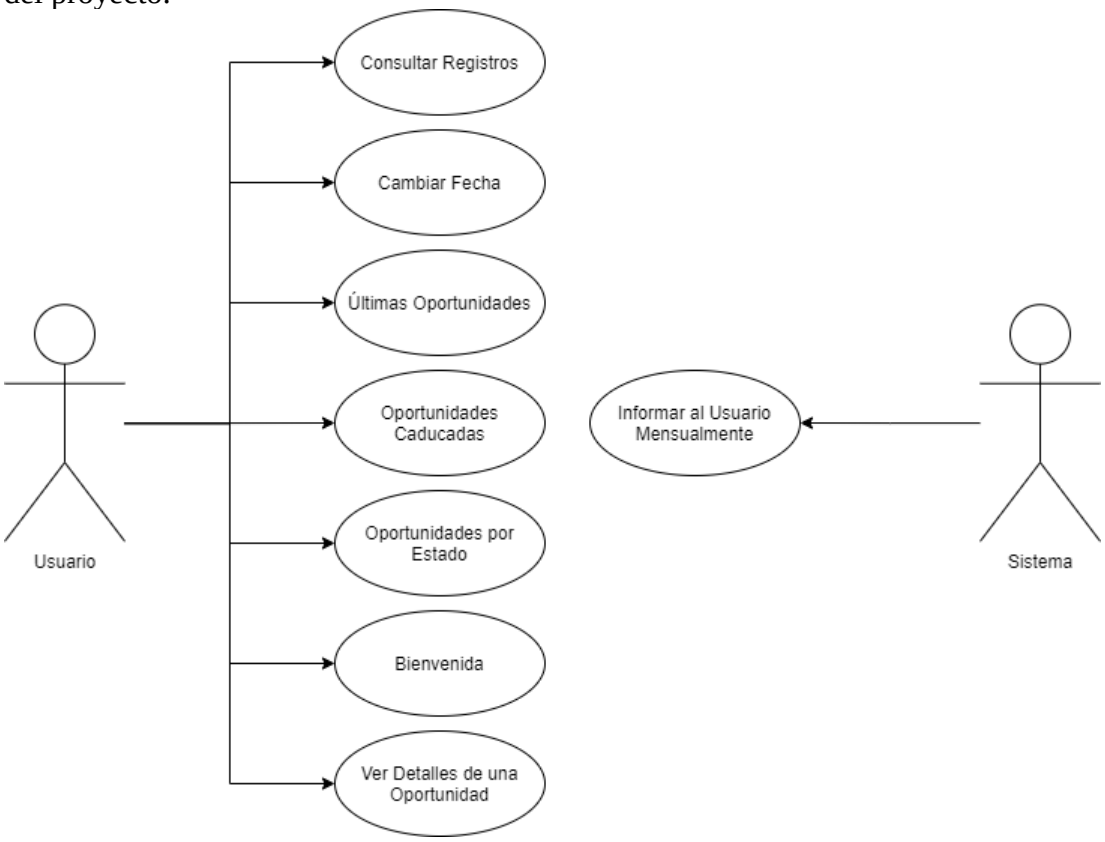

Figura 4 Diagrama de Casos de Uso

Caso de Uso	CU.1 Consultar Registros
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario podrá consultar la información de las diferentes entidades disponibles en el CRM.
Flujo básico	**1. Seleccionar la entidad a consultar** El usuario selecciona una entidad de la lista mostrada. **2. Confirmar consulta** El usuario confirma que quiere consultar los registros de la entidad seleccionada. **3. Mostrar datos** El sistema le muestra al usuario los datos de los registros deseados.
Flujos alternos	**1. El usuario selecciona una entidad no disponible** El sistema le muestra de nuevo las opciones disponibles al usuario, para que seleccione una de ellas. **2. El usuario cancela la consulta** El usuario decide que no quiere realizar la consulta y, por tanto, no confirma la consulta. Como consecuencia, el sistema no le muestra los datos de los registros.

Pre-condiciones	**1. Estar dentro del dialogo de consultar registros** El usuario debe estar en el dialogo de consultar registros. Para ello desde el dialogo principal puede decir consultar registros o pedir ayuda y seleccionar esta opción.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 3 Caso de Uso: Consultar Registros

Caso de Uso	CU.2 Cambiar Fecha
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario podrá cambiar la fecha prevista de inicio de una oportunidad concreta.
Flujo básico	**1. Seleccionar la oportunidad a cambiar** El usuario manda un mensaje con el código de la oportunidad que desea cambiar. **2. Envía la nueva fecha** El usuario envía la nueva fecha a establecer. **3. Confirmación de completado** El sistema le muestra al usuario un mensaje con los datos de la oportunidad que han sido actualizados.
Flujos alternos	**1. El usuario selecciona una oportunidad no disponible** El sistema le notifica de que el código de la oportunidad no es correcto. **2. El usuario introduce un valor que no corresponde con una fecha** El sistema notifica al usuario que el valor introducido no es una fecha. **3. El usuario no tiene permisos.** El usuario no tiene permisos para la oportunidad seleccionada, por tanto, el sistema le advierte de que no puede realizar la acción.
Pre-condiciones	**1. Estar dentro del dialogo de cambiar fecha** El usuario debe estar en el dialogo de cambiar fecha. Para ello desde el dialogo principal puede decir cambiar fecha o pedir ayuda y seleccionar esta opción.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 4 Caso de Uso: Cambiar Fecha

Caso de Uso	CU.3 Últimas Oportunidades
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario podrá consultar cuales son las últimas oportunidades creadas.
Flujo básico	**1. Seleccionar la opción últimas oportunidades** El usuario selecciona la opción últimas oportunidades, de alguna de las maneras posibles, desde el menú de ayuda o diciendo algo similar a "últimas oportunidades". **2. Enviar los datos al usuario** El sistema muestra las últimas oportunidades creadas al usuario. **3. Seleccionar una de las oportunidades** El usuario selecciona una de las oportunidades mostradas. **4. El sistema le muestra el detalle de la oportunidad.** El sistema le muestra al usuario información más detallada sobre la oportunidad seleccionada.
Flujos alternos	**1. El usuario no selecciona ninguna oportunidad** El sistema le devuelve al dialogo principal.
Pre-condiciones	**1. Estar dentro del dialogo principal** El usuario debe estar en el dialogo principal del bot.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 5 Caso de Uso: Últimas Oportunidades

Caso de Uso	CU.4 Oportunidades caducadas
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario podrá ver que oportunidades tiene caducadas o cercanas a caducar.
Flujo básico	**1. Seleccionar la opción oportunidades caducadas** El usuario selecciona la opción oportunidades caducadas por alguno de los métodos disponibles. **2. Muestra los datos de las oportunidades caducadas** El sistema muestra las oportunidades caducadas al usuario.
Flujos alternos	**1. El usuario no tiene permisos** El sistema le muestra un mensaje advirtiéndole de que no tiene permisos para realizar dicha acción.
Pre-condiciones	**1. Estar dentro del dialogo principal** El usuario debe estar en el dialogo principal del bot.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 6 Caso de Uso: Oportunidades Caducadas

Caso de Uso	CU.5 Oportunidades por Estado
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario podrá consultar las oportunidades que tengan un estado concreto.
Flujo básico	**1. Solicitar las oportunidades con un estado concreto** El usuario informa al sistema del tipo de oportunidades que quiere consultar. **2. Muestra oportunidades concretas** El sistema muestra las oportunidades con el estado acordado por el usuario.
Flujos alternos	**1. El usuario no tiene permisos** El sistema le muestra un mensaje advirtiéndole de que no tiene permisos para realizar dicha acción. **2. El usuario introduce un estado que no existe** El sistema le muestra las oportunidades con el estado más similar al introducido por el usuario.
Pre-condiciones	**1. Estar dentro del dialogo principal** El usuario debe estar en el dialogo principal del bot.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 7 Caso de Uso: Oportunidades por Estado

Caso de Uso	CU.6 Bienvenida
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario podrá saludar al bot.
Flujo básico	**1. Saludar** El usuario saluda al bot. **2. Bienvenida** El sistema da la bienvenida al usuario.
Flujos alternos	
Pre-condiciones	**1. Estar dentro del dialogo principal** El usuario debe estar en el dialogo principal del bot.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 8 Caso de Uso: Bienvenida

Caso de Uso	CU.7 Ver Detalles de una Oportunidad
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario podrá ver los detalles de una oportunidad concreta.
Flujo básico	**1. Seleccionar la oportunidad a consultar** El usuario selecciona una oportunidad mediante su código de oportunidad. **2. Muestra datos** El sistema muestra al usuario los datos más detallados de la oportunidad solicitada.
Flujos alternos	**1. El usuario no tiene permisos** El sistema le muestra un mensaje advirtiéndole de que no tiene permisos para realizar dicha acción.
Pre-condiciones	**1. Estar dentro del dialogo principal** El usuario debe estar en el dialogo principal del bot.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 9 Caso de Uso: Ver Detalles de una Oportunidad

Caso de Uso	CU.8 Informar al Usuario Mensualmente
Fuentes	Stakeholder principal
Actor	Usuario
Descripción	El usuario será notificado mensualmente de las oportunidades caducadas.
Flujo básico	**1. Muestra oportunidades caducadas** El sistema muestra al usuario de manera periódica las oportunidades caducadas.
Flujos alternos	**No tiene flujos alternativos**
Pre-condiciones	**1. Haber iniciado una conversación con el bot** Para que el bot te envíe las oportunidades que tienes caducadas mensualmente, será necesario que hayas establecido una conversación con él.
Post-condiciones	**1. El usuario vuelve al dialogo principal** Al terminar el caso de uso por cualquiera de los flujos, el usuario es devuelto al dialogo principal para poder seguir interactuando con el bot.

Tabla 10 Caso de Uso: Informar al Usuario Mensualmente

5.4. Arquitectura del chatbot

En la figura 5 podemos observar la arquitectura genérica de un chatbot en Azure. En dicha arquitectura el usuario realiza una petición al servicio del bot, el cual autentica y autoriza al usuario gracias a Azure y le devuelve un token o llave. Posteriormente se realiza la 'query' o consulta, siendo procesada por LUIS. Tras obtener los resultados de dicha consulta, se devuelve al servicio del bot, el cual almacena los logs correspondientes y el 'feedback' necesario y finaliza devolviéndole la respuesta al usuario.[8]

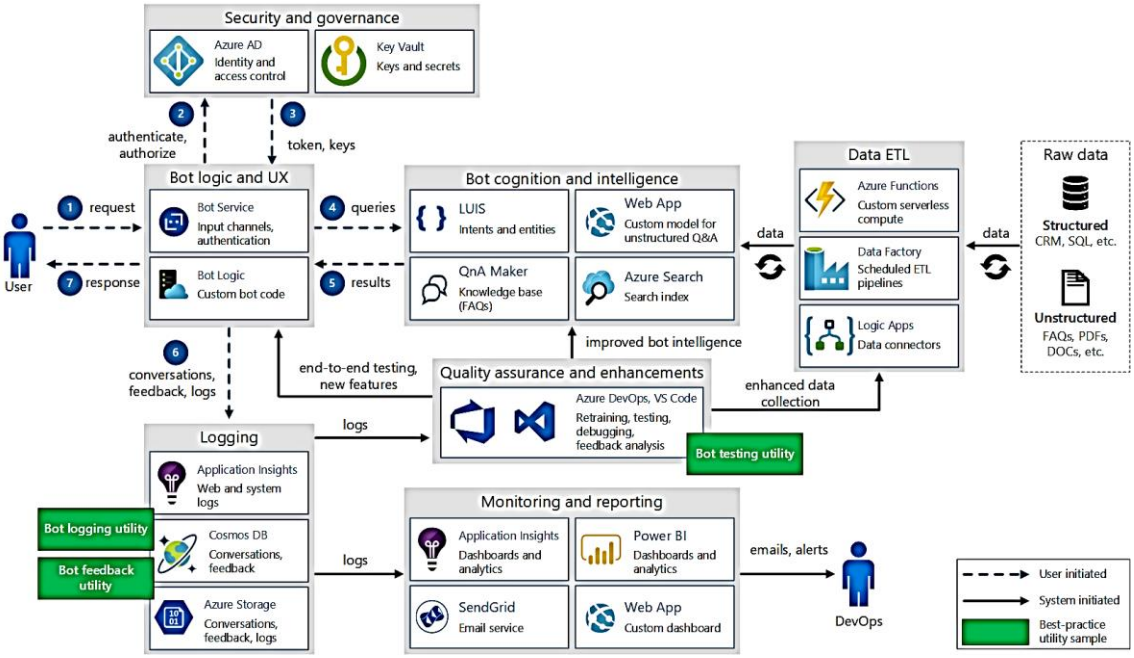

Figura 5 Arquitectura bot de Azure

5.5. Comunicación del proyecto

A continuación, en la figura 6, podemos ver el diagrama de secuencia de alto nivel, genérico y simplificado de la comunicación de la mayoría de las acciones que se realizan sobre el bot. En el diagrama podemos observar que el usuario envía un primer mensaje desde la aplicación de Teams. Esta se lo manda al chatbot, el cual lo procesa ayudándose de LUIS y llama a la función de Azure correspondiente al caso concreto que sea necesario ejecutar. La función de Azure procesa los datos y realiza la consulta que sea necesaria al CRM de Dynamics. El CRM devuelve los datos o error si no se ha podido acceder a ellos. Finalmente, la función devuelve un código HTTP de error o de que todo ha ido bien al chatbot, que procesa ese código enviándole vía Teams al usuario un mensaje informándole del error o con los datos solicitados.

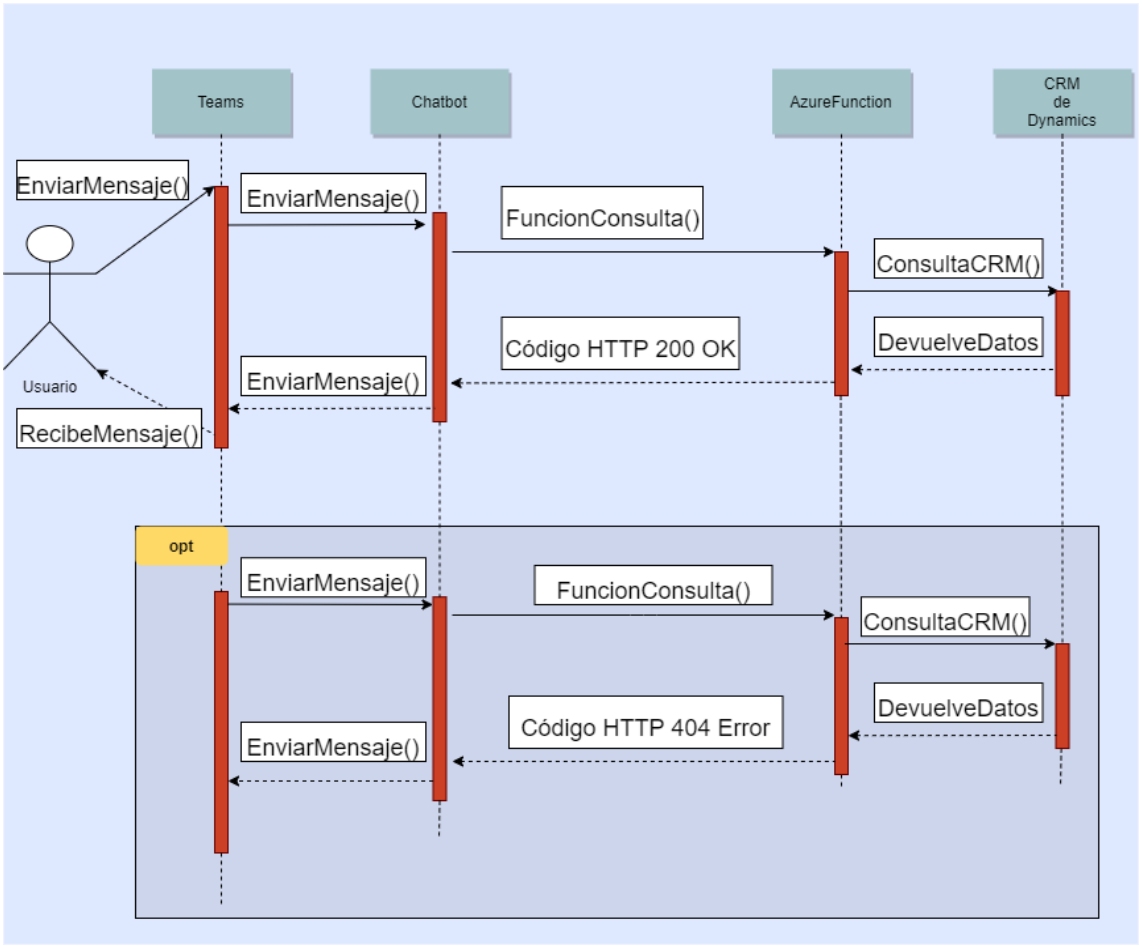

Figura 6 Diagrama de Secuencia

5.6. Diseño, desarrollo e implementación

Durante el desarrollo podemos destacar tres módulos principales. En primer lugar, el propio bot con su código en .Net, por otro lado, tenemos las funciones de Azure que invocan diferentes consultas sobre el CRM y por último la aplicación de LUIS para el reconocimiento y procesamiento del lenguaje natural. Es importante diferenciarlas porque cada una tiene un propósito y se realiza de una forma distinta. Primero de todo la parte principal del bot se desarrolla en la plataforma de .Net con código C#, pero debido a que la forma de realizar consultas al CRM no es compatible con esta versión, es necesario ejecutarlo a través de funciones de Azure las cuales si son compatibles con esta forma de hacer consultas. También se usan estas funciones para enviar un mensaje al usuario de forma periódica con las oportunidades caducadas en el CRM. Y por último la aplicación de LUIS la cual es necesaria para entender el lenguaje del usuario y que el bot no sea estático ni funcione por comandos establecidos, lo cual resultaría rígido y poco "humano", empeorando así la experiencia del usuario.

La arquitectura de la aplicación del bot sigue el modelo arquitectónico modelo vista controlador (MVC). Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

En la figura 7 podemos ver un ejemplo gráfico de esta arquitectura. [7]

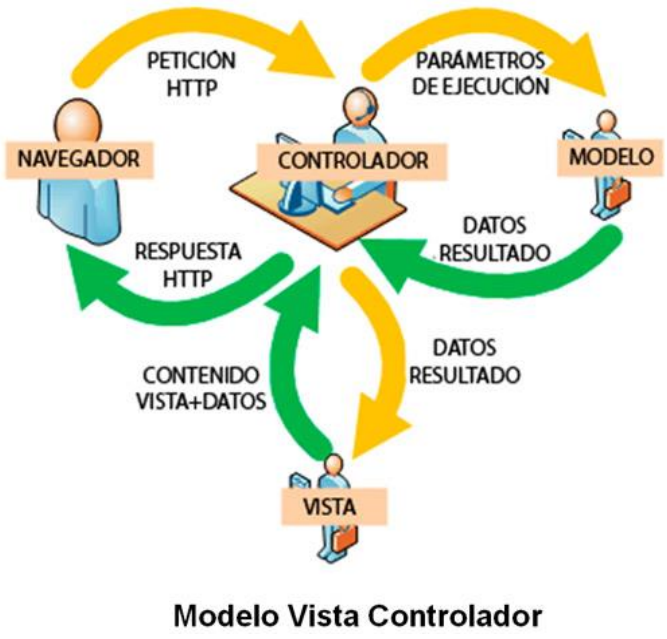

Figura 7 Modelo arquitectónico **[12]**

La aplicación de LUIS está formada por ‘intents’ que son las intenciones que reconocerá la aplicación. Estas intenciones son lo que pretende decir el usuario como, por ejemplo, saludar al bot. Cada ‘intent’ está compuesto por varias oraciones que se asocian a dicho ‘intent’. Una vez se han establecido las oraciones que formarán parte de un ‘intent’, se entrena la aplicación, para que reconozca oraciones similares a las asignadas y de esta forma entienda la intención del usuario, aunque no diga ninguna de las oraciones asignadas a dicho ‘intent’. Además de los ‘intents’, tenemos las entidades, por ejemplo, tenemos la entidad código de oportunidad, la cual nos sirve para reconocer cuando el usuario introduce un código de oportunidad en su oración. Gracias a que este código sigue una expresión ‘regex’, basta con asignar dicha expresión ‘regex’ en la entidad de LUIS y este ya lo reconocerá. La forma de integrar esta aplicación con nuestro bot consiste en importarla desde el navegador a un ‘json’ y después convertir ese ‘json’ a una clase de C# la cual añadimos a nuestro proyecto del bot.

En la figura 8 podemos ver algunos de los diferentes ‘intents’ que hay en la aplicación de LUIS.

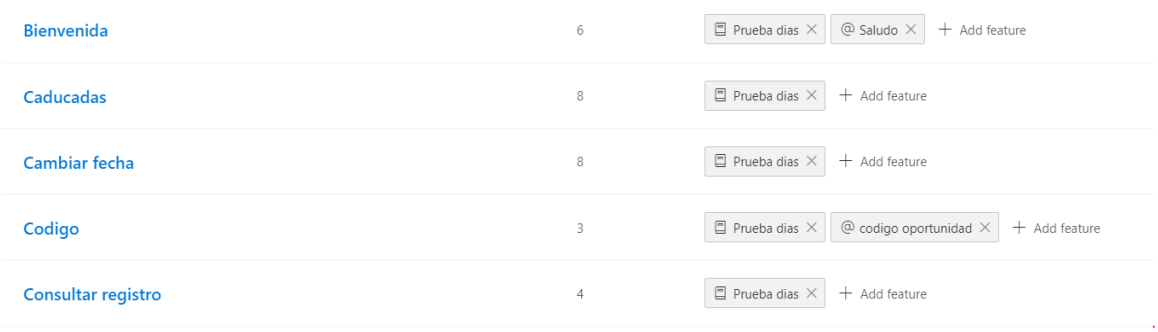

Figura 8 'Intents'

En la figura 9, se observan algunas de las frases de ejemplo a partir de las cuales el bot aprende a reconocer el ‘intent’ de oportunidades caducadas.

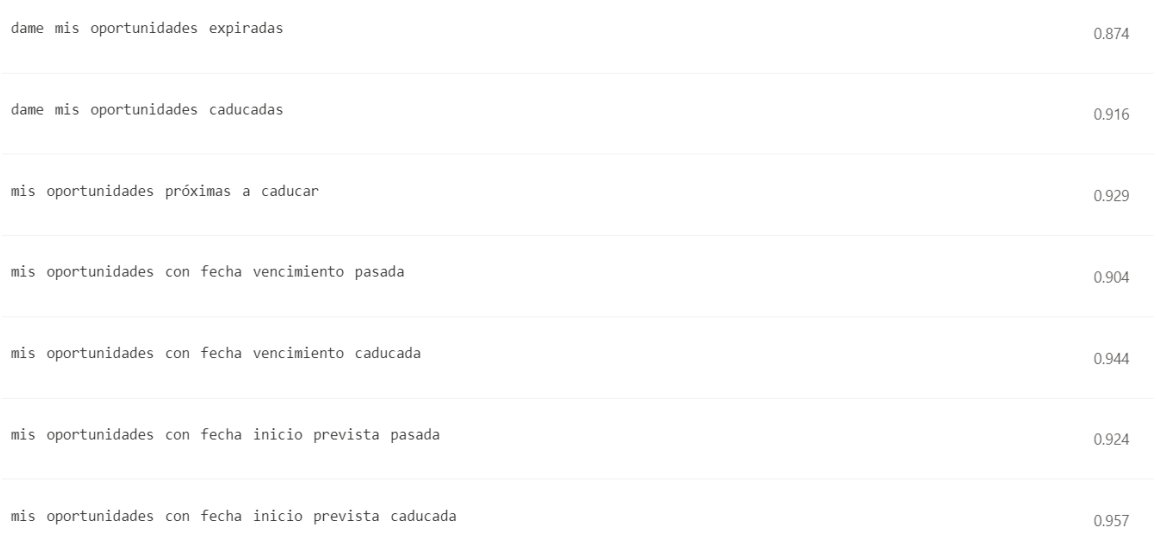

Figura 9 Frases de ejemplo

Seguidamente, en la figura 10, podemos observar el diagrama de clases de nuestro chatbot. El chatbot está formado por un dialogo principal que es 'MainDialog', el cual contiene otros diálogos a los que puede invocar. El dialogo principal, consta de un 'Recognizer' que se encarga de reconocer los diferentes 'intents' de la aplicación de LUIS y lanzarlos, gracias a la clase 'OportunidadesRecognizer' y más concretamente al método RecognizeAsync<T>(), donde T es del tipo Oportunidades, que es la clase generada por la aplicación de LUIS.
Por otra parte, tenemos los controladores, que heredan del controlador base de Microsoft ASP Net Core. Y por último tenemos las clases 'DialogBot' y 'DialogAndWelcomeBot' que se lanzan cuando llega el primer mensaje o cuando el usuario manda un mensaje, para guardar el estado de la conversación o cuando se quiere iniciar un nuevo dialogo.

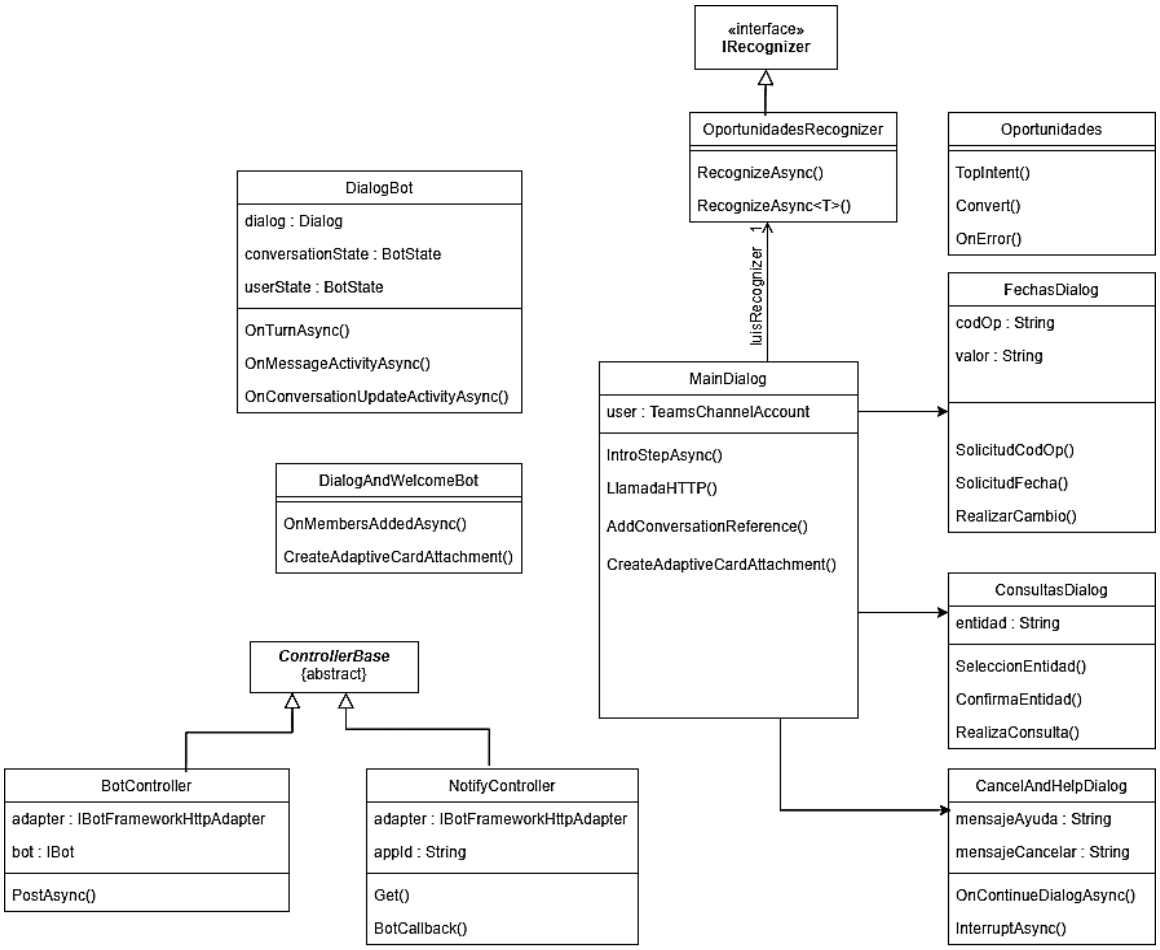

Figura 10 Diagrama de Clases

A continuación en la figura 11 podemos ver la estructura interna del proyecto del chatbot. En el apartado de 'Bots' tenemos las clases que se encargan de gestionar el envio de mensajes y reconocimiento de entrada de usuarios a la conversación. En el caso del 'DialogAndWelcomeBot', se encarga de gestionar la bienvenida del bot y el otro es el que gestiona el resto de mensajes.
Por otra parte tenemos las tarjetas 'Cards', que son los elementos que manda el bot en forma de tarjeta con imágenes texto botones y demás elementos. En este caso tenemos la tarjeta de bienvenida ya que es la única fija, porque las demás tarjetas que se usan son dinámicas en función de los datos que se consultan.

En el apartado de ‘CognitiveModels’, encontramos las clases de LUIS, ‘Oportunidades.cs’ que es la clase en C# y el .json es el modelo de datos que usa LUIS para reconocer e identificar las diferentes entradas de texto que envia el usuario.
Además tenemos un apartado de ‘Controllers’, que como su propio nombre indican se encargan de controlar el funcionamiento del bot, en este caso se encarga de controlar las solicitudes HTTP, en el caso del ‘NotifyController’ es la clase que se encarga de notificar al usuario cada cierto tiempo con el mensaje que contiene información sobre las funciones que puede realizar el chatbot y las oportunidades caducadas. Ambos controllers son servicios web.
A continuación en la carpeta ‘Dialogs’, tenemos los diferentes dialogos del bot, el ‘MainDialog’ que es el dialogo principal y en el que se encuentra el usuario inicialmente, despues tenemos el ‘CancelAndHelpDialog’ el cual como su propio nombre indica, se encarga de gestionar el dialogo cuando el usuario pide ayuda o desea cancelar un dialogo. El ‘ConsultasDialog’ y el ‘FechasDialog’, se encargan de dirigir el dialogo de las consultas de las distintas entidades y de cambiar la fecha de inicio de una oportunidad concreta respectivamente.
Añadido a todo esto tenemos la clase ‘AdapterWithErrorHandler’, la cual se encarga de gestionar los casos de error, mostrando un mensaje al usuario advirtiendole del problema.

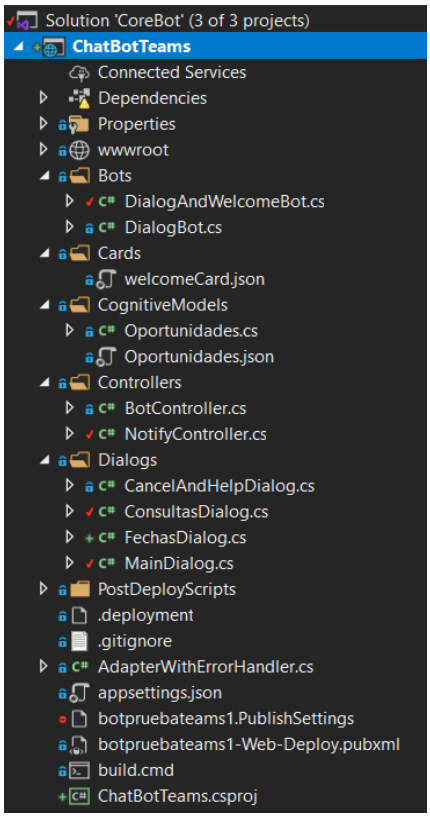

Figura 11 Estructura proyecto del bot 1

En la figura 12 tenemos la clase encargada de reconocer los diferentes 'intents' existentes en el modelo de LUIS, que es 'OportunidadesRecognizer'. También tenemos el 'Program' que es el 'main' que se encarga de iniciar todo el bot y el 'Startup' que es la clase encargada de iniciar y preparar todo para el arranque correcto.
Por último, podemos ver los dos diferentes proyectos de pruebas tanto unitarias como de integración que se han realizado para comprobar el correcto funcionamiento del bot.

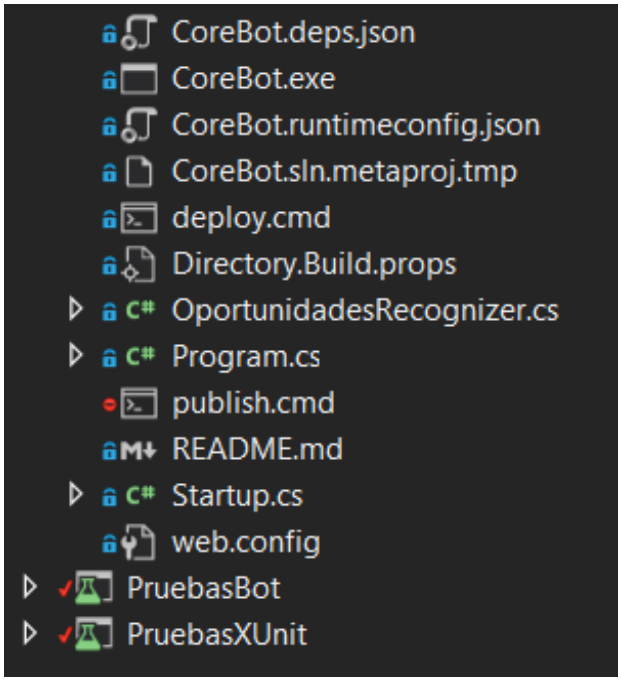

Figura 12 Estructura proyecto del bot 2

5.7. Pruebas

Para el desarrollo del bot se ha seguido un minucioso proceso de pruebas, para asegurarse de que todo funcionaba correctamente. La parte más importante de este proceso ha sido la interacción directa con el bot, para ver como respondía, comprobar si interpretaba bien el lenguaje natural y comprobar que todas las integraciones funcionasen.
A parte de este proceso se han seguido también los procesos habituales de pruebas automatizadas del software, entre los cuales se incluyen las pruebas unitarias, de integración y del sistema. Sin olvidarnos también de las pruebas de aceptación una vez terminado el producto. Estas últimas se detallan a continuación.

5.7.1. Pruebas unitarias

Se han desarrollado todas las pruebas unitarias necesarias y entre ellas, la prueba unitaria del método por el cual solicitas ayuda o cancelas un dialogo activo con el bot. De manera que al iniciar un dialogo con el bot, por ejemplo, diciéndole que quieres consultar los registros. En cualquier momento le puedes escribir ciertas palabras, con las cuales podrás cancelar el dialogo y volver al inicio de la conversación o pedirle ayuda sobre como interactuar con él si estás perdido.
Para el desarrollo de estas pruebas he utilizado la librería de XUnit, la cual nos permite hacer pruebas con varios datos a la vez de una manera muy sencilla. Simplemente tenemos que anotar el método como 'Theory' y poner después tanto 'InlineData' como conjuntos decidamos probar. En mi caso he probado todas las posibles combinaciones ya que no eran muchas y así nos aseguramos de que funciona con todas correctamente.
También se ha probado el caso en el que no le escribamos ninguna de las palabras que espera y que de esta forma no cancela el dialogo ni nos muestra la ayuda.

5.7.2. Pruebas de integración

Para el desarrollo de las pruebas de integración, he realizado todas las pruebas necesarias para asegurar el correcto funcionamiento de todos los elementos con los que interactúa el bot. Una de estas pruebas es la realizada sobre el diálogo de las fechas, que se comunica con la función de Azure, ya que es la parte del bot encargada de realizar las consultas. Siendo las funciones de Azure necesarias para interactuar con el CRM y obtener los datos de este. De esta forma se prueba la integración entre el bot, la aplicación de Teams, la función de Azure y el CRM, que son los elementos principales de este chatbot.
La función de la que hablamos es la encargada de cambiar la fecha de una oportunidad. En esta función se han probado varios casos de prueba, como son el caso en el que el usuario no tenga permisos para acceder al CRM o a la oportunidad concreta que se desea modificar. Comprobación que los datos introducidos por el usuario son correctos, es decir, que la oportunidad existe y que la fecha es realmente una fecha. Por último, también se ha probado el caso de éxito en el cual se modifica la fecha de la oportunidad concreta.
Para estas pruebas se ha utilizado la librería estándar de Visual Studio para el desarrollo de pruebas.

5.7.3. Pruebas del sistema

A continuación, se detallan las pruebas del sistema realizadas como comprobación de que todos los requisitos de la aplicación funcionan correctamente. Cabe destacar que los diferentes casos se pueden invocar con diferentes expresiones similares, gracias al uso de la inteligencia artificial.

PS01: Comprobar saludo

1. El usuario saluda al bot
2. El sistema devuelve un mensaje, dándole la bienvenida al usuario

PS02: Comprobar consulta de registros

1. El usuario solicita consultar información de los registros.
2. El sistema devuelve un mensaje, mostrándole las entidades disponibles a consultar al usuario.
3. El usuario selecciona una de las entidades a consultar.
4. El sistema le pregunta si desea consultar los registros.
5. El usuario confirma al sistema que quiere los registros.
6. El sistema devuelve los registros deseados al usuario.
   Como casos opcionales: el cliente puede solicitar ayuda sobre el dialogo o cancelarlo. También puede elegir en el paso 5 que no quiere consultarlos y cancelar así la consulta.

PS03: Cambiar fecha

1. El usuario solicita cambiar la fecha de una oportunidad.
2. El sistema devuelve un mensaje, pidiéndole que introduzca el código de la oportunidad que desea cambiar.
3. El usuario comunica al sistema el código de la oportunidad a cambiar.
4. El sistema le pregunta al usuario por el valor de la fecha que desea establecer.
5. El usuario comunica al sistema la fecha a establecer.
6. El sistema devuelve un mensaje informando de que se ha realizado correctamente el cambio.
   Como casos opcionales: el usuario introduce mal la fecha o el código de la oportunidad, por tanto, el sistema le informaría de que dato es erróneo. También se puede dar el caso de que el usuario no tenga permisos para realizar esta acción de lo cual también le informará el sistema

PS04: Últimas oportunidades

1. El usuario solicita las últimas oportunidades.
2. El sistema devuelve un mensaje con un botón por cada oportunidad.
3. El usuario selecciona una de las oportunidades mostradas por el sistema.
4. El sistema muestra al usuario la información detallada de la oportunidad seleccionada y la opción de cambiar fecha.
   Como casos opcionales: el usuario puede desplazarse por las últimas oportunidades para ver más de cinco que son las que se muestran inicialmente.

PS05: Oportunidades caducadas

1. El usuario solicita las oportunidades caducadas o cercanas a ello.
2. El sistema devuelve un mensaje con todas las oportunidades caducadas o cercanas a caducar.

PS06: Oportunidades por estado

1. El usuario solicita las oportunidades con un determinado estado, por ejemplo, las oportunidades abiertas.
2. El sistema devuelve un mensaje con todas las oportunidades que se encuentren en el estado solicitado, en este ejemplo, las que estén en estado 'abierta'.
   Como casos opcionales: el usuario puede introducir un estado que no exista y el bot le devolverá las oportunidades con el estado más similar al solicitado por el usuario.

PS07: Ver detalle de una oportunidad

1. El usuario solicita ver el detalle de una oportunidad.
2. El sistema devuelve toda la información detallada de la oportunidad concreta al usuario.

PS08: Mensaje proactivo

1. El usuario comienza una conversación con el bot.
2. El sistema le envía un mensaje de manera periódica con las oportunidades caducadas e información de las funciones en las que le puede ayudar sin necesidad de decirle nada.

5.7.4. Pruebas de aceptación

En cuanto a las pruebas de aceptación, se le ha proporcionado el chatbot al cliente el cual ha podido probar todo lo que ha estimado oportuno, asegurándose de que el software cumple con todo lo esperado y sin ningún error.

6. Uso de la aplicación e interfaz gráfica

En este apartado, se mostrarán imágenes de la interfaz gráfica y el funcionamiento del bot.

En la figura 13 podemos ver la bienvenida proporcionada por el bot, la cual nos muestra un mensaje de bienvenida, nos muestra una foto de CIC y nos muestra un botón con un enlace a través del cual podemos acceder a al portal del empleado de CIC, algo muy útil para los trabajadores.

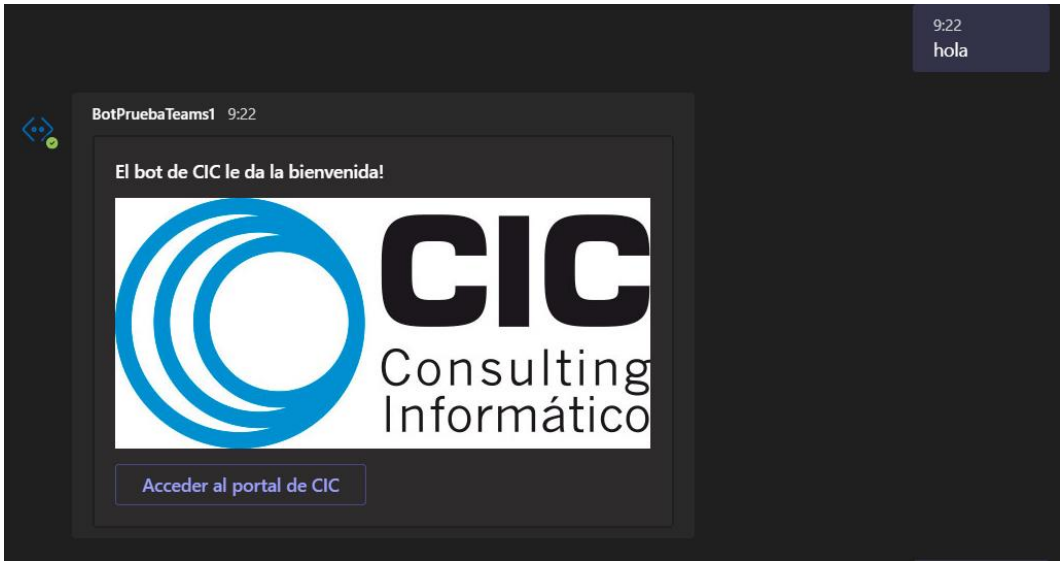

Figura 13 Interfaz bienvenida

A continuación, en la figura 14, tenemos el comando de ayuda general del bot y que también se activa cuando no reconoce lo que le hemos escrito, para que se puedan ver las posibilidades que tiene y como te puede ayudar. Nos muestra varios botones, en los cuales si pulsamos podremos realizar las funciones correspondientes a cada uno.

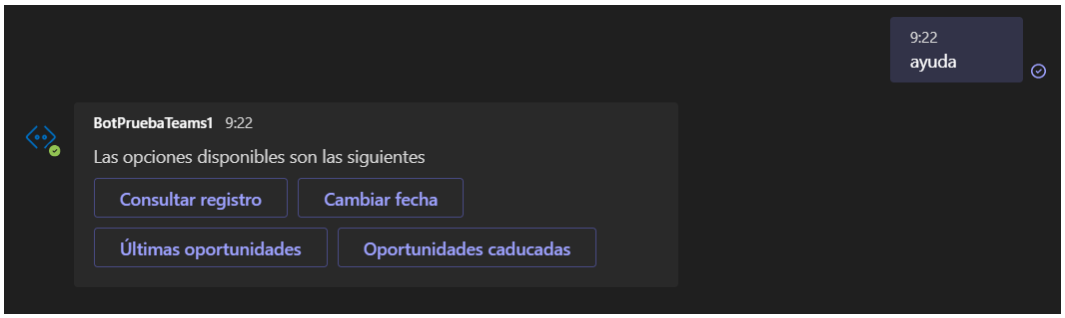

Figura 14 Interfaz de ayuda

En la figura 15 podemos ver la ejecución de una de las acciones que se muestran el menú de ayuda, en la cual se consulta con el CRM cuales son las últimas oportunidades creadas por el usuario en forma de botón, de manera que se seleccionamos una de ellas, podemos obtener el detalle esta. Dispone de flechas para moverte por las distintas oportunidades en caso de que quieras ver más de las que te aparecen inicialmente.

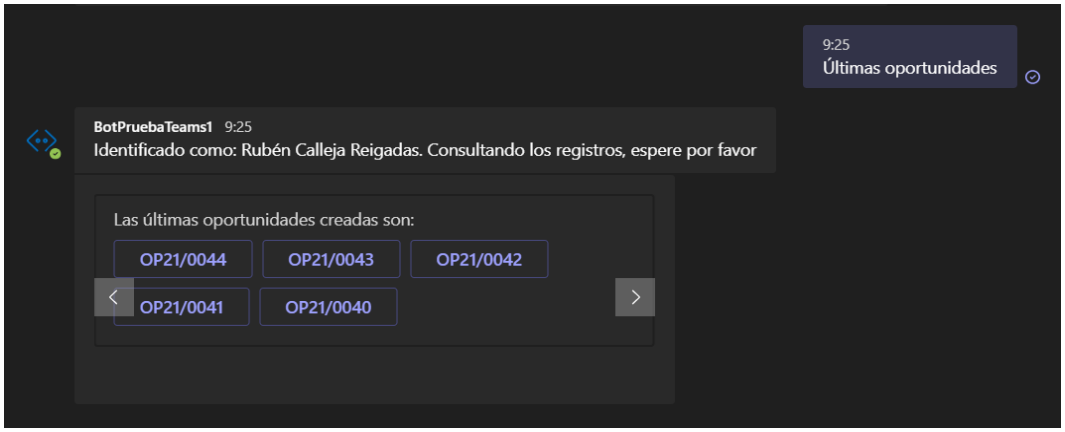

Figura 15 Interfaz de últimas oportunidades

En la figura 16, podemos ver como se muestran los datos de las oportunidades que se encuentran caducadas o cercanas a caducar. Se muestran en un formato de tabla para que al usuario le sea más fácil leer los datos y trabajar con ellos.

CodOp	Nombre	Valorado (€)	Importe (€)	Importe última oferta (€)
OP21/0043	P21/0042?	30.000,00	30.000,00	
OP20/0333	TEST1159		0,00	
OP21/0013	Test Miguel	6.000,00	6.000,00	6.000,00
OP21/0047	[ZARA] - PowerApps + Office 365 2022	10.450,00	11.000,00	
OP20/0326	test	55.477,60	69.347,00	
OP21/0046	test borrar		0,00	
OP20/0289	PRUEBA	53,82	234,00	234,00
OP20/0293	Hola	7,12	8,00	8,00
OP20/0287	IDbox - CEPSA - Mantenimiento de Licencia 2021	46.926,20	49.396,00	
OP20/0294	hola!!!	7.636,95	254.565,00	232.232,00

Figura 16 Oportunidades Caducadas

Como podemos observar en la figura 17, el bot nos va guiando de manera sencilla e intuitiva por los diálogos más complejos o largos. Nos muestra las opciones que tenemos disponibles y solo tenemos que seleccionar uno de los botones para ejecutar esa acción o también puedes escribir la opción deseada si lo prefieres.

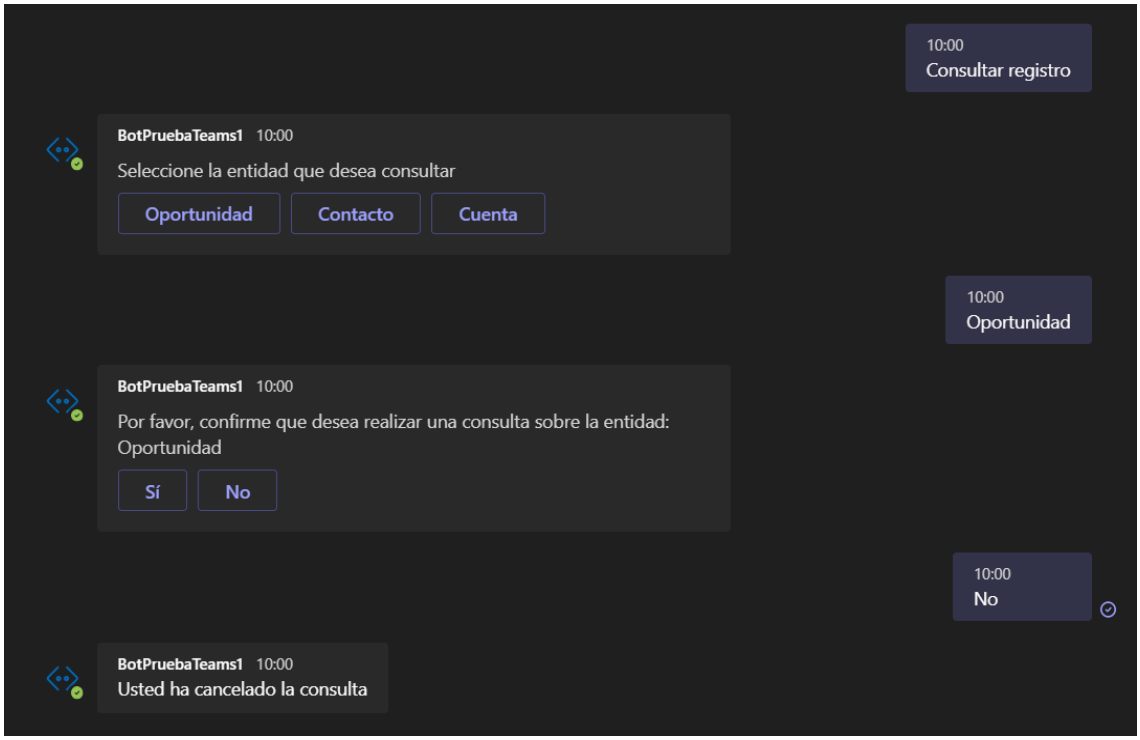

Figura 17 Guía por los diálogos

En la figura 18, podemos ver como si aun con estas facilidades nos encontramos perdidos, podemos solicitar ayuda al bot y nos dirá que podemos hacer en cada momento. Ya sea para seguir con el dialogo en el que nos encontramos o bien para cancelarle y volver al dialogo principal.

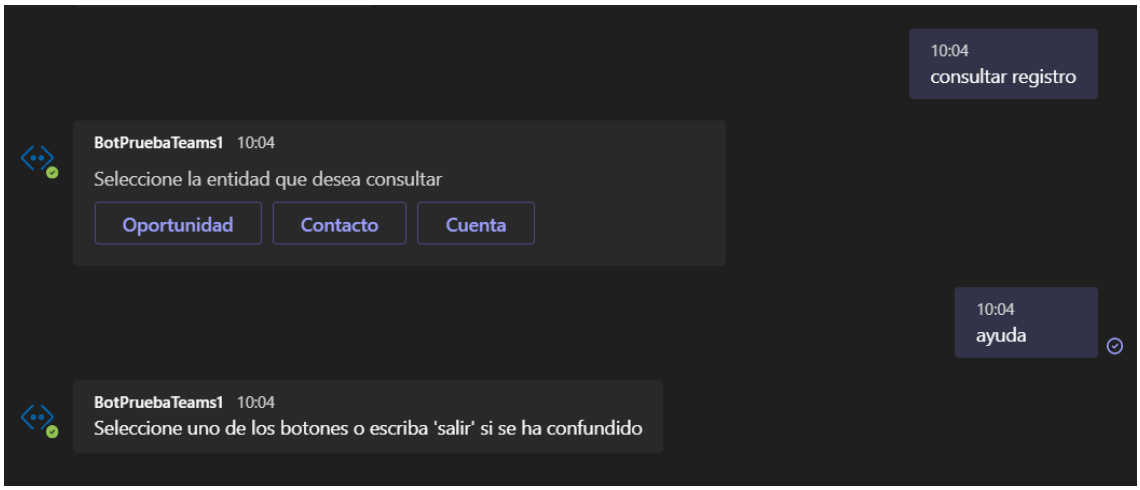

Figura 18 Ayuda en un diálogo

Siguiendo con lo anterior, en la figura 19, podemos ver cómo podemos cancelar cualquier dialogo en el que nos encontremos con palabras como salir o cancelar. En este caso, el bot nos informa de que nos cancela el dialogo y nos devuelve al dialogo principal.

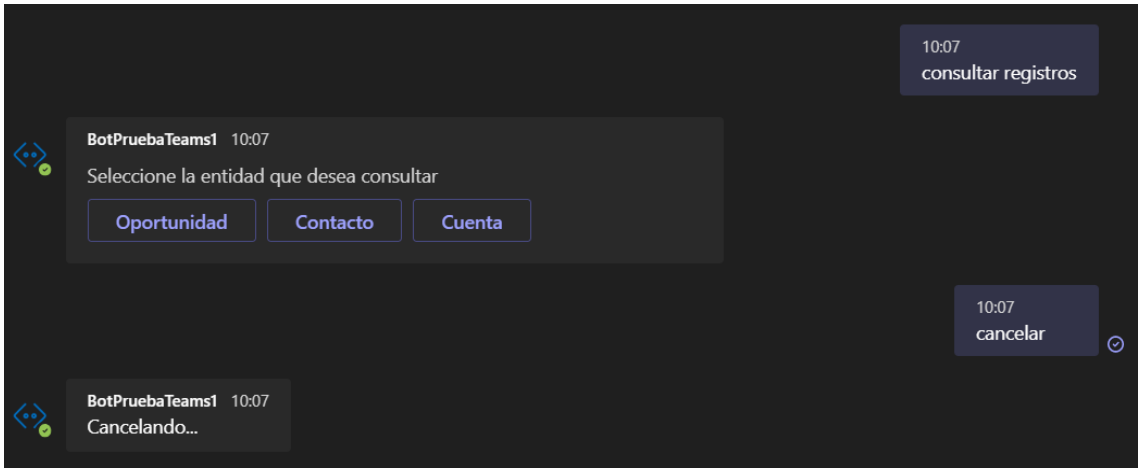

Figura 19 Cancelar un diálogo

7. Conclusiones y Trabajo Futuro

Durante este proyecto, he desarrollado un chatbot integrado con Teams y Dynamics para facilitar el uso del CRM a los empleados de CIC. En este chatbot se han implementado funcionalidades como dar la bienvenida al usuario, consultar registros del CRM, obtener las oportunidades caducadas o cercanas a caducar, cambiar la fecha de una oportunidad concreta, consultar los detalles de una oportunidad, notificar al usuario mensualmente cuales son las oportunidades que tiene caducadas, obtener cuales son las últimas oportunidades que se han creado y consultar las oportunidades con un determinado estado, por ejemplo, las oportunidades abiertas.

Durante el desarrollo me he enfrentado a diferentes retos, el primero de todos fue conseguir realizar consultas sobre el CRM desde el código del bot. El problema era la incompatibilidad entre donde estaba escrito el código del bot que era .Net Core y donde se realizaban las consultas que es .Net Framework. Para poder solventarlo cree una función de Azure a la cual se llamaba desde el bot siendo esta la encargada de realizar todas las consultas necesarias al CRM y procesar los datos, devolviéndoles posteriormente al bot.

Otro de los problemas encontrados fue obtener la información del usuario que está interactuando con el bot y poder realizar una consulta al CRM con sus credenciales. Para ello obtenemos el correo con el que está registrado en Teams, el cual coincide con el correo del usuario en el CRM. Y con el id del usuario en CRM realizamos un ‘impersonate’ es decir, realizar una consulta como si fuésemos ese usuario y así obtenemos solo las oportunidades a las que tiene acceso dicho usuario.

Por último, a la hora de mostrar los datos de las oportunidades caducadas no había forma de mostrarlo en formato tabla, ya que Teams no permitía tarjetas con formato tabla ni ‘Markdown’ y los mensajes de texto plano tampoco admitían el formato tabla. La forma de solucionar esto fue usar las tarjetas adaptativas, las cuales permiten establecer separadores entre columnas y filas simulando así el aspecto de una tabla y consiguiendo que el usuario pueda leer los datos de manera más sencilla. Para crear estas tarjetas con un número de filas dinámicas, he tenido que crear las clases correspondientes a los elementos del ‘json’ el cual genera la tarjeta y después creo la estructura de las tarjetas con estas clases y definitivamente lo convertimos a ‘json’ para poder retornar la tarjeta adaptativa.

En conclusión, gracias al trabajo desarrollado se ha conseguido facilitar mucho el trabajo de todos los usuarios del CRM de CIC y el de los posibles nuevos usuarios. Haciéndoles además poder trabajar de una forma más rápida y eficiente.

El trabajo futuro o algunas de las mejoras que se le pueden ir añadiendo a este proyecto son la mejora de la exposición de los datos, con una interfaz o estructura más visual, como la que se usa en las oportunidades. También está abierto a añadir nuevas funcionalidades que vayan surgiendo con el paso del tiempo debido a mejoras o cambios en el CRM. Entre otras, una de las líneas de futuro es realizar algo similar a lo que se ha hecho durante este proyecto con las oportunidades, pero esta vez con las ofertas.

A nivel más personal, gracias a este proyecto me he dado cuenta de lo rápido que cambian las tecnologías, ya que durante el desarrollo del proyecto ha habido alguna librería de la cual han sacado actualizaciones y nuevas formas de uso de las cuales es necesario estar atento e implementar. También me ha servido para conocer el mundo laboral y la forma de trabajar en una empresa real con proyectos reales. Más allá de los proyectos académicos desarrollados en la carrera. Con este proyecto también me he dado cuenta de todas las cosas que me ha proporcionado la carrera. Tanto a nivel de conocimientos teóricos y prácticos, como a nivel de independencia y ser capaz de solucionar problemas por mi cuenta.

8. Bibliografía

[1] Página oficial de Visual Studio. Disponible en: https://visualstudio.microsoft.com/es/vs/

[2] Página oficial de LUIS. Disponible en: https://www.luis.ai/

[3] Página oficial del Portal de Azure. Disponible en: https://portal.azure.com/

[4] Página oficial de GitLab. Disponible en: https://gitlab.com/

[5] Página oficial de Postman. Disponible en: https://www.postman.com/

[6] Página oficial de Teams. Disponible en: https://www.microsoft.com/es-es/microsoft-teams/group-chat-software

[7] Modelo vista controlador. Visto (23/06/2021). Disponible en: https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html

[8] Arquitectura genérica chatbot Azure. Visto (19/06/2021). Disponible en: https://docs.microsoft.com/es-es/azure/architecture/reference architectures/ai/conversational-bot

[9] Información sobre Dynamics. Visto (29/06/2021). Disponible en: https://dynamics.microsoft.com/

[10] Inteligencia artificial. Visto (21/06/2021). Disponible en: https://www.xataka.com/robotica-e-ia/que-inteligencia-artificial

[11] Imagen sobre la metodología de desarrollo agile. Disponible en: https://upload.wikimedia.org/wikipedia/commons/5/50/Agile_Project_Management_by_Planbox.png

[12] Imagen del modelo arquitectónico modelo vista controlador (MVC). Disponible en: https://si.ua.es/es/documentacion/asp-net-mvc-3/imagenes/introduccion/flujo-mvc.png