



Sistemas Informáticos Avanzados

Sistemas distribuidos y web

Tesis fin de Máster

Sheetchat - Generación de chatbots a partir de hojas de cálculo.

Autor

Haritz Medina

Director
Óscar Díaz



2016

Resumen

Las hojas de cálculo son herramientas que tienen un uso extendido con más de 750 millones de usuarios a lo largo del mundo. Permiten crear complejas simulaciones, modelar situaciones financieras o analizar y mostrar datos a clientes. Sin embargo son herramientas que dentro del mundo móvil son muy demandadas pero complejas de utilizar. Recientes estudios afirman que cerca de un 80 % de los usuarios de hojas de cálculo no disponen de un ordenador cuando necesitan consultar datos almacenados en hojas de cálculo. En este trabajo se ha definido un DSL y una aplicación llamada SheetChat que permite a un usuario definir y generar sus propios chatbots que le ayuden a consultar datos de su hoja de datos en una configuración móvil. Se han desarrollado tres casos de uso en tres contextos diferentes donde se evalúa la herramienta.

Palabras clave: SpreadSheets, Chatbot, Mobile Setting

Índice de capítulos

Resumen	I
Índice de capítulos	III
Índice de figuras	VII
1. Introducción	1
2. Introducción a los chatbots	5
2.1. Definición y contexto de uso	5
2.2. Plataformas para desarrollo de agentes conversacionales	7
3. Análisis y diseño de la solución	11
3.1. Requisitos	11
3.2. Modelo de características	12
3.2.1. Sheet	12
3.2.2. Chat	14
3.3. Modelo de datos	17
3.3.1. Metamodelo	17
3.3.2. Sintaxis concreta	18

ÍNDICE DE CAPÍTULOS

4. Implementación	21
4.1. Botkit	22
4.2. Lenguaje de implementación	24
4.3. Motor SQL para consultas. AlaSQL	24
 5. Casos de estudio	27
5.1. Ejemplo 1: Notas de asignaturas impartidas por un profesor	28
5.1.1. Hoja de cálculo con los datos	28
5.1.2. Análisis de preguntas	29
5.1.3. Definición del DSL	29
5.1.4. Ejemplo de uso del chatbot	31
5.2. Ejemplo 2: Calendario de sesiones en un congreso científico	33
5.2.1. Hoja de cálculo con los datos	33
5.2.2. Análisis de preguntas	33
5.2.3. Definicion del DSL	34
5.2.4. Ejemplo de uso del chatbot	35
5.3. Ejemplo 3: Búsqueda de restaurantes de Tripadvisor	35
5.3.1. Hoja de cálculo con los datos	36
5.3.2. Análisis de preguntas	37
5.3.3. Definicion del DSL	38
5.3.4. Ejemplo de uso del chatbot	39
 6. Trabajo Futuro y Conclusiones	43
6.1. Trabajo futuro	43
6.1.1. Diseño del DSL	44
6.1.2. Usabilidad de los Chatbot generados	44
6.1.3. Ecosistema de plataformas	45
6.1.4. Hojas de cálculo	45
6.2. Conclusiones	46

Anexos

A. Import.io: Extracción de datos tabulares a partir de la web	49
B. Wit.ai: Procesamiento del lenguaje natural orientado a bots conversacionales	51
Bibliografía	55
Agradecimientos	57

Índice de figuras

1.1. El problema de visualización de hojas de cálculo en dispositivos móviles debido al scrolling.	3
2.1. Un chatbot que informa sobre las elecciones españolas del 26-J ofreciendo información personalizada a cada usuario.	6
2.2. El chatbot define las opciones de interacción que ofrece mediante botones en lugar de esperar un mensaje textual.	8
3.1. Modelo de características de SheetChat.	13
3.2. Ejemplo de hoja de cálculo.	13
3.3. Funcionamiento de los filtros para realizar la búsqueda de la respuesta del chatbot en la hoja de cálculo.	16
3.4. Modelo de características de SheetChat.	19
3.5. DSL con la definición de las Sheet y un Intent con una Entity.	19
4.1. Diagrama de componentes UML de SheetChat.	22
4.2. Una conversación desarrollada con Botkit sobre Slack.	23
5.1. Hoja de cálculo del profesor para la asignatura de matemática.	28
5.2. Hoja de cálculo del profesor para la asignatura de física.	29
5.3. DSL con la definición de las Sheet y un intent a utilizar en el ejemplo de notas.	30

ÍNDICE DE FIGURAS

5.4. Definición de la pregunta para obtener la nota ponderada de los ejercicios de física del ejemplo de notas.	31
5.5. Interacciones del usuario a la hora de consultar las notas de Alberto.	32
5.6. El chatbot recomienda algunos nombres en caso de que no se defina o se defina un nombre inexistente para hacer el filtrado en la hoja de cálculo.	32
5.7. Hoja de cálculo con el programa del WISE 2015.	34
5.8. Implementación del chatbot de la conferencia. Cabe destacar remarcado en azul el intent con input multicolumna.	35
5.9. Interacción del usuario a la hora de preguntar por los eventos en una hora concreta (Imagen izquierda y central) y consulta respecto a un topic concreto (Imagen derecha).	36
5.10. Búsqueda de restaurantes de Miami en Tripadvisor	37
5.11. Hoja de cálculo con restaurantes de Miami extraídos de Tripadvisor.	38
5.12. DSL de SheetChat que describe el intent para la búsqueda por tipo de comida de los restaurantes de Tripadvisor.	39
5.13. DSL de SheetChat que describe el intent para la búsqueda por precio máximo (resaltado en azul) y tipo de comida de los restaurantes de Tripadvisor.	39
5.14. Interacción entre el usuario y el chatbot que recomienda restaurantes en base a un tipo de cocina.	40
5.15. El bot de Tripadvisor recomienda restaurantes italianos con precio menor a 15 dolares por petición del usuario.	41
A.1. Generador de URLs en las que extraer datos para import.io.	49
A.2. Pantalla de edición de los elementos HTML a extraer en la tabla generada por import.io.	50
B.1. Wit.ai infiriendo de dos frases cuales son los Intents correspondientes para cada uno de ellos y cuál es la entidad Alumno.	52
B.2. En la imagen de arriba Witai infiere erroneamente el Intent y no reconoce la entidad. En la inferior el usuario no ha proporcionado ninguna entidad, por lo que el chat tendrá que preguntar por ella.	52

B.3. Respuesta del servicio de wit.ai cuando se le proporciona una frase. Remarcado en morado está el grado de confianza referente a la entidad alumno que ha detectado, de azul la del Intent.	53
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

1. CAPÍTULO

Introducción

La inclusión del Smartphone se ha extendido hasta el punto de ser una herramienta indispensable en el día a día, tanto para comunicación [Montag et al., 2015], como para la búsqueda de información [Wang et al., 2016]. El uso del Smartphone en estos aspectos está superando a los sistemas de cómputo tradicionales como el PC o los portátiles. El Smartphone dispone actualmente una capacidad de trabajo similar a los PC, con la ventaja de la movilidad que ofrece. En la actualidad, con un Smartphone se pueden realizar la mayoría de tareas cotidianas que un usuario puede requerir, como leer el correo electrónico, comunicarse con sus seres queridos, consultar información en la web o realizar compras online.

La característica principal del Smartphone es la movilidad que ofrece frente a los PC o portátiles tradicionales. A pesar de las ventajas que ofrece, el uso de dispositivos móviles, estos acarrean la inclusión de nuevos factores que pueden dificultar su usabilidad [Zhang and Adipat, 2005] como son:

- **Contexto móvil:** el usuario puede cambiar su localización durante su uso. Asimismo, esto puede incluir interacción con su entorno, personas, objetos o el ambiente más próximo.
- **Conectividad:** la conectividad de los dispositivos móviles es lenta y poco fiable.
- **Tamaño de la pantalla:** la cantidad de información que se puede mostrar se reduce

debido al tamaño de la pantalla que llega a un máximo de 7“ (los conocidos como phablets¹).

- **Limitación de procesamiento:** los dispositivos móviles debido a su reducido tamaño disponen de menor capacidad de cómputo, lo que reduce el número de aplicaciones que son adecuadas para móviles.
- **Métodos de entrada de datos:** los dispositivos móviles ofrecen una peor usabilidad a la hora de introducir datos, el teclado táctil, los gestos y la voz son menos precisos y la inclusión de datos erróneos se incrementa [Flood et al., 2011b].

Debido a estas limitaciones, resulta complejo el uso de algunas aplicaciones en dispositivos móviles. Entre ellas destaca las hojas de cálculo, que son complejas debidas a las limitaciones de los Smartphone previamente descritas [Flood, 2010].

En la actualidad el uso de hojas de cálculo como Microsoft Excel o Google Spreadsheet es una de las herramientas más utilizadas en el manejo de información, en el ámbito empresarial, pero también a nivel personal. La potencia y versatilidad que ofrece es de sobra conocida, de ahí que exista gran cantidad de hojas de cálculo para el almacenamiento de datos. Actualmente cerca de 750 millones de usuarios utilizan Microsoft Excel para presentar y analizar datos [Investitech, 2015].

Un estudio reciente ha demostrado que el 79 % de los participantes (seleccionados a través del European Spreadsheet Risk Interest Group²³) requieren de acceso a hojas de cálculo en una configuración móvil [Flood et al., 2011a], especialmente en estos contextos de uso:

- En el día a día o tareas cotidianas
- Demostración de datos a clientes
- Cuando reciben un correo electrónico con una hoja de cálculo (dado que es el medio de compartición favorito para las hojas de cálculo).
- A la hora de debatir cambios urgentes con compañeros de trabajo.

¹Los phablet son dispositivos móviles denominados de esta manera por comprenderse en un tamaño mayor que los smartphones (hasta 5") y menor que los tablets (a partir de 7"): <https://en.wikipedia.org/wiki/Phablet>

²Sitio web de EuSpRiG: <http://www.cimglobal.com/Thought-leadership/Newsletters/Insight-e-magazine/Insight-Archive/Are-you-managing-your-spreadsheet-risk/>

³¿Cuáles son los riesgos de las hojas de cálculo? <http://www.cimglobal.com/Thought-leadership/Newsletters/Insight-e-magazine/Insight-Archive/Are-you-managing-your-spreadsheet-risk/>

Tal y como se puede apreciar en este estudio, el uso de las hojas de cálculo en una configuración móvil es diferente a la que se presenta en el uso de los ordenadores tradicionales. El uso más habitual es el de consultar o modificar datos de hojas de cálculo ya existentes, es decir, que rara vez se diseñan hojas de cálculo nuevas, o se cambia gran cantidad de información de las mismas.

Una vez conocidos los usos más habituales de las hojas de cálculo en un entorno móvil y las limitaciones del Smartphone, **¿Cuáles son las razones por las que resulta complejo interactuar con ellas en un entorno móvil?**

El limitado tamaño de la pantalla obliga al usuario a realizar muchas más interacciones cuando esta visualizando extensas hojas de cálculo. Esto provoca dos problemas [Flood et al., 2011b]:

- **Usabilidad pobre o limitada:** En la Figura 1.1 se presentan dos imágenes de una hoja de cálculo. Cada fila representa a un alumno y cada columna un parcial de una asignatura concreta. Como se puede observar, resulta tedioso obtener los exámenes de un alumno dado que hay que ir haciendo scrolling a lo largo de la hoja y rápidamente se pierde el foco de en qué fila está el dato asociado a ese alumno.
- **Sobrecarga cognitiva:** Los usuarios, debido a que solo pueden ver pequeñas porciones de la hoja de cálculo, tienen dificultades para conceptualizar la hoja de cálculo y como casa la porción que están visualizando en el conjunto global de la hoja.

The figure consists of two side-by-side screenshots of a mobile application interface titled "NotasAsignaturas".

Left Screenshot (Row Labels A-E):

Alumno	EjerEcuaciones	EjerTrigonometria	EjerFunciones1	EjerFunciones2
Adrian Arana	6	7	5.5	-
Santiago Rocha	8	6	6.5	-
Alberto Ballester	4.5	4	3.5	-
Jesus Barral	5.5	4	3.5	-
Alejandro Baquero	5	5.5	7	-
Maria Cuevas	8	9	7.5	-
Aurora Soler	6	4.5	3	-
Esther Espina	8	7.5	6.5	-
Jaime Brito	5	5.5	4.5	-

Right Screenshot (Row Labels G-K):

Integrales	ExamenEcuaciones	ExamenFunciones	ExamenDerivInt	NotaContinua
6.5	4.5	5.5	6	5.9
7.5	8.5	7	8	7.75
3.5	4	5	3.5	3.85
2.5	6	2.75	3	3.45
8	7	8	8	6.925
9.5	8.5	9.5	9	8.975
6.5	4.5	4	4.5	4.7
7	8	6.5	6	6.75
3.5	5.5	5.5	5	4.9

Figura 1.1: El problema de visualización de hojas de cálculo en dispositivos móviles debido al scrolling.

La idea principal de este trabajo se apoya en dos premisas:

- **Hay usuarios que trabajan con hojas de cálculo en entornos móviles,** especialmente para la consulta de los datos. La solución debe de ir enfocada a usuarios

como son los comerciales, administrativos, analistas,... que son los que principalmente hacen uso de esta herramienta en entornos móviles.

- **El acceso con el dispositivo móvil a las hojas de cálculo es complejo** o farragoso con las aplicaciones actualmente existentes (como Microsoft Excel⁴ o Google Sheets⁵). Sin embargo, como se ha mencionado previamente, el uso de las aplicaciones de chat se han adaptado perfectamente al día a día de los usuarios y están muy extendidas dentro de los dispositivos móviles. Son utilizadas principalmente para hablar con seres queridos o compañeros de trabajo [Montag et al., 2015], aunque también está creciendo el uso para interactuar con Chatbots⁶. Un Chatbot es un software capaz de resolver preguntas sobre un ámbito concreto realizadas en una interfaz textual. Se puede encontrar una introducción a los Chatbots en el Capítulo 2.

Por lo tanto, la pregunta a la que se trata de dar respuesta en este trabajo es **¿Cómo se puede utilizar un Chatbot como interfaz para extraer la información que un usuario requiere?**. Para ello este documento presenta el trabajo bibliográfico y de investigación que se ha desarrollado entorno a la generación de Chatbots para autoconsumo utilizando hojas de cálculo. En este Capítulo 1 se ha presentado el contexto en el que se desarrolla el proyecto y el problema a resolver. En el Capítulo 2 se presenta una introducción a los Chatbots, que son la interfaz de consulta que se utilizará para acceder a las hojas de cálculo. En el Capítulo 3 se analizarán los requisitos y las características que debe de tener el artefacto a implementar para resolver la problemática sobre la que se sostiene este trabajo. En el Capítulo 4 se hace hincapié en la implementación y las tecnologías utilizadas para el desarrollo del software. En el Capítulo 5 se presentan tres casos de estudio para evaluar la aplicabilidad de la solución adoptada. Finalmente en el Capítulo ?? se presenta el trabajo futuro que se va a realizar en el contexto de este trabajo y las conclusiones extraídas del mismo.

⁴Sitio web de las aplicaciones móviles de Microsoft Excel: <https://products.office.com/en/mobile/office>

⁵Sitio web de descarga de Google SpreadSheets para Android e iOS: <https://www.google.com/mobile/drive/>

⁶Tendencia de interés respecto al término “chatbot” en Google Trends: <https://www.google.com/trends/explore?q=chatbot>

2. CAPÍTULO

Introducción a los chatbots

En este capítulo se tratará de abordar uno de los conceptos fundamentales en los que se basa el trabajo realizado, los chatbots. En el Apartado 2.1 se describe qué es un chatbot y en qué contextos se utiliza. En el Apartado 2.2 se explicará las actuales plataformas para el desarrollo de chatbots, sus características y la solución que se ha adoptado en este trabajo.

2.1. Definición y contexto de uso

Un chatbot, bot conversacional o agente conversacional se puede describir como un software para automatizar infinidad de tareas que actualmente desarrollan los usuarios por sí mismos, como reservar un restaurante para cenar, añadir un evento al calendario u obtener información [Wagner, 2016].

Un concepto que habitualmente se confunde o se solapa (en cierta medida) con los chatbots son los asistentes personales o asistentes digitales. Ambas ideas comparten algunas características que se presentan a continuación:

- Tanto los chatbots como los asistentes personales disponen (o pueden disponer) de interfaz textual o por voz.
- Ambas ideas tienen como objetivo automatizar tareas cotidianas.
- Son capaces de integrar múltiples servicios.

Sin embargo, presentan una diferencia principal. Un chatbot está orientado a resolver ciertos problemas, digamos que es experto en un ámbito concreto, puede actuar como representante de una empresa o un servicio. Mientras tanto un asistente personal juega el papel de oráculo y tiene que lidiar con cualquier tipo de tarea [Wright, 2016].

Lo que las definiciones dejan claro es que un bot conversacional tiene un propósito específico y que en líneas generales es un software que debe de lidiar con o resolver tareas cotidianas del usuario.

A continuación se puede ver un ejemplo. En la Figura 2.1 se puede observar la interacción que tiene un usuario con un chatbot llamado Politibot. Politibot¹ es un agente conversacional que permite ofrecer información acerca de las elecciones generales de España. En la imagen izquierda Figura 2.1 se puede ver cómo el bot recaba cierta información del usuario (rango de edad, localización) para de esta manera ofrecer información más relevante para el elector. En la imagen derecha, el usuario pregunta por los resultados de las elecciones, a lo que el chatbot proporciona una respuesta personalizada, mostrando los datos de las elecciones generales, pero también los de su provincia.

Detrás de un chatbot de estas características se añan diferentes servicios: bases de datos con las preferencias de los usuarios, sistemas de localización o datos extraídos de servicios web para las noticias o los resultados.

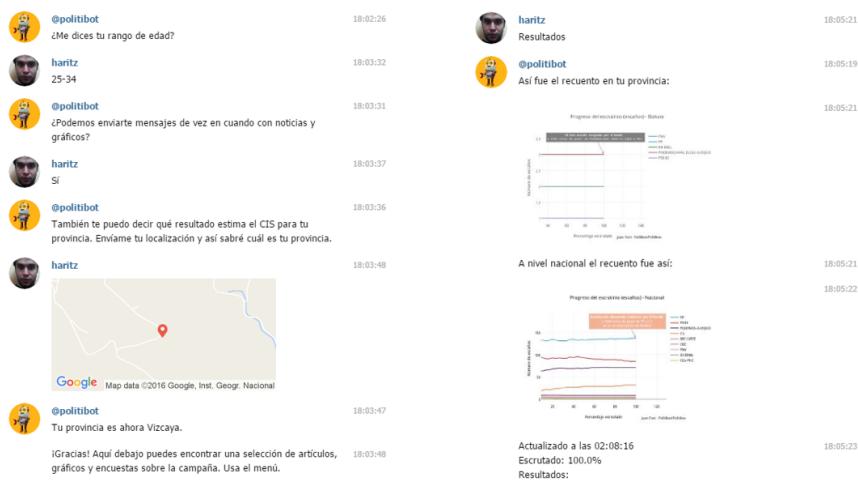


Figura 2.1: Un chatbot que informa sobre las elecciones españolas del 26-J ofreciendo información personalizada a cada usuario.

Cualquiera puede pensar que los bots son un concepto nuevo dentro de las tecnologías de

¹Sitio web de @politibot: <https://politibot.es/>

la información. Sin embargo, es una idea que lleva desde los comienzos de la informática (uno de los pioneros fue el proyecto ELIZA²), aunque en los últimos años está en crecimiento [Ferrara et al., 2016]. Esto es debido al aumento de las redes sociales y de los dispositivos interconectados dentro de lo que se conoce cómo el Internet de las Cosas³.

Los chatbots proporcionan una interfaz de comunicación en la que se reduce el coste frente a la interacción humana [Dans, 2016]. De igual manera también porque esta interacción ejerce menor presión en el usuario que quiere realizar consultas. Un bot está disponible para atender consultas 24 horas al día los 7 días de la semana, y puede atender simultáneamente consultas de múltiples usuarios, a diferencia de los humanos.

En el próximo Apartado 2.2 se hará hincapié en las plataformas que ofrecen chatbots y las características de los mismos.

2.2. Plataformas para desarrollo de agentes conversacionales

Como se ha mencionado en el apartado anterior, los chatbot existen desde hace varias décadas. Sin embargo, con el uso de las tecnologías móviles y el aumento del uso de aplicaciones de chat para conversar [Montag et al., 2015], ha hecho que los bots se hayan puesto en boga nuevamente.

Hay que diferenciar dos aspectos a la hora de hablar de plataformas para los chatbots. Por un lado existen las plataformas donde tiene el chatbot su interfaz. La interfaz de un chatbot es una aplicación o un servicio web por el cual chatea el usuario con el bot. Este apartado hablará de estas plataformas. Por otro lado está la plataforma de desarrollo de los bots, que es la librería o servicio que se utiliza para desarrollar un chatbot que después será desplegado en una o más plataformas de interfaz.

En referencia a la interfaz de los chatbots, en la actualidad muchas empresas ofrecen su plataforma como interfaz para interactuar con los chatbots. Entre ellas destacan: Facebook, Twitter, Telegram, Microsoft Skype o Slack. Sin embargo, empresas como Kik

²Proyecto ELIZA, un chatbot que simulaba a una psicóloga: <https://en.wikipedia.org/wiki/ELIZA>

³Utilización de chatbots como interfaces para el Internet de las cosas: <https://iot.telefonica.com/blog/using-smart-chatbots-as-an-iot-interface>

llevan desde hace años trabajando y ofreciendo su plataforma para el consumo de chatbots⁴.

La plataforma de desarrollo está muy ligada a la interfaz. Volviendo al ejemplo de Polibot, en la Figura 2.2 se observa como la interfaz de Telegram proporciona botones con las diferentes opciones que el usuario puede elegir para interactuar con el bot. Esta característica es particular de Telegram, que por ejemplo Slack no dispone. Sin embargo, otras plataformas ofrecen otras características que Telegram no soporta, por lo que hay que evaluar las diferentes plataformas antes de desarrollar un chatbot.



Figura 2.2: El chatbot define las opciones de interacción que ofrece mediante botones en lugar de esperar un mensaje textual.

Es por ello que existen muchas plataformas de desarrollo. Algunas de ellas, como Microsoft Bot Framework⁵ ofrecen soporte multiplataforma (Telegram, Slack, Skype, Messenger,...), otras como la API de Telegram⁶ es exclusiva para Telegram.

En este trabajo, se ha decidido trabajar con la librería Botkit⁷. Las principales razones son las siguientes:

- Es multiplataforma, actualmente soporta Slack, Facebook Messenger⁸ y Twilio IP Messaging⁹.

⁴ ¿Como predijo Kik el crecimiento del uso de los chatbots? <https://goo.gl/Cd50Vc>

⁵ Microsoft Bot Framework: <https://dev.botframework.com/>

⁶ Telegram bot API: <https://core.telegram.org/bots>

⁷ Sitio web de Botkit: <https://github.com/howdyai/botkit>

⁸ Sitio web de Facebook Messenger: <https://facebook.com/>

⁹ Sitio web de Twilio: <https://www.twilio.com/docs/api/ip-messaging>

- Es software libre, lo que permite ver el código fuente y modificarlo, además de que no tiene ningún coste económico.
- Es sencillo de desarrollar, permite abstraer bastante la implementación a bajo nivel de los chatbots, que es compleja debido a las llamadas asíncronas y las conexiones a múltiples servicios que trabajan por debajo.

Para profundizar en las características de implementación sobre Botkit es conveniente revisar el Capítulo 4.

3. CAPÍTULO

Análisis y diseño de la solución

En este capítulo se hablará del análisis y del diseño adoptado para resolver el problema de generar chatbots. Para ello, como se ha mencionado previamente, se ha definido un artefacto llamado SheetChat. En el Apartado 3.1 se hablará de los requisitos que ha de tener el artefacto que permita generar bots. En el Apartado 3.2 se realizará el análisis de las funcionalidades que ha de tener SheetChat. Finalmente, en el Apartado 3.3 se mostrará cual es el modelo de datos a definir para la generación de agentes conversacionales basados en hojas de cálculo.

3.1. Requisitos

En la captura de requisitos hay dos vertientes que hay que tener en cuenta al tratarse de un software generador. Por un lado, hay que prestar atención que el generador cumpla con los requisitos funcionales y no funcionales. Por otro lado, hay que asegurar que el software generado también cumple con los requisitos preestablecidos.

Cabe destacar, tal y como se mencionaba en el Capítulo 1, que el público objetivo que va a utilizar esta herramienta para el desarrollo va a ser gente con conocimientos de hojas de cálculo. Ellos mismos también serán los consumidores del chatbot que generen.

Dado que el usuario que utilizará SheetChat será un usuario con conocimientos para crear sus propias hojas de cálculo, se da por supuesto que éste **dispondrá de conocimientos de fórmulas matemáticas y de funciones de filtrado**. Es por ello que es apropiado propor-

cionarle un lenguaje que permita definir Chatbots con consultas y manipulación de datos más o menos potente.

Sin embargo, el usuario promedio de hojas de cálculo **no tiene conocimiento de los Chatbots**, qué es una API, un lenguaje de programación o procesamiento del lenguaje natural. Estos aspectos SheetChat deberá de abstraérselos al usuario.

Un aspecto fundamental es cómo se van a definir los Chatbots. La idea a largo plazo es que exista una herramienta gráfica que ayude a generar la sintaxis concreta de manera automática. Sin embargo dado el alcance de este trabajo, es conveniente **utilizar un lenguaje textual fácilmente legible por los seres humanos**.

En referencia al chatbot generado, este debe de cumplir con algunas características que se definen en el Capítulo 2. Estas funcionalidades pueden resultar obvias, pero rara vez se cumplen en los chatbots que existen en la red, provocando que la usabilidad se merme [Chan, 2016].

3.2. Modelo de características

El diseño de la solución que se ha adoptado para la generación de chatbots o agentes conversacionales utilizando hojas de cálculo se divide en dos partes claramente diferenciadas. Esto es debido al objetivo que se ha presentado en los anteriores apartados, que es generar un chatbot para la consulta de datos extraídos de hojas de cálculo. Tal y como se puede observar en la Figura 3.1, por un lado hay que definir el origen de datos, los Sheet u hojas de cálculo. Por otro lado el cómo se consultan los datos, es decir, el chat o flujo de conversación que se define para obtener los resultados.

En el Apartado 3.2.1 se hablará sobre cómo se manejan las hojas de cálculo para su posterior consulta mediante un chatbot. En el Apartado 3.2.2 se describirá cómo se define un chatbot, el tipo de consultas (ver Apartado 3.2.2) y el procedimiento para hacerlas. Asimismo se mostrará el cómo se realiza la humanización de un chatbot (ver Apartado 3.2.2), que permite mejorar la experiencia de usuario a la hora de conversar.

3.2.1. Sheet

La palabra Sheet de SheetChat hace referencia a las hojas de cálculo. Las hojas de cálculo son unas herramientas muy potentes que permiten, además de generar gráficas o trabajar

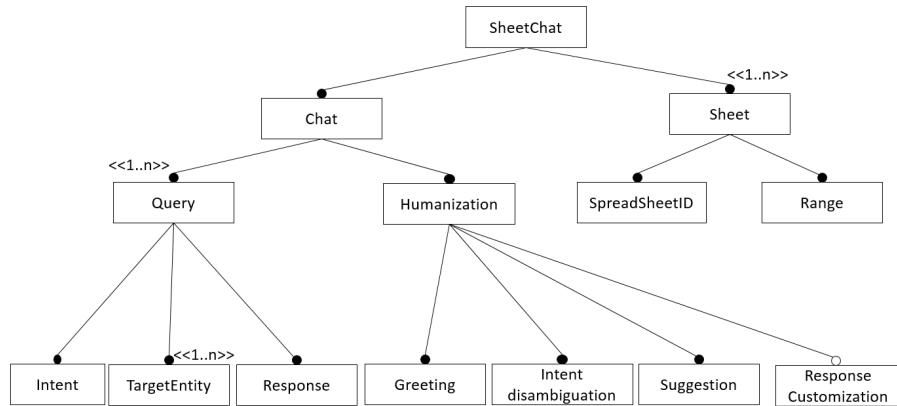


Figura 3.1: Modelo de características de SheetChat.

con complejas funciones matemáticas, el almacenar datos en forma matricial (filas y columnas). Las hojas de cálculo están orientadas al análisis de datos y las bases de datos al almacenamiento de las mismas [Philips, 2016]. Sin embargo, ambas almacenan datos basados en columnas y filas (o tuplas).

En el enfoque de nuestro trabajo, cada hoja de cálculo representa un conjunto de datos en forma de tabla. Al igual que sucede con los ficheros CSV, la primera fila es la cabecera de la tabla, donde se especifica cual es el nombre de cada columna (ver resaltado en azul de la Figura 3.2). Las posteriores filas se traducen en tuplas que serán los datos a consultar.

Se pueden definir como origen de datos tantas hojas de cálculo se requieran para el chatbot. La única condición es que no tengan el mismo nombre la hoja de cálculo (resaltado en morado en la Figura 3.2) y que los datos estén estructurados de una manera similar a lo que se puede observar en la Figura 3.2.

	A	B	C	D	E	F	G	H	I	J	N
1	Alumno	EjerEcuaciones	EjerTrigonometria	EjerFunciones1	EjerFunciones2	EjerDerivadas	EjerIntegrales	ExamenEcuaciones	ExamenFunciones	ExamenDerivadas	
2	Adrian Arana	6	7	5.5	6	7	6.5	4.5	5.5	6	
3	Santiago Rocha	8	6	6.5	6.5	7	7.5	8.5	7	8	
4	Alberto Ballester	4.5	4	3.5	3	2.5	3.5	4	5	3.5	
5	Jesus Barral	5.5	4	3.5	3	1.5	2.5	6	2.75	3	
6	Alejandro Baquero	5	5.5	7	7.5	7	8	6	7	8	
7	Maria Cuevas	8	9	7.5	9.5	9	9.5	8.5	9.5	9	
8	Aurora Sabater	6	4.5	3	4.5	5.5	6.5	4.5	4	4.5	
9	Esther Espina	8	7.5	6.5	4.5	6.5	7	8	6.5	6	
10	Jaime Brito	5	5.5	4.5	4	4	3.5	5.5	5.5	5	
11											

Figura 3.2: Ejemplo de hoja de cálculo.

A la hora de definir un chatbot se pueden definir más de un origen de datos. En la tecnología utilizada hay que mencionar que se definen hojas de cálculo en Google SpreadSheet¹. La principal ventaja que ofrece esta tecnología es que los datos residen en la nube, lo que permite ser accedidos desde cualquier plataforma y en cualquier momento y siempre se dispondrá de la versión más actualizada de los datos a consultar.

Los datos de las hojas de cálculo serán adaptadas a una base de datos que permite consultas SQL. Esto permite que se puedan hacer consultas con cierto nivel de riqueza sobre las hojas de cálculo (funciones matemáticas y funciones de filtrado). Estas consultas se definen en el sistema de conversación de SheetChat en el Apartado 3.2.2.

Cabe mencionar que SheetChat se encargará de hacer transparente la extracción (de Google SpreadSheets), transformación (a una base de datos en memoria) y actualización de los datos en tiempo real, tal y como se puede profundizar en el Apartado 4.3.

3.2.2. Chat

El Chat de la palabra SheetChat hace referencia a la conversación que se ha de definir. La conversación es la interfaz de consultas a la hoja de cálculo que tendrá el usuario. Una conversación consiste en un proceso donde el usuario y el chatbot intercambian mensajes hasta que el usuario consigue resolver su duda u obtener la información que requiere. Una conversación puede consistir en un mensaje o más, dependerá en este caso de si el chatbot puede recabar todos los datos necesarios para proporcionar la respuesta.

El proceso de definir un Chat consta de dos partes. En el Apartado 3.2.2 se estudia qué son las consultas en SheetChat y en el Apartado 3.2.2 se estudia las técnicas de humanización del bot que proporciona SheetChat.

Consultas

Las consultas en SheetChat funcionan como el nexo entre una conversación y la hoja de cálculos. Definir adecuadamente una consulta permite que el usuario pueda preguntar por algo en concreto y el bot pueda obtener una respuesta rápida y fiable en la hoja de cálculo.

Una consulta, tal y como se puede observar en la Figura 3.1 consta de tres partes, el Intent, las Entidades y la Respuesta. Definiendo estos tres aspectos, SheetChat es capaz

¹Sitio web de Google SpreadSheets: <https://www.google.com/sheets/about/>

de reconocer las preguntas que el usuario realice al chatbot y proporcionar una respuesta al mismo.

Para reconocer la pregunta que el usuario hace se definen dos aspectos, el Intent y las Entidades:

- El **Intent** representa la intención del usuario. “Saber si un alumno ha aprobado una asignatura”, “Saber a qué hora parte el próximo tren a un destino concreto” o “Informarse sobre los restaurantes con comida de un tipo concreto” son intenciones de usuario. Los Intents esperan como respuesta un concepto muy concreto, sin embargo la manera de preguntar sobre el mismo Intent puede ser bastante diferente: “Ha aprobado un alumno concreto la asignatura de Inglés?” o “Debe un alumno concreto presentarse a la segunda convocatoria de Inglés?” hacen referencia al mismo Intent. Sin embargo, para un sistema computacional es complejo desambiguar esto. En el Apartado [3.2.2](#) se profundiza en cómo resolver esta problemática.
- Las **Entidades** representan los elementos u objetos de un contexto determinado. En las preguntas de ejemplo definidas anteriormente, no hacían referencia a un objeto del contexto. En la pregunta “Ha aprobado un alumno concreto la asignatura de Inglés?” no se especificaba el nombre de a qué alumno se refiere el usuario. Las entidades vienen a representar en este contexto a los alumnos, Miguel, María, Juan,... En las consultas las entidades hacen la función de filtro, donde el Chatbot filtrará las notas de los alumnos en base a un alumno concreto. Cabe destacar que SheetChat, en caso de no reconocer todas las Entidades realizará preguntas para que el usuario le proporcione esa información.

Una consulta, como se ha mencionado antes, tiene una tercera pata que se le ha denominado Respuesta. La **Respuesta** especifica qué columnas del filtrado aplicado se deben de utilizar como respuesta. Es decir, que la respuesta debe de ser la columna NotaFinal y no la nota de un parcial, ya que se desea obtener si un alumno ha aprobado y la respuesta está en esa columna.

Se podría decir que los Intent y las Entidades funcionan como un filtrado horizontal mientras que la respuesta hace un filtrado vertical. En la Figura [3.3](#) se puede observar resaltado de color azul como mediante el Intent y la Entidad se ha realizado un filtrado de posibles respuestas quedando un sólo alumno que cumple con esos criterios de búsqueda. En la Figura [3.3](#) de color verde, la respuesta especifica en qué columna se encuentra el resultado a mostrar al usuario. Esta idea de búsqueda parte de la premisa de que los datos en una

fila están relacionados entre sí, es decir, en la fila resaltada en azul en la Figura 3.3 las calificaciones pertenecen a María Cuevas y no hay datos que no estén relacionados con María Cuevas.

A	B	C	D	E	F	G	H	I	J	K	L
Alumno	EjerEcuaciones	EjerTrigonometria	EjerFunciones1	EjerFunciones2	EjerDerivadas	EjerIntegrales	ExamenEcuaciones	ExamenFunciones	ExamenDerivInt	NotaContinua	ContAprobado
2 Adrian Arana	6	7	5.5	6	7	6.5	4.5	5.5	6	5.9	sí
3 Santiago Rocha	8	6	6.5	6.5	7	7.5	8.5	7	8	7.375	sí
4 Alberto Ballester	4.5	4	3.5	3	2.5	3.5	4	5	3.5	3.85	no
5 Jesus Barral	5.5	4	3.5	3	1.5	2.5	6	2.75	3	3.45	no
6 Alejandro Baquero	5	5.5	7	7.5	7	8	6	7	8	6.925	sí
7 María Cuevas	8	9	7.5	9.5	9	9.5	8.5	9.5	9	8.975	sí
8 Aurora Sabater	6	4.5	3	4.5	5.5	6.5	4.5	4	4.5	4.7	no
9 Esther Espina	8	7.5	6.5	4.5	6.5	7	8	6.5	6	6.75	sí
10 Jaime Brito	5	5.5	4.5	4	4	3.5	5.5	5.5	5	4.9	no
11											

Figura 3.3: Funcionamiento de los filtros para realizar la búsqueda de la respuesta del chatbot en la hoja de cálculo.

De igual manera se pueden realizar consultas enriquecidas, donde la entidad o el resultado fuesen un conjunto de más de una columna. Esto dependerá del dominio y de la representación de los datos. En los casos de estudio del Capítulo 5 se profundiza más en qué tipo de consultas enriquecidas pueden definirse. Para realizar estas consultas se trabaja con un subconjunto de funciones de SQL, del que se puede profundizar en el Apartado 4.3.

Humanización

La humanización de un chatbot consiste en hacer que un chatbot parezca más humano. Esto permite ofrecer una experiencia de usuario más próxima a una interacción con un humano real posible. Al fin y al cabo, un chatbot es un software y por muy humano que resulte, seguirá siendo un software, pero para un usuario no es lo mismo interactuar mediante comandos o mediante frases que le puedan salir de forma natural.

Para ello se ha trabajado en varios aspectos que permiten una experiencia de usuario más rica. Aunque se pueda mejorar, cabe recordar que el esfuerzo de humanizar conlleva un coste de tiempo de desarrollo. Si este coste es muy elevado, es posible que el usuario decida utilizar su aplicación de hojas de cálculo de toda la vida. Los cuatro aspectos en los que se ha hecho hincapié son los siguientes:

- **Saludo:** Cada Chatbot ha de disponer de un mensaje de bienvenida o ayuda. Además de ser una buena práctica, este mensaje proporciona feedback (aunque sea a un nivel muy básico) de las funcionalidades que tiene el chatbot. Esto puede resultar útil a pesar de ser un chatbot para autoconsumo, ya que si se está un periodo largo de tiempo sin usarlo, se le podría olvidar al usuario en qué consistía ese chatbot.

- **Desambiguación del Intent:** tal y como se ha comentado en el Apartado 3.2.2 es necesario ser capaz de interpretar cuál es el Intent al que se refiere el usuario. Para ello se ha utilizado un servicio que mediante procesamiento del lenguaje natural desambiguar Intents y reconocer Entidades. En el Anexo B hay más información acerca de este servicio denominado Wit.ai.
- **Sugerencias:** en una conversación puede surgir que el usuario al hacer una petición no haya introducido todas las Entidades que requiere el chatbot para proporcionar una respuesta. En estas situaciones el chatbot irá preguntando por cada una de las entidades que falten. Para mejorar la experiencia del usuario, el chatbot proporcionará sugerencias sobre posibles valores que hay registrados en la base de datos.
- **Personalización de la respuesta:** al trabajar con hojas de cálculo, el resultado de las consultas son celdas con un dato o un conjunto de datos concreto. A la hora de visualizar eso en una aplicación de mensajería instantánea resulta farragoso y bastante robótico. Es por eso, que se pueden definir frases que se completan con la respuesta que extraída de la hoja de cálculo. Es una técnica sencilla y que ayuda a una interpretación de la información más rápida.

3.3. Modelo de datos

Todas las características presentadas en el Apartado 3.2 se han de representar en un modelo de datos. En este apartado en primer lugar se presentará el metamodelo y después la sintaxis concreta que será la que el motor de SheetChat interprete para generar el bot.

3.3.1. Metamodelo

En la Figura 3.4 se puede observar cual es el Metamodelo definido para los chatbots generados por SheetChat. Un **SheetChat** consta de uno o más Sheets (u hojas de cálculo) y de un Chat.

Un **Sheet** está representado por un nombre de la hoja de cálculo, que debe de ser único, y el rango de los datos que conforman los datos que se quieren consultar.

Un **Chat** contiene una descripción del chatbot y un mensaje de saludo. El mensaje de saludo puede ser cualquier contenido textual, aunque como se ha mencionado en el Apartado 3.2.2, cuanto más descriptivo sea, mejor.

Un Chat contiene uno o más Intents. Cada **Intent** estará asociado a una hoja de cálculo donde deberá de buscar los datos, y contiene un identificador único, que será el que se defina en el entrenamiento de Wit.ai [B](#).

De igual manera un Intent está conformado por una o más Entity. Cada **Entity** tendrá que estar asociada a una o más columnas de la hoja de cálculo. El *Mask* de una Entity define el tipo de comparación que se debe de hacer con el valor de las columnas. Del Mask dependerá el filtrado que se aplica a una fila concreta. Para los valores de tipo String se pueden utilizar el “=” o el “LIKE”, mientras que para los valores numéricos se pueden hacer comparaciones de mayor, menor, mayor o igual, distinto de, etc. Las Entity permiten definir un mensaje personalizado para las preguntas que el chatbot puede hacer el usuario para recabar información acerca de la Entity a consultar. El *entityMissingMsg* es un mensaje que se muestra si no se ha encontrado en la pregunta inicial la entidad requerida. El *entityNotFoundMsg* hace referencia al caso donde el usuario ha proporcionado un valor para la entidad pero esta no se ha encontrado en la hoja de cálculo.

Por otro lado, un Intent tiene asociada una Response. Una **Response** consta de una o un conjunto de columnas o una columna extraída a partir de una fórmula matemática. El resto de parámetros son opcionales. El número de respuestas indica el número máximo de respuestas que puede dar el chatbot para esa respuesta. Es una funcionalidad útil si el filtrado genera muchas respuestas. *CustomResponseStructure* permite definir una respuesta con un texto personalizado que hace más amena su lectura. El *showColumnName* se puede utilizar en caso de que no se use el *CustomResponseStructure*. Esta se encarga de mostrar el nombre de la columna junto al valor de la celda respuesta. *NotFoundMessage* es un mensaje personalizado que se muestra al usuario en caso de que no se hayan encontrado resultados.

3.3.2. Sintaxis concreta

La sintaxis concreta está desarrollada en JSON. Entre los requisitos definidos, era conveniente que se utilizase un DSL definido en un lenguaje de fácil lectura y modificación por parte de un humano y sin duda JSON es de los mejores lenguajes para ello.

En la Figura [3.5](#) se observa la sintaxis concreta del caso de estudio descrito en el Apartado [5.1](#)

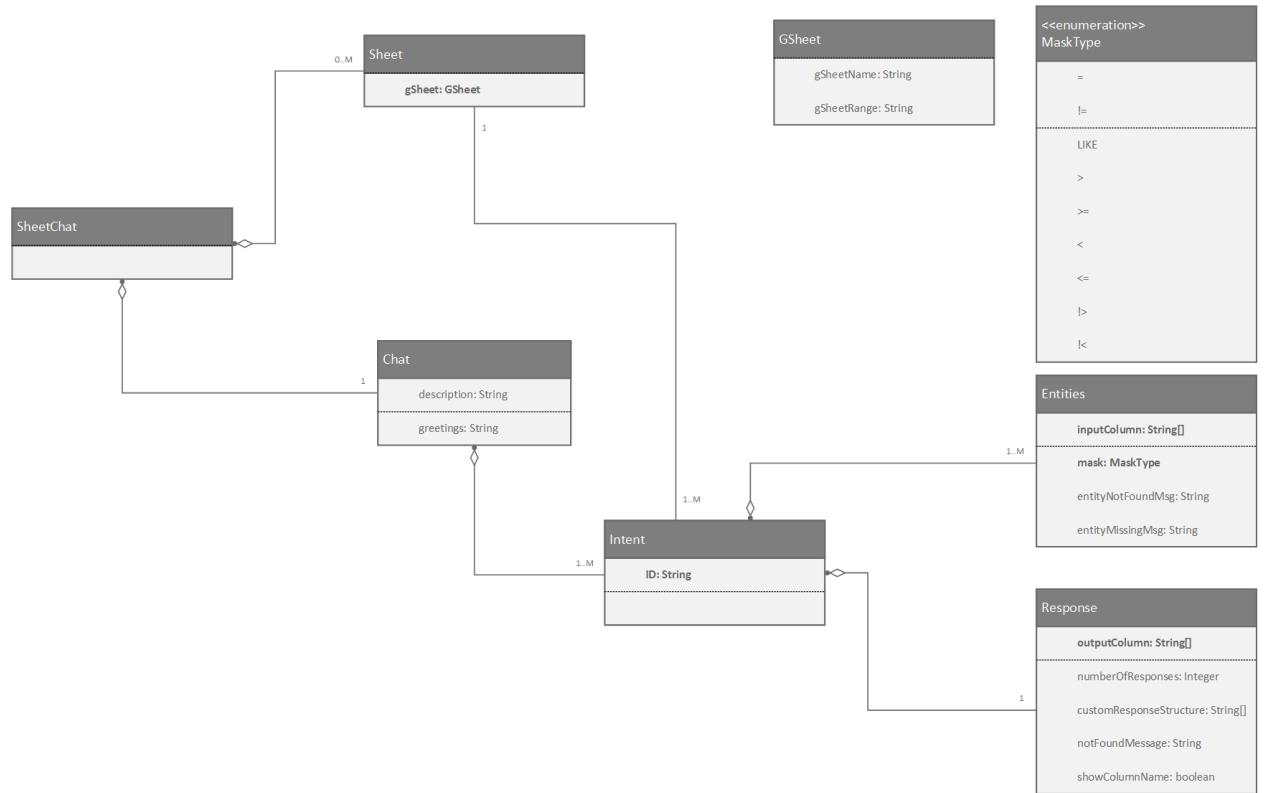


Figura 3.4: Modelo de características de SheetChat.

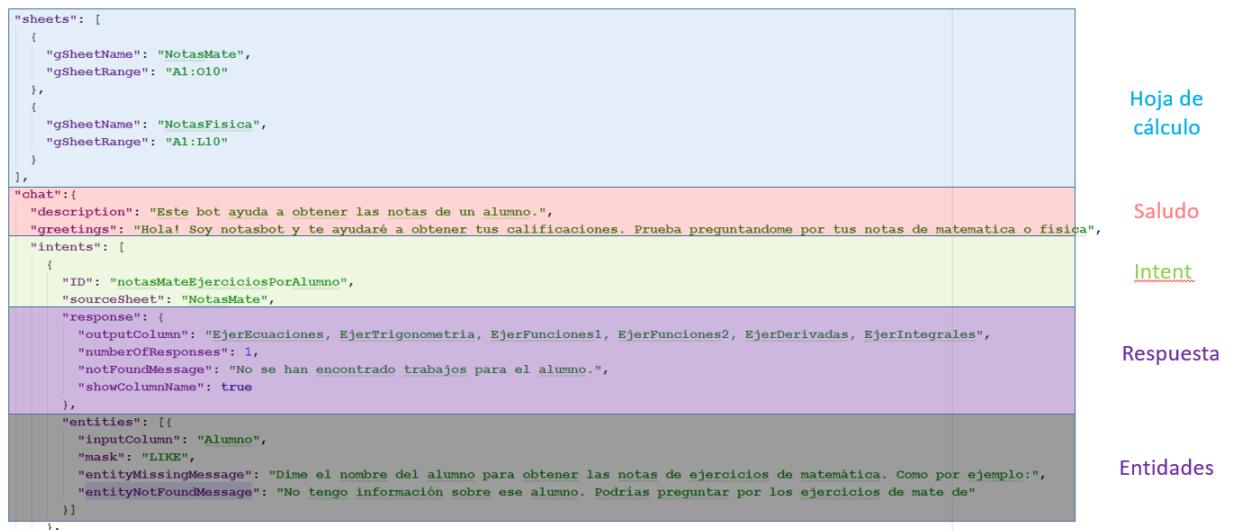


Figura 3.5: DSL con la definición de las Sheet y un Intent con una Entity.

4. CAPÍTULO

Implementación

En este capítulo se abordará los aspectos de implementación de SheetChat, que es un software que permite interpretar el DSL definido en el Capítulo 3 y desplegar un chatbot sobre la plataforma Slack.

Para abordar los aspectos de implementación, el capítulo se dividirá en tres secciones o apartados principales. En el Apartado 4.1 se presentará la plataforma de desarrollo de bots utilizada, Botkit, desarrollada en Javascript. En el Apartado 4.2 se hablará del lenguaje de programación utilizado y del ecosistema utilizado. Por último, en el Apartado 4.3 se hablará del motor SQL utilizado para permitir al DSL el uso de consultas enriquecidas con funciones matemáticas que se ha comentado en el Apartado 3.2.2.

Como se puede observar en la Figura 4.1, se han utilizado otros servicios web que forman parte del funcionamiento de SheetChat, pero de los que no se hablará en este capítulo. Estos son Wit.ai¹ y la API de Google Spreadsheets². Wit.ai se utiliza en la desambiguación de intents y reconocimiento de entidades, explicado en el Apartado 3.2.2. Sobre Wit.ai se puede profundizar en el Anexo B. La API de Google Spreadsheets permite la obtención de los datos de forma matricial de la hoja de cálculo alojada en Google Docs.

Cabe destacar que el código fuente y los ejemplos del Capítulo 5 están accesibles en el siguiente repositorio de Github: <https://github.com/haritzmedina/Sheetbot>.

¹Sitio web de Wit.ai: <https://wit.ai>

²Sitio web de la documentación de la API de Google SpreadSheet: <https://developers.google.com/sheets/>

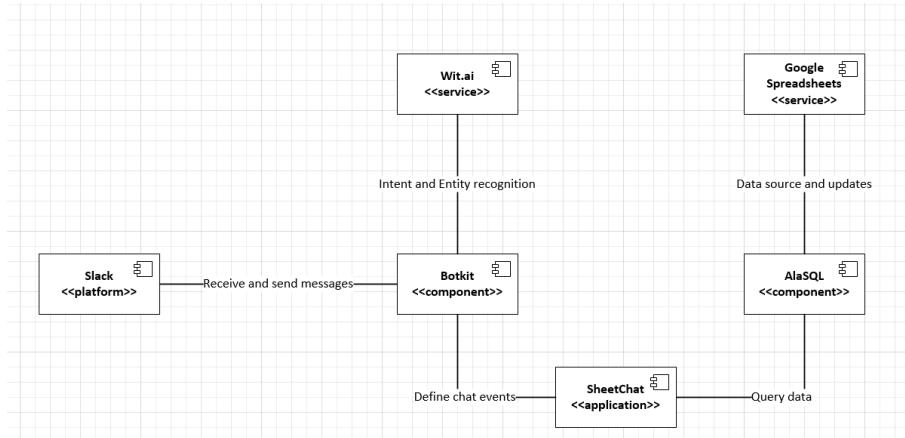


Figura 4.1: Diagrama de componentes UML de SheetChat.

4.1. Botkit

Botkit, es una librería que permite la programación de manera más sencilla de Chatbots distribuida bajo licencia MIT. Al igual que otras plataformas mencionadas en el Apartado 2.2, como Microsoft Bot Framework o la API de bots de Telegram, Botkit permite la abstracción de algunos aspectos de programación. Botkit es una librería desarrollada por Howdy.ai. Está programada en Javascript y funciona sobre Node.js³.

Botkit permite desarrollar bots para Slack, Facebook Messenger y Twilio. Como se ha mencionado previamente, en este trabajo se ha focalizado su uso sobre Slack. Para su uso simplemente será necesario crear un nuevo bot en Slack (o como se llama en Slack, una App⁴) e indicarle el token de acceso al bot.

Botkit presenta un sistema relativamente sencillo para crear bots conversacionales. Se basa en una programación orientada a eventos, de ahí que case perfectamente con Javascript, que es un lenguaje dirigido por eventos. Los tipos de eventos que se pueden definir son los siguientes:

- **Escuchar mensajes:** se puede establecer que Botkit esté a la escucha de un mensaje concreto y asociar una funcionalidad en caso de que se reciba el mensaje que se desea recibir. Se puede establecer en qué canales se desea escuchar dependiendo de la plataforma. En Slack los canales son: mensaje privado, mención a él en un grupo (donde esté añadido como miembro el bot) o mensaje recibido por cualquier canal.

³Node.js, es un entorno de ejecución de Javascript multi-plataforma: <https://nodejs.org/en/>

⁴Creación de apps o bots en Slack: <https://api.slack.com/bot-users>

- **Responder/Enviar mensajes:** una funcionalidad habitual cuando se recibe un mensaje es contestar con otro. En slack se puede definir un mensaje de respuesta más o menos elaborado, con emoticonos, textos coloreados, etc.

Botkit además de gestionar recepción y envíos de mensajes es capaz de entablar conversaciones. De esta manera se puede mantener la información intercambiada durante la conversación ya que puede ser relevante para dar una respuesta más precisa. En la Figura 4.2 se puede observar un ejemplo de conversación. En este caso el usuario ha pedido una pizza y el bot le va preguntando por el tipo de pizza, el tamaño y dónde debe de entregarse. Estos tres datos van asociados al mismo pedido, por lo que puede ser interesante utilizar la noción de conversación que Botkit proporciona.

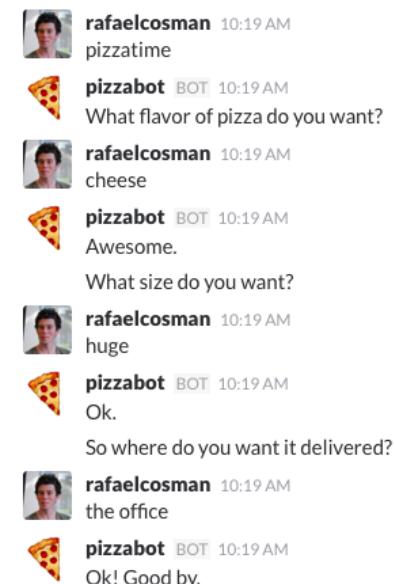


Figura 4.2: Una conversación desarrollada con Botkit sobre Slack.

En la generación de bots mediante SheetChat se ha utilizado esta característica de entablar conversación también. La creación de una conversación permite ir recabando las entidades necesarias para hacer el filtrado sobre los datos de la tabla y proporcionar el resultado que el usuario necesita. De igual manera, Botkit proporciona mecanismos de repetición de preguntas para el caso de que no haya podido interpretar la información que el usuario le ha proporcionado.

Como se ha mencionado previamente, en Botkit el funcionamiento reside en que necesita escuchar un mensaje para poder comunicar algo. Para poder decidir si la conversación que hay que abrir pertenece a un Intent o a otro se ha utilizado la técnica de *Intent Disambiguation* (ver Apartado 3.2.2) mediante la herramienta de procesamiento del lenguaje

natural Wit.ai. Para aunar el uso de Botkit con el de Wit.ai existe un middleware que se ha tenido que adaptar para este trabajo llamado [botkit-middleware-witai](#).

4.2. Lenguaje de implementación

El lenguaje de implementación utilizado es Javascript. En gran medida impuesto por la librería en la que se apoya SheetChat y que se acaba de exponer, Botkit, pero también porque es un lenguaje adecuado para el desarrollo de agentes conversacionales. En particular se ha utilizado el estándar ECMAScript2015⁵ o ECMA 6, que viene a ser la nueva versión de Javascript.

Tal y como se ha mencionado antes, el desarrollo se ha realizado sobre Node.js. Cabe destacar que los bots se ejecutan de manera local y se conectan a las diferentes APIs de los servicios de mensajería instantánea, como puede ser en este caso Slack. La idea es que esta aplicación resida en un servidor dedicado que permita su ejecución dando disponibilidad al bot de manera continua.

4.3. Motor SQL para consultas. AlaSQL

Una de las piezas fundamentales y sobre las que se basa la teoría de este proyecto son las hojas de cálculo y las consultas y filtrados que se le pueden realizar. Google SpreadSheet dentro de su API no proporciona filtrados de ningún tipo, ni un lenguaje potente para realizar consultas específicas sobre los datos. Es por ello que se ha decidido utilizar una implementación de SQL para javascript, llamada AlaSQL⁶.

El método de trabajo con AlaSQL en SheetChat es sencillo. Las filas y columnas que se reciben de Google SpreadSheet se traducen a tablas relacionales, cada tabla está representada por los valores de una hoja de cálculo. En caso de que los datos se actualicen, estos serán actualizados en la tabla de la base de datos en memoria de AlaSQL.

AlaSQL permite el almacenamiento en ficheros, importar datos de CSV, etc. Aunque en este caso sólo se ha utilizado el almacenamiento en memoria de los datos de las hojas de cálculo. Esto reduce en gran medida el tamaño de los datos que un Chatbot puede tener,

⁵Sitio web de ECMAScript2015: <http://www.ecma-international.org/ecma-262/6.0/>

⁶AlaSQL, una implementación de base de datos SQL en memoria en javascript: <http://alasql.org/>

pero resulta suficiente para un Chatbot de auto-consumo, y la velocidad de consulta es mucho mayor.

Cuando un usuario realiza una petición de unos datos al chatbot, estas consultas se traducen a sentencias SQL que serán ejecutadas sobre AlaSQL. Basándonos en el modelo de características de la Figura 3.1, se utiliza el response para definir las columnas a mostrar, la hoja de cálculo del intent como la tabla de la base de datos y las entidades como filtros. Suponiendo que la respuesta que da el Chatbot es “Examen1, Examen2, Examen3”, la hoja de cálculo es “Notas” y la entidad requerida es “Alumno”, la consulta se traduciría a algo similar a esta sentencia SQL: *SELECT Examen1, Examen2, Examen3 FROM Notas WHERE Alumno='X'*.

De esta manera se pueden realizar consultas más elaboradas, con funciones típicas de SQL como SUM, AVG, MAX, LIKE,... siempre que se mantenga la estructura de la consulta. Se descartó la idea de permitir al usuario que pudiese definir sus propias sentencias SQL limitándolas a la estructura SELECT-FROM-WHERE por las siguientes razones:

- La idea, de cara al futuro, es poder generar la sintaxis concreta de SheetChat mediante una herramienta gráfica que haga mucho más sencilla la especificación de los bots y no tenga el usuario que aprender el DSL.
- La idea no es consultar datos que requieran un procesamiento muy complejo, ya que para el procesamiento de fórmulas matemáticas complejas ya está la propia hoja de cálculo. El objetivo es poder prestar unas entidades derivadas de manera sencilla.

5. CAPÍTULO

Casos de estudio

En este apartado se evaluará el funcionamiento del DSL definido para la creación de un SheetChat. Para ello se tienen que tener en cuenta todas las características previamente mencionadas, la definición del origen de datos, la creación de una conversación que nos proporcione una petición de datos de entrada y su salida asociada; y no menos importante, los recursos que nos permitan humanizar el bot para ofrecer una experiencia de usuario agradable.

Para ello se han elaborado tres ejemplos. El primero de ellos es dado las notas de dos asignaturas impartidas por un profesor, el poder preguntar por las notas de los alumnos. El segundo de los ejemplos permite dado un calendario de sesiones de un congreso científico, en este caso extraido del WISE de 2015¹, poder obtener información sobre qué sesiones hay en los diferentes slots (u horarios). Por último, el tercer ejemplo permite, dada una hoja de cálculo autogenerada de una búsqueda de restaurantes de Miami en el sitio web Tripadvisor, obtener restaurantes por tipo de comida.

Los tres casos de estudio tendrán la misma estructura. En primer lugar se abordará el problema que el usuario tiene. En segundo lugar se realizará un análisis de las preguntas o cuestiones que tendrá que ser capaz de resolver el chatbot. Posteriormente se hará hincapié en la definición del DSL que tendrá que hacer el usuario. Finalmente se mostrará un ejemplo de interacción entre el chatbot generado y el usuario.

¹Calendario con las diferentes sesiones del congreso WISE: <http://www4.cis.fiu.edu/wise2015/@schema.html>

5.1. Ejemplo 1: Notas de asignaturas impartidas por un profesor

Para el primer ejemplo la audiencia objetivo es un profesor de instituto que dispone de las notas de sus alumnos almacenadas en hojas de cálculo. Él es profesor de dos asignaturas Matemática y Física.

5.1.1. Hoja de cálculo con los datos

Para el almacenamiento de las notas de sus alumnos dispone de dos hojas de cálculo, una con las notas de Matemática (ver Figura 5.1) y otra para las calificaciones de la asignatura de Física (ver Figura 5.2).

Tal y como se puede observar en la Figura 5.1 el profesor tiene las notas de cada uno de sus alumnos que va rellenando a medida que se les evalúa de los diferentes aspectos de las asignaturas. Por lo tanto, cada fila representa a un alumno y cada columna a cada concepto a evaluar de la asignatura.

En el método de evaluación empleado por el profesorado se tiene en cuenta trabajos de clase o ejercicios y exámenes durante la evaluación continua. En caso de que la media ponderada de estos parciales supere un 5 el alumno habrá aprobado mediante la evaluación continua. En caso contrario, tendrá que presentarse a un examen final con todo el temario del curso. A modo de ejemplo se puede visualizar en la Figura 5.1 como el alumno Adrian Arana ha aprobado la evaluación continua con un 5.9 mientras que Alberto Ballester tuvo que presentarse al examen final para aprobar dado que en la evaluación continua su nota era de 3.85.

Alumno	EjercCuestiones	EjerTrigonometria	EjerFunciones1	EjerFunciones2	EjerDerivadas	EjerIntegral	ExamenEjercicios	ExamenDerivadas	NotaContinua	ContAprobado	ExamenFinal	NotaFinal	Aprobado
2 Adrian Arana	6	7	5.5	6	7	6.5	5.5	6	5.9 si	0	5.9 si	7	
3 Santiago Rocha	8	6	6.5	6.5	7	7.5	8.5	7	8	7.375 si	0	7.375 si	
4 Alberto Ballester	4.5	4	3.65	3	2.5	3.65	4	5	3.85 no	5.3	5.3 si	6.35	
5 Jesus Lopez	5	4	3.5	3	1.5	2.5	6	2.75	3	3.85	4	4 no	
6 Alejandro Bejarano	5	5.5	7	7.5	8	6	7	8	6.925 si	0	6.925 si	9.5	
7 Maria Cuevas	8	9	7.5	9.5	9	9.5	8.5	9	8.975 si	9.5	9.5 si	9.5	
8 Aurora Sabater	6	4.5	3	4.5	5.5	6.5	4.5	4	4.5 no	4.5	4.7 no	4.7 no	
9 Esther Espina	8	7.5	6.5	4.5	6.5	7	6	6.5	6.75 si	0	6.75 si	6.75 si	
10 Daniel Otero	5	5.5	4.5	4	3.5	5.5	5.5	5	4.9 no	6	6.56		

Figura 5.1: Hoja de cálculo del profesor para la asignatura de matemática.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Alumno	EjerCinemática	EjerDinámica	EjerEnergía	EjerGravitatorio	Examen1	Examen2	NotaContinua	ContinuaAprobado	ExamenFinal	NotaFinal	Aprobado
2	Adrián Arana	5	5	5.5	6.5	7	6.5	6.25	si	0	6.25	si
3	Santiago Rocha	8	7.5	7	6.5	6	7	6.775	si	0	6.775	si
4	Alberto Ballesster	5	4	3.5	4	5	5	4.6	no	5.5	5.5	si
5	Jesus Barral	3.5	3	4.5	0	4	4.5	3.625	no	4	4	no
6	Alejandro Baquero	5	6.5	6	7	6.5	6	6.275	si	0	6.275	si
7	Maria Cuevas	8	7.5	7	8.5	8.5	9	8.325	si	0	8.325	si
8	Aurora Sabater	1.5	3.5	3	4.5	4.5	4	3.9	no	5.5	5.5	si
9	Esther Espina	7.5	6	5.5	4	5.5	4.5	5.225	si	0	5.225	si
10	Jaime Brito	7	7.5	7	8.5	7.5	9	7.975	si	8.5	8.5	si
11												
12												

Figura 5.2: Hoja de cálculo del profesor para la asignatura de física.

5.1.2. Análisis de preguntas

Una vez conocidos los datos almacenados, el profesor que quiere generar un chatbot tiene que tener en cuenta qué tipo de análisis quiere hacer sobre los datos, o dicho de otra manera, qué consultas va a realizar sobre los datos. Debería de ser capaz de decidir cuáles son las preguntas más habituales sobre esos datos o las que más frecuentemente le piden sus alumnos. Dado que son asignaturas diferentes y su método de evaluación no es exactamente el mismo, las preguntas que se puede llegar a realizar también son diferentes.

A continuación se recogen algunas de las preguntas que podría hacer el profesor:

- **¿Cuál es la nota que tiene un alumno en cada uno de los ejercicios de mate/física?** Es habitual preguntar por la nota en uno de los ejercicios o en el conjunto de los mismos a lo largo de la evaluación.
- **¿Cuál es la nota de un alumno en los exámenes parciales de mate/física?** Al igual que sucede con los ejercicios, es interesante conocer las notas de los exámenes parciales.
- **¿Se debe de presentar un alumno al examen final de mate?** O dicho de otra manera, ¿ha aprobado la evaluación continua?
- **¿Qué nota debe de sacar un alumno en los exámenes de física para aprobar la asignatura?** O preguntado de otra manera, ¿Qué nota ponderada tiene en los ejercicios de evaluación continua?
- **¿Ha aprobado un alumno la asignatura de física?** De esta manera se sabe si un alumno ha de presentarse a la segunda convocatoria de física o no.

5.1.3. Definición del DSL

Como se ha comentado en el Capítulo 3 hay que definir mediante el uso del DSL de SheetChat el origen de los datos y los intents para cada una de las consultas que se quiera

extraer de la hoja de cálculo previamente definidas en el Apartado 5.1.2.

Si se observa la Figura 5.3 se puede observar cómo es la definición de las hojas de cálculo a consultar. De igual manera se define una descripción para el chatbot y un mensaje de bienvenida que nos ayude a recordar alguna de las funcionalidades del bot. Posteriormente se muestra un intent de los que deberá de definir el profesor que vaya a utilizar el chatbot.

El intent a resolver es el de obtener la nota de los ejercicios de mate de un determinado alumno. El origen de datos por tanto será la hoja de NotasMate. Se define que las columnas que el bot debe de responder son las de los ejercicios. También se define que sólo se desea un resultado, ya que sólo se pregunta por un alumno concreto. Esta característica es útil para mostrar un número determinado de resultados. En este caso cada alumno se identifica inequívocamente por el nombre y apellido. También se indica que se muestre el nombre de la columna (en este caso el nombre del ejercicio).

Dentro del intent también se definen las entidades de filtrado. En este caso se define la columna Alumno. El tipo de filtrado utilizado es LIKE, que tiene la misma funcionalidad que el like de SQL. Esto permite no tener que introducir nombre y apellido del alumno, si no introducir parcialmente su nombre a la hora de filtrar. En lugar de exigir al usuario escribir Adrian Arana, podrá preguntar por las notas de Adrian o de Arana. Por último se definen mensajes que permitan al usuario preguntar de una manera más amigable (o humana, ver Apartado 3.2.2) por las entidades que hacen falta.

"sheets": [
{			
"gSheetName": "NotasMate",			
"gSheetRange": "A1:O10"			
},			
{			
"gSheetName": "NotasFisica",			
"gSheetRange": "A1:L10"			
}			
],			
"chat": {			
"description": "Este bot ayuda a obtener las notas de un alumno.",			Hoja de cálculo
"greetings": "Hola! Soy notasbot y te ayudaré a obtener tus calificaciones. Prueba preguntandomo por tus notas de matemática o física",			
"intents": [
{			
"ID": "notasMateEjerciciosPorAlumno",			
"sourceSheet": "NotasMate",			
"response": {			
"outputColumn": "EjerEcuaciones, EjerTrigonometria, EjerFunciones1, EjerFunciones2, EjerDerivadas, EjerIntegrales",			
"numberOfResponses": 1,			
"notFoundMessage": "No se han encontrado trabajos para el alumno.",			
"showColumnName": true			
},			
"entities": [{			
"inputColumn": "Alumno",			
"mask": "LIKE",			
"entityMissingMessage": "Dime el nombre del alumno para obtener las notas de ejercicios de matemática. Como por ejemplo:",			
"entityNotFoundMessage": "No tengo información sobre ese alumno. Podrías preguntar por los ejercicios de mate de"			
}]			
},			

Figura 5.3: DSL con la definición de las Sheet y un intent a utilizar en el ejemplo de notas.

Algunas preguntas son más difíciles de formular debido a la naturaleza de los datos. Sin embargo, SheetChat ofrece algunos mecanismos que permiten resolver funciones matemáticas a la hora de realizar consultas sobre los datos. Se puede observar en la pregunta relacionada con obtener la nota ponderada. La nota ponderada de los ejercicios de Física el profesor lo tiene definido de una manera determinada. Esto no es una columna como tal dentro de la hoja de cálculo, si no que es una entidad derivada. En la Figura 5.4 se observa que dentro del response hay una expresión matemática que define qué es la nota ponderada de los ejercicios. De igual manera, en las respuestas se puede proporcionar un mensaje personalizado para mejorar la humanización del bot ofreciendo una respuesta más sencilla de interpretar por un humano.

```
{
  "ID": "fisicaPonderadaEjerciciosPorAlumno",
  "sourceSheet": "NotasFisica",
  "response": {
    "outputColumn": "EjerCinematica*0.05+EjerDinamica*0.15+EjerEnergia*0.1+EjerGravitatorio*0.1 AS Ponderada, Alumno",
    "numberOfResponses": 1,
    "notFoundMessage": "No se han encontrado resultados para el alumno.",
    "customResponseStructure": ["El alumno %s tiene una nota ponderada en ejercicios de fisica de %s sobre 4 puntos.", "Alumno", "Ponderada"]
  },
  "entities": [
    {
      "inputColumn": "Alumno",
      "mask": "LIKE",
      "entityMissingMessage": "¿De qué alumno quieres obtener la nota ponderada de ejercicios de fisica? Estos son algunos alumnos que tengo:",
      "entityNotFoundMessage": "Ups! no conozco a ese alumno. Podrias preguntar por la nota de ejercicios de fisica de"
    }
  ]
},
```

Figura 5.4: Definición de la pregunta para obtener la nota ponderada de los ejercicios de física del ejemplo de notas.

5.1.4. Ejemplo de uso del chatbot

A continuación se puede visualizar cómo sería un ejemplo de interacción entre un ser humano y el bot que se ha generado a partir del DSL de SheetChat en el ejemplo de las notas. En la Figura 5.5 se puede observar un intercambio de mensajes para obtener información respecto al alumno Alberto. En primer lugar se ha saludado al chatbot para que este proporcione su mensaje de bienvenida. Esta interacción no es necesaria si se conocen cuales son las características del chatbot. Posteriormente se le han preguntado por los resultados de alberto en matemática: si habia aprobado la evaluación continua y las notas que ha obtenido para conocer la causa de su suspenso. De igual manera se han realizado dos preguntas respecto a las notas de física.

En este caso el chatbot ha inferido cuales eran los intents del usuario y a su vez la entidad necesaria para cada una de ellas, en este caso el nombre del alumno. Si se obtienen de la frase de manera adecuada los intents y las entidades el chatbot proporciona la respuesta. En caso de que se haya inferido el intent pero no las entidades, preguntará por ellas (ofre-

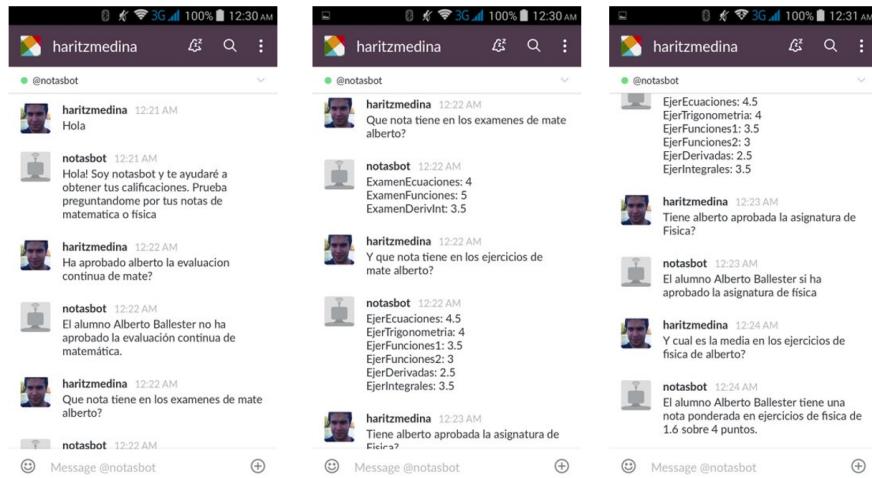


Figura 5.5: Interacciones del usuario a la hora de consultar las notas de Alberto.

ciendo sugerencias que ayuden al usuario). Esto sucede en la Figura 5.6 dado que no se le había proporcionado ningún nombre a la hora de preguntar por la media en los ejercicios de física. De igual manera, si se le proporciona un nombre no existente, el chatbot le seguirá ofreciendo sugerencias.



Figura 5.6: El chatbot recomienda algunos nombres en caso de que no se defina o se defina un nombre inexistente para hacer el filtrado en la hoja de cálculo.

5.2. Ejemplo 2: Calendario de sesiones en un congreso científico

Los congresos disponen de un calendario complejo, donde hay trabajos más o menos interesantes o relevantes con la rama de especialización que tiene el asistente. Es por ello que acudir a las sesiones más afines a tu trabajo es importante. Es interesante disponer de información in situ de cuales son las próximas charlas que habrá, dónde o quién las presenta. Sin embargo, los sitios webs rara vez están preparados para su navegación por el móvil o se pierde mucho tiempo en encontrar los eventos a los que se desea acudir. Es una información relevante y que se desea conocer en el momento.

Este ejemplo se centra en ofrecer una alternativa en forma de bot mediante una hoja de cálculo creada a partir del horario de la conferencia WISE 2015².

5.2.1. Hoja de cálculo con los datos

Como se ha mencionado previamente la hoja de cálculo es creada a partir de una tabla con el programa disponible en el sitio web. En la Figura 5.7 se puede observar el calendario de las diferentes sesiones que están presentes en el congreso. Cada sesión tiene asociada un día y un slot (una franja horaria) donde se exponen trabajos. En algunos de los slots se produce un solapamiento de múltiples sesiones, por lo que es interesante que el usuario conozca cuales son las charlas para elegir la más interesante.

5.2.2. Análisis de preguntas

Como se ha comentado previamente, el usuario del bot tiene como objetivo poder conocer el programa del congreso durante la estancia en él. Algunas de las preguntas que le pueden surgir durante el evento están recogidas a continuación:

- **¿Cuáles son las charlas en un horario concreto, es decir, qué eventos hay en esa franja horaria?** De los eventos que existan en esa franja horaria, el asistente al congreso podría decantarse por la que más interesante le resulte.

²Programa del WISE 2015 (Web Information System Engineering) <http://www4.cis.fiu.edu/wise2015/@schema.html>

	A Day	B Slot	C Session1	D Session2	E Session3	F Session4
1	Sunday, November 1st	7:30-9:00	Room: Paris Breakfast			
2	Sunday, November 1st	9:00-10:00	Chair: S. Iyengar Keynote 1: Hardware Accelerators for the Web			
3	Sunday, November 1st	10:30-12:00	\$1:Big Data Techniques	Room: Cannes Chair: Dech-Banchai S2:Social Issues of Web Applications	Room: Marseille Chair: Yunchun Zhang S3:Data/Hidden Web	Room: Paris Chair: Daren Chen & William Wei Song S51:Data Quality and Trust in Big Data
4	Sunday, November 1st	12:00-13:30	Room: Nice Chair: Jie Shao Lunch	Room: Cannes Chair: Dech-Banchai S2:Social Issues of Web Applications	Room: Marseille Chair: Yunchun Zhang S3:Data/Hidden Web	Room: Paris Chair: Daren Chen & William Wei Song S51:Data Quality and Trust in Big Data
5	Sunday, November 1st	13:30-15:00	\$4:Big Data and Mining Techniques	Room: Cannes Chair: Marek Buszlejewicz S5:Social Network Computing	Room: Cannes Chair: Frederique Lafosse S7:Social Web	Room: Paris Chair: Shaohong Zhang S52:Data Quality and Trust in Big Data
6	Sunday, November 1st	15:30-17:00	Room: Nice Chair: Naoya Takada S6:Big Data Annotations	Room: Cannes Chair: Frederique Lafosse S7:Social Web	Room: Marseille Chair: Eduardo B. Fernandez S8:Secure Web Systems Architectures Using Secu	Room: Paris Chair: Sriji Yella S53:Data Quality and Trust in Big Data
7	Sunday, November 1st	17:30-18:30	Room: Paris Reception			
8	Sunday, November 1st	18:30-19:00	Room: Paris Break			
9	Monday, November 2nd	7:30-9:00	Room: Monte Carlo Chair: Tao Li ata Science Research at FIU (S. S. Iyengar, Nagaiah Rishi, Steve Luis, and Radu Jianu)			
10	Monday, November 2nd	9:00-10:00	Room: Nice Chair: Ming Tang S8:Web Data Integration and Mashups	Room: Cannes Chair: Shu-Ching Chen S9:Web Privacy and Security	Room: Monte Carlo Chair: Bishik Naras S10:Integration of Web with IoT	
11	Monday, November 2nd	10:30-12:00	Room: Paris Lunch			
12	Monday, November 2nd	12:00-13:30	Room: Nice Chair: Junhu Wang S11:Web Data Models	Room: Cannes Chair: Dimitris Therasios S12:Web Information Retrieval	Room: Monte Carlo Chair: Binyamin Ho Karng S13:Web-based recommendations	
13	Monday, November 2nd	13:30-15:00	Room: Nice Chair: Toshiaki Amagasa S14:Linked Open Data	Room: Cannes Chair: Wojciech Cellary S15:Web Information Extraction	Room: Monte Carlo Chair: Mark A. Finlayson T1:The WordNet Database: Form, Function, and Use	
14	Monday, November 2nd	15:30-17:00				
15	Monday, November 2nd	17:00-17:30	Room: Monte Carlo Ballroom I&II			
16				

Figura 5.7: Hoja de cálculo con el programa del WISE 2015.

- **¿Qué eventos hay relacionados con un tema (o topic) en particular?** De esta manera el usuario sabrá con una simple pregunta en qué horarios están programadas charlas interesantes para él.
- **¿Quiero conocer en qué sesiones participa una persona?** Los investigadores conocen el trabajo de algunas personas y puede interesarle saber si estas personas participan como expertas en la materia de una determinada sesión.

5.2.3. Definicion del DSL

El diseño del chatbot en este ejemplo constará de una única hoja de cálculo y de dos intents. A pesar de que previamente se han definido 3 cuestiones, sólo hace falta definir dos intents debido a que la obtención de las sesiones relacionadas con un tema en concreto o presidido por un chairman concreto se pueden unir en una. Esto es debido a la naturaleza de los datos. Una columna sesión tiene como valor un string que incluye ambas informaciones. A la hora de realizar el filtrado, se filtra por valor de la columna que se desee. Visto en un ejemplo, observando la sesión 1 del domingo en el slot 10:30-12:00, es lo mismo filtrar por el topic *Hardware Accelerators for the Web* por la habitación *Paris* o el chairman *Iyengar*, se obtendrá como resultado el dia y slot de esa sesión.

Por otro lado, en este ejemplo, surge la novedad de filtrar por más de una columna. En la Figura 5.8 se observa remarcado en color azul el entity definido. En este caso se han definido cuatro columnas que actuarán como filtro. A la hora de preguntar por el topic o chairman, el chatbot buscará si se dan positivos en cualquiera de las cuatro columnas.

En caso de que el valor introducido por el usuario dé positivo por alguna de las cuatro columnas, se le mostrará el día y la hora (o slot) de ese evento.

```

"sheets": [
  {
    "gSheetName": "Schedules",
    "gSheetRange": "A1:F22"
  }
],
"chat": {
  "description": "WISE conference sessions related bot.",
  "greetings": "This bot helps you to retrieve information related with the WISE 2015 conference. Try typing 'Show sessions'.",
  "intents": [
    {
      "ID": "retrieveSlotByTopic",
      "sourceSheet": "Schedules",
      "response": {
        "outputColumn": "Day, Slot",
        "notFoundMessage": "Events not found",
        "numberOfResponses": 5,
        "customResponseStructure": ["An event at $s on $s is scheduled.", "Slot", "Day"]
      },
      "entities": [
        {
          "inputColumn": ["Session1", "Session2", "Session3", "Session4"],
          "mask": "LIKE",
          "entityMissingMessage": "Search for a topic, room or chairman, for example:",
          "entityNotFoundMessage": "Topic, room or chairman not found, try another one:"
        }
      ]
    }
  ]
},
  
```

Figura 5.8: Implementación del chatbot de la conferencia. Cabe destacar remarcado en azul el intent con input multicolumna.

5.2.4. Ejemplo de uso del chatbot

A continuación se presenta un ejemplo de interacción real con el chatbot desarrollado para el programa de las conferencias. Para la obtención de las sesiones en una franja horaria concreta, tal y como se ha mencionado en la implementación del chatbot, se requiere de dos parámetros (el día y el slot). En la Figura 5.9 el Chatbot guía al usuario mediante preguntas para obtener la información suficiente de a qué slot se refiere el usuario. En este caso realiza una pregunta respecto a los días y otra respecto al slot. Si el usuario hubiese añadido esa información a la hora de hacer la pregunta, el chatbot podría haber inferido estas entidades y ahorrarse los pasos de preguntarlo. Esto es lo que sucede en la pregunta de la imagen derecha, donde el usuario ha proporcionado un criterio de búsqueda.

5.3. Ejemplo 3: Búsqueda de restaurantes de Tripadvisor

En este tercer caso de estudio se muestra además de un nuevo contexto de uso, el uso de un origen de datos web. En la actualidad la mayoría de información se puede recabar en

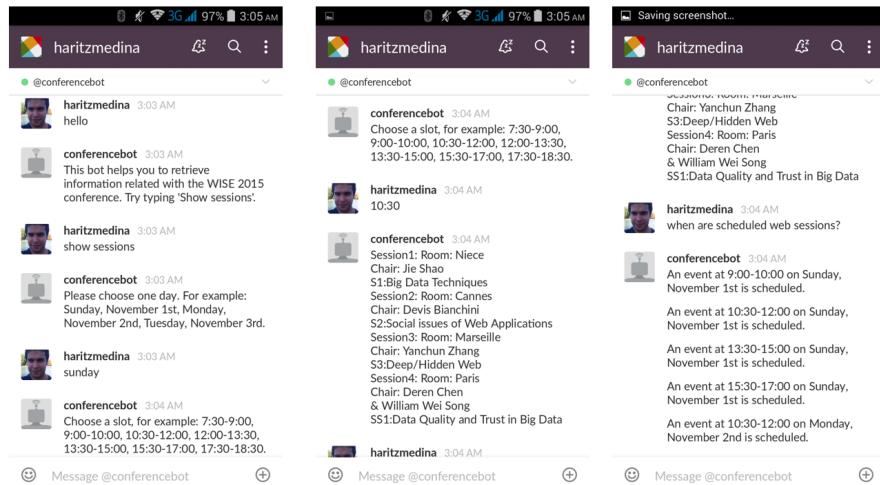


Figura 5.9: Interacción del usuario a la hora de preguntar por los eventos en una hora concreta (Imagen izquierda y central) y consulta respecto a un topic concreto (Imagen derecha).

la red, sin embargo esta no suele tener una estructura utilizada en un ámbito general como son las tablas de bases de datos o las hojas de cálculo.

En este ejemplo se mostrará cómo se pueden realizar búsquedas en restaurantes extraídos del sitio web Tripadvisor. Tripadvisor es un sitio web de opinión sobre hoteles, restaurantes, y otros lugares de ocio. En su web se pueden filtrar los resultados en base a localización (como una ciudad o provincia), características del sitio (número de estrellas de un hotel o tipo de comida de un restaurante) y otros muchos aspectos. Concretamente en este ejemplo se ha realizado una búsqueda de restaurantes en Miami (Estados Unidos), que es lo que se puede observar en la Figura 5.10.

La idea para este ejemplo es demostrar que estos datos en la web pueden ser extraídos a una hoja de cálculo y generar un chatbot mediante la herramienta SheetChat.

5.3.1. Hoja de cálculo con los datos

Como se ha mencionado al comiendo de este caso de estudio, el objetivo es poder representar estos datos en forma de hoja de cálculo. Para ello en la web existen múltiples herramientas de web scraping (extracción de información de la web). A pesar de que la web de tripadvisor muestre los restaurantes en un formato más amigable para el ser humano, si que todos los restaurantes comparten una estructura similar. En la Figura 5.10 se observa que todos los restaurantes tienen un hyperlink con el nombre del restaurante donde pinchando saldría la ficha del restaurante en cuestión. Cada restaurante tiene una

The screenshot shows the TripAdvisor search results for restaurants in Miami. At the top, there are several category filters: 'Populares' (highlighted in green), 'Cenas memorables', 'Desayunos rápidos', 'Comidas informales', and 'Cafeterías y cafés'. Below these are sorting options: 'Ordenar por:' (Clasificación selected), 'Nombre', 'Precio (menor a mayor)', and 'Precio (mayor a menor)'. A map of Miami with a pin is on the left, and a sidebar on the left lists filters for 'Restaurantes', 'Reservas', 'Cocinas y platos', and 'Comidas'. The main list displays four restaurants:

- Crust**: 1 de 3.334 Restaurantes en Miami. Rating: 4.5 stars (247 reviews). 'Excelente!!' 23/08/2016. 'Exquisito' 05/06/2016. Precio: 6 € - 20 €. Cocina: Italiana, Pizza, Menú para celíacos.
- Vero Italian Restaurant**: 2 de 3.334 Restaurantes en Miami. Rating: 4.5 stars (457 reviews). 'excellent' 27/07/2016. 'Comida fresca, con gusto a Italia' 10/07/2016. Precio: 6 € - 20 €. Cocina: Italiana, Menú para celíacos.
- CVI.CHE 105**: 3 de 3.334 Restaurantes en Miami. Rating: 4.5 stars (2.001 reviews). 'Almuerzo en paroja' 12/09/2016. 'MUST EN MIAMI AREA' 12/09/2016. Precio: 10 € - 27 €. Cocina: Peruana, Latina, Marisco, Sopas, Sudamericana.
- Whole Foods Market**: 4 de 3.334 Restaurantes en Miami. Rating: 4.5 stars (284 reviews). 'Almorzo' 08/09/2016. 'Comprando' 08/09/2016. Precio: 10 € - 27 €. Cocina: Latinoamericana.

Figura 5.10: Búsqueda de restaurantes de Miami en Tripadvisor

imagen asociada, un número de opiniones, dos breves opiniones, el precio promedio del menú del restaurante y el tipo de comida. Precisamente, el encontrar patrones y extraer datos es el objetivo de herramientas como Import.io³.

El funcionamiento de import.io permite extraer una tabla con los resultados de la búsqueda de restaurantes de Miami. En el Anexo A se puede profundizar en el funcionamiento de la herramienta. Para el ejemplo, simplemente la hoja de cálculo que obtenemos es la presentada en la Figura 5.11. Mediante import.io se han extraído 400 restaurantes en Miami. En la tabla se pueden observar las columnas sobre las que podremos realizar preguntas posteriormente al bot, nombre del restaurante (Name), sitio web con la ficha de Tripadvisor (url), rango de precios de los menús (RangePrice) o tipo de comida (Cuisines).

5.3.2. Análisis de preguntas

Tras observar los datos extraídos de Tripadvisor, el usuario podría hacer diferentes preguntas. En este caso se ha decidido que pueden ser interesantes las que se presentan a continuación:

³Import.io extracción de datos tabulares a partir de sitios web.

Figura 5.11: Hoja de cálculo con restaurantes de Miami extraídos de Tripadvisor.

- **¿Me podrías ayudar a buscar restaurantes con comida italiana?** Al igual que se puede preguntar por italiana, se puede preguntar por cualquier tipo de comida que se haya extraído de los restaurantes situados en Miami.
 - **¿Qué restaurantes vegetarianos hay por menos de 45 dolares?** Similar a la pregunta anterior, con la condición de precio máximo, ya que son dos aspectos que se suelen mirar frecuentemente a la hora de buscar un restaurante, si gusta la comida y el precio.
 - **¿Qué opinión hay sobre un restaurante concreto?** Mediante la opinión de otros usuarios extraída del sitio web, el usuario del Chatbot puede deducir si un restaurante es bueno o no.

5.3.3. Definicion del DSL

Para este ejemplo se trabajará la definición de las interacciones para las dos primeras cuestiones previamente definidas. En la Figura 5.12 se ve la definición para la primera de las preguntas, dónde la entidad de entrada es la columna *Cuisines* (tipo de cocina) y en la salida se mostrará un mensaje personalizado con el nombre, el tipo de comida, el rango de precio y un link para abrir la ficha en Tripadvisor.

Se ha observado que el resultado que ofrece el chatbot (junto con la previsualización de links de la plataforma Slack) hace que el chatbot al ofrecer la respuesta sea muy verbose. Por esa razón, se ha decidido que se limite a tres el número de respuestas máximas que va a recibir el usuario.

De igual manera, en la Figura 5.13 se muestra la definición utilizando el DSL de Sheet-Chat donde el usuario hará peticiones respecto a un precio máximo y un tipo de comida concretos, es decir, hay dos entidades participantes a la hora de realizar el filtrado entre los 400 restaurantes. Como novedad cabe destacar que el valor del campo Mask en lugar de ser LIKE (que sirve para comparar strings), se utiliza el símbolo < para designar que es

```

"intents": [
  {
    "ID": "restaurantInfoPerCuisine",
    "sourceSheet": "Restaurants",
    "response": {
      "outputColumn": "Name, RangePrice, Cuisines, Name_link",
      "numberOfResponses": 3,
      "customResponseStructure": ["The %s restaurant has %s food and it cost between %s. More info via: %s",
        "Name", "Cuisines", "RangePrice", "Name_link"]
    },
    "entities": [
      {
        "inputColumn": "Cuisines",
        "mask": "LIKE",
        "entityMissingMessage": "Which type of cuisine would you like? For example:",
        "entityNotFoundMessage": "Cuisine style not found, try with:"
      }
    ]
  },
]

```

Figura 5.12: DSL de SheetChat que describe el intent para la búsqueda por tipo de comida de los restaurantes de Tripadvisor.

un valor numérico y que el filtro rechazará los resultados mayores que el precio máximo delimitado por el usuario a la hora de interaccionar con el Chatbot, tal y como se muestra en el Apartado 5.3.4.

```

{
  "ID": "restaurantsCheaperThanPerCuisine",
  "sourceSheet": "Restaurants",
  "response": {
    "outputColumn": "Name, RangePrice, Cuisines, Name_link",
    "numberOfResponses": 3,
    "customResponseStructure": ["The %s restaurant has %s food and it cost between %s. More info via: %s",
      "Name", "Cuisines", "RangePrice", "Name_link"]
  },
  "entities": [
    {
      "inputColumn": "Cuisines",
      "mask": "LIKE",
      "entityMissingMessage": "Which type of cuisine would you like? For example:",
      "entityNotFoundMessage": "Cuisine style not found, try with:"
    },
    {
      "inputColumn": "MaxPrice",
      "mask": "<",
      "entityMissingMessage": "Set maximum price. For example:",
      "entityNotFoundMessage": "Not found cheaper than that price. Can you set a higher price as: "
    }
  ]
}

```

Figura 5.13: DSL de SheetChat que describe el intent para la búsqueda por precio máximo (resaltado en azul) y tipo de comida de los restaurantes de Tripadvisor.

5.3.4. Ejemplo de uso del chatbot

Tras el diseño del Chatbot mediante el DSL de SheetChat a continuación se presenta cómo sería la experiencia de uso de este chatbot por parte de un usuario.

En la Figura 5.14 el usuario en primer lugar saluda al Chatbot y este le ofrece alguna de

sus funcionalidades por si acaso el usuario no se acuerda de qué trata este chatbot o qué funcionalidades tenía. Posteriormente se dispone a preguntar por restaurantes italianos. El chatbot le responde con tres resultados, el número máximo de resultados que se indicó en el diseño del chatbot (ver Apartado 5.3.3).

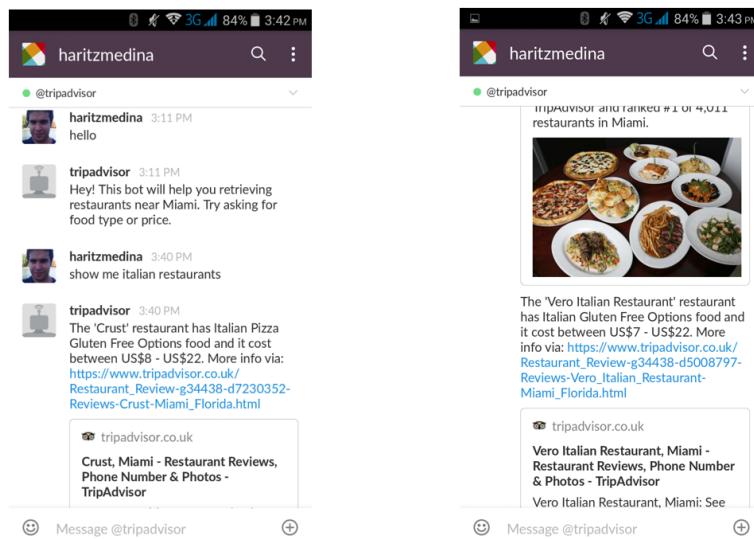


Figura 5.14: Interacción entre el usuario y el chatbot que recomienda restaurantes en base a un tipo de cocina.

En la interacción de la Figura 5.15 se observa que al usuario las recomendaciones de los restaurantes italianos le ha resultado cara. Por lo tanto decide preguntar por restaurantes con precio menor a 15 dolares. El chatbot ha detectado que el usuario está preguntando por restaurantes con un precio máximo y un tipo de cocina determinado. El chatbot restringe las respuestas en base a ese nuevo filtro y muestra el único restaurante que cumple esas características. En caso de que no hubiese ningún restaurante que cumpliese ambos criterios preguntaría por las entidades nuevamente.

De esta manera, mediante el uso de un chatbot, el usuario evita tener que aplicar filtros en el sitio web de Tripadvisor que en una configuración móvil suele resultar en líneas generales bastante farragoso.

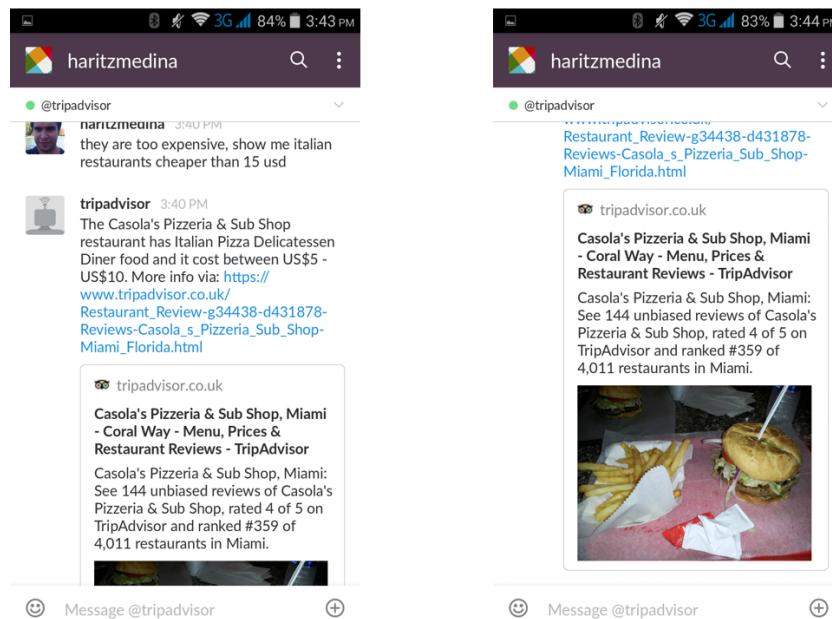


Figura 5.15: El bot de Tripadvisor recomienda restaurantes italianos con precio menor a 15 dólares por petición del usuario.

6. CAPÍTULO

Trabajo Futuro y Conclusiones

Este capítulo se divide en dos apartados. En el Apartado 6.1 se hablará del trabajo futuro y de las mejoras que se puedan aplicar a este trabajo. En el Apartado 6.2 se extraerán unas conclusiones obtenidas durante el desarrollo del trabajo.

6.1. Trabajo futuro

El trabajo presentado en este proyecto sirve como prueba conceptual para demostrar que la solución al problema detectado puede ser factible. Está claro que un componente que demostrará, si además de ser factible, es útil es la evaluación de la propia herramienta con usuarios de hojas de cálculo. Es por ello que resulta difícil de vaticinar cuales son los siguientes pasos a dar en el desarrollo de esta idea, ya que irán claramente asociada a esa evaluación y feedback que puedan proporcionar los usuarios.

Sin embargo, a lo largo del desarrollo se han detectado algunos aspectos de mejora en lo que a la herramienta se refiere. En concreto se han dividido en los siguientes aspectos: enriquecimiento del diseño del DSL, usabilidad de los chatbot generados, el ecosistema de plataformas utilizado para su desarrollo y la actualización de los datos de las hojas de cálculo.

6.1.1. Diseño del DSL

El DSL diseñado permite la creación de consultas abstrayendo al usuario del conocimiento de un lenguaje como es SQL. A pesar de que esto resulta positivo, se ha detectado que el actual diseño del DSL no permite realizar consultas que puede que sean bastante habituales, especialmente sobre datos tabulares extraídos de sitios web de manera automática.

De igual manera, el lenguaje de tipado JSON, a pesar de que sea legible por el humano y fácilmente generable, no es un lenguaje con el que el usuario se sienta excesivamente cómodo. Esto es debido a que el usuario final de esta herramienta es posible que no conozca JSON. A pesar de que no hay una evaluación de por medio, puede resultar más adecuado utilizar hojas de cálculo también para definir los chatbots haciéndolo más intuitivo para los usuarios.

Como se ha comentado a lo largo del trabajo, la idea es que una herramienta, posiblemente gráfica, sea capaz de ayudar (o sustituir) en la elaboración de este esquema para definir las propiedades necesarias para la creación de un Sheetchat. En esta herramienta será fundamental ver cuánto se requiere para la realización de un chatbot que cubra las necesidades de los usuarios.

6.1.2. Usabilidad de los Chatbot generados

El sistema de sugerencias desarrollado es bastante espartano, ya que se utiliza una función de aleatorización para ir mostrando diferentes sugerencias cada vez que el usuario introduzca entidades que no existen. Sin embargo puede ser que el usuario esté cometiendo un error tipográfico y que el bot le esté indicando constantemente que esa entidad no existe, pero no que se trate de un error tipográfico. En este caso, la experiencia de usuario mejoraría considerablemente utilizando lo que en los buscadores web se denomina el "quizá quiso decir...".

A pesar de que se ha trabajado en aspectos de humanización, hay que ver si son realmente útiles o suficientes. También sería conveniente ver si hay más patrones de diseño de chatbots de manera que se pueda equiparar los chatbots que hay en el mercado con los que se generan con esta herramienta. Aplicando un patrón de diseño se obtienen chatbots con funcionalidades similares, lo que simplificaría el proceso de aprendizaje para usar nuevos chatbot.

6.1.3. Ecosistema de plataformas

Sería conveniente realizar un estudio en mayor profundidad de las plataformas y librerías para desarrollo y despliegue de agentes conversacionales.

Para este trabajo se ha utilizado Botkit, que es una librería muy sencilla y muy potente junto con el middleware de Wit.ai. Sin embargo, su desarrollo no es demasiado continuado y solo ofrece soporte para tres plataformas como son Slack, Facebook Messenger y Twilio.

Entre las alternativas, Microsoft Bot Framework está cogiendo mucha fuerza debido a que es capaz de generar bots para muchas más plataformas. Sería interesante también ver qué servicios para proveer chatbots son las más usadas. La tendencia actual es que Facebook Messenger y Slack son de las más utilizadas ¹.

6.1.4. Hojas de cálculo

El principal handicap de las hojas de cálculo que tienen datos extraídos de sitios web mediante herramientas automatizadas es que debe el usuario actualizarlas de manera manual cada cierto tiempo.

En la actualidad para solventar ese problema se ha detectado la existencia de dos herramientas que habría que ver si pueden ser interesantes implantarlas en el ecosistema de SheetChat.

Las propias hojas de cálculo de google tienen una funcionalidad que es exportar una tabla html de un sitio web ² y que se actualizará de manera automática. A pesar de que en la mayoría de los casos no trabaje bien, ya que depende de la correcta estructuración de la tabla html, puede servir para datos que están bien estructurados.

Además de los sitios web html como fuente de información, los servicios web RESTful o SOAP también contienen una cantidad de información fácilmente accesible. Existe una solución que permite extraer hojas de cálculo dado un endpoint de un servicio RESTful [Chang and Myers, 2014].

¹Encuesta a la comunidad de desarrolladores de chatbots: <http://venturebeat.com/2016/09/14/early-results-of-bot-community-survey-show-messenger-and-slack-as-the-developers-top-platforms/>

²Tutorial de uso de la función de Google Sheets importhtml: <https://mashe.hawksey.info/2012/09/reshaping-importhtml-data-in-google-spreadsheet-using-query-and transpose-formula/>

6.2. Conclusiones

El desarrollo de este trabajo ha permitido ofrecer una respuesta a la problemática de cómo acceder a datos de una hoja de cálculo en un entorno móvil.

Lo novedoso de la solución es que utiliza el lenguaje natural como interfaz para realizar consultas a una hoja de cálculo [Flood, 2010]. La generación de consultas SQL a partir de las definiciones de Intents y Entidades proporciona una abstracción sobre el propio lenguaje SQL. Para gente que no tenga conocimientos de SQL poder consultar en bases de datos utilizando el lenguaje natural puede resultar muy ventajoso. Sin embargo, a la vez que la generación de consultas SQL a partir de Intents y Entidades tiene sus ventajas, la limitación que tiene también queda patente.

Los aspectos de humanización en un chatbot son fundamentales, incluso siendo para de autoconsumo, en el que el usuario/desarrollador conozca todos los aspectos del chatbot. Estos aspectos de humanización presentan una interfaz más cercana al usuario y mejoran la experiencia de usuario reduciendo costes temporales especialmente en el proceso de aprendizaje del uso del chatbot.

Aplicando patrones de diseño como el de obligar a definir un mensaje de bienvenida se resuelven muchos problemas. Esta cuestión, aunque parezca increíble, hay gran cantidad de bots de terceros en la web que aún no la resuelven. Esto provoca que un usuario descarte la posibilidad de usar estos chatbots. La razón principal es que el usuario no puede saber ni tan siquiera para qué fue diseñado el bot.

Anexos

A. ANEXO

Import.io: Extracción de datos tabulares a partir de la web

Import.io es una herramienta que permite extraer datos de un sitio web de manera sencilla y convertirlos a datos en forma de tabla, concretamente a ficheros CSV. A estos artefactos de extracción de datos se les denomina extractores. Un extractor es representado por un número de URLs o direcciones web que tendrá como origen de datos y el método de extracción de datos, es decir, qué elementos de un sitio web corresponden a qué columna de la tabla que se va a extraer.

Import.io es capaz de inferir todas las URLs que pueden ser relevantes proporcionándole algunos ejemplos si todas estas comparten alguna característica o estructura similar. En la Figura A.1 se puede observar cómo se han generado URLs con las diferentes páginas de búsqueda de restaurantes en Miami en el sitio web Tripadvisor¹

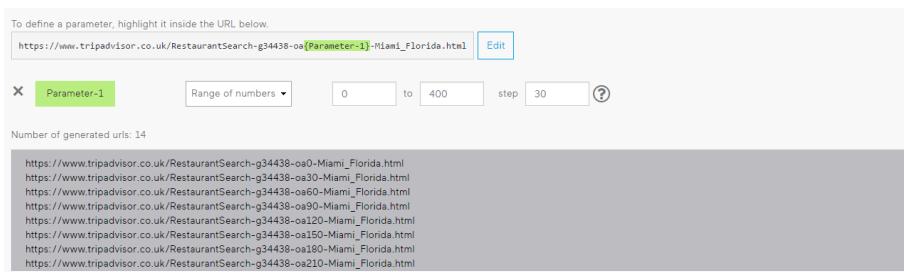


Figura A.1: Generador de URLs en las que extraer datos para import.io.

¹Búsqueda de restaurantes en Miami en Tripadvisor: https://www.tripadvisor.co.uk/RestaurantSearch-g34438-oa30-Miami_Florida.html

Import.io mediante el uso de heurísticos define qué datos siguen un patrón concreto dentro de un sitio web y los marca como candidatos a formar parte de la tabla resultante. Sin embargo, una característica muy interesante es poder definir las columnas a qué elementos del sitio web hacen referencia. En la Figura A.2 está en el panel superior marcada la columna de *Image* y te marca el elemento HTML con el que hacer el matching.

#	Image	Name	Ranking	NumberOfReviews	Review1	Review1Date	Review2	Review2Date	RangePrice	Cuisines
1		Crust	1 de 3.329 Restaurantes en Miami	4.5	236 opiniones	23/09/2016	Evaluation: 05/08/2016	Precio: 8 \$ - 22 \$	Mapa Fotos de los visitantes (102)	Cocina: Italiana Reservar
2		Vero Italian Restaurant	2 de 3.329 Restaurantes en Miami	4.5	456 opiniones	Excelente! 27/07/2016	Comida fresca, con gusto a Italia! 10/07/2016	Precio: 8 \$ - 22 \$	Mapa Fotos de los visitantes (115)	Cocina: Italiana Reservar
3		CVICHE 106	3 de 3.329 Restaurantes en Miami	4.5	972 opiniones	Deliciosa mariscada en envolver! 03/09/2016	Especialidad: Especialidad 03/09/2016	Precio: 11 \$ - 30 \$	Mapa Fotos de los visitantes (559)	Cocina: Peruana Latina Marisco Sopas Sudamericana Reservar
4		Whole Foods Market	4 de 3.329 Restaurantes en Miami	4.5	276 opiniones	Un super mercado! 31/08/2016				

Figura A.2: Pantalla de edición de los elementos HTML a extraer en la tabla generada por import.io.

Como última característica, cabe destacar que import.io, en su versión de pago, permite la actualización automática de los cambios que se produjesen en la hoja de cálculo resultante. Es decir, en caso de que el sitio web cambiase, algo muy frecuente en sitios web de esta índole, import.io se encargaría de escanear y extraer la tabla con el formato que se haya preestablecido.

Al igual que existe import.io, existen otras herramientas para la extracción de datos en forma tabular:

- **importhtml de Google Spreadsheets**², menos potentes, pero mucho más sencilla de utilizar y con contenido actualizable.
- **Gneiss Spreadsheet** permite definir consultas y extraer datos en forma de tabla de un servicio web RESTful API [Chang and Myers, 2014] con la funcionalidad de auto-actualización de los datos.

²Extracción de datos con importhtml de Google Spreadsheets: <https://mashe.hawksey.info/2012/09/reshaping-importhtml-data-in-google-spreadsheet-using-query-and-transpose-formula/>

B. ANEXO

Wit.ai: Procesamiento del lenguaje natural orientado a bots conversacionales

Wit.ai se define como una herramienta de procesamiento del lenguaje, ya sea para texto como para voz. El objetivo de esta herramienta es funcionar de nexo entre el lenguaje que comprenden los humanos y las funcionalidades que dispone un sistema software, como lo puede ser un chatbot. Trabaja con diferentes lenguajes como el inglés o el castellano (en fase de desarrollo actualmente, aunque funciona relativamente bien). La principal virtud es que con poco trabajo Wit.ai empieza a funcionar como es esperado. Wit.ai funciona a base de entrenamiento, es decir, el desarrollador de la aplicación debe de ir introduciendo frases que el usuario utilizaría para hacer peticiones al sistema. A medida de que disponga de más frases más preciso será encontrando las intenciones (Intents) o entidades que los usuarios hayan introducido.

Respecto al trabajo referente a SheetChat, se ha delegado la tarea de desambiguación de Intents y reconocimiento de entidades a Wit.ai. Wit.ai permite tanto introducir frases de manera manual o esperar a recibir frases de los usuarios e ir corrigiendo desambiguaciones en las que se haya podido equivocar Wit.ai para mejorar su sistema de reconocimiento. En la Figura B.1 se pueden observar dos mensajes introducidos por alguno de sus usuarios. El sistema ha reconocido tanto la entidad alumno como el Intent al que se refería el usuario. Sin embargo en la Figura B.2 Wit.ai ha sido incapaz de obtener toda la información necesaria para resolver la pregunta del usuario. En la imagen superior ha reconocido erróneamente el Intent y no ha visto la entidad alumno (lo que indica que requiere más entrenamiento con frases de ese tipo). En la imagen inferior si ha sido capaz de reconocer

adecuadamente el Intent, pero no ha podido obtener la entidad alumno, por lo que tendrá que preguntar por ella para dar respuesta al Intent.

Figura B.1: Wit.ai infiriendo de dos frases cuales son los Intents correspondientes para cada uno de ellos y cuál es la entidad Alumno.

Figura B.2: En la imagen de arriba Witai infiere erroneamente el Intent y no reconoce la entidad. En la inferior el usuario no ha proporcionado ninguna entidad, por lo que el chat tendrá que preguntar por ella.

Como se ha mencionado previamente, Wit.ai funciona en base a una confianza en las detecciones que ha hecho. En SheetChat se exige que el grado de confianza ha de ser mayor que 0.5 sobre 1 para que una inferencia se de por válida. En la Figura B.3 se puede observar como el grado de confianza de que *María* es un alumno es de 0.99 y que el Intent es *fisicaPonderadaPorEjercicio* únicamente de 0.71. Si se le ha entrenado adecuadamente, es muy probable que *María* sea un alumno y, con una confianza menor, que lo que se quiere es obtener la media ponderada de los ejercicios de física.

Al igual que existe Wit.ai existen numerosos servicios de este tipo en Internet, como

```
{  
    "entities": {  
        "alumno": [  
            {  
                "confidence": 0.996138881423843,  
                "start": 54,  
                "end": 59,  
                "body": "maria",  
                "value": {  
                    "value": "Maria"  
                },  
                "entity": "alumno"  
            }  
        ],  
        "intent": [  
            {  
                "confidence": 0.71759894323797,  
                "value": "fisicaPonderadaEjerciciosPorAlumno",  
                "entity": "intent"  
            }  
        ]  
    },  
    "confidence": null,  
    "_text": "me puedes dar la media de los ejercicios de fisica de maria?",  
    "intent": "default_intent",  
    "intent_id": "57d6b4ab-bf5c-4ee0-9245-60727fb5b2bf"  
}
```

Figura B.3: Respuesta del servicio de wit.ai cuando se le proporciona una frase. Remarcado en morado está el grado de confianza referente a la entidad alumno que ha detectado, de azul la del Intent.

son Luis.ai¹ (Language Understanding Intelligent Service) desarrollado por Microsoft o Api.ai² especializado a conversaciones con interfaz de voz.

¹Sitio web de luis.ai: <https://www.luis.ai/>

²Sitio web de api.ai: <https://api.ai/>

Bibliografía

- [Chan, 2016] Chan, C. (2016). Why does your chatbot suck? <https://uxmag.com/articles/why-does-your-chatbot-suck>. Accedido: 2016-09-13.
- [Chang and Myers, 2014] Chang, K. S.-P. and Myers, B. A. (2014). A spreadsheet model for using web service data. *Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on*, pages 169–176.
- [Dans, 2016] Dans, E. (2016). El bot como interfaz preferida. <https://www.enriquedans.com/2016/06/el-bot-como-interfaz-preferida.html>. Accedido: 2016-09-11.
- [Ferrara et al., 2016] Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, 59(7).
- [Flood, 2010] Flood, D. (2010). Natural language processing for spreadsheet information retrieval. *Natural Language Processing and Information Systems*, 5723:309–312.
- [Flood et al., 2011a] Flood, D., Harrison, R., and Daid, K. M. (2011a). Spreadsheets on the Move : An Evaluation of Mobile Spreadsheets.
- [Flood et al., 2011b] Flood, D., Harrison, R., and Nosseir, A. (2011b). Useful but tedious: An evaluation of mobile spreadsheets. *Ppig.Org*.
- [Investitech, 2015] Investitech (2015). 27 microsoft excel experts predict the future of excel in business intelligence. <http://www.investintech.com/resources/blog/archives/5718-experts-predict-the-future-of-excel-in-business-intelligence.html>. Accedido: 2016-09-13.

- [Montag et al., 2015] Montag, C., Błaszkiewicz, K., Sariyska, R., Lachmann, B., Andone, I., Trendafilov, B., Eibes, M., and Markowitz, A. (2015). Smartphone usage in the 21st century: who is active on WhatsApp? *BMC research notes*, 8:331.
- [Philips, 2016] Philips, G. (2016). Excel vs. access – can a spreadsheet replace a database? <http://www.makeuseof.com/tag/excel-vs-access-can-spreadsheet-replace-database/>. Accedido: 2016-09-11.
- [Wagner, 2016] Wagner, K. (2016). Bots, explained. <http://www.recode.net/2016/4/11/11586022/what-are-bots>. Accedido: 2016-09-11.
- [Wang et al., 2016] Wang, D., Xiang, Z., and Fesenmaier, D. R. (2016). Smartphone Use in Everyday Life and Travel. *Journal of Travel Research*, 55(1):52–63.
- [Wright, 2016] Wright, D. (2016). Are digital assistants and chat bots the same thing? <https://www.linkedin.com/pulse/digital-assistants-chat-bots-same-thing-david-wright>. Accedido: 2016-09-13.
- [Zhang and Adipat, 2005] Zhang, D. and Adipat, B. (2005). Challenges, methodologies, and issues in the usability testing of mobile applications. *International Journal of Human-Computer Interaction*, 18(3):293–308.

Agradecimientos

En primer lugar, me gustaría agradecer a mi director de proyecto Óscar Díaz por confiar en mi para la realización de este TFM. Ha sabido guiarme correctamente a lo largo de la elaboración del trabajo, atendiendo a las consultas con un trato humano excepcional.

Asimismo, me gustaría agradecer a mis compañeros de trabajo dentro del grupo Onekin. Estos no solo me han brindado con su ayuda cuando la he necesitado, si no que también me han abierto las puertas de su grupo de par en par haciéndome sentir uno más del equipo.

De igual manera, quiero agradecer a mis compañeros de clase, de los que en poco tiempo he adquirido mucho conocimiento que me será útil para desarrollarme tanto profesionalmente como en el ámbito personal.

Por último, agradecer a mi familia, amigos, a Ainhoa y a mi entorno más cercano que una vez más han confiado en mi y me han respaldado en todo lo que he necesitado.