UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

título del TFG

Presentado por Álvaro Delgado Pascual
en Universidad de Burgos — 18 de mayo de 2020
Tutor: D. Raúl Marticorena Sánchez

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. Raúl Marticorena Sánchez, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Álvaro Delgado Pascual, con DNI 71363793Z, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 18 de mayo de 2020

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

Introducción

Descripción del contenido del trabajo y del estrucutra de la memoria y del resto de materiales entregados.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

Conceptos teóricos

A continuación se va a detallar qué es un asistente de voz, una *Skill*, que elementos tiene y cómo funcionan. También se explicará el funcionamiento de *Mycroft*, que es el asistente de voz que se ha usado para el desarrollo del proyecto.

3.1. Asistente de voz

Un asistente de voz es un programa que actúa para un usuario, al que ayuda automatizando y relizando tareas con una interacción hombre-máquina mínima. La interacción entre el asistente y la persona debe ser una interacción lo más natural posible, es decir, la persona se comunica con la voz como es habitual y el asistente procesa, interpreta y responde de la misma forma.

3.2. Skill

Una *Skill* es una aplicación diseñada para un asistente de voz. Estas *skills* tienen diferentes elementos que hacen que funcionen, como la *wake word*, las *utterances*, los *intents* y los *dialogs*.

Wake Word

La *wake word* es una palabra o palabras que tiene que decir el usuario para que el asistente comience a escuchar. En el asistente de voz que se ha usado para el proyecto, la *wake word* es configurable, pero por defecto es "Hey Mycroft".

Utterance

Una *utterance* es la frase completa que dice el usuario a continuación de la *wake word* y que sirve como inicio del proceso de búsqueda de la *Skill* correspondiente a la frase dicha. Un ejemplo de *utterance* sería: ¿que temperatura hace en Burgos?

Intent

Un *intent* es la parte de la frase o *utterance* que el asistente de voz detecta como la parte necesaria para la invocación de una *Skill*. Cada Skill tiene un *intent* asociado. Siguiendo con el ejemplo anterior, el intent sería: temperatura Burgos

Dialog

Este término sí que es diferente de otros asistentes. Un *dialog* es una frase que dice el propio asistente, normalmente en respuesta a una pregunta por parte del usuario. Los *dialog* son independientes de las *Skill*, pero es normal que cada *Skill* tenga un *dialog*. Es el último paso cuando usamos una *Skill*, pero no tiene por qué estar presente. Finalizando con el ejemplo, el *dialog* sería: Hay 11ºC en Burgos.

3.3. Mycroft

Mycroft es un asistente de voz de código abierto y software libre. Está disponible para sistemas operativos basados en Linux, Android, Raspberry Pi y dispositivos propios de Mycroft, como el Mark 1. Es una aplicación cliente que requiere una instalación previa, a diferencia de otros asistentes de voz que se ejecutan en la nube. Al ser de código abierto te permite explorar y cambiar su implementación y diseño, no como en asistentes de voz como Alexa que son una caja negra en la que no puedes ver nada. Al ser una aplicación cliente existen diferentes componentes que tienen su propia responsabilidad dentro de la aplicación.

Voice

Este componente es el encargado de transformar la voz a texto (*speech-to-text* o *STT*) y el texto a voz (*text-to-speech* o *TTS*). Para el STT, Mycroft usa por defecto el motor de Google, ya que se necesita que la transformación sea rápida y precisa. Para añadir una capa adicional de

privacidad, todas las peticiones de STT pasan por un proxy de los servidores de Mycroft. Así, Google no puede detectar si hay una persona haciendo miles de peticiones o son miles de personas haciendo pocas peticiones. Además del motor de Google, Mycroft permite usar otros motores de STT como Mozilla DeepSpeech o Kaldi. En cuanto al TTS, se puede configurar desde la página del dispositivo. Las opciones de British Male y American Female usan Mimic 1, American Male usa Mimic 2 y Google Voice usa la voz de la API de Google Translate. Además de estos motores configurables desde la web, se puede usar Google TTS o Mycrosoft Azure, entre otros.

Skills

Este componente usa principalmente un servicio, llamado *intent service*, que es el encargado de, dada una *utterance*, hacer *match* entre la *utterance* y una *skill*. Esto es posible gracias a los *intent parser* que usa Mycroft, Adapt y Padatious. Adapt es una aplicación de código abierto ligera diseñada para usarse en dispositivos con recursos limitados, lo que es muy útil para Mycroft. Padatious es una aplicación basada en redes neuronales y *machine learning* y es más efectiva y fácil de usar que Adapt, además de ofrecer más funcionalidad.

MessageBus

El MessageBus es el componente que permite que el resto de componentes se comuniquen entre sí. Es un *websocket* que se encarga de pasar la información entre el componente de Skills y Voice. Cuando el usuario dice una frase, el componente Voice lo transforma a texto mediante el *TTS*, lo pasa al componente Skills mediante el MessageBus y éste decide qué *Skill* hay que ejecutar. Una vez ejecutada la *Skill*, se envía la respuesta al componente Voice, que la transforma a voz mediante el *STT*.

Técnicas y herramientas

4.1. REST API

Una API [4](*Application Programming Interface*) es un conjunto de definiciones y protocolos usados para desarrollar e integrar el software de las aplicaciones. REST [1] es una interfaz entre sistemas y que usa HTTP para obtener datos o realizar distintas operaciones sobre esos datos en todos los formatos posibles.

4.2. Moodle

Moodle es una plataforma de aprendizaje que permite a docentes y alumnos impartir y recibir clases a distancia, así como facilitar la gestión de los cursos. Es una herramienta de código abierto y por lo tanto cualquiera puede utilizarlo. En su página se pueden encontrar versiones de prueba como Mount Orange School

Es la plataforma en la que está basada UBUVirtual y es ampliamente utilizada en todo el mundo. Además posee una API con servicios web bastante completa que facilita mucho el trabajo.

WebServices

Web Services es el nombre que se usa en Moodle para referirse a la API REST. Estos *Web Services* facilitan la interacción con Moodle para los programadores, quienes no tienen que recurrir a otros métodos como el *web scraping* que requieren un mayor tiempo para implementarlos. Los *Web Services* basan su funcionamiento en un *token*, que es una cadena de

caracteres asociada a un usuario y que se obtiene a través de la función **moodle_mobile_app**. Gracias a este *token* se puede acceder al resto de funciones del *web service*, las cuáles te permiten, por ejemplo, obtener los eventos del calendario, cursos de un usuario o sus notas.

4.3. Metodología de desarrollo

Para el desarrollo del proyecto se ha empleado metodología ágil, en concreto SCRUM. Se ha adoptado una estrategia de desarrollo iterativa, con sprints bimensuales en los que se han realizado las entregas parciales. Las reuniones de estado del proyecto se han realizado semanalmente, a diferencia de lo normal en SCRUM que se realizan diariamente.

4.4. Python

Python es un lenguaje de programación interpretado, dinámico, multiplataforma y multiparadigma, soportando orientación a objetos, programación imperativa y programación funcional.

4.5. GitHub

GitHub [3]es una plataforma de desarrollo colaborativo para alojar proyectos usando el sistema de control de versiones Git. Ya que se ha empleado SCRUM se ha utilizado ZenHub, que es una herramienta para la administración de proyectos que se integra con GitHub y ofrece herramientas para la metodología agil como el *Kanban* o distintos gráficos que te muestran la evolución del proyecto entre otras cosas.

4.6. Atom

Atom es un editor de código fuente de código abierto multiplataforma desarrollado por GitHub. Tiene integrado control de versiones Git y se le pueden añadir plugins.

Plugins

Los plugins que le he añadido a Atom son atom-ide-ui que mejora la UI de Atom y añade la funcion de IDE. Para que el IDE funcione con Python también se ha instalado el plugin ide-python.

4.7. PyQt5

PyQt5 es un binding (una adaptación de una biblioteca para que sea usada en otro lenguaje) de Qt para Python. Qt es un framework multiplataforma orientado a objetos para desarrollar programas que utilizan interfaces gráficas de usuario. Es software libre y de código abierto.

Qt Designer

Es una herramienta para diseñar rápidamente interfaces gráficas a través de los *Widgets* de Qt. Es muy útil para crear prototipos rápidos de lo que quieres hacer, mediante la funcionalidad de drag-and-drop para poner los componentes de la interfaz y te permite traducir el prototipo creado a un lenguaje de programación como C++ o Python.

4.8. requests

Requests es una biblioteca para Python que permite enviar peticiones HTTP. Ha sido extremadamente útil para este proyecto gracias a su simplicidad de uso y cantidad de funcionalidades que tiene.

4.9. LaTeX

4.10. MiKTeX

MiKTeX es una distribución de LaTeX que está siempre actualizada, es fácil de instalar e incluye muchos paquetes.

4.11. TeXstudio

TeXstudio es un editor de LaTeX y un entorno de desarollo integrado (IDE), de código abierto. Ofrece varios servicios muy útiles, como resaltado de sintaxis o la corrección ortográfica. Es por esto que TeXstudio es una opción más atractiva que otros editores de LaTeX.

4.12. Java

4.13. Amazon Web Services

Los Amazon Web Services son un conjunto de servicios de computación en la nube. En concreto, de todos estos servicios se utilizó Amazon Lambda, una plataforma de serverless computing, que es un modelo de ejecución de computación en la nube en el que el proveedor proporciona el servidor y gestiona los recursos necesitados por el código ejecutado, lo que permite que el usuario pague únicamente por lo que necesita, en lugar de alquilar un servidor. Lambda ejecuta código como respuesta a eventos y es la base de los AWS.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende comentar los aspectos importantes y problemas que han surgido con la realización del proyecto, así como las decisiones de añadir o no más funcionalidad a la aplciación.

5.1. Utilización de AmazonWebServices y Alexa para la realización del proyecto

La idea inicial era utilizar los servicios web de Amazon (Amazon Lambda) y la consola de desarrollador (Amazon Developer) para crear la Skill de Alexa y hostearla en sus servidores.

Ya que nunca había usado estos servicios ni había desarrollado ninguna Skill o aplicación asistente de voz similar, leyendo la documentación de Alexa encontré una guía de cómo crear tu primera skill[2]. En un principio utilicé solo la guía para configurar la Skill dentro de Amazon Lambda y Amazon Developer, usando el código que había hecho ya usando los webservices de Moodle.

Problemas con Alexa

Sin embargo, cuando fui a hacer la prueba de esta primera versión de la Skill, no conseguía ninguna respuesta. Así que, al ser la primera vez que usaba la libreria de Amazon Skills Kit y la primera vez también que hacía un programa de este tipo, utilicé el código de prueba básico, que es un hello world para la Skill, que viene en la guía que estaba siguiendo. Y, a pesar

de seguir todos los pasos, intentándolo varias veces por si me había saltado algo, seguía sin recibir ninguna respuesta.

En uno de esos reintentos por conseguir que funcionara, me di cuenta de que salía un aviso de que el archivo con el proyecto de la Skill era demasiado pesado (superaba los 10MB). Aún no tengo claro que fuera ese el problema, ya que ese proyecto tenía lo mínimo necesario para hacer una Skill, pero de todas formas decidí continuar con el proyecto por otro camino.

5.2. Mycroft como alternativa

Esta herramienta es de código abierto, por lo que tuve la posibilidad de ver como están implementados los diferentes servicios del asistente de voz, modificarlos, etc. Además, para crear una Skill de Alexa, estás restringido en cierto modo de qué puedes y qué no puedes hacer, problema que no existe con Mycroft que te da libertad total para crear lo que quieras. Tampoco tiene las restricciones de tamaño que me impedían continuar con el proyecto en Alexa, así que por todas estas características decidí volver a empezar pero usando Mycroft esta vez.

Pero con Mycroft no son todo ventajas. De momento esta aplicación no se puede instalar en Windows, teniendo que usar una máquina virtual de Ubuntu o distribución de Linux para usarla, aunque se puede usar en otras plataformas como Android, Raspberry Pi y Docker. Al ser una aplicación cliente el usuario necesita instalarla para usar la Skill.

5.3. Aplicación cliente

En el anterior apartado he comentado la libertad que te ofrece Mycroft. Pues bien, esta herramienta se ejecuta como una aplicación cliente, a diferencia de Alexa que es un servicio en la nube y que necesitas emplear los servicios de Amazon para usar la Skill. Y, aunque Mycroft también depende de servicios, como el de text-to-speech o el de speech-to-text, y por ello una conexión a internet, se puede crear una aplicación cliente usando estos servicios ya que la propia aplicación de Mycroft es cliente.

Así que debido a esto el proyecto tomó un camino distinto al inicial, teniendo la posibilidad de crear una aplicación gráfica que resultaría en una aplicación más atractiva y completa en general.

5.4. Comunicación entre procesos

En el inicio de esta nueva etapa surgió un problema que no había planteado cuando decidí usar Mycroft como alternativa a Alexa. Al ejecutar la aplicación gráfica para loguearse en Moodle e invocar una Skill, se crea un nuevo programa con esa Skill, por lo que la comunicación entre la aplicación gráfica y Skill se complicó. Como solución utilicé comunicación entre procesos mediante sockets, lo que resultó más sencillo de lo que en un primer momento me había imaginado.

5.5. Deshabilitar módulos del asistente

Otra adición importante en la aplicación es la posibilidad de activar/desactivar distintos *intents*. Sin embargo, en el estado actual de Mycroft no es posible hacer cambios de este nivel de granularidad, pudiendo únicamente desactivar o activar *Skills* completas. Así que la solución fue separar la *Skill* en varias, en función de qué hacen, consiguiendo tres *Skills*, una para los eventos del calendario, otra para distintas acciones de un curso y otra para las notas.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos.* URL: https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos (visitado 12-05-2020).

[2] *Developing Your First Skill.* URL: https://developer.amazon.com/es-ES/docs/alexa/alexa-skills-kit-sdk-for-java/develop-your-first-skill.html (visitado 30-03-2020).

[3] *GitHub — Wikipedia, The Free Encyclopedia.* URL: http://es.wikipedia.org/w/index.php?title=GitHub&oldid=125798761 (visitado 12-05-2020).

[4] *¿Qué es una API?* URL: https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces (visitado 12-05-2020).