



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**UBUAssistant
Interacción por voz con la
plataforma Moodle**



Presentado por Álvaro Delgado Pascual
en Universidad de Burgos — 25 de junio
de 2020

Tutor: D. Raúl Marticorena Sanchez



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 UBUAssistant interacción por voz con la plataforma Moodle.

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, 25 de junio de 2020

Vº. Bº. del Tutor:

D. Raúl Marticorena Sánchez

Resumen

Los asistentes de voz son una nueva tecnología cuyo uso se ha disparado durante los últimos años. Estos asistentes nos permiten realizar distintas tareas de una forma muy cómoda, interactiva y empleando poco tiempo. Tareas como consultar el tiempo, una búsqueda en Internet, la compra... en definitiva una gran variedad de tareas, en muchos campos distintos.

En el ámbito de la educación, la enseñanza *online* y las aulas virtuales también han tenido un importante desarrollo tecnológico, siendo Moodle un referente en este campo. Sin embargo, los asistentes de voz no se han utilizado en este área, a pesar de sus características y posibilidades que ofrecen para los estudiantes. Es por ello que ha surgido la idea de este proyecto, para dar a los usuarios de estas plataformas de educación una nueva forma de interactuar con ellas, de una forma más rápida y cómoda.

UBUAssistant es una aplicación cliente desarrollada en Python utilizando el asistente de voz Mycroft. Esta aplicación permite que el usuario, mediante comandos de voz o a través de texto, obtenga información sobre el curso académico sin necesidad de conocer como funciona Moodle ni saber cómo tiene que navegar por la plataforma para obtener esta información.

Descriptores

Asistente de voz, Moodle, REST, intent, utterance, TTS, STT, skill

Abstract

Voice assistants are a new technology whose use has exploded over the last few years. These assistants allow us to perform varying tasks in an interactive and very comfortable way. Tasks such as check the weather, an Internet search, go shopping... in short a wide variety of tasks in many of fields.

In the educational environment, online teaching and virtual classrooms have had an important technological development too, Moodle being a reference in this field. However, voice assistants have not been used in this area, despite of its features and the possibilities they bring to students. This is the reason that the idea of the project was conceived, to bring the users of these educational platforms a new way of interacting with them, in a quicker and easier way.

UBUAssistant is a client application developed in Python using Mycroft as the voice assistant. This application allows the user to obtain information of the academic course through voice commands or text input, without the need of knowing what is Moodle or what steps he/she needs to do in order to retrieve this information.

Keywords

Voice assistant, Moodle, REST, intent, utterance, TTS, STT, skill

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
Conceptos teóricos	5
3.1. Asistente de voz	5
3.2. Skill	5
Técnicas y herramientas	7
4.1. REST API	7
4.2. Web Scraping	7
4.3. Moodle	7
4.4. Metodología de desarrollo	8
4.5. Mycroft	8
4.6. Python	10
4.7. GitHub	10
4.8. Atom	11
4.9. PyQt5	11
4.10. requests	12

4.11. L ^A T _E X	12
4.12. Análisis de software	13
4.13. Zotero	13
4.14. PlantUML	13
4.15. Amazon Web Services	14
Aspectos relevantes del desarrollo del proyecto	15
5.1. Utilización de AmazonWebServices y Alexa para la realización del proyecto	15
5.2. Mycroft como alternativa	16
5.3. Aplicación cliente	16
5.4. Comunicación entre procesos	17
5.5. Webservices de Moodle	17
5.6. Deshabilitar módulos del asistente	17
5.7. Skill mycroft-volume.mycroftai	17
Trabajos relacionados	19
Conclusiones y Líneas de trabajo futuras	21
7.1. Conclusiones	21
7.2. Líneas de trabajo futuras	22
Bibliografía	23

Índice de figuras

4.1. Arquitectura de Mycroft	10
4.2. Interfaz de Qt Designer	12
4.3. Resultados del análisis de SonarCloud	13

Índice de tablas

Introducción

Los asistentes de voz son una tecnología muy nueva y que no se ha utilizado en el campo de los sistemas de gestión de aprendizaje, aún teniendo unas características que los hacen muy interesantes para el ámbito de la educación. La posibilidad de obtener información importante para el estudiante a través de un comando de voz facilita mucho la interacción con este tipo de plataformas de educación.

Este proyecto se basa en la utilización de un asistente de voz para realizar diferentes tareas en una plataforma de Moodle, como es UBUVirtual. Estas tareas son recuperar los eventos del calendario de un curso, de un día en concreto o en las próximas dos semanas, leer los foros de un curso y recuperar las notas de un usuario tanto finales como de un curso en concreto. La utilización de estas nuevas tecnologías permiten obtener información de la plataforma de Moodle de una forma más interactiva y cómoda, sin tener que navegar por la web.

En esta memoria se van a ir explicando de forma ordenada los conceptos necesarios para entender el proyecto, las técnicas y herramientas utilizadas para su realización y los problemas y decisiones que han surgido durante el desarrollo del mismo.

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.

2.1. Objetivos generales

- Diseñar e implementar una aplicación que permita a los usuarios de una plataforma de Moodle interactuar con esta.
- Conseguir que la interacción con el asistente sea lo más natural posible.
- Programar la aplicación para que añadir futura funcionalidad o realizar cambios sea sencillo.

2.2. Objetivos técnicos

- Aplicar los conocimientos adquiridos como patrones de diseño, buenas prácticas de programación, etc.
- Familiarizarme con el uso de una REST API, como por ejemplo los *web services* de Moodle.
- Acostumbrarme a empezar un proyecto desde cero, tanto en el diseño de este como en su implementación.
- Utilizar las herramientas existentes en el asistente de voz.

Conceptos teóricos

A continuación se va a detallar qué es un asistente de voz, una *Skill*, qué 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 realizando 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: "¿qué 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 es particular de Mycroft. 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 11°C en Burgos".

Técnicas y herramientas

4.1. REST API

Una API [20] (*Application Programming Interface*) es un conjunto de definiciones y protocolos usados para desarrollar e integrar el software de las aplicaciones. REST [3] 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. Web Scraping

Web scraping [19] es una técnica utilizada para obtener información de páginas web simulando la navegación de una persona en la web. Este recurso se utiliza cuando no hay otra opción para obtener datos de una página web. En este proyecto se implementó para obtener la sesión del usuario en UBUVirtual y conseguir datos que no podía obtener mediante otro medio. Finalmente esta técnica no se ha utilizado aunque se ha dejado implementada para posibles usos futuros.

4.3. Moodle

Moodle [13] es una plataforma de gestión 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 cuales te permiten, por ejemplo, obtener los eventos del calendario, cursos de un usuario o sus notas.

4.4. 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* normalmente de dos semanas 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.

4.5. Mycroft

Mycroft[14] 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. Su diseño está pensado para que sea modular, permitiendo a los usuarios cambiar sus componentes.

Al ser de código abierto te permite explorar y cambiar su implementación y diseño, no como otros asistentes de voz que son una caja negra en la que no puedes ver nada. Al ser una aplicación cliente existen diferentes componentes, como se puede observar en la figura 4.1, que tienen su propia responsabilidad dentro de la aplicación.

Voice[23]

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 Microsoft Azure, entre otros.

Skills[23]

Este componente usa principalmente un servicio, llamado servicio de intents (*intent service*), que es el encargado de, dada una frase o *utterance*, emparejarla con una *skill*. Esto es posible gracias a los analizadores de intents (*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[11]

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 los componentes 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*.

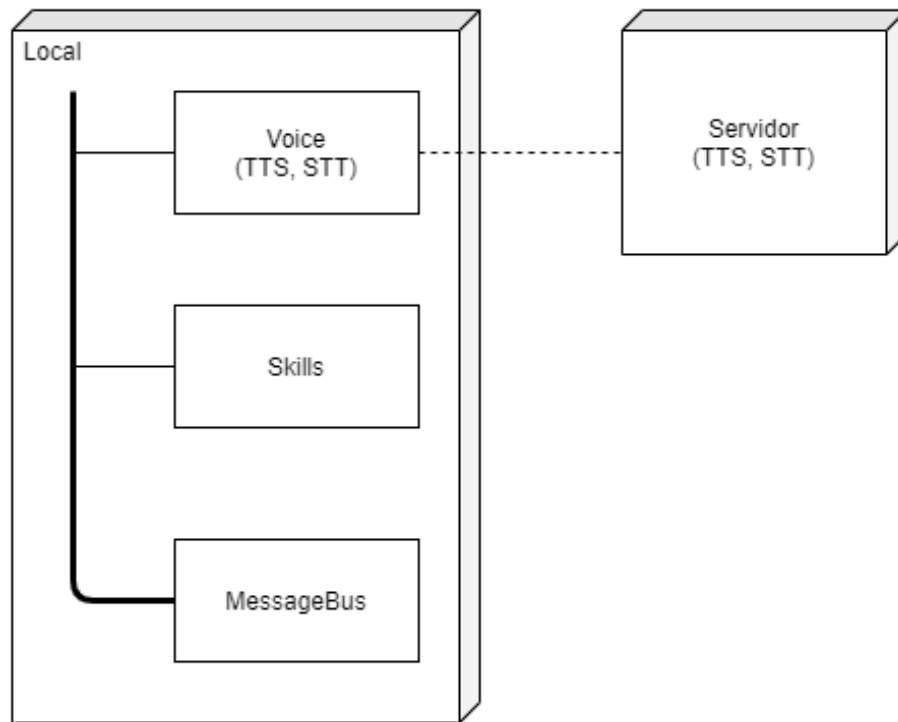


Figura 4.1: Arquitectura de Mycroft

4.6. Python

Python [17] es un lenguaje de programación interpretado, dinámico, multiplataforma y multiparadigma, soportando orientación a objetos, programación imperativa y programación funcional.

4.7. GitHub

GitHub [7] 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 ágil como el *Kanban* o distintos gráficos que te muestran la evolución del proyecto entre otras cosas.

4.8. Atom

Atom [4] 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* para añadir distintas funciones.

Plugins

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

4.9. PyQt5

PyQt5 [16] 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

QtDesigner [18] 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 arrastrar y soltar (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.

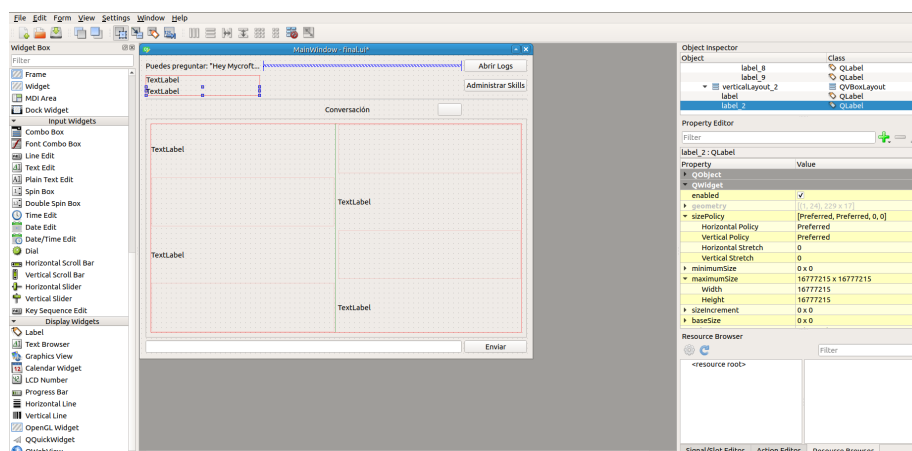


Figura 4.2: Interfaz de Qt Designer

4.10. 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.

4.11. L^AT_EX

L^AT_EX[8] es un sistema de composición de textos que se utiliza para crear documentos con una alta calidad tipográfica. Se utiliza comúnmente en artículos y libros científicos. Para esta memoria se ha utilizado la distribución MiKTeX con el editor TeXstudio.

MiKTeX

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

TeXstudio

TeXstudio [24] es un editor de L^AT_EX y un entorno de desarrollo 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 L^AT_EX.

4.12. Análisis de software

El análisis de software[2] es un proceso mediante el cual se analiza el código fuente teniendo en cuenta propiedades como seguridad, robustez, fiabilidad o mantenibilidad. Es algo muy importante en cuanto al desarrollo software permitiendo al programador identificar posibles errores en el código. Para el proyecto se han utilizado las herramientas **Codacy** y **SonarCloud**

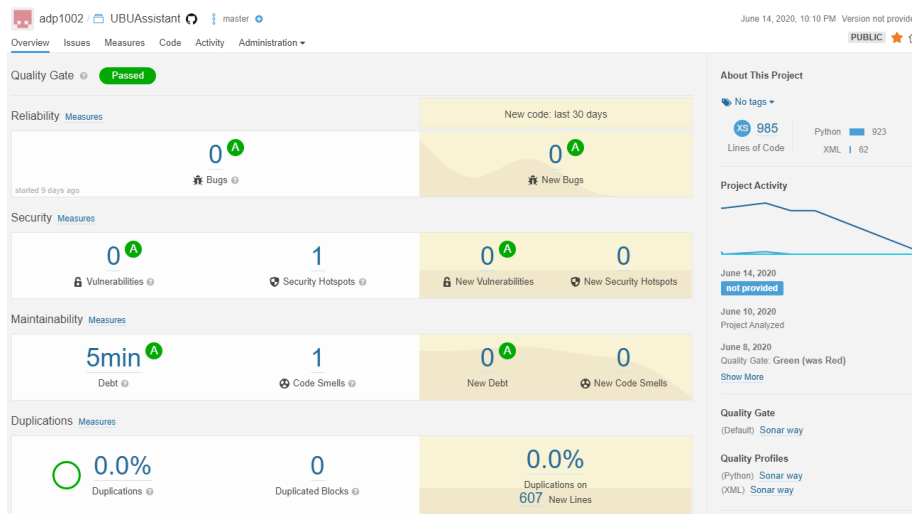


Figura 4.3: Resultados del análisis de SonarCloud

4.13. Zotero

Zotero [25] es un gestor de referencias bibliográficas, de código libre y abierto y multiplataforma. Además de la versión cliente también dispone de una **página** para generar referencias bibliográficas. Ha resultado muy útil para generar la bibliografía en formato BibTeX y es muy cómodo de usar.

4.14. PlantUML

PlantUML [15] es una herramienta de código abierto que permite la creación de diagramas UML desde un lenguaje de texto plano. Con PlantUML solo tienes que especificar las relaciones que hay en el diagrama y te genera automáticamente el diagrama en una imagen.

4.15. Amazon Web Services

Los Amazon Web Services [1] son un conjunto de servicios de computación en la nube. En concreto, de todos estos servicios se utilizó Amazon Lambda para gestionar los eventos originados por la activación de la *skill*. Amazon Lambda es una plataforma de *serverless computing*[21], que es un modelo de ejecución de computación en la nube en el que el proveedor proporciona el servidor y gestiona los recursos necesarios 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. Al final no se han utilizado en el proyecto, pero se estudió su funcionamiento durante el desarrollo del prototipo de *skill* para Alexa.

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 aplicación.

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 alojarla 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* [5]. En un principio solo utilicé la guía para configurar la *Skill* dentro de Amazon Lambda y Amazon Developer, usando el código que había hecho ya utilizando 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 librería 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 cómo están implementados los diferentes servicios del asistente de voz, modificarlos, etc. Además, para crear una *Skill* de Alexa, estás restringido en cierto modo 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, de lo que se hablará en el capítulo 7, líneas de trabajo futuras. 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. Webservices de Moodle

Uno de los objetivos de este proyecto es que el usuario pueda consultar sus calificaciones, sin embargo solo existía un *web service* que permitía obtener las calificaciones finales de los cursos del usuario. Entonces, a falta de un *web service* que me permitiera obtener las calificaciones completas de cada curso del usuario, se optó por utilizar *web scraping*.

Un tiempo después de haber implementado esta forma de obtener las calificaciones del usuario Moodle actualizó sus *web services* y en esta actualización existía un *web service*, `gradereport__user__get__grade__items` [9], que permitía obtener las calificaciones de un curso de un usuario, así que deseché la implementación usando *web scraping* y lo implementé usando este *web service* para continuar con la compatibilidad de la aplicación para cualquier Moodle.

5.6. 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.

5.7. Skill mycroft-volume.mycroftai

De un día para otro empecé a tener problemas con el sonido en general de la aplicación. A veces el micrófono no estaba activo, otras veces se

desactivaba tras detectar la *wake word*, el audio del TTS se silenciaba, etc. Sin embargo no había realizado ningún cambio relacionado con el audio o Mycroft en sí, así que tras mucho probar, buscar en los foros de Mycroft y mirar los logs de la aplicación encontré un post que hablaba sobre problemas muy similares a los que estaba teniendo y que estaban relacionados con la *skill* **mycroft-volume.mycroftai**, que es una *skill* propia de Mycroft. Tras desactivarla para comprobar que ese era el problema todo empezó a funcionar como debería así que por esta razón decidí dejar esta *skill* desactivada para la aplicación.

Trabajos relacionados

El campo de este proyecto está muy inexplorado, aunque si que hay algún trabajo bastante similar a este proyecto.

ALEXA SKILL VOICE INTERFACE FOR THE MOODLE LEARNING MANAGEMENT SYSTEM[10] es un proyecto de una *skill* para Alexa que permite a los usuarios acceder a anuncios de un curso, notas y eventos próximos.

Sin embargo, en el campo de los asistentes de voz existen múltiples proyectos en gran variedad de campos. En la [página](#) de Mycroft para conseguir nuevas *skills* puedes por ejemplo [controlar las luces](#), [controlar las horas de buses](#) o [jugar a un juego](#).

Además, el uso de los asistentes de voz se puede escalar a proyectos mucho mayores que los anteriores. Por ejemplo, la universidad de Saint Louis instaló 2300 *Echo Dot*[22], que son los dispositivos para el asistente de voz Alexa, y crearon una *skill personalizada* que ofrecía muchas posibilidades para los estudiantes, como preguntar por eventos y organizaciones, hora de cierre de la biblioteca o dónde se puede comer.

Las ventajas del proyecto de la universidad de Saint Louis respecto al mío son la facilidad de uso, ya que los estudiantes solo tienen que aprender a utilizar la *skill* para tener acceso a la información que esta les provee, y que, siendo un proyecto mucho más grande hecho por la universidad, los usuarios tienen acceso a más funcionalidad, por ejemplo saber las cafeterías disponibles para comer. La desventaja principal sería el coste del proyecto, la instalación de los 2300 *Echo Dot* rondaría los 80.500€ si cada dispositivo les cuesta 35€, que es lo que valen en la página de Amazon. Solo eso costaría 16 veces más que este proyecto.

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

Una de las conclusiones que he sacado de este proyecto es que, aunque te encuentres con grandes problemas como fue el prototipo de skill para Alexa, en este tipo de proyectos es importante no abandonar y buscar otro camino, porque aunque parezca una pérdida de tiempo o algo problemático, puede ser que llegues a un mejor resultado que el esperado, como por ejemplo poder hacer una interfaz gráfica tras el cambio a Mycroft.

Otra conclusión a la que se ha llegado es que es importante dedicarle más tiempo a pensar qué es lo que quieres hacer que ponerse a hacer sin tener un plan, porque al final acabas dedicando más tiempo arreglando y cambiando cosas que el que dedicas cuando piensas las cosas.

Como conclusión final, se puede afirmar que se han cumplido los objetivos del proyecto. El producto final es una aplicación que permite al usuario interactuar con una plataforma de Moodle mediante voz y texto, configurar el asistente de voz y multitud más de funcionalidad.

Se han aplicado muchos de los conocimientos aprendidos durante el grado, como puede ser la interacción hombre-máquina, buenas prácticas de programación, defectos de diseño, refactorización de código, etc.

También se ha aprendido mucho en cuanto a la realización de interfaces gráficas, con las que no tenía apenas experiencia.

En definitiva, gracias a lo aprendido en la carrera se ha creado una aplicación funcional, útil y cumpliendo con los objetivos del proyecto. Ha

sido una experiencia bastante agradable, aunque a veces te encuentres con problemas es muy satisfactorio solventarlos y te motiva para continuar con el proyecto y explorar un campo completamente desconocido para mí, como eran los asistentes de voz, es muy entretenido y se aprende mucho.

7.2. Líneas de trabajo futuras

Durante la realización del proyecto han surgido varias ideas sobre cómo se puede mejorar la aplicación.

Una de las mayores desventajas que tuvo cambiar de Alexa a Mycroft es que Mycroft no tiene soporte para Windows. Sin embargo, es posible ejecutar Mycroft desde Docker[6], que permite automatizar el despliegue de aplicaciones, por lo que se podría ejecutar la aplicación con Docker y utilizarla conectándose al contenedor desde Windows. Además también es posible utilizar Mycroft desde Android, que facilitaría mucho su uso para los estudiantes.

Otra posible continuación del desarrollo es guardar los datos que se obtienen de Moodle en ficheros locales de forma que se pueda utilizar la aplicación sin conexión a Internet. Usar la aplicación *offline* reduciría bastante la funcionalidad de la aplicación, principalmente el servicio de STT, aunque gracias a que Mycroft es modular se pueden cambiar estos elementos. De hecho, el motor de TTS por defecto en Mycroft, Mimic[12] permite instalarse y utilizarse de forma local. En este [post del foro](#) de Mycroft se detalla bastante qué cambios son necesarios para utilizar Mycroft sin conexión.

Una de las formas más claras de mejorar la aplicación es aumentar la funcionalidad de la aplicación, permitiendo al usuario obtener más información de Moodle.

Por último se puede mejorar la interacción con el usuario. Esto implica mejorar la interfaz gráfica, que puede ser añadiendo funcionalidad o mejorar la claridad visual, y mejorar la interacción con Mycroft, de forma que las conversaciones sean más "humanas".

Bibliografía

- [1] *Amazon Web Services*. Marzo de 2020. URL: https://es.wikipedia.org/w/index.php?title=Amazon_Web_Services&oldid=124664856 (visitado 14-06-2020).
- [2] *Análisis de software*. Agosto de 2019. URL: https://es.wikipedia.org/w/index.php?title=An%C3%A1lisis_de_software&oldid=118062970 (visitado 14-06-2020).
- [3] *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 12-06-2020).
- [4] *Atom (software)*. Marzo de 2020. URL: [https://es.wikipedia.org/w/index.php?title=Atom_\(software\)&oldid=124393266](https://es.wikipedia.org/w/index.php?title=Atom_(software)&oldid=124393266) (visitado 12-06-2020).
- [5] *Developing Your First Skill / Alexa Skills Kit*. URL: </es-ES/alexa/techdoc-template> (visitado 12-06-2020).
- [6] *Docker (software)*. Junio de 2020. URL: [https://es.wikipedia.org/w/index.php?title=Docker_\(software\)&oldid=126606072](https://es.wikipedia.org/w/index.php?title=Docker_(software)&oldid=126606072) (visitado 16-06-2020).
- [7] *GitHub*. Junio de 2020. URL: <https://es.wikipedia.org/w/index.php?title=GitHub&oldid=126716929> (visitado 12-06-2020).
- [8] *LaTeX*. Junio de 2020. URL: <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=126802021> (visitado 14-06-2020).

- [9] *[MDL-64298] gradereport_user_get_grade_items fails when "Course Grade Settings"when "Show Feedbacks set to "False"for user reports - Moodle Tracker*. URL: <https://tracker.moodle.org/browse/MDL-64298> (visitado 12-06-2020).
- [10] Michelle Layne Melton. "ALEXA SKILL VOICE INTERFACE FOR THE MOODLE LEARNING MANAGEMENT SYSTEM". En: (), pág. 109. URL: http://libres.uncg.edu/ir/asu/f/Melton_Michelle_2019_Thesis.pdf.
- [11] *MessageBus*. URL: <https://mycroft-ai.gitbook.io/docs/mycroft-technologies/mycroft-core/message-bus> (visitado 14-06-2020).
- [12] *Mimic*. URL: <https://mycroft-ai.gitbook.io/docs/mycroft-technologies/mimic-overview> (visitado 16-06-2020).
- [13] *Moodle*. Mayo de 2020. URL: <https://es.wikipedia.org/w/index.php?title=Moodle&oldid=126448465> (visitado 14-06-2020).
- [14] *Mycroft (software)*. Septiembre de 2019. URL: [https://es.wikipedia.org/w/index.php?title=Mycroft_\(software\)&oldid=119343893](https://es.wikipedia.org/w/index.php?title=Mycroft_(software)&oldid=119343893) (visitado 12-06-2020).
- [15] *PlantUML*. Junio de 2020. URL: <https://en.wikipedia.org/w/index.php?title=PlantUML&oldid=962483456> (visitado 14-06-2020).
- [16] *PyQt*. Mayo de 2020. URL: <https://es.wikipedia.org/w/index.php?title=PyQt&oldid=126506459> (visitado 12-06-2020).
- [17] *Python*. Mayo de 2020. URL: <https://es.wikipedia.org/w/index.php?title=Python&oldid=126115328> (visitado 14-06-2020).
- [18] *Qt Designer Manual*. URL: <https://doc.qt.io/qt-5/qtdesigner-manual.html> (visitado 14-06-2020).
- [19] *Qué es el web scraping*. Enero de 2019. URL: <https://aukera.es/blog/web-scraping/> (visitado 14-06-2020).
- [20] *¿Qué es una API?* URL: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces> (visitado 12-06-2020).
- [21] *Serverless computing*. Mayo de 2020. URL: https://en.wikipedia.org/w/index.php?title=Serverless_computing&oldid=959625452 (visitado 14-06-2020).
- [22] *SLU Alexa Project*. en. URL: <https://www.slu.edu/news/2018/august/slu-alexa-project.php> (visitado 22-06-2020).
- [23] *Technology Overview*. URL: <https://mycroft-ai.gitbook.io/docs/mycroft-technologies/overview> (visitado 14-06-2020).

- [24] *TeXstudio*. Octubre de 2019. URL: <https://es.wikipedia.org/w/index.php?title=TeXstudio&oldid=120138946> (visitado 12-06-2020).
- [25] *Zotero*. Junio de 2020. URL: <https://es.wikipedia.org/w/index.php?title=Zotero&oldid=126818652> (visitado 14-06-2020).