



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**
título del TFG



Presentado por Nombre del alumno
en Universidad de Burgos — 22 de abril
de 2020

Tutor: nombre tutor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Nombre del alumno, con DNI dni, 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, 22 de abril 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 general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Skill	5
Técnicas y herramientas	7
4.1. Metodología de desarrollo	7
4.2. Moodle	7
4.3. PyQt5	7
4.4. requests	8
4.5. MiKTeX	8
4.6. TeXstudio	8
Aspectos relevantes del desarrollo del proyecto	9
5.1. Utilización de AmazonWebServices y Alexa para la realización del proyecto	9
5.2. Problemas con Alexa	10
5.3. Mycroft como alternativa	10
5.4. Aplicación cliente	10
5.5. Comunicación entre procesos	10

Trabajos relacionados	13
Conclusiones y Líneas de trabajo futuras	15

Índice de figuras

Índice de tablas

Introducción

Descripción del contenido del trabajo y del estructura 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 una *Skill* de un asistente de voz, 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. Skill

Una *Skill* es una aplicación pero para un asistente de voz, por lo que funciona mediante voz, tanto hablar como escuchar. Estas *skills* tienen diferentes elementos que hacen que funcionen, como las *utterances*, los *intents*, los *dialogs* y la *wake word*

Wake Word

La *wake word* es una palabra o palabras que tiene que decir el usuario para que la *Skill* comience a escuchar.

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.

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.

Dialog

Este término sí que es diferente de otros asistentes como Alexa. Un *dialog* es una frase que dice el propio asistente. Los *dialog* son independientes de las *Skill*, pero es normal que cada *Skill* tenga un *dialog*.

3.2. Mycroft

Mycroft es un asistente de voz de código abierto y software libre. Está disponible para sistemas operativos basados en Linux, Android, en una 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 Microsoft 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. Metodología de desarrollo

Para el desarrollo del proyecto se ha utilizado 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.

4.2. 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**

4.3. PyQt5

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

4.4. requests

Requests es una biblioteca para Python que permite enviar peticiones HTTP de forma muy sencilla.

4.5. MiKTeX

4.6. TeXstudio

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 hostearla en sus servidores.

Ya que nunca había usado nunca 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.

Amazon Web Services

Los Amazon Web Services, AWS a partir de ahora, son un conjunto de servicios de computación en la nube. En concreto, de todos estos servicios se utilizó Amazon Lambda, que es una plataforma sin servidor basada en eventos, que es la base de los AWS. Lambda ejecuta código como respuesta a eventos y gestiona los recursos necesarios por el código ejecutado.

Amazon Alexa

Es un servicio de voz en la nube para dispositivos de Amazon y dispositivos de terceros que usan Alexa.

5.2. Problemas con Alexa

5.3. Mycroft como alternativa

Esta herramienta es de código abierto, por lo que tengo 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. Al ser una aplicación cliente el usuario necesita instalarla para usar la Skill.

5.4. 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 necesitas usar los servicios de Amazon para usar la Skill. Y aunque Mycroft también depende de servicios de text-to-speech y 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 gracias a esto el proyecto tomó un camino distinto al inicial, teniendo la posibilidad de crear una aplicación gráfica más atractiva en general.

5.5. 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

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.