UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

UBUVoiceAssistant v2 Documentación Técnica

Presentado por Rodrigo Garcia Martin
en Universidad de Burgos — 15 de junio de 2021
Tutor: Raúl Marticorena Sánchez

Índice general

Índice general I

Índice de figuras III

Índice de tablas IV

Apéndice A Plan de Proyecto Software 1
A.1. Introducción 1
A.2. Planificación temporal 1
A.3. Estudio de viabilidad 2

Apéndice B Especificación de Requisitos 3
B.1. Introducción 3
B.2. Objetivos generales 3
B.3. Catalogo de requisitos 3
B.4. Especificación de requisitos 5

Apéndice C Especificación de diseño 7
C.1. Introducción 7
C.2. Diseño de datos 7
C.3. Diseño procedimental 9
C.4. Diseño arquitectónico 9

Apéndice D Documentación técnica de programación 11
D.1. Introducción 11
D.2. Estructura de directorios 11
D.3. Manual del programador 12

D.4. Compilación, instalación y ejecución del proyecto 15
D.5. Pruebas del sistema 16

Apéndice E Documentación de usuario **17**
E.1. Introducción 17
E.2. Requisitos de usuarios 17
E.3. Instalación 17
E.4. Manual del usuario 20

Bibliografía **23**

Índice de figuras

Índice de tablas

C.1. Diccionario de datos de Conversation 8
C.2. Diccionario de datos de Message 8
C.3. Diccionario de datos de User 9
C.4. Diccionario de datos de Course 9

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apartado se detallan la planificación temporal que se ha seguido para realizar el proyecto, junto con la metodología utilizada, y la viabilidad del proyecto.

A.2. Planificación temporal

Para realizar el proyecto se ha usado una metodología ágil llamada SCRUM, pero reduciendo el número de personas a 1, y realizando reuniones semanales, para adaptarlo a las necesidades y a las características del trabajo.

Sprint 1

El primer sprint duró aproximadamente dos semanas (del 12 al 24 de febrero).

La reunión previa al sprint se dedicó para establecer los objetivos del proyecto. La mayor parte del sprint se dedicó a revisar el código inicial del proyecto, revisar errores, identificar posibilidades de mejora y oportunidades para incluir funcionalidades completamente nuevas. Principalmente, consultar cómo funciona el docker proporcionado por Mycroft y comprobar cómo se puede integrar e iniciar la documentación.

Sprint 2

El segundo sprint dura dos semanas (del 24 de febrero al 10 de marzo)

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apéndice se explican los objetivos del proyecto y los requisitos que se han definido para llevarlo a cabo.

B.2. Objetivos generales

El principal objetivo del proyecto es mejorar la versión anterior del asistente del UBUVoiceAssistant. Para cumplirlo, se proponen algunas mejoras, como una mejora en el proceso de instalación y configuración inicial o en la interacción con el asistente. También se propone conseguir una mayor robustez en la aplicación, obtener más información de Moodle e incluso poder enviar datos a la plataforma.

B.3. Catalogo de requisitos

Los nuevos requisitos que se han añadido a la aplicación son los siguientes:

Requisitos funcionales

- RF-1 Obtener datos de Moodle: La aplicación debe poder obtener distintos datos de Moodle.
    - RF-1.1 Obtener los mensajes privados: La aplicación debe poder leer los últimos mensajes que ha recibido el usuario de la misma.

    - RF-1.2 Buscar en los foros: La aplicación debe poder buscar en los foros los hilos relacionados con una frase que diga el usuario.

- RF-2 Enviar datos a Moodle: La aplicación debe poder enviar algunos datos a nuestra plataforma de Moodle.

    - RF-2.1 Enviar mensajes privados: La aplicación debe poder enviar mensajes privados a otros usuarios con los que se tiene una conversación abierta o que comparten un curso con el usuario de la aplicación.

- RF-3 Mostrar un sistema de ayuda: La aplicación debe poder indicar al usuario cómo interactuar, sugiriéndole algunas frases que puede decir.

- RF-4 Mejorar la instalación: Se deben mejorar varios aspectos para hacer más cómodo el proceso de puesta en marcha.

    - RF-4.1 Crear un instalador: Se debe crear un programa que prepare todas las dependencias necesarias y coloque todos los archivos en las ubicaciones necesarias para el correcto funcionamiento de la aplicación.
    - RF-4.2 Crear un asistente de emparejamiento: La aplicación debe guiar al usuario por el proceso de emparejamiento del cliente con el servidor de Mycroft.

Requisitos no funcionales

- RNF-1 Soporte para Windows: La aplicación debe poder ser ejecutada en Windows 10.

- RNF-2 Naturalidad: La interacción de la aplicación con el usuario debe ser lo más natural posible.

- RNF-3 Robustez: Se debe mejorar la robustez de la aplicación, haciéndola más tolerante a fallos.

- RNF-4 Internacionalización: La aplicación tiene que permitir añadir idiomas nuevos fácilmente.

- RNF-5 Calidad de código: La calidad del código debe ser buena, y tener una documentación completa.

B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

C.1. Introducción

En este apartado se detalla el diseño del proyecto, indicando claramente cuáles han sido los cambios más importantes con respecto a la versión anterior.

C.2. Diseño de datos

El modelo de datos que se usa dentro de la aplicación está fuertemente ligado con la estructura de datos que se usa dentro de Moodle. Sólo se ha creado el modelado necesario para almacenar los datos que se usan en las diferentes skills. Todos los archivos fuente que componen el modelo de datos se encuentran en la carpeta src/UBUVoiceAssistant/model. Las clases que lo forman son las siguientes:

- **Course** guarda los datos relacionados con los cursos.
- **Forum** almacena los foros de un curso.
- **Discussion** representa cada uno de los hilos de un foro.
- **GradeItem** guarda las notas.
- **Event** representa los eventos del calendario.
- **User** almacena todos los datos relacionados con el usuario.

- Se ha añadido la clase **Conversation** que almacena las conversaciones por mensaje privado.

- La nueva clase **Message** representa cada mensaje de una conversación.

Atributo	Tipo de dato	Descripción
conversation_id	int	Identificador de la conversación
isread	bool	Indica si hay mensajes sin leer
unreadcount	int	Número de mensajes pendientes
name	str	Nombre de la conversación
subname	str	Subtítulo de la conversación
members	dict	Diccionario de miembros Son objetos de tipo User La clave es el id del usuario
messages	dict	Diccionario de mensajes Son objetos de tipo Message La clave es el id del mensaje

Tabla C.1: Diccionario de datos de Conversation

Atributo	Tipo de dato	Descripción
message_id	int	Identificador del mensaje
useridfrom	int	Identificador del usuario que ha mandado este mensaje
text	str	Contenido del mensaje Contiene algunas etiquetas HTML
timecreated	int	Momento de envío del mensaje Timestamp en formato Unix

Tabla C.2: Diccionario de datos de Message

Atributo	Tipo de dato	Descripción
user_id	int	Identificador del usuario
courses	dict	Diccionario de los cursos del usuario Son objetos de tipo Course La clave es el id del curso Está vacío si no representa al usuario de la aplicación
fullname	str	Nombre completo del usuario

Tabla C.3: Diccionario de datos de User

Atributo	Tipo de dato	Descripción
course_id	str	Identificador del curso
name	str	Nombre del curso
grades	list	Lista que contiene las calificaciones del usuario de ese curso
events	list	Lista que contiene los eventos del curso
forums	list	Lista que contiene los foros de ese curso
participants	dict	Diccionario de los participantes Son objetos de tipo User La clave es el id del usuario

Tabla C.4: Diccionario de datos de Course

Las clases “Forum”, “Discussion”, “GradeItem” y “Event” no han sufrido cambios.

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este capítulo se van a mencionar todos los aspectos relevantes que es necesario conocer para poder seguir trabajando en una versión futura de la aplicación.

D.2. Estructura de directorios

Para organizar el proyecto, se ha hecho de la siguiente manera:

- /docs/: Carpeta que contiene todos los documentos e imágenes para generar los archivos PDF con la memoria y anexos.
- /scripts/: Varios scripts que se han usado para automatizar algunas acciones frecuentes, como la ejecución del programa desde la versión en desarrollo, la generación de la plantilla para las traducciones, o para realizar las pruebas de calidad de código.
- /testing/: Algunos archivos de prueba que he ido generando a lo largo del proyecto para probar algunas funcionalidades sin tener que lanzar el programa completo.
- /src/UBUVoiceAssistant/: La carpeta base del código fuente del proyecto. Tiene las siguientes carpetas en su interior:

- GUI: Contiene los archivos necesarios para la interfaz gráfica
- GUI/old_files: Archivos que hacían funcionar la anterior interfaz gráfica. No se usan, pero se han dejado como referencia.
- GUI/forms: En esta carpeta se encuentran los formularios generados con QtDesigner.
- GUI/forms/chat_window_html: Los documentos html, javascript y CSS necesarios para mostrar los bocadillos en la ventana de chat con el asistente.
- imgs: Contiene las imágenes que se usan en la interfaz gráfica.
- lang: Los archivos para traducir la interfaz gráfica. Tiene la plantilla llamada "translation_template.pot" y las carpetas para cada uno de los idiomas. Siguen la estructura *languageCode_countryCode*/LC_MESSAGES/UBUVoiceAssistant
- model: Archivos que representan los datos de Moodle
- prototipo: Archivos para un prototipo de skill para Alexa que se usaron en la anterior versión del proyecto.
- skills: Contiene las *Skills* del proyecto.
- util: Archivos que ofrecen varias funcionalidades para el proyecto.
- webservice: Código encargado de realizar las peticiones web al servidor de Moodle

D.3. Manual del programador

Como entorno de programación se recomienda usar como sistema operativo la última versión normal de Ubuntu, aunque también es posible usar la última LTS o cualquiera de las distribuciones que derivan de ellas. Es necesario instalar las dependencias de python3 que hemos usado en el proyecto, Mycroft, QtDesigner, Poedit y un editor de código como Visual Studio Code con algunas extensiones recomendadas para facilitar el trabajo.

Python3

Python3 ya viene preinstalado en Ubuntu, pero para que la aplicación funcione correctamente necesitamos instalar las bibliotecas usadas en el proyecto. Esto se puede hacer escribiendo los siguientes comandos en una terminal:

- `sudo apt-get install python3-pip python3-pyqt5 \ python3-pyqt5.qtwebengine git gettext mypy pylint -y`
- `sudo pip3 install mycroft-messagebus-client babel`

  En el caso de que se vayan a modificar o crear skills nuevas, es recomendable ejecutar los siguientes comandos para tener autocompletado dentro del IDE:
- `sudo apt-get install libfann-dev python3-dev swig -y`
- `sudo pip3 install adapt-parser padatious`

Mycroft

Para poder usar Mycroft es necesario tener una cuenta en su página web. También será necesario descargar el código de Mycroft desde su repositorio. Lo podemos hacer ejecutando los siguientes comandos:

- `sudo mkdir -p /usr/lib/mycroft-core`
- `sudo chown SUDO_USER /usr/lib/mycroft-core`
- `git clone https://github.com/MycroftAI/mycroft-core.git \ /usr/lib/mycroft-core`
- `/usr/lib/mycroft-core/dev_setup.sh -sm`

Tras introducir el último comando se abrirá una instalación interactiva de Mycroft. En todas las preguntas deberemos responder escribiendo la letra Y en la terminal.

Cuando termine, podemos emparejarlo abriendo la consola de Mycroft escribiendo los siguientes comandos:

- `cd /usr/lib/mycroft-core`
- `./start-mycroft.sh debug`

Tras esto, se mostrará en la terminal los registros de las operaciones que realiza Mycroft en la parte superior, mientras que en la parte inferior aparecen los registros de los mensajes intercambiados entre el usuario y Mycroft. Pasados unos segundos o unos minutos, dependiendo de la velocidad tanto del equipo como de la conexión a internet, se iniciará el proceso de

emparejamiento. En caso de no realizarse de manera automática, podemos escribir "pair my device" para iniciarlo manualmente. Mycroft recitará y escribirá lo que tenemos que hacer en este punto. Tendremos que ir a la página web de Mycroft, y en la parte superior derecha, hacer click en **Add device**. Deberemos introducir el código de 6 caracteres en el campo que pone **Pairing Code**. En la parte inferior deberemos seleccionar **Google Voice** como motor de voz. Una vez hecho, Mycroft dirá que se ha emparejado correctamente y puede empezar a funcionar. Podemos cerrar la interfaz de Mycroft pulsando *ctrl + c* o escribiendo `:exit`.

QtDesigner

Para instalar QtDesigner deberemos escribir el siguiente comando en una terminal:

- `sudo apt-get install qttools5-dev-tools`

Una vez hecho esto, podremos acceder al programa desde el menú de aplicaciones del sistema.

– Completar explicación de uso –

Poedit

Para editar los archivos de traducción de la interfaz es recomendable usar un programa que las muestre de forma visual y ofrezca algunas herramientas, como Poedit.

Para instalarlo, es tan sencillo como hacer:

- `sudo apt install poedit`

– Completar pequeño manual de uso –

Visual Studio Code

Si bien es posible usar cualquier editor de texto, voy a describir el proceso para instalar Visual Studio Code, ya que es uno de los editores más completos que existen a día de hoy.

Para descargarlo tendremos que ir a su página web, descargamos el archivo .deb, y hacemos doble click sobre él para instalarlo. En caso de que

no funcione, podemos abrir una terminal, navegar hasta la carpeta donde se encuentra el archivo usando el comando `cd` y realizar la instalación usando `sudo apt install ./nombreDelArchivo.deb`.

Una vez instalado, lo abrimos desde el menú de aplicaciones del sistema, hacemos click en el icono de las extensiones de la barra lateral e instalamos las siguientes:

- `Python`, de Microsoft. Nos permite usar funciones de autocompletado en el IDE.
- `Python Docstring Generator`, de Nils Werner. Nos permite generar la documentación de los métodos, a partir de los argumentos definidos en sus cabeceras.
- `Pylance`, de Microsoft. Mejora el autocompletado, permitiendo tener en cuenta el tipado.
- `Code Spell Checker`, de Street Side Software. Añade un corrector con las palabras inglesas.
- `Spanish - Code Spell Checker`, de Street Side Software. Añade el español como paquete de idioma a la anterior extensión.

D.4. Compilación, instalación y ejecución del proyecto

Descargamos el repositorio del proyecto desde la web usando el comando `git clone https://github.com/rogama25/UBUVoiceAssistant`.

Posteriormente, debemos usar los siguientes comandos para crear las carpetas necesarias y colocar los archivos en las rutas correctas:

- `cd UBUVoiceAssistant`
- `mkdir -p ~/.config/UBUVoiceAssistant`
- `mkdir -p ~/.config/mycroft`
- `cp -r ./src/UBUVoiceAssistant/skills/. /opt/mycroft/skills`
- `sudo mkdir -p /usr/lib/UBUVoiceAssistant`
- `sudo cp -r ./src/UBUVoiceAssistant/. /usr/lib/UBUVoiceAssistant`

Para ejecutarlo desde el entorno de desarrollo, escribimos:

- `cd src`
- `python3 -m UBUVoiceAssistant.GUI.main`

D.5. Pruebas del sistema

Para realizar las pruebas de calidad de código lo podemos hacer con las herramientas Pylint y mypy. Sin embargo, debido a que en las carpetas de las skills de Mycroft se suelen usar guiones y que no son caracteres válidos en Python, es necesario renombrar temporalmente las carpetas para que tengan un guión bajo.

Después podremos hacer:

- `pylint src -ry`
- `mypy src`

Para las pruebas de funcionalidad lo podemos realizar conectándonos a un servidor de Moodle cualquiera. Durante el desarrollo se ha usado tanto UBUVirtual, como Mount Orange School como un servidor Moodle instalado en la propia máquina, usando la documentación oficial.

Apéndice E

Documentación de usuario

E.1. Introducción

En este apartado se explica cómo se puede instalar y usar el proyecto.

E.2. Requisitos de usuarios

- Sistema operativo Windows 10 u Ubuntu 20.04 o superior.
- Conexión a internet
- 3GB de memoria RAM
- 5GB de espacio en disco libres

E.3. Instalación

Windows Susbsystem for Linux

En el caso de que hayamos optado por usar Windows 10 como sistema operativo, es necesario que instalemos este sistema para poder ejecutar la aplicación. Si se va a usar Ubuntu, saltar a la siguiente sección.

Instalación de WSL

Para comenzar, deberemos conocer la versión de Windows 10 que estamos ejecutando en nuestro equipo. Para ello, abrimos el menú de configuración de

Windows, hacemos click en Sistema" y luego en Acerca de". En esa pantalla deberemos fijarnos que tengamos un sistema operativo de 64 bits y tengamos la versión 1607 o superior.

Ahora, tenemos que buscar Powershell" en el menú de inicio y la ejecutaremos como administrador. En esa ventana, escribimos el siguiente comando:

```
dism.exe /online /enable-feature `
/featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

En el caso de que tengamos la versión 1903 o superior, con la compilación 18362 o superior, es recomendable instalar WSL2 para obtener un mayor rendimiento. En el caso de no tenerlo, saltamos al apartado de la instalación de Ubuntu para WSL.

Lo siguiente es activar la virtualización, usando este comando en la Powershell que habíamos abierto:

```
dism.exe /online /enable-feature `
/featurename:VirtualMachinePlatform /all /norestart
```

En este punto, reiniciamos el ordenador. Después del reinicio, instalamos el paquete de actualización del kernel de Linux, desde este enlace.

Abrimos otra Powershell como administrador y establecemos el uso de WSL2 por defecto:

```
wsl -set-default-version 2
```

Instalación de Ubuntu para WSL

Para instalar Ubuntu en el Windows Subsystem for Linux, tenemos que descargarlo desde la Microsoft Store, accesible en este enlace. Una vez instalado, es probable que se abra de manera automática una terminal en la que se nos pedirá elegir un nombre de usuario y una contraseña. En el caso de que no se abra automáticamente, podremos abrirla desde el menú de inicio, en el acceso directo a Ubuntu.

Instalación de Pulseaudio y servidor X11

Para que el susbsistema pueda recibir sonido del micrófono y mostrar interfaces gráficas debemos instalar lo siguiente:

VcXserv, descargable desde este enlace. Deberemos añadir al .bashrc del WSL las siguientes líneas:

`export DISPLAY=:0` (únicamente si no usamos WSL2)

`export DISPLAY=$(awk '/nameserver / {print $2; exit}' \
/etc/resolv.conf 2>/dev/null):0` (si usamos WSL2)

`export LIBGL_ALWAYS_INDIRECT=1`

Pulseaudio, descargable desde este enlace. Tendremos que editar los siguientes archivos:

`etc/pulse/default.pa`, donde tendremos que añadir:

`load-module module-native-protocol-tcp auth-ip-acl=127.0.0.1`

`etc/pulse/daemon.conf`, donde cambiaremos el `exit-idle-time` a -1.

Ejecutamos el programa bin/pulseaudio.exe y añadimos la siguiente línea al .bashrc: `export PULSE_SERVER=tcp:127.0.0.1`

Descarga del repositorio

Para descargar el código de la aplicación se puede hacer usando un navegador o usando la terminal. En el caso de que estemos usando Windows Subsystem for Linux, es recomendable usar la terminal.

Usando un navegador web

Desde cualquier navegador de internet, hay que ir a la página web del repositorio.

Allí, en la parte derecha, haz click en la sección "Releases" y, en el desplegable de "Assets", descarga el "Source code (.zip)" que aparezca en la versión más reciente.

Una vez se complete la descarga, primero hay que descomprimirlo y luego abrir una terminal donde nos moveremos hasta la carpeta que se ha creado con el nombre de la versión usando el comando `cd`.

Allí hay que ejecutar el comando "sudo ./install.sh install" y esperar a que termine. Puede que tarde varios minutos, dependiendo de la velocidad de internet y del equipo, ya que descarga y configura todos los componentes necesarios para ejecutar el programa.

Usando la terminal

Desde una terminal, ejecutamos los siguientes comandos:

- `sudo apt install git`
- `git clone https://github.com/rogama25/UBUVoiceAssistant.git`
- `cd UBUVoiceAssistant`
- `sudo ./install.sh install`

E.4. Manual del usuario

Después de haber completado la instalación, en el caso de que estemos usando Ubuntu como sistema operativo, tendremos un icono en su lanzador con el que podremos abrir nuestro programa. Si estamos usando el Windows Susbsystem for Linux o si no aparece el icono en el lanzador de aplicaciones, deberemos escribir `UBUVoiceAssistant` para ejecutar el programa.

La primera ventana que veremos es la de inicio de sesión. En la esquina superior derecha podemos elegir el idioma de la aplicación y en la parte inferior tendremos los campos donde debemos introducir los credenciales de Moodle. Deberemos especificar también la dirección web del servidor, siendo en nuestro caso `https://ubuvirtual.ubu.es`. Las direcciones deben incluir el protocolo http o https.

Una vez hayamos iniciado sesión por primera vez (puede tardar varios minutos mientras Mycroft instala algunos de sus componentes), se nos mostrará la ventana de emparejamiento. En ella aparecen los pasos que debemos realizar para vincular el cliente de Mycroft que se ha instalado en el equipo con sus servicios web. El procedimiento es el siguiente:

- Abrir en un navegador de Internet la página web de Mycroft.
- Registrarnos o iniciar sesión en una cuenta que ya tengamos.
- En la parte superior derecha, donde aparece el icono del perfil, hacer click en “Add Device”.
- Escribir el código dicho por Mycroft en el campo que pone “Pairing Code”.
- En la parte inferior, seleccionar “Google Voice” y guardar los cambios.

Pasados unos segundos, se debería abrir la ventana con la interfaz de chat. En ella se muestra la conversación que tengamos con el asistente de

voz. A partir de este punto podremos decir nuestras órdenes a través del micrófono. Al decir "Hey Mycroft", deberíamos oír un sonido que significa que ha detectado la wake word y está grabando la siguiente frase que digamos. También podemos silenciar el micrófono mediante el botón que hay en la interfaz, o escribir nuestras órdenes por teclado en el campo inferior. Haciendo click en el botón de la parte superior derecha de la ventana, podemos activar y desactivar las skills que queramos.

– Faltan imágenes –

Bibliografía