

ESTUDI I IMPLEMENTACIÓ D'UN ASSISTENT DE VEU INDEPENDENT DEL NÚVOL

Document:

Memòria

Autor:

Oriol Jorba Olmeda

Director:

Jordi Sellarès González

Titulació:

Grau en Enginyeria de Sistemes Audiovisuals

Convocatòria:

Primavera, 2021

TREBALL FINAL D'ESTUDIS

Índex

1. Introducció	6
1.1 Justificació del projecte	6
1.2 Antecedents	7
1.3 Objectius	8
1.4 Abast del projecte.....	8
1.5 Descripció general.....	9
2. Dissenys	10
2.1 Metodologia emprada	10
2.2 Recursos utilitzats	11
2.3 Instal·lació i configuració del sistema operatiu Raspbian.....	11
2.4 Instal·lació de Picroft a la Raspberry Pi	15
2.5 Instal·lació Mycroft-core en Raspbian.....	17
2.6 Comparació de Rendiment Picroft en Raspbian i la imatge de Picroft	21
2.6.1 Picroft Imager	21
2.6.2 Picroft en Raspbian.....	22
2.7 Instal·lació Mozilla DeepSpeech.....	23
2.8 DeepSpeech en mode servidor	27
2.9 Mycroft amb DeepSpeech mode servidor	30
2.10 Text-To-Speech.....	33
2.11 Configuració del Backend <i>Offline</i>	34
2.11.1 Personal Backend	34
2.11.2 Ovos-Local-Backend.....	36
2.12 Mycroft en català	40
2.12.1 DeepSpeech en Català	40
2.12.2 DeepSpeech-Català en mode servidor	42
2.12.3 Mycroft amb DeepSpeech- Català en mode servidor	44
2.12.4 Text-to-Speech en català	45
2.12.5 Mycroft en català.....	46

2.12.6 Backend amb Mycroft en català	47
2.12.7 <i>Skill</i> Coses Vivents	50
3. Resultats.....	53
3.1 Àmbit d'utilització.....	53
3.2 Validació dels dissenys	53
3.3 Descripció del funcionament.....	55
3.4 Aplicacions del projecte.....	56
4. Comentaris finals	57
4.1 Pla de treball.....	57
4.2 Llista de materials	57
4.3 Pressupost	58
4.4 Impacte mediambiental i aspectes ètics	58
4.5 Fites aconseguides	59
4.6 Conclusions.....	59
4.7 Millores futures	60
5. Bibliografia.....	61

Índex de Figures i Taules

Figura 1 Connexions de la Raspberry Pi.....	9
Figura 2 Procés ordre/resposta amb Mycroft	10
Figura 3 Interfície de Raspberry pi Imager	12
Figura 4 Fitxers creats al Imager	13
Figura 5 Connexió amb la Raspberry Pi per SSH	14
Figura 6 Escriptori de Raspbian.....	15
Figura 7 Interfície Picroft.....	16
Figura 8 Configuració del dispositiu a Mycroft IA	17
Figura 9 Interfície de Mycroft en Raspbian.....	19
Figura 10 Resposta de Mycroft a petició de l'hora	20
Figura 11 Resposta de Mycroft a petició de la temperatura	20
Figura 12 Prova amb Imatge de Picroft.....	21
Figura 13 Prova amb Picroft en Raspbian.....	23
Figura 14 Configuració de als.conf.....	25
Figura 15 Primera prova amb DeepSpeech	26
Figura 16 Segona prova amb DeepSpeech	27
Figura 17 Interfície de DeepSpeech en mode servidor	29
Figura 18 Primera prova amb DeepSpeech com a servidor	30
Figura 19 Resposta del servidor a l'origen	30
Figura 20 Configuració de Mycroft.json.....	31
Figura 21 Primera prova de Mycroft amb DeepSpeech en mode servidor	32
Figura 22 Segona prova de Mycroft amb DeepSpeech en mode servidor.....	32
Figura 23 Tercera prova de Mycroft amb DeepSpeech en mode servidor.....	33
Figura 24 Ovos-Local-Backend en funcionament.....	39
Figura 25 Funcionament de Mycroft totalment offline.....	39
Figura 26 Prova del funcionament de DeepSpeech-Català.....	42
Figura 27 Servidor de DeepSpeech en català.....	43
Figura 28 Resposta del servidor DeepSpeech en català.....	44
Figura 29 Primera prova de Mycroft amb DeepSpeech en català	44
Figura 30 Canvi d'idioma de Mycroft.....	45
Figura 31 Configuració de Festival en Mycroft.conf	46
Figura 32 Primera prova de Mycroft en català.....	47
Figura 33 Segona prova amb Mycroft en català.....	47
Figura 34 Primera prova amb Mycroft en català offline	50

Figura 35 Configuració fitxer __init__.py de la skill coses-vivents	51
Figura 36 Modificació de la classe LivingThings	52
Figura 37 Comprovació del funcionament de Coses Vivents	52
Taula 1 Comprovació del funcionament.....	54
Taula 2 Diagrama de Gantt.....	57
Taula 3 Pressupost material	58
Taula 4 Pressupost total	58

1. Introducció

1.1 Justificació del projecte

Últimament, els assistents de veu digitals formen part del nostre dia a dia, ja que han rebut un gran desenvolupament i investigació per part de grans empreses relacionades amb el sector digital. Igual que el teclat o les pantalles, són un recurs per comunicar-se amb el dispositiu que facilita la interacció amb l'usuari en certes situacions en que altres formes de comunicació màquina-usuari no son bones com podria ser conduint o cuinant, situacions en les que no es tenen les mans lliures. És una tecnologia que porta molts anys en procés de desenvolupament per no ha sigut fins la última dècada que ha aparegut en món quotidià, ja que havia estat frenada per la dificultat de reconèixer veu i, fins cert punt, de produir-la.

Juntament amb mòbils, ordinadors, tablets,... els assistents ens permeten millorar la nostra experiència, connectant els artefactes llistats anteriorment, i molts més, entre si, i a la vegada creant connexions entre aplicacions, la qual cosa ens permet sol·licitar qualsevol informació o ordre quan vulguem¹. Tot el que anteriorment havíem de teclejar o buscar navegant per internet pel nostre compte, ara és tan fàcil com demanar-li al assistent de veu que busqui per nosaltres tota la informació que necessitem; o faci les accions necessàries, sempre amb limitacions ja que no deixa de ser una Intel·ligència artificial.

Com s'ha dit anteriorment, grans empreses són les que aposten cada dia més per millorar aquest sector i a la vegada treure'n un benefici, ja sigui econòmic o per recol·lectar dades dels seus usuaris.

Veient aquesta situació, és interessant fer un estudi per poder trobar un assistent de veu independent al núvol, que pugui operar sense estar connectat a Internet, puix que li donaria a l'usuari la oportunitat de poder utilitzar un assistent de veu en situacions en que no és possible la connexió a la xarxa, com en un túnel per exemple. Una altra avantatge de fer-lo *offline* és la seguretat de saber que no hi ha cap individu o empresa en el servidor mirant el que un pregunta i sol·licita, és a dir, és més privat. Això implica que els diferents servidors s'executin de manera local al dispositiu i disposin de totes les dades necessàries per operar. D'aquesta manera s'aconseguirà un assistent de veu capaç d'operar *offline* i evitant donar informació a servidors remots.

¹Asistente de voz, una gran revolución tecnológica. https://www.elespanol.com/imprescindibles/20191231/asistente-voz-gran-revolucion-tecnologica/452705612_0.html. Accessed May 6, 2021.

1.2 Antecedents

Un assistent de veu és un software que permet a un usuari consultar, ordenar i configurar comandes per interaccionar amb un propi dispositiu o altres. Tot això amb una interacció home-màquina. Es basa en una comunicació verbal per part de la persona que la màquina processa, interpreta i respon de la mateixa forma.

Com molts altres softwares o dispositius, els assistents de veu tenen un origen militar. Podríem dir que el primer reconeixement de veu realitzat per una màquina va ser l'any 1962, quan IBM va presentar el IBM Shoebox. Aquesta va ser una espècie de computadora que era capaç de reconèixer fins a setze paraules parlades i els dígit del 0 al 9. Al llarg dels anys la tecnologia ha anat avançant permetent-nos avui en dia poder accedir a diferents assistents de veu com són Alexa (Amazon) o la pròpia Siri (Apple), els quals poden interactuar de forma complexa i intel·ligent amb el usuari.

Avui en dia, 4 són les empreses que més aposten per aquesta tecnologia. La primera va ser Apple, amb la seva peculiar i reconeguda Siri l'any 2012. Google va ser la següent en apostar per aquesta revolució, donant a conèixer Google Now que més tard es coneixeria com a Google Assistant. Al 2015, Microsoft va donar a conèixer a Cortana, la qual es pot trobar en la majoria d'ordinadors que tenen un sistema operatiu de Microsoft. Finalment, també el 2015, va aparèixer a través de Amazon, Alexa, la qual és utilitzada per una gran quantitat d'usuaris ja que és un dispositiu domèstic independent activat per veu.

Aquesta evolució constant va portar a la creació del assistent de veu en el que es basa aquest projecte, Mycroft². Aquest és un software lliure i de codi obert. El codi obert facilita la col·laboració en diversos aspectes, per exemple, utilització de programes auxiliars alternatius o internacionalització. Com que els programes estan instal·lats localment encara que es descarreguin d'internet(però es pot fer una instal·lació 100% local). El seu sistema és modular, és a dir, que connecta diversos serveis de manera flexible i configurable com poden ser el *Speech-to-text* o el *Text-to-speech*. El seu model de negoci es basa en la venda de *hardware* la qual cosa fa que l'assistent en si mateix sigui "neutral" pel que fa a les aplicacions.

En resum, es pot veure que aquesta tecnologia s'ha anat millorant durant els últims anys fent que sigui avui en dia, una eina imprescindible per a molts usuaris, en el cas d'Espanya, més d'un 10% de la població. Les actualitzacions i les millores fan que cada

² The History of Mycroft: Origin Story. <https://mycroft.ai/blog/history-of-mycroft-origin-story/>. Accessed May 6, 2021.

cop, més gent es vegi atreta per el servei que brinden aquests artefactes. Si aquest creixement segueix així, ens uns anys es podria parlar dels assistents de veu com quelcom imprescindible per a les vides de la majoria de la població.

1.3 Objectius

Aquest estudi té com a finalitat principal aconseguir que l'assistent de veu Mycroft funcioni de forma independent al núvol i que pugui operar sense necessitat d'estar connectat a internet. Aquestes necessitats impliquen que els diferents servidors que permeten el funcionament de l'assistent funcionin de forma local al dispositiu i disposin de totes les dades necessàries per operar.

Per poder aconseguir aquest objectiu ens hem marcat les següents metes:

- Instal·lació local de l'assistent de veu Mycroft
- Instal·lació d'un sistema *speech-to-text* (per exemple, *deepspeech*)
- Aconseguir els fitxers de dades necessaris pel reconeixement de veu
- Instal·lació i configuració d'un "personal backend" que ens permeti replicar a nivell local algunes de les funcionalitats de *home.Mycroft.ai*.
- Instal·lació d'un sistema *text-to-speech*.
- Configuració de Mycroft de manera que utilitzi exclusivament els serveis locals.
- Realització d'un prototip d'altaveu intel·ligent.
- Programació de *skills* que funcionin offline.
- Millores diverses al sistema que demostrin la potència de l'aproximació emprada

1.4 Abast del projecte

La base d'aquest estudi és desenvolupar un assistent de veu basat en la intel·ligència artificial Mycroft amb el material i software necessari per garantir uns mínims de funcionament d'aquest programa sense connexió a internet.

Aquest estudi garantirà unes funcions bàsiques per part de l'assistent com podria ser la hora, localització o preguntes sobre persones o localitzacions famoses que es puguin trobar a Viquipèdia. Com a demostració de les possibilitats del sistema, s'intentarà

realitzar un reconeixement de veu en Català, també offline, que pugui reconèixer les preguntes gestionades per la *skill* 'Living Things'³, la qual es traduirà al català.

Finalment, es realitzaran proves per garantir el funcionament correcte tant del prototip en anglès com en català.

1.5 Descripció general

Tot el desenvolupament del projecte es veu englobat al voltant de la Raspberry Pi 4, ja que aporta unes molt bones característiques per tenir bàsicament coses d'embedded, les quals són 4GB de RAM, un processador de 64-bit, un consum de 5V/3A , dos ports HDMI, dos ports USB 2.0 i dos més 3.0, un port Gigabit Ethernet, Bluetooth 5.0, so integrat, .Però a la vegada també d'ordinador convencional per poder realitzar el projecte amb bon rendiment i amb mínims problemes. Mycroft té una versió per aquest prototip, així que la instal·lació no ha de ser cap problema.

A nivell de *hardware*, els dispositius necessaris no són complexes, puix que és necessari, apart de la Raspberry pi, un micròfon amb entrada USB, uns altaveus amb entrada Jack i un ordinador amb connexió a internet. També seria possible realitzar-ho sense el ordinador, i fer-ho amb un cable HDMI connectat a un monitor extern.

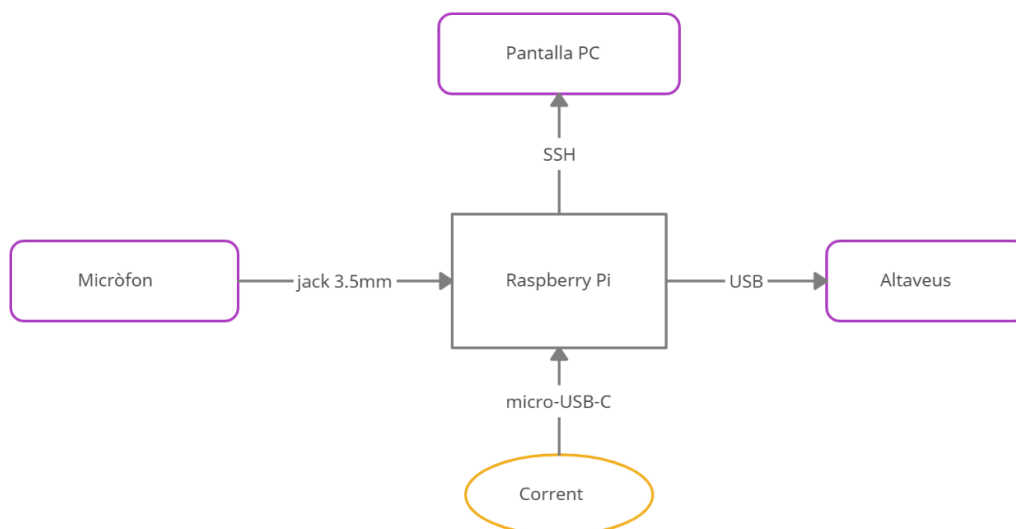


Figura 1 Connexions de la Raspberry Pi

³ Treball de fi de grau de Judith Font Muñoz <http://hdl.handle.net/2117/331119>

Inicialment, aquest assistent de veu funciona conjuntament amb els servidors de Mycroft, on envia la informació que li diem per micròfon o per teclat, la sintetitza, i retorna la informació que s'ha sol·licitat. Dins d'aquest procés, hi ha diferents accions que du a terme el software del dispositiu, les quals s'hauran de modificar. Aquestes accions són: el convertidor de veu a text (*stt*), el convertidor de text a veu (*tts*) i finalment el *backend*, el qual farà la funció de servidor local. L'ordre d'execució és el següent:

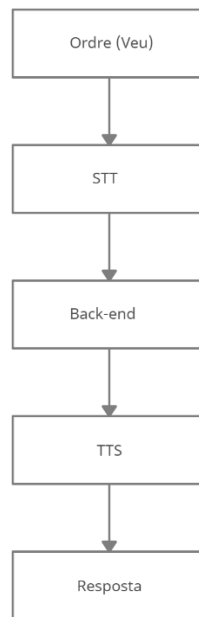


Figura 2 Procés ordre/resposta amb Mycroft

Un cop acabat el procediment de convertir aquest assistent de veu, de *online* a *offline*, es realitzaran proves per poder fer compatible aquest *software* en català, per a una certa *skill* prèviament programada.

2. Dissenys

2.1 Metodologia emprada

Com hem dit anteriorment, aquest projecte funciona al voltant de la Raspberry Pi, ja que es un dispositiu que permet la utilització de tots els perifèrics necessaris i el *software* funciona perfectament en el seu sistema operatiu. Es un dispositiu que s'utilitza per molts altres projectes i té un gran registre de documentació ja sigui escrita o en vídeos.

El primer pas serà instal·lar el sistema operatiu Raspbian a la Raspberry Pi a través d'una targeta SD, on hi copiarem un *imager* d'aquest sistema. A partir d'aquí s'instal·laran els diferents programes necessaris, sempre seguint les instruccions que recomanen els desenvolupadors per evitar qualsevol error a l'hora de compilar o engegar l'assistent de veu. Sempre pot haver-hi el cas en que un tutorial o instrucció sigui antic o erroni, per això mateix sempre consultarem la major quantitat d'opinions i recomanacions possibles davant aquests problemes. A més a més, davant de diferents camins que es puguin agafar durant el projecte, sempre s'intentarà comprovar totes les opcions possibles, com podrà ser amb els sintetitzadors de veu o els *backends*.

Durant el projecte, s'avaluarà contínuament el funcionament de les diferents parts del assistent en mode *online* i *offline*, per poder veure les manques de l'un o l'altre. Això ens permetrà avaluar els resultats del projecte respecte al servidor de Microsoft IA. A més, també s'intentarà comparar el funcionament del dispositiu en Català i Anglès i així poder veure les diferències entre l'un i l'altre.

Finalment, es revisaran tots els errors i obstacles que han anat apareixent durant el projecte per poder donar una valorització més exacte i fiable per a futur avenços o continuacions d'aquest projecte.

2.2 Recursos utilitzats

En aquest projecte s'ha treballat principalment amb la Raspberry Pi 4, però s'ha utilitzat altres recursos no comentats prèviament. El primer de tot, i probablement el més important és internet. Totes les cerques per informació i dubtes han sigut majoritàriament a través de la xarxa. A nivell de hardware també ha sigut molt important l'ordinador portàtil Lenovo ideapad 330, amb el qual ha sigut necessari per poder cercar per internet, majoritàriament amb Chrome, i guardar tota la documentació del projecte.

Finalment, els perifèrics utilitzats han sigut els altaveus amb connexió jack 3.5mm, micròfon amb sortida USB, el monitor extern al portàtil i un parell de targetes SD per instal·lar el sistema operatiu tant de Picroft com de Raspbian.

2.3 Instal·lació i configuració del sistema operatiu Raspbian

Per començar el projecte, s'ha hagut d'instal·lar el sistema operatiu Raspbian a la Raspberry Pi. Raspbian és una versió de Debian que a la vegada és una distribució de

Linux, adaptada a la Raspberry Pi. És el sistema operatiu recomanat per el fabricant per al desenvolupament de l'aplicació. El primer pas es instal·lar el Raspberry Pi Imager de la pagina web de Raspberry⁴. En el nostre cas ens hem baixat la versió per Windows, però també es accessible tant per macOS com per Ubuntu..

Un cop executat ens apareixerà la següent finestra al obrir-lo.

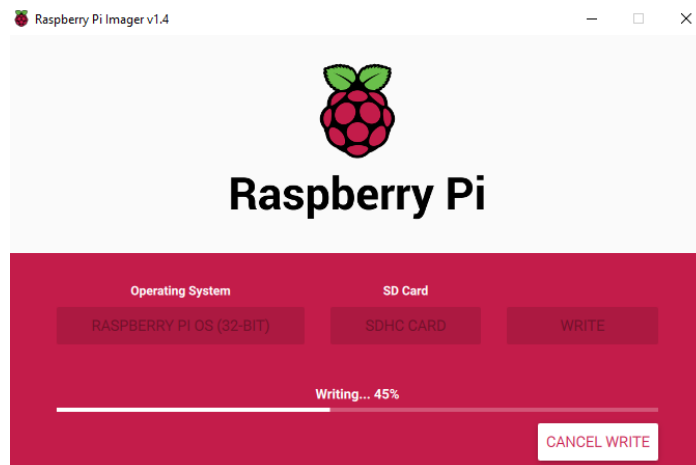


Figura 3 Interfície de Raspberry pi Imager

Per poder realitzar la instal·lació, es necessita introduir la targeta SD al ordinador per que la aplicació la detecti. Un cop introduïda, es seleccionarà el sistema operatiu, el qual serà el Raspberry Pi OS (32 bits), i la targeta SD que s'ha introduït anteriorment. Per finalitzar aquest primer procés, s'haurà de clicar al botó *write* i la imatge del sistema operatiu es començarà a instal·lar en la targeta. Un cop s'hagi instal·lat, es retirarà la targeta de l'ordinador.

Aquest projecte necessitarà la funció d'un protocol anomenat SSH, el qual realitza una administració remota a través de la xarxa, fent que es pugui compartir programes gràfics a través d'aquesta.

Per poder aconseguir aquesta interacció entre l'ordinador i la Raspberry pi, serà necessari tornar a introduir la targeta SD l'ordinador i crear un fitxer buit dintre d'aquesta amb el nom `ssh`, sense cap extensió. Es crearà un altre fitxer de text amb el nom `wpa_supplicant.conf`, al qual si que se li introduirà informació per poder connectar el dispositiu a internet de forma remota. El text és el següent:

```
country=es  
  
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
```

⁴ Teach, Learn, and Make with Raspberry Pi. <https://www.Raspberrypi.org/software/>. Accessed May 21, 2021.

```

update_config=1

network={

scan_ssid=1

  ssid="nom de la xarxa"

  psk="clau de la xarxa"

}

```

A ssid s'introduirà el nom de la nostra xarxa i a psk la contrasenya. Un cop realitzat aquest pas, hauríem de veure els dos fitxers en la targeta.











 ssh	20/12/2020 16:56	Archivo	0 KB
 start.elf	26/11/2020 17:30	Archivo ELF	2.869 KB
 start_cd.elf	26/11/2020 17:30	Archivo ELF	771 KB
 start_db.elf	26/11/2020 17:30	Archivo ELF	4.674 KB
 start_x.elf	26/11/2020 17:30	Archivo ELF	3.610 KB
 start4.elf	26/11/2020 17:30	Archivo ELF	2.162 KB
 start4cd.elf	26/11/2020 17:30	Archivo ELF	771 KB
 start4db.elf	26/11/2020 17:30	Archivo ELF	3.627 KB
 start4x.elf	26/11/2020 17:30	Archivo ELF	2.904 KB
 wpa_supplicant.conf	20/12/2020 17:02	Archivo CONF	1 KB

Figura 4 Fitxers creats al Imager

El segon pas en aquest punt és la configuració de Raspbian. Per poder fer-ho, s'ha d'introduir la targeta SD a la Raspberry PI i connectar aquesta a una font d'alimentació i paral·lelament al *router* de la xarxa local a través d'un cable *ethernet*. El *router* li donarà una direcció IP a la Raspberry. Per poder saber aquesta direcció, serà necessari instal·lar el programa nmap a través de la seva pàgina⁵. Un cop instal·lat i executat, buscarem la IP utilitzant un objectiu 192.168.1.1/24 amb un perfil *Ping scan*. En el moment que cliqui escanejar, apareixeran varies IP's, on es buscarà la corresponent a la Raspberry Pi. En el nostre cas és la 192.168.1.48.

Per continuar amb la configuració, és necessari obrir un terminal de Windows per escriure-hi les comanes indispensables per poder utilitzar Raspbian de forma remota. És tan senzill com escriure CMD al buscador de Windows.

⁵ Nmap: the Network Mapper - Free Security Scanner. <https://nmap.org/>. Accessed May 21, 2021.

Un cop obert el terminal, s'intentarà connectar amb la Raspberry a través de SSH amb la comanda `ssh pi@192.168.1.48`.

```
C:\Users\orijo>ssh pi@192.168.1.48
The authenticity of host '192.168.1.48 (192.168.1.48)' can't be established.
ECDSA key fingerprint is SHA256:3cJD0H/yd3USPsKtWQ1xJmka+ZcX5RxVwsEtlyIL18.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.48' (ECDSA) to the list of known hosts.
pi@192.168.1.48's password:
```

Figura 5 Connexió amb la Raspberry Pi per SSH

Al intentar connectar-se, demana una contrasenya la qual és *Raspberry*. Aquesta haurà de ser canviada per motius de seguretat quan s'accedeixi a la interfície gràfica. Quan s'hagi introduït aquesta clau, ja es podrà interactuar amb la màquina de forma remota.

Per poder veure la interfície gràfica, farà falta instal·lar un programa més anomenat VNC Viewer, el qual també es pot descarregar a través de la seva pàgina web oficial⁶. A continuació obrirem la configuració de la Raspberry via comanda.

```
sudo raspi-config
```

Un cop dins, s'haurà de anar a les *interfacing options* i seleccionar VNC, per poder veure a través d'aquesta opció, l'escriptori de Raspbian.

Finalment, s'ha d'obrir l'aplicació VNC Viewer i introduir la IP de la Raspberry. Demanarà un usuari, el qual es *pi*, i una contrasenya que és *Raspberry*. Després d'introduir-los ja es podrà veure la interfície gràfica de Raspbian i es podran realitzar les configuracions de l'idioma i la connexió a internet sense cable *ethernet*. Per comprovar la connexió sense fil, es pot tornar a l'aplicació nmap i veure que el *router* li ha donat una nova direcció IP a la Raspberry Pi

⁶ Descargue VNC Viewer | VNC Connect. <https://www.realvnc.com/es/connect/download/viewer/>. Accessed May 21, 2021.

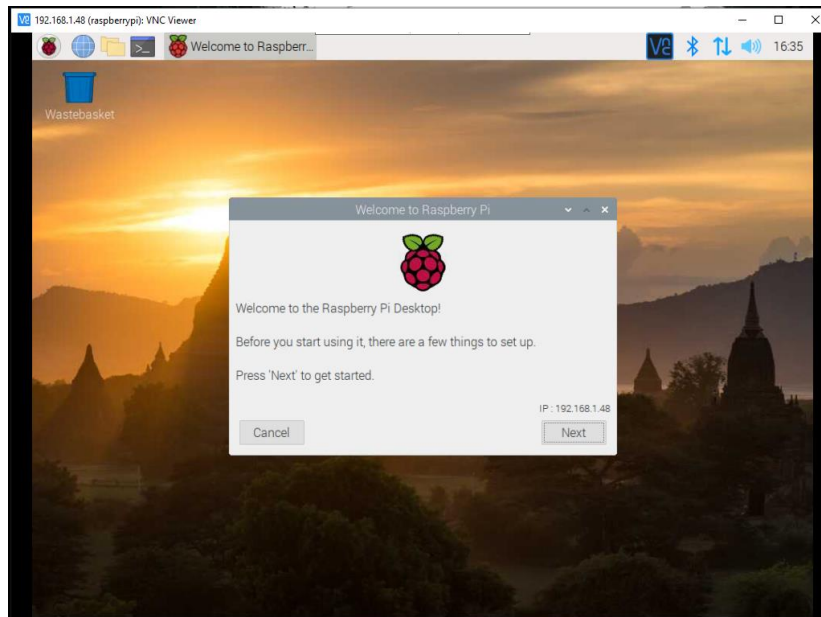


Figura 6 Escriptori de Raspbian

2.4 Instal·lació de Picroft a la Raspberry Pi

Abans de començar a interactuar amb Raspbian i Mycroft conjuntament, s'ha decidit provar el funcionament de l'assistent de veu de forma autònoma al sistema operatiu. Per poder realitzar això, el primer pas es baixar la imatge de Mycroft per a Raspberry Pi, el qual s'anomena Picroft. És tant senzill com dirigir-se a la pàgina web⁷ de Mycroft i seleccionar *Download Picroft*. Cal dir que Mycroft no ven un dispositiu per a desenvolupament, sinó que la unió de Raspberry Pi més Picroft es el sistema que recomanen per desenvolupar els *skills* de l'assistent.

Un cop descarregat, s'haurà de repetir el mateix procés realitzat amb la còpia de Raspbian a la targeta però aquest cop amb la nova imatge de Picroft. En aquest cas, la imatge de Picroft no ve per defecte amb el Raspberry Pi Imager, així que s'haurà de seleccionar del directori on s'ha baixat. En aquest cas també s'haurà de crear els fitxers *ssh* i *wpa_supplicant.conf* per poder tenir una connexió sense fil i s'haurà de connectar inicialment la Raspberry al *router* via cable *ethernet*.

A continuació, per poder accedir a Picroft, s'ha d'obrir un terminal de Windows i escriure la comanda per accedir a la Raspberry via *ssh*.

```
ssh pi@192.168.1.xx
```

⁷Get Started - Mycroft. <https://Mycroft.ai/get-started/>. Accessed May 23, 2021.

Després d'introduir la comada, demanarà una contrasenya que per defecte, en Mycroft, és *Mycroft*.

```

MYCROFT
Picroft

*****
** Picroft enclosure platform version: Buster Keaton - Pork Pi
** mycroft-core: 20.8.1 ( master )
*****

Mycroft is your open source voice assistant. Full source
can be found at: /home/pi/mycroft-core

Mycroft-specific commands you can use from the Linux command prompt:
mycroft-cli-client      Command line client, useful for debugging
mycroft-msm            Mycroft Skills Manager, to manage your Skills
mycroft-start          Launch/restart Mycroft services
mycroft-stop           Stop Mycroft services

Scripting Utilities:
mycroft-listen         Activate the microphone to listen for a command
mycroft-speak <phr>   Have Mycroft speak a phrase to the user
mycroft-say-to <utt>  Send an utterance to Mycroft as if spoken by a user

Mycroft's Python Virtual Environment (venv) control:
mycroft-pip            Install a Python package within the venv
mycroft-venv-activate Enter the venv
mycroft-venv-deactivate Exit the venv

Skill Development:
mycroft-msk            Mycroft Skills Kit, create and share Skills
mycroft-skill-testrunner Run integration tests on Mycroft Skills

Other:
mycroft-config        Manage your local Mycroft configuration files
mycroft-mic-test      Record and playback to directly test microphone
mycroft-help          Display this message

For more information, see https://mycroft.ai/documentation

```

Figura 7 Interfície Picroft

Per defecte, l'aplicació fa una configuració inicial tant del micròfon com del altaveu on s'han de seleccionar varies opcions com el tipus de sortida y entrada d'aquests dispositius. Un cop realitzada la configuració, Picroft dona un codi per poder registrar el dispositiu que estem utilitzant a la seva pàgina web, anomenada anteriorment per poder baixar la imatge del software. En aquest cas és necessari crear un compte per poder registrar el dispositiu. Quan el compte estigui creat, s'ha d'enregistrar un dispositiu, i aquí serà un s'introduirà el codi i les diferents especificacions com la localització i la zona horària.

Configure your new device

Pairing Code *

Code spoken by device

Name *

Placement

Must be unique

e.g. Kitchen, Bedroom, Office

Geographical Location

Country

Spain

Region

Catalonia

City

Terrassa

Time Zone

Europe/Madrid

Voice

British Male

American Male

Google Voice

Wake Word

Hey Mycroft

Christopher

Hey Ezra

Hey Jarvis

Figura 8 Configuració del dispositiu a Mycroft IA

Un cop el dispositiu està configurat, Picroft permetrà interaccionar amb l'assistent de veu de forma totalment fluida i sense cap problema. Se li poden fer preguntes tant de temperatura com relacionades amb localitzacions i personatges, aquest respondrà tant per veu com per text.

Per defecte, la comanda per cridar al assistent es "Hey, Mycroft", aquesta es pot canviar per altres com "Hey, Jarvis". Un cop es digui aquesta ordre, Picroft estarà preparat per escoltar l'ordre que se li vulgui donar. S'ha de tenir en compte que aquest no és un assistent de veu tan desenvolupat com Siri o Alexa, així que pot ser que alguns cops entengui malament el que se li digui, o no acabi de reaccionar del tot bé a certes comandes dites per veu.

Aquest procés és necessari per veure el funcionament de Picroft i veure els seus punts forts i els fluixos. Més endavant aquest software serà necessari per fer comparacions amb el que serà instal·lat en el sistema operatiu de Raspbian prèviament instal·lat.

2.5 Instal·lació Mycroft-core en Raspbian

Després d'instal·lar el sistema operatiu Raspbian i poder veure el funcionament de Picroft, el següent pas es combinar aquests dos passos, que és crear la nostra pròpia imatge equivalent a Picroft a partir del codi que Mycroft ha publicat.

El primer pas és tornar a introduir la targeta SD on s'havia instal·lat Raspbian i engegar la Raspberry Pi. Serà necessari obrir l'aplicació VNC i introduir la IP de la placa i la seva contrasenya per poder connectar-s'hi. Un cop dins, el següent pas es obrir un terminal i

seguir les instruccions donades per part de la pàgina web GitHub en el seu apartat de Mycroft IA⁸.

En primer lloc es crearà una clonació dels repositoris que es troben l'aplicatiu web anteriorment comentat amb les comandes necessàries.

```
cd ~/
git clone https://github.com/MycroftAI/Mycroft-core.git
cd Mycroft-core
bash dev_setup.sh
```

La comanda 'cd' permet l'usuari moure's pels directoris i fitxers del sistema escrivint, cd + directori.

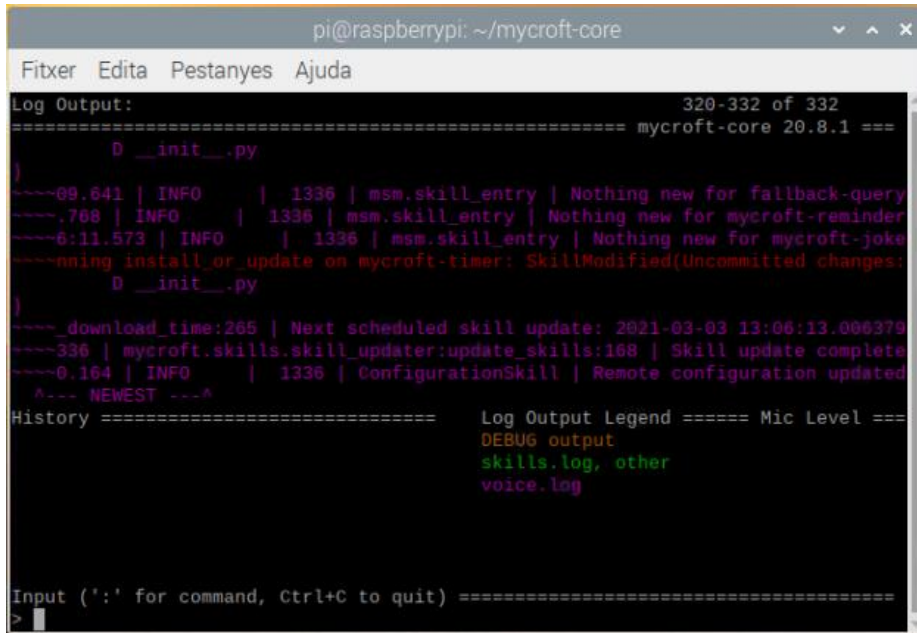
Després d'haver descarregat tots els fitxers necessaris i d'haver construït els executables per poder tenir Mycroft en el Raspbian, és molt important saber com fer-lo funcionar. La comanda per engegar-lo, és una de les més important de tot el projecte, puix que serà utilitzada reiterades vegades durant el procés.

```
cd ~/Mycroft-core
./start-Mycroft.sh debug
```

Aquest *debug* començarà a fer funcionar uns serveis en *background*, escoltant el micròfon i realitzant *skills* d'entre altres funcions. Es pot aturar aquests serveis amb una comanda que normalment no s'utilitzarà massa, ja que sempre que es vulgui sortir de la interfície de Mycroft, es farà a través de la instrucció ctrl+C, la qual ja aturarà tots aquests processos.

Un cop inicialitzat el *debug*, apareixerà una interfície molt semblant a la de la imatge de Picroft en l'altre SD. Es pot teclejar `:skills` per poder veure quines son les *skills* que estan en funcionament en aquell precís moment.

⁸GitHub - MycroftAI/Mycroft-core: Mycroft Core, the Mycroft Artificial Intelligence platform. <https://github.com/MycroftAI/Mycroft-core>. Accessed March 4, 2021.



```

pi@raspberrypi: ~/mycroft-core
Fitxer Edita Pestanyes Ajuda
Log Output: 320-332 of 332
===== mycroft-core 20.8.1 =====
  D __init__.py
}
---09.641 | INFO      | 1336 | msm.skill_entry | Nothing new for fallback-query
---.768 | INFO      | 1336 | msm.skill_entry | Nothing new for mycroft-reminder
---6:11.573 | INFO      | 1336 | msm.skill_entry | Nothing new for mycroft-joke
---oming install_or_update on mycroft-timer: SkillModified(Uncommitted changes)
  D __init__.py
}
---_download_time:265 | Next scheduled skill update: 2021-03-03 13:06:13.000379
---336 | mycroft.skills.skill_updater:update_skills:168 | Skill update complete
---0.164 | INFO      | 1336 | ConfigurationSkill | Remote configuration updated
^--- NEWEST ---^
History ===== Log Output Legend ===== Mic Level ===
          DEBUG output
          skills.log, other
          voice.log
Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 9 Interfície de Mycroft en Raspbian

Igual que en el punt anterior, s'haurà d'enregistrar el dispositiu a la pàgina web de Mycroft, amb el codi que dona per pantalla, per poder interactuar amb l'assistent de forma completa.

Per poder comprovar que funciona, s'haurà d'interaccionar a través del micròfon, així que serà necessari fer-li una pregunta que faci funcionar una *skill*, per exemple el timer, la qual s'encarrega de retornar la hora actual a la geolocalització que se li ha donat al dispositiu.

Primer es cridarà a Mycroft amb la frase per que escolti el que se li vol dir, per defecte "Hey, Mycroft" i a continuació, quan aparegui un text verd que indiqui que el dispositiu està escoltant, se li farà la pregunta: "what time is it?" que vol dir: "quina hora és?". Mycroft retornarà la resposta tant per text per pantalla com per els altaveus.

```

pi@raspberrypi: ~/mycroft-core
Fitxer Edita Pestanyes Ajuda
Log Output: 263-275 of 275
===== mycroft-core 20.8.1 =====
----tent_services.pاداتious_service:train:81 | Training... (single_thread=False)
----croft.skills.intent_services.pاداتious_service:train:83 | Training completa.
----961 | mycroft.skills.skill_manager:send:64 | New Settings meta to upload.
----ings_update:193 | All settings meta has been processed or upload has started
----manager: start settings update:195 | Skill settings downloading has started
----oft_session:get:74 | New Session Start: 027453a7-cfa8-431a-b6a5-54bb7f042df1
----F0 | 967 | __main__:handle_wakeword:71 | Wakeword Detected: hey jarvis
----44 | INFO | 967 | __main__:handle_record_begin:41 | Begin Recording...
----45.735 | INFO | 967 | __main__:handle_record_end:49 | End Recording...
---- | 967 | __main__:handle_utterance:76 | Utterance: ['what time is it']
----treemap.org/search?q=Europe%2FMadrid&format=jsonv2&addressdetails=1&limit=1
A... NEWEST ...A
History ===== Log Output Legend ===== Mic Level ==
what time is it          DEBUG output
>> Currently eleven fifty four skills.log, other
                           voice.log
Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 10 Resposta de Mycroft a petició de l'hora

També és possible cridar a la *skill* de la temperatura o *weather*, la qual retorna informació detallada de la temperatura a la localització del dispositiu.

```

pi@raspberrypi: ~/mycroft-core
Fitxer Edita Pestanyes Ajuda
Log Output: 291-303 of 303
===== mycroft-core 20.8.1 =====
----961 | QuestionsAnswersSkill | Answer from fallback-wolfram-alpha.mycroftai
Removing event fallback-query.mycroftai:QuestionQueryTimeout
Removing event fallback-query.mycroftai:QuestionQueryTimeout
---- | INFO | 961 | QuestionsAnswersSkill | Timeout occurred check responses
----61 | QuestionsAnswersSkill | Handling with: fallback-wolfram-alpha.mycroftai
----.302 | INFO | 961 | WolframAlphaSkill | Setting information for source
----F0 | 967 | __main__:handle_wakeword:71 | Wakeword Detected: hey jarvis
----96 | INFO | 967 | __main__:handle_record_begin:41 | Begin Recording...
----47.834 | INFO | 967 | __main__:handle_record_end:49 | End Recording...
----67 | mycroft.client.speech.listener:transcribe:239 | list index out of range
----eech.listener:transcribe:240 | Speech Recognition could not understand audio
A... NEWEST ...A
History ===== Log Output Legend ===== Mic Level ==
what's the weather in terrassa DEBUG output
>> The weather in Terrassa, Spain, skills.log, other
currently includes no precipitation voice.log
with no cloud cover data available
a wind speed of 0 meters per second
and a temperature of 13 degrees
Celsius
Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 11 Resposta de Mycroft a petició de la temperatura

Poden aparèixer certs problemes amb aquest software en el moment en que se li realitzen una gran quantitat de peticions seguides. Un error que ha aparegut dues vegades és un relacionat amb el bus de dades, puix que al posar Mycroft sota pressió, realitzant moltes preguntes, ha acabat col·lapsant. Es pot entendre que la culpa pugui ser de la Raspberry Pi, ja que aquesta no te la capacitat de operar com ho fa un ordinador convencional, ja que la memòria RAM és més escassa igual que la capacitat de càlcul.

També es pot donar el cas en que Mycroft no respongui a la primera a les peticions de l'usuari; és normal ja que com s'acaba de remarcar, aquest dispositiu no té la capacitat de realitzar peticions de forma òptima degut a les seves limitacions obvies. Per això a vegades s'hauran de repetir un parell de cops o més les peticions que se li facin a l'assistent.

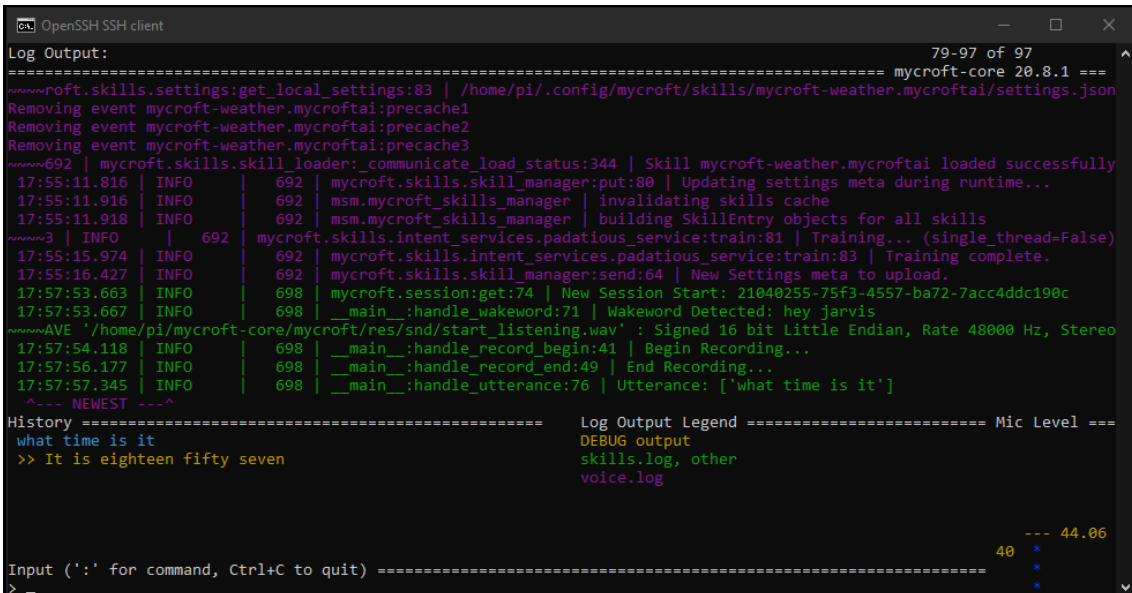
En resum, es pot comprovar que en general, Mycroft, funciona de forma correcta instal·lat en el sistema Raspbian de la Raspberry pi. Poden aparèixer alguns errors prèviament comentats, però res que compliqui severament el desenvolupament de la interacció assistent-usuari.

2.6 Comparació de Rendiment Picroft en Raspbian i la imatge de Picroft

A continuació, per veure com rendeix l'assistent de veu tant en la imatge cremada en una targeta SD com el instal·lat en el sistema operatiu Raspbian, s'han dut a terme una sèrie de proves per comparar les dues opcions.

2.6.1 Picroft Imager

La primera prova realitzada ha sigut d'acció-reacció, per veure com respon a les preguntes de l'usuari dirigides a fer reaccionar les *skills*.



```

===== mycroft-core 20.8.1 =====
mycroft.skills.settings:get_local_settings:83 | /home/pi/.config/mycroft/skills/mycroft-weather.mycroftai/settings.json
Removing event mycroft-weather.mycroftai:precache1
Removing event mycroft-weather.mycroftai:precache2
Removing event mycroft-weather.mycroftai:precache3
692 | mycroft.skills.skill_loader:communicate_load_status:344 | Skill mycroft-weather.mycroftai loaded successfully
17:55:11.816 | INFO | 692 | mycroft.skills.skill_manager:put:80 | Updating settings meta during runtime...
17:55:11.916 | INFO | 692 | msm.mycroft_skills_manager | invalidating skills cache
17:55:11.918 | INFO | 692 | msm.mycroft_skills_manager | building SkillEntry objects for all skills
693 | INFO | 692 | mycroft.skills.intent_services.padatious_service:train:81 | Training.. (single_thread=False)
17:55:15.974 | INFO | 692 | mycroft.skills.intent_services.padatious_service:train:83 | Training complete.
17:55:16.427 | INFO | 692 | mycroft.skills.skill_manager:send:64 | New Settings meta to upload.
17:57:53.663 | INFO | 698 | mycroft.session:get:74 | New Session Start: 21040255-75f3-4557-ba72-7acc4ddc190c
17:57:53.667 | INFO | 698 | __main__:handle_wakeword:71 | Wakeword Detected: hey jarvis
17:57:54.118 | INFO | 698 | __main__:handle_record_begin:41 | Begin Recording...
17:57:56.177 | INFO | 698 | __main__:handle_record_end:49 | End Recording...
17:57:57.345 | INFO | 698 | __main__:handle_utterance:76 | Utterance: ['what time is it']
^-- NEWEST ^--
History ===== Log Output Legend ===== Mic Level ===
what time is it                               DEBUG output
>> It is eighteen fifty seven                 skills.log, other
                                              voice.log
                                              --- 44.06
                                              40
Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 12 Prova amb Imatge de Picroft

En aquest cas ha tardat un parell d'intents fer-lo reaccionar a la seva frase de crida, "Hey, Mycroft", que com s'ha remarcat en el punt anterior, és una problema que es pot

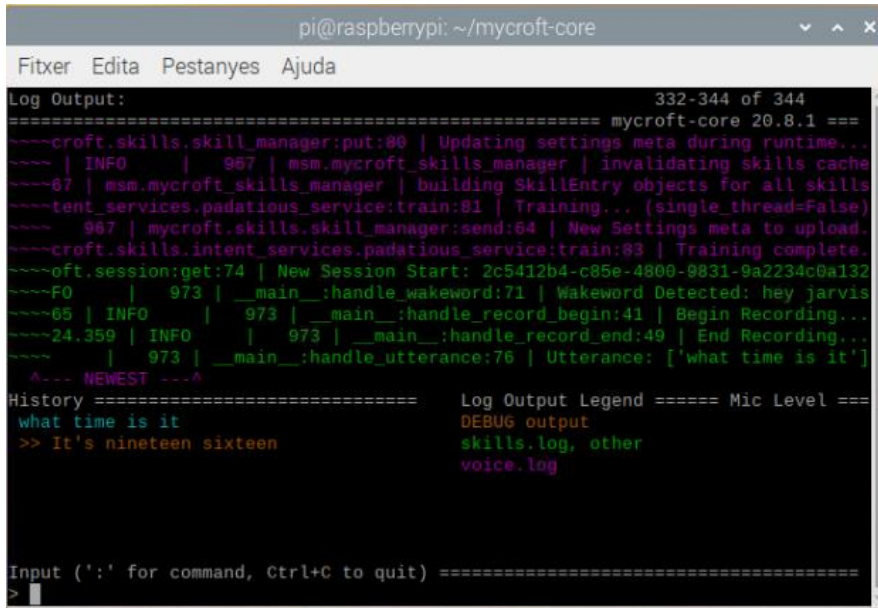
esperar i no implica grans complicacions. Un cop ha començat a escoltar, se li ha preguntat quina hora era en aquell precís instant. Com es pot veure en la figura anterior, ha escoltat perfectament el que se li ha preguntat, “what time is it?”, i la seva resposta ha sigut l’hora correcta a la zona horària del centre d’Europa. La reacció ha sigut ràpida, d’uns dos segons aproximadament. Pot passar que la sensibilitat del micròfon sigui massa alta o baixa, per aquest motiu poden aparèixer els problemes a l’hora de cridar-lo i que tardi més d’un intent en reaccionar. En el nostre cas depenent de com estigués configurada la sensibilitat, funcionava millor o pitjor, no hi ha una mesura exacta per determinar quina es la opció més bona.

Finalment, amb la sensibilitat ajustada, s’han realitzat 10 preguntes seguides de temes diferents, ja fos del temps, la hora o d’algun personatge famós. Ha respost correctament a totes les peticions al primer intent. Així que es pot treure com a conclusió que són varis els factors que determinen que funcioni millor o pitjor l’assistent a l’hora de respondre, però que intentant corregir-los dintre de lo possible, fa que el seu funcionament sigui molt millor.

2.6.2 Picroft en Raspbian

La primera diferència que es pot veure és a l’hora d’engegar la interfície, ja que en aquest cas tarda bastant més en posar a punt l’assistent, més o menys uns trenta segons.

Un cop està a punt, aquest reacciona correctament si la sensibilitat del micròfon és correcte i òptima. Se li ha fet la pregunta de l’hora com en el cas anterior i la resposta ha sigut correcte igual que ha entès perfectament el que se li ha dit.



```

pi@raspberrypi: ~/mycroft-core
Fitxer Edita Pestanyes Ajuda
Log Output: 332-344 of 344
===== mycroft-core 20.8.1 =====
~--croft.skills.skill_manager:put:80 | Updating settings meta during runtime...
~-- | INFO | 967 | msm.mycroft_skills_manager | invalidating skills cache
~--67 | msm.mycroft_skills_manager | building SkillEntry objects for all skills
~--tent_services.padatious_service:train:81 | Training... (single_thread=False)
~-- 967 | mycroft.skills.skill_manager:send:64 | New Settings meta to upload.
~--croft.skills.intent_services.padatious_service:train:83 | Training complete.
~--oft.session:get:74 | New Session Start: 2c5412b4-c85e-4800-9831-9a2234c0a132
~--F0 | 973 | __main__:handle_wakeword:71 | Wakeword Detected: hey jarvis
~--65 | INFO | 973 | __main__:handle_record_begin:41 | Begin Recording...
~--24.359 | INFO | 973 | __main__:handle_record_end:49 | End Recording...
~-- | 973 | __main__:handle_utterance:76 | Utterance: ['what time is it']
^--- NEWEST ---^
History ===== Log Output Legend ===== Mic Level ==
what time is it          DEBUG output
>> It's nineteen sixteen skills.log, other
                           voice.log
Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 13 Prova amb Picroft en Raspbian

A l'hora de fer-li les 10 preguntes també ha funcionat correctament, però sí que s'ha pogut veure que tarda una mica més en respondre que en l'altre cas, uns 2 o 3 segons més.

Una situació curiosa que apareix en ambdós casos és al moment de fer-li repetides preguntes. Si se li fa una pregunta i directament no es deixa temps amb la següent, és a dir, preguntes molt seguides, l'assistent reacciona perfectament a la primera crida. En canvi, si es deixen més de 10 segons, aquests tarda més en respondre. No hem pogut trobar una raó amb pes per descriure aquest efecte, però podria estar relacionat amb algun *timer* que manté a l'assistent.

En resum, es pot dir que el funcionament de les dues opcions és molt similar. El Picroft instal·lat en la targeta SD és més ràpid a l'hora de respondre, però no significa quelcom tan important com per decantar-se per l'un o l'altre. Pot ser que sigui degut a algun tipus de configuració de la Raspberry Pi que no s'ha dit a terme en aquest projecte.

2.7 Instal·lació Mozilla DeepSpeech

El primer pas per començar a ser més independents dels servidors de Mycroft, és trobar un reconeixedor de veu, per poder transformar les paraules que entren al micròfon en

forma d'ones, en un text exacte del que s'ha dictat. Aquest és un procés que du a terme Mycroft de forma online per defecte.

Com s'ha dit anteriorment, és necessari que tots els processos que du a terme l'assistent es realitzin de forma *offline*, així que s'ha de seleccionar un reconeixedor que es pugui adaptar a Mycroft i que es pugui executar a la Raspberry Pi. El més recomanat en la pàgina web de l'assistent de veu és el Mozilla DeepSpeech, puix que posseeix força més documentació i te opinions més positives que no pas Kaldi, un altre opció de sintetitzador de veu. També cal dir que la seva potència fa que tingui una base de dades de mostres d'àudio en diferents idiomes molt important

Un cop decidit que s'utilitzarà Mozilla DeepSpeech, s'haurà de seguir els passos indicats a GitHub⁹, on s'hauran de realitzar les següents comandes en el terminal:

```
sudo apt install git python3-pip python3-scipy python3-numpy
python3-pyaudio libatlas3-base
pip3 install deepspeech --upgrade
mkdir ~/dspeech
cd ~/dspeech
curl -LO
https://github.com/mozilla/DeepSpeech/releases/download/v0.9.3/d
eepspeech-0.9.3-models.tflite
curl -LO
https://github.com/mozilla/DeepSpeech/releases/download/v0.9.3/d
eepspeech-0.9.3-models.scorer
curl -LO
https://github.com/mozilla/DeepSpeech/releases/download/v0.9.3/a
udio-0.9.3.tar.gz
tar xvf audio-0.9.3.tar.gz
source ~/.profile
```

Es pot veure que s'ha utilitzat dues comandes diferents per instal·lar fitxers, `apt` i `pip`. La primera s'utilitza per descarregar i instal·lar paquets de repositoris del sistema(Raspbian), mentre que `pip` s'utilitza per descarregar i instal·lar paquets directament des de PyPI, el qual és un administrador de paquets de Python. El 3 es necessari per indicar que es vol utilitzar Python 3.0 o major.

⁹GitHub - touchgadget/DeepSpeech: Install Mozilla DeepSpeech on a Raspberry Pi 4. <https://github.com/touchgadget/DeepSpeech>. Accessed May 31, 2021.

Amb aquestes comandes realitzades, s'instal·larà un directori /DeepSpeech en el directori /pi, on es trobaran totes les dependències necessàries per fer funcionar el stt. També és poden transcriure tres fitxers de prova que donen els mateixos desenvolupadors.

```
deepspeech --model deepspeech-0.9.3-models.tflite --scorer  
deepspeech-0.9.3-models.scorer --audio audio/2830-3980-0043.wav
```

```
deepspeech --model deepspeech-0.9.3-models.tflite --scorer  
deepspeech-0.9.3-models.scorer --audio audio/4507-16021-0012.wav
```

```
deepspeech --model deepspeech-0.9.3-models.tflite --scorer  
deepspeech-0.9.3-models.scorer --audio audio/8455-210777-  
0068.wav
```

Després d'aquest últim pas, serà necessari modificar el fitxer `alsa.conf`, el qual assigna, entre altres coses, el micròfon. Per defecte està seleccionat el Device 0, però s'haurà de canviar al número 3, ja que estarem utilitzant un micròfon amb connexió USB i la opció de Device 3 és per a transferència d'àudio via HDMI o Jack. Les comandes per configurar-ho són les següents:

```
sudo nano /usr/share/alsa/alsa.conf
```

Un cop obert el fitxer, s'haurà de buscar les línies on digui `defaults.ctl.card 3` i `defaults.pcm.card 3` i canviar els números 0 per el número 3 quedant de la següent manera:

```
# show all name hints also for definitions without hint {} section  
defaults.namehint.showall on  
# show just basic name hints  
defaults.namehint.basic on  
# show extended name hints  
defaults.namehint.extended on  
#  
defaults.ctl.card 3  
defaults.pcm.card 3  
defaults.pcm.device 0  
defaults.pcm.subdevice -1  
defaults.pcm.nonblock 1  
defaults.pcm.compat 0  
defaults.pcm.minperiodtime 5000 # in us  
defaults.pcm.ipc_key 5678293
```

Figura 14 Configuració de `alsa.conf`

Finalment, per acabar amb aquesta instal·lació prèvia a les proves, s'han d'instal·lar els exemples de DeepSpeech.

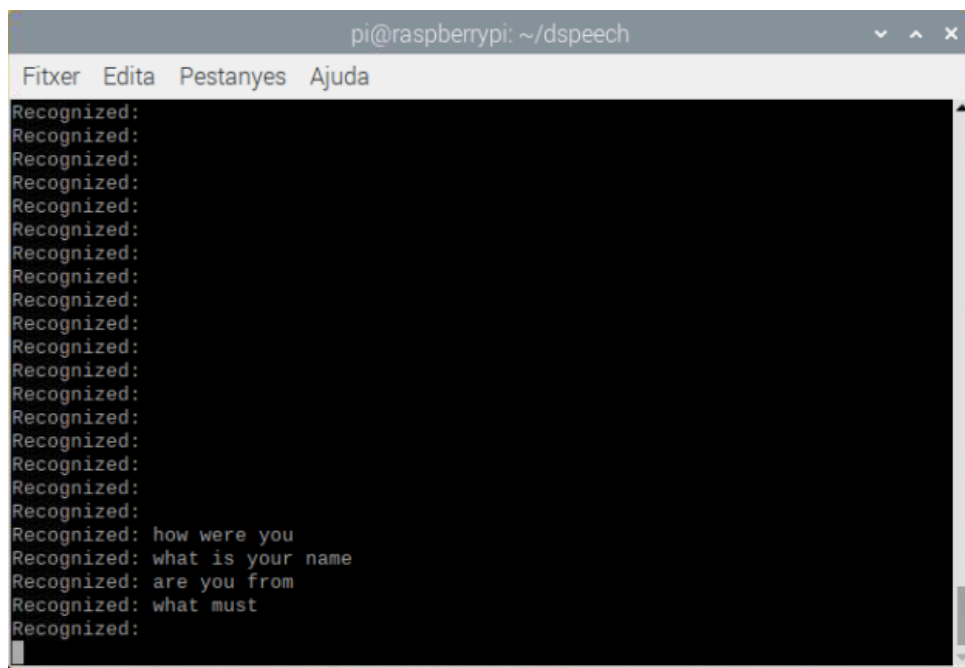
```
git clone https://github.com/mozilla/DeepSpeech-examples
```

```
pip3 install halo webrtcvad --upgrade
```

```
python3 examples/mic_vad_streaming/mic_vad_streaming.py -m deepspeech-0.9.3-models.tflite -s deepspeech-0.9.3-models.scorer
```

Amb l'última comanda apareixerà una interfície on es veurà una icona a l'esquerra. I aquesta icona va girant, voldrà dir que el micròfon està escoltat. Aquesta icona girarà només en el moment en que el micròfon detecti qualsevol so, ja sigui soroll o la veu del usuari, puix que en aquest cas, no existeix cap frase o paraula per avisar de que qui l'estigui utilitzant vagi a parlar, com a Mycroft.

A continuació, s'han realitzat una sèrie de proves per poder veure el funcionament d'aquest *software*. En algunes situacions no acaba de detectar del tot bé la frase que se li vol dir com es pot veure en la següent figura. De les quatre frases dites, només n'ha convertit bé una i les demés han tingut algun error mínim. Les quatre frases han sigut: 'how are you', 'what is your name', 'where are you from' i 'what time is it'. El resultat ha sigut el següent:



```
pi@raspberrypi: ~/dspeech
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized:
Recognized: how were you
Recognized: what is your name
Recognized: are you from
Recognized: what must
Recognized:
```

Figura 15 Primera prova amb DeepSpeech

Com es pot veure en la *Figura 15*, la primera frase no l'ha entès del tot bé, ja que ha confon *are* amb *were*. Es pot entendre aquesta equivocació, ja que són dues paraules amb força similitud. La segona frase l'ha entès a la perfecció i no ha tingut cap error. En el tercer cas, no ha captat la primera paraula de la frase. Finalment, en la última ha entès només la primera paraula i no la resta. Podem dir que a primera vista no sembla un stt massa fiable però en la tercera frase s'ha pogut veure que l'únic error que ha tingut ha

sigut no entendre o captar la primera paraula, així que hem decidit dir sempre una paraula abans de expressar la frase que vulguem, per exemple la paraula "Hello". El resultat amb aquest mètode ha sigut el següent:

```
Recognized:  
Recognized:  
Recognized: how are you  
Recognized: hello what is your name  
Recognized: where are you from  
Recognized: what time is it  
Recognized:  
Recognized:
```

Figura 16 Segona prova amb DeepSpeech

Com es pot veure en la *Figura 16*, aquest cop sí que ha entès totes les frases, puix que l'única equivocació ha sigut captar el "hello" de la segona frase, però no es tindrà en compte com a error. En principi, la obligació de dir una paraula prèvia a dir la frase, no hauria de ser necessària amb Mycroft, ja que aquest ja té la seva pròpia frase per fer que el micròfon escolti.

2.8 DeepSpeech en mode servidor

Totes les comandes per veu que se li diguin a DeepSpeech, d'alguna forma han d'arribar a Mycroft. Per això s'ha de fer funcionar DeepSpeech en mode servidor i així poder enviar totes les comandes a través de la xarxa local.

El primer pas d'aquest apartat serà instal·lar deepspeech server des de GitHub¹⁰. Totes les comandes necessàries es troben explicades perfectament en anglès i en diferents formes. En aquest cas, no farà falta instal·lar el deepspeech ja que aquest s'ha instal·lat prèviament, i s'haurà de saltar fins el pas en que s'instal·la el servidor, que és la següent comanda.

```
pip3 install deepspeech-server
```

El següent pas és crear i configurar el fitxer `config.json`. Per fer-ho es crearà aquest fitxer en el mateix directori `/dspeech` des de `/pi`.

```
cd dspeech
```

```
nano config.json
```

¹⁰GitHub - MainRo/deepspeech-server: A testing server for a speech to text service based on mozilla deepspeech. <https://github.com/MainRo/deepspeech-server>. Accessed June 2, 2021.

Quan s'obri el fitxer, serà necessari escriure-hi la configuració del servidor, que és la següent:

```
{
  "deepspeech": {
    "model" : "deepspeech-0.9.3-models.tflite",
    "scorer" : "deepspeech-0.9.3-models.scorer",
    "beam_width": 500,
    "lm_alpha": 0.931289039105002,
    "lm_beta": 1.1834137581510284
  },
  "server": {
    "http": {
      "host": "0.0.0.0",
      "port": 8080,
      "request_max_size": 1048576
    }
  },
  "log": {
    "level": [
      { "logger": "deepspeech_server", "level": "DEBUG"}
    ]
  }
}
```

Per poder comprovar que la configuració ha sigut correcte, s'han realitzat certes proves per veure que realment, DeepSpeech pot funcionar com a servidor. Primer s'ha d'instal·lar un software que permeti enregistrar àudios, com és el cas de Sox.

```
apt install sox
```

I per enregistrar l'àudio és tan senzill com escriure la següent comanda, amb una freqüència de mostratge de 16000 Hz, puix que és el mostratge que va millor amb DeepSpeech després d'haver realitzat una sèrie de proves amb altres freqüències.

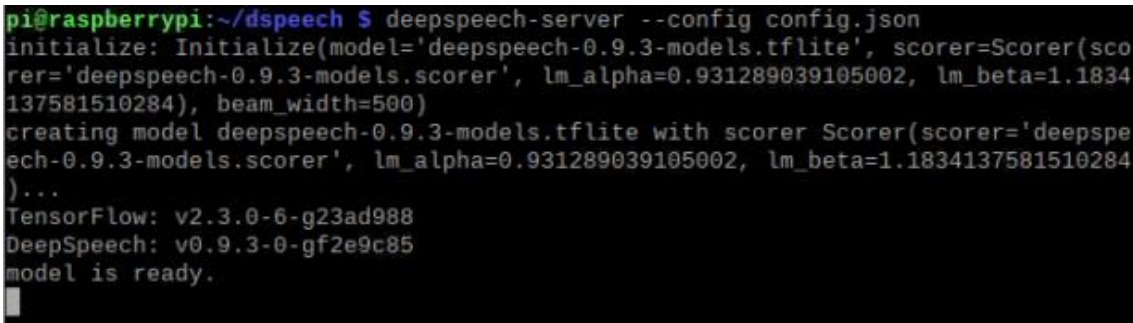
```
rec -r 16k audio.wav
```

L'àudio enregistrat haurà de ser curt, per no superar la quantitat de bytes indicada en el fitxer config.json, i clar per que deepspeech ho entengui a la perfecció.

Per començar amb la prova, seran necessàries dues pestanyes del terminal. En la primera s'obrirà deepspeech en mode servidor, amb una comanda que cridarà a la configuració creada anteriorment, i que farà funcionar el servidor en la xarxa local. La comanda es durà a terme des de el directori /dspeech.

```
deepspeech-server --config config.json
```

Un cop introduïda, apareixerà una interfície indicant que el servidor està preparat per escoltar i rebre informació.



```
pi@raspberrypi:~/dspeech $ deepspeech-server --config config.json
initialize: Initialize(model='deepspeech-0.9.3-models.tflite', scorer=Scorer(scorer='deepspeech-0.9.3-models.scorer', lm_alpha=0.931289039105002, lm_beta=1.1834137581510284), beam_width=500)
creating model deepspeech-0.9.3-models.tflite with scorer Scorer(scorer='deepspeech-0.9.3-models.scorer', lm_alpha=0.931289039105002, lm_beta=1.1834137581510284)...
TensorFlow: v2.3.0-6-g23ad988
DeepSpeech: v0.9.3-0-gf2e9c85
model is ready.
```

Figura 17 Interfície de DeepSpeech en mode servidor

Un cop els servidor estigui escoltant en el port 8080, s'haurà de realitzar una comanda CURL, en l'altre pestanya, per poder enviar la informació a través de la xarxa local. S'haurà de realitzar des de el directori on s'ha guardat el àudio anteriorment registrat.

```
curl -X POST --data-binary @audio.wav http://localhost:8080/stt
```

Al moment en que s'envia aquesta comanda, la informació del fitxer.wav viatja a través de la xarxa local per arribar a DeepSpeech, el qual s'encarrega de traduir l'àudio a text, un cop fet, retornarà el resultat a la direcció d'origen com a resposta.

```
pi@raspberrypi:~ $ cd dspeech
pi@raspberrypi:~/dspeech $ -deepspeech-server --config config.json
bash: -deepspeech-server: no s'ha trobat l'ordre
pi@raspberrypi:~/dspeech $ deepspeech-server --config config.json
initialize: Initialize(model='deepspeech-0.9.3-models.tflite', scorer=Scorer(scorer='deepspeech-0.9.3-models.scorer', lm_alpha=0.931289039105002, lm_beta=1.1834137581510284), beam_width=500)
creating model deepspeech-0.9.3-models.tflite with scorer Scorer(scorer='deepspeech-0.9.3-models.scorer', lm_alpha=0.931289039105002, lm_beta=1.1834137581510284)...
TensorFlow: v2.3.0-6-g23ad988
DeepSpeech: v0.9.3-0-gf2e9c85
model is ready.
STT error: File format b'...' not understood.
STT result: hello
```

Figura 18 Primera prova amb DeepSpeech com a servidor

La paraula gravada en el fitxer.wav era "Hello", així que es pot confirmar que el servidor ha llegit correctament el fitxer i ha pogut entendre la paraula a la perfecció. L'últim pas serà veure si l'origen ha pogut rebre la resposta correctament.

```
pi@raspberrypi:~/dspeech $ curl -X POST --data-binary @prova10.wav http://localhost:8080/stt
hellopi@raspberrypi:~/dspeech $
```

Figura 19 Resposta del servidor a l'origen

Com es pot veure en la *Figura 19*, ha rebut la paraula correctament, així que es pot afirmar que DeepSpeech en mode servidor funciona correctament. El següent pas serà comprovar aquest funcionament juntament amb l'assistent i sense tenir que dependre dels servidors de de Mycroft.

2.9 Mycroft amb DeepSpeech mode servidor

Fins el moment, l'assistent de veu ha funcionat de forma online, enviant tota la informació que rebia a través del micròfon cap als servidor de Mycroft per a que aquesta reconegui el text enviat per l'usuari, ja sigui per veu o teclat. Aquest punt serà el primer en que es veuran ja les primeres funcionalitats de Mycroft en mode *offline*. El procés és força senzill però han aparegut alguns problemes de configuració que es veuran més endavant i que podrien haver complicat el projecte de no haver deduït la causa.

Tot el procés que es du a terme en aquest pas, està també documentat pas per pas en la pàgina web de Mycroft IA¹¹. Per fer-ho, serà necessari dirigir-se al directori on es troba totes les dependències de Mycroft, i un cop allà trobar el fitxer `Mycroft.config`.

```
cd Mycroft-core/bin
./Mycroft-config edit user
```

Amb l'última comanda es podrà editar la configuració de Mycroft respecte la xarxa i modificar a quina direcció envia la informació. En el cas d'aquest projecte, es vol enviar tota la informació a la xarxa local per que DeepSpeech pugui convertir la veu de l'usuari en text. La configuració és la següent:

```
"stt": {
  "deepspeech_server": {
    "uri": "http://localhost:8080/stt"
  },
  "module": "deepspeech_server"
}
```

Com es pot veure en el codi, la direcció es la pròpia xarxa local `localhost` en el port 8080 on estarà escoltat DeepSpeech.

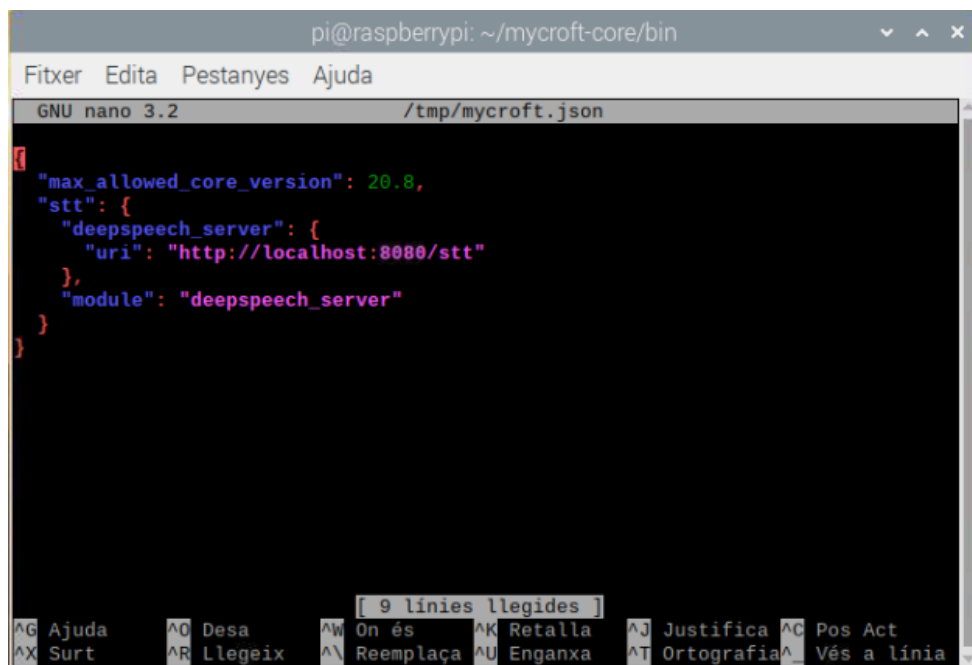


Figura 20 Configuració de Mycroft.json

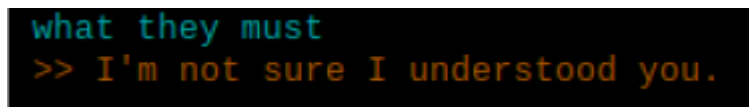
¹¹Speech-To-Text - Mycroft AI. <https://Mycroft-ai.gitbook.io/docs/using-Mycroft-ai/customizations/stt-engine>. Accessed March 16, 2021.

Un cop guardat aquest fitxer, vindrà el pas clau d'aquest punt el qual és comprovar si el funcionament és correcte. Per començar aquest comprovació, s'ha d'obrir DeepSpeech en mode servidor per que estigui escoltat en el port 8080 de la xarxa local el que li envii Mycroft. Per fer-ho, s'ha d'obrir una nova pestanya en el directori `/home/pi`, dirigir-se al directori de DeepSpeech i allà obrir el servidor. S'haurà de veure una interfície idèntica a la de la *Figura 17*.

```
cd dspeech
deepspeech-server --config config.json
```

Amb el servidor ja en funcionament, només fa falta obrir Mycroft-core en l'altre pestanya, des de el directori `/Mycroft-core` i introduir la comanda `./start-Mycroft.sh debug` per posar en funcionament l'assistent.

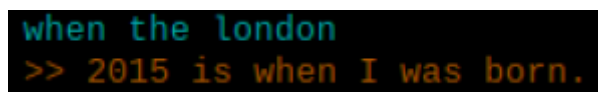
Un cop obert i amb Mycroft funcionant, se li farà una petició a l'assistent com per exemple quina hora és, en anglès "What time is it?".



```
what they must
>> I'm not sure I understood you.
```

Figura 21 Primera prova de Mycroft amb DeepSpeech en mode servidor

Com es pot veure, hi ha hagut un problema de comprensió, puix que la frase que ha entès per part de l'usuari ha sigut incorrecte i sense cap sentit. Podria ser un error puntual, ja que com s'ha dit anteriorment, aquest no és un assistent de veu tant desenvolupat com altres de més renom en el mercat. Així que la segona petició de prova ha sigut, "on està Londres?", en anglès "where is London?"



```
when the london
>> 2015 is when I was born.
```

Figura 22 Segona prova de Mycroft amb DeepSpeech en mode servidor

Com es pot veure en la *Figura 22*, no ha entès la petició correctament ja que no te res a veure amb la pregunta realitzada per l'usuari i a més la resposta tampoc és la indicada amb aquesta pregunta. Resumint, aquest és un cas bastant caòtic i el qual podria donar bastants problemes a curt termini.

La primera opció i la més probable, seria un error amb la freqüència de mostratge, puix que era un paràmetre que s'introduïa en la comanda per enregistrar els fitxers.wav per després enviar-los al servidor i fer les proves del DeepSpeech en mode servidor. Aquest

fitxers s'enviaven amb una Fm de 16 kHz, així que s'ha de comprovar amb quina freqüència esta configurat Pulse, el programa que gestiona la l'àudio del dispositiu.

Per poder accedir a la configuració de Pulse, el primer pas es dirigir-se al directori on es troba la configuració d'aquest programa. Un cop allà obrir el document de configuració i modificar-lo.

```
cd /etc/pulse
```

```
sudo nano daemon.conf
```

Un cop dins del fitxer s'hauran de buscar les següents línies de codi:

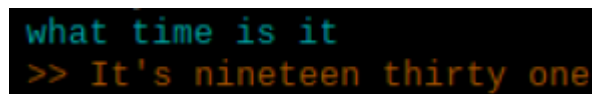
```
default-sample-format = s16le
```

```
default-sample-rate = 44100
```

```
alternate-sample-rate = 48000
```

On posa `alternative-sample-rate` s'haurà de canviar per 16000. D'aquesta manera quan Mycroft llegeixi fitxers provinents de DeepSpeech els podrà analitzar correctament i no farà falta canviar la `default-sample-rate`, què és la que funciona per defecte.

Després d'editar el fitxer, s'ha tornat a comprovar el funcionament de l'assistent i el resultat ha sigut el següent:



```
what time is it  
>> It's nineteen thirty one
```

Figura 23 Tercera prova de Mycroft amb DeepSpeech en mode servidor

Com es pot veure, aquest cop si que ha funcionat correctament, així que la configuració es deixarà tal i com s'ha modificat anteriorment. Amb el reconeixement de veu vinculat el següent pas serà el *text-to-speech*.

2.10 Text-To-Speech

En el cas del procés de convertir el text en veu de forma *offline*, és força més senzill que el procés de convertir la veu en text.

En un inici vàrem pensar en algunes opcions per fer aquesta funció. El stt que més s'indicava per internet en general era el Mimic. El que no sabíem és què aquest ja venia inclòs amb Mycroft per defecte. Així que si en la configuració del dispositiu en la pàgina

web de Mycroft s'indicava que es volia la veu de *UK Male*, aquest ja funcionaria de forma *offline*.

Per això mateix, l'únic canvi que s'ha de fer és anar a la configuració del dispositiu a Mycroft IA i indicar que es vol la veu de *UK Male*. D'aquesta forma ja es podrà operar amb un *speech-to-text* totalment fora dels servidors de l'assistent.

2.11 Configuració del Backend *Offline*

El pas final per a que Mycroft funcioni de forma totalment *offline*, és instal·lar un *backend* per a que funcioni de forma independent dels servidor de l'assistent. El *backend* és la part del sistema que s'ocupa de l'accés a les dades i de la gestió d'aquestes, però en el nostre cas no durà a terme totes les funcions que feien els servidors de Mycroft, només les necessàries per a que Mycroft pensi que està intercanviant dades amb els servidors al núvol quan en realitat ho fa amb el *backend* local. En aquest projecte s'han investigat dues opcions per poder dur a terme aquest procés. Una de les opcions és Personal-Backend i l'altre OVOS-Local-Backend. Cal remarcar que a diferència d'altres assistents de veu, Mycroft instal·la els *skills* de manera local, la qual cosa fa possible que es puguin utilitzar de forma *offline*.

2.11.1 Personal Backend

Personal Backend serà la primera opció en ser provada, cal dir que en un principi, segons les opinions vistes per internet, OVOS és més fiable, per això es deixarà en segon lloc ja que es començarà amb la solució proposada per l'empresa. Cal dir que aquest primer, encara està en desenvolupament i és normal que el seu funcionament no sigui òptim.

Per instal·lar-lo a Raspbian cal seguir les instruccions donades en el GitHub¹² de Mycroft IA.

```
git clone https://github.com/MycroftAI/personal-backend
cd personal-backend
pip3 install .
```

¹² 1. GitHub - MycroftAI/personal-backend: WORK IN PROGRESS: A Flask personal backend alternative for running your own version of <https://home.Mycroft.ai>. <https://github.com/MycroftAI/personal-backend>. Accessed June 13, 2021.

Un cop instal·lats tots els fitxers i dependències necessàries, s'ha de crear un document de configuració per al motor de la següent manera.

```
mkdir Mycroft-core/Mycroft/personal_backend
```

```
nano personal_backend.conf
```

Dins del fitxer hi anirà el següent codi:

```
{  
  "backend_port": 6712,  
  "ssl": true,  
  "ssl_key":  
  "/home/user/.Mycroft/personal_backend/certs/MycroftPersonalServer.key",  
  "ssl_cert":  
  "/home/user/.Mycroft/personal_backend/certs/MycroftPersonalServer.crt",  
  "mail_port": 465,  
  "mail_server": "smtp.googlemail.com",  
  "mail_user": "xxx@gmail.com",  
  "mail_password": "xxx"  
}
```

Aquí s'especifica el port d'entrada i sortida de dades del motor com la configuració del ssl.

El següent pas es canviar la configuració de Mycroft per a que deixi d'enviar i rebre dades del núvol i faci totes les operacions de forma local amb el Personal Backend. És necessari dirigir-se a la configuració de l'assistent per realitzar els canvis.

```
nano /Mycroft-core/Mycroft/configuration/Mycroft.conf
```

Dins del fitxer només es modificarà la secció del servidor, puix que és l'encarregada de definir la direcció IP on s'envien les dades.

```
// Address of the REMOTE server  
// Override: none  
"server": {  
  "url": "http://0.0.0.0:6712",  
  "version": "v1",  
  "update": true,
```

```
"metrics": true  
},
```

Amb la configuració realitzada, només cal posar a funcionar el backend, el DeepSpeech-servidor i Mycroft a la vegada. Per poder engegar el Personal Backend, primer s'ha de crear un fitxer en el mateix directori amb el codi necessari per fer-lo funcionar.

```
Nano start_personal_server.py
```

Dins del fitxer va el següent codi:

```
from personal_Mycroft_backend.backend import start_backend  
  
start_backend()
```

Un cop configurat només queda realitzar la comanda per iniciar-lo.

```
python3 start_personal_server.py
```

Al realitzar certes proves per veure si funcionava bé o no, aquest funciona de forma força limitada puix que algunes vegades no acaba de inicialitzar Mycroft de forma correcta o tarda molt en captar el que se li està dient el usuari. A la vegada, quan ha aconseguit captar el que se li deia, la resposta ha sigut molt lenta.

2.11.2 Ovos-Local-Backend

El segon motor backend en ser provat és el Ovos-local-backend. El seu funcionament és molt semblant al de l'anterior igual que la seva configuració, puix que Personal-Backend és un *fork* d'aquest, d'aquí la similitud. Això passa gràcies a que és un ecosistema *open source*. Per instal·lar-lo, s'han de seguir els passos especificats en el GitHub¹³ de JarbasIA. En aquesta web s'indica que es pot descarregar directament amb pip, però ho realitzarem des de la font ja que és més fiable.

```
git clone https://github.com/OpenVoiceOS/OVOS-local-backend  
  
cd OVOS-personal-backend  
  
pip3 install .
```

Com en el Personal-backend, també s'ha de crear un fitxer de configuració propi del motor en un directori que s'ha de crear.

```
cd ./config
```

¹³ 1. GitHub - OpenVoiceOS/OVOS-local-backend: WIP - flask personal backend alternative to home.Mycroft. <https://github.com/OpenVoiceOS/OVOS-local-backend>. Accessed June 13, 2021.

Mkdir json_database

Nano ovos_backend.json

Dins del fitxer, s'escriurà el següent codi:

```
"stt": {
  "module": "google"
},
"backend_port": 6712,
"geolocate": false,
"override_location": false,
"api_version": "v1",
"data_path": "~",
"record_utterances": false,
"record_wakewords": false,
"wolfram_key": "BUNDLED_DEMO_KEY",
"owm_key": "BUNDLED_DEMO_KEY",
"default_location": {
  "city": {
    "code": "Lawrence",
    "name": "Lawrence",
    "state": {
      "code": "KS",
      "name": "Kansas",
      "country": {
        "code": "US",
        "name": "United States"
      }
    }
  }
},
"coordinate": {
  "latitude": 38.971669,
  "longitude": -95.23525
},
```

```
"timezone": {  
  "code": "America/Chicago",  
  "name": "Central Standard Time",  
  "dstOffset": 3600000,  
  "offset": -21600000  
}  
}  
}
```

Com es pot veure, el port d'escolta del backend és el mateix en el que opera Personal Backend.

Després de crear aquest document, el següent pas és configurar la configuració de Mycroft.conf.

```
cd
```

```
nano Mycroft-core/Mycroft/configuration/Mycroft.conf
```

S'haurà de modificar la part del server per que aquest funcioni amb Ovos i no amb el servidor de Mycroft IA.

```
"server": {  
  "url": "http://127.0.0.1:6712",  
  "version": "v1",  
  "update": true,  
  "metrics": true,  
  "sync_skill_settings": true  
},
```

Amb aquesta configuració, el funcionament de Mycroft amb Ovos hauria de ser ja possible. Pot ser que apareguin certs errors relacionats amb el SSL, per això s'han de crear uns certificats en un directori a crear.

```
mkdir ssl
```

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem  
-days XXX
```

On XXX és un número enter qualsevol.

Per encendre el motor, cal escriure la següent comanda des de el directori de Ovos, DeepSpeech en mode servidor i finalment Mycroft.

```
ovos-local-backend
```

Un cop tot obert i funcionant, es podrà veure com el *backend* va fent les seves funcions de GET i PUT, per rebre i retornar dades.

```

19:08:33.393 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:08:33] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/location HTTP/1.1" 200 -
19:08:33.408 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:08:33] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/location HTTP/1.1" 200 -
19:08:33.414 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:08:33] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/location HTTP/1.1" 200 -
19:08:43.265 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:08:43] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/skill/settings HTTP/1.1" 200 -
19:09:33.418 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:09:33] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/setting HTTP/1.1" 200 -
19:09:33.438 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:09:33] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/location HTTP/1.1" 200 -
19:09:43.286 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:09:43] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/skill/settings HTTP/1.1" 200 -
19:10:33.504 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:10:33] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/setting HTTP/1.1" 200 -
19:10:33.519 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:10:33] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/location HTTP/1.1" 200 -
19:10:43.306 - werkzeug - INFO - 127.0.0.1 - - [14/Jun/2021 19:10:43] "GET /v1/
device/e461d22c-804b-4adb-91e8-a386dd789eb4/skill/settings HTTP/1.1" 200 -
  
```

Figura 24 Ovos-Local-Backend en funcionament

Dins de Mycroft, es farà la comprovació de que tot funciona correctament de forma conjunta tant amb el motor *backend* com amb el stt.

```

Log Output: 121-130 of 130
===== mycroft-core 21.2.0 =====
~~~| INFO      | 3888 | msm.mycroft_skills_manager | invalidating skills cache
~~~8 | msm.mycroft_skills_manager | building SkillEntry objects for all skills
~~~4.943 | INFO      | 3888 | __main__:on_ready:185 | Skills service is ready.
~~~ent_services.padatious_service:train:81 | Training... (single_thread=False)
~~~roft.skills.intent_services.padatious_service:train:83 | Training complete.
~~~ 3888 | mycroft.skills.skill_manager:send:66 | New Settings meta to upload.
~~~ngs_update:197 | All settings meta has been processed or upload has started
~~~anager:_start_settings_update:199 | Skill settings downloading has started
  ^--- NEWEST ---^
History ===== Log Output Legend ===== Mic Level ==
what time is it          DEBUG output          115 -* - 147.22
>> twelve thirty three  skills.log, other
                          voice.log
                          *
                          *
                          *
                          *
                          *
                          *
Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 25 Funcionament de Mycroft totalment offline

La pregunta ha sigut “what time is it” i com es pot veure, ha contestat correctament dient l'hora de la zona que té configurada. El temps de reacció és molt ràpid i no tarda gens en respondre en comparació amb el motor anterior.

2.12 Mycroft en català

L'última part d'aquest projecte té un objectiu molt clar; poder fer funcionar la *skill* Living Things en català juntament amb Mycroft *offline*, per demostrar la potència d'aquesta aproximació. La idea és configurar una nova targeta SD amb Raspbian de la mateixa manera que en el punt 2.3.

Un cop aquesta targeta tingui configurat el sistema operatiu Raspbian, amb connexió a internet de forma remota, serà necessari instal·lar l'assistent de veu de Mycroft en Raspbian, Picroft. El procés d'instal·lació es troba descrit en el punt 2.4 d'aquest projecte i no hi ha canvis respecte aquest punt.

2.12.1 DeepSpeech en Català

Com s'ha dit en el punt anterior, s'ha d'aconseguir que Mycroft pugui reconèixer una *skill* en català, per això hi ha tres punts molt importants i diferents a la instal·lació per defecte en anglès que són: un veu-a-text en català, un text-a-veu en català i configurar Mycroft de forma que pugui processar la informació en català.

El primer punt a resoldre és el stt en català. En aquest cas, s'ha escollit deepspeech-català, que es bàsicament Mozilla DeepSpeech però amb la capacitat de poder reconèixer veu en català. Per instal·lar-lo, és necessari escriure les següents comandes les quals baixaran tots els arxius necessaris del GitHub¹⁴ de DeepSpeech-Català.

```
git clone https://github.com/ccoreilly/deepspeech-catala
```

Com és pot veure no és la mateixa comanda que recomana la pàgina web, però al fer-ho com ho deien ells, no s'instal·lava correctament, així que s'ha passat directament a aquest mètode. Per comprovar si el programa funciona, s'ha de escriure la següent comanda on prèviament s'ha d'haver gravat un fitxer.wav amb una frase en català, com per exemple “Bon dia”.

¹⁴GitHub - ccoreilly/deepspeech-catala: Deepspeech ASR Model for the Catalan Language. <https://github.com/ccoreilly/deepspeech-catala>. Accessed June 7, 2021.


```
deepspeech --model deepspeech-catala.pbmm --scorer kenlm.scorer -  
-audio file.wav
```

En el primer intent hi ha hagut un error relacionat amb els fitxers .pbm i .scorer. Primer de tot, s'ha de substituir el fitxer .pbm per un .tflite, ja que degut a opcions de conversió de models, funciona millor aquest tipus de fitxer.

Per baixar-los, es crea un directori nou en el directori /pi, amb el nom stashify, per reconèixer que s'han baixat d'aquest GitHub¹⁵. S'utilitza la comanda `wget` ja que permet baixar fitxers d'una pàgina web només indicant el url de la localització del fitxer.

```
Cd
```

```
mkdir stashify
```

```
Cd stashify
```

```
wget https://storage.cloud.google.com/stashify-  
static/deepspeech\_cat/output\_graph.pbmm
```

```
wget https://storage.googleapis.com/stashify-  
static/deepspeech\_cat/kenlm.scorer
```

Un cop realitzades aquestes comandes, s'ha de tornar a realitzar l'ordre per fer funcionar DeepSpeech-Català amb un fitxer.wav, però indicant els nous fitxers instal·lats de stashify.

```
deepspeech --model output_graph.tflite--scorer kenlm.scorer --  
audio audio.wav
```

La frase gravada en audio.wav ha sigut "bon dia" i el resultat ha sigut el següent:

¹⁵GitHub - stashify/deepspeech_cat: Deepspeech Catalan. https://github.com/stashify/deepspeech_cat. Accessed June 7, 2021.

```

pi@raspberrypi:~/stashify $ deepspeech --model output_graph.tflite --scorer ke
lm.scorer --audio audio.wav
Loading model from file output_graph.tflite
TensorFlow: v1.15.0-24-gceb46aa
DeepSpeech: v0.7.0-0-g3fbbca2
Loaded model in 0.00214s.
Loading scorer from files kenlm.scorer
Loaded scorer in 0.000522s.
Warning: original sample rate (48000) is different than 16000hz. Resampling mi
ght produce erratic speech recognition.
Running inference.
bon dia
Inference took 1.633s for 1.963s audio file.
  
```

Figura 26 Prova del funcionament de DeepSpeech-Català

Com es pot veure en la *Figura 23*, la resposta que ha donat és “bon dia”, així que el funcionament és correcte, i es pot passar al següent pas que seria fer-lo funcionar en mode servidor.

2.12.2 DeepSpeech-Català en mode servidor

Com en l'apartat 2.8, per que Mycroft pugui fer servir DeepSpeech, és necessari que aquest funcioni en mode servidor. Així que serà necessari seguir els mateixos passos indicats en l'apartat que s'acaba de comentar.

```

Pip3 install deepspeech-server
Cd dspeech
Nano config.json
  
```

En el moment que s'obre el fitxer `config.json`, aquest cop el codi serà diferent, puix que el model i el scorer seran els descarregats des de stashify.

```

{
  "deepspeech": {
    "model" : "output_graph.tflite",
    "scorer" : "kenlm.scorer",
    "beam_width": 500,
    "lm_alpha": 0.931289039105002,
    "lm_beta": 1.1834137581510284
  },
  "server": {
    "http": {
      "host": "0.0.0.0",
      "port": 8080,
    }
  }
}
  
```

```
"request_max_size": 1048576  
}
```

Després de realitzar aquest últim pas, cal obrir dues pestanyes en el terminal. En la primera, s'obrirà el DeepSpeech en mode servidor. Pot ser que aquest doni un error de llibreria, per es pot solucionar amb la següent comanda:

```
sudo apt-get install libatlas-base-dev
```


Un cop solucionat el problema, quedarà engegat el servidor com es feia en el apartat 2.8. Aquest s'haurà de cridar des de el directori /stashify, ja que és on es troben els fitxers model i scorer, així que s'haurà de realitzar una drecera al fitxer `config.json` de DeepSpeech per que pugui detectar-lo.

```
ln -s cd ../Mycroft/config.json ./  
deepspeech-server --config config.json
```

Si tot ha sortit correctament, haurà d'aparèixer la frase "model is ready", que voldrà dir que el servidor està escoltant en el port 8080 de la xarxa local.

En l'altre pestanya, s'introduirà la comanda necessària per poder enviar un fitxer.wav al servidor per que aquest el desxifri i retorni el resultat per pantalla. Es pot fer des de el directori /stashify, puix que aquest ja conté un audio.wav amb la frase "bon dia".

```
Rec -r 16k audio.wav  
curl -X POST --data-binary @audio.wav http://localhost:8080/stt
```



```
pi@raspberrypi:~/stashify $ deepspeech-server --config config.json  
initialize: Initialize(model='output_graph.tflite', scorer=Scorer(scorer='ken  
.scorer', lm_alpha=0.931289039105002, lm_beta=1.1834137581510284), beam_width=  
30)  
creating model output_graph.tflite with scorer Scorer(scorer='kenlm.scorer',  
alpha=0.931289039105002, lm_beta=1.1834137581510284)...  
TensorFlow: v1.15.0-24-gceb46aa  
DeepSpeech: v0.7.0-0-g3fbbca2  
model is ready.  
STT result:  
STT result: bon dia
```

Figura 27 Servidor de DeepSpeech en català

Com es pot veure en l'anterior figura, DeepSpeech ha entès correctament la frase així que no hi ha hagut problema en el procés de fer arribar el fitxer al servidor.

```

pi@raspberrypi:~/stashify $ curl -X POST --data-binary @audio.wav http://localhost:8080/stt
bon dia
pi@raspberrypi:~/stashify $
  
```

Figura 28 Resposta del servidor DeepSpeech en català

En la *Figura 25* es pot veure que el servidor ha tornat la resposta correctament, indicant que el procés ha sigut correcte.

2.12.3 Mycroft amb DeepSpeech- Català en mode servidor

Aquest apartat només és una prova per confirmar que Mycroft i DeepSpeech en català poden funcionar junts. De moment no es veurà que contesti a cap comanda, puix que l'únic objectiu d'aquest punt es vincular-los.

El procés és gairebé idèntic que en el punt 2.9, així que es seguiran els passos d'aquell apartat. Més endavant ja es modificarà l'idioma tant del stt com del tts.

```

Log Output: 129-138 of 138
===== mycroft-core 21.2.0 =====
~~~8 | msm.mycroft_skills_manager | building SkillEntry objects for all skills
~~~8.141 | INFO | 4458 | __main__:on_ready:185 | Skills service is ready.
~~~ent_services.padatious_service:train:81 | Training... (single_thread=False)
~~~ | INFO | 4464 | __main__:handle_utterance:76 | Utterance: ['bon dia']
~~~roft.skills.intent_services.padatious_service:train:83 | Training complete.
~~~0:40.064 | INFO | 4458 | QuestionsAnswersSkill | Searching for bon dia
~~~ INFO | 4458 | QuestionsAnswersSkill | Timeout occured check responses
~~~ 4458 | mycroft.skills.skill_manager:send:66 | New Settings meta to upload.
^--- NEWEST ---^
History ===== Log Output Legend ===== Mic Level ==
bon dia          DEBUG output
>> I don't know what that means. skills.log, other
voice.log

Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 29 Primera prova de Mycroft amb DeepSpeech en català

Com es pot veure en la darrera figura, DeepSpeech ha entès perfectament el que s'ha dit i ho ha enviat a Mycroft, però aquest no ha entès que vol dir, ja que no està configurat per entendre aquesta llengua.

2.12.4 Text-to-Speech en català

Després de configurar el stt, cal saltar al següent pas que es convertir el text en veu per que l'assistent pugui retornar la informació necessària. Per fer-ho, es necessita un programa que pugui i estigui configurat per poder retornar veu en català. El motor que s'ha escollit es el festival, ja que en comparació amb altres, la seva qualitat és la millor.

En un inici es va intentar treballar descarregant una extensió de festival en català, el seu nom es Festcat¹⁶ i esta desenvolupada pel grup de tecnologies de la parla de la Universitat Politècnica de Catalunya, però finalment, es va poder instal·lar directament des dels repositoris de la distribució.

Gràcies a la informació trobada en el GitHub de JarbasLingua¹⁷, s'ha pogut desenvolupar aquest punt del projecte d'una forma més ràpida i senzilla.

Per començar si no s'ha instal·lat prèviament Festival i es vol instal·lar amb l'extensió de català, serà necessària la següent comanda:

```
sudo apt-get -y install festival festvox-ca-ona-hts lame
```

Amb el motor instal·lat, només cal modificar la configuració de Mycroft per poder fer-ho funcionar correctament. També es canviarà la llengua de l'assistent en català per que així entengui que no estem en el seu mode normal en anglès.

```
Nano Mycroft-core/Mycroft/configuration/Mycroft.conf
```

Un cop obert el fitxer, es canviarà la llengua de "en-us" a "ca-es", en la secció de "lang", que es troba en les primeres línies del document.

```
// Language used for speech-to-text and text-to-speech.  
// Code is a BCP-47 identifier (https://tools.ietf.org/html/bcp47), lowercase  
// TODO: save unmodified, lowercase upon demand  
"lang": "ca-es",
```

Figura 30 Canvi d'idioma de Mycroft

Després s'ha d'afegir Festival com a motor de text-a-veu per que l'assistent el detecti, així que s'afegiran les següents línies de codi en el fitxer.

```
{  
  "tts": {
```


¹⁶FestCat: Descarregueu les veus. <http://festcat.talp.cat/download.php>. Accessed June 9, 2021.

¹⁷Mycroft-catalan.conf/readme-ca.md at main · JarbasLingua/Mycroft-catalan.conf · GitHub. <https://github.com/JarbasLingua/Mycroft-catalan.conf/blob/main/readme-ca.md#configuració-del-motor-tts>. Accessed June 9, 2021.

```

"module": "festival",
"festival": {
  "lang": "catalan",
  "encoding": "ISO-8859-15//TRANSLIT"
}
}
}

```



```

"tts": {
  // Engine. Options: "mimic", "mimic2", "google", "marytts", "fatts", "esp$
  // "spdsay", "responsive_voice", "yandex", "polly", "mozilla"
  "pulse_duck": false,
  "module": "festival",
  "polly": {
    "voice": "Matthew",
    "region": "us-east-1",
    "access_key_id": "",
    "secret_access_key": ""
  },
  "festival": {
    "lang": "catalan",
    "encoding": "ISO-8859-15//TRANSLIT"
  },
  "mimic": {

```

Figura 31 Configuració de Festival en Mycroft.conf

2.12.5 Mycroft en català

Per comprovar que funciona correctament, s'ha de posar en marxa Mycroft i donar-li alguna ordre, com per exemple preguntar-li quina hora és. Per cridar a l'assistent encara serà necessari dir-li "Hey Mycroft", però es pot modificar per que respongui a una crida en català.

```

Log Output: 313-322 of 322
===== mycroft-core 21.2.0 =====
~~~3.673 | INFO | 21691 | msm.skill_entry | Nothing new for fallback-query
~~~23.736 | INFO | 21691 | msm.skill_entry | Nothing new for mycroft-stock
~~~.719 | INFO | 21691 | msm.skill_entry | Nothing new for mycroft-pairing
~~~.539 | INFO | 21691 | msm.skill_entry | Nothing new for mycroft-naptime
~~~436 | INFO | 21691 | msm.skill_entry | Nothing new for fallback-unknown
~~~258 | INFO | 21691 | msm.skill_entry | Nothing new for mycroft-spelling
~~~ INFO | 21691 | msm.skill_entry | Nothing new for mycroft-configuration
~~~| INFO | 21691 | msm.skill_entry | Nothing new for mycroft-audio-record
^-- NEWEST ---^
History ===== Log Output Legend ===== Mic Level ==
quina hora a@s DEBUG output
>> les vint i quaranta-nou skills.log, other
voice.log
--- 122.66
86
Input (':' for command, Ctrl+C to quit) =====
>
  
```

Figura 32 Primera prova de Mycroft en català

Com es pot veure en la Figura 30, respon correctament a la petició de l'hora i retorna la resposta tant per pantalla com per altaveus. Se li poden formular més preguntes com per exemple quina és la capital d'Espanya.

```

quina a@s la capital d'espanya
>> La capital d'Espanya és Madrid,
Espanya
  
```

Figura 33 Segona prova amb Mycroft en català

Veient els dos últims resultats, es pot deduir que està utilitzant *skills* per poder retornar aquesta informació. Fins que l'assistent no es configuri en mode totalment *offline* no es podrà saber si les operacions de traducció de les *skills* les realitza en el núvol o en el propi sistema. Com que Festival es un motor que s'ha hagut d'instal·lar prèviament i no venia configurat amb Mycroft, també es dona per entès que funcionarà sense accés al servidor de Mycroft.

2.12.6 Backend amb Mycroft en català

Amb el veu-a-text i el text-a-veu ja configurats en català, només falta l'últim pas per poder fer aquest el funcionament assistent totalment de forma *offline*. El sistema de backend que s'utilitzarà serà el mateix que en el procés principal d'aquest projecte, el OVOS backend. Com s'ha pogut veure en l'apartat 2.11, el seu funcionament és millor comparat amb altres *backends* i ha funcionat correctament.

Els passos a seguir seran molt similars als del apartat anteriorment anomenat. Primer s'ha d'instal·lar el OVOS.


```
git clone https://github.com/OpenVoiceOS/OVOS-local-backend  
  
cd OVOS-personal-backend  
  
pip3 install .
```

Un cop descarregat, s'ha de realitzar una petita configuració.

```
cd ./config  
  
mkdir json_database  
  
nano ovos_backend.json
```

Al obrir-se aquest fitxer, s'han d'escriure les línies de codi que s'especifiquen en el punt 2.11.2 {

```
"stt": {  
  "module": "google"  
},  
"backend_port": 6712,  
"geolocate": false,  
"override_location": false,  
"api_version": "v1",  
"data_path": "~",  
"record_utterances": false,  
"record_wakewords": false,  
"wolfram_key": "BUNDLED_DEMO_KEY",  
"owm_key": "BUNDLED_DEMO_KEY",  
"default_location": {  
  "city": {  
    "code": "Lawrence",  
    "name": "Lawrence",  
    "state": {  
      "code": "KS",  
      "name": "Kansas",  
      "country": {  
        "code": "US",  
        "name": "United States"      }  
    }  
  }  
}
```



```

    }
  }
},
"coordinate": {
  "latitude": 38.971669,
  "longitude": -95.23525
},
"timezone": {
  "code": "America/Chicago",
  "name": "Central Standard Time",
  "dstOffset": 3600000,
  "offset": -21600000
}
}
}

```

Els següent pas és dirigir-se al document Mycroft.conf per modificar el servidor i que aquest interactui amb OVOS en comptes de amb Mycroft IA.

```
cd
```

```
nano Mycroft-core/Mycroft/configuration/Mycroft.conf
```

L'únic canvi necessari serà modificar la configuració del servidor per indicar-li que es dirigeixi a al port 6712 de la xarxa local on estarà escoltant Ovos.

```

"server": {
  "url": "http://127.0.0.1:6712",
  "version": "v1",
  "update": true,
  "metrics": true,
  "sync_skill_settings": true
},

```

Un punt molt important i que com en l'apartat 2.11 pot donar molts errors, és escriure `http` en comptes de `https` ja que aquest protocol no funciona correctament amb aquest backend. També serà necessari o per precaució, copiar els fitxers `cert.pem` i

key.pem de la altra targeta SD on tenim el Raspbian amb Mycroft en anglès, per si donés algun problema relacionat amb el SSL.

Després de completar tots aquest passos, només queda comprovar el funcionament de tots els punts anteriors conjuntament. Primer s'ha d'obrir el DeepSpeech en català i el Ovos per que es posin a escolat cadascun al seu port configurat de la xarxa local. LA comanda per engegar ovos és la següent:

```
ovos-local-backend
```

El següent pas es obrir Mycroft i realitzar-li una petició, com per exemple quina hora és.

```

Log Output: 154-163 of 163
===== mycroft-core 21.2.0 =====
---- | 5762 | __main__:handle_wakeword:71 | Wakeword Detected: hey mycroft
----2 | INFO | 5762 | __main__:handle_record_begin:41 | Begin Recording...
---- 5756 | mycroft.skills.skill_manager:send:66 | New Settings meta to upload.
----ngs_update:197 | All settings meta has been processed or upload has started
----manager:start_settings_update:199 | Skill settings downloading has started
----9.396 | INFO | 5762 | __main__:handle_record_end:49 | End Recording...
---- | 5762 | __main__:handle_utterance:76 | Utterance: ['quina hora ës']
----ries recordar-me {reminder}" for mycroft-reminder.mycroftai:Reminder.intent
^--- NEWEST ---^
History ===== Log Output Legend ===== Mic Level ==
quina hora ës          DEBUG output
>> les quatre i set    skills.log, other
                        voice.log
                        --- 108.36
                        76
Input (':' for command, Ctrl+C to quit) =====
>

```

Figura 34 Primera prova amb Mycroft en català offline

Com es pot veure en l'anterior figura, ha respost correctament a la petició. Com que Mycroft està configurat per donar l'hora de Kansas, aquest dona una hora diferent a la de la nostra localització, però el fet de realitzar la petició, després DeepSpeech ho entengui i envii la informació a Ovos per que aquest la sintetitzi i faci retornar una resposta tot de forma *offline*, ha sigut tot un èxit.

2.12.7 Skill Coses Vivents

Com a últim pas d'aquest projecte, es durà a terme una adaptació de la *skill* Living Things, desenvolupada en el marc d'un estudi general sobre assistents de veu¹⁸, al català. Aquesta *skill* bàsicament retorna una curiositat aleatòria d'animals o plantes.

¹⁸ Treball de fi de grau de Judith Font Muñoz <http://hdl.handle.net/2117/331119>

El funcionament de Coses Vivents serà idèntic al de Living Things, així que només farà falta incloure unes certes línies de codi per que pugui fer-ho en català. El primer pas ha sigut copiar la *skill* des de una *Pendrive* al directori de les *skills* en Mycroft. Es pot fer tant per interfície gràfica com per comanda.

```
mv living-things-skill /Mycroft-core/skills
```

Un cop Living Things estigui copiat al directori de les *skills*, es farà una còpia del directori la qual s'anomenarà *coses-vivents-skill*.

```
Cp -r living-things-skill cosas-vivents-skill
```

El següent pas és dirigir-se al fitxer `__init__.py` on es troba tot el funcionament de la *skill* i les diferents respostes a peticions que se li puguin fer.

```
cd cosas-vivents-skill
```

```
sudo nano __init__.py
```

Dins del fitxer es veu una funció `get_fact`, la qual és la encarregada de escollir quina resposta retornarà la *skill*. Per adaptar el codi al català, s'han de traduir totes les frases que es veuen en la funció, indicant amb una condició que si l'idioma és el català, respongui amb una frase d'aquest idioma i sinó en anglès. Aquestes frases no haurien d'estar en el `__init__.py` sinó a un programa internacionalitzat, no necessàriament per a assistents de veu.

```
from mycroft import MycroftSkill, intent_handler
from random import choice

def get_fact(language = 'en', category = None):
    if language == 'ca':
        animal = [
            "El cor d'una gamba es localitza en el seu cap.",
            'Un cargol pot dormir durant tres anys',
            'Els llímacs tenen quatre nassos.',
            'És possible hipnotitzar una granota rascant-li la panxa',
            'Una vaca produeix gairebé 200.000 gots de llet durant la seva vida' ]
        plant = [
            'Deixar la pell a les patates quan les cuines és més sa ja que conté totes les vitamines',
            'Banana vol dir dir dits en àrab',
            'Les pastanagues eren originalment de color lila',
            'El baobab que es troba a Àfrica pot enmagatzemar de 1.000 a 120.000 litres d'aigua a la seva tija',
            "El bambú la planta llenyosa que creix més ràpidament; pot créixer 90 centímetres en un sol dia"]

        if category == "animal":
            return choice(animal)
        elif category == "plant":
            return choice(plant)
        else:
            return choice(animal + plant)

    else:
        animal = [
            'The heart of a crayfish is located in its head',
            'A snail can sleep for three years',
            'Slugs have four noses.',
            'It is possible to hypnotize a frog by rubbing its belly',
            'A cow produces about 200,000 glasses of milk during its life']
        plant = [
            'Leaving the skin on potatoes when you cook them is safer because it contains all the vitamins',
            'Banana means fingers in Arabic',
            'Carrots were originally purple',
            'The baobab tree, found in Africa, can store 1,000 to 120,000 liters of water in its trunk',
            "Bamboo is a woody plant that grows the fastest; it can grow 90 centimeters in a single day"]

        if category == "animal":
            return choice(animal)
        elif category == "plant":
            return choice(plant)
        else:
            return choice(animal + plant)
```

Figura 35 Configuració fitxer `__init__.py` de la *skill* *coses-vivents*

Com es pot veure en la *Figura 33*, si l'idioma és el català, aquest retornarà una d'aquestes dades en aquesta llengua. També s'ha de modificar el nom de la classe a Coses Vivents, i lo mateix en el retorn de la funció `create_skill`.

```
class CosesVivents(MycroftSkill):
    def __init__(self):
        MycroftSkill.__init__(self)
```

Figura 36 Modificació de la classe LivingThings

Per últim s'han de crear els `fitxers.voc`. És tant senzill com traduir els que estan escrits en anglès; en el cas d'aquest *skill* és l'únic que s'ha de traduir. Aquest fitxers s'encarreguen de determinar el vocabulari de la *skill* per el moment en que l'usuari doni a entendre que vol utilitzar-la, aquests la cridin. Primer s'ha d'anar al directori `/vocab` on hi ha el vocabulari per a diferents idiomes. S'haurà de copiar la carpeta `en-us` a una que s'anomenarà `ca-es`.

```
cd vocab
cp en-us ca-es
cd ca-es
```

Un cop en el directori, és tan senzill com traduir totes les paraules i frases del anglès al català i la configuració de la *skill* estarà feta. Per comprovar que tot el procés ha sortit bé, se li farà una petició a Mycroft per que retorni una dada sobre plantes o animals. Per cridar la *skill* direm "Digues-me una dada".

```
History ===== Log Output Legend ===== Mic Level ==
digues me una dada          DEBUG output
>> Els llimacs tenen quatre nassos. skills.log, other
                               voice.log
                               --- 291.52
                               220
Input (':' for command, Ctrl+C to quit) =====
>
```

Figura 37 Comprovació del funcionament de Coses Vivents

Com es pot veure en la figura anterior, al preguntar-li per una dada, aquest ha retornat una de les opció aleatòries, així que el funcionament ha sigut correcte i Coses Vivents funciona perfectament amb Mycroft.

3. Resultats

3.1 Àmbit d'utilització

Per començar, cal remarcar que aquest és un prototip del que podria arribar a ser un assistent de veu amb el sistema de Mycroft totalment *offline*. Aquest projecte ha permès el desenvolupament de dos assistents de veu *offline*, un en anglès i l'altre en català. La Raspberry Pi ha donat la versatilitat per poder desenvolupar diferents parts del projecte a la vegada amb diferents targetes SD.

En principi es tracta d'un simple estudi però es podria utilitzar en tots els àmbits comercials on es requereixi un assistent de veu que no pugui comptar amb una connexió ininterrompuda a Internet per realitzar les seves funcions, Alguns exemples serien: automoció, navegació marítima, aparells que hagin de funcionar en localitzacions remotes, ... També pot ser útil en el cas d'aquells aparells que estiguin en llocs sensibles i en els quals calgui preservar la privacitat de les converses.

3.2 Validació dels dissenys

Per poder validar el funcionament de l'assistent *offline* s'han fet diferents comprovacions per veure com reacciona en certes situacions, com a l'hora en que se'l crida, funcionament dels *skills*, problemes al escoltar o el diferents temps de reacció al reconèixer y sintetitzar la veu. S'han fet les proves tant per l'assistent en català com en anglès.

Tasca	Mycroft en anglès	Mycroft en català
Resposta a la paraula clau	La reacció és momentània, ja sigui a la primera crida o qualsevol de les següents	La reacció és momentània, ja sigui a la primera crida o qualsevol de les següents
Comunicació amb l'usuari	Entén gairebé totes les peticions. Falla en algunes segurament per la pronunciació. Alguns noms propis no els entén, però aquets acostumen a ser en altres llengües.	Algunes peticions les entén i altres no. Falla bastant a l'hora d'entendre noms propis com el nom d'una ciutat o d'un personatge reconegut.

Skills	De forma <i>offline</i> pocs són els <i>skills</i> que funcionen. La de wikipedia segueix funcionant, igual que el de la hora i la localització. Aquests dos últims amb la configuració predeterminada que se li configura	En aquest cas només funcionen tres <i>skills</i> de forma <i>offline</i> , que són el Coses Vivents, localització i la hora. Aquestes són molt limitades puix que depenen del lloc i l'hora que estan configurats.
Reacció a una petició	La reacció és ràpida i sense problemes. El reconeixement del text és ràpid i amb pocs errors	La reacció és ràpida. Just després de que acabi d'enregistrar el missatge, només tarda un parell de segons en reconèixer el text
Temps de resposta	Tarda una mica més que amb les peticions, però només un parell de segons més.	Aquest temps és una mica més llarg que el de petició, però no són més de 3 o 4 segons, així que es pot dir que és ràpid.
DeepSpeech	Entén gairebé tot el que l'usuari li comunica. Com s'ha dit anteriorment, alguns noms propis els pot arribar a confondre.	Com s'ha dit en la comunicació amb l'usuari, entén tot menys els noms propis. Aquests els confon amb altres paraules que no tenen res a veure amb la paraula original.
<i>Text-to-Speech</i>	Retorna una veu masculina robòtica. La pronunciació és acceptable però no comparable amb les veus d'altres assistents com Siri o Alexa.	Retorna una veu femenina robòtica. No acaba de pronunciar bé algunes paraules però s'entén perfectament.

Taula 1 Comprovació del funcionament

Com es pot veure en la *Taula 1*, estem parlant d'un assistent força limitat a unes funcions molt concretes, però s'ha aconseguit que aquestes funcions es puguin dur a terme de forma *offline*, és a dir, no és necessària cap comunicació entre l'assistent i el núvol per dur a terme aquestes tasques. Gràcies al procés que comença per l'usuari indicant una instrucció per el micròfon, aquest envia la informació a l'assistent el qual la envia al programa que reconeix la veu, en aquest cas el Mozilla DeepSpeech. Quan el stt retorna una resposta a Mycroft, aquest intenta entendre quina és la instrucció de l'usuari. Quan ho ha entès, pensa en la resposta més adequada utilitzant un *skill*. Quan ja sap la resposta, la retorna en forma de veu a través del sintetitzador de veu, Mimic per el cas en anglès i Festival pel cas en català. El control de totes aquestes accions les ha dut a terme el *backend* Ovos-Local-Backend, el qual ha fet de substitut de les funcions dels servidors de Mycroft, almenys les necessàries per que l'assistent pensés que estava intercanviant informació amb el núvol.

Tot aquest procés ha permès una interacció assistent-usuari totalment *offline*, fent que pugui funcionar en situacions sense connexió a internet com podria ser un túnel o altres circumstàncies en que l'accés a la xarxa no fos possible. A més a més, permet la seguretat de la privacitat de l'usuari, puix que tota la informació que sol·licita a l'assistent no surt en cap moment de la xarxa local.

3.3 Descripció del funcionament

El funcionament en els dos casos, tant en anglès com en català, és idèntic. Per una connexió via ssh, és necessari el PC, micròfon i altaveus o auriculars. El primer pas és introduir la targeta SD a la Raspberry Pi, connectar-la a la font d'alimentació i engegar-la. Un cop encesa, des de l'ordinador s'haurà d'obrir l'aplicació VNC Viewer, la qual permetrà la visualització de la interfície gràfica del sistema operatiu Raspbian. Un cop oberta l'aplicació, s'haurà d'introduir la direcció IP desitjada per a connectar-se i esperar a que demani l'usuari i la contrasenya.

Amb els passos anteriors realitzats, apareixerà en la pantalla l'escriptori de Raspbian, en el qual s'haurà d'accedir a un terminal de comandes, i dins d'aquest obrir 3 pestanyes, una per Mycroft, la segona per al *speech-to-text* i la última per al *backend*. En la del stt, que en aquest projecte és el DeepSpeech, es necessari dirigir-se a la carpeta on es troben els fitxers `.tflite`, `.scorer` i el fitxer `config.json` amb la configuració del DeepSpeech per que pugui funcionar en mode servidor. Allà, la comanda per posar-lo a funcionar és: `deepspeech-server --config config.json`.

Amb DeepSpeech ja funcionant, el segon pas és engegar Ovos-Local-Backend, aquest s'obre dirigint-se a directori amb els eu nom i escrivint la comanda: `ovos-local-backend`.

Finalment, s'ha d'obrir Mycroft-core. És tant senzill com dirigir-se al directori de Mycroft-core i escriure la següent comanda: `./start-Mycroft.sh debug`.

Quan la interfície de Mycroft ja està en funcionament, només queda indicar a través del micròfon qualsevol petició, dintre de les possibilitat limitades que té amb aquesta configuració. Per avisar de que anirem a donar una instrucció, s'ha de dir la *wake up word* que son les paraules per a que escolti el que se li dirà, per defecte és "Hey, Mycroft". Després de la *wake up word* ja es pot indicar el que es vol dir, tant per el cas en català com en anglès.

3.4 Aplicacions del projecte

Aquest projecte pot servir com a base per a futurs desenvolupaments relacionats amb aquesta temàtica. Tant empreses com desenvolupadors independents podrien agafar idees o bàsicament seguir amb el projecte per millorar-lo. Hi ha varies temàtiques en les quals es podria aplicar, com altaveus intel·ligents, GPSs, electrodomèstics, vehicles o telèfons mòbils i ordinadors.

4. Comentaris finals

4.1 Pla de treball

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	
Estudi Previ i configuració Raspberry Pi	■	■																			
Instal·lació Raspbian i Picroft (amb respectives proves)		■	■																		
Instal·lació Picroft en Raspbian (amb respectives proves)			■	■	■																
Entrega Charter					■																
Instal·lació DeepSpeech com a servidor de Speech-to-text per a Mycroft.					■	■	■														
Instal·lació d'un sistema text-to-speech							■	■	■												
1r Seguiment									■	■	■	■									
Configuració de mycroft per l'ús exclusiu de serveis locals										■	■	■	■								
2n Seguiment												■									
Realització d'un prototip d'altaveu intel·ligent											■	■	■	■	■						
3r Seguiment															■						
programació skills online																■	■	■			
Millores																				■	■
Documentació																				■	■
Lliurament																					■

Taula 2 Diagrama de Gantt

Aquest diagrama s'ha dissenyat per distribuir la feina del projecte fos més equitativa possible i assegurar la seva realització a temps.

4.2 Llista de materials

La següent llista mostra els materials que s'han necessitat per realitzar el projecte:

- Raspberry Pi 4 Computer Model B amb 4GB RAM.
- Cable de corrent i carcassa amb ventilador per a la Raspberry Pi 4
- 3 targetes SD de 16, 32 i 64 GB de memòria.
- Micròfon de la marca Bopuroy amb cable USB.
- Auriculars o altaveus amb cable Jack 3.5mm.
- Ratolí amb connexió Bluetooth.
- Ordinador portàtil Lenovo ideapad 330

4.3 Pressupost

Per realitzar aquest projecte, s'han de contar dues parts molt important a nivell monetari. La primera és el cost del material, el qual engloba el cost de tots els materials utilitzats durant la realització d'aquest treball i es veu exposat en la següent taula

Material	Unitats	Preu €	Preu total
Raspberry Pi 4	1	57.00	57.00
Cable a corrent + carcassa	1	15.99	15.99
Tarjeta SD 16 GB	1	4.60	4.60
Tarjeta SD 32 GB	1	6.90	6.90
Tarjeta SD 64 GB	1	14.99	14.99
Micròfon	1	24.89	24.89
Auriculars	1	30.00	30.00
Ratolí	1	9.39	9.39
Ordinador portàtil	1	800.00	800.00
Total			963.76

Taula 3 Pressupost material

En segon lloc, i per poder calcular el pressupost total, s'ha de tenir en compte el sou del treballador per l'estudi, segons les hores realitzades. Aquestes han sigut aproximadament 300. Amb la següent taula es podrà veure aquest cost total:

Costos	€/hora	Hores	€
Material			963.76
Sou	8	300	2400
Total			3363.76

Taula 4 Pressupost total

Com es pot veure en la *Taula 4*, la inversió que s'hauria de fer per aquest projecte és de 3363,76 euros.

4.4 Impacte mediambiental i aspectes ètics

Ha sigut un treball que s'ha realitzat 100% des de casa. La feina s'ha pogut dur a terme sense cap dany al medi ambient ni cap altre activitat èticament qüestionable. Remarcant també que l'assistent de veu *offline* funciona al 100% sobre *software* lliure.

4.5 Fites aconseguides

Les fites que s'han aconseguit durant el projecte han sigut les següents:

- Instal·lació de totes les parts per que Mycroft funcioni *offline*
- Correcte funcionament de Mycroft en mode *offline*
- Versió de Mycroft *offline* en català i internacionalització de la *skill* Living Things

4.6 Conclusions

Amb aquest projecte s'han pogut treure varies conclusions respecte al funcionament d'un assistent de veu, en concret Mycroft. Aquest ha donat moltes facilitats gràcies al seu codi obert, puix que facilita la col·laboració en diversos aspectes. A part també comentar les facilitats que posen els desenvolupadors per incentivar el desenvolupament de millores o nous projectes a partir del seu producte.

Respecte als diferents passos necessaris per a que l'assistent funcioni de forma *offline*, es pot dir que en la gran majoria, la documentació ha sigut abundant o almenys amb les instruccions i explicacions necessàries per dur a terme qualsevol instal·lació o canvi en la configuració però no hi havia un tutorial que expliqués tots els passos, la qual cosa ha fet que integrat tots els elements hagi estat un repte. Cal dir que en situacions en que apareixien errors, no hi ha hagut massa documentació o referències al respecte, sinó que s'havia de buscar per internet errors semblants d'altres projectes i intentar adaptar-ho a aquest treball.

Tot l'ecosistema de Mycroft està basat, en gran part, en codi en Python o que es pot utilitzar des de Python. Aquest llenguatge també és el recomanat per la Raspberry Pi foundation i està molt ben suportat per Raspbian. Apart d'això, el fet de comptar amb un gestor de paquets propi ha facilitat molt la integració de tot el software necessari.

Una altra característica de l'ecosistema de Mycroft és que és *open source*. Això ha portat a fer un ús intensiu del gestor de codi git i dels repositoris online com GitHub, des d'on es pot descarregar el codi font de tots els programes necessaris.

En el tema més important del treball, és a dir, el funcionament de l'assistent en mode *offline*, ha sigut molt satisfactori. S'ha trobat la manera de poder construir tot un procés al voltant de Mycroft per que aquest funcioni de independent al núvol. Tot el conjunt de programes necessaris per aquest procés han fet possible que en cap moment, la informació sol·licitada per l'usuari sortís de la xarxa local.

En resum, l'assistent Mycroft permet una gran quantitat de possibilitats de desenvolupament al seu voltant, fent que es puguin crear nous *skills* ràpidament i adaptar-los al sistema d'una forma molt fàcil. Tampoc hi ha complicacions si no detecta els servidors de l'empresa, s'adapta fàcilment a qualsevol via que li permeti intercanviar dades amb l'usuari, com ha sigut en aquest projecte amb la combinació de l'assistent i un *backend* local.

4.7 Millores futures

Al finalitzar el treball s'han pogut pensar i deduir diferents camins futurs per seguir amb el seu desenvolupament. Una idea molt interessant podria ser seguir desenvolupant *skills* que s'adaptessin a l'assistent en català, quelcom semblant a Coses Vivents però amb un desenvolupament més complexa i més funcionalitats. També traduir les *skills* actuals que no estan encara traduïdes al català.

Per altra banda, també hi ha la opció d'adaptar aquest projecte a diferents tecnologies com podrien ser sensors. Un exemple molt interessant podria ser lligar Mycroft *offline* a un sensor de temperatura i crear un *skill* per que comuniqués la temperatura actual. Eventualment aquesta línia es podria expandir per crear un estació meteorològica o una alarma de la llar.

Finalment una extensió interessant seria convertir l'assistent *offline* en una interfície especialment adaptada a algun tipus de maquinària com, per exemple, un vehicle, amb *skills* especialment pensades per interactuar amb les funcionalitats específiques de la màquina.

5. Bibliografia

1. Speech-To-Text - Mycroft AI. <https://Mycroft-ai.gitbook.io/docs/using-Mycroft-ai/customizations/stt-engine>. Accessed March 16, 2021.
2. Releases · mozilla/DeepSpeech · GitHub. <https://github.com/mozilla/DeepSpeech/releases>. Accessed March 6, 2021.
3. GitHub - touchgadget/DeepSpeech: Install Mozilla DeepSpeech on a Raspberry Pi 4. <https://github.com/touchgadget/DeepSpeech>. Accessed March 4, 2021.
4. Descargue VNC Viewer | VNC Connect. <https://www.realvnc.com/es/connect/download/viewer/>. Accessed May 21, 2021.
5. Nmap: the Network Mapper - Free Security Scanner. <https://nmap.org/>. Accessed May 21, 2021.
6. Teach, Learn, and Make with Raspberry Pi. <https://www.Raspberrypi.org/>. Accessed May 21, 2021.
7. GitHub - ccoreilly/deepspeech-catala: Deepspeech ASR Model for the Catalan Language. <https://github.com/ccoreilly/deepspeech-catala>. Accessed May 11, 2021.
8. deepspeech-server · PyPI. <https://pypi.org/project/deepspeech-server/>. Accessed March 6, 2021.
9. Mycroft-catalan.conf/readme-ca.md at main · JarbasLingua/Mycroft-catalan.conf · GitHub. <https://github.com/JarbasLingua/Mycroft-catalan.conf/blob/main/readme-ca.md#configuració-del-motor-tts>. Accessed June 9, 2021.
10. Metadata in json format with speaker info - DeepSpeech - Mozilla Discourse. <https://discourse.mozilla.org/t/metadata-in-json-format-with-speaker-info/46337/13>. Accessed March 11, 2021.
11. SLL handshake error? · Issue #37 · MycroftAI/personal-backend · GitHub. <https://github.com/MycroftAI/personal-backend/issues/37>. Accessed April 16, 2021.
12. Get Started - Mycroft. <https://Mycroft.ai/get-started/>. Accessed May 23, 2021.
13. The History of Mycroft: Origin Story. <https://Mycroft.ai/blog/history-of-Mycroft-origin-story/>. Accessed May 6, 2021.

14. FestCat: Descarregueu les veus. <http://festcat.talp.cat/download.php>. Accessed June 9, 2021.
15. GitHub - ccoreilly/deepspeech-catala: Deepspeech ASR Model for the Catalan Language. <https://github.com/ccoreilly/deepspeech-catala>. Accessed June 7, 2021.
16. Mimic - Mycroft AI. <https://Mycroft-ai.gitbook.io/docs/Mycroft-technologies/mimic-overview>. Accessed March 19, 2021.
17. GitHub - MycroftAI/personal-backend: WORK IN PROGRESS: A Flask personal backend alternative for running your own version of <https://home.Mycroft.ai>. <https://github.com/MycroftAI/personal-backend>. Accessed March 25, 2021.
18. GitHub - MycroftAI/Mycroft-core: Mycroft Core, the Mycroft Artificial Intelligence platform. <https://github.com/MycroftAI/Mycroft-core>. Accessed March 4, 2021.
19. GitHub - MycroftAI/personal-backend: WORK IN PROGRESS: A Flask personal backend alternative for running your own version of <https://home.Mycroft.ai>. <https://github.com/MycroftAI/personal-backend>. Accessed June 13, 2021.
20. GitHub - touchgadget/DeepSpeech: Install Mozilla DeepSpeech on a Raspberry Pi 4. <https://github.com/touchgadget/DeepSpeech>. Accessed May 31, 2021.
21. GitHub - OpenVoiceOS/OVOS-local-backend: WIP - flask personal backend alternative to home.Mycroft. <https://github.com/OpenVoiceOS/OVOS-local-backend>. Accessed March 25, 2021.
22. GitHub - OpenVoiceOS/OVOS-local-backend: WIP - flask personal backend alternative to home.Mycroft. <https://github.com/OpenVoiceOS/OVOS-local-backend>. Accessed June 13, 2021.
23. GitHub - MainRo/deepspeech-server: A testing server for a speech to text service based on mozilla deepspeech. <https://github.com/MainRo/deepspeech-server>. Accessed June 2, 2021.
24. GitHub - stashify/deepspeech_cat: Deepspeech Catalan. https://github.com/stashify/deepspeech_cat. Accessed June 7, 2021.
25. Asistente de voz, una gran revolución tecnológica. https://www.elespanol.com/imprescindibles/20191231/asistente-voz-gran-revolucion-tecnologica/452705612_0.html. Accessed May 6, 2021.