



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

**Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa**

Estudi comparatiu d'assistents virtuals per a altaveus intel·ligents

Treball Fi de Grau

Grau en Enginyeria de Sistemes Audiovisuais

30 de Juny de 2020

Autor:

Judith Font Muñoz

Tutor:

Jordi Sellarès González

Convocatòria de lliurament del TFG:

Juny 2020

Índex

1. Introducció.....	4
1.1. Justificació del projecte	4
1.2. Antecedents	4
1.3. Objectius	6
1.4. Abast del projecte	6
1.5. Descripció general.....	7
2. Desenvolupament	8
2.1. Metodologia emprada	8
2.2. Recursos utilitzats.....	9
2.3. Instal·lació del sistema operatiu Raspbian a la Raspberry Pi	9
2.4. Instal·lació d'Alexa (Amazon).....	11
2.4.1 Alexa en la Raspberry Pi	11
2.5. Instal·lació de Google Assistant (Google)	16
2.5.1 Google Assistant en la Raspberry Pi	16
2.6. Mycroft en la Raspberry Pi.....	20
2.6.1 Creació de Mycroft en la Raspberry Pi	21
2.7 Programació de Skills	22
2.7.1 Alexa	23
2.7.2 Google Assistant	35
2.7.3 Mycroft	39
3. Resultats	45
3.1. Àmbit d'utilització	45
3.2. Comprovació dels assistents.....	45
3.3. Descripció del funcionament	49
3.4. Aplicacions de l'estudi.....	51
4. Comentaris finals.....	52
4.1. Pla de treball.....	52
4.2. Llista de materials	52
4.3. Pressupost.....	53
4.4. Impacte mediambiental, aspectes ètics,	53
4.5. Fites aconseguides	54
4.6. Conclusions	54
4.7. Millores futures.....	55
5. Bibliografia Addicional	56

Índex de Figures i Taules

Figura 1 Comparativa dels diferents dispositius de veu.....	5
Figura 2 Explicació de que esta connectat a la Raspberry Pi.....	7
Figura 3 Procediment d'un diàleg amb l'assistent de veu bàsic.....	8
Figura 4 Sistema operatiu descarregat per la Raspberry Pi.....	10
Figura 5 Programa balenaEtcher.....	10
Figura 6 Altaveu intel·ligent d'Alexa, primera generació.....	11
Figura 7 Dades del dispositiu creat per Alexa.....	12
Figura 8 Localització d'on es troba l'arxiu de credencials en Alexa.....	14
Figura 9 Dispositiu Google Home.....	16
Figura 10 Dispositiu Mycroft.....	20
Figura 11 Inici de Mycroft.....	21
Figura 12 Opcions en la compte de Mycroft.....	22
Figura 13 Diagrama de flux.....	23
Figura 14 Procés que realitza Alexa quan l'usuari fa la seva proposta.....	24
Figura 15 Esquema de com seria una conversa Usuari i Alexa per invocar una Skill.....	29
Figura 16 Fitxa d'inici per a la mostra de Tide Pooler.....	30
Figura 17 Finestra quan vols crear una nova skill a Alexa.....	31
Figura 18 Exemples predeterminats que es poden escollir per crear la skill a Alexa.....	31
Figura 19 Utterances de skill creada.....	32
Figura 20 Exemple de conversa amb la skill creada.....	34
Figura 21 Validació de la Skill de Amazon.....	35
Figura 22 Inicialització del un projecte.....	37
Figura 23 Opcions per escollir la tipologia de la skill.....	37
Figura 24 Inicialització del un projecte.....	37
Figura 25 Primer pas per poder crear l'action a google.....	38
Figura 26 Excel creat amb les preguntes i respostes del Trivial.....	38
Figura 27 Test de l'action de google.....	39
Figura 28 Directori test de la Skill de Mycroft.....	45
Figura 29 Inicialització de Alexa.....	49
Figura 30 Inicialització de google.....	50
Figura 31 Inicialització de Mycroft.....	51
Taula 1 Comportament dels dispositius a situacions quotidianes.....	48
Taula 2 Diagrama de Gantt.....	52
Taula 3 Pressupost del material utilitzat en el projecte.....	53
Taula 4 Pressupost total.....	53

1. Introducció

1.1. Justificació del projecte

Recentment els assistents de veu digitals han tingut un gran desenvolupament, ja que han crescut en nombre i en prestacions.

En aquests moments, internet té una gran importància, ja que una gran quantitat de dispositius ho utilitzen, com ara ordinadors, mòbils, tablets,... però la tendència actual és que molts més productes, els que es coneixen com *appliances*, també s'hi puguin connectar. Per exemple, tots els electrodomèstics poden esdevenir "intel·ligents", és a dir, que puguin adaptar totes les accions que realitzen i automatitzar-les. En aquests casos és especialment útil una interfície de veu, ja que són especialment útils quan els dispositius no siguin ni ordinadors ni dispositius tàctils.

Les grans empreses són les que desenvolupen l'anomenada *Internet of Things (IoT)*, però també hi hauria d'haver lloc per a petits desenvolupadors que poguessin crear els seus productes connectats a la *IoT*. En el desenvolupament d'aquests productes potser molt interessant incorporar un assistent de veu i, de fet, hi ha un cert nombre d'opcions, cadascuna amb els seus avantatges i els seus inconvenients.

És pertinent fer un estudi comparatiu dels diferents assistents de veu per poder saber quina seria la millor opció per a tots aquells petits desenvolupadors, empreses més petites o gent que ho fa per afició. Un estudi d'aquestes característiques seria una ajuda a l'hora de triar la millor opció per implementar una interfície de veu en un nou producte.

1.2. Antecedents

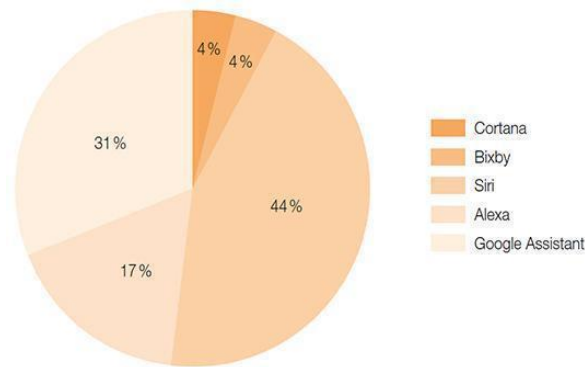
Es pot veure que durant els últims anys hi ha hagut una gran evolució en els assistents de veu, que cada vegada han agafat més força perquè són una gran ajuda pels usuaris.

Aquests assistents son una gran innovació i ofereixen un important servei als usuaris. A partir del 2016, més del 20% de les cerques s'han fet per veu. Els serveis que ofereixen aquests programes estan basats en les dades de l'usuari, la seva ubicació i l'accés a la seva informació. En el futur, s'utilitzaran d'una forma molt més àmplia per a tots els usuaris¹.

L'any 2018 el més utilitzat va ser Siri. En aquests moments existeixen més de 70.000 *skills* de Amazon i més d'1 milió de Google Assistant als Estats Units. Una *skill* és una funcionalitat extra que poden ser afegides a partir de tercers que es poden instal·lar als assistents de veu².

¹ «Asistente de voz, una gran revolución tecnológica». https://www.elespanol.com/imprescindibles/20191231/asistente-voz-gran-revolucion-tecnologica/452705612_0.html (accedit juny 15, 2020).

² «3. Asistentes virtuales de voz | Marketing XXI». <https://www.marketing-xxi.com/voice-search-asistentes-voz-altavoces-inteligentes-seo-sem/asistentes-virtuales-voz> (accedit juny 19, 2020).



Fuente: Elaboración propia basada en Kinsella y Mutchler (2018a).

Figura 1 Comparativa dels diferents dispositius de veu.

L'evolució d'aquests dispositius ha tingut una gran evolució, ja que el primer dispositiu reconeixia 16 paraules i els números del 0 al 9. Actualment, els assistents són a tot arreu perquè Apple té a Siri, Amazon a Alexa, Google a Google Assistant, Samsung a Bixby i Microsoft a Cortana. Totes les cerques es fan a partir d'intel·ligència artificial o *Machine learning*. D'aquesta manera es poden tenir uns resultats molt més específics respecte els que està demanant el consumidor³.

En l'actualitat hi ha 4 empreses molt importants que aposten per la intel·ligència artificial. El primer que va començar va ser Siri l'any 2012 i actualment té diferents accions que tots els seus usuaris utilitzen dia a dia. Mesos més tard va arribar Google on primerament va sortir Google Now i als 4 anys van presentar Google Assistant. Aquest assistent permet respostes visuals, recopilació d'informació a partir de cerques a través de la càmera, entre d'altres. En 2015, va sortir Cortana de Microsoft amb la finalitat per ajudar als seus usuaris a realitzar tasques bàsiques i per últim el 2015 va sortir Alexa de Amazon, primer a Estats Units i després al mercat europeu. Alexa és un dispositiu domèstic independent activat per veu⁴.

En l'actualitat, Amazon Alexa, Assistent de Google i Siri són els millors assistents virtuals del moment. Després d'una comparació entre ells de comptabilitat i de si els dispositius poden respondre a diferents respostes genèriques. En Alexa, un dels punts forts és el poder demanar una pregunta i de seguida et respon sense haver de reactivar-la. A vegades, té tendència de respondre preguntes errònies i no està preinstal·lat a telèfons mòbils. En canvi en Google, és un dispositiu amb més precisió, amb una gran capacitat d'ampliació, ja que es pot configurar amb diferents llengües, té la compatibilitat de Google, mentre que Alexa utilitza Bing per fer les cerques d'internet⁵.

Mycroft és un dispositiu en que l'usuari no s'ha de preocupar per la seva privacitat, ja que no guarda res en els seus historials sobre les tasques que se li han demanat. Té un motor Open

³«Los altavoces y asistentes de voz están revolucionando el modo de comprar». <https://tentulogo.com/los-altavoces-y-asistentes-de-voz-estan-revolucionando-el-modo-de-comprar/> (accedit juny 19, 2020).

⁴«La evolución de los Asistentes Virtuales en 8 años». <https://blogthinkbig.com/asistentes-virtuales-usuarios> (accedit juny 19, 2020).

⁵«The Best Voice Assistants of 2020 | Reviews.com». <https://www.reviews.com/home/smart-home/best-voice-assistant/> (accedit juny 19, 2020).

Source i serà capaç de fer tot el que li ordenin encara que sempre salvaguardant la nostra privacitat, altrament, el producte que s'ofereix val 190 dòlars⁶.

En conclusió, el que es pot comprovar és de que durant l'última dècada s'han anat desenvolupant molt els assistents de veu. Encara que, cada vegada n'hi ha més cadascun d'ells tenen la seva funcionalitat personalitzada, per tal que no tots serveixen per els mateixos serveis. No obstant, tots aquests dispositius han tingut diferents actualitzacions per tenir una funcionalitat més òptima i, així doncs, poder satisfer els seus usuaris. Els assistents cada vegada estan agafant més força i segur que esdevindran un dispositiu essencial pel consum de tothom.

1.3. Objectius

La finalitat d'aquest estudi és determinar quin és el millor dispositiu de veu per a que petits desenvolupadors, empreses més petites o gent que ho fa per afició l'incorporin als seus productes o projectes, basant-nos en diverses consideracions, com ara disponibilitat, adaptabilitat o respecte a la privacitat, entre d'altres.

Per aconseguir-ho ens hem marcat els següents objectius:

- Instal·lar els assistents en una placa de desenvolupament Raspberry Pi i descriure aquest procés.
- Inscriure'ns als respectius portals de desenvolupament i avaluar la seva funcionalitat.
- Configurar i programar el dispositiu, mitjançant la creació d'una *skill*(aplicació de veu) d'exemple, i documentar el procés.
- Explorar totes les possibilitats que tenen els dispositius, per exemple, l'automatització, accés a la media, és a dir, la ràdio, Spotify, Skills,...
- Comparar el suport d'idiomes que permet l'assistent de veu.
- Avaluar la privacitat que ofereixen.
- Determinar la capacitat de processar les dades *offline*.

1.4. Abast del projecte

Aquest estudi no comporta en el desenvolupament d'un producte com a tal, però sí que necessitem disposar d'un prototip per fer l'avaluació dels diferents assistents. Utilitzarem material, el més genèric possible per realitzar-lo consistent en un hardware senzill, assequible i àmpliament difós i el software més bàsic que es pugui instal·lar. En aquest sentit, l'estudi inclou la realització d'un prototip funcional d'altaveu intel·ligent al qual se li puguin instal·lar els diferents assistents que volem avaluar.

Apart de la creació d'un prototip aquest projecte inclou la programació d'una *skill* de prova per a cadascun dels assistents estudiats , a partir de la documentació i els exemples disponibles.

⁶«Mycroft, el asistente virtual con inteligencia artificial Open Source - Domótica en Casa». <https://domoticaencasa.es/mycroft-asistente-virtual-inteligencia-artificial-open-source/> (accedit juny 19, 2020).

Això ens permetrà avaluar la potència i la facilitat d'ús de l'entorn de desenvolupament associat a cada assistent (SDK).

Finalment, l'estudi inclou proves tant a l'assistent en general com al *skill* de prova per poder realitzar una comparativa entre els diferents assistents.

1.5. Descripció general

La single board computer (SBC) Raspberry Pi té les característiques idònies per ser la base del prototip ja que és un equip amb una gran difusió, rendiment acceptable, mides contingudes i que ha esdevingut un estàndard en el món del prototipatge. Per aquest motiu, els principals assistents digitals tenen versions per a aquest dispositiu, especialment quan es combina amb el sistema operatiu que recomana el fabricant, Raspbian. A més caldrà connectar uns altaveus i un micròfon. En el cas dels altaveus en tindrem prou amb uns altaveus genèrics ja que la RPi disposa d'una sortida de jack d'àudio però en el cas del micròfon caldrà disposar d'un micròfon USB. En definitiva es crearà un prototip de dispositiu intel·ligent de veu a partir d'aquests materials.

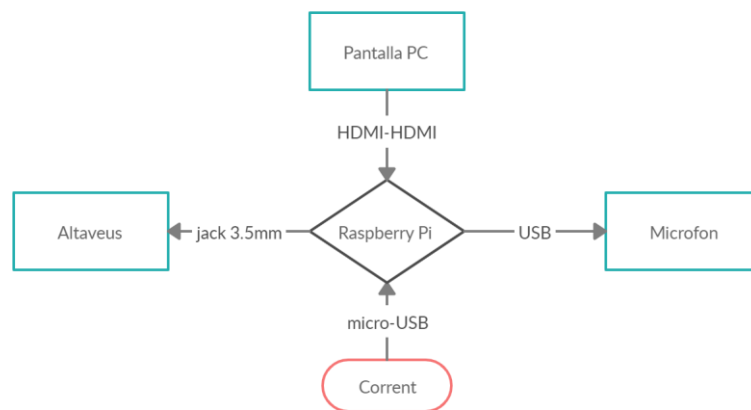


Figura 2 Explicació de que esta connectat a la Raspberry Pi.

Els assistents de veu són programes de software basats en intel·ligència artificial capaços de respondre diferents comandes de veu per l'execució de diferents tasques i preguntes. Aquests assistents permeten, la domòtica de l'habitatge, reproducció de música, consultar informació entre d'altres. A partir d'aquestes dades s'implementarà 3 assistents de veu a partir de les webs oficials d'aquests o amb webs de tercers que tinguin un seguiment bastant actual. Sempre s'haurà de realitzar una cerca prèvia per trobar la millor informació de implementació de software a la Raspberry Pi.

Una vegada posat en funcionament els 3 assistents de veu, s'hauria de fer una comprovació de diferents aspectes de la seva funcionalitat. Començant per la realització de diferents tasques quotidianes que un usuari pogués realitzar en el seu dia a dia.

Per fer-ho es programarà una *skill* per poder fer l'estudi de com es desenvolupa i dels passos que cal fer per a que es pugui utilitzar des del dispositiu. Aquesta *skill* haurà de detectar

diferents “utterances” (parts del diàleg que donen indicacions al dispositiu sobre el que volem que realitzi) i realitzar els “intents” (part condicional del codi) corresponents.

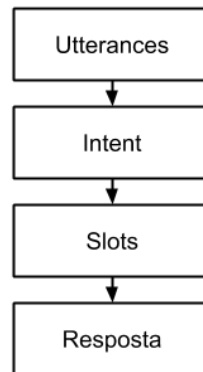


Figura 3 Procediment d'un diàleg amb l'assistent de veu bàsic.

També, s'establirà un conjunt de proves per poder veure quin dels assistents té les millors característiques per realitzar funcions quotidianes, com podria ser tant obrir un canal de ràdio, per a que s'escolti una emissora o fer una conversa amb el mateix dispositiu.

2. Desenvolupament

2.1. Metodologia emprada

Per fer la realització de l'estudi utilitzarem una Raspberry Pi, per tal que un dispositiu on es pot adaptar els diferents dispositius de veu. S'ha de crear tota la part de *Hardware* a partir de diferents materials. S'ha escollit la RPi perquè és ideal pel *Software*. Molts usuaris utilitzen Raspberry Pi per el desenvolupament de diferents tasques i les grans empreses donen suport per poder crear els assistents de veu a partir d'aquest dispositius, ja que poden donar ajudes als *developers*.

Instal·larem tres assistents de veu, l'Alexa de Amazon, Google Assistant de Google i per últim, Mycroft a partir de diferents fonts que s'han estat buscant i contrastant per arribar a que funcioni cada un dels assistents de veu. Com els assistents de veu sempre van evolucionant i van tenint actualitzacions pot passar que s'hagi de modificar algun pas per a que puguin funcionar. Habitualment, no hi ha un tutorial al 100% que serveixi. El que es farà és consultar els tutorials més importants i escollir un procediment que funcioni correctament i fer aquests passos a la Raspberry Pi i valorar l'eficàcia dels passos efectuats.. A vegades, al fer la cerca podem trobar-nos amb tutorials que estiguin obsolets, ja que no s'han actualitzat i el que es vol fer és la cerca de totes les comandes més actualitzades.

En aquesta cerca de diferents guies, que es poden trobar a internet a partir de fòrums o diferents webs per trobar-los.

Un cop trobat tot, el pas a pas de l'assistent, s'avalua com va el seu entorn, la programació, el funcionament,... La millor manera de realitzar-ho és fer el desenvolupament i com va arribar a tractar i buscar tots els obstacles que hi ha per solucionar-ho de la millor manera possible.

Per una banda, s'ha de tenir en compte la privacitat que dona cada un dels dispositius i com pot afectar al usuari. Per poder fer-ho, ens haurem de basar a partir de diferents fonts, com per exemple article de premsa especialitzada.

Per una altra banda, s'ha dissenyat una *skill* bàsica amb els condicionals bàsics que són necessaris per crear-les. Per crear-les es necessita tenir en compte les *utterances* que es s'han creat, per tal que a partir d'aquest crida s'extreuen el intent i veure si té algun *slot*, i així doncs, processar i doni una resposta.

Finalment, es fa una valoració de tots els obstacles que s'ha obtingut i com poden afectar a l'usuari durant el procés i es troben les solucions.

2.2. Recursos utilitzats

Durant tot el procés s'ha utilitzat diferents recursos, però el més important ha estat Internet perquè s'han realitzat totes les cerques d'informació, per demanar informació a diferents experts.

Un recurs també important es que a partir d'un ordinador portàtil, ASUS R510V, on s'ha realitzat tot el desenvolupament, a més a més, de tot els programes, terminals o diferents utilitzats com, *Putty*, per tenir la RPi connectada des de l'ordinador, els diferents entorns webs per crear les *skills* de cada dispositiu. S'ha fet ús del navegador com Opera o Chrome per fer els servidors webs de cada assistent, com Alexa i Google, ja que Mycroft es fa directament des del terminal.

Per la part de Hardware s'han utilitzat la Raspberry com material important, altaveus amb connexió jack 3.5 mm, un micròfon amb entrada USB, una pantalla d'ordinador, per poder connectar la RPi si no s'utilitza mitjançant l'ordinador.

2.3. Instal·lació del sistema operatiu Raspbian a la Raspberry Pi

La primera tasca que s'ha hagut de realitzar és la configuració de la Raspberry Pi per a que tingui una funcionalitat.

A partir de la web de Raspberry Pi oficial⁷, en l'apartat de Sistema Operatiu de Raspberry Pi s'ha descarregat el SO amb menys configuracions possibles, és a dir, el més bàsic, *Raspbian Buster Lite*.

⁷ «Raspberry Pi Downloads - Software for the Raspberry Pi». <https://www.raspberrypi.org/downloads/> (accedit abril 7, 2020).

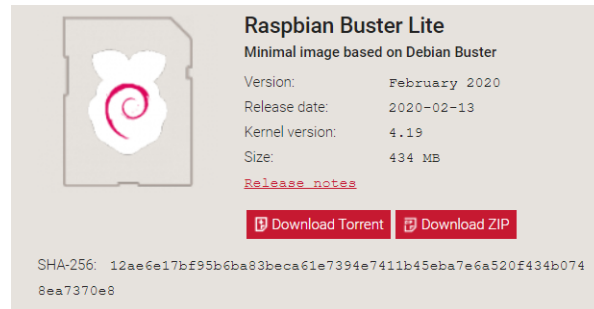


Figura 4 Sistema operatiu descarregat per la Raspberry Pi.

Després s'ha introduït a una targeta SD al programa *balenaEtcher* descarregat⁸ i després ja es té el dispositiu configurat. Aquest programa és un utilitari lliure i codi obert que s'utilitza per escriure arxius d'imatges per poder crear targetes de memòria *Live* i unitat flash USB.

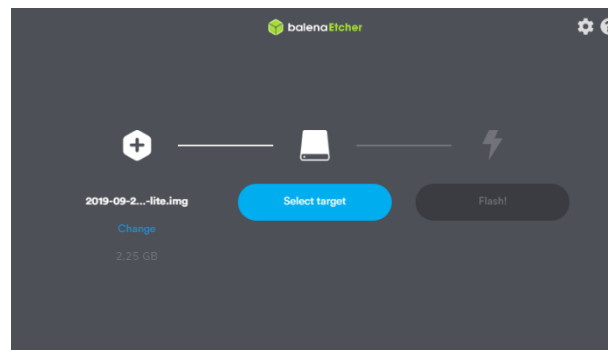


Figura 5 Programa balenaEtcher

Una vegada hem obtingut la imatge a la targeta SD. S'ha de configurar per tenir uns bàsics.

Entrem a la RaspberryPi:

Usuari: pi i contrasenya: raspberry, són valors que es donen per defecte

Després de entrar, el primer que s'ha d'anar a configuracions:

```
sudo raspi-config
```

Una vegada dintre de les configuracions s'ha de canviar:

- Teclat: 4 - Localisation Option → I3-Change Keyboard LayOut → Generic 105-key PC (intl.) → Other → Spanish → Spanish → Catalan (Spain, with middle-dot L) → The default for the keyboard layout → No compose key.
- Zona horària: 4-Localisation Option → I2-Change TimeZone → Europe → Madrid.
- Wifi-Country: - Localisation Option → I4 - Change Wi-fi Country → ES Spain.
- Contrasenya: En la primera opció que hi ha posa: Change User Password, l'usuari la pot canviar per la que vulgui.

Després de totes les configuracions s'ha de fer una actualització del sistema.

⁸ «balenaEtcher - Flash OS images to SD cards & USB drives». <https://www.balena.io/etcher/> (accedit abril 7, 2020)

```
Sudo apt-get update
```

```
Sudo apt-get upgrade
```

Depenen de si l'usuari vol un escriptori en la Raspberry o no, s'ha d'introduir una comanda o no, si es vol crear l'escriptori a partir de la comanda:

```
sudo apt-get install xserver-xorg raspberrypi-ui-mods
```

Després s'ha de tornar a sudo raspi-config i anar a la tercera opció, Boot Options → Dekstop / CLI → Dekstop Autologin. D'aquesta manera l'usuari no hauria d'indicar la contrasenya cada vegada que encén la Raspberry Pi.

2.4. Instal·lació d'Alexa (Amazon)

Alexa és un assistent virtual que està controlat per veu. Aquest assistent de veu ha estat creat per Amazon. Va ser creat al novembre de 2014 i es trobava en els dispositius d'Altaveus Intel·ligents. Les funcions d'Alexa depenen de dos elements claus. Per una banda, es troben totes les comandes de veu que ja hi són integrades, com pot ser demanar la hora demanar el temps entre d'altres. Per altra banda, es troben les *skills* que són complements que s'instal·len i es poden afegir moltes més funcionalitats per ampliar les possibilitats de l'assistent..

Per invocar-lo es diu el seu nom, "Alexa", i després demanen el que l'usuari necessita en aquell moment i et donarà una resposta.



Figura 6 Altaveu intel·ligent d'Alexa, primera generació.

2.4.1 Alexa en la Raspberry Pi

La instal·lació d'Alexa s'ha realitzat a partir de la web oficial de Amazon. La web de *developers*⁹, és per la creació de *skill* per l'assistent, la desenvolupament d'aplicacions i jocs, inclou funcions i serveis que permetin repetir comandes en els dispositius de l'usuari i en el centre de desenvolupadors es poden trobar eines, documentació i el codi de mostra per poder crear aplicacions.

⁹«Iniciar sesión[1] «Alexa Voice Service». <https://developer.amazon.com/alexa/console/avs/home?> (accedit abril 7, 2020).

Primer de tot, s'ha de registrar un nou producte¹⁰. Per poder arribar a crear-lo s'ha d'anar a l'apartat d'*Alexa Voice Service Developer Console*. En el moment que estem en aquesta pàgina, s'ha d'accedir als productes i crear un de nou. S'han d'anar indicant el nom del producte, la ID i per a quina finalitat es crea el producte. En aquest cas, es el dispositiu on conté l'Alexa incorporada. Les característiques més rellevants, com per exemple, un dispositiu de mans lliures. S'ha d'indicar que no serà un producte comercial ni per una empresa ni que hi ha una compte de *AWS IoT Core Accounts*. Després s'ha de crear un *Security profile* on es posa un nom, la descripció i una vegada generat s'ha d'activar en l'enllaç¹¹.

D'aquesta manera es genera un Client ID, un Client secret i un Amazon ID que després serà imprescindible en el moment de la configuració a la Raspberry Pi. També t'has de descarregar el fitxer *.json* del producte.

AlexaPI

Client ID amzn1.application-oa2-client.9929d42197a9488180d41b25332249e3	Amazon ID AFL4P81T2WACE
Client secret add275a24cc2de5ebd7a02d42db69eada8e811c9908f7624e72d01833e69c8fa	

Figura 7 Dades del dispositiu creat per Alexa

Un cop creat, ja ens passem amb la configuració de la Raspberry Pi, on la hem configurat prèviament amb el sistema operatiu de la Raspberry pi, Raspbian. Comença la configuració d'Alexa en el dispositiu¹².

Per començar, hem de configurar l'entorn de configuració.

```
cd /home/pi/  
mkdir sdk-folder  
cd sdk-folder  
mkdir sdk-build sdk-source third-party sdk-install db
```

Fer una petita actualització del paquets de la Raspberry Pi

```
cd /home/pi/  
sudo apt-get update  
sudo apt-get upgrade
```

S'instal·len totes les dependències de la Raspberry pi.

¹⁰«Register a Product | Alexa Voice Service». <https://developer.amazon.com/es-ES/docs/alexa/alexa-voice-service/register-a-product.html> (accedit abril 12, 2020).

¹¹«Register a Product | Alexa Voice Service». <https://developer.amazon.com/es-ES/docs/alexa/alexa-voice-service/register-a-product.html#create-a-security-profile> (accedit maig 28, 2020).

¹²«Set Up the AVS Device SDK on Raspberry Pi | AVS Device SDK». <https://developer.amazon.com/es-ES/docs/alexa/avs-device-sdk/raspberry-pi.html> (accedit maig 28, 2020).

```
sudo apt-get -y install \ git gcc cmake build-essential libsqlite3-dev  
libcurl4-openssl-dev libfaad-dev \ libssl-dev libsoup2.4-dev libgcrypt2-dev  
libgstreamer-plugins-bad1.0-dev \ gstreamer1.0-plugins-good libasound2-dev  
doxygen
```

Una vegada fet tot això, es descàrrega l'entorn de AVS Device SDK, ja que s'ha de clonar un escriptori de GitHub. Però per poder fer-ho s'ha d'instal·lar Python al dispositiu.

```
sudo apt-get install python-pip  
sudo apt-get install python3-pip  
pip install commentjson  
cd /home/pi/sdk-folder/sdk-source  
git clone --single-branch git://github.com/alexa/avs-device-sdk.git  
cd /home/pi/sdk-folder/third-party  
git clone git://github.com/Sensory/alexa-rpi.git
```

Després de carregar totes les clonacions s'ha de córrer la llicència per a que pugui funcionar.

```
cd /home/pi/sdk-folder/third-party/alexa-rpi/bin/  
./license.sh
```

Un cop realitzat s'ha de crear el SDK utilitzant el terminal de la Raspberry Pi, per realitzar-lo s'ha de fer la generació de dependències de compilació utilitzant la comanda *cmake*.

```
cd /home/pi/sdk-folder/sdk-build  
cmake /home/pi/sdk-folder/sdk-source/avs-device-sdk \  
-DSENSORY_KEY_WORD_DETECTOR=ON \  
-DSENSORY_KEY_WORD_DETECTOR_LIB_PATH=/home/pi/sdk-folder/third-party/alexa-  
rpi/lib/libnsr.a \  
-DSENSORY_KEY_WORD_DETECTOR_INCLUDE_DIR=/home/pi/sdk-folder/third-  
party/alexa-rpi/include \  
-DGSTREAMER_MEDIA_PLAYER=ON \  
-DPORTAUDIO=ON \  
-DPORTAUDIO_LIB_PATH=/home/pi/sdk-folder/third-  
party/portaudio/lib/.libs/libportaudio.a \  
-DPORTAUDIO_INCLUDE_DIR=/home/pi/sdk-folder/third-party/portaudio/include  
make SampleApp
```

Al realitzar les comandes de *cmake* s'ha de executar l'*script* de configuració. Primer de tot s'ha d'obrir el fitxer que es troba a:

```
home/pi/sdk-folder/sdk-source/avs-device-sdk/tools/Install
```

En aquest fitxer s'ha d'emplenar amb el .json que s'ha hagut de descarregar anteriorment. En la guia de *developer*, informen com fer-ho si no s'ha pogut entendre correctament¹³. S'ha de descarregar un fitxer que es troba en la web de Alexa després de fer el registre del nou producte.

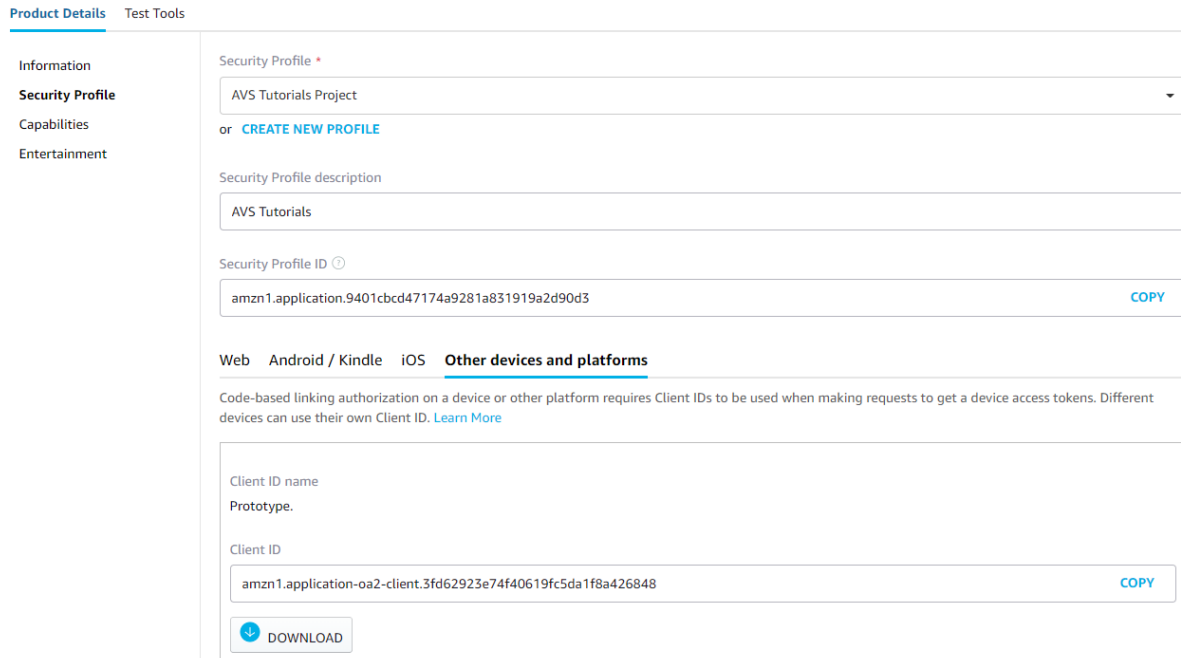


Figura 8 Localització d'on es troba l'arxiu de credencials en Alexa.

Un pas important és agafar el fitxer .json i s'introdueix al fitxer que s'acaba de crear. Executem `genCongif.sh` a partir del directori:

```
$ HOME / my_project / source / avs-device-sdk / tools / Install
```

En aquest últim fitxer es poden trobar diferents conceptes del producte configurat. Llavors, per introduir totes les dades que es necessiten s'han d'utilitzar aquesta comanda i després s'ha de modificar un segon fitxer.

```
cd /home/pi/sdk-folder/sdk-source/avs-device-sdk/tools/Install bash
genConfig.sh config.json 12345 \
/home/pi/sdk-folder/db \
/home/pi/sdk-folder/sdk-source/avs-device-sdk \
/home/pi/sdk-folder/sdk-build/Integration/AlexaClientSDKConfig.json \ -
DSDK_CONFIG_MANUFACTURER_NAME="raspberrypi" \ -
DSDK_CONFIG_DEVICE_DESCRIPTION="raspberrypi"
```

```
nano /home/pi/sdk-folder/sdk-build/Integration/AlexaClientSDKConfig.json
```

S'ha d'introduir `"cb1AuthDelegate":{` al davant del fitxer.

La configuració del micròfon on s'ha de configurar un fitxer per indicar la entrada de la targeta del micròfon i dels altaveus.

¹³«Register a Product | Alexa Voice Service». <https://developer.amazon.com/es-ES/docs/alexa/alexa-voice-service/register-a-product.html#create-a-security-profile> (accedit maig 28, 2020).

```
nano ~/.asoundrc
pcm.!default {
    type asym
    playback.pcm {
        type plug
        slave.pcm "hw:0,0"
    }
    capture.pcm {
        type plug
        slave.pcm "hw:1,0"
    }
}
```

Una vegada realitzada totes les configuracions comença l'execució i l'autorització de l'aplicació d'exemple. Primer de tot ens hem de desplaçar a la carpeta *sdk-build*, on es realitzarà la inicialització de l'entorn.

```
cd /home/pi/sdk-folder/sdk-build PA_ALSA_PLUGHW=1 ./SampleApp/src/SampleApp
./Integration/AlexaClientSDKConfig.json DEBUG9
```

Amb aquesta comanda, sortirà una URL que al entrar, s'ha d'iniciar la sessió de *developers* de Amazon, on s'ha fet el registre del producte. En el mateix enllaç es donarà un codi específicat, per posar en el terminal, ja que te'l demanarà i s'autoritzarà la mostra.

Per tornar a executar l'aplicatiu de mostra, s'ha de executar:

```
cd /home/pi/sdk-folder/sdk-build PA_ALSA_PLUGHW=1 ./SampleApp/src/SampleApp
./Integration/AlexaClientSDKConfig.json ../third-party/alexa-rpi/models
```

En el cas d'Alexa no s'ha hagut de fer una gran recerca, ja que la seva web oficial suporta que els petits *developers* puguin realitzar Alexa en un dispositiu que no sigui comprant els seus productes. Així doncs, et donen la facilitat de fer Alexa en una Raspberry Pi o en diferents dispositius¹⁴.

¹⁴«Set Up the AVS Device SDK on Raspberry Pi | AVS Device SDK». <https://developer.amazon.com/es-ES/docs/alexa/avs-device-sdk/raspberry-pi.html> (accedit abril 12, 2020).

2.5. Instal·lació de Google Assistant (Google)

L'assistent de veu de Google es diu Google Assistant. Aquest assistent ens ajuda a fer una cerca a Internet, fer programació de diferents esdeveniments i alarmes. També es pot escoltar música, demanar per veure vídeos o fer de traductor. Per invocar-la s'ha de dir "Ok Google" més la petició corresponen.



Figura 9 Dispositiu Google Home.

El dispositiu el que fa son respostes a les preguntes que fa l'usuari.

2.5.1 Google Assistant en la Raspberry Pi

En el cas de Google, no tenen suport oficial des de fa uns mesos a que es pugui crear cap assistent de veu amb una Raspberry o un altre dispositiu. Fa temps, Google va intentar donar el suport que els *developers* poguessin tenir les seves Raspberry pi, però per assumptes comercials ja no donen aquest suport.

Hi ha gent que n'ha fet versions on s'ha explicat els passos de com poder realitzar-ho però sense actualitzacions, per tant totes les pàgines de tercers pot ser que estiguin desactualitzades o tingui algun inconvenient per seguir-les.

Al realitzar una cerca per trobar una web de tercers on expliquen les diferents opcions per poder realitzar-ho a partir de diferents pàgines web de diferents usuaris. Un dels inconvenients que hi ha es la nul·la actualització que hi ha, ja que pot ser que Google canviï diferents paràmetres i ja no es pugui obtenir l'assistent. En el meu cas he utilitzat dues pàgines de externs¹⁵.

Per fer tota la configuració hem de crear un projecte amb un compte de Google i habilitar la API de Google Assistant.

Després d'aquest dos passos s'ha de registrar el model que hem de crear i l'habilitació de l'API, es troba a l'apartat *API & Serveis*. Abans de poder fer qualsevol cosa hem d'acceptar tots els termes per poder crear el dispositiu¹⁶.

Per últim, s'ha de descarregar les credencials de *.json* i deixar-les guardades per després. A més a més de la configuració d'àudio en la Raspberry Pi.

¹⁵«Your own Raspberry Pi Google Assistant by Janne Spijkervet -». <https://hakin9.org/your-own-raspberry-pi-google-assistant/> (accedit juny 19, 2020).

¹⁶


```
arecord -l
```

```
aplay -l
```

En aquestes dues comandes hem de buscar la *card* i el *device* del micròfon i els altaveus per després poder agregar aquests valors al fitxer que es crearà.

```
sudo nano /home/pi/.asoundrc
```

```
pcm.!default {
    type asym
    capture.pcm "mic"
    playback.pcm "speaker"
}
pcm.mic {
    type plug
    slave {
        pcm "hw:<card number>,<device number>"
    }
}
pcm.speaker {
    type plug
    slave {
        pcm "hw:<card number>,<device number>"
    }
}
```

En el meu cas el micròfon era *card:1* i *device:0* i amb els altaveus *card:0* i *device:1*. Després es realitza una comprovació per veure si funciona correctament.

```
speaker-test -t wav
```

```
arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw out.raw
```

```
aplay --format=S16_LE --rate=16000 out.raw
```

El pas número 3, es per crear un nou directori de google assistant i afegir dintre les credencials que hem descarregat en el pas anterior, per tal que es creï un fitxer per fer-ho.

```
mkdir ~/googleassistant
```

```
nano ~/googleassistant/credentials.json
```

Una vegada creat el directori, s'ha d'instal·lar tots els paquets de Python3 i de Google amb les seves llibreries necessàries.

```
sudo apt-get install python3-dev python3-venv
```

```
python3 -m venv env && env/bin/python -m pip install --upgrade pip setuptools  
--upgrade && source env/bin/activate
```

```
python3 -m pip install --upgrade wheel
```

```
python -m pip install --upgrade google-assistant-library google-assistant-  
sdk[samples]
```

Quan ja ho tenim realitzat hem de fer que ens autoritzin el Google Assistant. Primer s'ha d'instal·lar l'autorització amb l'API de Google Assistant i després afegim una comanda on et donarà una URL i d'allà has de donar a permetre a tots els termes i permisos per a que funcioni.

```
python -m pip install --upgrade google-auth-oauthlib[tool]
```

```
google-oauthlib-tool
```

```
--client-secrets ~/googleassistant/credentials.json \
```

```
--scope https://www.googleapis.com/auth/assistant-sdk-prototype \
```

```
--scope https://www.googleapis.com/auth/gcm \
```

```
--save --headless
```

Si per algun cas quan s'obre la URL, dóna error, vol dir que no s'ha habilitat alguna API o que no hem donat permís per a que funcionin totes les APIS. Per solucionar-ho només s'ha d'anar a la part de credencials i fer una comprovació de totes elles.

Hem d'instal·lar diferents paquets de so perquè pugui funcionar tot correctament.

```
sudo apt-get install libportaudio2
```

```
sudo apt-get install pulseaudio
```

Una vegada s'ha obtingut que dóna l'enllaç, s'ha de crear un nou fitxer i introduir-li unes especificacions, per a que es pugui escoltar.

```
sudo nano /etc/systemd/system/pulseaudio.service
```

```
[Unit]
```

```
Description=PulseAudio Sound Server in system-wide mode
```

```
[Service]
```

```
Type=forking
```

```
PIDFile=/var/run/pulse/pid
```

```
ExecStart=/usr/bin/pulseaudio --system --disallow-exit=1 \
```

```
--disable-shm=1 --fail=1 --daemonize
```

```
[Install]
```

```
WantedBy=multi-user.target
```

En un altre fitxer que ja existeix s'ha de comentar una línia amb un #.

```
sudo nano/etc/pulse/default.pa
```

```
#load-module module-suspend-on-idle
```

Un cop fet, es comença habilitar el servei de *pulseaudio* i es crea un usuari pi.

```
sudo systemctl --system enable pulseaudio.service
```

```
sudo adduser pi pulse-access
```

L'últim pas, s'ha de crear un arxiu `sudo nano start_assistant.sh` utilitzant les ID que s'han obtingut al principi al crear el projecte a la web de la consola de Google. On s'ha de completar amb ID del projecte i el model del dispositiu.

```
#!/bin/bash
```

```
source /home/pi/env/bin/activate
```

```
googlesamples-assistant-pushtotalk --project-id 'my-project-id' --device-  
model-id 'my-device-model-id'
```

I després de crear-lo es fa l'executable, `sudo chmod +x start_assistant.sh`

Es realitza un nou fitxer:

```
sudo nano /etc/systemd/system/assistant.service
```

On s'ha d'omplir amb:

```
[Unit]
```

```
Description=Google Assistant
```

```
Wants=network-online.target
```

```
After=network-online.target
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/bin/bash /home/pi/start_assistant.sh
```

```
Restart=on-abort
```

```
User=pi
```

```
Group=pi
```

[Install]

```
WantedBy=multi-user.target
```

I per últim habilitem i comencem el servei de Google Assistant.

```
sudo systemctl enable assistant.service
```

```
sudo systemctl start assistant.service
```

Sempre es pot tenir un control de l'estat del dispositiu.

```
sudo systemctl start assistant.service
```

D'aquesta manera es pot saber si està en funcionament, en repòs o si té problemes.

2.6. Mycroft en la Raspberry Pi

Mycroft és un assistent de veu virtual de codi obert que ens permet tenir privacitat. Volen que els dispositius d'aquesta marca tinguin una experiència amb el dispositiu de veu sense cap problema amb la seva privacitat.



Figura 10 Dispositiu Mycroft.

Les empreses de Amazon i Google tenen un nivell de negoci molt contrari al de Mycroft, perquè aquestes dues, sempre tenen un objectiu com en el cas d'Amazon es poder arribar a vendre diferents productes i la utilització de les seves eines com Amazon Music o Amazon entre d'altres i a Google es seu propòsit es la publicitat, ja que quan un usuari fa una cerca a Google sempre es trobarà amb anuncis. Així doncs, agafaran informació privada dels usuaris per poder facilitar els gustos i mostrar-ho a partir de vendes de productes o anuncis.

En canvi, Mycroft és tot el contrari. Aquesta petita empresa el que vol vendre es el *Hardware* del assistent de veu. Mycroft només ven la part de *Hardware* per això tenen un dispositiu genèric, utilitza el seu *Software* propi. Encara que donen la facilitat d'obtenir un *Software* de manera gratuïta, Picroft, és a dir, et donen la part de *Software* per has de facilitar tu el *Hardware* per utilitzar-ho.

2.6.1 Creació de Mycroft en la Raspberry Pi

Per poder configurar Mycroft en la Raspberry Pi, la compta oficial ens dóna el pas a pas de com realitzar-la.¹⁷

Ens dóna uns passos a seguir on primer de tot ens hem de crear un compte de Mycroft. D'aquesta manera pots tenir en el teu perfil tots els dispositius que tens a càrrec¹⁸.

En segon lloc, s'ha de fer la instal·lació i l'aparellament de Picroft que és una plataforma de desenvolupament de Mycroft gratuïta que ofereix als desenvolupadors. Assumeixen que la Raspberry Pi serà el dispositiu de desenvolupament i així doncs, distribueixen una imatge amb Picroft inclòs.

Una vegada estigui la imatge operativa a l'encendre la Raspberry Pi amb la imatge descarregada ha de sortir així:

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 12 07:23:47 2018

MYCROFT
Picroft

*****
** Picroft enclosure platform version: Lighting preview
** mycroft-core: 18.8.0 ( dev )
*****

Welcome to Picroft. This image is designed to make getting started with
Mycroft quick and easy. Would you like help setting up your system?
  Y)es, I'd like the guided setup.
  N)ope, just get me a command line and get out of my way!
Choice [Y/N]:
```

Figura 11 Inici de Mycroft.

Aquesta imatge aporta una petita introducció on et posa les petites configuracions que s'han de fer.

Per començar, s'ha de configurar l'àudio i el micròfon, perquè soni per pantalla. S'ha d'indicar on es troben els altaveus i com es localitzen i al mateix amb el micròfon. Et donen diferents opcions i has d'escollir la que pertoqui per a cada element.

Una vegada realitzat t'enviarà directament al Mycroft o sinó s'ha d'indicar amb la comanda per poder parlar amb el dispositiu.

`mycroft-cli-client`

Una vegada allà, sortirà una pantalla amb un codi. Aquest codi es per registrar el producte a la web i tot pugui estar operatiu.

¹⁷ «Get Started - Mycroft». <https://mycroft.ai/get-started#setup>, (accedit maig 30, 2020)

¹⁸ «Mycroft AI Single Sign On».

<https://sso.mycroft.ai/login?redirect=https:%2F%2Fhome.mycroft.ai%2F> (accedit maig 31, 2020).

Com abans, ja l'usuari s'ha creat el compte ha d'anar al icona de perfil, que es troba a dalt a la dreta, i anar a *Add Device*. Allà es crea el dispositiu al gust de l'usuari i s'introdueix el codi per a que funcioni des de la Raspberry Pi.

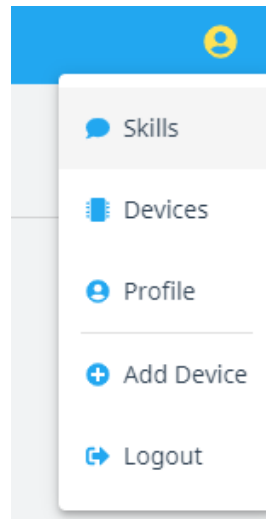


Figura 12 Opcions en la compte de Mycroft.

També s'ha de tenir en compte que, ha d'estar connectat a una xarxa wifi.

Una vegada creada la màquina, començarà a funcionar correctament. A més a més, Mycroft té diferents comandes especials per diferents *Add Device* funcions del dispositiu.

- `mycroft-cli-client`: Aquesta comanda inicialitza el dispositiu, per a que l'usuari pugui demanar el que necessiti en aquell moment.
- `mycroft-help`: Comanda que dona informació.
- `mycroft-mic-test`: Comanda que torna a executar la prova de micròfon.
- `mycroft-msk`: Executa el kit d'habilitats de Mycroft.
- `mycroft-msm`: Executa el gestor d'habilitats de Mycroft.
- `mycroft-pip`: Execució de pip dins de l'entorn de Mycroft Python (venv).
- `mycroft-say-to`: S'utilitza sobretot, quan no funciona el micròfon. Quan poses aquesta comanda s'envia a Picroft.
- `mycroft-setup-wizard`: Aquesta comanda torna a executar una configuració guiada.
- `mycroft-skill-testrunner`: Comanda que executa el *Testrunner*.
- `mycroft-speak`: Aquesta comanda fa que Mycroft parli amb *Text to Speech*.
- `mycroft-venv-deactivate`: Comanda que desactiva l'entorn virtual de Mycroft.
- `mycroft-wipe`: Amb aquesta comanda es restaura Picroft a l'estat de fàbrica. S'eliminarà el dispositiu i totes les configuracions que s'haurien fet.

2.7 Programació de Skills

Una *skill* o una *action*, és on l'usuari convoca i el dispositiu respon a base de unes respostes que s'han estat configurant anteriorment. Poden ser jocs, entreteniment, notícies, informació,...

En una *skill* o *action* es poden fer:

- Capacitat de prendre decisions.
- Facilitat per treballar en equip.
- Persistència i constància per obtenir els resultats desitjats.
- Bona capacitat de comunicació.
- Ser un gran apassionat pel treball que s'ocupa.

Depenen del dispositiu té una finalitat de com fer-les i quin procés es necessita per poder crear-les i utilitzar-les.

La idea principal de les skills que es volen crear són sobre curiositats sobre plantes i animals. És una habilitat bastant senzilla on només es crida a la *skill* a partir de les *utterances* que s'hauràn de crear en la *skill*. Per poder diferenciar quan l'usuari demana un tipus o un altre s'han de crear un intent, *living_beigns*, on s'hauran de crear diferents *slots* dintre d'aquests. Aquests *slots* són planta i animal.

La creació de intent es poder fer la diferencia entre que necessita l'usuari i depenen com faci la crida tindrà una resposta aleatòria a partir del *slot* que s'ha indicat a la *utterance*.

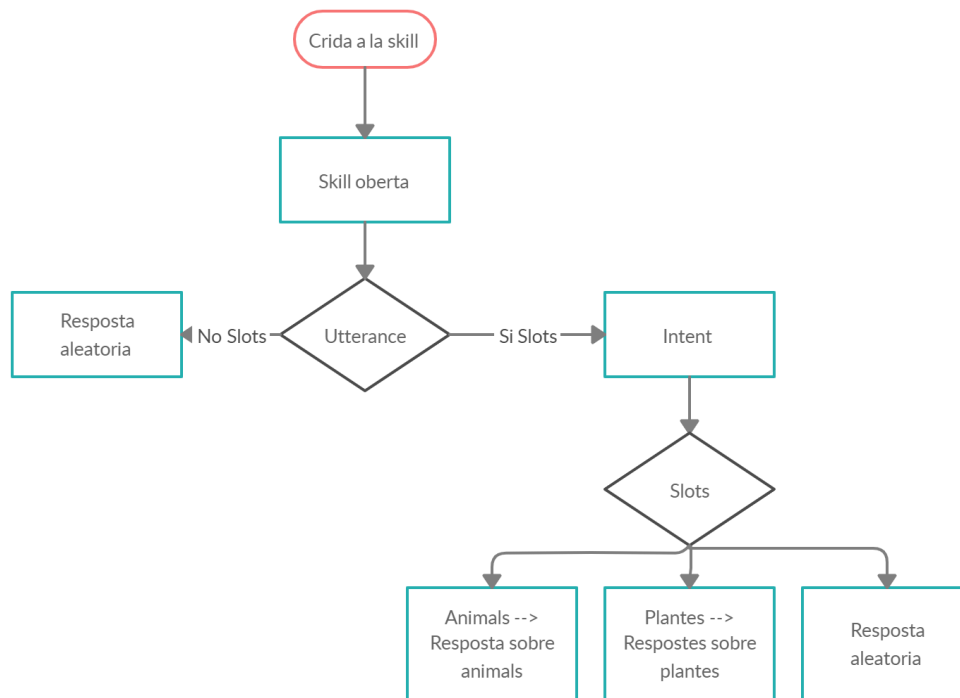


Figura 13 Diagrama de flux.

2.7.1 Alexa

Una habilitat d'Alexa té un model de interacció o una interfície de l'usuari de veu i una lògica d'aplicació. És a dir que quan un client parla, el que processa és un discurs en el context del vostre model d'interacció per determinar la sol·licitud del client. Tot seguit, Alexa envia una sol·licitud amb la proposta lògica d'aplicació d'habilitats actuant sobre ella. Per proporcionar la lògica de l'usuari com a un servei al núvol que s'allotja a Alexa, AWS o un altre servidor.

Un AWS és una plataforma del núvol que proporciona Amazon per als seus *developers* i així puguin fer les seves *skills*.



Figura 14 Procés que realitza Alexa quan l'usuari fa la seva proposta.

Les diferents funcions que pot fer les habilitats d'Alexa són:

- **Tipus d'habilitats múltiples.** És la tria del tipus d'habilitat que l'usuari vol depenent de com és l'habilitat que es vol crear. Es poden utilitzar APIs i/o models d'interacció que ja hi són integrats per a propòsits per a coses específics, com per exemple, una casa intel·ligent, o bé, la utilització de tipus d'habilitats personalitzades per poder crear diferents experiències que es poden imaginar. Es defineix el model d'interacció amb una habilitat.
- **API d'aplicació.** Alexa té incloses diferents API d'aplicació que l'usuari pot utilitzar al back-end per rebre les sol·licituds dels clients d'Alexa i informar-la de com ha de respondre. Les API també es poden utilitzar per a una personalització de la *skill*, vendre productes, aprofitar avantatges d'Alexa o enllaçar comptes de clients.
- **Kits de Software de Desenvolupament.** Aprofitar els ASK SDKs per a una reducció de complexitat del codi. Els SDK per Java, Node.js i Python tenen una funcionalitat específica del llenguatge per a tasques habituals. Així doncs, permetrà en segon pla el codi i no la lògica de la *skill* que l'usuari tingui en ment.
- **Eines de desenvolupament.** Hi ha una consola per als desenvolupadors que es pot utilitzar a la interfície de línia de ordres ASK (CLI) per crear, gestionar, testejar i publicar *skills*. També es poden integrar aquestes funcions en els sistemes amb l'API de gestió d'habilitats.
- **Analytics.** Es poden aprofitar les analítiques d'ASK per comprendre el que funciona bé i les oportunitats de millora. ASK proporciona una àmplia gamma de mètriques, inclosos els clients, intencions, sessions, declaracions i latència.

Alexa té un conjunt de capacitats integrades, és a dir, les *skills*. Aquestes habilitats poden reproduir música, respondre preguntes, proporcionar previsions meteorològiques i consultes a alguna web.

Tots els usuaris d'Alexa tenen accés a totes les noves habilitats fent preguntes o sol·licituds. Es poden crear *skills* que proporcionin als usuaris molts tipus diferents de *skills*. Per exemple:

- Preguntes específiques. Exemple, “*Alexa, preguntar sobre el temps que farà demà a Barcelona*”.
- Challenges d'usuaris de jocs o trencaclosques. Per exemple, “*Alexa juga a Akinator*”
- Control de llums i diferents dispositius de la llar. Per exemple, “*Alexa, encén la llum del menjador*”
- Proporció de contingut de text o àudio per a la publicació d'informació flash d'un client. Exemple: “*Alexa, dona'm una informació breu*”.

Es poden crear *skills* de diferents finalitats i per això, un *developer* ha de decidir que farà la seva *skill* com a primer pas. El kit que té Alexa per fer aquestes *skills* és molt variat i l'usuari pot escollir el que millor li convingui.

1. Es pot crear una *skill* que pot gestionar qualsevol tipus de sol·licitud. Per exemple, la cerca d'informació d'un servei web, la integració d'un servei web per demanar alguna cosa, demanar un cotxe a Uber, demanar menjar per *JustEat*, jugar a diferents jocs interactius, és a dir, qualsevol cosa que es pugui pensar.

Per poder crear-la es necessita fer amb una *custom skill*, ja que es defineixen unes peticions per gestionar la *skill* i les paraules que els usuaris diuen per invocar aquestes sol·licituds.

2. Una *skill* que permet a l'usuari controlar i consultar tots els electrodomèstics intel·ligents habilitats al *cloud*, com les llums, el televisor, els panys de portes, càmeres, termòstats i televisions intel·ligents.
El que es pot fer amb aquests es:

- Demanar apagar o encendre els llums.
- Canviar la intensitat de la llum o el color d'aquesta.
- Canviar la temperatura del termòstat.
- Veure en quina posició està el pany, és a dir, bloquejat o desbloquejat.
- Pujar o baixar el volum de la televisió.
- Canviar el canal de televisió.

Aquesta *skill* es crea a partir de *Smart Home API* ja que defineix les peticions que es poden gestionar les *skills* i les paraules que els usuaris diuen per poder invocar-les.

3. Una *skill* que permet a l'usuari controlar els serveis de vídeo habilitats al cloud.
 - Reproducció d'una pel·lícula.
 - La cerca d'un canal de televisió.
 - Canvi de canal.
 - Play, pausa, avançar o rebobinar un contingut de vídeo.

Per crear aquest tipus de *skills* es necessita la *Video Skill API*, defineix les peticions que pot gestionar l'habilitat i les paraules dels usuaris poden dir per fer les invocacions de les sol·licituds.

4. Una *skill* que proporciona contingut original per a una publicació flash d'un client.

Es crea a partir de la *Briefing Skill API*, ja que aquesta API defineix les paraules que diuen els usuaris per invocar la informació o peticions de notícies i el seu format del contingut per a que Alexa pugui proporcionar-lo.

5. Una *skill* que permet seleccionar, escoltar i controlar el contingut d'àudio que es transmet a través d'un dispositiu habitat per Alexa. Quan l'usuari vol fer una *skill* d'aquest estil utilitza *Music Skill API* per fer-la. És un model amb una interacció de veu que es defineix i es maneja pel desenvolupador mateix. Alexa interpreta les indicacions dels usuaris i envia missatges a la vostra habilitat que comuniquen aquestes sol·licituds.

Per crear un servei en el *cloud-based* que es gestionen les sol·licituds el tipus d'habilitat i l'amfitrió al cloud. El servei que dona Alexa encamina les sol·licituds entrants al servei que li pertoca.

A partir del tipus de skill requereix un servei diferent.

- Per una Custom Skill, es pot codificar amb una funció lambda AWS o un servei web:
 - *AWS Lambda, d'Amazon Web Service*, és un servei que us permet la execució de codi al cloud sense gestionar servidors. El que fa Alexa es enviar les sol·licituds d'usuari del codi i el codi pot inspeccionar la sol·licitud, fer totes les accions necessàries i després s'envia la resposta. Es poden escriure les funcions de Lambda amb diferents codis com *Node.js, Java, Python, C# o Go*. Una funció Lambda és una funció que no està lligat a cap identificador¹⁹.
 - Una altra opció, *web service* i allotjar-lo amb un proveïdor qualsevol d'allotjament en el cloud. La *web service* ha d'acceptar sol·licituds mitjançant HTTPS. En aquest cas, el que fa Alexa és enviar sol·licituds al servei web i aquest realitza les accions necessàries i envia una resposta. Podeu escriure el vostre servei web en qualsevol idioma.
 - Independentment de com creï el servei, també s'ha de crear *interaction model* personalitzat per a l'habilitat. En aquest cas, les *peticions* que pot gestionar la skill i les *paraules que els usuaris poden dir* per invocar aquestes sol·licituds.
- Per obtenir una habilitat que controla dispositius domèstics intel·ligents com llums, termòstats i dispositius d'entreteniment, podeu utilitzar l'*API Smart Home Skill*. En aquest cas, desenvolueu una funció Lambda AWS que accepta les directrius del dispositiu d'Alexa:
 - Proporciona el codi per gestionar directives en una funció Lambda AWS.
 - La skill rep sol·licituds en forma de *directrius de dispositiu* per controlar un dispositiu en particular. A més a més, el codi gestiona la sol·licitud adequadament. Per exemple, obrir les llums o pujar el volum de la televisió.
 - Totes les interaccions de veu amb l'usuari són gestionades per l'*API Smart Home Skill*. No s'han de definir les paraules per utilitzar l'habilitat.

¹⁹ «Expresión lambda - Wikipedia, la enciclopedia libre». https://es.wikipedia.org/wiki/Expresión_lambda (accedit juny 29, 2020).

→ Per obtenir una habilitat que controla el contingut de vídeo, podeu utilitzar l'6. En aquest cas, es desenvolupa una funció lambda que accepta les directrius del dispositiu d'Alexa:

- Proporcionar el codi per gestionar directives en una funció Lambda AWS.
- La skill rep sol·licituds en forma de *device directives* pel control d'un servei de vídeo. Aleshores, el codi gestiona les sol·licituds adequadament. Per exemple, demanar la reproducció d'una pel·lícula o una sèrie.
- Totes les interaccions de veu estan gestionades per l'API Video Skill. No s'ha de definir les paraules que es diuen per utilitzar el skill.

→ Per obtenir una habilitat que proporciona contingut com ara notícies, llistes o comèdia per presentar informació breu per part del client, podeu utilitzar l'API *Flash Briefing Skill*. En aquest cas, es crea l'habilitat a la consola de desenvolupadors i es configura un o més fluxos JSON o RSS que continguin el contingut:

- Per rebre el contingut com a part del seu *briefing flash*, un client habilita la seva *skill* d'informació flash a l'aplicació Alexa i activa un o més *feeds* de contingut.
- Totes les interaccions de veu amb l'usuari estan gestionades per l'API *Flash Briefing Skill*. No s'ha de definir les paraules que els usuaris diuen per utilitzar l'habilitat.
- Proporcionar un o més fluxos de contingut fiables en format RSS o JSON. El contingut pot ser contingut d'àudio que Alexa reproduïx al client o contingut de text que Alexa llegeix al client. Hauríeu de posseir el contingut o tenir els drets per distribuir-lo.

En el meu cas s'ha creat una *skill* customitzada a partir del servei que facilita Amazon amb el seu propi servidor.

Al haver escollit una *custom skill* i es vol dissenyar i crear s'ha de tenir en compte en el moment de la creació:

- Conjunt d'intencions (*intents*) que representen accions per a que usuaris puguin realitzar amb la seva *skill*. Aquests propòsits representen la funcionalitat bàsica de la *skill*.
- Un conjunt de *sample utterances* que especifiquen les paraules i les frases que els usuaris poden dir per invocar aquestes intencions. Assigna aquests indicis segons els propòsits. Aquest mapatge forma el *model d'interacció* per a l'habilitat.
- Un *nom d'invocació* que identifica l'habilitat. L'usuari inclou aquest nom en iniciar una conversa amb la *skill*. L'usuari escull quin serà el nom d'invocació.
- Si escau, un conjunt d'imatges, fitxers d'àudio i fitxers de vídeo que es volen incloure a la *skill*. S'han de guardar en un lloc accessible públicament de manera que cada element sigui accessible mitjançant un URL únic.

- Un *servei cloud-based* que accepta aquestes intencions com a sol·licituds estructurades i després actua sobre elles. Aquest servei ha d'estar accessible a través d'Internet. Proporciona un punt final per al vostre servei quan configureu la *skill*.
- Una configuració que combina tots els elements anteriors per tal que Alexa pugui encaminar les sol·licituds al servei per a la vostra habilitat. Podeu crear aquesta configuració a la *developer console*.

Per exemple, una *skill* per obtenir informació sobre les mareas pot definir una intenció anomenada *OneshotTideIntent* per representar la sol·licitud de l'usuari per cercar informació de marea per a una determinada ciutat costanera.

Aquesta intenció es correspon en diversos casos d'exemple, com ara:

- *OneshotTideIntent* obté una marea alta
- *OneshotTideIntent* obté una marea alta de {City}
- *OneshotTideIntent* dona'm informació sobre {City}
- *OneshotTideIntent* quan hi ha una marea alta a {City}
- ...

Es podrien crear molts més *utterances*, tantes com l'usuari desitgi. L'usuari podria començar la conversa per invocar la *skill* tal que:

Usuari: "Alexa, quan hi ha una marea alta a Seattle en **Tide Pooler**." En aquesta invocació es pot veure tota la primera part es la crida que s'ha implementat i després el nom de la *skill* per poder invocar-la.

En el moment que es parla amb un dispositiu que conté Alexa, fa el següent:

1. El discurs de l'usuari es transmet al servei d'Alexa al cloud.
2. Alexa reconeix que aquesta sol·licitud representa la intenció de *OneshotTideIntent* per a l'habilitat de "Tide Pooler".
3. Alexa estructura aquesta informació en una sol·licitud (en concret un *IntentRequest* en aquest exemple) i envia aquesta sol·licitud al servei definit per a Tide Pooler. La sol·licitud inclou el valor "Seattle" com a "Ciutat".
4. El servei "Tide Pooler" rep la sol·licitud i fa una acció adequada (cercant informació sobre les mareas per a la data actual a Seattle a <http://tidesandcurrents.noaa.gov/>).
5. "Tide Pooler" envia al servei Alexa una resposta estructurada amb el text per parlar amb l'usuari.
6. El dispositiu habilitat d'Alexa torna a respondre a l'usuari:

Tide Pooler: Avui a Seattle, la primera marea alta serà al voltant de les 2:46 del matí i arribarà a la màxima a uns 8 peus...

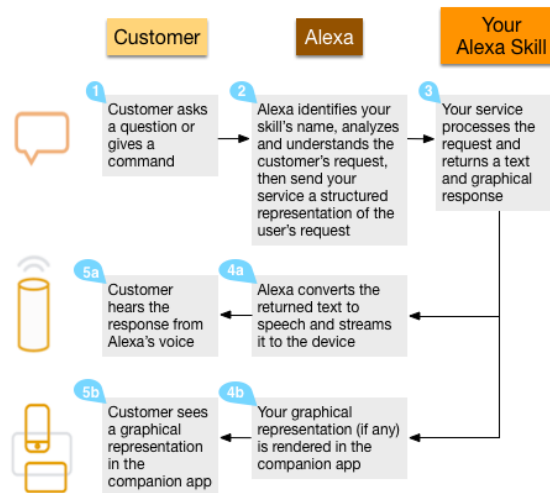


Figura 15 Esquema de com seria una conversa Usuari i Alexa per invocar una Skill.

Una *custom skill* sol rebre una pregunta o altra informació de l'usuari i després respon amb una resposta o alguna acció, com ara demanar un cotxe o una pizza. Les invocacions que poden fer els usuaris per demanar la *skill* són:

- Alexa , *obté una marea alta per a seattle* de Tide Pooler.
- Alexa , *Demana Receptes Com puc fer una truita?*
- Alexa , *pregunteu als horòscops diaris sobre el Taure.*
- Alexa , *dóna deu punts a Stephen* mitjançant el marcador de puntuació.

També es pot interactuar amb Alexa sense fer cap pregunta o una sol·licitud específica sobre la *skill* que s'ha creat.

- Alexa , *Open Tide Pooler.*
- Alexa , *Parlem amb les receptes.*
- Alexa , *Juga a Trivia Master.*
- Alexa , *mantenedor de puntuació d'inici.*

Els usuaris poden utilitzar aquesta opció si no saben o no recorden la sol·licitud exacta que volen fer. En aquest cas, la *skill* normalment retorna un missatge de benvinguda que proporciona als usuaris una ajuda breu sobre com utilitzar la *skill*. Son petites ajudes per l'usuari.

Si la vostra habilitat necessita més informació per completar una sol·licitud, podeu tenir una conversa molt més específica amb l'usuari:

- **Usuari:** *Alexa, diga'm la marea alta de Tide Pooler (Tot i que els mapes "obteniu la marea alta" al OneShotTideIntent, l'usuari no va especificar la ciutat. Tide Pooler ha de recopilar aquesta informació per continuar.)*
- **Tide Pooler:** *informació sobre marees de quina ciutat? (Ara Alexa escolta la resposta de l'usuari. Per a un dispositiu amb un toc de llum, com Amazon Echo, el dispositiu s'il·lumina per donar un toc visual)*
- **Usuari:** *Seattle*
- **Tide Pooler:** *Avui a Seattle, la primera marea alta serà a...*
La interacció finalitza, després de la conversa realitzada entre assistent i usuari.

L'aplicació Alexa d'Amazon és una aplicació gratuïta disponible per als navegadors web de Windows OS, Android, iOS i escriptori. Aquesta aplicació és rellevant per a la vostra habilitat personalitzada de dues maneres:

- L'aplicació mostra *fitxes de detall d'habilitats* per a totes les habilitats publicades. Els usuaris revisen aquestes targetes per a conèixer què fa la seva *skill* i com utilitzar-la quan decideixen si voleu habilitar la *skill*. Aquest procés es facilita proporcionant informació útil sobre la *skill* a l'hora de preparar-la per a la publicació.
- L'aplicació mostra *targetes de casa* que descriuen o milloren les interaccions de veu de l'usuari amb Alexa. Els usuaris poden veure més tard aquestes targetes, per obtenir més informació sobre la interacció o refrescar la seva memòria sobre la resposta d'Alexa.

Una cosa que s'ha de tenir en compte es que la *skill* pot incloure contingut a les targetes per fer les respostes.

Per exemple, l'habilitat de *Tide Pooler* envia una targeta d'inici que conté la informació de mareas que l'usuari sol·licita. La targeta d'inici ofereix a l'usuari una manera de veure la informació de la marea sense fer una altra sol·licitud de veu.

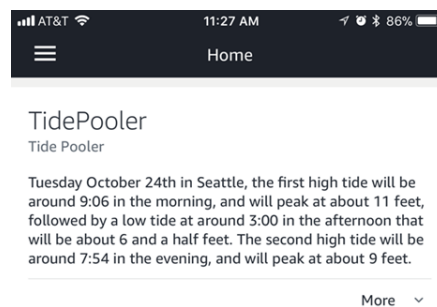


Figura 16 Fitxa d'inici per a la mostra de Tide Pooler.

També hi ha l'opció de crear *skills* que incorporen veu, pantalla i tacte per dispositius amb Alexa que tinguin una pantalla.

La *skill* es poden utilitzar els fitxers d'àudio amb AudioPlayer. Les skills dissenyades per un dispositiu també poden utilitzar imatges i fitxers de vídeo.

Tots aquests recursos externs han d'estar disponibles en un lloc web accessible públicament. Cada element té referència amb una URL únic que utilitza HTTPS.

En el cas d'aquest projecte s'ha escollit fer una skill personalitzada amb funció Lambda AWS amb el servidor web d'Amazon, ja que només executa el codi quan es necessita i s'executa automàticament, de manera que no cal proveir o executar servidors contínuament.

L'ús de la funció Lambda per al servei elimina la complexitat perquè:

- No cal administrar ni gestionar cap dels recursos de càlcul del servei.
- No necessiteu certificat SSL.
- No heu de verificar si les sol·licituds venen del servei Alexa. L'accés per executar la vostra funció està controlat per permisos dins d'AWS.

- AWS Lambda executa el codi només quan ho necessiteu i faci una escala amb l'ús que us faci, de manera que no hi ha necessitat de proveir o executar servidors contínuament.
- Alexa xifra les seves comunicacions amb Lambda utilitzant TLS.
- És un servei gratuït.

AWS Lambda admet diferents llenguatges com:

- Node.js
- Python
- Java

Totes les noves *skills* es poden crear amb diferents llenguatges i l'usuari pot escollir si fer-ho en un sol idioma o més d'un. Si s'utilitza el *SDK Kit de Skills* d'Alexa per node o Python. Es pot utilitzar unes plantilles ja dissenyades que te les facilita el dipòsit d'aplicacions sense servidors.

Per crear una *skill* des de zero ens hem de:

1. Crea un compte de AWS, si encara no la té el usuari.
2. Iniciar la sessió a la consola de gestió d'AWS i aneu a AWS Lambda.
3. S'ha d'entrar a Alexa developer console i dintre de la pestanya hi ha una casella que posa nova *skill*.

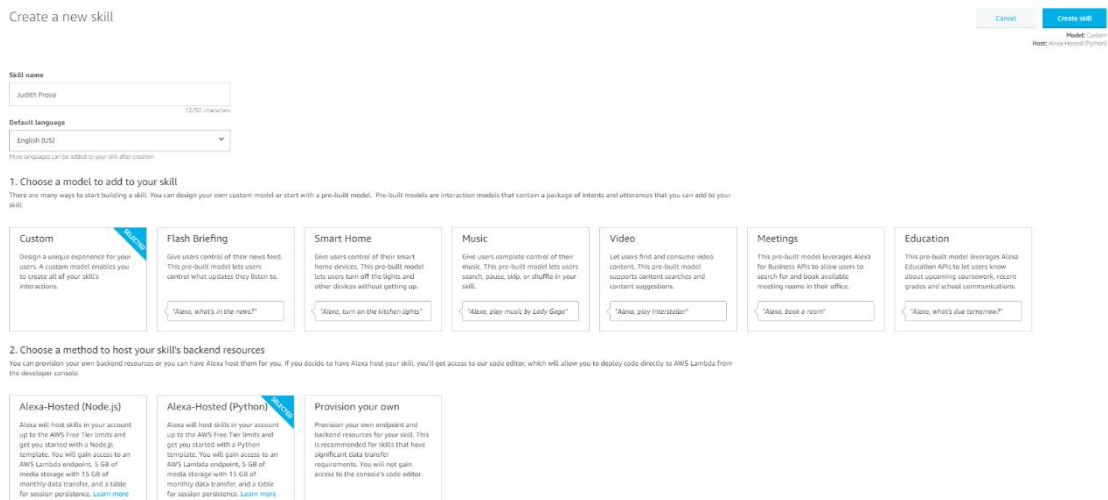


Figura 17 Finestra quan vols crear una nova skill a Alexa.

4. Allà dintre introdueixes les dades i esculls quin tipus de *skill* vols.
5. Un cop fet, passem a la següent pestanya i ens diu si volem escollir alguna plantilla que estan de exemples per l'usuari. En el meu cas, he escollit la *skill* de *Fact Skill*.

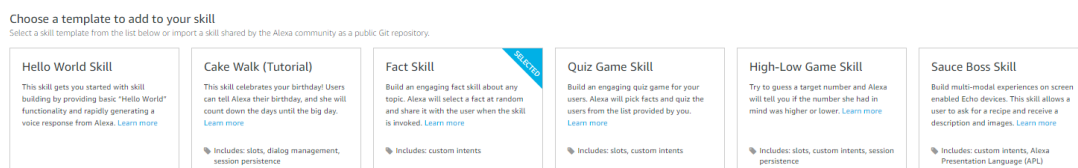


Figura 18 Exemples predeterminats que es poden escollir per crear la skill a Alexa.

Es configurarà i el primer pas que s'ha de fer és canviar la invocació de la *skill* perquè no en té. En el meu cas la *skill* es diu *living things*.

A més a més, en la funció que es crea *GetNewFactIntent* on se li introdueixen *Sample Utterances*, així quan la crides i/o parles amb ella, s'ha de fer a partir d'aquestes utterances que ha creat l'usuari.

També s'ha de tenir en compte que se li agregarà un Slot amb la finalitat de que quan un demana Alexa una cosa més específica li doni la resposta del slot que ha demanat i no una resposta aleatòria del que s'ha demanat prèviament.

La meua *skill* té un slot type que es diu *Living beigns* on els valors són animal i planta. Per cridar-lo a la funció *GetNewFactIntent*, a baix d'aquesta hi ha un apartat que posa *Slots*

Usings, llavors allà es posa un nom per quan es facin la crida i dius que el seu *slot* types és..., en el meus cas *living_beigns*.

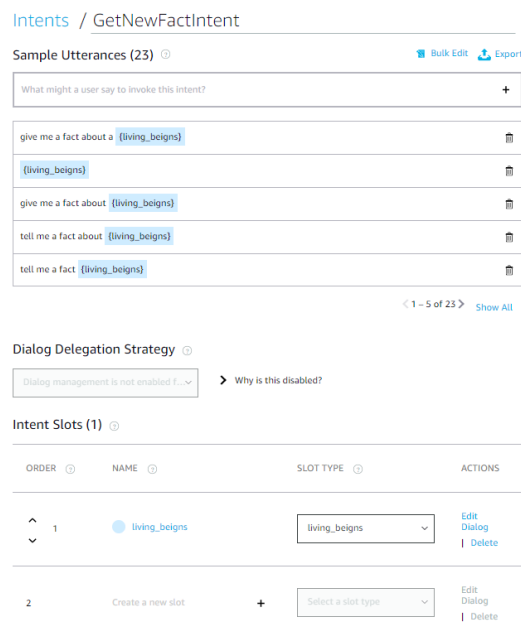


Figura 19 Utterances de skill creada.

Crída de *slots* a la funció *GetNewFactIntent*. Una vegada realitzat a les *samples utterances* pots cridar els *slots* a partir de {}.

Al finalitzar el *front-end* de la *skill* toca el *back-end*, la part del codi de la habilitat creada. En la pestanya code es pot veure el codi que donen amb la plantilla escollida, doncs es fa la configuració a partir d'aquest.

En el meu cas, només he introduït aquestes línies de codi.

Per una banda, s'han cridat dues llistes per posar les curiositats d'animals i de plantes per separat i un altre que es la unió de les dues per quan no demana una cosa específica.

ANIMALS = [

'The heart of a shrimp is located in its head.',


```
'A snail can sleep for three years.',  
'Slugs have four noses.',  
'It is possible to hypnotize a frog by placing it on its back and gently  
stroking its stomach.',  
'A cow gives nearly 200,000 glasses of milk in a lifetime.'  
]  
PLANTS = [  
    'Leaving the skin on potatoes while cooking is healthier as all the  
    vitamins are in the skin.',  
    'Banana is an Arabic word for fingers.',  
    'Carrots were originally purple in colour.',  
    'The baobab tree found in Africa can store 1,000 to 120,000 litres of  
    water in its swollen trunk.',  
    'Bamboo is the fastest-growing woody plant in the world; it can grow 35  
    inches in a single day.'  
]  
FACTS = ANIMALS + PLANTS
```

Dintre d'una classe anomenada *GetNewFactHandler*. Unes línies més de codi per invocar en una nova variable *lb_value*, el valor que dona quan l'usuari demana un *slot* en concret i després s'ha creat un condicional per diferència quan es un valor aleatori, un animal, una planta i com a última opció et respon una frase de que hi ha una cosa errònia quan estàs creant el codi.

```
dades = handler_input.request_envelope.request  
lb_value = dades.intent.slots['living_beings'].value  
print(lb_value)  
  
if lb_value is None or "any" in lb_value:  
    random_fact = random.choice(FACTS)  
elif "animal" in lb_value:  
    random_fact = random.choice(ANIMALS)  
elif "plant" in lb_value:  
    random_fact = random.choice(PLANTS)  
else:  
    random_fact = "If you hear this something has gone really wrong"
```

Un exemple de com quedaria una conversa amb aquesta *skill* seria:

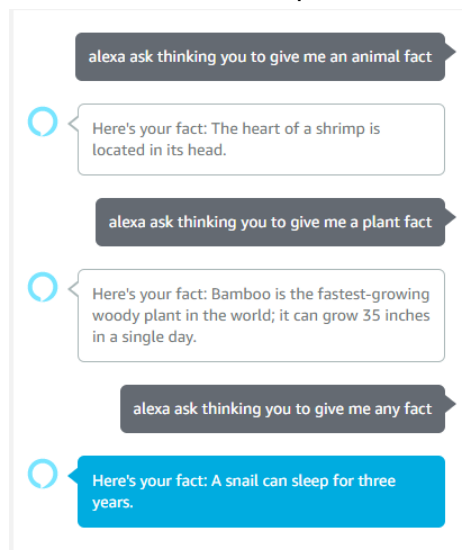


Figura 20 Exemple de conversa amb la skill creada.

Un cop feta la habilitat que l'usuari ha realitzat es pot publicar per poder baixar-la a diferents dispositius de diferents usuaris i si la poden utilitzar.

S'ha d'anar a l'apartat *Distribution* i anar omplint tota informació necessària per poder fer la publicació. S'ha d'escriure el nom de la *skill*, una breu explicació i una de més detallada, diferents exemple de com es farien les crides de la *skill*, una imatge per l'icona, la categoria. Després de desar i continuar s'ha d'especificar si:

- S'ha de pagar per aquesta *skill*.
- Si és necessària informació personal, com correu electrònic, contrasenyes, data d'aniversari, ...
- Si és per menors de 13 anys, és a dir, una habilitat infantil.
- Si conté publicitat.

En el meu cas a totes les opcions he dit que he certificat el compliment d'exportació i s'han de donar unes instruccions per a que Amazon pugui realitzar un petit test.

Com a últim apartat és la disponibilitat, on s'ha d'especificar si es per l'àmbit públic o privat, si és per totes les regions o només en uns països en concret.

Just després, ens enviarà a l'apartat de Validation on el farem córrer i es pot observar si n'hi ha errors o tot és correcte. Quan no es troben errors, es realitzarà enviament, per a que es verifiqui i es publiqui.

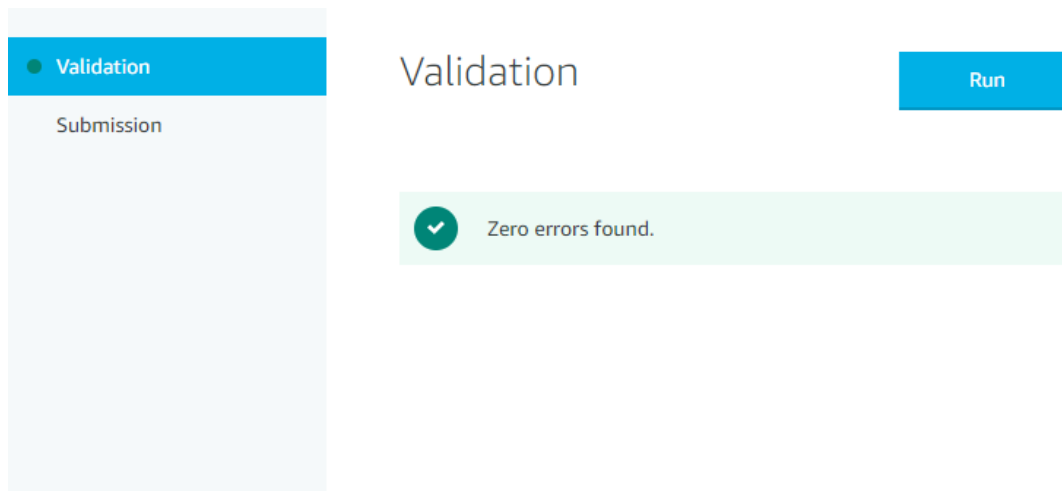


Figura 21 Validació de la Skill de Amazon.

A partir d'aquest moment que s'envii s'ha d'esperar que es certifiqui i es publiqui la *skill* al mercat i es pugui utilitzar amb els dispositiu desitjat.

2.7.2 Google Assistant

Actions de Google és una plataforma que permet crear accions al Assistent de Google. Aquestes accions són interaccions que es connecten amb l'assistent de veu i així també poden permetre donar resposta a una necessitat concreta, com per exemple, demanar el temps de un lloc determinat o fer una compra online. D'altra banda, es poden fer servir en múltiples dispositius, com mòbils, televisors o rellotges de *smartwatches*.

Els usuaris poden invocar accions pel seu nom, però també es poden afegir diferents frases per poder indicar d'una forma més exacta el que es vol fer. Com per exemple, una *action* de temps que es vol saber si plourà o no en algun lloc determinat. Llavors, el que s'hauria de fer es invocar amb el nom d'aquesta acció, per exemple, "Tempestes" o dir "Ok Google, parla amb Tempestes per saber si plourà demà a Barcelona".

No cal instal·lar per utilitzar una acció, però si s'ha d'enviar a producció perquè sigui aprovada. Però sí que hem d'utilitzar una acció abans de que sigui publicades en el mateix compte on l'hem creat.

D'altra banda, una de les parts del desenvolupament de les accions és saber com planejar com els usuaris poden descobrir-les.

Per últim, s'ha de dir que Google proporciona analítiques de l'ús, descobriment de les accions, les converses que es donen i la seva llargada.

Els conceptes bàsics per a una acció de Google són:

- **Assistent de Google:** assistent virtual de Google per permetre converses entre els usuaris i Google, donant resposta a alguna necessitat.
- **Dialogflow:** eina de Google per desenvolupar experiències conversacionals que puguin utilitzar-se amb l'Auxiliar de Google. Utilitza *machine learning* per analitzar

l'input de l'usuari i identificar el que vol fer per poder respondre de la manera més correcta.

- **Agent:** projecte de *DialogFlow* per administrar certes experiències en un o múltiples idiomes.
- **Intent:** es tracta de l'objectiu que té l'usuari quan parla amb l'Auxiliar de Google. Els agents de *Dialogflow* es creen amb dos Intents per defecte: el de benvinguda i el de fallback, que s'executa quan no s'identifica què va intentar utilitzar. És recomanable no esborrar-los, sinó modificar-los perquè s'adapti a les nostres necessitats.
- **Invocació:** és quan un usuari inicia la interacció amb una acció. Pot fer-ho explícitament, indicant el seu nom, o implícitament, indicant què vol fer. A més, existeixen els esdeveniments, que són una característica que permet que una acció s'invoqui programàticament en comptes de directament per l'usuari.
- **Paràmetre:** concepte que proporciona l'usuari a través de la seva interacció amb l'Auxiliar de Google i què farem servir per donar una resposta correcta a la necessitat. Per exemple, si volem comprar-ne deu cartrons de suc per al dilluns, es podria identificar "cartrons de suc", "deu" i "dilluns" com a paràmetres. Aquests paràmetres poden ser d'un tipus definit com una entitat.
- **Entitat:** es tracta de representacions de conceptes reals i que es poden utilitzar en les frases d'entrenament perquè *Dialogflow* identifiqui correctament els paràmetres. Per exemple, es podria tenir una entitat que representi les opcions de compra amb "cartró de llet", "batut" i "gelat" com a opcions. *Dialogflow* proporciona algunes entitats per defecte, però també podem afegir les nostres.
- **Fulfillment:** lògica que administra un esdeveniment per realitzar l'acció corresponent. Un fulfillment pot ser una aplicació externa, un servei, ...
- **Frases d'entrenament:** exemples de possibles interaccions de l'usuari per trucar a un intent en concret. Aquestes frases són utilitzades per un algoritme que entrena a l'agent perquè pugui entendre el que l'usuari vol. És recomanable proporcionar el menys 20 frases.

El funcionament de Google assistent consisteix, primer de tot en elaborar la petició que s'enllaça amb un agent. Una vegada identificat l'agent utilitza, s'intentarà aparellar el que l'usuari sol·licita amb un intent. Si s'aconsegueix, s'intentarà recuperar els paràmetres necessaris. Si no tots els paràmetres obligatoris s'han proporcionat o si no s'han identificat, s'han de sol·licitar a l'usuari.

Tot seguit, s'ha de retornar una resposta. En el cas que sigui una resposta estàtica o amb una lògica mínima es farà directament. No obstant, per a una complexitat o la utilització de bases de dades o APIs externes, serà necessari un *fulfillment*, el qual farà el tractament necessari.

Per crear la acció de google, el primer cas que s'ha de fer és la connexió amb la consola de *Actions de Google*, i al accedir es demanarà un compte de Google.

Una vegada connectats, es crea un nou projecte, proporcionant un nom, el país i l'idioma.

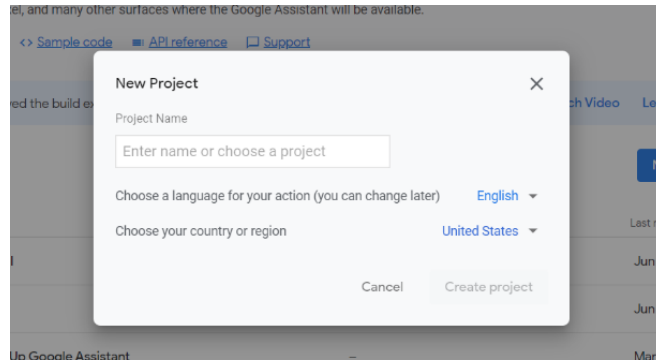


Figura 22 Inicialització del un projecte.

En la següent pantalla s'ha de escollir l'opció *Conversational* de totes les opcions que hi ha, per tal que per realitzar una *action* personalitzada com es el nostre cas.

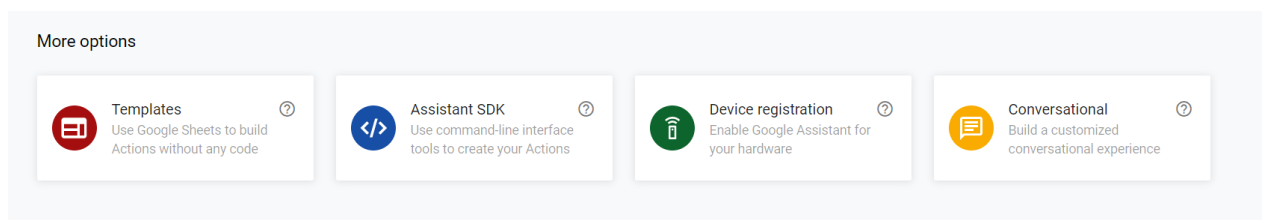


Figura 23 Opcions per escollir la tipologia de la skill

Dintre dels projecte, s'ha d'anar a *Invocation*, que es troba al menú del costat esquerra i s'escriu el nou que es vol per invocar a l'acció que s'ha creat. Es dona a guardar.

El següent pas, es crear la acció per al que anem a *Actions* que es troba al menú i afegim una. S'ha haurà de escollir la opció de *custom* i directament s'obrirà una pestanya nova amb *Dialogflow*.

Una vegada s'ha entrat a *Dialogflow* s'ha d'escollir l'idioma i la zona horària.

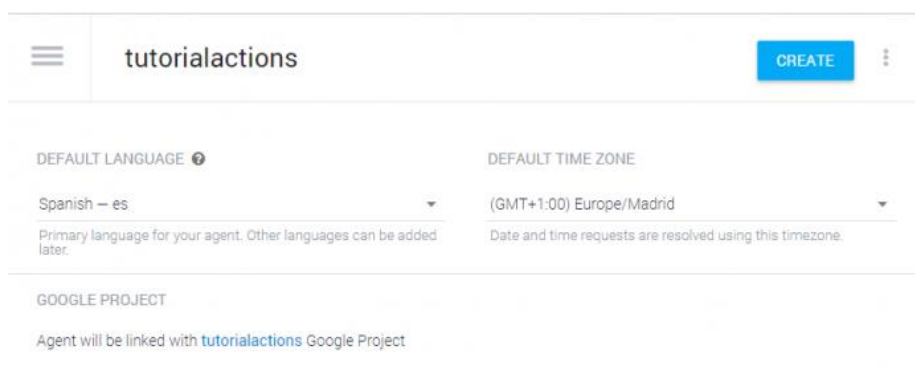


Figura 24 Inicialització del un projecte.

Després, s'ha de fer la configuració del Intent. S'ha de crear un de nou introduïm tot el necessari. Quina son les frases d'entrenament i els paràmetres que s'han d'utilitzar. Es comença a indicar tots els paràmetres, per el qual baixem fins la secció corresponent. On s'introdueixen si és necessari els nous paràmetres que siguin necessaris i després s'introdueixen les frases d'entrenament.

Una vegada realitzat aquests passos toca la part de *Fulfilment*, primer de tot s'ha d'habilitar. Un cop fet, s'ha de canviar el codi pell que l'usuari vulgui.

S'ha de tenir en compte que el nom del intent després ha d'estar reflexat al codi i no es pot canviar, sinó no cridarà bé a la funció.

Un cop fet tocarà fer les diferents proves per comprovar que la *action* realitzada s'està executant correctament.

En el meu cas, no he pogut realitzar aquest tipus d'*action* pel fet que quan vols editar el codi et reenvia a la consola de google cloud i el fet d'introduir una targeta de dèbit o crèdit, editar-lo s'ha decidit buscar una diferent forma a fer-ho, ja que un dels objectius que es volen aconseguir és fer una acció amb una inversió inicial baixa, és a dir nul·la.

La solució ha sigut realitzat una *action* a partir d'una plantilla de Google i que no s'hagi d'utilitzar en cap moment la consola de *Google Cloud*.

Per començar s'ha creat un nou projecte i en compte d'escollir *Conversational* escollim *templates*. Allà hi hauran diferents opcions i s'escull *Trivia*. Després en una nova pestanya clicarem a *Get Started* i es seguiran les pautes que donen.

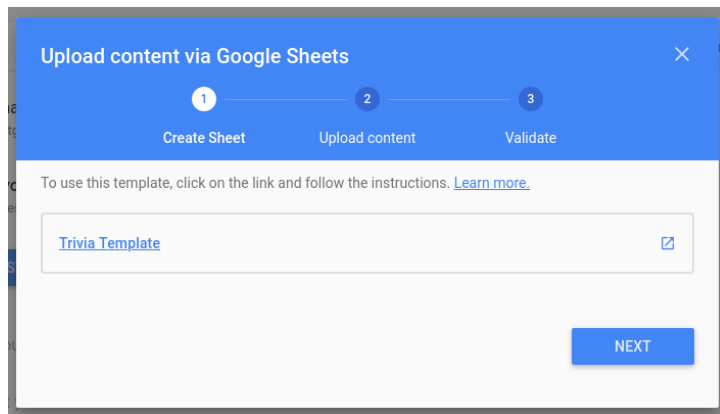


Figura 25 Primer pas per poder crear l'accion a google.

S'obre una nova pestanya on es crea un Excel, es fa una còpia d'aquest i a partir d'allà es va editant l'Excel al gust de cada usuari.

Question	Correct Answer	Incorrect Answer 1	Incorrect Answer 2	Follow Up
Note that synonyms for answers that aren't just partial matches can be separated with the character. Remember to change your game title and settings in the Configuration tab.				
How long is a chameleon's tongue?	10 inches	7 inches	16 inches	On average, a chameleon's tongue is roughly twice the length of its body.
What instrument does Eric Clapton play?	Guitar	Keyboards	Trumpet	
The TV show American Idol aired for how many seasons?	15	10	24	
How long can an alligator survive in captivity?	50 years	45 years	30 years	
The world's tallest waterfall is?	Angel Falls	Victoria Falls	Niagara Falls	The falls are 2230 feet in height.
Which author created the character James Bond?	Ian Fleming	Robert Ludlum	Sean Connery	He was an English author, journalist and naval intelligence officer.
Which year was the moon landing?	1969	1970	1966	
What is the capital of Texas?	Austin	Dallas	Houston	
How many countries are in South America?	12	22	33	
Which of these US states is farthest West?	Alaska	Nebraska	Hawaii	
How many gigabytes are in a terabyte?	1024 1k		1000	256
What are the official languages of Canada?	English and French French Swedish and French French English and Spanish Spanish and English			
Yankee Stadium is in which part of New York City?	The Bronx	Brooklyn	Queens	
Which country has the most roads?	United States USA US	India	China	
Wikipedia was launched in 2001.	VERDADERO	FALSO		Wikipedia was launched on 15 January 2001, two days after the domain was registered by Jimmy W.

Figura 26 Excel creat amb les preguntes i respostes del Trivial.

Una vegada acabat, es van fer comprovacions per poder validar-ho en l'apartat *Test*. També s'ha de donar una descripció de l'acció i donar la informació necessària per a que un cop

estigui creada i validada es pugui publicar i els usuaris puguin utilitzar-la com la resta d'actions que hi ha.

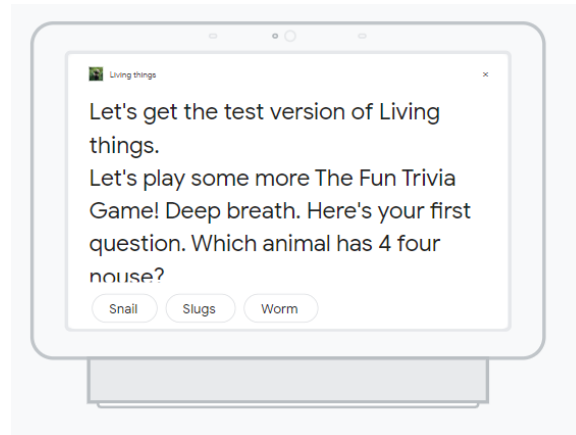


Figura 27 Test de l'acció de google.

En el cas de google, al crear el producte com l'acció amb la mateixa compte, es pot utilitzar directament al nostre dispositiu, és a dir, si invoques l'acció amb "Talk to Living things" directament et farà el trivial de preguntes sobre animals i plantes.

2.7.3 Mycroft

Les *skills* de Mycroft, són les aplicacions de veu que ofereixen diferents funcionalitats pels usuaris. Per poder crear una *skill* s'ha de tenir la experiència d'una tècnica bàsica, de d'instal·lació i de tenir Mycroft per poder realitzar-la.

Les habilitats en Mycroft s'escriuen a partir del llenguatge de programació de Python. Totes elles quedaran allotjades en Github, que l'usuari s'ha hagut de crear prèviament. Encara que, s'han de tenir uns bàsics per com funciona la plataforma. *Mycroft Skills Kit* també automatitza molts dels aspectes de la gestió de depòsit en *Github*.

Per realitzar una *skill* s'ha de tenir en compte les terminologies que s'han d'utilitzar:

- Utterance: Una utterance, declaració en català, és una frase que l'usuari diu, després que l'usuari digui la paraula per invocar la *skill*. Per exemple: "quin és el temps que fa a Barcelona?".
- Dialog: un diàleg és una frase de la qual ens parla Mycroft. Diferents habilitats tindran diferents diàlegs, segons el que faci l'habilitat. Per exemple, en una habilitat meteorològica, un diàleg pot ser *el.minimum.temperature.is.dialog*.
- Intent: Mycroft coincideix amb les declaracions que un usuari parla amb una habilitat determinada a una intenció de la declaració demanda. Per exemple, si un usuari parla "Hey Mycroft, com és el clima a Toronto?" a continuació, la intenció s'identificarà com a meteorològica i es combinarà amb la *skill* de Meteorologia. Quan desenvolueu noves habilitats, heu de definir nous propòsits.

Per poder instal·lar o desinstal·lar una *skill* que ja estigui creada hi ha dues opcions a fer-ho, demanant-lo a Mycroft directament, com introduint-lo al terminal.

```
Hey Mycroft, install {skill name}
```

```
Hey Mycroft, uninstall {skill name}
```

```
$ msm install skill-name
```

```
$ msm remove skill-name
```

Per la creació d'una *skill* a Mycroft s'ha de tenir en compte:

- L'usuari ha de tenir un coneixement bàsic sobre la programació de Python.
- Un compte de GitHub.
- Treballar amb una versió de Mycroft que anteriorment l'usuari s'ha instal·lat al seu dispositiu, seleccionat.

Com anteriorment s'ha mencionat, *Mycroft Skills Kit* és una comanda que ajuda a crear una nova *skill*.

`Mycroft-msk-create`

Aquesta comanda és un *script* interactiu que realitza unes preguntes i genera una nova plantilla de la *skill*. Aquesta plantilla es pot utilitzar com a *skill* directament o es pot ampliar fins que l'usuari la tingui amb les funcionalitats pertinents.

Per crear la plantilla per la *skill* es demanarà:

1. Nom: El nom de la nova *skill*. El nom ha de ser curt, generalment amb 22 caràcters de longitud. El nom també ha de ser únic. Es pot consultar el *Marketplace* per veure quines altres habilitats ja existeixen.
2. Frases d'exemples, conegudes com a *utterances*. Declaracions que espereu que els usuaris diguin a Mycroft, al qual us respondrà la vostra habilitat.
3. Diàleg de resposta. El diàleg amb què respondrà la vostra habilitat.
4. Breu descripció. Una descripció d'una sola línia de menys de 40 caràcters.
5. Descripció llarga. Pot ser el més curt o el més llarg.
6. Autor. el més sovint és el vostre nom i/o nom d'usuari de Github. Per facilitar-ho et fan entrar en un enllaç de github per crear un *token* on l'usuari a de dir el que pot fer amb la *skill* i si es podria editar la *skill* o tenir modificacions d'aquesta.
7. Categories. El mercat de competències de Mycroft al qual pertany la vostra habilitat. És important tenir en compte que la primera categoria que seleccioneu es definirà com a categoria predeterminada. Aquí és on la vostra habilitat apareixerà més sovint al *Marketplace*.
8. Les etiquetes, proporcionen un mitjà addicional per a que els usuaris puguin cercar o descobrir habilitats rellevants. A diferència de les categories, es poden configurar les etiquetes com l'usuari desitgi.

Després d'introduir aquestes dades, es demanarà si voleu crear una representació de Github per a la vostra habilitat. Això us proporciona una manera senzilla d'emmagatzemar les vostres habilitats i us serà obligatori si decidiu publicar la vostra habilitat al *Marketplace*.

Si heu completat tots aquests passos, la vostra habilitat s'haurà creat al directori `/opt/mycroft/skills` del dispositiu i per últim demana que s'introdueixi un icona per la app i un color que s'ha d'indicar en forma hexadecimal.

A la següent secció, explorarem cada component que compon la vostra habilitat.

Una vegada que estigui realitzada la plantilla es pot entrar dintre i modificar-la per a que surti la *skill* de la manera que l'usuari desitgi. L'estructura de la *skill* conté diferents fitxers i directoris.

```
$ ls -l
```

```
drwxr-xr-x 3 kris kris 4096 Oct  8 22:21 dialog
-rw-r--r-- 1 kris kris  299 Oct  8 22:21 __init__.py
-rw-r--r-- 1 kris kris 9482 Oct  8 22:21 LICENSE
-rw-r--r-- 1 kris kris  283 Oct  8 22:21 README.md
-rw-r--r-- 1 kris kris  642 Oct  8 22:21 settingsmeta.yaml
drwxr-xr-x 3 kris kris 4096 Oct  8 22:21 vocab
```

En els següents directoris, vocab, dialog i locale contenen subdirectoris per a cada idioma parlat que pot suportar.

Quan es crea una habilitat per defecte conté un subdirectori amb un únic idioma, l'anglès de Estats Units.

En el directori dialog, hi haurà un fitxer a la subdirecció d'idiomes (per a cada tipus de diàleg que utilitzarà l'habilitat. Actualment, contindrà totes les frases que introduïu quan creeu l'habilitat. Per exemple:

```
Sample1.dialog
```

Quan se li encarrega utilitzar un diàleg particular, Mycroft triarà una d'aquestes línies a l'atzar. Això s'acosta més a la parla natural. És a dir, moltes frases similars signifiquen el mateix.

Per exemple la invocació de la *skill*:

- Diga'm una curiositat
- Necesito saber una curiositat sobre plantes
- Diga'm alguna cosa sobre animals.

En el directori de vocab es defineixen un o més propòsits. Aquest directori està organitzat per l'idioma.

Cada habilitat defineix un o més propòsits. Aquests es defineixen al directori vocab. El directori vocab està organitzat per idioma, igual que el directori de diàleg.

En breu coneixerem els propòsits. De moment, podem veure que dins del directori vocab es pot trobar diversos tipus d'arxius:

- Els fitxers *.intent* utilitzats per definir els Padatious Intents. En aquest fitxer es troba les utterances que s'han introduït quan s'ha creat la *skill*.
- Els fitxers *.voc* defineixen paraules clau utilitzades principalment en Intents adaptats.
- Els fitxers *.entity* defineixen una entitat amb nom que també s'utilitza en Intents adaptats.

En el directori *locale* és la combinació dels directoris de *dialog* i de *vocal* en un únic directori.

El fitxer `__init__.py` es on defineix la major de la *skill* creat amb llenguatge Python. En aquest fitxer. Es crida els *intents* que s'han creat i indica quina resposta vol que respongui el dispositiu. Per fer-ho s'han d'utilitzar els intents on s'inicia la funció a partir de `@intent_handler`. D'aquesta forma es crida l'*intent* d'una forma clara.

Per exemple en la *skill Hello World* un intent el declaren:

Un gestor d'intencions padatius es va activar mitjançant una llista de frases de mostra.

```
@intent_handler(IntentBuilder('ThankYouIntent').require('ThankYouKeyword'))
```

```
def handle_thank_you_intent(self, message):
```

```
    self.speak_dialog("welcome")
```

La funció demana dos parametres:

- Auto: referència al propi objecte *HelloWorldSkill*.
- Missatge: un missatge entrant del bus de missatges.

Normalment també tindreu un mètode `stop()`, detecta una intenció d'aturada.

El bloc final de codi de la nostra habilitat és la funció *create_skill* que retorna la nostra nova habilitat:

A més a més, el fitxer *LICENCE* conté el text complet de la llicència a la qual es distribueix l'habilitat. No està obligat per a que treballi l'habilitat, però totes les habilitats enviades al mercat han de ser publicades sota una llicència de codi obert adequat.

El fitxer *README*, conté informació que es pot llegir sobre la vostra habilitat humana. La informació d'aquest fitxer s'utilitza per generar l'entrada d'Habilitats al Marketplace. Podeu trobar més informació sobre aquest fitxer a la secció Enviament del mercat.

Aquest fitxer, *settingsmeta.yaml*, defineix la configuració que estarà disponible per a un usuari a través del seu compte a *Home.Mycroft.ai*.

Una vegada realitzada el codi de tots els fitxers, s'ha de testejar a partir d'entrar a la màquina, és a dir, `mycroft-cli-client` i intenta invocar-la a partir de les *utterances* que s'han facilitat per poder veure com reacciona el dispositiu.

S'ha de tenir en compte que la *skill* quedarà guardada en el directori de l'usuari de github que s'ha donat en el moment de crear a *skill*. Tanmateix, en el mateix dispositiu on s'ha creat.

Un cop es tenen localitzats els diferents arxius, s'ha d'anar al fitxer `__init__.py`, amb aquest fitxer tenim l'estructura principal de tota la *skill* perquè introduïm tot el codi amb llenguatge Python i s'especifica que es que fa la *skill*.

S'importen llibreries, es visualitza la crida que farà l'usuari i s'escollirà una resposta en mode aleatori.

```
from os.path import dirname, join
```

```
from adapt.intent import IntentBuilder
```

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

def get_fact(language = 'en', category = None):
    animal = [
        'The heart of a shrimp is located in its head.',
        'A snail can sleep for three years.',
        'Slugs have four noses.',
        'It is possible to hypnotize a frog by placing it on its back and gently
stroking its stomach.',
        'A cow gives nearly 200,000 glasses of milk in a lifetime.' ]
    plant = [
        'Leaving the skin on potatoes while cooking is healthier as all the
vitamins are in the skin.',
        'Banana is an Arabic word for fingers.',
        'Carrots were originally purple in colour.',
        'The baobab tree found in Africa can store 1,000 to 120,000 litres of
water in its swollen trunk.',
        'Bamboo is the fastest-growing woody plant in the world; it can grow 35
inches in a single day.']
    if category == "animal":
        return choice(animal)
    elif category == "plant":
        return choice(plant)
    else:
        return choice(animal + plant)

livingthings_types = ['animal', 'plant']
class LivingThings(MycroftSkill):
    def __init__(self):
        MycroftSkill.__init__(self)
```

```
def speak_fact(self, lang, category):
    self.speak(get_fact(language=lang, category=category))

@intent_handler(IntentBuilder("FactIntent").require("Fact"))
def handle_general_fact(self, message):
    selected = choice(livingthings_types)
    self.speak_fact(self.lang[:-3], selected)

@intent_handler(IntentBuilder("AnimalFactIntent").require("Fact")
                .require("Animal"))
def handle_animal_fact(self, message):
    self.speak_fact(self.lang[:-3], 'animal')

@intent_handler(IntentBuilder("PlantFactIntent").require("Fact")
                .require("Plant"))
def handle_plant_fact(self, message):
    self.speak_fact(self.lang[:-3], 'plant')

@intent_handler(IntentBuilder("MineralFactIntent").require("Fact")
                .require("Mineral"))
def handle_mineral_fact(self, message):
    self.speak_fact(self.lang[:-3], 'mineral')

def stop(self):
    pass

def create_skill():
    return LivingThings()
```

En el directori de test s'afegeixen 4 exemples per saber com serien les crides a la *skill* i en el directori de *vocab* tenir els *slots* de que s'han escollit, en el meu cas, animals, plantes i *facts* en mode genèric.

```
(.venv) pi@picroft:~/mycroft-core/skills/living-things-skill/test/intent $ ls
sample1.intent.json  sample3.intent.json
sample2.intent.json  sample4.intent.json
(.venv) pi@picroft:~/mycroft-core/skills/living-things-skill/test/intent $ cat sample1.intent.json
{
  "utterance": "give me a fact",
  "intent_type": "FactIntent",
  "intent": {
    "Fact": "fact"
  }
}
```

Figura 28 Directori test de la Skill de Mycroft.

Per poder testejar la *skill* s'ha d'anar al dispositiu i provar des d'allà, és a dir, obrir mycroft-client i fer les crides que s'han introduït en el directori test. En el cas de Mycroft, no és necessari publicar la *skill* perquè directament es treballa en el mateix dispositiu i si anteriorment vincles la *skill* al github pots donar accés a diferents

3. Resultats

3.1. Àmbit d'utilització

D'entrada, el dispositiu que s'ha realitzat és senzillament un mitjà de proves. D'aquesta manera tenim diferents dispositius a partir de només el canvi de targetes SD. En altres paraules, al fer l'estudi s'han pogut estudiar els diferents assistents només amb un Raspberry Pi.

El que es vol buscar és un *hardware* que sigui versàtil per poder fer un canvi de màquina amb la millor facilitat possible, amb transparència i així, poder saber tot el que passava en cada moment i gràcies a la Raspberry Pi es pot comprovar, ja que és un mini-processador a la vegada que té totes les facilitats per convertir-se en un ordinador.

3.2. Comprovació dels assistents

Per fer la validació de cada disseny s'ha volgut realitzar un conjunt de proves on el que es vol valorar és com respon a tot el que pot realitzar en la vida quotidiana, com és una comunicació entre usuari i dispositiu, si demanes que t'obri una *skill*, encendre la ràdio entre d'altres.

	Alexa	Google Home	Mycroft
Paraula clau per invocar al dispositiu	“Alexa”	“Ok Google”	“Hey Mycroft”
Problemes per comunicar-se usuari-dispositiu	Cap problema, sempre que l'usuari vocalitza i parli de forma adequada i coherent.	El primer cop que es parla amb Google respon en anglès i després ja respon en castellà, encara que s'ha configurat per a que parli en castellà.	Depenen de com se li expressi el que l'usuari necessita en el moment pot ser que arribi a dir que no entén el que se li ha demanat o que es torni a demanar el mateix però amb altres paraules.
Respon a tot?	Sí, però si es diuen insults o paraules malsonants es calla i no respon	A vegades es bloqueja i no vol sortir de l'acció en la que es troba, però normalment funciona bastant correcte.	Si, no te cap problema, encara que a vegades quan comença a parlar es menja la primera paraula.
Test de micròfon, altaveus, S'escolta correctament?	El micròfon i els altaveus han estat configurats prèviament i d'aquesta manera Alexa respon a tot correctament.	Fent la configuració per obtenir el dispositiu és suficient.	S'ha fet una configuració prèvia tant dels altaveus com la del micròfon perquè el moment de l'acció s'escolti, processi i respongui el que pertoca per cada crida al dispositiu.
Té problemes per escoltar?	No, però depenent de que se li demani, a vegades és molt millor especificar-ho més.	A vegades et deixa sense resposta. Encara que la persona vocalitzi i sigui el més específic possible.	Si, però depenen de com li preguntis o formulis la frase i l'acció que li demanis.
T'indica el temps si el demanes?	Si l'usuari diu en la App de Amazon on ha registrat també el producte en la aplicació del mòbil es pot configurar la direcció de l'usuari i pot indicar-te el	Et dona el temps perfectament.	Et diu el temps amb els graus que fan i si plou o fa un dia solejat.

	temps directament, sinó et dirà que pots fer per tenir-ho directament i a més a més, et preguntarà d'on vols la informació del temps.		
Obre les llums o l'electrodomèstic endolla't a un interruptor especial?	Sempre que configuris des de la app del telèfon. Pots demanar a Alexa que actuï en els electrodomèstics de la casa com les llums o la televisió.	Té l'opció d'introduir llums o electrodomèstics a partir de l'aplicació de google home.	No entén el que li estic demanant. "I'm sorry. I don't understand." És a dir, encara no té l'opció de poder tenir el control de llums o electrodomèstics de casa.
Si demanes un canal de radio, te'l posa?	Et posa la emissora que tu li diguis, encara que depenen de la emissora has d'especificar-la una mica més. Per exemple, la emissora de "Los 40", has de dir la comunitat.	"Perdona, todavía no se como ayudarte."	Per poder escoltar la ràdio s'ha de tenir constància d'instal·lar les <i>skills</i> que són per poder escoltar-la. Per exemple, s'ha d'instal·lar la <i>skill</i> de Spotify per poder escoltar música.
Quines poden ser les seves limitacions?	A vegades, has de ser una mica específic quan li demanes informació sobre algun tema ja que depèn el que demanis ho pot agafar com a resposta molt genèrica o et respongui una resposta molt diferent el que has demanat.	Que no té l'abast de poder respondre tot el que se li demana.	Depenen de com l'expressis les coses es pot confondre i llavors et diu que ni t'entén o que ho expressis d'una altra forma.

<p>Com correspon a les <i>Skill</i> que ja són creades o creades per el mateix usuari?</p>	<p>Perfectament, encara que prèviament s'ha de configurar la <i>skill</i> a partir de la app de Alexa en el telèfon mòbil.</p>	<p>S'obre correctament, però després et comença a contestar amb anglès i quan no t'entén pot arribar a parlar en castellà, o donar-te les instruccions de com funciona l'<i>action</i>.</p>	<p>El fer la creació de la <i>skill</i> creada, en el moment que invoques la <i>skill</i> respon directament a ella sense cap problema. A més a més de les habilitats que ja estan preinstal·lades o que l'usuari pot instal·lar-les amb les respostes corresponents de cada una.</p>
<p>Una activació fàcil, és a dir, quan es crida amb la seva paraula clau. Li costa contesta o respon al moment?</p>	<p>En el moment que s'inicia amb "Alexa" et comença a escoltar i et respon sense cap problema. Encara que si triges o comences a parlar molt lentament pot ser que et deixi d'escoltar i es posi en mode repòs.</p>	<p>En el cas de Google, al moment de fer la crida, "Ok Google", es contesta correctament, encara que hi ha excepcions, si es parla molt ràpid i no es vocalitza, o es parla amb ell amb paraules no adequades.</p>	<p>En el moment que se li dona una acció a Mycroft necessita processar per poder respondre amb la resposta adequada. Però intentar processar-ho el més ràpid possible.</p>
<p>Si l'usuari parla amb el dispositiu amb paraules malsonants, per exemple. Li insulta, com reacciona</p>	<p>Es desconnecta i es sent un soroll per que tu puguis saber que s'ha desactivat. Com si tornés a un estat de repòs a la espera d'una nova crida.</p>	<p>Google el que fa és dir-te que si tens algun problema que et queixis a la web de google.</p>	<p>"I'm sorry, i don't understand".</p>

Taula 1 Comportament dels dispositius a situacions quotidianes.

Una de les parts més importants de totes és la privacitat que deixen cada un dels disponibles, ja que en molts casos els dispositius van captant informació dels usuaris, ja que van gravant totes les converses que hi ha al costat del dispositiu.

Fent una cerca per poder veure opinions de diferents articles periodístics es pot comprovar, que per exemple en Google, poden saber qui som, on hem estat, qui son els teus amics, plans de futur, és a dir, tot el nostre dia a dia, si tens un viatge programat, etc.

En canvi, Mycroft, dona la seguretat de que no guarda cap informació de totes les tasques que l'usuari realitza. Tampoc ho guarda de forma anònima. Mycroft només es processa quan li parles i no es guarda cap informació quan l'usuari té una conversa amb ell.

Una part important de la comunicació amb els diferents dispositius és que s'ha de vocalitzar bé sinó poden confondre el que els hi demanes i llavors o et responen amb "No te entiendo, lo siento" o una resposta similar i tenir una resposta molt diferent al que es demana o directament no fer-te cas.

Tant Alexa com Mycroft tenen suport de les webs oficials de cada un dels dispositius i així van realitzant actualitzacions cada vegada que canviant de mètodes o passos a seguir. En canvi, Google fa un any aproximadament es va deixar de donar suport a *google assistants* i als petits desenvolupadors per a que poguessin fer un dispositiu en un *Hardware* que no fos el que venen ells. No obstant, els tres dispositius tenen el suport de tenir diferents idiomes per a que els usuaris puguin comunicar-se amb ell. Encara que Mycroft té menys idiomes i alguns estaven en fase *Beta* com es l'espanyol.

3.3. Descripció del funcionament

Per a cada un dels dispositius tenen un inici bastant semblant per poder iniciar-lo. Primer de tot, s'ha d'introduir la targeta SD a la Raspberry Pi. A més a més de tenir connectats els altaveus, el micròfon, el teclat i el ratolí si és necessari. Després depenent de quin dispositiu volem iniciar tenen un diferent mètode.

En Alexa, quan iniciem la Raspberry Pi, entrem al terminal i llavors s'ha de entrar a la carpeta `sdk-build`. Per fer-ho s'ha d'utilitzar la comanda .

```
cd /home/pi/sdk-folder/sdk-build PA_ALSA_PLUGHW=1 ./SampleApp/src/SampleApp
./Integration/AlexaClientSDKConfig.json ../third-party/alexa-rpi/models
```

Una vegada hem introduït les comandes al terminal sortirà per pantalla el procés de com es connecta i surt per pantalla.



Figura 29 Inicialització de Alexa.

La imatge superior ens indica que Alexa ja està operativa i es pot parlar amb ella, a més a més et donen diferents opcions per poder fer com parlar amb ella sense haver de dir-li a Alexa al principi, configuracions d'ella com l'idioma, si necessites ajuda, si es necessita informació,...

A partir d'allà, funciona com un dispositiu corrent on l'usuari pot establir una conversa amb ella. Demanar el temps que fa, voler saber informació d'un artista o d'una ciutat, voler jugar a jocs que prèviament has d'haver activat amb l'aplicació que proporcionen i com si es vol parlar amb ella com si fos una persona.

Per poder activar a Alexa es necessita dir primer Alexa i després el que l'usuari necessiti sinó no farà cas.

Per altra banda, amb Google Home, al obrir la Raspberry Pi directament s'ha d'habilitar la consola i encendre-la.

```
sudo systemctl enable assistant.service
```

```
sudo systemctl start assistant.service
```

Un cop encès google, ja es pot comunicar-se amb ell, és a dir, per poder activar-lo s'haurà de dir "Ok Google" i després el que l'usuari demani. Normalment el primer cop que parla Google ho farà en anglès i després començarà en castellà, igualment, la primera vegada pot ser no et respon correctament però les següents sí.

```
pi@raspberrypi:~$ sudo systemctl enable assistant.service
pi@raspberrypi:~$ sudo systemctl start assistant.service
pi@raspberrypi:~$ sudo systemctl status assistant.service
● assistant.service - Google Assistant
   Loaded: loaded (/etc/systemd/system/assistant.service; enabled; vendor preset
   Active: active (running) since Fri 2020-06-12 19:20:00 CEST; 1 weeks 2 days a
   Main PID: 483 (bash)
     Tasks: 8 (limit: 2200)
    Memory: 60.5M
   CGroup: /system.slice/assistant.service
           └─483 /bin/bash /home/pi/start_assistant.sh
             └─489 /home/pi/env/bin/python /home/pi/env/bin/googlesamples-assistan

Jun 12 19:20:22 raspberrypi bash[483]: INFO:root:Finished playing assistant resp
Jun 12 19:20:22 raspberrypi bash[483]: INFO:root:Recording audio request.
Jun 21 22:09:11 raspberrypi bash[483]: INFO:root:End of audio request detected.
Jun 21 22:09:11 raspberrypi bash[483]: INFO:root:Stopping recording.
Jun 21 22:09:11 raspberrypi bash[483]: INFO:root:Finished playing assistant resp
Jun 21 22:09:11 raspberrypi bash[483]: INFO:root:Recording audio request.
Jun 21 22:09:20 raspberrypi bash[483]: INFO:root:End of audio request detected.
Jun 21 22:09:20 raspberrypi bash[483]: INFO:root:Stopping recording.
```

Figura 30 Inicialització de google

En el cas de Mycroft hi han dos opcions, una que es posi un tutorial d'ajudar per configurar els altaveus i micròfon, i just quan s'acaba la configuració entra directament a la màquina per poder interactuar amb ella, que directament entri a la maquina o a partir d'una comanda per cridar-la.

```
mycroft-cli-client
```

```

20:54:52.386 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-version-checker
20:54:53.555 | INFO | 792 | ConfigurationSkill | Remote configuration updated
20:54:53.703 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-ip
20:54:55.551 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-audio-record
20:54:56.849 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-naptime
20:54:58.096 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-speak
20:55:00.315 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-timer
20:55:01.973 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-support-helper
20:55:03.959 | INFO | 792 | msm.skill_entry | Nothing new for mycroft-volume
20:55:04.740 | INFO | 792 | mycroft.skills.skill_updater:log_next_download_time:248 | Next scheduled skill update: 2020-06-21 21:55:04.738390
20:55:04.743 | INFO | 792 | mycroft.skills.skill_updater:update_skills:163 | Skill update complete
^--- NEWEST ---^
History ===== Log Output Legend ===== Mic Level ==
DEBUG output
skills.log, other
voice.log
--- 1.50
1
Input (':' for command, Ctrl+C to quit) =====
>

```

Figura 31 Inicialització de Mycroft.

A partir d'allà, tots els dispositius estan operatius des de que comença una conversa amb el dispositiu corresponent. Ja es pot demanar qualsevol cosa per poder demanar-li.

Tots els dispositius tenen l'opció d'escollir entre diversos idiomes encara que Google és el que té més opcions i Mycroft és el que menys i algun d'ells estan en proves encara. En la part de Alexa es pot fer directament quan s'obre l'aplicatiu hi ha diferents opcions de configuracions i en una d'elles es pot escollir idiomes i totes les configuracions que hi ha. En canvi per fer el canvi d'idiomes de Google s'ha de fer a partir de la seva web de *Action de Google*.²⁰ Per Mycroft es canvia a partir del terminal de la màquina.

```
mycroft-config set lang "it-it"
```

El "it-it", seria en italià però es poden escollir entre diferents idiomes com l'espanyol, l'alemany, l'italià, portuguès entre d'altres.

Tots els dispositius tenen una funcionalitat diferents entre elles, llavors cadascuna té diferències per activar-los, cridar-los o com demanar-li el que l'usuari desitgi.

3.4. Aplicacions de l'estudi

Aquest estudi pot ser útil en el món *maker*, a les petites empreses, als desenvolupadors independents o els *hobbyists*, per poder preveure quin seria l'assistent més apropiat i pugui escollir l'opció més adequada per a projectes, per exemple, en els següents camps:

- Altaveus intel·ligents.
- Electrodomèstics.
- Automoció.
- Domòtica.
- Juguines.
- Telefonia mòbil.
- *Smartwatches*.

²⁰ «Actions on Google». <https://console.actions.google.com/u/0/> (accedit juny 14, 2020)

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
Configuració Raspberry Pi	■	■																		
Instal·lació Alexa			■	■																
Instal·lació Google Assistant				■	■	■														
Instal·lació Mycroft						■	■	■												
Disseny general d'un skill de prova					■	■	■	■												
Adaptació a Alexa, Google Assistant i Mycroft								■	■	■	■	■	■							
Proves i valoració										■	■	■	■	■	■	■				
Deteccions de millores													■	■	■	■	■	■		
Redacció de l'estudi										■	■	■	■	■	■	■	■	■	■	■

Taula 2 Diagrama de Gantt.

La distribució de feina que s'ha realitzat pensant en distribuir la càrrega de treball de manera efectiva. La feina ha estat dividida a partir de diferents blocs temàtics.

4.2. Llista de materials

Per realitzar aquest projecte s'han utilitzat diferents materials:

- Raspberry Pi 3b, amb 1GB RAM. Un processador Broadcom BCM2837BO, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.
- Diferents cables:
 - Cable de corrent. Que entra amb connexió USB.
 - Cable HDMI-HDMI. Per poder endollar a una pantalla.
 - Cable Ethernet. Per tenir xarxa Wifi al dispositiu.
- Altaveus. Genèrics amb cable jack de 3.5mm amb cable de corrent.
- Micròfon. Marca Logitech. Amb cable USB.
- Targetes SD. Mínim de 8 GB o superior.
- Pantalla. Samsung que tingui connexió amb HDMI.
- Teclat. Que es connecti per cable USB.
- Ratolí. Marca Logitech amb cable USB.

4.3. Pressupost

Per poder fer aquest treball s'han hagut de fer 250 hores de treball per poder realitzar l'estudi que s'ha fet, a més a més del material utilitzat. S'ha tingut un acord on l'usuari que ha realitzat aquesta cerca tindrà un sou de 8 euros a l'hora.

Per una banda, hauríem de tenir en compte tot el material que s'ha hagut d'utilitzar per poder realitzar aquest estudi.

Material	Unitats	Preu €	Preu total
Raspberry Pi 3b	1	40,23	40,23
Altaveus	1	14,99	14,99
Cable de corrent de Raspberry Pi	1	9,99	9,99
Regleta	1	9,99	9,99
Cable HDMI-HDMI	1	8,99	8,99
Cable Ethernet	1	8,99	8,99
Micròfon	1	8,97	8,97
Targetes SD	3	9,79	29,37
Pantalla	1	99,99	99,99
Teclat	1	9,98	9,98
Ratolí	1	4,99	4,99
Ordinador portàtil	1	500	500
Total			746,48

Taula 3 Pressupost del material utilitzat en el projecte.

Per altra banda, també s'ha de tenir en compte, a més a més, del material, el sou que s'ha ingressat al treballador per l'estudi.

Costos	€/hora	Hores	€
Material			746,48
Sou	8	250	2000
			2746,48

Taula 4 Pressupost total.

Finalment, per poder realitzar aquest estudi s'hauria de fer una inversió inicial de 2746,48 €.

4.4. Impacte mediambiental, aspectes ètics, ...

No es té constància de cap dany inusual al medi ambient. Tampoc hi ha hagut constància d'activitats èticament qüestionables (aplicacions militars, experimentació amb animals, ...). El treball ha tingut un impacte mediambiental mínim perquè des de casa s'ha pogut realitzar gairebé tota la feina sense problemes.

4.5. Fites aconseguides

Les fites que s'han pogut assolir en aquest estudi són:

- Instal·lació de prototips de diferents assistents de veu en una placa de desenvolupament de Raspberry pi.
- Configuració de cadascun dels assistents de veu.
- Creació i implementació de *skills* en cadascun dels assistents de veu.
- Assoliment d'una funcionalitat correcta de cada dispositiu.
- Proves intensives del comportament en qualsevol situació de cada assistent de veu.

4.6. Conclusions

Tots els assistents de veu tenen l'aparença de ser oberts però no ho són, en el sentit de que es poden instal·lar en diferents dispositius de desenvolupament, com per exemple, la Raspberry Pi però a la pràctica sempre hi ha problemes.

- I. L'entorn SDK canvia sovint, perquè els diferents assistents es van actualitzant cada cert temps.
- II. Hi ha diferents controls, amb codis de informació, on indicar les identificacions que s'estan creant.
- III. És necessari un compte de desenvolupament gestionat per *webapp*. El funcionament també canvia amb freqüència.

Tota la informació que es pot trobar a internet es bastant dispersa i moltes vegades està desfasada. Encara que, es compensa per la documentació *online* oficial que en general és de bastant qualitat.

Amazon i Google són empreses grans que treballen molt al núvol. Amazon amb AWS i Google amb Google Cloud Platform. En canvi, Mycroft és tot el contrari, totes les *skills* s'instal·len en el mateix dispositiu. Totes les *skills* de Mycroft es troben a una mateixa carpeta en el mateix dispositiu, i així es té més control del codi i no estan tots els programes al núvol.

La programació en Mycroft és molt més amigable pels desenvolupadors per tenir una interfície "pitònica", és a dir, es programa a partir de Python i fent un ampli ús de llibreries i classes de manera que amb poques línies es pot arribar a aconseguir una *skill*. En una posició intermèdia es pot trobar a Amazon, ja que està molt més orientada a Javascript, on s'han de cridar moltes funcions per poder executar una *skill* senzilla. Per últim, tenim a Google que si no disposen d'un compte bancari per carregar les eventuais despeses no es pot programar i només es poden realitzar *actions* molt bàsiques.

El cost econòmic és molt variat entre les empreses, és a dir, a Google si no dones les teves dades bancàries no pots programar *actions* a Google Cloud Platform, Amazon és molt més tolerant mentre sigui un desenvolupament, però que la web d'experts AWS té un cost anual per utilitzar-la. D'altra banda, Mycroft, té un cost de Hardware bastant elevat i a més a més, té un cost mensual si és necessari el suport d'ells.

Un dels punts més importants és la funcionalitat dels diferents assistents de veu. La millor comprensió de les comandes és la d'Alexa, que entèn gairebé totes les comandes mentre que Google i Mycroft moltes vegades no arriben entendre el que es demana. S'ha de tenir en compte que si no es vocalitza cap dels assistents arribarà entendre la crida que s'ha realitzat. En canvi, Google es qui té més idiomes de compatibilitat per utilitzar-lo. Per defecte, Alexa i Google tenen les mateixes *skills* genèriques. En el moment que responen els assistents de veu Alexa i Google s'entenen correctament, en canvi, Mycroft, moltes vegades, es menja el principi de les seves respostes.

La privacitat dels assistents de veu es bastant nul·la pel fet que de bon començament ja et demanen informació personal sobre l'usuari i sempre estan escoltant encara que en estat de repòs només responen si els crides per la seva paraula clau. Pel contrari, Mycroft és diferent, només es processa quan li parles i no es guarda cap informació quan l'usuari té una conversa amb ell.

Per mi el millor de tots els assistents que s'ha estudiat seria una barreja entre Alexa i Mycroft ja que la comunicació és molt bona i en canvi, per Mycroft realitzar tota la part en el mateix dispositiu sense necessitat de *webs app* ni el núvol fa que tinguis un major control del teu dispositiu sense tenir en compte tercers.

4.7. Millores futures

Aquest treball es pot continuar de moltes i molt interessant maneres, com per exemple:

- Funcionament offline. Està per veure fins a quin punt podem tenir un assistent fora de la xarxa. Aquest estudi seria interessant, per exemple, per a l'àmbit de l'automoció, on un automòbil pot circular per una carretera sense cobertura. En teoria, Mycroft podria funcionar d'aquesta manera però cal fer una configuració bastant complexa per poder assolir-ho. Una bona continuació seria estudiar-ho.
- L'aprofundització de tota la part de programació. En el meu cas s'ha realitzat una skill molt bàsica. S'hauria de fer un cert ventall de skills de proves i després programar-ho en diferents assistents de veu per poder tenir una comparativa de programacions.
- Realització de proves per poder saber fins a quin punt estan interconnectats els diferents perfil de l'usuari. Per exemple, fer una cerca des de Alexa i després anar a la web de Amazon per poder saber, si a partir de la cerca que s'ha realitzat pot ser que ens influèncii en la web per comprar o no. Un altre cas seria amb Google i els anuncis que poden aparèixer després de fer diferents crides a Google Assistant.
- Fer un estudi molt més complet, incloent més assistents de veu com Siri, Cortana, Bixby,...

5. Bibliografia Addicional

- [1] «3. Asistentes virtuales de voz | Marketing XXI». <https://www.marketing-xxi.com/voice-search-asistentes-voz-altavoces-inteligentes-seo-sem/asistentes-virtuales-voz> (accedit juny 19, 2020).
- [2] «Actions on Google». <https://console.actions.google.com/u/0/?pli=1> (accedit juny 19, 2020).
- [3] «Actions on Google». <https://console.actions.google.com/u/0/> (accedit juny 19, 2020).
- [4] «Actions on Google». <https://console.actions.google.com/u/0/?pli=1> (accedit juny 19, 2020).
- [5] «Alexa». <https://developer.amazon.com/es-ES/docs/alexa/alexa-voice-service/activate-security-profile.html> (accedit juny 19, 2020).
- [6] «Alexa Voice Service». <https://developer.amazon.com/alexa/console/avs/home?> (accedit juny 28, 2020).
- [7] «Alexa Voice Service (AVS) SDK Overview | AVS Device SDK». <https://developer.amazon.com/es-ES/docs/alexa/avs-device-sdk/overview.html> (accedit juny 19, 2020).
- [8] «Asistente de voz, una gran revolución tecnológica». https://www.elespanol.com/imprescindibles/20191231/asistente-voz-gran-revolucion-tecnologica/452705612_0.html (accedit juny 19, 2020).
- [9] I. Aleksandra, «Asistente de voz, una gran revolución tecnológica», *31 diciembre*, 2019.
- [10] «balenaEtcher - Flash OS images to SD cards & USB drives». <https://www.balena.io/etcher/> (accedit juny 19, 2020).
- [11] «Build Skills with the Alexa Skills Kit | Alexa Skills Kit». <https://developer.amazon.com/es-ES/docs/alexa/ask-overviews/build-skills-with-the-alexa-skills-kit.html> (accedit juny 19, 2020).
- [12] «Cheatsheet | Font Awesome». <https://fontawesome.com/cheatsheet> (accedit juny 19, 2020).
- [13] «colors - Mycroft». <https://mycroft.ai/colors> (accedit juny 19, 2020).
- [14] «Controls d'activitat». <https://myaccount.google.com/activitycontrols> (accedit juny 19, 2020).
- [15] «Create a Trivia Action using templates | Actions on Google Templates». <https://developers.google.com/assistant/templates/trivia> (accedit juny 19, 2020).
- [16] «Development Setup - Mycroft AI». <https://mycroft-ai.gitbook.io/docs/skill-development/introduction> (accedit juny 19, 2020).

- [17] «Download Raspberry Pi OS for Raspberry Pi». <https://www.raspberrypi.org/downloads/raspberry-pi-os/> (accedit juny 19, 2020).
- [18] «Finalize Your Alexa Skill - Beginner Alexa Developer Tutorial». <https://developer.amazon.com/es-ES/alexa/alexa-skills-kit/get-deeper/tutorials-code-samples/build-an-engaging-alexa-skill/module-7> (accedit juny 19, 2020).
- [19] «Font Awesome 6 Pre-order». <https://fontawesome.com/6?next=%2Fcheatsheet> (accedit juny 19, 2020).
- [20] «Get Started - Mycroft». <https://mycroft.ai/get-started#setup>, (accedit juny 19, 2020).
- [21] «Get Started with the Alexa Skills Kit - Alexa Skills Kit Official Site». <https://developer.amazon.com/en-US/alexa/alexa-skills-kit/start> (accedit juny 19, 2020).
- [22] «Glossary of terms - Mycroft AI». <https://mycroft-ai.gitbook.io/docs/about-mycroft-ai/glossary> (accedit juny 19, 2020).
- [23] «Google Cloud Platform». <https://console.cloud.google.com/home/dashboard?project=holaquetal-37d26&pli=1> (accedit juny 19, 2020).
- [24] «google-assistant-demo and googlesamples-assistant-hotword segmentation fault - Raspberry Pi Forums». <https://www.raspberrypi.org/forums/viewtopic.php?t=249368> (accedit juny 19, 2020).
- [25] «Hola Google! vamos a aprender Actions on Google (Parte 1)». <https://medium.com/@leslysandra/hola-google-vamos-a-aprender-actions-on-google-parte-1-bdf13b988a75> (accedit juny 20, 2020).
- [26] «Hola Google! vamos a aprender Actions on Google (Parte 2)». <https://medium.com/@leslysandra/hola-google-vamos-a-aprender-actions-on-google-parte-2-660d008416b> (accedit juny 20, 2020).
- [27] «Host a Custom Skill as an AWS Lambda Function | Alexa Skills Kit». <https://developer.amazon.com/es-ES/docs/alexa/custom-skills/host-a-custom-skill-as-an-aws-lambda-function.html> (accedit juny 19, 2020).
- [28] «How to add Alexa's custom skills to Alexa Raspberry Pi? - Stack Overflow». <https://stackoverflow.com/questions/57773861/how-to-add-alexa-custom-skills-to-alexa-raspberry-pi> (accedit juny 20, 2020).
- [29] «Introducción a Actions on Google - Adictos al trabajo». <https://www.adictosaltrabajo.com/2019/04/26/introduccion-a-actions-on-google/> (accedit juny 25, 2020).
- [30] «Introducción a Actions on Google - Adictos al trabajo». <https://www.adictosaltrabajo.com/2019/04/26/introduccion-a-actions-on-google/> (accedit juny 19, 2020).

- [31] «La evolución de los Asistentes Virtuales en 8 años». <https://blogthinkbig.com/asistentes-virtuales-usuarios> (accedit juny 19, 2020).
- [32] «Llega la nueva Raspberry Pi 3 Model B+: mismo precio, pero más velocidad y WiFi de doble banda». <https://www.xataka.com/ordenadores/llega-la-nueva-raspberry-pi-3-model-b-mismo-precio-pero-mas-velocidad-y-wifi-de-doble-banda> (accedit juny 19, 2020).
- [33] «Los altavoces y asistentes de voz están revolucionando el modo de comprar». <https://tentulogo.com/los-altavoces-y-asistentes-de-voz-estan-revolucionando-el-modo-de-comprar/> (accedit juny 19, 2020).
- [34] «Mattermost». https://chat.mycroft.ai/login?redirect_to=%2Fcommunity%2Fchannels%2Fskills (accedit juny 19, 2020).
- [35] «Mycroft AI Single Sign On». <https://sso.mycroft.ai/login?redirect=https:%2F%2Fhome.mycroft.ai%2F> (accedit juny 28, 2020).
- [36] «Mycroft AI Single Sign On». <https://sso.mycroft.ai/login?redirect=https:%2F%2Fhome.mycroft.ai%2F> (accedit juny 19, 2020).
- [37] «Mycroft Skills Kit - Mycroft AI». <https://mycroft-ai.gitbook.io/docs/mycroft-technologies/mycroft-skills-kit> (accedit juny 19, 2020).
- [38] «Mycroft, el asistente virtual con inteligencia artificial Open Source - Domótica en Casa». <https://domoticaencasa.es/mycroft-asistente-virtual-inteligencia-artificial-open-source/> (accedit juny 20, 2020).
- [39] «Mycroft, el asistente virtual con inteligencia artificial Open Source - Domótica en Casa». <https://domoticaencasa.es/mycroft-asistente-virtual-inteligencia-artificial-open-source/> (accedit juny 19, 2020).
- [40] YÚBAL FM, «No TitleLlega la nueva Raspberry Pi 3 Model B+: mismo precio pero más velocidad y WiFi de doble banda», 2018. <https://www.xataka.com/ordenadores/llega-la-nueva-raspberry-pi-3-model-b-mismo-precio-pero-mas-velocidad-y-wifi-de-doble-banda>.
- [41] «Picroft - Mycroft AI». <https://mycroft-ai.gitbook.io/docs/using-mycroft-ai/get-mycroft/picroft#tested-hardware> (accedit juny 19, 2020).
- [42] «Privacidad en asistentes de voz, ¿un freno para su desarrollo? | Kanlli». <https://www.kanlli.com/estrategia-marketing-digital/privacidad-en-asistentes-de-voz/> (accedit juny 19, 2020).
- [43] «Qué es Alexa, qué puedes hacer con él y qué dispositivos son compatibles». <https://www.xataka.com/basics/que-alexa-que-puedes-hacer-que-dispositivos-compatibles> (accedit juny 28, 2020).
- [44] «Raspberry Pi Downloads - Software for the Raspberry Pi». <https://www.raspberrypi.org/downloads/> (accedit juny 19, 2020).

- [45] «Register a Product | Alexa Voice Service». <https://developer.amazon.com/es-ES/docs/alexa/alexa-voice-service/register-a-product.html#create-a-security-profile> (accedit juny 28, 2020).
- [46] «Register a Product | Alexa Voice Service». <https://developer.amazon.com/es-ES/docs/alexa/alexa-voice-service/register-a-product.html> (accedit juny 19, 2020).
- [47] «Set Up the AVS Device SDK on Raspberry Pi | AVS Device SDK». <https://developer.amazon.com/es-ES/docs/alexa/avs-device-sdk/raspberry-pi.html> (accedit juny 19, 2020).
- [48] «Sign in to GitHub · GitHub». https://github.com/login?return_to=https%3A%2F%2Fgithub.com%2Fsettings%2Ftokens (accedit juny 19, 2020).
- [49] «Skills de Alexa: qué son, cómo instalarlos y cuáles son los mejores». <https://www.xataka.com/basics/skills-alexa-que-como-instalarlos-cuales-mejores> (accedit juny 25, 2020).
- [50] «The apple music on the Amazon echo speakers starting from December 17 – Moviru». <https://moviru.com/2018/12/17/the-apple-music-on-the-amazon-echo-speakers-starting-from-december-17/> (accedit juny 25, 2020).
- [51] «The apple music on the Amazon echo speakers starting from December 17 – Moviru». <https://moviru.com/2018/12/17/the-apple-music-on-the-amazon-echo-speakers-starting-from-december-17/> (accedit juny 19, 2020).
- [52] «The Best Voice Assistants of 2020 | Reviews.com». <https://www.reviews.com/home/smart-home/best-voice-assistant/> (accedit juny 19, 2020).
- [53] «Understand Custom Skills | Alexa Skills Kit». <https://developer.amazon.com/es-ES/docs/alexa/custom-skills/understanding-custom-skills.html> (accedit juny 19, 2020).
- [54] «Voice Application Development Guidelines – Plasma Bigscreen IMG». <https://plasma-bigscreen.org/development-guidelines/#getting-started> (accedit juny 19, 2020).
- [55] «Web developers de Amazon». <https://developer.amazon.com/alexa/>.
- [56] «Web Importer | Mendeley». <https://www.mendeley.com/reference-management/web-importer/> (accedit juny 19, 2020).
- [57] «Your First Skill - Mycroft AI». <https://mycroft-ai.gitbook.io/docs/skill-development/introduction/your-first-skill> (accedit juny 19, 2020).
- [58] «Your own Raspberry Pi Google Assistant by Janne Spijkervet -». <https://hakin9.org/your-own-raspberry-pi-google-assistant/> (accedit juny 19, 2020).