



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE GRAU

TÍTOL: ASSISTENT PERSONAL PER L'EPSEVG BASAT EN ALEXA

AUTORS: MARTÍNEZ IZQUIERDO, MIRIAM

DATA DE PRESENTACIÓ: GENER, 2021

COGNOMS: MARTÍNEZ IZQUIERDO

NOM: MIRIAM

TITULACIÓ: ENGINYERIA INFORMÀTICA

PLA: 2016

DIRECTOR-CODIRECTOR: VIDAL FERRÉ, RAFAEL
GUASCH MURILLO, DANIEL

DEPARTAMENT: 744 - ENGINYERIA TELEMÀTICA

QUALIFICACIÓ DEL TFG

TRIBUNAL

PRESIDENT

SECRETARI

VOCAL

DATA DE LECTURA:

Aquest Projecte té en compte aspectes mediambientals: Sí No

RESUM

L'ús dels assistents per veu s'ha tornat una rutina diària per moltes persones. Aquests assistents permeten una interacció per veu amb els diferents dispositius que el tenen integrat. Per tant, una interacció més similar a la humana, fet que fa que el seu ús sigui més còmode i fàcil. Aquests assistents són capaços de realitzar moltes tasques, com per exemple, consultar i retornar la informació demanada per l'usuari.

Donada aquesta situació, en aquest treball es planteja el desenvolupament d'una skill de l'assistent per veu Alexa, de la companyia Amazon. Una skill és una funcionalitat extra per l'assistent, equivalent a una aplicació per telèfons mòbils. La skill desenvolupada permet preguntar a l'assistent per informació relacionada amb l'activitat acadèmica de l'EPSEVG. Com per exemple, l'horari de classe d'una assignatura o les lectures de TFG previstes en les que participa un professor en el dia actual, entre d'altres. En aquesta memòria es mostren les diferents fases realitzades per al correcte desenvolupament d'aquesta skill. Aquestes fases són l'estudi de l'arquitectura de la plataforma Amazon que fa possible la creació de skills, el seu disseny que contempla tan el seu funcionament com la definició dels diferents casos d'ús, l'estudi de les eines a utilitzar, el seu desenvolupament, la seva validació i la posterior publicació i manteniment.

En conclusió, el resultat aconseguit és una skill que fa possible la interacció amb l'usuari de manera senzilla i proporciona diferents dades públiques sobre l'EPSEVG. Alhora s'aconsegueix una documentació detallada que aprofundeix en el funcionament d'Alexa i de les skills, que ha de permetre en el futur, la seva publicació i manteniment.

Paraules clau (màxim 10):

Assistent per veu	Alexa	Amazon Web Services	Reconeixement per veu
Intel·ligència artificial	Skill d'Alexa	Interacció per veu	Altaveu intel·ligent

ABSTRACT

The use of voice assistants has become a daily routine for many people. These assistants allow for speech interaction with different devices that have integrated it. Therefore, more similar to human interaction, making its use more comfortable and easier. These assistants are able to perform many tasks such as, querying and returning the information requested by the user.

In view of this situation, this work raises the development of a skill of Alexa, the Amazon's voice assistant. A skill is an extra functionality for the assistant, equivalent to an application for mobile phones. The developed skill allows you to ask the assistant for academic information about EPSEVG. For example, the class schedule of a subject or the TFG's readings in which a teacher participates on the current day, among others. This work's memory shows the different phases performed for the correct development of this skill. These phases are the study of the architecture of the Amazon platform that makes it possible to create skills, its design of operation and the definition of the different cases of use, the study of tools to use, its development, validation, publication, and maintenance.

In conclusion, the result is a skill that makes speech interaction with the user possible in a simple way and provides different public data about the EPSEVG. At the same time, detailed documentation is obtained that deepens the operation of Alexa and the skills which must allow its publication and maintenance in the future.

Keywords (10 maximum):

Voice assistant	Alexa	Amazon Web Services	Speech recognition
Artificial intelligence	Alexa's skill	Speech interaction	Smart speaker

SUMARI

1. INTRODUCCIÓ	14
1.1. Objectius del treball	14
2. ESTAT DE L'ART	14
2.1. Assistents per veu.....	14
2.1.1. Cronologia	15
2.2. Altaveus intel·ligents	17
3. ALEXA : LA INTEL·LIGÈNCIA ARTIFICIAL D'AMAZON	17
3.1. Plataforma Alexa.....	18
3.2. Dispositius.....	18
3.3. Servei de veu Alexa	18
3.4. Skills.....	20
3.4.1. Tipus.....	21
3.4.2. Model d'interacció	24
4. DISSENY DE LA SKILL	25
4.1. Funcionalitat.....	25
4.2. Casos d'ús	25
4.2.1. Consulta informació professors	26
4.2.2. Consulta informació horari de consulta dels professors	27
4.2.3. Consulta informació departaments	27
4.2.4. Consulta informació aules.....	29
4.2.5. Consulta informació setmana actual.....	29
4.2.6. Consulta informació horari assignatures actuals	29
4.2.7. Consulta informació exàmens.....	30
4.2.8. Consulta informació lectures TFG	32
5. EINES UTILITZADES PER A LA IMPLEMENTACIÓ	34
5.1. Front end.....	34
5.1.1 Alexa Skills Kit	34
5.2. Back end	34
5.2.1. AWS Lambda.....	34
5.2.1.1. Node.js	35

5.2.1.2. Amazon CloudWatch.....	36
5.2.2. MySQL.....	36
5.2.2.1 HeidiSQL.....	36
6. IMPLEMENTACIÓ DE LA SKILL.....	37
6.1. Arquitectura de la skill.....	37
6.2. Funcionament.....	38
6.3. Creació de la skill.....	39
6.4. Front end. Creació del model d'interacció.....	41
6.4.1. Nom d'invocació.....	41
6.4.2. Intents.....	41
6.4.3. Sample utterances.....	44
6.4.4. Slots.....	45
6.4.5. Guardar i construir el model d'interacció.....	49
6.5. Back end.....	49
6.5.1. Base de dades MySQL.....	49
6.5.2. Funció Lambda.....	49
6.5.2.1. Dependències.....	50
6.5.2.2. Connexió amb la skill <i>Asistente EPSEVG</i>	51
6.5.2.1. Controladors d'intents.....	52
6.5.2.3. Controlador de l'intent Lectures TFG.....	53
7. VALIDACIÓ DE LA SKILL.....	57
CONCLUSIONS.....	60
AGRAÏMENTS.....	62
REFERÈNCIES.....	63
BIBLIOGRAFIA.....	65
ANNEX.....	66
A. Tractament de les dades per una correcta síntesi de veu d'Alexa.....	66
A.1. Número de telèfon.....	66
A.2. Correu electrònic.....	66
A.3. Codi de les aules.....	67
A.4. Dies de la setmana i mesos.....	67

A.5. Hores amb sistema horari de 24h	69
B. Manual d'usuari	70
B.1. Informació sobre professors	70
B.2. Informació sobre assignatures	71
B.3. Informació sobre calendari	72
B.4. Informació sobre la localització d'aules	72
B.5. Informació sobre les lectures de TFG actuals	73
B.6. Informació sobre departaments	73
B.7. Peticions genèriques	74
C. Manual per l'actualització de dades	74
C.1. Actualitzacions en qualsevol període de curs	74
C.1.1. Actualització professors	74
C.1.2. Actualització departaments	76
C.2. Actualitzacions per canvi de quadrimestre	76
C.2.1. Actualització assignatures	76
C.2.2. Actualització grups	77
C.2.3. Actualització estudiants	77

SUMARI DE FIGURES

Figura 1. IBM Shoebox.....	15
Figura 2. Cronograma de l'evolució dels assistents per veu l'última dècada.....	16
Figura 3. Vendes d'altaveus intel·ligents 2018 Q3 – 2020 Q1 a nivell global (en milions).....	17
Figura 4. Arquitectura AVS amb les diferents parts d'una skill.....	20
Figura 5. Diagrama d'interacció de l'usuari amb una custom skill.....	23
Figura 6. Exemple de model d'interacció d'una custom skill fictícia.....	25
Figura 7. Especificació del cas d'ús informació professors.....	26
Figura 8. Especificació del cas d'ús horari consultes.....	27
Figura 9. Especificació del cas d'ús informació departaments.....	28
Figura 10. Especificació del cas d'ús horari assignatures.....	29
Figura 11. Especificació del cas d'ús informació departaments.....	31
Figura 12. Especificació del cas d'ús sobre lectures TFG.....	32
Figura 13. Programació síncrona vs programació asíncrona.....	35
Figura 14. Arquitectura de la skill Asistente EPSEVG.....	37
Figura 15. Funcionament de la skill Asistente EPSEVG.....	38
Figura 16. Pàgina principal d'Alexa Developer Console, creació skill.....	39
Figura 17. Estat inicial dels intents del model d'interacció amb la plantilla Start from Scratch.....	40
Figura 18. Nom d'invocació per la skill Asistente EPSEVG.....	41
Figura 19. Creació d'un nou intent.....	42
Figura 20. Intents personalitzats creats a la skill Asistente EPSEVG.....	43
Figura 21. Sample utterances definits per l'intent LecturesTFG.....	44
Figura 22. Creació d'un tipus d'slot.....	45

Figura 23. Obtenció dels valors corresponents pel tipus d'slot xProfesor de la skill Asistente EPSEVG utilitzant HeidiSQL	47
Figura 24. Importació dels valors d'un tipus d'slot a través d'un arxiu CSV	47
Figura 25. Tipus d'slot xProfesor amb els valors importats correctament d'un arxiu CSV	48
Figura 26. Slots inclosos a l'intent Lectures TFG de la skill Asistente EPSEVG	48
Figura 27. Barra de tasques del model d'interacció	49
Figura 28. Contingut de l'arxiu package.json de la funció Lambda	50
Figura 29. Dashboard de la funció Lambda asistentEPSEVG	51
Figura 30. Configuració endpoint de la skill Asistente EPSEVG	52
Figura 31. Llistat dels controladors de tots els intents creats al model d'interacció de la skill	52
Figura 32. Implementació del controlador de l'intent Lectures TFG	53
Figura 33. Arxiu tipus JSON amb la descripció de la sol·licitud realitzada per l'usuari	55
Figura 34. Implementació de la funció que retorna informació demanada de la base de dades	56
Figura 35. Flux de veu entre l'usuari i la skill.....	57
Figura 36. Arxiu JSON generat pel servei Alexa on es descriu la petició realitzada per l'usuari.....	58
Figura 37. Arxiu JSON generat al back end que conté la resposta	59
Figura 38. Número de telèfon formatejat per una correcta síntesi de veu.....	66
Figura 39. Correu electrònic formatejat per una correcta síntesi de veu	67
Figura 40. Codi d'aula formatejat per una correcta síntesi de veu	67

Figura 41. Obtenció del dia de la setmana dels valors "hoy" o "mañana" del tipus d'slot xDia.....	68
Figura 42. Tractament del dia de la setmana retornat per la BD per una correcta síntesi de veu	68
Figura 43. Obtenció de la paraula corresponent al mes donat una data (tipus date)	69
Figura 44. Tractament del sistema horari 24h per una correcta síntesi d'Alexa	70
Figura 45. Valors del tipus d'slot xProfesor	75
Figura 46. Variables any del curs i quadrimestre	76

SUMARI DE TAULES

Taula 1. Serveis oferts pel servei de veu Alexa	19
Taula 2. Intents obligatoris creats a la skill Asistente EPSEVG	42

GLOSSARI

En aquesta memòria s'utilitzen paraules tècniques o abreviatures, a continuació es troben les seves definicions.

- **Assistents per veu:** és un agent de software que mitjançant reconeixement per veu, permet a les persones interactuar amb un ordinador, un smartphone o altres dispositius a través de comandes de veu.
- **Altaveus intel·ligents:** és un dispositiu intel·ligent sense fils amb un assistent per veu integrat. Incorpora un micròfon per a poder realitzar el reconeixement de la parla, un altaveu per a poder reproduir la veu sintetitzada i permet connexió via Wi-Fi per a fer possible la comunicació amb el cloud. D'aquesta manera és possible estendre les seves funcionalitats més enllà de la reproducció d'àudio.
- **Alexa:** és l'assistent per veu d'Amazon ubicat al núvol i disponible a tots els dispositius de la marca. Permet crear experiències de veu naturals i oferir als usuaris una forma més fàcil e intuïtiva d'interactuar amb la tecnologia.
- **Amazon Echo Dot:** és un altaveu intel·ligent fabricat per Amazon que té integrat l'assistent per veu Alexa.
- **Intel·ligència artificial (IA/AI):** és una part de la informàtica dedicada al desenvolupament d'algorismes que permeten a una màquina prendre decisions intel·ligents, comportant-se com si tingués una intel·ligència humana.
- **Skill d'Alexa:** és una funcionalitat extra que es pot afegir a qualsevol dispositiu amb el servei d'Alexa integrat. Es pot dir que és un equivalent a les aplicacions per mòbils.
- **AWS Lambda:** *Amazon Web Services (AWS) Lambda* és un servei informàtic que permet executar codi sense aprovisionar ni administrar servidors, només executa el codi quan és necessari i s'escala de manera automàtica.
- **Funció Lambda:** és el conjunt de configuracions i codi que s'executa a l'AWS Lambda.
- **Comprensió de la parla:** en anglès *Speech Language Udenstarding, SLU*. És la interpretació de signes transmesos per una senyal de veu.

Juntament amb la comprensió del llenguatge natural, NLU, el seu objectiu és obtenir una representació conceptual d'oracions en llenguatge natural.

- **Síntesi de veu:** és la producció artificial de parla humana, en anglès *text-to-speech*, en referència a la capacitat de convertir text en parla.
- **Model d'interacció:** per una skill, és l'estructura que determina les sol·licituds a les que pot donar resposta i controlar les paraules que els usuaris diuen per invocar aquestes sol·licituds.
- **Front end:** és la part d'una aplicació que interactua amb els usuaris.
- **Back end:** és la capa d'accés a dades que conté la lògica d'una aplicació. El conjunt d'accions que la formen com per exemple la comunicació amb un servidor, és transparent per l'usuari.
- **Invocation name:** és el nom que utilitza Alexa per a identificar una skill.
- **Intent:** és cadascuna de les peticions a les que la skill pot donar resposta.
- **Sample utterance:** és un conjunt de frases probables que l'usuari pot dir per a cridar un intent.
- **Slot:** és un argument dins d'un sample utterance, habitualment necessari per a poder identificar la informació que busca l'usuari.
- **Especificació:** és la definició formal o semi-formal d'un sistema.
- **UML:** és un llenguatge visual per a especificar, construir o documentar un sistema.
- **Cas d'ús:** és la descripció de les accions que realitzarà un sistema amb l'usuari per aconseguir un objectiu específic dins d'un determinat escenari.
- **Esdeveniment de sistema:** és una acció dins d'un cas d'ús.
- **Base de dades:** és una col·lecció de dades estructurades.
- **Codi obert:** es diu del software que té accessible el seu codi font i pot ser modificat i distribuït per qualsevol persona. Es pot dir que és una alternativa al software comercial de pagament.
- **Javascript:** és un llenguatge script orientada a objectes.

- **Node.js:** és un entorn d'execució de Javascript.
- **NPM:** és un gestor de paquets Javascript de Node.js. S'utilitza des de la línia de comandes i permet tenir un control sobre els paquets o llibreries instal·lats en un projecte.
- **Dependència:** en informàtica, és una aplicació o llibreria requerida per un altre projecte o programa.
- **SDK d'Alexa:** és un kit de software de codi obert ofert per Alexa amb diferents utilitats en el desenvolupament d'una skill.

1. INTRODUCCIÓ

1.1. Objectius del treball

L'objectiu principal que es vol aconseguir en aquest treball és donar a conèixer la tecnologia dels assistents per veu. La intenció és oferir un servei d'atenció per a la comunitat de l'Escola i a possibles visitants amb informació d'interès sobre la mateixa.

Per a assolir aquest objectiu, es realitzarà un estudi pel disseny d'una skill per l'assistent per veu Alexa i la seva posterior implementació. Això es tradueix en els següents objectius concrets:

- Entendre la lògica de funcionament de l'assistent per veu Alexa
- Estudiar com desenvolupar una skill
 - Llenguatges de programació
 - Documentació sobre APIs
- Identificar la informació d'interès que oferirà la skill
 - Sistemes d'informació de l'Escola
 - Altres sistemes d'informació
- Estudiar com dissenyar la skill
 - Proposta i tria de casos d'ús
- Implementar la skill
 - Desenvolupament dels casos d'ús
- Validar la skill

L'estructura de la memòria està basada en aquests objectius. Cada secció presenta com s'ha assolit un d'aquests objectius plantejats, afegint l'estat de l'art a l'inici de la memòria i una sèrie d'annexos per a completar la informació. Hi ha tres annexos, el primer detalla aspectes de la implementació que són útils, i els altres dos descriuen un manual d'usuari i un manual de manteniment de la skill respectivament.

2. ESTAT DE L'ART

2.1. Assistents per veu

Un assistent per veu és un agent de software que mitjançant reconeixement per veu, permet a les persones interactuar amb un ordinador, un smartphone o altres dispositius a través de comandes de veu. El mode principal d'interacció és la veu, però depenent del dispositiu, l'assistent ho pot combinar amb altres, com per exemple

l'ús del teclat o una pantalla tàctil. La interacció per veu pot permetre als usuaris iniciar serveis automatitzats i executar tasques d'una manera més ràpida e intuïtiva.

2.1.1. Cronologia

L'ús del reconeixement de veu dels assistents no és res nou. Anys enrere, concretament l'any 1962, IBM va presentar a la Fira Mundial de Seattle un dispositiu anomenat Shoebox capaç de realitzar funcions matemàtiques reconeixent comandes per veu. Va reconèixer els díigits del 0 al 9 i 16 paraules parlades que permetien operacions com "més", "menys", "total" entre d'altres. El dispositiu tenia la mida i la forma d'una capsa de sabates, s'utilitzava parlant per un micròfon que convertia els sons de la veu en impulsos elèctrics, aquests es classificaven i activava la màquina sumadora que estava adjunta a través d'un sistema de relés.



Figura 1. IBM Shoebox
Font: pàgina oficial IBM [1]

Durant la dècada de 1970, investigadors de la Universitat Carnegie Mellon a Pittsburgh, Pensilvania van llançar el sistema de reconeixement de veu Harpy que podia reconèixer aproximadament 1.000 paraules. Durant aquest període es va iniciar la experimentació de reconeixement de veu independentment del locutor.

A mitjans de la dècada de 1980, IBM aconsegueix que la seva màquina Tangora reconegui aproximadament 20.000 paraules parlades amb capacitat d'adaptar-se a la veu de l'usuari. Però encara era necessària una parla clara, sense soroll de fons i amb pauses entre paraules. Tangora utilitzava els models ocults de Markov, HMM, això li permetia una predicció dels pròxims fonemes basada en patrons.

L'any 1997, va sortir el primer producte de dictat continu, Dragon Naturally Speaking. Això significava que el locutor no havia de fer pauses entre paraules, aquest software podia comprendre aproximadament 100 paraules i transformar-ho en contingut llegible.

A la dècada dels 2000, es va començar a utilitzar l'aprenentatge automàtic proporcionant avenços importants. Google va combinar l'última tecnologia amb la computació basada en el núvol per compartir dades i millorar la precisió dels algoritmes d'aprenentatge automàtic. Això va fer possible el llançament de l'aplicació Google Voice Search per iPhone l'any 2008, aquesta aplicació va mostrar millores en els nivells de precisió respecte totes les tecnologies per reconeixement de veu anteriors.

Des de 2010, la tecnologia per reconeixement de veu ha evolucionat i s'ha integrat en la societat molt significativament, integrant-se en el dia a dia de les persones.

Siri es publica com una aplicació mòbil per a iPhone l'any 2010. Apple en veure el potencial, va comprar l'empresa i va integrar Siri com a assistent per veu en tots els seus productes amb capacitat de veu.

Apareix l'any 2012 l'assistent per veu Nina, de la companyia Nuance. No va tenir gaire èxit com a assistent per veu pel consumidor, però moltes empreses l'utilitzen per a interactuar amb els seus clients. Un any més tard, Microsoft anuncia l'assistent per veu Cortana, i l'integra a Windows, Xbox i altres productes.

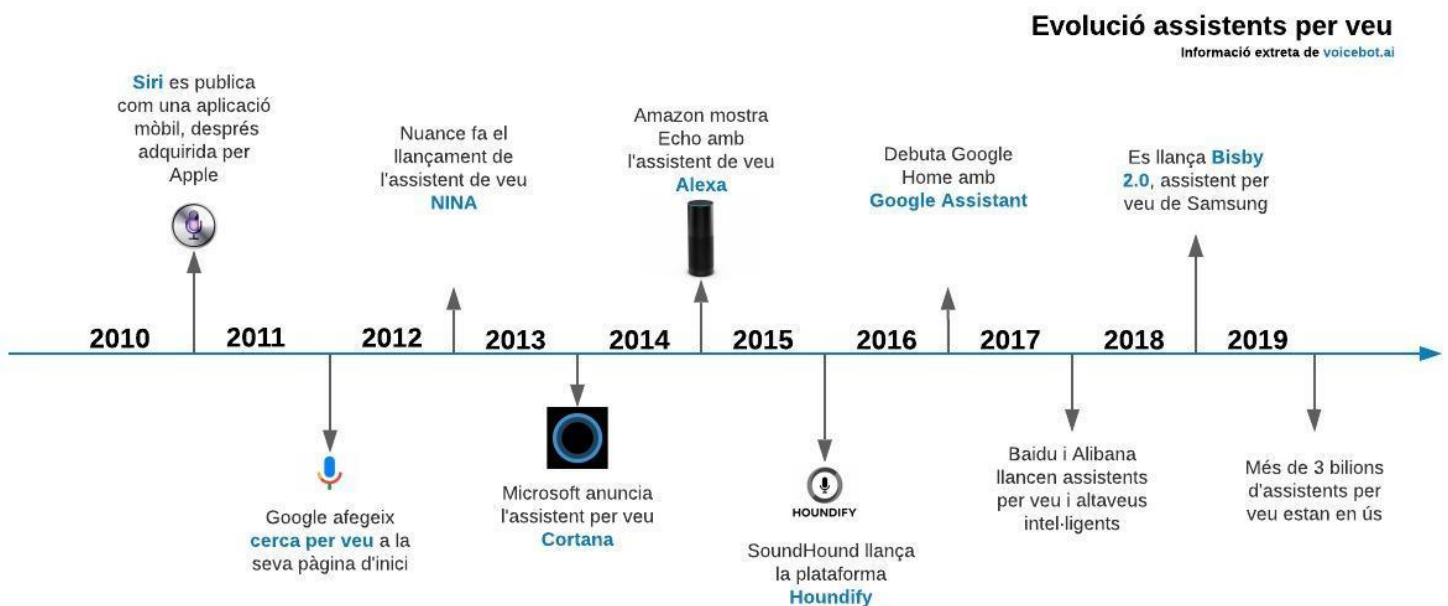


Figura 2. Cronograma de l'evolució dels assistents per veu l'última dècada
Font: informació extreta de voicebot.ai [2]
Creat amb Lucidchart [3]

L'any 2014, Amazon llença l'altaveu intel·ligent Echo amb l'assistent per veu Alexa. Aquí comença l'era dels altaveus intel·ligents i la gran integració dels assistents per veu amb la vida quotidiana de les persones.

Google presenta Google Home amb l'assistent per veu de Google com a competència d'Alexa l'any 2016. Un any més tard també apareixen altres altaveus intel·ligents amb els seus assistents com els de Baidu i Alibaba.

Al febrer de 2019, l'empresa Juniper Research, dedicada a la investigació i pronòstic del mercat tecnològic va estimar que estaven en ús aproximadament 3.250 milions d'assistents per veu. També ha realitzat una estimació per l'any 2023 on aquest ús

podria arribar a 8 bilions d'assistents. Els dispositius més comuns on s'usen aquests assistents serien els altaveus intel·ligents i els telèfons.

2.2. Altaveus intel·ligents

L'aparició dels assistents per veu anys enrere en grans empreses com Google i Amazon, entre d'altres, va fer aparèixer els altaveus intel·ligents a preus econòmics, fent possible l'accés a quasi tothom que volgués disposar d'un o varis pel seu ús personal. L'usuari, mitjançant el seu dispositiu, pot obtenir informació, automatitzar tasques entre d'altres només interactuant amb la veu.

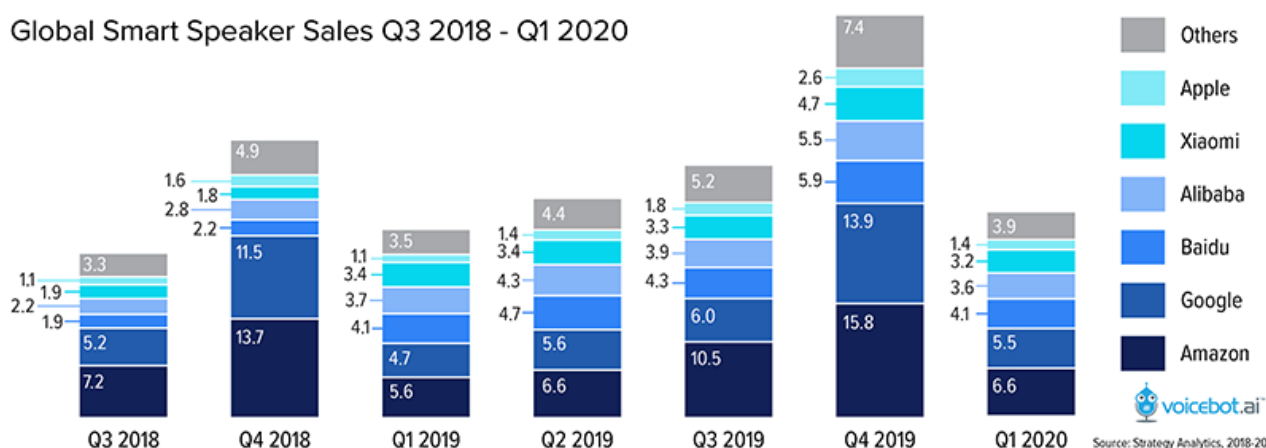


Figura 3. Vendes d'altaveus intel·ligents 2018 Q3 – 2020 Q1 a nivell global (en milions)
Font: imatge extreta de voicebot.ai [4]

Com podem observar a la *Figura 3*, les vendes d'altaveus intel·ligents són molt altes. El trimestre de vendes més baix va ser l'any 2018 entre el mes de juliol i setembre amb 22.8 milions d'altaveus venuts i el trimestre més alt l'any 2019 entre el mes d'octubre i desembre.

Per tant, podem afirmar que actualment els altaveus intel·ligents s'han integrat en la societat i formen part de la vida quotidiana de moltes persones.

3. ALEXA : LA INTEL·LIGÈNCIA ARTIFICIAL D'AMAZON

Alexa és l'assistent per veu d'Amazon ubicat al núvol i disponible a tots els dispositius de la marca. Permet crear experiències de veu naturals i oferir als usuaris una forma més fàcil e intuïtiva d'interactuar amb la tecnologia. A l'hora d'adquirir un dispositiu, Alexa conté una sèrie de funcionalitats bàsiques anomenades skills. Es poden afegir més funcionalitats instal·lant les que estan disponibles a la botiga de skills d'Amazon

o creant la teva pròpia skill. Això és possible perquè Amazon ofereix a tothom un conjunt d'eines i documentació per a poder desenvolupar skills i poder publicar-les a la botiga de skills.

3.1. Plataforma Alexa

Amazon és la companyia que proveeix la plataforma on desenvolupadors contribueixen creant skills utilitzant el mateix entorn de treball, llibreries, funcionalitats i serveis que l'assistent per veu. Això és possible perquè Amazon dona accés a tota la tecnologia de la plataforma Alexa, per exemple, es poden utilitzar totes les eines complexes de reconeixement de veu i la seva comprensió sense haver-les de construir.

Per altra banda, també conformen aquest entorn els fabricants de hardware. Aquests poden integrar el servei d'Alexa en els seus dispositius si compleixen uns mínims requeriments com són contenir un altaveu, micròfon i connexió a Internet.

Finalment, els usuaris poden comprar els dispositius Echo o hardware específic per l'us d'Alexa fabricat per Amazon. Això permet tenir accés a l'aplicació de configuració dels dispositius e instal·lar les funcionalitats que necessitin des de la botiga de skills d'Amazon.

3.2. Dispositius

Amazon produeix diversos tipus de dispositius habilitats per Alexa, tots aquests connectats a l'entorn de treball d'Alexa. Entre aquests dispositius podem trobar l'Echo Dot, el dispositiu emprat per a la realització d'aquest projecte, l'Echo Plus, l'Echo Auto, l'Echo Show entre d'altres. Els requisits que compleixen tots aquests dispositius per a ser compatibles amb l'entorn d'Alexa són disposar d'un micròfon, un altaveu, connexió a Internet i una sèrie de components hardware per al tractament de les seqüències de veu rebudes i per les que han de ser reproduïdes. Aquest tractament de seqüències de veu consisteix en transmetre les gravacions d'àudio al servei d'Alexa, processar-ho i així poder enviar les respostes en format d'àudio a través de l'altaveu.

3.3. Servei de veu Alexa

El servei de veu d'Alexa al núvol, AVS, està dissenyat per a imitar conversacions reals, el que els usuaris li diuen són comandes de veu per a que aquest servei realitzi les tasques sol·licitades. Ofereix una sèrie de serveis que són utilitzats per les skills d'Alexa:

Reconeixement de la parla <i>Automated Speech Recognition (ASR)</i>	Reconeix sons com a paraules, realitza la conversió de veu a text.
Comprensió del llenguatge natural <i>Natural Language Understanding (NLU)</i>	Entén la sol·licitud de l'usuari a partir de les paraules reconegudes. Aquest servei utilitza tècniques complexes d'intel·ligència artificial.
Comunicació AVS - Skill	Transmet l'intent detectat a la skill i espera la resposta en format text.
Síntesi de veu <i>Text-To-Speech (TTS)</i>	Converteix la resposta textual en veu i la reproduïx.

Taula 1. Serveis oferts pel servei de veu Alexa

El servei de veu d'Alexa és l'encarregat de dur a terme la conversació, és a dir, d'entendre les gravacions de veu que li arriben, comunicar-se amb la skill corresponent i de convertir la resposta en format text en veu. La veu d'Alexa es generada per una llarga xarxa de memòria neuronal i artificial.

Per altra banda, en els dispositius integrats amb Alexa no es descarreguen les diferents skills, aquests dispositius simplement envien i reben informació del "núvol". Al núvol és on es troba tot el codi. Quan s'instal·la una skill el servei d'Alexa crea un enllaç entre aquest usuari, dispositiu i skill.

Finalment, podem veure a la *Figura 4* l'arquitectura del servei de veu Alexa amb les diferents parts de les skills.

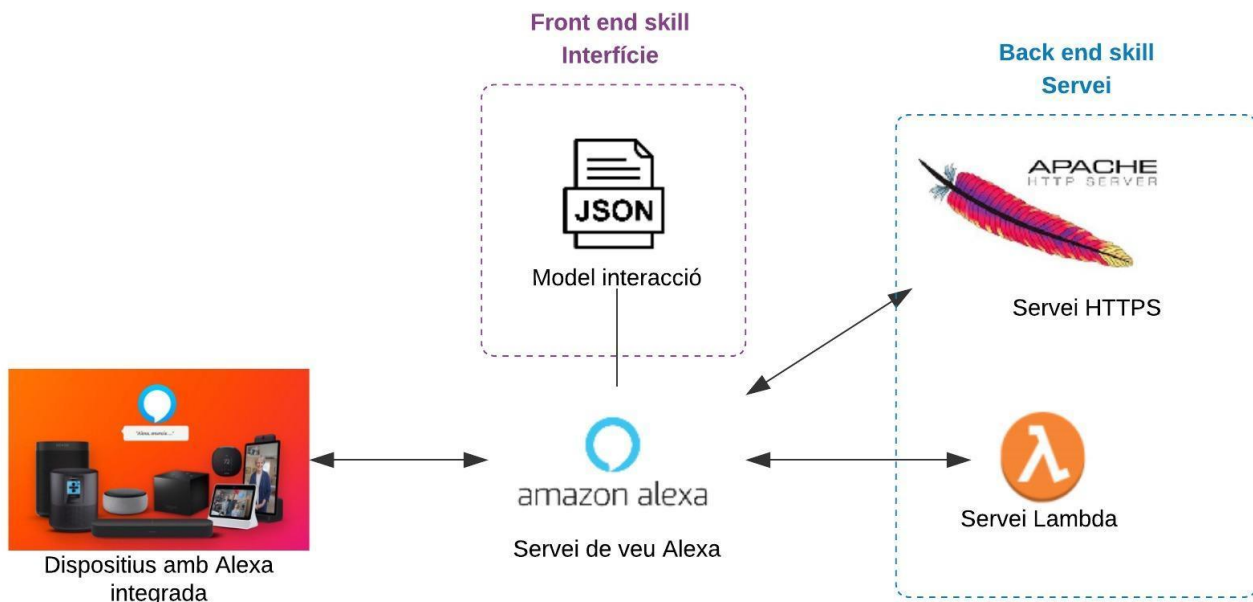


Figura 4. Arquitectura AVS amb les diferents parts d'una skill
Creat amb Lucidchart [3]

Tota skill està formada d'una part d'interfície, és a dir, front end i d'una part de servei, back end. Com podem veure a la *Figura 4*, la part de front end seria un arxiu JSON que correspon amb la descripció del model d'interacció, explicat més endavant a la *secció 3.4.2*. En el cas del back end, estarà format per un servidor.

3.4. Skills

Les skills d'Alexa com s'ha comentat anteriorment, són les diferents funcionalitats que pot oferir l'assistent Alexa, per tant, es pot fer un paral·lelisme entre elles i les aplicacions dels telèfons mòbils. Aquestes skills permeten ampliar les funcionalitats bàsiques d'Alexa i per tant, permeten la personalització per part de l'usuari de la interfície d'usuari per veu.

Amazon proporciona una sèrie de skills bàsiques en el moment d'adquirir un dispositiu, però l'usuari pot trobar milers de skills a la botiga d'Amazon. El funcionament d'aquesta és similar a la plataforma Play Store, on es distribueixen les diferents aplicacions per a sistemes operatius Android i com a desenvolupador pots publicar les teves aplicacions.

3.4.1. Tipus

Existeixen diferents models de skills, aquests models determinen com s'integra la skill amb el servei de veu d'Alexa i el que es necessita desenvolupar. Cada skill té un model d'interacció que determina les sol·licituds que la skill pot resoldre i les paraules que els usuaris utilitzen per fer aquestes sol·licituds. El desenvolupador el pot definir mitjançant una custom skill o pot fer ús de la resta de skills predissenyades on les possibles sol·licituds i expressions ja estan predefinides.

Smart home skills

Serveixen per a controlar dispositius domèstics intel·ligents com llums, endolls, termòstats, televisors intel·ligents entre d'altres. Proporcionen de forma senzilla pels desenvolupadors i usuaris el control i monitorització dels dispositius connectats al núvol.

Aquesta skill no permet al desenvolupador tenir tot el control sobre l'experiència d'usuari, però té un desenvolupament més simple perquè no necessita crear la interfície d'usuari. Contenen unes sol·licituds que poden efectuar anomenades *device directives* com encén/apaga, incrementa/decrementa la temperatura entre d'altres i com a desenvolupador ha de definir com respon a cada directiva.

Per part dels usuaris aquestes skills són fàcils d'invocar perquè no necessiten recordar cap nom, simplement poden realitzar sol·licituds parlant a Alexa, com per exemple, "Alexa, enciende las luces de la cocina".

Flash briefing skill

Aquestes skills proporcionen una descripció general de les notícies o altres continguts configurats per l'usuari a la seva aplicació Alexa. El desenvolupador no té el control del model d'interacció de la interfície de veu. L'API de skills Flash Briefing defineix paraules per a que els usuaris invoquin aquestes sol·licituds com "Alexa, dime las noticias".

El desenvolupador s'encarrega de definir una o més fonts de contingut, aquest contingut pot ser àudio que es reproduïx a l'usuari o text que Alexa llegeix.

Video skills

Per als dispositius amb pantalla, permet mostrar contingut de vídeo fàcilment sense invocar a una skill específica.

Assistent personal per l'EPSEVG basat en Alexa
Miriam Martínez Izquierdo

El desenvolupador s'encarrega de definir les sol·licituds que la skill pot gestionar com ara reproduir i cercar contingut de vídeo i definir com es mostren els resultats de la cerca.

Music skills

Permet proporcionar contingut d'àudio com cançons, llistes de reproducció o emissores de ràdio. L'API d'aquest tipus de skill gestiona les paraules que un usuari pot dir per demanar i controlar el contingut d'àudio i reconèixer les sol·licituds d'aquesta skill.

El desenvolupador ha d'implementar una funció de Lambda on gestiona les sol·licituds i respon enviant contingut d'àudio per l'usuari.

Custom skills

Una skill personalitzada és el tipus de skill que s'ha escollit per aquest projecte, permet obtenir un major control sobre la experiència d'usuari. És el tipus de skill més flexible que Amazon permet desenvolupar, però també el més complex perquè s'ha de proporcionar el model d'interacció que explicarem més endavant a la *secció 3.4.2*.

En aquest tipus de skill s'ha d'implementar la part d'interfície i la part de servei, és a dir, el seu front end i el seu back end. Per tant, per a desenvolupar una custom skill s'ha de crear el model d'interacció i el servei allotjat al núvol que accepta les sol·licituds definides al model d'interacció i completa les accions oportunes sobre elles per donar una resposta.

Per a implementar el servei es pot utilitzar AWS Lambda o implementar un servei web que accepti sol·licituds i envii les respostes al servei de veu d'Alexa complint una sèrie de requisits [5].

La *Figura 5* ens mostra un diagrama del flux d'interacció entre una custom skill i l'usuari, aquest diagrama és la descripció de les accions de l'usuari i les respostes del sistema.

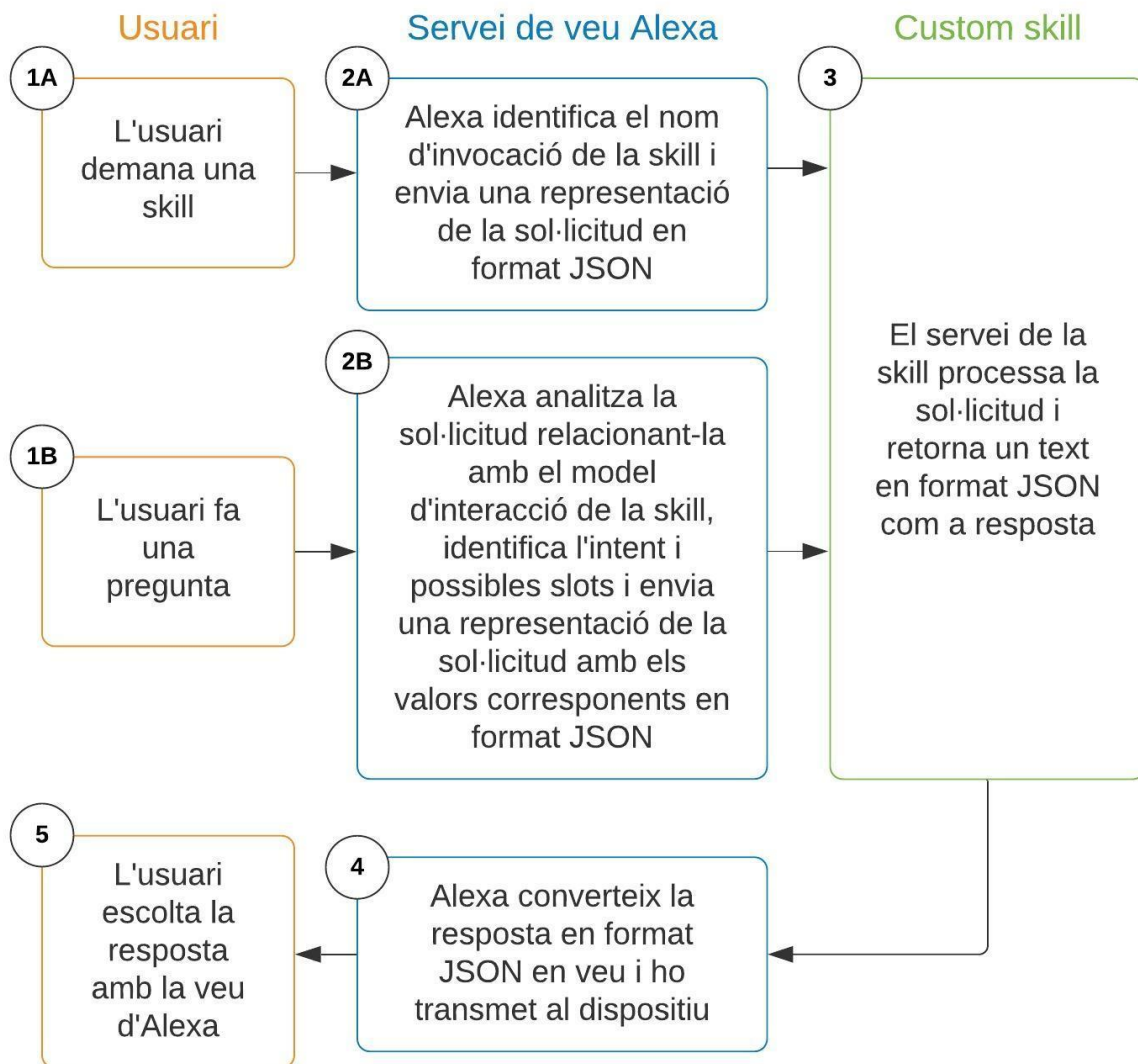


Figura 5. Diagrama d'interacció de l'usuari amb una custom skill
Font: basat en el diagrama d'interacció de la documentació oficial d'Alexa [6]
Creat amb Lucidchart [3]

1A-B. La parla de l'usuari es transmet al servei de veu d'Alexa al núvol.

2A. Alexa reconeix el nom d'invocació de la skill i envia al servei d'aquesta skill una representació de la sol·licitud d'invocació en format JSON.

2B. Alexa reconeix l'intent de la skill que representa el que ha dit l'usuari i estructura la informació de la sol·licitud en un arxiu JSON que inclou també el valor de l'slot si n'hi ha i ho envia al servei de la skill.

3. El servei de la skill rep la informació de la sol·licitud i realitza l'acció de l'intent sol·licitat i envia la resposta que dir-li a l'usuari de manera estructurada en format JSON.

4. El servei de veu Alexa rep la resposta en format text i ho converteix en veu.
5. El dispositiu reproduïx la veu d'Alexa.

3.4.2. Model d'interacció

El model d'interacció per una skill és l'estructura que determina les sol·licituds a les que pot donar resposta i controlar les paraules que els usuaris diuen per invocar aquestes sol·licituds. Aquesta estructura queda representada en un arxiu JSON, aquest inclourà totes les possibilitats d'interacció amb l'usuari disponibles per la skill.

Tots els tipus de skill comentats a la *secció 3.4.1.* menys els custom skills tenen un model d'interacció predefinit. En el cas del custom skill s'ha de crear un model d'interacció personalitzat.

Per a la creació d'un model d'interacció personalitzat, s'han de definir:

- El nom que utilitza Alexa per a identificar la skill, anomenat ***invocation name***.
- Les sol·licituds a les que la skill pot donar resposta. Aquestes sol·licituds són les anomenades ***intents***. Un intent representa una acció que correspon a la sol·licitud demanada per un usuari. Estan formats per unes expressions de mostra, anomenades ***sample utterances*** i opcionalment per arguments anomenats ***slots***.
 - **Sample utterances** : un conjunt de probables frases que l'usuari pot dir per a cridar un intent. Hi ha d'haver tantes frases com sigui possible per a ajudar a relacionar l'intent corresponent quan l'usuari ho sol·licita.
 - **Slots** : són arguments dins d'un sample utterance per a poder conèixer la informació que busca l'usuari. Cal assignar-li un tipus a l'hora de crear-los amb una sèrie de valors. Amazon ofereix tipus predefinits.

Per acabar d'entendre el model d'interacció i les diferents parts que el formen, a la *Figura 6*, es mostra un exemple del que podria ser un model d'interacció d'una custom skill orientada a un centre mèdic.

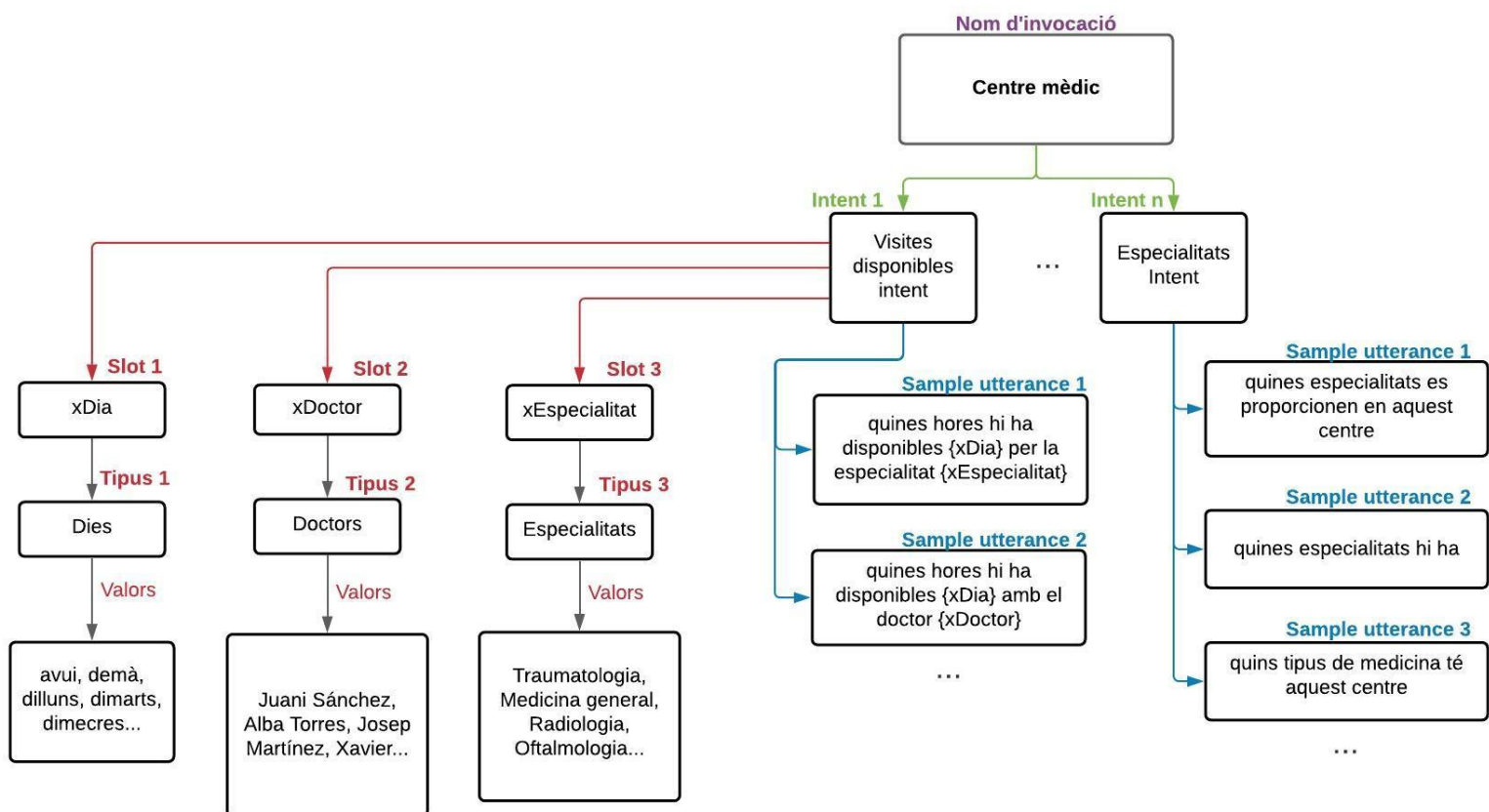


Figura 6. Exemple de model d'interacció d'una custom skill fictícia
Creat amb Lucidchart [3]

4. DISSENY DE LA SKILL

4.1. Funcionalitat

La funcionalitat de la skill desenvolupada en aquest projecte és la d'accedir a dades de l'Escola disponibles a la pàgina web d'aquesta a través de l'assistent Alexa mitjançant la interacció de veu. L'objectiu és oferir un servei d'atenció amb informació d'interès sobre l'Escola a la seva comunitat i a possibles visitants.

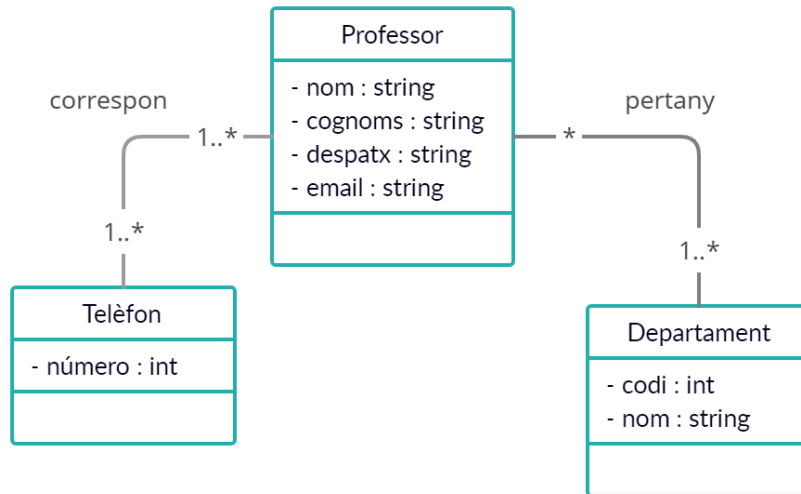
4.2. Casos d'ús

En aquest projecte s'han definit una sèrie de casos d'ús que es basen en informació pública i disponible a la pàgina web de l'Escola.

4.2.1. Consulta informació professors

La responsabilitat d'aquest cas d'ús es retornar les dades que es demanen sobre un professor de l'Escola. Aquestes dades poden ser el número de telèfon, el despatx, l'horari de consultes i el departament al que pertany.

A la *Figura 7* es troba representada l'especificació que correspondria a aquest cas d'ús.



*Figura 7. Especificació del cas d'ús informació professors
Creat amb Creately [7]*

Com es mostra a la especificació, un professor té com atributs el seu nom i cognoms, el despatx i un e-mail, també pot tenir més d'un número de telèfon i un telèfon pot correspondre a més d'un professor. Cada professor pertany com a mínim a un departament i un departament està format per més d'un professor.

Aquest cas d'ús tractaria els següents esdeveniments de sistema segons la informació demanada:

- Retornar número de telèfon.
- Retornar despatx.
- Retornar nom del departament al que pertany.
- Retornar e-mail.

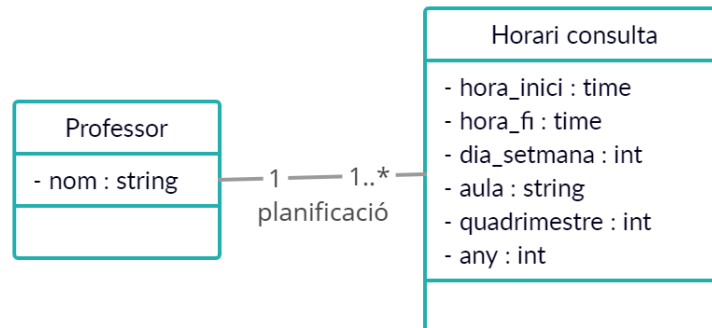
L'usuari ha d'indicar la informació que vol rebre i especificar el nom i cognom del professor. Les precondicions per a poder complir la responsabilitat de cada esdeveniment de sistema, és a dir, retornar la informació demanada són:

- Existeix el professor de nom i cognom indicats al sistema de l'Escola.
- Existeix la informació demanada d'aquest professor al sistema.

4.2.2. Consulta informació horari de consulta dels professors

Aquest cas d'ús té la responsabilitat de retornar les dades demanades sobre l'horari de consultes del curs actual d'un professor.

La *Figura 8* mostra l'especificació necessària per aquest cas d'ús.



*Figura 8. Especificació del cas d'ús horari consultes
Creat amb Creately [7]*

Com es mostra a l'especificació, un professor pot tenir un o diferents horaris de consulta durant un any i quadrimestre determinats. La informació sobre un horari de consulta és el dia de la setmana, l'aula, hora d'inici i fi de la disponibilitat.

Aquest cas d'ús tracta l'esdeveniment de sistema encarregat de retornar la informació sobre els horaris d'un professor específic en l'any i quadrimestre actual. Les precondicions per a que es compleixi la seva responsabilitat són:

- Existeix el professor de nom i cognom indicats al sistema de l'Escola.
- Existeix un o diferents horaris de consulta per aquest professor l'any i quadrimestre actuals.

4.2.3. Consulta informació departaments

Aquest cas d'ús té la responsabilitat de retornar les dades demanades sobre un departament de l'Escola. Aquestes dades poden ser el codi del departament, el nom o el nom del responsable del departament a l'Escola.

La *Figura 9* mostra l'especificació necessària per aquest cas d'ús.

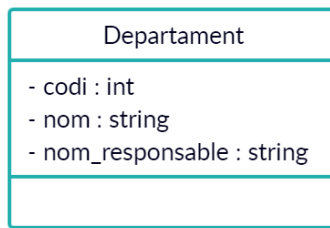


Figura 9. Especificació del cas d'ús informació departaments
Creat amb Creately [7]

Com es mostra a la especificació, un departament té un codi, un nom i el nom del responsable d'aquest a l'Escola.

Aquest cas d'ús pot tractar els següents esdeveniments de sistema:

- Retorna el codi.

Precondicions

1. Existeix un departament al sistema de l'Escola amb el nom especificat per l'usuari.
2. Existeix el codi d'aquest departament al sistema de l'Escola.

- Retorna el nom.

Precondicions

1. Existeix un departament al sistema de l'Escola amb el codi especificat per l'usuari.
2. Existeix el nom d'aquest departament al sistema de l'Escola.

- Retorna el nom del responsable a l'Escola.

Precondicions

1. Existeix un departament al sistema de l'Escola amb el codi especificat per l'usuari.
2. Existeix el responsable d'aquest departament al sistema de l'Escola.

Si les precondicions definides es compleixen, l'esdeveniment de sistema podrà complir la seva responsabilitat de retornar la informació demanada per l'usuari.

4.2.4. Consulta informació aules

Aquest cas d'ús té la responsabilitat de retornar l'edifici i la planta on està ubicada un aula que pertany a l'Escola. L'única precondició que s'ha de complir per a que l'esdeveniment de sistema encarregat de retornar la informació d'un aula ho realitzi és que es faciliti un codi d'aula que existeixi en algun edifici de l'Escola.

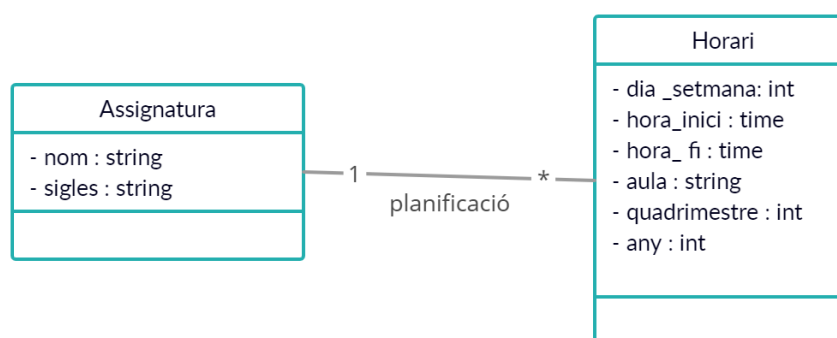
4.2.5. Consulta informació setmana actual

Aquest cas d'ús és responsable de retornar el número de setmana i ordre del dia actual. L'única precondició que s'ha de complir per a que l'esdeveniment de sistema encarregat retorni la informació demanada és que existeixi un número de setmana i ordre pel dia actual, és a dir, que sigui un dia amb docència.

4.2.6. Consulta informació horari assignatures actuals

La responsabilitat d'aquest cas d'ús és retornar les dades demanades sobre l'horari d'una assignatura.

La *Figura 10* mostra l'especificació necessària per aquest cas d'ús.



*Figura 10. Especificació del cas d'ús horari assignatures
Creat amb Creately [7]*

Com es mostra a l'especificació, una assignatura té com a atributs el seu nom i les sigles. Aquesta assignatura tindrà diferents horaris que representen el dia de la setmana, l'aula, l'hora d'inici i l'hora de fi de la classe, el quadrimestre i l'any en què es realitza.

Aquest cas d'ús pot tractar els diferents esdeveniments de sistema:

- Retorna tots els horaris del curs actual d'una assignatura.
- Retorna els horaris del curs actual d'un dia de la setmana d'una assignatura.
- Retorna tots els horaris del curs actual d'un grup concret d'una assignatura.

L'usuari ha d'indicar la informació que vol rebre i especificar el nom o sigles de l'assignatura i si és el cas, el dia o grup de classe del que vol conèixer l'horari. Les precondicions per a poder complir la responsabilitat de cada esdeveniment de sistema, és a dir, retornar la informació demanada són:

- Existeix una assignatura al sistema de l'Escola amb el nom/sigles especificat per l'usuari.
- Existeixen horaris per aquesta assignatura al sistema de l'Escola l'any i quadrimestre actuals i si s'ha especificat, pel dia o grup de classe indicats.

Si les precondicions definides es compleixen, l'esdeveniment de sistema podrà complir la seva responsabilitat de retornar la informació demanada per l'usuari.

4.2.7. Consulta informació exàmens

La responsabilitat d'aquest cas d'ús és retornar les dades sobre el tipus d'examen demanat. Els tipus d'examen són examen parcial, final i de reavaluació. Per tant, l'usuari especificarà el tipus d'examen i l'assignatura a la que correspon i es retornaran les dades sobre aquest examen.

La *Figura 11* mostra l'especificació necessària per aquest cas d'ús.

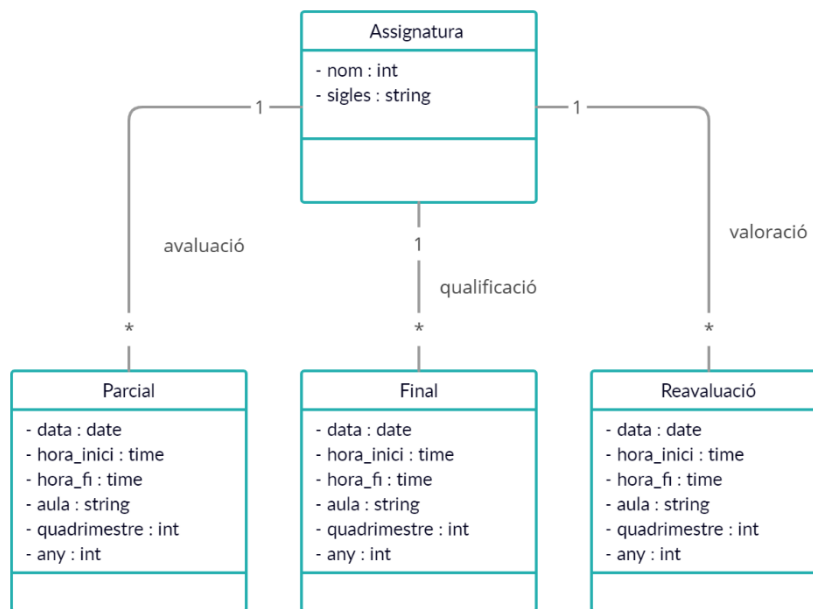


Figura 11. Especificació del cas d'ús informació departaments
Creat amb Creately [7]

Com es mostra a l'especificació, una assignatura té com a atributs el seu nom i les sigles. Aquesta assignatura tindrà un examen parcial, un final i una reavaluació per cada any i quadrimestre que es realitzi. Les dades que contenen tots els tipus d'examen són la data de realització, l'hora d'inici, l'hora de fi, l'aula, el quadrimestre i l'any en que es realitza.

Aquest cas d'ús pot tractar els diferents esdeveniments de sistema:

- Retorna les dades de l'examen parcial d'una assignatura.
- Retorna les dades de l'examen final d'una assignatura.
- Retorna les dades de l'examen de reavaluació d'una assignatura.

L'usuari ha d'indicar el tipus d'examen i especificar el nom o sigles de l'assignatura a la que correspon per poder obtenir les dades. Les precondicions per a poder complir la responsabilitat de cada esdeveniment de sistema, és a dir, retornar la informació demanada són:

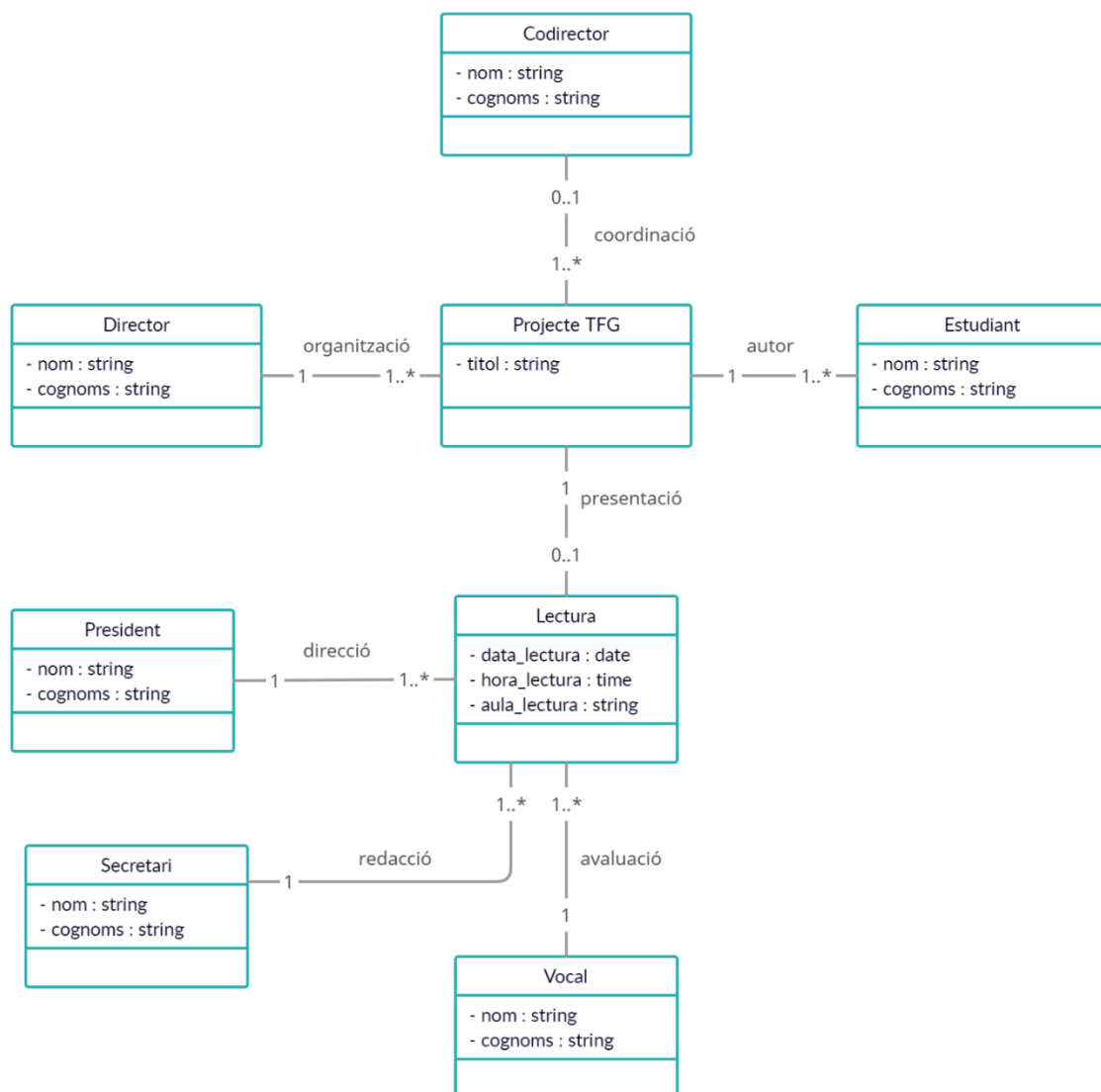
- Existeix una assignatura al sistema de l'Escola amb el nom/sigles especificat per l'usuari.
- Existeix el tipus d'examen per aquesta assignatura al sistema de l'Escola l'any i quadrimestre actuals.

Si les precondicions definides es compleixen, l'esdeveniment de sistema podrà complir la seva responsabilitat de retornar la informació demanada per l'usuari.

4.2.8. Consulta informació lectures TFG

La responsabilitat d'aquest cas d'ús és retornar les dades demanades per l'usuari sobre la lectura d'un TFG. Les dades disponibles sobre la lectura d'un TFG són el nom i cognoms de l'estudiant/s, el nom i cognoms dels diferents professors participants a la lectura amb el seu rol, la data, hora i aula on es realitza. Els diferents rols dels professors poden ser director, codirector, president, vocal o secretari.

A la *Figura 12* es mostra l'especificació necessària per aquest cas d'ús.



*Figura 12. Especificació del cas d'ús sobre lectures TFG
Creat amb Creately [7]*

Aquest cas d'ús pot tractar diferents esdeveniments de sistema:

- Retorna els noms i cognoms dels alumnes que tenen prevista una lectura del TFG el dia actual.

Precondicions

1. Existeix una o diverses lectures de TFG amb data del dia actual al sistema.
- Retorna les dades sobre la lectura del TFG d'un alumne determinat.
Aquestes dades són la data, hora i aula on es realitza la lectura i els noms i cognoms de tots els participants en aquesta amb el seu rol.

Precondicions

1. Existeix una lectura de TFG per l'estudiant indicat en el curs actual al sistema .
 2. Existeix un director i opcionalment, un codirector per aquesta lectura de TFG al sistema.
 3. Existeixen els membres del tribunal d'aquesta lectura al sistema.
- Retorna la informació de les lectures de TFG on participa el dia actual un professor determinat.
Aquesta informació és el rol que té el professor a la lectura, el nom i cognoms de l'estudiant/s i l'hora i aula on es realitza.

Precondicions

1. Existeix al sistema una o diverses lectures de TFG amb data del dia actual on el professor indicat participa amb qualsevol rol dels possibles.

Si les precondicions definides es compleixen, l'esdeveniment de sistema podrà complir la seva responsabilitat de retornar la informació demanada per l'usuari.

5. EINES UTILITZADES PER A LA IMPLEMENTACIÓ

5.1. Front end

El front end és la part d'una aplicació web que interactua amb els usuaris. En el cas d'una skill, la part que forma el front end seria el model d'interacció vist a la *secció 3.4.2.*, aquesta és la interfície de la skill.

L'eina utilitzada per a la implementació d'aquesta part ha estat Alexa Skills Kit.

5.1.1 Alexa Skills Kit

L'Alexa Skill Kit, ASK, és un conjunt de documentació, eines, plantilles de codi i APIs que permet desenvolupar skills de forma ràpida i senzilla. Per a poder utilitzar aquesta eina és necessari un compte d'Amazon [8]. Una vegada iniciem sessió podrem accedir a l'Alexa Developer Console.

L'Alexa Developer Console és un entorn de desenvolupament a la web, per tant, és el lloc web principal a l'hora de crear una skill. A la pàgina principal d'aquest entorn web trobarem un llistat amb les nostres skills creades i diferent informació sobre el nostre compte com a desenvolupador. Alexa Developer Console ens permet el desenvolupament del model d'interacció, la integració amb el nostre servei, la realització de proves, la distribució de la nostra skill, la certificació i les analítiques una vegada està publicada i activa.

5.2. Back end

El back end és la part d'una aplicació web transparent per l'usuari on es produeix l'accés a les dades e implementa la lògica per fer-ho. Per tant, aquesta part és la del servei, en aquest cas el servei de veu d'Alexa, la base de dades de l'Escola i el servidor que realitza la comunicació entre aquests.

Les eines utilitzades per a la implementació del back end ha estat AWS Lambda i una còpia de la base de dades MySQL de l'Escola.

5.2.1. AWS Lambda

AWS Lambda és una plataforma informàtica sense servidor proporcionada per Amazon com a part d'Amazon Web Services, AWS. És un servei informàtic que executa codi en resposta a esdeveniments, només quan es necessita, i gestiona automàticament els recursos informàtics requerits per aquest codi. Aquesta eina

permet la creació de serveis back end propis que funcionin amb el nivell de seguretat, rendiment i escala d'AWS.

AWS Lambda s'encarrega de l'administració completa dels recursos informàtics, inclosos el manteniment del servidor i sistema operatiu, l'aprovisionament de capacitat, la implementació de seguretat i la monitorització de codi i els seus registres.

El codi que s'executa a AWS Lambda s'anomena funció de Lambda. Cada funció de Lambda té les seves pròpies configuracions i el seu codi, i pot ser vinculada amb la nostra skill. Això vol dir, que la preocupació del desenvolupador es redueix a implementar el codi, pujar-lo a una funció Lambda i AWS s'encarrega d'executar-lo quan sigui necessari.

Les funcions de Lambda són compatibles amb llenguatges com JavaScript, Python, C#, Java, Go, i Ruby, en aquest projecte s'ha escollit el llenguatge JavaScript amb l'entorn d'execució Node.js.

5.2.1.1. Node.js

Node.js és un entorn d'execució de JavaScript orientat a esdeveniments asíncrons. És considerat lleuger, eficient i pot proporcionar una alta escalabilitat. El principal motiu per haver escollit aquest entorn ha estat que és altament recomanable en aplicacions que implementen *realtime* en els seus serveis, com per exemple chatbots, videoconferències, assistents per veu entre d'altres.

Javascript és un llenguatge no bloquejant, és a dir, les tasques que es realitzen no es queden bloquejades esperant ser finalitzades. S'anomena programació asíncrona perquè el codi asíncron no espera a la realització completa de les tasques, sinó que continua amb l'execució, veure *Figura 13*. Per tant, s'ha de gestionar aquesta asincronia en alguns casos, com per exemple en els que s'ha d'esperar una resposta. En aquest projecte, aquesta gestió de l'asincronia es realitza mitjançant promeses i el mecanisme `async/await`.

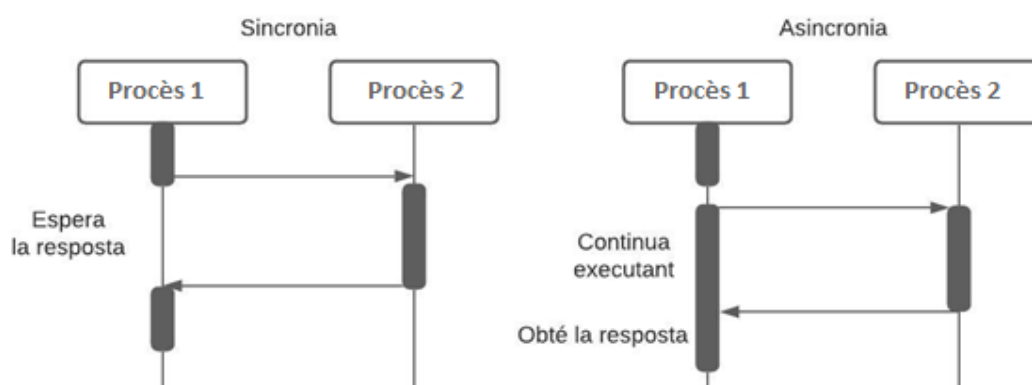


Figura 13. Programació síncrona vs programació asíncrona
Creat amb Lucidchart [3]

Promeses

Les promeses és la manera més actual de gestionar l'asincronia a Javascript aconseguint que el codi asíncron es comporti com si fos síncron. Aquestes promeses es creen en una funció per a poder ser consumida, és a dir, rebre el resultat en un altre funció.

Per implementar una promesa, es crea un nou objecte *Promise* on se li passa per paràmetre una funció amb dos callbacks. El primer, *resolve*, s'utilitzarà quan es compleixi per retornar el valor. El segon, *reject*, quan es rebutgi, per retornar l'error.

Async/await

Async/await funciona conjuntament amb les promeses per a una gestió completa de l'asincronia. La funció que consumirà una funció amb promesa s'ha de definir com *async* i abans de cridar a la funció que ens retorna el resultat de la promesa, s'ha d'escriure *await*. D'aquesta manera es poden cridar les funcions que retornen promeses com si retornessin valors síncrons. Si la promesa no retorna res, es rebrà un *undefined*.

5.2.1.2. Amazon CloudWatch

Amazon CloudWatch és l'eina que monitoritza els recursos i les aplicacions d'AWS que executa a temps real. Ens permet tenir un control a temps real dels registres, *logs*, de la nostra funció Lambda a temps real. D'aquesta manera es pot validar el codi gràcies al registre de les sol·licituds gestionades per la funció Lambda.

5.2.2. MySQL

MySQL és un sistema de gestió de base de dades relacionals de codi obert amb un model client-servidor. Una base de dades és una col·lecció de dades estructurades. Al ser relacional, el conjunt de dades són organitzats en forma de taules i cada taula pot estar relacionada amb un altre.

S'utilitza sintaxis SQL per a poder realitzar operacions CRUD a la base de dades, és a dir, crear, llegir, eliminar i actualitzar. En aquest projecte només es necessita fer operacions de consulta, llegir dades.

5.2.2.1 HeidiSQL

HeidiSQL, inicialment conegut com MySQL-Front, és un software lliure i de codi obert que permet connectar-se a servidors MySQL per administrar base de dades.

L'usuari ha d'iniciar sessió en un servidor MySQL local o remot. En aquest cas es connecta remotament a la còpia de la base de dades de l'Escola a la que tenim accés amb l'usuari i contrasenya corresponents. D'aquesta manera podem visualitzar de forma més clara que per terminal les dades de les diferents taules que conté i provar les diferents queries de consulta que es necessitin implementar.

6. IMPLEMENTACIÓ DE LA SKILL

6.1. Arquitectura de la skill

Com s'ha explicat anteriorment a la *secció 3.4.1.*, les custom skills estan formades per una part d'interfície, front end, i una de servei, back end. L'arquitectura de la skill *Asistente EPSEVG* creada es mostra gràficament a la *Figura 14.*

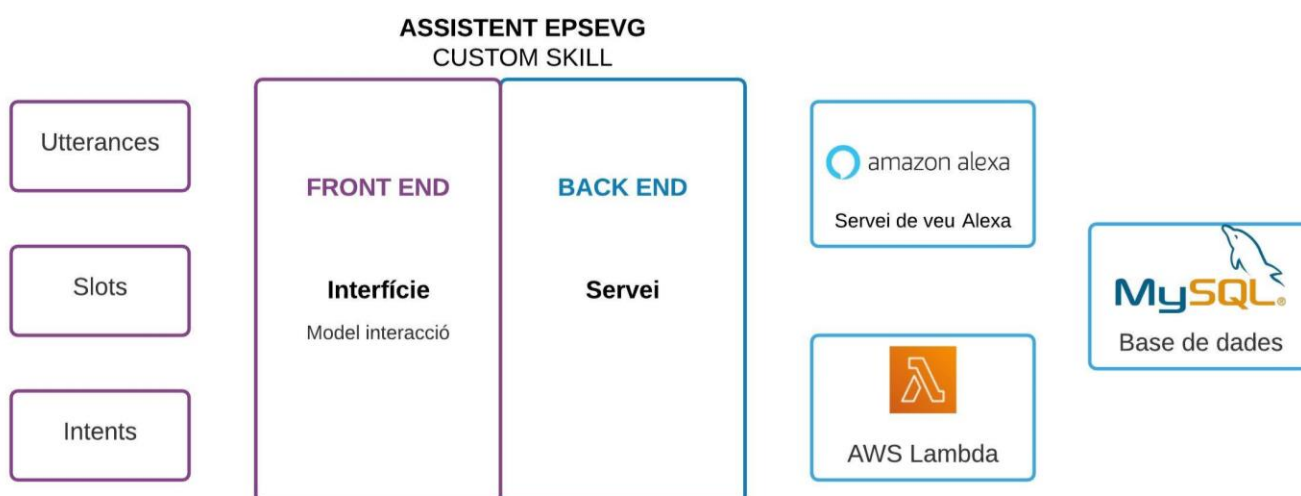


Figura 14. Arquitectura de la skill Asistente EPSEVG
Creat amb Lucidchart [3]

La part de front end és la que interactua amb l'usuari. Per tant, en una skill correspon al model d'interacció, un arxiu en format JSON que conté els seus components, intents, sample utterances i slots. L'explicació més detallada es troba a la *secció 3.4.2.*

La part de back end és la part transparent per l'usuari i encarregada de l'accés a les dades e implementació de la lògica.

En aquest cas, es treballa amb una funció Lambda en un entorn Node.js, a la *secció 5.2.1.* s'explica amb més detall. Aquesta funció Lambda, codificada en Javascript, és l'encarregada de l'obtenció de les dades i de la comunicació amb la skill i el servei de veu d'Alexa, AVS. En aquest projecte, es treballa amb una còpia de la base de dades MySQL de l'Escola. S'ha realitzat una còpia d'aquesta amb només la informació

pública i accessible també disponible a la pàgina web. En aquesta còpia s'ha exclòs informació confidencial o que pugui suposar una violació de la privacitat dels membres de la comunitat de l'Escola.

Totes les dades sobre l'Escola tractades a la skill es consulten directament a la base de dades, una altra opció podria ser parsejar la informació disponible a la pàgina web, però per motius de persistència de les dades s'ha decidit que és millor opció obtenir-les a partir de la base de dades.

6.2. Funcionament

Donada l'arquitectura anterior, s'explica mitjançant el diagrama de la *Figura 15* el funcionament de la custom skill desenvolupada.

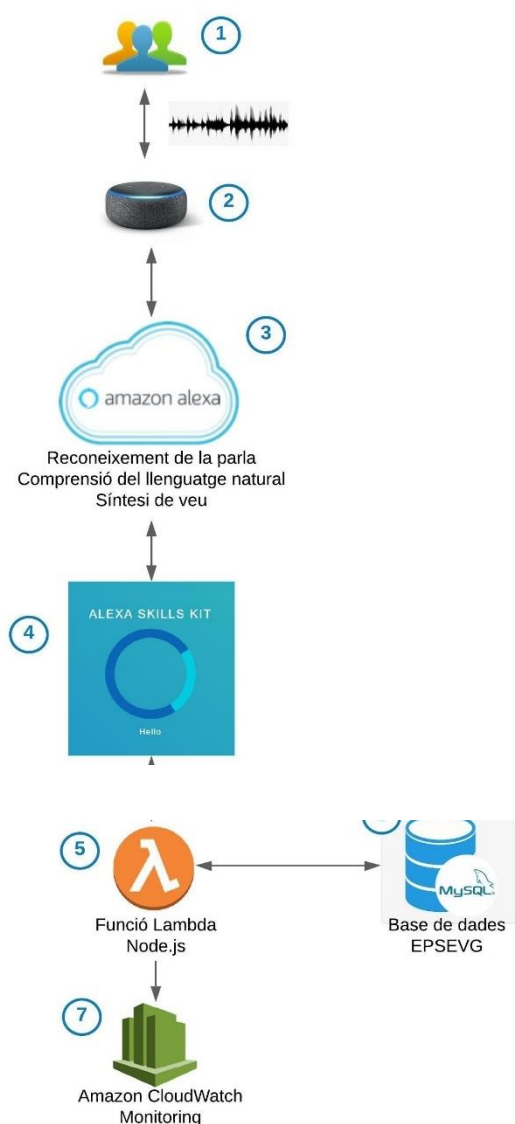


Figura 15. Funcionament de la skill Asistente EPSEVG
Creat amb Lucidchart [3]

1. Els usuaris interactuen amb Alexa mitjançant l'Amazon Echo Dot utilitzant la veu.
2. L'Amazon Echo Dot escolta la paraula "Alexa" i s'activa en el moment que la reconeix.
3. El servei de veu d'Alexa realitza el procés de comprensió de la parla (SLU), incloent el reconeixement de veu automatitzat (ASR), comprensió del llenguatge natural (NLU) i la síntesi de veu (TTS) que realitza la conversió de text a veu.
4. Alexa Skills Kit és l'eina on es creen les skills i permet una integració amb AWS Lambda. Com que la nostra skill és una custom skill tenim un model d'interacció personalitzat.
5. La funció de Lambda integrada a la skill és l'encarregada de processar les diferents sol·licituds que li envia el servei de veu d'Alexa en format JSON i retornar-li la resposta en el mateix format.
6. La còpia de la base de dades MySQL de l'Escola ens proporcionarà les dades necessàries per a respondre a les sol·licituds de l'usuari.
7. Amazon CloudWatch Monitoring ens mostra la monitorització de la nostra skill a temps real.

6.3. Creació de la skill

Posteriorment a la creació del compte de desenvolupador d'Amazon, acció definida a la *secció 5.1.1.*, es té accés a l'Alexa Developer Console on es crearà la custom skill [9].

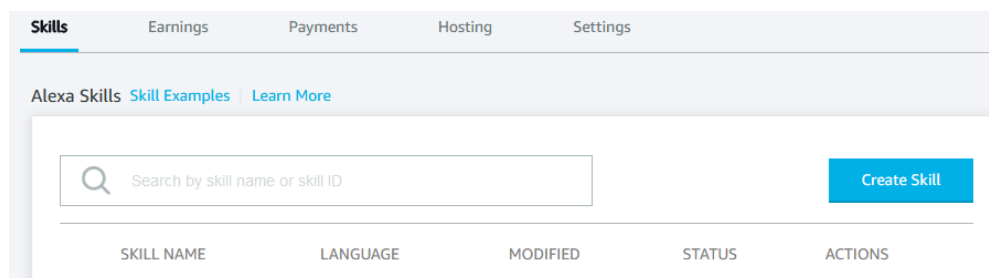


Figura 16. Pàgina principal d'Alexa Developer Console, creació skill

Al clicar sobre el botó 'Create skill' ens redirigeix cap a la pàgina de creació d'una skill. Ens demana el nom de la skill. Aquest nom no serà el nom d'invocació, simplement

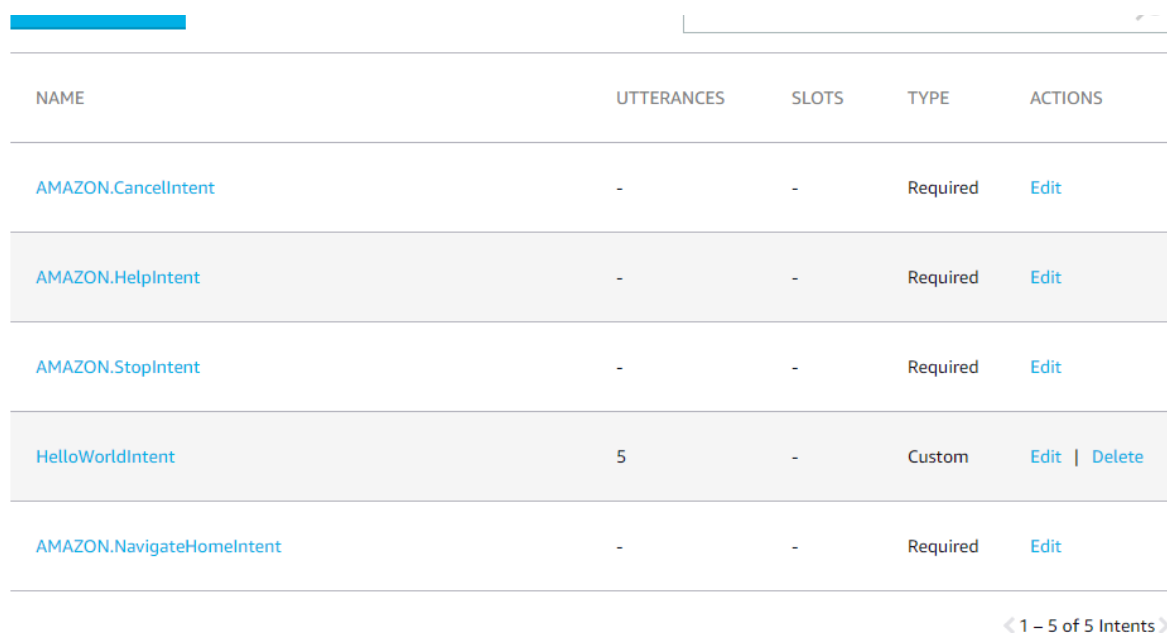
és el nom que es visualitzarà a la botiga de skills d'Amazon una vegada publicada. S'ha escollit *'Asistente EPSEVG'*. Seguidament s'ha d'escollir un llenguatge, aquest és el que s'utilitzarà pel reconeixement de veu, comprensió de la parla i síntesi. A dia d'avui no està disponible el català, així que s'ha escollit l'espanyol *'Spanish (ES)'*.

A continuació ens demana escollir el model d'interacció que s'afegirà a la skill, ens dona diferents opcions de tipus de skill. S'escolleix la *'Custom'* perquè, tal com s'ha comentat anteriorment (veure secció 3.4.1.) no té cap model d'interacció predefinit. S'haurà de crear posteriorment, això fa possible un major control sobre l'experiència d'usuari, és a dir, una interacció personalitzada amb l'usuari. Aquesta flexibilitat en la interacció és el que necessitem.

També s'ha d'escollir el back end que s'utilitzarà a la skill. En aquest cas, com s'utilitza una funció Lambda de la plataforma AWS, es selecciona *'Provision your own'*.

Per últim, ofereix l'opció d'afegir una plantilla a la nostra skill. En aquest cas s'escull *'Start from Scratch'* on es proporciona un model d'interacció amb els quatre intents obligatoris per a totes les skills sense cap sample utterance i un intent d'exemple anomenat *'HelloWorldIntent'* que conté cinc exemples de sample utterances tal i com es mostra a la *Figura 17*.

Intents



NAME	UTTERANCES	SLOTS	TYPE	ACTIONS
AMAZON.CancelIntent	-	-	Required	Edit
AMAZON.HelpIntent	-	-	Required	Edit
AMAZON.StopIntent	-	-	Required	Edit
HelloWorldIntent	5	-	Custom	Edit Delete
AMAZON.NavigateHomeIntent	-	-	Required	Edit

◀ 1 - 5 of 5 Intents ▶

Figura 17. Estat inicial dels intents del model d'interacció amb la plantilla Start from Scratch

6.4. Front end. Creació del model d'interacció

Gràcies al disseny previ de la skill, explicat a la *secció 4*, es pot implementar d'una manera més senzilla i clara el model d'interacció. Per a facilitar la comprensió només s'explicarà detalladament l'intent que gestiona les lectures de TFG a les *seccions 6.4.3 i 6.4.4*.

6.4.1. Nom d'invocació

El nom d'invocació de la skill és el nom amb el que es sol·licitarà a Alexa que obri la nostra skill. Ha d'estar format com a mínim de dues paraules, el que s'ha indicat en la skill és *'asistente epsevg'*. Per tant, els usuaris podran dir a Alexa "Alexa, *obre asistente epsevg*" per a poder accedir a la skill de l'Escola.

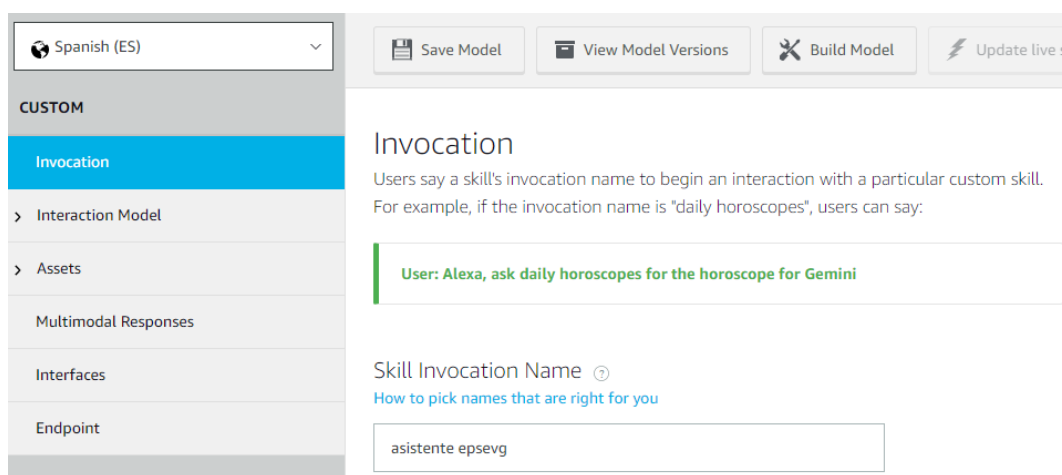


Figura 18. Nom d'invocació per la skill Asistente EPSEVG

6.4.2. Intents

Com s'ha definit anteriorment, els intents d'una skill són les sol·licituds a les que la skill pot donar resposta. Formats per sample utterances i de manera opcional, slots.

Alexa ofereix una sèrie d'intents estàndard per a accions generals com cancel·lar, demanar ajuda, detenir, entre d'altres. A més, per accions més complexes o pròpies, es poden crear intents propis.

Per a crear un nou intent, s'ha de clicar a *Interaction Model* al menú de l'esquerra, seguidament a *Intents* i fer clic al botó d'afegir intent. S'haurà de decidir si es vol crear un intent propi o escollir un dels estàndard que ofereix Alexa.

Add Intent

An intent represents an action that fulfills a user's spoken request. [Learn more](#) about intents.

Create custom intent [?]

Use an existing intent from Alexa's built-in library [?]

[Learn more](#) about using built-in intents.

Figura 19. Creació d'un nou intent

És important conèixer que en totes les skills hi ha 4 intents que són obligatoris i s'han d'indicar els sample utterances de cadascun. En el cas d'escollir la plantilla comentada anteriorment en el moment de crear una skill, si s'explora el model d'interacció es veu com s'han afegit aquests intents obligatoris, però sense cap sample utterance.

Intents obligatoris	Funció	Sample utterances
Amazon.CancelIntent	Cancel·lar alguna interacció en procés.	'pausa'
Amazon.HelpIntent	Sol·licitar ajuda a la skill.	'cómo funciona', 'ayuda', 'necesito ayuda', ...
Amazon.StopIntent	Detenir completament la skill.	'quita', 'apaga', 'apaga asistente epsevg', ...
Amazon.NavigateHomeIntent	Tornar a l'inici.	'inicio', 'reinicia', ...

Taula 2. Intents obligatoris creats a la skill Asistente EPSEVG

A la skill *Asistente EPSEVG* hi ha 13 intents en total, inclosos els 4 intents obligatoris. De la resta, 8 dels intents són propis i el darrer és un intent estàndard que ofereix Alexa anomenat *Amazon.RepeatIntent* que permet repetir la resposta que ha contestat Alexa.

Els intents personalitzats correspondrien als casos d'ús definits a la *secció 4*:

- Professors Intent: retorna la informació demanada d'un professor.
- Horari consultes Intent: retorna els horaris de consulta d'un professor.
- Departaments Intent: retorna la informació demanada d'un departament.
- Localització espais Intent: retorna la localització d'un aula.
- Calendari Intent: retorna la setmana i ordre del dia actual.
- Horari assignatura Intent: retorna els horaris d'una assignatura.
- Exàmens Intent : retorna la informació d'un examen d'una assignatura.
- Lectures TFG Intent: retorna la informació demanada sobre lectures TFG.

ProfesoresIntent	3	2	Custom
HorarioAsignaturaIntent	12	4	Custom
CalendarioIntent	2	-	Custom
DepartamentosIntent	9	2	Custom
LocalizacionEspaciosIntent	4	1	Custom
ExámenesIntent	2	2	Custom
LecturasTFGIntent	4	2	Custom
HorarioConsultasIntent	4	1	Custom

Figura 20. Intents personalitzats creats a la skill Asistente EPSEVG

A la *Figura 20* es poden veure els intents personalitzats creats a la skill, la primera columna indica el nom de l'intent, la segona el número de sample utterances, la tercera el número d'slots i l'última el tipus d'intent.

6.4.3. Sample utterances

Els sample utterances són les preguntes possibles que pot realitzar l'usuari a la skill per invocar a un intent determinat, es recomana afegir uns 10 depenent dels slots que tinguin i de la similitud que tinguin amb la resta de sample utterances dels altres intents. A més, s'ha de ser conscient que Alexa utilitza intel·ligència artificial, per tant, està en un constant entrenament, no necessita un gran número de sample utterances perquè anirà aprenent.

En el cas de la skill *Asistente EPSEVG* s'ha comprovat que si s'afegeixen més sample utterances en alguns intents pot ser contraproduent, ja que podrien arribar a semblar-se amb els d'altres intents, provocant que Alexa s'equivoci a l'hora de detectar quin intent s'invoca.

A l'hora de definir els sample utterances s'ha de tenir en compte que no poden contenir caràcters diferents a lletres a excepció de '{ i '}' que s'utilitzen per indicar un slot.

La *Figura 21* mostra els sample utterances definits per a l'intent que gestiona les lectures de TFG.



Figura 21. Sample utterances definits per l'intent LecturesTFG

En aquest intent s'ha decidit definir només 4 sample utterances per evitar conflictes amb els definits a d'altres intents.

Si es compara el dissenyat a la *secció 4.2.8.* pel cas d'ús de consulta d'horaris de TFG, podem comprovar que hem seguit els esdeveniments de sistema definits:

- Retorna els noms i cognoms dels alumnes que tenen prevista una lectura del TFG el dia actual. Correspon al següent sample utterance:
 - 'qué lecturas de tfg se realizan hoy'
- Retorna les dades sobre la lectura del TFG d'un alumne determinat. Correspon al següent sample utterance:
 - 'cuándo es la lectura del tfg del estudiante {xEstudiante}'
- Retorna la informació de les lectures de TFG on participa el dia actual un professor determinat. Correspon als següents sample utterances:
 - 'qué lecturas de tfg tiene el profesor {xProfesor}'
 - 'en qué lecturas de tfg participa el profesor {xProfesor}'

6.4.4. Slots

Els slots són arguments dins d'un sample utterance utilitzats per a poder conèixer la informació que busca l'usuari i posteriorment, ser utilitzats al back end de la skill per a obtenir les dades desitjades. Cal assignar-li un tipus d'slot a l'hora de crear-los, ja que aquest tipus és el que conté una sèrie de valors.

S'ha de diferenciar entre slot i tipus d'slot. El que afegim a un intent i per tant, s'utilitza als diferents sample utterances són slots, a aquests se'ls ha d'assignar un tipus d'slot que és el que conté els diferents valors possibles.

Per a crear un tipus d'slot, s'ha de clicar a *Assets* situat al menú de l'esquerra, després *Slot Types* i escollir si volem un slot propi o un dels que ofereix Alexa. Els tipus d'slot que ofereix Alexa contenen una sèrie de valors predefinits.

En aquesta skill, només s'han utilitzat tipus d'slots propis afegint valors manualment u obtinguts de la base de dades de l'Escola.

Create a custom slot type with values

Custom slot types with values define a representative list of possible values, IDs and synonyms.

Use an existing slot type from Alexa's built-in library [Ⓢ]

[Learn more about using built-in slot types.](#)

Figura 22. Creació d'un tipus d'slot

Ens els tipus d'slot propis, s'han d'afegir els diferents valors manualment i un per un o es poden importar d'un arxiu CSV seguint un format. Els valors dels slots tenen 3 camps:

- Valor: aquests valors són els que el servei de veu d'Alexa compararà amb els rebuts per l'usuari.
- ID: aquests valors són opcionals, si hi ha, han de ser únics per cada valor afegit. Quan els valors de l'slot són extrets d'una base de dades, és interessant guardar un valor que identifiqui el valor a la base de dades per a posteriors consultes al back end de la skill.
- Sinònims: aquests valors són opcionals, serien maneres possibles de dir el mateix valor. Si n'hi ha, també els utilitzarà per a comparar amb els rebuts per l'usuari.

Si es volen importar els valors des d'un arxiu CSV, s'ha de tenir en compte que la primera columna d'aquest correspondrà als valors, la segona als IDs i les posteriors als sinònims.

A la skill *Asistente EPSEVG* s'han creat 11 tipus d'slots propis, 2 d'ells utilitzats a l'intent que s'està detallant, lectures TFG. Aquests 2 tipus d'slots són *xEstudiante* i *xProfesor*.

Aquests 2 tipus d'slot, s'han creat i posteriorment s'han afegit els valors corresponents. Aquests valors s'han importat a través d'un arxiu CSV generat amb l'eina d'administració de base de dades, HeidiSQL, amb els resultats de la corresponent consulta a la còpia de la base de dades de l'Escola.

La consulta utilitzada per obtenir els valors del tipus d'slot *xProfesor*:

```
SELECT DISTINCT concat(nom, ' ', cognoms), dni, nom, cognoms FROM  
340_personal WHERE dni IN(SELECT DISTINCT dni FROM 340_personal_epsevg  
WHERE incid='A' and perfil='PDI');
```

Aquesta consulta retorna una primera columna amb els noms i cognoms dels professors que estan actius a l'Escola actualment, una segona columna amb un valor únic per cada professor, una tercera columna amb només els noms i una última columna amb només els cognoms.

La *Figura 23* mostra com obtenir els valors corresponents mitjançant la consulta explicada. Per generar l'arxiu CSV des de HeidiSQL, s'ha de fer clic on posa *Exportar filas de la cuadrícula* al desplegable que es mostra.

Assistent personal per l'EPSEVG basat en Alexa Miriam Martínez Izquierdo

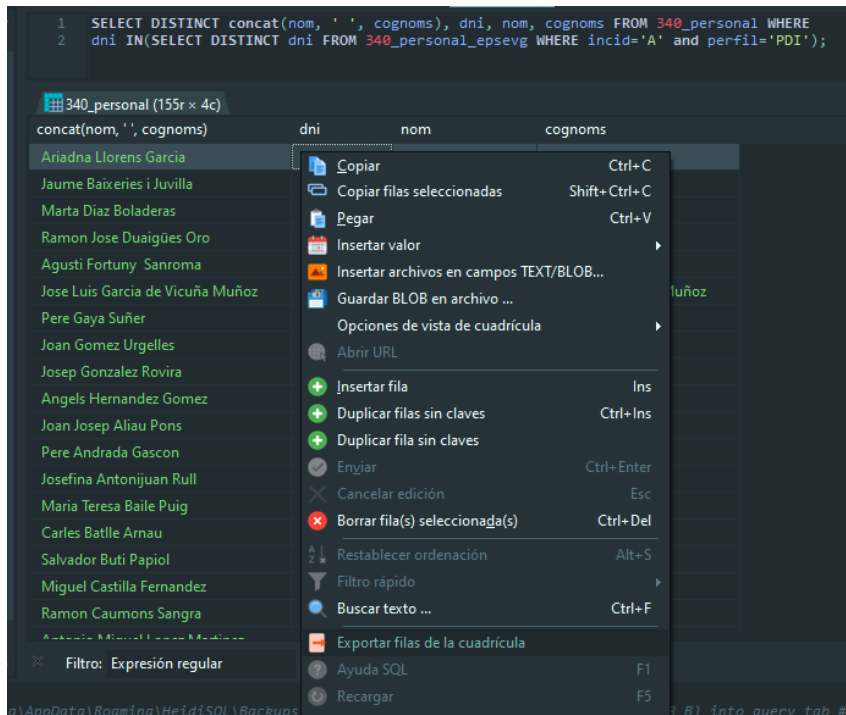


Figura 23. Obtenció dels valors corresponents pel tipus d'slot xProfesor de la skill Asistente EPSEVG utilitzant HeidiSQL

Una vegada obtingut l'arxiu CSV amb els valors corresponents, s'afegeix al tipus d'slot creat. Situat a la pàgina del tipus d'slot, es fa clic a *Bulk Edit* i s'obre la finestra mostrada a la Figura 24, es selecciona l'arxiu CSV on es tenen els valors.

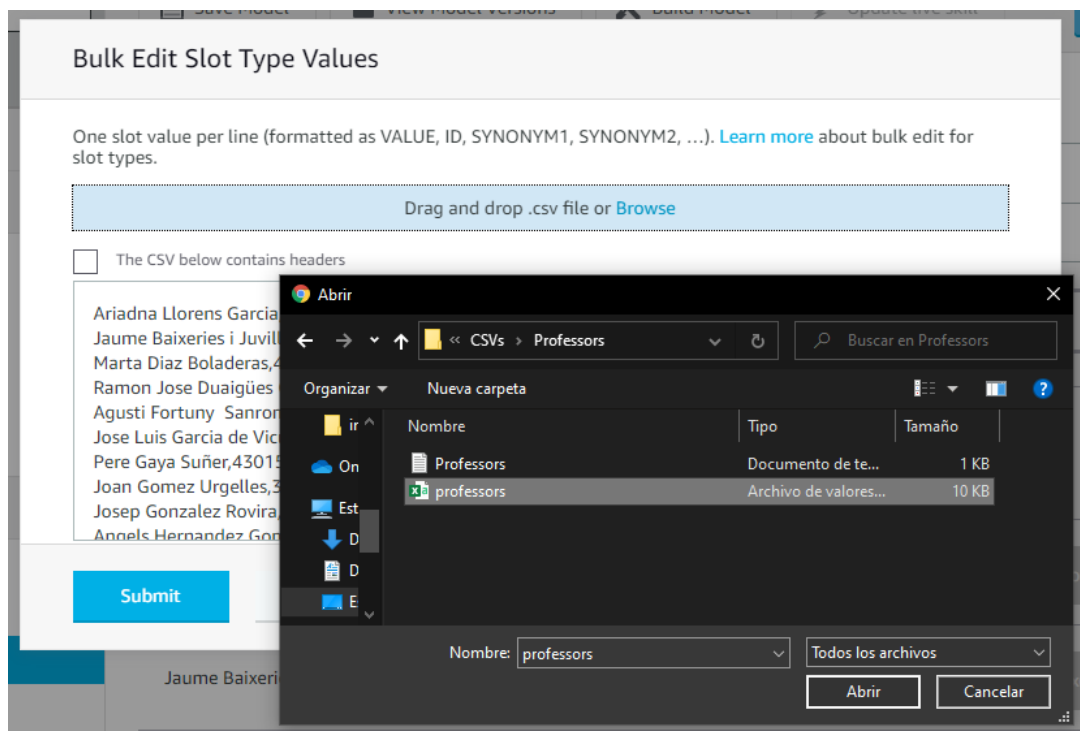


Figura 24. Importació dels valors d'un tipus d'slot a través d'un arxiu CSV

Es comprova, com es mostra a la *Figura 25*, que s'hagin importat correctament els valors al tipus d'slot, si n'hi ha algun, ens informará en el moment de fer *Build Model*, s'explicarà a la *secció 6.4.5*.

Slot Types / xProfesor

Custom slot types with values define a representative list of possible values, IDs and synonyms.

Slot Values (155) ? [Bulk Edit](#) [Export](#)

+

VALUE ?	ID (OPTIONAL) ?	SYNONYMS (OPTIONAL) ?			
Ariadna Llorens Garcia	522	Add synonym	+	Ariadna x Llorens Garcia x	
Jaume Baixeries i Juvilla	369	Add synonym	+	Jaume x Baixeries i Juvilla x	
Marta Diaz Boladeras	462	Add synonym	+	Marta x Diaz Boladeras x	
Ramon Jose Duaigües Oro	408	Add synonym	+	Ramon Jose x Duaigües Oro x	

Figura 25. Tipus d'slot xProfesor amb els valors importats correctament d'un arxiu CSV

Una vegada es tenen els tipus d'slot necessaris a un intent creats amb els valors corresponents, es creen els slots a l'intent. En aquest cas els slots *xProfesor* i *xEstudiante* de l'intent lectures TFG, els noms dels slots i els tipus d'slot no és necessari que siguin iguals. Posteriorment a la creació, s'assigna el tipus d'slot, el resultat es mostra a la *Figura 26*.

Intent Slots (2) ?

ORDER ?	NAME ?	SLOT TYPE ?	ACTIONS
1	● xEstudiante	xEstudiante	Edit Dialog Delete
2	● xProfesor	xProfesor	Edit Dialog Delete

Figura 26. Slots inclosos a l'intent Lectures TFG de la skill Asistente EPSEVG

6.4.5. Guardar i construir el model d'interacció

La pàgina de la skill que s'està desenvolupant, conté una barra de tasques del model d'interacció que ens permet guardar el model, visualitzar les diferents versions, construir-lo i avaluar-lo.

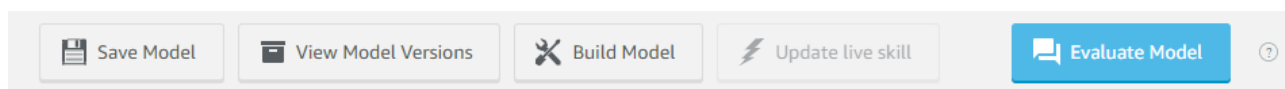


Figura 27. Barra de tasques del model d'interacció

En el moment de construir el model d'interacció actual, si n'hi ha algun error, els mostrarà. En el cas d'una construcció satisfactòria també mostrarà un avís.

La tasca per avaluar el model, avalua els diferents sample utterances sense necessitat d'haver construït encara el model.

6.5. Back end

De la mateixa manera que en el front end, gràcies al disseny previ de la skill a la secció 4, es poden implementar les funcionalitats de cada intent de manera més organitzada i clara. Aquest back end és l'encarregat de donar resposta, en format JSON, al servei de veu d'Alexa per fer la síntesi de veu. Per aconseguir les dades sol·licitades per l'usuari, es realitzen consultes a la base de dades. Un aspecte important recollit a l'Annex A és el tractament de les dades rebudes per una correcte síntesi de veu d'Alexa, equivalent a una bona experiència d'usuari.

Per a facilitar la comprensió i seguir amb el mateix intent explicat a la secció 6.4, es detallarà la funcionalitat implementada per l'intent Lectures TFG.

6.5.1. Base de dades MySQL

Com s'ha comentat anteriorment a la secció 6.1, la base de dades que s'utilitza en aquest projecte és una còpia de la base de dades MySQL de l'Escola amb la informació necessària actualitzada.

6.5.2. Funció Lambda

Com s'ha detallat anteriorment a la secció 5.2.1., la funció de Lambda creada, és el codi que s'executa quan és necessari a AWS Lambda. Està escrita en Javascript i configurada amb un temps d'execució Node.js. Aquesta funció permet gestionar tot el

codi del back end, realitzar les tasques corresponents a la sol·licitud de cada intent i donar resposta. El fitxer principal d'aquesta funció és *index.js*.

6.5.2.1. Dependències

La nostra funció Lambda necessita l'SDK d'Alexa per a poder enrutar les peticions que arriben en format JSON a través del servei d'Alexa cap a la funció controlador d'aquest intent que s'està demanant i poder executar el codi per així retornar la informació sol·licitada. Per altra banda, com s'obtenen les dades a través de la connexió a una base de dades MySQL, es necessita el mòdul de mysql per a poder realitzar les corresponents connexions i funcions pròpies de mysql.

Per resoldre aquestes dues dependències s'utilitzarà NPM, el gestor de paquets Javascript de Node.js. Aquesta eina s'utilitza des de la línia de comandes i permet tenir un control sobre els paquets o llibreries instal·lats en un projecte.

Per tant, primerament descarreguem el .zip del codi de la funció, es fa clic a *Acciones*, situat a la barra de tasques superior, posteriorment clic a *Exportar la funció*, que obrirà una finestra i clic *Descargar paquete de implementación*.

D'aquesta manera, està disponible localment el codi de la funció, es descomprimeix el .zip. Seguidament s'obre un terminal situat a la carpeta pare del nostre projecte de la funció Lambda. Executem les següents comandes per a la instal·lació dels dos mòduls explicats anteriorment:

```
npm install --save ask-sdk
```

```
npm install mysql
```

Per a la comprovació de la correcta instal·lació, obrim els arxius 'package.json' que correspon a la descripció en format JSON del nostre projecte funció Lambda. Es revisa que estiguin les següents dependències:

```
{
  "name": "asistenteepsevg",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "Miriam Martínez Izquierdo",
  "license": "ISC",
  "dependencies": {
    "ask-sdk": "^2.10.0",
    "mysql": "^2.18.1"
  }
}
```

Figura 28. Contingut de l'arxiu package.json de la funció Lambda

Es pot comprovar com a dependències surten ask-sdk i mysql, és a dir, ha anat satisfactòriament. Una vegada comprovat, es comprimeix i es torna a pujar el projecte a la funció Lambda, a l'apartat *Código de la función, Acciones, Cargar un archivo .zip*.

6.5.2.2. Connexió amb la skill *Asistente EPSEVG*

La connexió entre la skill i la funció de Lambda aconseguix que la funció de Lambda vinculada rebi les sol·licituds d'intents corresponents quan un usuari interactua amb la skill d'Alexa vinculada. La representació de la sol·licitud en format JSON creada pel servei de veu d'Alexa amb el model d'interacció definit permet obtenir paràmetres necessaris, com valors dels slots, per la lògica implementada a la funció Lambda i així aconseguir generar una resposta en format JSON.

Per a realitzar aquesta connexió amb la skill *Asistente EPSEVG*, s'ha d'afegir el trigger d'Alexa Skills Kit a la funció Lambda, en aquest cas, anomenada *asistenteEPSEVG* com es mostra a la *Figura 29*.

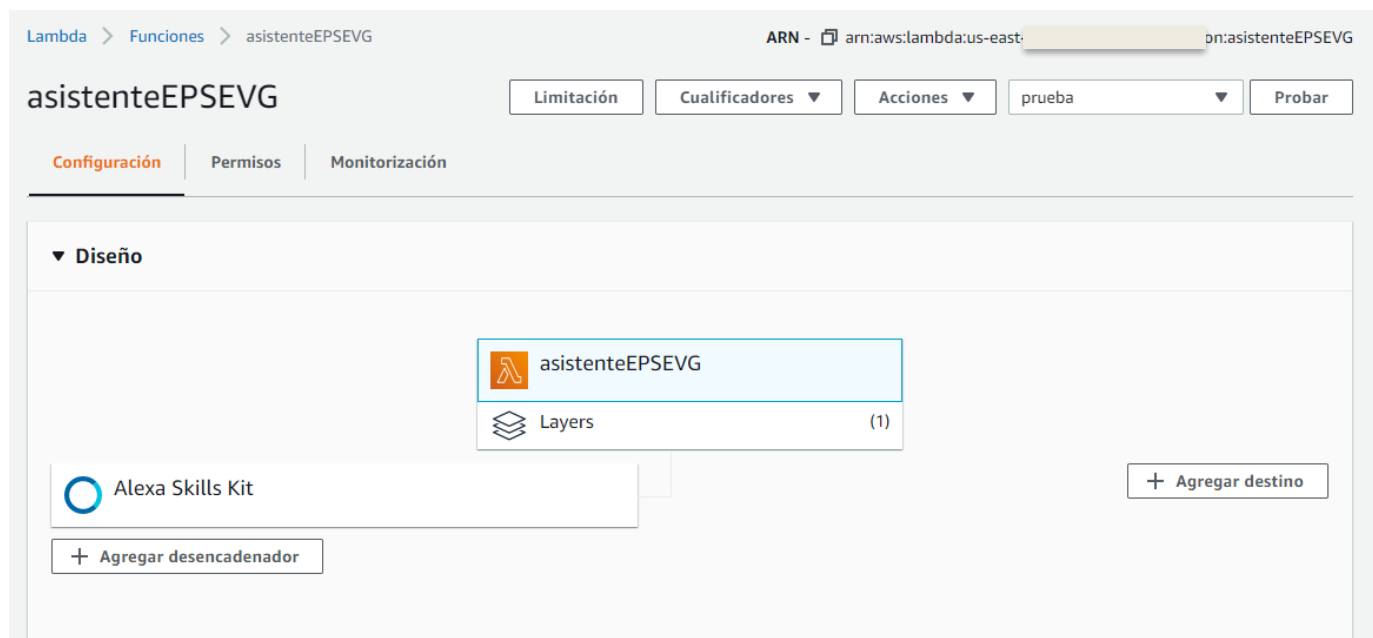


Figura 29. Dashboard de la funció Lambda *asistenteEPSEVG*

El següent pas és tornar a la Alexa Developer Console, clicar *Endpoint* al menú de l'esquerra i afegir l'ARN de la funció de Lambda que apareix a la *Figura 27* a la cantonada superior dreta a *Default Region* com es mostra a la *Figura 28*.

Service Endpoint Type

Select how you will host your skill's service endpoint. Best practices in choosing lambda regions. [Learn more here](#)

AWS Lambda ARN [?]
(Recommended)

Your Skill ID [?] amzn1.ask.skill.4caaddb4-4c78-
[Copy to Clipboard](#)

Default Region [?]
(Required) arn:aws:lambda:us-east-1:
:asistenteEF

Figura 30. Configuració endpoint de la skill Asistente EPSEVG

Per últim, tornem al Dashboard de la funció de Lambda, cliquem sobre el trigger d'Alexa Skills Kit que hem afegit per a configurar-lo. Afegim el ID de la nostra skill, apareix a la *Figura 30*.

Una vegada realitzats aquests tres passos, ja tenim la funció Lambda i la skill connectades.

6.5.2.1. Controladors d'intents

Els controladors d'intents, *handler intent*, són les funcions a les que l'SDK d'Alexa crida depenent de la sol·licitud d'intent arribada. Aquests controladors estan definits a l'arxiu *index.js*. Per cada intent definit al model d'interacció s'ha d'afegir el seu controlador en aquest arxiu.

```
exports.handler = Alexa.SkillBuilders.custom()
  .withSkillId("amzn1.ask.skill.4caaddb4-4c78-")
  .addRequestHandlers(
    LaunchRequestHandler,
    RepeatIntentHandler,
    HelpIntentHandler,
    CancelAndStopIntentHandler,
    SessionEndedRequestHandler,
    ProfesoresIntentHandler,
    DepartamentosIntentHandler,
    HorarioAsignaturaIntentHandler,
    ExamenesIntentHandler,
    HorarioConsultasIntentHandler,
    CalendarioIntentHandler,
    LocalizacionEspaciosIntentHandler,
    LecturasTFGIntentHandler
  )
  .addErrorHandlers(ErrorHandler)
  .lambda();
```

Figura 31. Llistat dels controladors de tots els intents creats al model d'interacció de la skill

6.5.2.3. Controlador de l'intent Lectures TFG

Per a una comprensió més fàcil del funcionament de la funció Lambda es planteja un exemple. L'usuari diu *'qué lecturas de tfg tiene el profesor Daniel Guasch'*, el servei de veu d'Alexa genera un arxiu de tipus JSON amb la descripció de la sol·licitud de l'intent Lectures TFG. Posteriorment, l'SDK gestiona aquest arxiu JSON i crida al corresponent controlador, en aquest cas, *LecturasTFGIntentHandler* i s'executa el codi corresponent, *Figura 32*.

```
const LecturasTFGIntentHandler = {
  canHandle(handlerInput){
    return handlerInput.requestEnvelope.request.type === 'IntentRequest'
      && handlerInput.requestEnvelope.request.intent.name === 'LecturasTFGIntent';
  },
  async handle(handlerInput){
    const currentIntent = handlerInput.requestEnvelope.request.intent;

    let _profesor = currentIntent.slots.xProfesor.resolutions;
    let _estudiant = currentIntent.slots.xEstudiante.resolutions;

    let data = new Date();
    data = data.getFullYear() + "-" + (data.getMonth()+1) + "-" + data.getDate() + " " + "00:00:00";

    if(_profesor == undefined && _estudiant == undefined) speechOutput = await mysqlHelper.getLecturasHoy(data);
    else{
      if(_profesor == undefined){
        if(_estudiant.resolutionsPerAuthority[0].status.code == "ER_SUCCESS_NO_MATCH")
          speechOutput = "Lo siento, no he encontrado ningún estudiante con ese nombre.";
        else{
          _estudiant = _estudiant.resolutionsPerAuthority[0].values[0].value;
          speechOutput = await mysqlHelper.getLecturasEstudiante(_estudiant);
        }
      }
      else{
        if(_profesor.resolutionsPerAuthority[0].status.code == "ER_SUCCESS_NO_MATCH")
          speechOutput = "Lo siento, no he encontrado ningún profesor con ese nombre.";
        else{
          _profesor = _profesor.resolutionsPerAuthority[0].values[0].value;
          speechOutput = await mysqlHelper.getLecturasProfesor(data, _profesor);
        }
      }
    }

    return handlerInput.responseBuilder
      .speak(speechOutput)
      .reprompt('¿Puedo ayudar en algo más?')
      .getResponse();
  }
};
```

Figura 32. Implementació del controlador de l'intent Lectures TFG

Donats els sample utterance definits per aquest intent, a la *secció 6.4.3.*, es realitza un retorn de dades diferents depenent de quin ha estat cridat per l'usuari. Per a saber quin ha estat cridat, s'han d'analitzar els valors dels slots recuperats per l'arxiu JSON de la petició que envia el servei d'Alexa, des del controlador de l'intent hi tenim accés.

Per tant, en aquest intent hi ha 3 possibilitats d'informació demanada, es pot analitzar a través dels valors dels slots rebuts:

- Té un slot del tipus *xProfesor* amb el nom del professor com a valor.
- Té un slot del tipus *xEstudiante* amb el nom de l'estudiant com a valor.
- No té cap slot.

Aquesta part d'analitzar els slots de l'arxiu JSON generat pel servei d'Alexa és la que s'ha implementat al controlador *LecturasTFGIntentHandle*.

Com es pot comprovar a la *Figura 33*, dels 2 slots possibles que té aquest intent, l'slot *xEstudiante* no té cap valor i l'slot *xProfesor* sí. Per generar l'arxiu JSON, Alexa busca si el valor escoltat coincideix amb algun valor dels tipus d'slot afegits al model d'interacció. Si coincideix amb algun, mostra el seu valor. En aquest cas, el nom del professor (Daniel Guasch Murillo) i el seu id.

Aquest fitxer JSON generat pel servei d'Alexa on detecta l'intent de lectures TFG, fa que l'SDK pugui cridar al controlador *LecturasTFGIntentHandler*. S'executa el codi, i per tant, agafa els valors dels slots obtinguts. En aquest cas, entraria a la condició on la variable *_estudiant* té un valor indefinit (perquè a JSON no té cap valor). Posteriorment, es recupera el valor de l'slot *xProfesor* (Daniel Guasch Murillo i el seu id). Això significa que existeix a l'Escola, i per tant, es crida a la funció *getLecturesProfesor(data, _profesor)*. Aquesta funció, passant-li com a paràmetres la data actual i el valor de l'slot *xProfesor* ens retornarà la informació demanada consultant a la base de dades.

En el cas de que el nom del professor no coincidís amb cap professor existent a l'Escola, per tant, que no existís com a valor a l'slot *xProfesor*. El servei de veu d'Alexa hagués retornat a l'arxiu JSON, una resolució de l'slot *xProfesor* amb codi d'estat "*ERR_SUCCESS_NO_MATCH*".

```
"intent": {
  "name": "LecturasTFGIntent",
  "confirmationStatus": "NONE",
  "slots": {
    "xProfesor": {
      "name": "xProfesor",
      "value": "daniel guasch",
      "resolutions": {
        "resolutionsPerAuthority": [
          {
            "authority": "amzn1.er-authority.echo-sdk.amzn1.ask.skil",
            "status": {
              "code": "ER_SUCCESS_MATCH"
            },
            "values": [
              {
                "value": {
                  "name": "Daniel Guasch Murillo",
                  "id": "397( )"
                }
              }
            ]
          }
        ]
      }
    }
  },
  "confirmationStatus": "NONE",
  "source": "USER",
  "slotValue": {
    "type": "Simple",
    "value": "daniel guasch",
    "resolutions": {
      "resolutionsPerAuthority": [
        {
          "authority": "amzn1.er-authority.echo-sdk.amzn1.ask.",
          "status": {
            "code": "ER_SUCCESS_MATCH"
          },
          "values": [
            {
              "value": {
                "name": "Daniel Guasch Murillo",
                "id": "397( )"
              }
            }
          ]
        }
      ]
    }
  }
},
"xEstudiante": {
  "name": "xEstudiante",
  "confirmationStatus": "NONE",
  "source": "USER"
}
```

Figura 33. Arxiu tipus JSON amb la descripció de la sol·licitud realitzada per l'usuari

Assistent personal per l'EPSEVG basat en Alexa
Miriam Martínez Izquierdo

La funció `getLecturesProfesor` es troba dins del directori `helpers`, a l'arxiu `mysqlHelper.js`. En aquest arxiu es defineixen totes les funcions necessàries per a establir una connexió amb la base de dades i obtenir la informació demanada a través de la consulta corresponent.

```
mysqlHelper.prototype.getLecturasProfesor = (data, profesor) => {
  return new Promise((resolve, reject) => {
    const con = mysql.createConnection(dades_bd);

    let sql = 'SELECT DISTINCT vw_membres_projectes_340.rol_projecte, vw_membres_projectes_340.rol_tribunal, vw_dades_estudiants_340.nom, vw_c
    let var_sql = [data, profesor.id];

    con.query(sql, var_sql, (err, result) => {
      con.destroy();
      if(err){
        console.log(err);
        return reject(err);
      }

      let speechOutput;

      if(result[0] == undefined) speechOutput = "Para hoy no tiene prevista ninguna lectura de TFG.";
      else{
        let n = 0;
        speechOutput = "Participas en la lectura de TFG ";
        while(result[n] != undefined){
          speechOutput += "del estudiante " + result[n].nom + " " + result[n].primer_cognom + " " + result[n].segon_cognom + " como ";

          //S'obté el rol amb el que participa a la lectura
          if(result[n].rol_projecte == "Director") speechOutput += "director a las " + result[n].hora_lectura;
          else if(result[n].rol_projecte == "Codirector") speechOutput += "codirector a las " + result[n].hora_lectura;
          else if(result[n].rol_tribunal == "President") speechOutput += "presidente a las " + result[n].hora_lectura;
          else if(result[n].rol_tribunal == "Secretari") speechOutput += "secretario a las " + result[n].hora_lectura;
          else if(result[n].rol_tribunal == "Vocal") speechOutput += "vocal a las " + result[n].hora_lectura;

          //Es formateja la variable que conté el codi de l'aula per a la correcta síntesi de veu d'Alexa
          let aula = result[n].lloc_lectura;
          if(aula.substr(0, 2) == 'VG'){
            let formatted = aula.substr(0, 1);
            let i = 1;
            while(isNaN(aula.substr(i, 1))){
              formatted += '.' + aula.substr(i, 1);
              ++i;
            }
            formatted += '.' + aula.substr(i, aula.length);

            speechOutput += " en el aula " + formatted;
          }else speechOutput += " de forma no presencial";

          if(result[n+1] == undefined) speechOutput += ". ";
          else if(result[n+2] == undefined) speechOutput += " y ";
          else speechOutput += ", ";

          ++n;
        }
      }

      resolve(speechOutput);
    })
  })
}
```

Figura 34. Implementació de la funció que retorna informació demanada de la base de dades

Per a retornar la resposta creada a la funció `getLecturesProfesor`, s'ha de gestionar el codi per a un funcionament síncron d'aquesta part, per tant, s'ha utilitzat les promeses i l'`async/await` de Javascript detallat a la secció 5.2.1.1.

A la variable `speechOutput` es on es guarda la resposta, cadenes de text, amb la que el servei d'Alexa realitzarà la síntesi de veu. Per una correcte síntesi de veu per part d'Alexa s'han de formatejar alguns valors rebuts amb la consulta de la base de dades. Com per exemple, el cas de retornar un número de dia de la setmana, una data del tipus *Date* entre d'altres.

7. VALIDACIÓ DE LA SKILL

La comprovació del correcte funcionament de la skill s'ha realitzat mitjançant el simulador que incorpora l'Alexa Developer Console i l'altaveu intel·ligent d'Amazon, l'Echo Dot de 3a generació proporcionat per l'Escola.

Per a poder utilitzar la skill a l'altaveu intel·ligent quan està en fase de desenvolupament, s'ha de vincular el dispositiu al mateix compte on s'està desenvolupant aquesta skill.

D'altra banda, el simulador que ofereix Alexa Developer Console mostra de manera visual la interacció entre l'usuari i Alexa. Es poden veure els arxius JSON que genera el servei d'Alexa amb la interacció de l'usuari i també l'arxiu JSON que rep com a resposta per part del back end desenvolupat. Per a iniciar aquest simulador, s'ha d'anar a la pàgina web de la skill i clicar a *Test* situat a la barra de tasques de la skill.

Seguint amb l'exemple explicat durant l'apartat 6 d'implementació, l'intent *LecturesTFG*, podem veure el que ens mostra el simulador d'Alexa. Per a realitzar aquesta prova, s'ha modificat manualment el codi per establir com dia actual el dia 05/02/2021. Si s'utilitzés el dia actual real Alexa només contestaria que no hi ha cap lectura prevista per aquest professor.

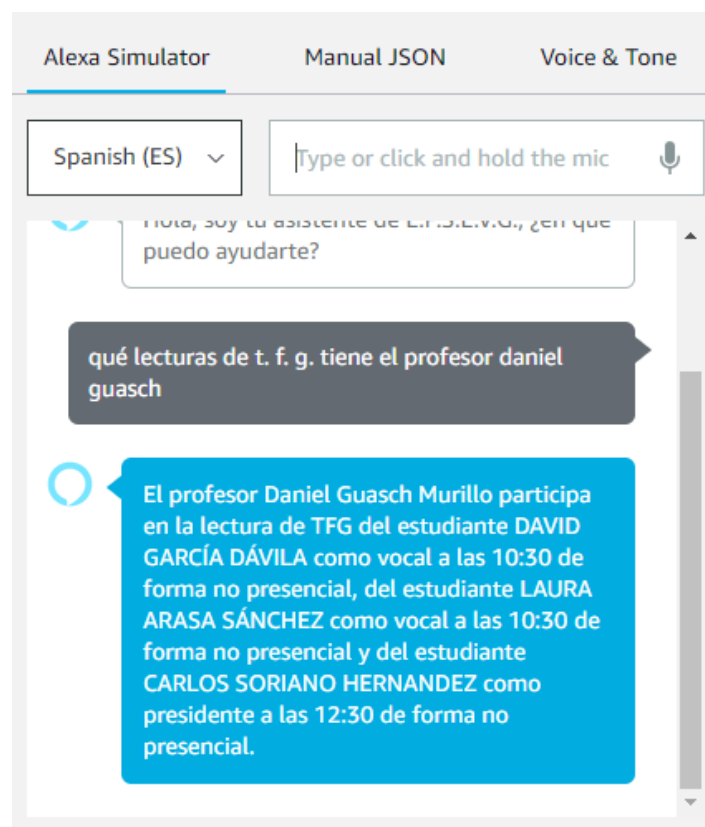


Figura 35. Flux de veu entre l'usuari i la skill

Com es pot veure a la *Figura 35*, el simulador mostra el flux de veu entre l'usuari i Alexa de manera visual, en format text, però també per veu. Això és útil per a realitzar comprovacions. Com per exemple, veure que entén Alexa quan es parla pel micròfon o per veure el redactat de la resposta que sintetitza Alexa.



El profesor Daniel Guasch Murillo participa en la lectura de TFG del estudiante DAVID GARCÍA DÁVILA a las 10:30 de forma no presencial y del estudiante CARLOS SORIANO HERNANDEZ como presidente a la

Skill Invocations | Viewing: 1 / 1

```
92  "name": "LecturasTFGIntent",
93  "confirmationStatus": "NONE",
94  "slots": {
95    "xProfesor": {
96      "name": "xProfesor",
97      "value": "daniel guasch",
98      "resolutions": {
99        "resolutionsPerAuthority": [
100       {
101         "authority": "amzn1.er-authority.echo-sdk.amzn1.ask.skill.4caaddb4-4c78-4c",
102         "status": {
103           "code": "ER_SUCCESS_MATCH"
104         },
105         "values": [
106           {
107             "value": {
108               "name": "Daniel Guasch Murillo",
109               "id": "397"
110             }
111           ]
112         }
113       }
114     ]
115   },
116   "confirmationStatus": "NONE",
117   "source": "USER",
118   "slotValue": {
119     "type": "Simple",
120     "value": "daniel guasch",
121     "resolutions": {
122       "resolutionsPerAuthority": [
123         {
124           "authority": "amzn1.er-authority.echo-sdk.amzn1.ask.skill.4caaddb4-4c",
125           "status": {
```

Figura 36. Arxiu JSON generat pel servei Alexa on es descriu la petició realitzada per l'usuari

Per altra banda, com es pot veure a la *Figura 36*, també ens mostra l'arxiu JSON que genera el servei d'Alexa amb la descripció de la sol·licitud rebuda per part de l'usuari. Aquí podem comprovar si detecta bé els valors dels slots, si en té la sol·licitud demanada i si detecta l'intent que correspon amb el sample utterance dit per l'usuari.

como vocal a las 10:30 de forma no presencial, del estudiante LAURA ARASA SÁNCHEZ como vocal a las 10:30 de forma no presencial.



```
1 {
2   "body": {
3     "version": "1.0",
4     "response": {
5       "outputSpeech": {
6         "type": "SSML",
7         "ssml": "<speak>El profesor Daniel Guasch Murillo participa en la lectura de TFG
8       },
9       "reprompt": {
10        "outputSpeech": {
11          "type": "SSML",
12          "ssml": "<speak>¿Puedo ayudar en algo más?</speak>"
13        }
14      },
15      "shouldEndSession": false,
16      "type": "_DEFAULT_RESPONSE"
17    },
18    "sessionAttributes": {},
19    "userAgent": "ask-node/2.10.1 Node/v12.20.1"
20  }
21 }
```

Figura 37. Arxiu JSON generat al back end que conté la resposta

Com es pot veure a la *Figura 37*, aquest simulador també ens permet visualitzar l'arxiu JSON que rep el servei d'Alexa com a resposta del nostre back end i que és el que sintetitzarà Alexa.

Finalment, també s'han validat tots els intents implementats mitjançant proves amb l'altaveu intel·ligent, Echo Dot [10]. També s'ha redactat un manual d'usuari amb les diferents preguntes que es poden realitzar per a demanar informació a la skill, *Annex B*.

Finalment, s'ha realitzat un vídeo [11] on es pot veure l'Echo Dot utilitzat i escoltar aquesta interacció entre l'usuari i Alexa amb la skill *Asistente EPSEVG*.

CONCLUSIONS

En aquest treball es podrien diferenciar cinc parts, cadascuna relacionada amb els objectius fixats en un principi.

En primer lloc, la realització d'un estudi sobre els assistents per veu, més concretament Alexa i la seva plataforma per a desenvolupar skills. Gràcies a la realització d'aquesta part s'ha pogut aprofundir en l'arquitectura d'aquesta plataforma, concretament en l'arquitectura i funcionament d'una custom skill. Part que s'ha intentat deixar reflectida durant l'elaboració del treball.

En segon lloc, la realització d'un estudi sobre les diferents eines que ofereix Amazon per al desenvolupament d'una skill. També es va realitzar una primera aproximació a les futures eines per al desenvolupament del back end de la skill.

En tercer lloc, la valoració de quines dades mostrades a la pàgina web de l'Escola eren més rellevants en aquest context d'Alexa. Una vegada decidit, realitzar una especificació amb els diferents casos d'ús i els esdeveniments de sistema que podria tractar cadascun. És a dir, dissenyar la skill.

En quart lloc, la decisió final de les eines a utilitzar per a la implementació i la posterior implementació de la skill. Una vegada comprovada la possible utilització d'una còpia de la base de dades de l'Escola per a l'obtenció de les dades, es va començar a implementar. Aquesta implementació ha estat complerta satisfactòriament.

Per últim, la validació de la skill. Per tant, la comprovació del correcte funcionament del desenvolupat durant el projecte. I a la vegada, la comprovació de la correcta aplicació dels coneixements sobre les eines utilitzades, l'arquitectura i funcionament de la plataforma Alexa.

En conclusió, es pot dir que s'han assolit els objectius plantejats inicialment de manera satisfactòria. Però evidentment el cicle de la skill no s'ha tancat, per tant seguidament s'exposaran les possibles futures tasques i millores a realitzar.

Primerament, la skill no ha estat publicada a la botiga de skills d'Amazon. Per tant, el següent pas seria publicar-la, pujar-la a producció [12]. Com que aquesta skill seria un producte oficial de l'Escola i s'ha d'elaborar una política de privacitat com a requisit, no s'ha realitzat aquest pas. Un aspecte important a tenir en compte és el manteniment d'aquesta skill una vegada publicada, concretament les possibles actualitzacions de les dades, detallat a l'*Annex C*.

En segon lloc, el model d'interacció és una part fonamental a la skill. Donat que el servei de veu d'Alexa interactua millor o pitjor en gran part en funció d'aquest model

Assistent personal per l'EPSEVG basat en Alexa
Miriam Martínez Izquierdo

creat de zero en el cas de la custom skill. Per tant, seria molt bona opció seguir treballant en ell per a poder millorar-lo.

Per últim, a les primeres fases d'estudi sobre el funcionament de les skills i diferents eines disponibles es va plantejar la idea de poder utilitzar una API que tractés les dades de la base de dades depenent de la necessitat i no obtenir-les directament. Aquest canvi seria una millora molt important en termes de seguretat, donat que, una API seria un extra de seguretat. Una API ofereix més seguretat que un accés directe a la base de dades.

AGRAÏMENTS

M'agradaria agrair al director del treball, Rafael Vidal Ferré i al codirector, Daniel Guasch Murillo, per haver donat els feedbacks constructius corresponents durant l'elaboració del treball i per transmetre una motivació constant. Això últim, tret característic dels dos i motiu d'haver contactat inicialment per la realització del meu treball final de grau.

Per altra banda, un agraïment especial a Víctor Agramunt, de serveis TIC de l'Escola. Sense ell no hagués estat possible la obtenció de les dades de la base de dades de l'Escola, aspecte essencial en l'elaboració d'aquest treball. Destacar també la seva gran implicació i entusiasme durant el desenvolupament del treball, la seva disposició a realitzar reunions per a resoldre dubtes sobre l'estructura de les dades entre d'altres.

Finalment, agrair a tots els professors i companys que he arribat a tenir durant tota l'etapa, perquè part de tot el que he après al final és producte de cadascun d'ells. Tant la base teòrica i pràctica adquirida, com tots els punts de vista i consells rebuts.

REFERÈNCIES

[1] Imatge del IBM Shoebox . Pàgina web IBM. [Consulta: 20/11/2020]

https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html

[2] Eric Hal Schwartz. (31 desembre 2019). Cronograma dels assistents per veu l'última dècada. Pàgina web Voicebot. [Consulta: 29/11/2020]

<https://voicebot.ai/2019/12/31/the-decade-of-voice-assistant-revolution/>

[3] Diagrames creats a través de Lucidchart [Consulta: 16/01/2021]

<https://www.lucidchart.com/>

[4] Bret Kinsella. (25 maig 2020). Gràfica de barres sobre les vendes d'altaveus intel·ligents. Pàgina web Voicebot. [Consulta: 29/11/2020]

<https://voicebot.ai/2020/05/25/global-smart-speaker-growth-cools-in-q1-as-pandemic-leads-to-declining-china-sales-amazon-retains-top-spot-says-strategy-analytics/>

[5] Requisits per la utilització d'un servei web propi. Pàgina web Amazon Developer. [Consulta: 12/09/2020]

<https://developer.amazon.com/en-US/docs/alexa/custom-skills/host-a-custom-skill-as-a-web-service.html>

[6] Diagrama d'interacció d'una custom skill. Pàgina web Amazon Developer. [Consulta: 12/09/2020]

<https://developer.amazon.com/en-US/docs/alexa/custom-skills/understanding-custom-skills.html>

[7] Diagrames UML creats a través de Creately. [Consulta: 17/10/2020]

<https://creately.com/>

[8] Registre Amazon Alexa. Pàgina web Amazon Alexa. [Consulta: 12/09/2020]

Assistent personal per l'EPSEVG basat en Alexa
Miriam Martínez Izquierdo

https://www.amazon.com/ap/register?clientContext=134-4971092-2989462&showRememberMe=true&openid.pape.max_auth_age=7200&openid.identity=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&marketPlaceId=ATVPDKIKX0DER&pageId=amzn_dante_us&openid.pape.preferred_auth_policies=Singlefactor&openid.return_to=https%3A%2F%2Fdeveloper.amazon.com%2Fes-MX%2Falexa%2Falexa-skills-kit&prevRID=F98Q8KR98B250X48T7NR&openid.assoc_handle=amzn_dante_us&openid.mode=checkid_setup&prepopulatedLoginId=&failedSignInCount=0&openid.claimed_id=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&openid.ns=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0

[9] Creació skill. Pàgina web Amazon Developer. [Consulta: 13/09/2020]

<https://developer.amazon.com/alexa/console/ask#/skill/create/>

[10] Proves validació skill. Pàgina web Amazon Developer. [Consulta: 02/01/2021]

<https://developer.amazon.com/en-US/docs/alexa/custom-skills/test-and-debug-a-custom-skill.html>

[11] Vídeo propi de la validació de la skill. Pàgina web Google Drive. [Consulta: 28/01/2021]

https://drive.google.com/file/d/1aXUFtk8aFCSaY5vm2_Hi1X4s6aSOonv5/view?usp=sharing

[12] Certificació i publicació de la skill. Pàgina web Amazon Developer. [Consulta: 03/01/2020]

<https://developer.amazon.com/en-US/docs/alexa/certify/certify-your-skill.html>

BIBLIOGRAFIA

Documentació Amazon Developer

<https://developer.amazon.com/es/documentation>

Documentació AWS Lambda

https://docs.aws.amazon.com/es_es/lambda/?id=docs_gateway

Documentació Javascript

<https://devdocs.io/javascript/>

Documentació MySQL

<https://dev.mysql.com/doc/>

Documentació Node.js

<https://nodejs.org/en/docs/>

ANNEX

A. Tractament de les dades per una correcta síntesi de veu d'Alexa

En aquest treball, les dades que necessita la skill les consulta a la còpia de la base de dades de l'Escola. Això vol dir que es recuperen amb el format i tipus que estigui a la base de dades. Hi ha casos en els que aquest format o aquest tipus de dades pot provocar que l'usuari no entengui què vol dir Alexa. A l'hora de realitzar la síntesi de veu per part del servei d'Alexa ho fa literal o no s'acaba d'entendre. Seguidament s'exposaran els casos trobats en aquest treball i l'opció escollida pel seu tractament.

Aquest tractament de les dades, i per tant, la correcta síntesi de veu d'Alexa, ajuda a donar una millor experiència d'usuari. Per aquesta raó, és un aspecte important a tenir en compte.

A.1. Número de telèfon

Quan es recupera el número de telèfon d'algun professor a la base de dades, el format és "931234567". Quan el servei de veu d'Alexa ho sintetitza l'interpreta com a un número enter i per tant diu el seu valor, no és una bona pràctica perquè l'usuari no espera aquest tipus de resposta. Per resoldre-ho, des de la mateixa funció del fitxer *mysqlHelper.js* de la funció Lambda, tractem aquesta dada rebuda i generem la resposta amb el valor formatejat.

```
telf = result[0].telfl;  
var formatted = telf.substr(0, 3) + '-' + telf.substr(3, 3) + '-' + telf.substr(6,4);  
speechOutput = "El teléfono de " + profesor.name + " es " + formatted;
```

Figura 38. Número de telèfon formatejat per una correcta síntesi de veu

En aquest cas, com mostra la *Figura 38*, el que es fa és donar-li un format "931-234-567", d'aquesta manera el servei d'Alexa ho sintetitza correctament i l'usuari obté la resposta desitjada.

A.2. Correu electrònic

El domini dels correus electrònics dels professors de l'Escola és "@upc.edu", Alexa ho sintetitza dient "upc" amb una lectura literal de la paraula. Per resoldre-ho es va decidir que lletregés tot el domini, no aconsegueix la manera humana en la que es diu, però l'usuari ho pot entendre.

```
res = result[0].email;
var n = res.search("@");
var formatted = res.substr(0, n+1);
var i = n+1;
while(i < res.length){
    if(res.substr(i, 1) != '.' && res.substr(i-1, 1) != '.') formatted += ' ' + res.substr(i, 1);
    else formatted += res.substr(i, 1);
    ++i;
}
console.log(formatted);
speechOutput = "El correo electrónico de " + profesor.name + " es " + formatted;
}
```

Figura 39. Correu electrònic formatejat per una correcta síntesi de veu

A.3. Codi de les aules

En el cas dels codis de les aules succeeix algo similar a l'apartat A.2.. Per exemple, el codi d'aula "VGA111", Alexa reproduiria literalment "VGA" i el número si que el tractaria de forma correcta. La solució és simple, que lletregi "VGA", això ho aconseguim formatejant el codi d'aula rebut, com es mostra a la Figura 40.

```
aula = result[i].codaul;
formatted = aula.substr(0, 1);
n = 1;
while(isNaN(aula.substr(n, 1))){
    formatted += ' ' + aula.substr(n, 1);
    ++n;
}
formatted += ' ' + aula.substr(n, aula.length);
```

Figura 40. Codi d'aula formatejat per una correcta síntesi de veu

A.4. Dies de la setmana i mesos

En el cas dels dies de la setmana, no dies d'una data, s'han de tenir en compte 2 casos que s'explicaran més endavant.

La base de dades assigna els valors de l'1 al 5 per dilluns a divendres respectivament, els valors de dissabte i diumenge no es contemplen perquè no hi ha activitat docent a l'Escola. Per una altra banda, al model d'interacció existeix un tipus d'slot personalitzat, *xDia*, amb els valors dels dies de la setmana i el seu respectiu ID. Els valors estan introduïts manualment seguint la lògica de la base de dades. Per tant,

els valors “els dilluns” a “els divendres” tenen assignats els IDs 1-5 respectivament i els valors “hoy”, “mañana” tenen assignats el 0 i el 6 respectivament.

Els 2 casos de tractament de dades són:

1. La petició d'un horari per un dia concret. L'usuari dirà el dia de la setmana del que vol saber l'horari, per tant això serà un slot del tipus *xDia*. En el cas que no sigui “hoy” o “mañana” no s'ha de fer res perquè coincideix amb les dades de la base de dades. En cas contrari, s'haurà de tractar al controlador de l'intent, al fitxer *index.js* (més detall a la secció 6.5.2.3.).

```
if(_dia.id == 0){
  var d = new Date();
  var n = d.getDay();

  if(semana == undefined) speechOutput += await mysqlHelper.getHorari(n, _grup, _assign);
  else speechOutput = await mysqlHelper.getHorari(n, _grup, _assign);
} else if(_dia.id == 6){
  var d = new Date();
  var n = d.getDay() +1;

  if(semana == undefined) speechOutput += await mysqlHelper.getHorari(n, _grup, _assign);
  else speechOutput = await mysqlHelper.getHorari(n, _grup, _assign);
```

Figura 41. Obtenció del dia de la setmana dels valors “hoy” o “mañana” del tipus d'slot *xDia*

A Javascript s'obté la data actual amb “new Date()”. A més, té una funció on retorna el dia de la setmana (1-7, dilluns-diumenge), això vol dir que en el cas de ser el valor 0 (“hoy”), enviarem a la funció que consulta a la base de dades el valor del dia actual. En cas de ser 6 (“mañana”), enviarem el dia actual incrementant 1.

2. Quan es consulten els horaris d'una assignatura, la base de dades et retorna el dia que es realitza la classe amb aquest format numèric. Per a la correcta síntesi de veu d'Alexa, s'ha de tractar i convertir en la paraula adequada, resultat amb el codi que mostra la *Figura 42*.

```
var dia_nom = 0;
if(result[i].dia != dia_nom){
  if(result[i].dia == 1) speechOutput += "Los lunes, ";
  else if(result[i].dia == 2) speechOutput += "Los martes, ";
  else if(result[i].dia == 3) speechOutput += "Los miércoles, ";
  else if(result[i].dia == 4) speechOutput += "Los jueves, ";
  else if(result[i].dia == 5) speechOutput += "Los viernes, ";
  dia_nom = result[i].dia;
}
```

Figura 42. Tractament del dia de la setmana retornat per la BD per una correcta síntesi de veu

En el cas dels mesos, succeeix algo semblant amb el cas 2 dels dies. Quan es demana alguna data, la base de dades retorna un tipus *date*, és a dir "0000-00-00 00:00:0000". El dia es pot recuperar sense cap problema i Alexa ho sintetitzarà bé, però el mes és interessant que no el sintetitzi en format número. Aquest canvi és senzill de realitzar, només s'ha d'analitzar quin número de mes és i donar la paraula corresponent com a resposta, a la *Figura 43*. Aquest canvi, ajuda a tenir una millor experiència d'usuari, en lloc de dir "el dia 5 del mes 2", Alexa dirà "el dia 5 del mes de febrer".

```
let data = result[0].fechaini;
let dia = data.getDate();
let mes = data.getMonth()+1;
if(mes == 01) mes = "enero";
else if(mes == 02) mes = "febrero";
else if(mes == 03) mes = "marzo";
else if(mes == 04) mes = "abril";
else if(mes == 05) mes = "mayo";
else if(mes == 06) mes = "junio";
else if(mes == 07) mes = "julio";
else if(mes == 08) mes = "agosto";
else if(mes == 09) mes = "septiembre";
else if(mes == 10) mes = "octubre";
else if(mes == 11) mes = "noviembre";
else if(mes == 12) mes = "diciembre";

speechOutput = "El examen " + examen + " de " +
  asignatura.name + " es el dia " + dia + " de "
  + mes + " a las " + result[0].h_i + " en el aula "
  + result[0].codaul + ". ";
```

Figura 43. Obtenció de la paraula corresponent al mes donat una data (tipus date)

A.5. Hores amb sistema horari de 24h

La base de dades de l'Escola utilitza el sistema horari de 24h. Això provoca que si retornem directament aquest valor a Alexa, les hores posteriors a les 12, la resposta sintetitzada pel servei de veu d'Alexa serà "a las 17" en comptes de "a les 5 de la tarda". El que es realitza és una resta (- 12) a les hores i també s'afegeix a la resposta "per la tarda"/"de la tarda", el codi implementat es pot veure a la *Figura 44*.

```
let hora = undefined;
if(result[n].hora_lectura.substr(0, 2) > 13){
    hora = result[n].hora_lectura.substr(0, 2) - 12 +
    result[n].hora_lectura.substr(2, result[n].hora_lectura.length);
}

if(hora == undefined)    speechOutput += result[n].hora_lectura;
else speechOutput += hora + " de la tarde";
```

Figura 44. Tractament del sistema horari 24h per una correcta síntesi d'Alexa

B. Manual d'usuari

Com s'ha comentat a la memòria, la skill *Asistente EPSEVG* té implementats 13 intents, que són les diferents peticions que pot resoldre la skill. En aquest manual es pretén donar a conèixer com invocar a cadascun d'aquests intents per a obtenir la informació desitjada.

Dividirem aquest manual en diferents seccions depenent de la informació que es vol demanar.

B.1. Informació sobre professors

La informació sobre els professors que et pot retornar la skill són el correu electrònic, el número de telèfon, el seu despatx, el departament al que pertany i el seu horari de consultes actual.

- Per conèixer el número de telèfon:

¿Cuál es el número de teléfono de NombreProfesor?

- Per conèixer el correu electrònic:

¿Cuál es el correo electrónico de NombreProfesor?

- Per conèixer el despatx:

¿Cuál es el despacho de NombreProfesor?

¿En qué despacho está NombreProfesor?

- Per conèixer el departament al que pertany:

¿En qué departamento está NombreProfesor?

¿A qué departamento pertenece NombreProfesor?

¿Cuál es el departamento de NombreProfesor?

- Per conèixer l'horari de consultes actual:

¿Cuál es el horario de consultas de NombreProfesor?

¿Cuándo tiene consultas NombreProfesor?

¿Cuándo tiene consultas el profesor NombreProfesor?

¿Cuál es el horario de consultas del profesor NombreProfesor?

B.2. Informació sobre assignatures

La informació sobre les assignatures que pot retornar la skill són l'horari complet d'una assignatura el quadrimestre actual, l'horari d'un dia de la setmana concret i l'horari complet d'un grup en concret. Per altra banda també pot retornar la data, hora i aula dels examens, parcial, final o reavaluació d'una assignatura.

- Per conèixer l'horari complet d'una assignatura:

¿Cuándo toca clase de NombreAsignatura?

¿Cuál es el horario de la asignatura NombreAsignatura?

¿Cuál es el horario de NombreAsignatura?

¿Cuándo y dónde toca clase de NombreAsignatura?

- Per conèixer l'horari d'una assignatura un dia concret de la setmana:

¿Cuándo toca clase de NombreAsignatura NombreDia?

¿Cuándo toca NombreDia clase de NombreAsignatura?

¿Cuál es el horario de la asignatura NombreAsignatura NombreDia?

*NombreDia = los viernes, hoy, mañana, los lunes...

- Per conèixer l'horari d'una assignatura d'un grup concret de classe:

¿Cuándo toca clase de NombreAsignatura del grupo CódigoGrupo?

- Per conèixer la informació sobre el parcial d'una assignatura:

¿Cuándo es el parcial de NombreAsignatura?

¿Cuándo es el examen parcial de NombreAsignatura?

- Per conèixer la informació sobre el final d'una assignatura:

¿Cuándo es el parcial de NombreAsignatura?

¿Cuándo es el examen parcial de NombreAsignatura?

- Per conèixer la informació sobre la revaluació d'una assignatura:

¿Cuándo es el examen de reevaluación de NombreAsignatura?

B.3. Informació sobre calendari

La informació sobre el calendari que pot retornar la skill és el número de setmana i ordre del dia actual.

¿Qué semana es?

¿En qué semana estamos?

B.4. Informació sobre la localització d'aules

La informació sobre la localització d'aules que pot retornar la skill és l'edifici i planta on es troba l'aula. Si estàs a l'edifici corresponent, t'ho informa.

¿En qué edificio está CódigoAula?

¿Dónde está CódigoAula?

¿Dónde está la clase CódigoAula?

¿Dónde está el aula CódigoAula?

B.5. Informació sobre les lectures de TFG actuals

La informació sobre les lectures de TFG que pot retornar la skill és la data, hora, lloc, director i codirector, membres del tribunal de la lectura d'un estudiant. Per altra banda, pot retornar totes les lectures que té previstes un professor el dia actual amb la data, hora, lloc, nom de l'estudiant del TFG i rol amb el què participa el professor. Per últim, també es pot consultar els noms dels estudiants que tenen lectura de TFG el dia actual.

- Per conèixer la informació sobre la lectura de TFG d'un estudiant:

¿Cuándo es la lectura del tfg del estudiante NombreEstudiante?

- Per conèixer la informació sobre les lectures previstes el dia actual d'un professor:

¿En qué lecturas de tfg participa el profesor NombreProfesor?

¿Qué lecturas de tfg tiene el profesor NombreProfesor?

- Per conèixer els noms dels estudiants que tenen la lectura de TFG el dia actual:

¿Qué lecturas de tfg se realizan hoy?

B.6. Informació sobre departaments

La informació sobre els departaments de l'Escola que pot retornar la skill és el nom complet del departament donat el codi, el codi donat el nom complet i el responsable de l'Escola en aquest departament.

- Per conèixer el nom complet d'un departament:

¿Cuál es el departamento CódigoDepartamento?

- Per conèixer el codi d'un departament:

¿Cuál es el código del departamento NombreDepartamento?

¿Cuál es el código de NombreDepartamento?

- Per conèixer el nom del responsable d'un departament:

¿Quién es el responsable del departamento CódigoDepartamento?

¿Quién es el jefe del departamento de NombreDepartamento?

¿Quién es el responsable del departamento de NombreDepartamento?

B.7. Peticions genèriques

Hi ha una sèrie de peticions genèriques que es poden realitzar, com es mostra a la secció 6.4.2. Aquestes peticions són sortir de la skill, tallar o repetir la resposta reproduïda per Alexa i rebre ajuda sobre la informació que ofereix.

- Per sortir de la skill: “apaga”
- Tallar la resposta: “pausa”
- Per repetir la resposta: “repite”
- Per demanar ajuda: “ayuda”

C. Manual per l'actualització de dades

Poden aparèixer casos on es necessita actualitzar dades estàtiques a la skill. Aquestes dades poden ser, el quadrimestre i any actuals, si no s'actualitzen aquestes variables sempre retornaria les dades del quadrimestre 1 de l'any 2020, l'actual a la base de dades en el moment de realitzar la skill. Altres dades estàtiques de la skill són els diferents valors dels tipus d'slot que potser s'han de canviar durant el quadrimestre actual.

Per tant, analitzem què s'ha de fer en cas d'haver d'actualitzar diferents dades.

C.1. Actualitzacions en qualsevol període de curs

C.1.1. Actualització professors

Es pot produir el cas en què un professor es doni de baixa o alta a l'Escola. En aquest cas, s'hauran d'actualitzar els valors del tipus d'slot *xProfessor*.

Per a dur a terme aquesta actualització s'haurà de generar un arxiu CSV amb els resultats obtinguts mitjançant la següent consulta a la base de dades de l'Escola:

Assistent personal per l'EPSEVG basat en Alexa
Miriam Martínez Izquierdo

```
SELECT DISTINCT concat(nom, ' ', cognoms), dni, nom, cognoms FROM 340_personal  
WHERE dni IN(SELECT DISTINCT dni FROM 340_personal_epsevg WHERE incid='A' and  
perfil='PDI');
```

El que conté aquest CSV és el nom i cognoms del professor a la primera columna, l'identificador d'aquest a la segona columna, i les darreres columnes serien només el nom i a l'última només els cognoms. El procediment a seguir per importar els valors del CSV al tipus d'slot corresponent està explicat a la memòria a la [secció 6.4.4](#).

D'aquesta manera ens assegurem que s'han actualitzat tots els possibles canvis de la gestió de professors i ara només estan com a valors els professors actius actualment.

En el cas de què la modificació sigui mínima i estigui controlada, es pot realitzar el canvi manualment. És a dir, afegir els valors manualment en el cas d'un alta o en el cas de baixa, cercar el professor en qüestió i eliminar-lo.

Slot Types / xProfesor

Custom slot types with values define a representative list of possible values, IDs and synonyms.

Slot Values (155) ⓘ

 Bulk Edit  Export

Search 




VALUE ⓘ	ID (OPTIONAL) ⓘ	SYNONYMS (OPTIONAL) ⓘ			
Ariadna Llorens Garcia	522 <input type="text"/>	Add synonym	+	Ariadna x Llorens Garcia x	
Jaume Baixeries i Juvilla	369 <input type="text"/>	Add synonym	+	Jaume x Baixeries i Juvilla x	
Marta Diaz Boladeras	462 <input type="text"/>	Add synonym	+	Marta x Diaz Boladeras x	

Figura 45. Valors del tipus d'slot xProfesor

En cas de baixa d'un professor, es cerca aquest mitjançant la barra de cerca i es fa clic a l'icona del contenidor. Altrament, en el cas d'alta s'introdueix el nom i cognoms a la capsa de text i es fa clic sobre la icona d'afegir, llavors s'indica l'identificador d'aquest i com a sinònims el nom només, un altre amb els cognoms només.

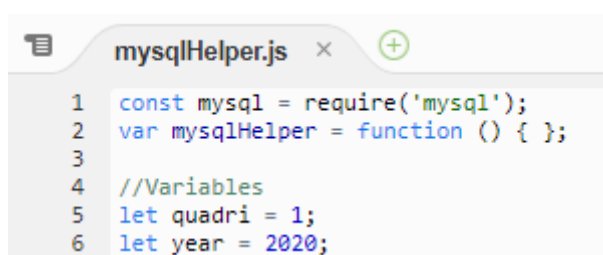
C.1.2. Actualització departaments

Si hi hagués algun canvi puntual en el nom d'un departament per exemple, es podria modificar manualment accedint a la pàgina web del tipus d'slot *xDepartamento*, d'igual manera que es mostra a la *Figura 45* pel tipus d'slot *xProfesor* a la *secció C.1.1*.

C.2. Actualitzacions per canvi de quadrimestre

Un moment d'actualització crític és quan es canvia de quadrimestre, perquè si no es canvien les següents variables del back end, seguiria retornant tota la informació del quadrimestre anterior.

Ens situem al back end de la skill, concretament a l'arxiu dins del projecte de la funció Lambda i del directori helpers, *mysqlHelper.js*. Una vegada en aquest arxiu, només s'han de modificar els següents valors de les variables *quadri* i *year* pels actuals.



```
mysqlHelper.js x (+)
1  const mysql = require('mysql');
2  var mysqlHelper = function () { };
3
4  //Variables
5  let quadri = 1;
6  let year = 2020;
```

Figura 46. Variables any del curs i quadrimestre

Aquests valors de les variables són els que s'utilitzen per a realitzar les diferents consultes a la base de dades tractades a totes les funcions implementades en aquest arxiu. Per tant, no s'ha d'anar funció per funció canviant-ho, simplement es modifiquen aquestes variables.

C.2.1. Actualització assignatures

Al produir-se un canvi de quadrimestre, les assignatures que teníem com a valor al tipus d'slot *xAssignatura* ara canvien. Per tant, s'ha de generar l'arxiu CSV amb els resultats obtinguts mitjançant la següent consulta canviant *any_curs* i *quadri_actual* pels valors corresponents al curs i quadrimestre actuals.

```
SELECT DISTINCT left(sigles_ud, 4), codi_upc_ud, nom, LOWER(sigles_ud) FROM
vw_unitats_docents_pro_340 WHERE codi_upc_ud IN (SELECT codass FROM
340_assigany WHERE anyass=any_curs AND quaass=quadri_actual);
```

Una vegada generat l'arxiu CSV, s'han d'importar aquests valors al tipus d'slot *xAssignatura*, com està explicat a la memòria a la *secció 6.4.4*.

C.2.2. Actualització grups

Al produir-se un canvi de quadrimestre, els diferents codis de grup de totes les assignatures de l'Escola poden canviar. Per aquest motiu, s'ha de generar un arxiu CSV amb els resultats obtinguts mitjançant la següent consulta canviant any_curs i quadri_actual pels valors corresponents al curs i quadrimestre actuals.

```
SELECT DISTINCT codgrup FROM 340_horari WHERE anyaca=any_curs AND  
quadri= quadri_actual;
```

Una vegada generat l'arxiu CSV, s'han d'importar aquests valors al tipus d'slot *xGrup*, com està explicat a la memòria a la *secció 6.4.4*.

C.2.3. Actualització estudiants

Els estudiants que estan al tipus d'slot *xEstudiante* només són els que tenen prevista una lectura de TFG el quadrimestre actual. Per tant, al canviar de quadrimestre, s'han d'actualitzar aquests valors. S'ha de generar un arxiu CSV amb els resultats obtinguts mitjançant la següent consulta a la base de dades canviant any_curs i any_lectura pels anys corresponents:

```
SELECT DISTINCT concat(nom, ' ', primer_cognom), codi_estudiant, CONCAT(nom,  
' ', primer_cognom, ' ', segon_cognom) FROM vw_dades_estudiants_340 WHERE  
codi_estudiant IN(SELECT DISTINCT codi_estudiant FROM vw_projectes_340  
WHERE curs_lectura= any_curs data_lectura LIKE '% any_lectura %');
```

Una vegada generat l'arxiu CSV, s'han d'importar aquests valors al tipus d'slot *xEstudiante*, com està explicat a la memòria a la *secció 6.4.4*.