Alex Moriana Betran

Comparativa de diferents mètodes de programar una skill per a la Alexa: Sleeping Helper

TREBALL DE FI DE GRAU

dirigit per Maria dels Àngels Moncusí Mercadé

Grau d'Enginyeria Informàtica

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Resum.

Aquest projecte té com a objectiu comparar i fer l'estudi sobre la creació d'una Skill d'Alexa mitjançant dos mètodes diferents per a crear-la.

Per tal de fer una bona comparació es desenvolupa una Skill d'Alexa amb la seva pròpia SDK anomenada **sleeping helper** que recull dades de l'usuari per seguir els patrons de comportament relacionats amb el son. Aquesta Skill es desenvolupa utilitzant Python i aprofita la base de dades DynamoDB per emmagatzemar les dades de l'usuari. A més a més, té un potencial considerable per esdevenir una Skill altament funcional i beneficiosa, ja que podria abordar una àmplia gamma de necessitats de l'usuari i contribuir a millorar la qualitat de vida ajudant als problemes relacionats amb la qualitat del son i el seu impacte en la desorganització com la manca de productivitat i el benestar general.

La mateixa Skill s'implementarà també amb la plataforma Voiceflow, una plataforma originalment feta per a persones que no saben programar gens o tinguin un nivell baix de programació. Aquesta plataforma ha ajudat molt al desenvolupament de moltes noves aplicacions de veu per Alexa i l'Assistent de Google.

En resum, aquest treball és un estudi de la creació d'una nova Skill d'Alexa, **sleeping helper** que té com a objectiu millorar la qualitat del son i el benestar general de l'usuari mitjançant el seguiment dels hàbits de son i l'alerta en cas de mals hàbits de son. S'ha proposat desenvolupar aquesta Skill de dues formes diferents i comparar-les deixant aquesta documentació com ajuda per als següents programadors júnior d'Alexa.

Resumen.

Este proyecto tiene como objetivo comparar y hacer el estudio sobre la creación de una Skill de Alexa mediante dos métodos diferentes para crearla.

Para hacer una buena comparación se desarrolla una Skill de Alexa con su propia SDK llamada **sleeping helper** que recoge datos del usuario para seguir los patrones de comportamiento relacionados con el sueño. Esta Skill se desarrolla utilizando Python y aprovecha la base de datos DynamoDB para almacenar los datos del usuario. Además, tiene un potencial considerable por devenir una Skill altamente funcional y beneficiosa, puesto que podría abordar una amplia gama de necesidades del usuario y contribuir a mejorar la calidad de vida ayudando a los problemas relacionados con la calidad del sueño y su impacto en la desorganización como la carencia de productividad y el bienestar general.

La misma Skill se implementará también con la plataforma Voiceflow, una plataforma originalmente hecha para personas que no saben programar nada o tengan un nivel bajo de programación. Esta plataforma ha ayudado mucho al desarrollo de muchas nuevas aplicaciones de voz por Alexa y el Asistente de Google.

En resumen, este trabajo es un estudio de la creación de una nueva Skill de Alexa, **sleeping helper** que tiene como objetivo mejorar la calidad del sueño y el bienestar general del usuario mediante el seguimiento de los hábitos de sueño

y lo alerta en caso de malos hábitos de sueño. Se ha propuesto desarrollar esta Skill de dos formas diferentes y compararlas dejando esta documentación como ayuda para los siguientes programadores júnior de Alexa.

Abstract.

This project aims to compare and carry out the study on the creation of an Alexa Skill using two different methods to create it.

In order to make a good comparison, an Alexa Skill is developed with its own SDK called **sleeping helper** that collects user data to follow sleep-related behavior patterns. This Skill is developed using Python and takes advantage of the DynamoDB database to store user data. In addition, it has considerable potential to become a highly functional and beneficial Skill, as it could address a wide range of user needs and contribute to improving quality of life by helping problems related to sleep quality and its impact on disorganization such as lack of productivity and general well-being.

The Skill itself will also be implemented with the Voiceflow platform, a platform originally made for people who do not know how to program anything or have a low level of programming. This platform has greatly helped the development of many new voice applications for Alexa and the Google Assistant.

In summary, this work is a study of the creation of a new Alexa Skill, **sleeping helper** that aims to improve sleep quality and the general well-being of the user by tracking sleep habits and alerting in case of bad sleep habits. It has been proposed to develop this Skill in two different ways and compare them leaving this documentation as a help for the next Alexa junior programmers.

Índex

Índex de figures

Índex de codi

1 Introducció

Aquest treball consisteix en l'estudi de dos possibles mètodes de desenvolupament que s'ha de seguir per a l'implementació d'una Skill per a l'assistent intel·ligent d'Amazon anomenada Alexa. S'utilitzarà dues formes per a crear un prototip de Skill, ja que fer una Skill complexa no és la finalitat d'aquest treball. Aquesta Skill pot donar peu a una Skill més potent que serveixi d'assistent virtual i calendari.

La Skill que s'implementarà no només recopilarà i emmagatzemarà dades rellevants sobre els hàbits de son dels usuaris, sinó que també proporcionarà consells personalitzats a través d'una interfície de veu. A més a més, el projecte té la intenció en un futur d'implementar una integració amb l'alarma, la qual cosa permetrà automatitzar encara més el procés d'establiment d'un horari de son òptim.

La part pràctica d'aquest projecte està desenvolupada en anglès per a poder en un futur ser accessible a la botiga de Skills d'Alexa de forma global. També l'anglès ajuda en el desenvolupament de la Skill, ja que gran part de la programació es treballa amb anglès, sobretot per part de la documentació, i afegir un llenguatge addicional com el castellà o el català en un futur seria completament possible.

1.1 Context

Els assistents virtuals han experimentat un impressionant creixement en popularitat, i un dels pioners en aquesta àrea va ser Siri, desenvolupat per Apple i llançat el 2011. La seva inclusió als dispositius mòbils va impulsar la seva adopció massiva, permetent als usuaris realitzar tasques mitjançant comandes de veu. Ràpidament, altres competidors com Alexa d'Amazon i Google Assistant van entrar en escena, incorporant assistents virtuals als seus propis telèfons intel·ligents.

Inicialment, els assistents de veu eren vistos com una característica d'entreteniment o per a consultes senzilles com recordatoris i previsió del temps. No obstant això, les empreses que els van desenvolupar van adonar-se que només un 22% dels usuaris feien un ús diari d'aquests assistents. Per motivar una utilització més freqüent, van començar a sorgir els altaveus intel·ligents, una nova categoria de dispositius que incorporaven els assistents de veu i oferien una qualitat d'àudio superior a la dels telèfons mòbils.

Els altaveus intel·ligents es van convertir ràpidament en un element essencial en les llars modernes, ja que permetien als usuaris accedir fàcilment a les funcions dels assistents virtuals a través de la veu des de qualsevol habitació de la casa. La característica més popular d'aquests altaveus era la seva capacitat per reproduir música, oferint una experiència acústica excepcional.

Amb el temps, la comunitat va començar a crear una àmplia gamma de skills pels assistents virtuals, que van des de jocs entretinguts fins a informació en temps real sobre notícies: esports, entre d'altres. Aquestes habilitats addicionals van convertir els altaveus intel·ligents en dispositius encara més versàtils i útils pels usuaris.

Encara que Google també ofereix versions dels seus altaveus intel·ligents, es va decidir decantar per desenvolupar una Skill específicament per a Amazon Alexa. Aquesta decisió es va basar en la disponibilitat de més documentació i una àmplia base d'aplicacions ja existents en el mercat d'Alexa. Amazon ofereix un complet kit que inclou totes les eines

essencials per al desenvolupament d'una Skill, conegut com a Alexa Skills Kit (ASK). Aquest kit proporciona un conjunt d'eines, documentació detallada, diverses mostres de codi i una *API*[1] per ajudar als desenvolupadors a crear habilitats per la Alexa. Per part de Google, ofereixen un conjunt d'eines i documentació necessària pels desenvolupadors que desitgin crear funcionalitats per al seu assistent, assegurant que els creadors tinguin el suport necessari per desenvolupar amb èxit noves funcionalitats per a Google Assistant.

1.2 Motivació

Aquest projecte troba la seva motivació en la necessitat d'ajudar a les persones que vulguin fer una Skill d'Alexa completament nova, millorant l'assistent de veu durant el procés i l'experiència dels usuaris. Aquesta motivació rau en la creença que la tecnologia pot ser una eina efectiva per afrontar problemes i oferir solucions personalitzades als usuaris i programadors, proporcionant documentació i ajuda accessible per als futurs creadors de noves Skills d'Alexa.

1.3 Objectius

Els objectius d'aquest projecte es deriven d'una preocupació principal: fer una Skill d'Alexa per ajudar a les persones, això no obstant, el primer pas és aprendre a crear la Skill de la millor manera possible i de forma efectiva. Per a fer-ho, s'ha decidit desenvolupar una Skill bàsica explicant els diferents procediments realitzats durant el procés.

La resposta que es vol donar amb aquest projecte és esbrinar quina seria la millor forma de crear una Skill d'Alexa. Es Poden utilitzar diverses eines de desenvolupament com per exemple la ASK d'Amazon o d'altres com el Voiceflow, que és una plataforma de disseny de converses amb sistemes intel·ligents.

En resum, aquest projecte té l'objectiu clar de documentar la creació d'una Skill d'Alexa per millorar la qualitat del son i el benestar general de les persones mitjançant l'ús de la tecnologia d'assistència virtual Alexa, oferint una solució personalitzada per afrontar els problemes relacionats amb el son i destacant les diferències entre les dues metodologies utilitzades.

1 API: interfície de programació per aplicacions formada per definicions i protocols per integrar software.

2 Coneixements Previs al desenvolupament

2.1 Que és Alexa

Alexa és un servei de veu ubicat en el núvol d'Amazon, disponible als dispositius d'Amazon. Aquest servei disposa d'un servei d'habilitats i funcionalitats anomenades Skills, que permet als usuaris expandir i millorar la seva experiència amb el producte.

2.2 Que és una Skill

Una Skill és una aplicació que pot ser activada mitjançant comandes de veu, integrades en dispositius Alexa. Es poden crear aplicacions amb grans habilitats com ara donar les notícies, reproduir música o dictar receptes de cuina.

Segons Adi Agashe, Program Manager a Microsoft, "Alexa està construïda basant-se en el processament del llenguatge natural (PLN), un procediment per convertir la parla en paraules, sons i idees", una tecnologia de *machine learning*[2] que brinda a les computadores la capacitat d'interpretar, manipular i comprendre el llenguatge humà.

Abans de començar a entrar en com s'estructura el *backend*[3] d'una Skill cal entendre com s'interactua amb Alexa.

2.3 Com funciona la Skill

Per interactuar amb Alexa s'ha de generar una frase que pugui diferenciar els components d'aquesta i que l'assistent sàpiga interpretar. Hi ha diferents parts en una frase dirigida a l'assistent virtual Alexa, aquests components es poden veure a la Figura 1.

Figura 1. Estructura d'una ordre a Alexa

Una frase dirigida a Alexa està formada per la següent estructura:

1. **Paraula d'activació (Wake Word**): Aquesta part és la paraula que fa que el dispositiu s'activi i sàpiga que hi haurà una interacció seguidament.
2. **Paraula indicant tipus d'acció (launch):** Aquesta part segueix de la paraula d'activació i serveix per indicar a l'assistent el tipus de petició que s'està fent.
3. **Nom de la Skill (Skill invocation Name):** Aquesta part especifiquem el nom de la Skill que demanarem a l'assistent. Aquest nom és únic, ja que aquest és l'identificador de la Skill. En el cas que dues o més Skills tinguessin el mateix identificador Alexa no sabria a quina Skill redirigir la petició.

2 Machine learning: Aprenentatge automàtic de patrons o dades que permet a la intel·ligència artificial aprendre sense necessitar ser ensenyada.
3 Backend: part lògica d'un projecte, accions que passen durant una execució, però no es veu com passa.

4. **Acció dins la Skill (Utterance):** Aquesta part és on específiques l'acció que demanes a la Skill.

Per a comprendre què són i com funcionen les utterances, primer és essencial definir els intents d'una Skill. Un intent es l'acció que l'usuari vol aconseguir, en el codi es tradueix com a funció, ja que Cada intent delimita un comportament específic. Una Skill té múltiples intents, un per a cada funcionalitat o comportament que volguem implementar.

Per exemple, si es vol desenvolupar una Skill, extremadament bàsica, que simplement ens saludi o ens digui "hola" cada vegada que s'obre. La seva única funció és dir "hola". Com ja s'ha assenyalat, una funció equivaldria a un concepte, de manera que es tindría un únic intent que fa la funció de dir "hola". Si es desitja que, quan es tanqui, aquesta digui "adeu", això constituiria un segon intent.

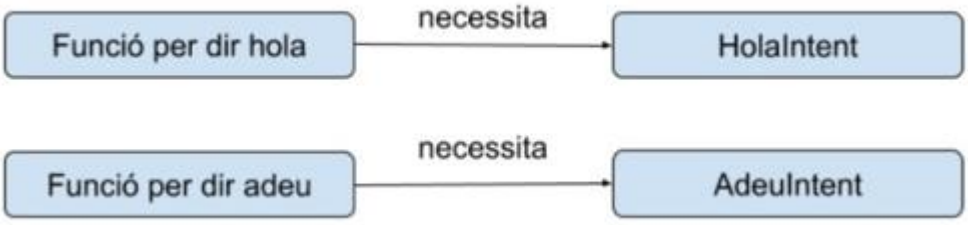


Figura 2. Crida d'un intent

Per a cada funció que existeixi en una Skill, es necessita un intent per activar-la i perquè Alexa sàpiga quan executar cada una de les funcions. L'activació d'un intent, s'ha de realitzar mitjançant les utterances, tal com es pot observar, varies utterances poden correspondre al mateix intent. Aquesta interacció es pot observar de forma gràfica a la Figura 3.

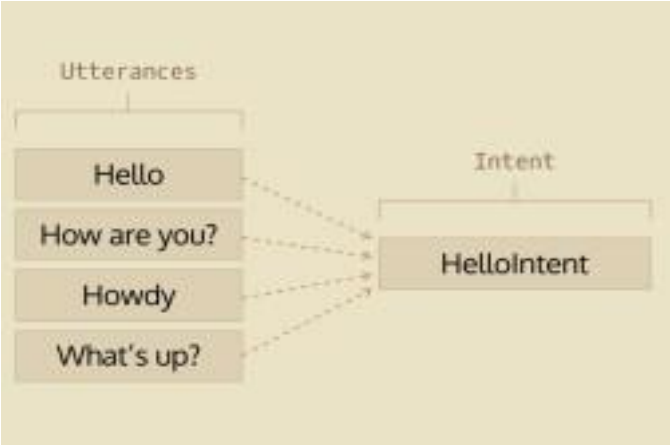


Figura 3. Crida a un intent mitjançant utterances

2.4 Entorns de desenvolupament

2.4.1 SDK o Software Development Kit

Una SDK o Software Developer Kit, és un conjunt definit d'eines i llibreries que els desenvolupadors poden utilitzar per a construir aplicacions en sistemes operatius específics, en aquest cas Alexa.

Alexa disposa d'una SDK que té la funció d'ajudar al desenvolupador a crear una Skill d'una forma fàcil i ràpida.

Aquesta SDK forma part d'un conjunt més gran anomenat ASK o Alexa Skills Kit, que és un conjunt d'eines, documentació i APIs proporcionades per Amazon per a desenvolupar aquestes Skills.

Aquest SDK és útil ja que no t'has de centrar en el codi base sinó que pots utilitzar totes les funcionalitats que des d'Amazon s'ofereixen de forma gratuïta. Amb la SDK d'Alexa es pot crear el projecte i la seva estructura principal.

2.4.2 AWS Lambda

El concepte d'AWS Lambda és un servei de computació sense servidor i orientat a esdeveniments que permet executar codi per a gairebé qualsevol mena d'aplicació o servei backend sense necessitat d'*hostejar*[4] o gestionar servidors. Es pot activar Lambda des de més de 200 serveis d'AWS i aplicacions de Software com a Servei (SaaS), i només es paga pel que es fa servir. Amb AWS Lambda, la forma d'executar el codi és escrivint i pujant el codi com un fitxer .zip o una imatge de contenidor. La Lambda respon automàticament a les sol·licituds d'execució de codi a qualsevol escala, des d'una dotzena d'esdeveniments al dia fins a centenars de milers per segon. A més, els costos funcionen pel temps de computació que s'ha utilitzat en lloc d'hostejar la infraestructura anticipadament que necessitarem. També permet interactuar amb bases de dades com DynamoDB.

2.4.3 Voiceflow

Voiceflow és una plataforma basada en web que permet dissenyar, prototipar i llançar assistents conversacionals per a diversos canals i casos d'ús. Pots utilitzar-la per crear *chatbots*[5] per a llocs web, aplicacions de missatgeria o plataformes de xarxes socials, o assistents de veu per a altaveus intel·ligents, telèfons o cotxes.

Voiceflow ofereix un enfocament sense codi, permetent als usuaris crear interaccions de veu mitjançant una interfície visual d'arrossegar i deixa anar. Aquesta plataforma té com a objectiu facilitar la creació ràpida de prototips i col·laboració multidisciplinària.

2.5 Intents proporcionats per Amazon

Amazon ofereix intents bàsics que ajuda als desenvolupadors a crear i gestionar noves Skills. Aquests intents es poden usar o no, depenent de cada situació. Seguidament s'expliquen alguns dels més importants.

- AMAZON.CancelIntent: Aquest intent es fa servir quan es vol cancel·lar una acció realitzada.
- AMAZON.HelpIntent: Intent per a demanar ajuda.
- AMAZON.StopIntent: Intent per a parar l'execució de l'intent.
- AMAZON.NavigateHomeIntent: Intent per tornar al començament de la

4 Hostejar: proporcionar els recursos d'emmagatzematge i computació que mantenen un servei actiu
5 Chatbots: programes que simulen una conversa amb humans amb respostes automàtiques

Skill.

2.6 Slots

Un slot és una variable que es relaciona amb un intent i permet a Alexa entendre informació sobre la sol·licitud. Per exemple, en una Skill que proporciona l'horòscop diari a les persones, la sol·licitud de l'usuari podria ser l'expressió "Dona'm l'horòscop per a Leo". En aquest exemple, "Leo" seria el slot personalitzat.

Amazon registra les paraules, però com que la interpretació de sons requereix molta potència computacional, les gravacions de veu s'envien als servidors d'Amazon per a ser analitzades de forma més eficient.

Els servidors d'Amazon, descomposen la frase en sons individuals i consulta en una base de dades que conté la pronunciació de diferents paraules per a comprar-les individualment amb els diferents sons rebuts.

Aleshores identifica les paraules importants per comprendre les tasques i funcions corresponents a realitzar. Per exemple, si Alexa detecta paraules com "esport" o "bàsquet", aquesta obrirà l'aplicació d'esports. A l'acabar l'analisi als servidors d'Amazon, la informació torna al dispositiu Alexa, en aquest moment Alexa repetirà el mateix procés descrit anteriorment, però en ordre invers, per tal d'activar la resposta adequada.

La definició de les paraules de l'intent s'han d'indicar en diferents slots, les paraules s'indiquen dintre de claudàtors ({}). La definició del tipus de slot consisteix a predir totes les possibles paraules que un usuari pot dir per aquell intent. Aquestes possibilitats són guardades en un fitxer *csv*[6] i es puja a la web d'Amazon. Un cop Alexa capta una paraula acceptada dintre d'algun slot, guarda la resposta de l'usuari. Cal remarcar que per als slots personalitzats, no creats per Amazon, les paraules que utilitzem per a definir els possibles valors d'aquesta es comporten com una "guia" per a Alexa. l'ajuden a entendre quin tipus de paraules volem en aquest intent, per exemple els colors: groc i blau, Alexa entendrà el color vermell també. Malgrat això, no ha estat la metodologia escollida, perquè podia donar a errors en alguna ocasió, ja que es pot indicar la sensibilitat de l'exactitud que es vol que fagin les crides amb les utterances. En el nostre cas, s'ha escollit una sensibilitat alta, és a dir, un major nivell de filtratge.

Amazon proporciona molts slots ja creats amb les paraules correctes, com per exemple els de la Figura 4.

6 Csv: arxiu de text pla que emmagatzema dades delimitades per comes

Slot Type	Short Description
AMAZON.AlphaNumeric	(Public beta) Converts letters, numbers, hyphens and periods into alphanumeric sequences. For example, converts "sixteen a." to 16a. You can use this slot type with all locales except Arabic (SA).
AMAZON.DATE	Converts words that indicate *dates* into a date format. For example, converts "today", "tomorrow" or "july" to 2015-07-00T9.
AMAZON.DURATION	Converts words that indicate *durations* into a numeric duration. For example, converts "five minutes" to PT5M.
AMAZON.FOUR_DIGIT_NUMBER	Converts words that represent *four-digit numbers* into digits. For example, converts "six oh four five" to 6045.
AMAZON.NUMBER	Converts numeric words into digits. For example, converts "five" to 5.
AMAZON.Ordinal	(Public beta) Converts ordinal numbers into digits. For example, converts "first" to 1 and "second" to 2.
AMAZON.PhoneNumber	(Public beta) Converts the numbers or words that represent a phone number into a string format without punctuation.
AMAZON.TIME	Converts words that indicate time into a time value. For example, converts "four in the morning", to 04:00 and "two p m" to 14:00.

Figura 4. Slots predefinits per Amazon

2.7 Comunicació des de l'usuari fins a la Skill

Alexa es comunica amb el servei utilitzat a través d'un mecanisme de sol·licitud-resposta mitjançant *HTTP*[7] i de protocols criptogràfics *SSL/TLS*[8], que formen HTTPS. Quan un usuari interactua amb una Skill Alexa, el servei rep una sol·licitud *POST*[9] que conté un cos *JSON*[10], que és el mètode utilitzat per a definir el model d'interacció de la Skill.

Finalment, el cos de la sol·licitud conté les propietats necessàries perquè el servei realitzi la lògica per la que ha estat programat i generi una resposta amb format JSON.

7 HTTP: protocol de comunicació que permet la transferència de dades entre dos sistemes.
8 SSL/TLS: protocols de seguretat utilitzats per establir una connexió xifrada entre un usuari i un servidor.
9 POST: mètode HTTP que envia dades a un servidor per crear/actualitzar un recurs al servidor.
10 JSON: fitxer que emmagatzema estructures de dades i objectes senzills

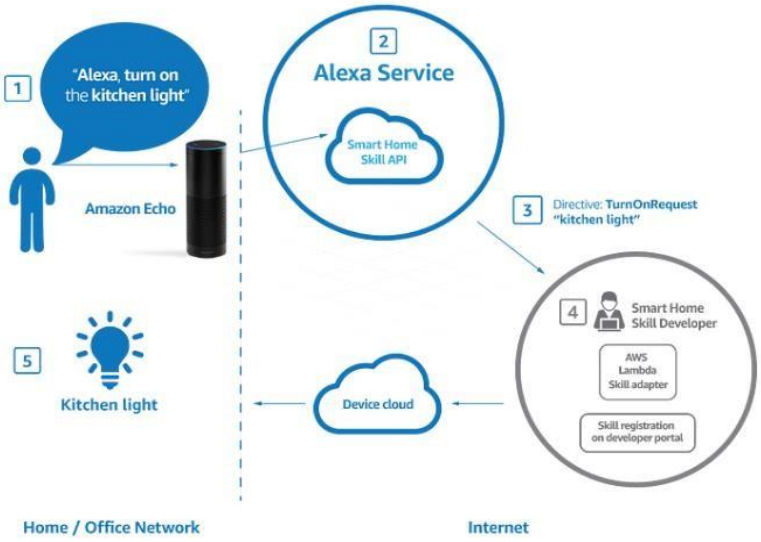


Figura 5. Representació de crida fins al Back-End

3 Implementació de la Skill amb la SDK i la AWS Lambda

La SDK s'utilitza per a definir l'estructura de la Skill, els intents i les seves utterances, els tipus de slots i el nom d'invocació de la Skill.

El servei Lambda d'AWS s'utilitza per escriure el codi, la lògica del projecte on es defineixen les funcionalitats de cada intent i es faran les connexions amb les bases de dades i recursos corresponents.

La part del backend està hostejada per la lambda, però també podria estar hostejada a la consola SDK en cas de no voler utilitzar la Lambda. En el nostre cas, gràcies a la Lambda, sempre es tindrà en el futur la possibilitat d'escalar el projecte i utilitzar altres serveis oferts per AWS, ja que a causa de les limitacions i les restriccions que té la SDK d'Alexa és més difícil de treballar amb aquesta, en canvi, la base de dades que s'utilitza, DynamoDB, ja ve integrada dintre dels serveis d'AWS.

3.1 Entorns de desenvolupament SDK d'Alexa i AWS Lambda

3.1.1 SDK d'Alexa

A la SDK es pot treballar de diferents formes, configurant la Skill amb la mateixa plataforma o configurant la Skill mitjançant un fitxer .json, totes dues opcions són vàlides.

La SDK es compon, principalment, per la construcció del projecte i la definició del model d'iteracció, per la consola de proves, que simula a un aparell Alexa, i per a un directori de fitxers de codi per a poder programar la Skill. La part lògica de la Skill es troba, a la Lambda d'AWS.

En aquest projecte s'ha configurat la Skill tal com s'ha indicat anteriorment, definint el nom d'invocació **sleeping helper**, la Skill tindrà quatre intents personalitzats:

- SleepHoursIntent
- ReasonIntent
- moodIntent
- useChatGPTIntent

I dos slots:

- mood
- reasons_diff

3.1.2 AWS Lambda

La Lamba proporciona una interfície gràfica per a crear, configurar i gestionar les funcions, s'anomena Lambda Console. Les configuracions que més destaquen de la Lambda són les següents:

Un **desencadenant** és la raó per la qual la funció s'executa, per a que una funció Lambda s'executi, ha d'haver-hi algun esdeveniment. En el cas d'aquest projecte, el desencadenant és l'inici de la Skill d'Alexa.

Una **destinació** s'utilitza per a rebre registres de crida de la funció Lambda als serveis d'AWS.

Una **capa** serveix per a poder importar una o més llibreries, en aquest cas s'importarà la del "openai". Les capes són beneficioses per a poder reutilitzar-les amb altres codis usant simplement la capa corresponent, fent que el projecte sigui molt més lleuger i ràpid.

A la consola de la funció Lambda s'inclou un editor de codi, on es troba el codi d'aquest projecte. En aquest editor es pot tant provar tant els intents i les funcions de la Skill com desplegar aquesta lògica a la Skill connectada.

Per a poder fer ús del **DynamoDB** dintre del codi, s'han de configurar els permisos de la Lambda, per tal de garantir la seguretat de la informació gestionada i els accessos d'aquesta Lambda als serveis desitjats. Per a poder donar permisos a la Lambda, s'ha de crear un **rol** al servei de IAM d'AWS, que serveix per controlar els accessos als recursos d'AWS.

Un cop creat el rol, s'han de crear o afegir polítiques "attach policies" i cercar pel nom DynamoDB, tot seguit afegir els accessos que es necessitin, lectura i escriptura en aquest cas, també es poden afegir permisos absoluts o "FullAccess". En finalitzar aquest procés, es podrà accedir al servei especificat des de la funció Lambda escollida.

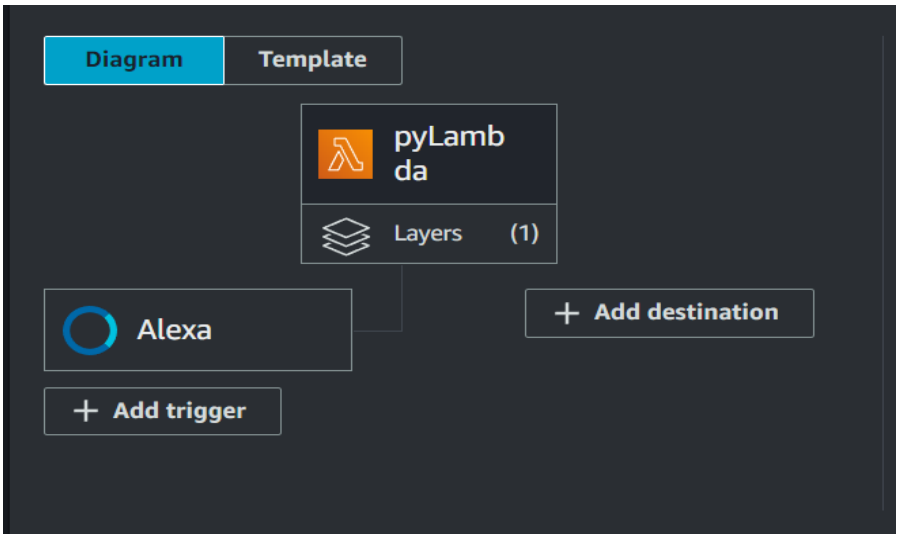


Figura 6. Diagrama estructural d'una Funció Lambda

3.2 Aspectes generals i descripció de la Skill personalitzada

L'objectiu d'aquesta Skill és ajudar l'usuari a portar un millor horari de son i aconsellar-li fent una consulta a la Intel·ligència artificial, en la mesura del possible, per a complir aquest objectiu. Una Skill té certs requisits en el moment de crear-la, un d'ells és tenir un nom únic i sense lletres majúscules.

Així doncs, el nom d'invocació de la Skill és **sleeping helper**, tal com s'indica en descriure aquesta.

La funcionalitat primària d'aquesta Skill és portar un control de les hores de son del dia a dia, preguntar a l'usuari com és sent al despertar i si sap el motiu de trobar-se així. A partir d'aquí, es pot derivar a altres funcionalitats com possibles problemes del son i algun consell o, per exemple, demanar a la Skill que pregunti a la IA algun consell per a intentar millorar la situació.

Dintre de la Skill hi ha un intent que connecta amb la IA, la Skill genera un prompt amb les condicions i resultats de les preguntes formulades des de l'execució de la Skill i espera una resposta de la intel·ligència artificial per a finalment poder transmetre-la a l'usuari.

3.3 Requeriments de la Skill

S'ha de disposar de:

- **Una interfície intuïtiva i amigable** per a garantir la facilitat d'ús, permetent als usuaris amb diferents nivells de coneixement tècnic navegar i accedir a les funcionalitats sense esforç.
- Pel que fa a la **funcionalitat**, s'ha de proporcionar una àmplia gamma de característiques i capacitats per satisfer les diverses necessitats dels usuaris:
  - **Seguiment del son**
  - creació del *prompt*[11] a la IA.
- **Capacitat d'evolucionar** per incorporar nous coneixements, tecnologies i retroalimentació de l'usuari, assegurant una experiència en contínua millora i compromesa amb l'usuari.

3.4 Decisions de disseny

La decisió de disseny més important ha estat la de fer el projecte per separat, utilitzant la SDK i la Lambda, en comptes d'utilitzar únicament la SDK, altres decisions a destacar han estat:

- **Escalabilitat:** S'ha treballat amb la SDK i la Lambda per separat, en comptes de treballar únicament amb la SDK a causa de la possibilitat d'escalar l'aplicació en un futur.
- **Amazon Web Services** (Ecosistema d'aplicacions més ampli): En treballar amb AWS Lambda es pot diferenciar la part lògica del *model d'interacció*[12] i treballar amb un ecosistema d'aplicacions més ampli (AWS), el que comporta un processament de dades i funcionament de la lògica molt més eficient.
- **DynamoDB** L'ús d'una Base de Dades ha estat essencial per a poder guardar els valors registrats de forma diària de forma separada i eficient.
- **Atributs** (valors temporals durant l'execució de la Skill)**:** Ja que amb Alexa es pot utilitzar aquesta eina d'*atributs de sessió*[13], s'ha utilitzat per estalviar crides a la Base de Dades durant la mateixa sessió. Permetent un cost temporal, computacional i monetari inferior.
- **Sobreescriptura de fitxers:** El directori de treball de l'editor de text de la funció Lambda és exclusivament accessible en forma de lectura, el que significa que el codi de la funció no pot modificar o sobreescriure aquests fitxers. Per tant, per a poder modificar aquests fitxers durant l'execució,

11 Prompt: text que s'utilitza com entrada d'un sistema informàtic o programa.

12 Model d'interacció: cicle de vida d'un programa dividit en processos més petits anomenats iteracions.

13 Atributs de sessió: valors que es guarden durant tota l'execució de la Skill.

s'accedeix als fitxers auxiliars mitjançant un servei d'Amazon anomenat S3 Bucket17. On si es poden modificar els fitxers, ja que no formen part del directori arrel del projecte i podem continuar utilitzant-los de la mateixa forma.

3.5 Disseny

És essencial mantenir una estructura de projectes adequada per a la funcionalitat de la Skill, l'estructura de fitxers i directoris que s'usarà en el nostre projecte es la que es genera per defecte.

3.5.1 Estructura

El "project root folder" d'un projecte d'Alexa es divideix en els següents directoris:

- .ask: Carpeta on es troba el fitxer ask-states.json on es pot trobar la Skill Id i altres metadades.
- lambda: Carpeta on es troben els fitxers de codi i de la funció de la lambda que fa que la Alexa funcioni. Dintre d'aquesta carpeta hi ha fitxers com:
  - lambda_function.py: conte el codi dels *handlers*[14] per a la Skill
  - local_debugger: conte les funcions per a poder debugar el codi a la consola.
- skill-package: Informació de la Skill; nom, descripció, tipus, converses, models d'interacció, el fitxer *build*[15], etc.

3.5.2 Connexió entre la Lambda i la SDK

Per a fer la integració de la Skill mitjançant una funció Lambda d'AWS primer s'ha de crear una nova funció especificant el llenguatge que es vol utilitzar.

Després s'ha d'afegir un desencadenant a la funció, aquest serà la Skill d'Alexa, s'haurà d'afegir el "id" de la Skill. Aquest pas serveix per indicar que la funció lambda s'ha d'activar a l'executar la Skill d'Alexa com a *endpoint*[16], és a dir, el seu "Amazon Resource Name" o (ARN), que permetrà que la Skill d'Alexa funcioni a partir del codi de la funció Lambda amb la que esta relacionada.

14 Handler: encarregats de gestionar un o diversos tipus de sol·licituds entrants.

15 Build: Procés de traduir el codi llegible per humans a un programa executable.

16 Endpoint: fi de l'adreça on es dirigeixen les sol·licituds per al processament lògic.

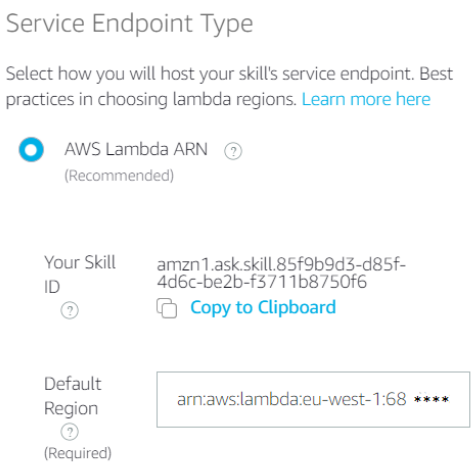


Figura 7. Configuració del Endpoint de la Skill

3.5.3 Flux d'execució de la Skill

A la figura 8 es pot observar quin seria el flux d'execució de la Skill, començant des de la invocació de la Skill i continuant pels intents fins a arribar al final.

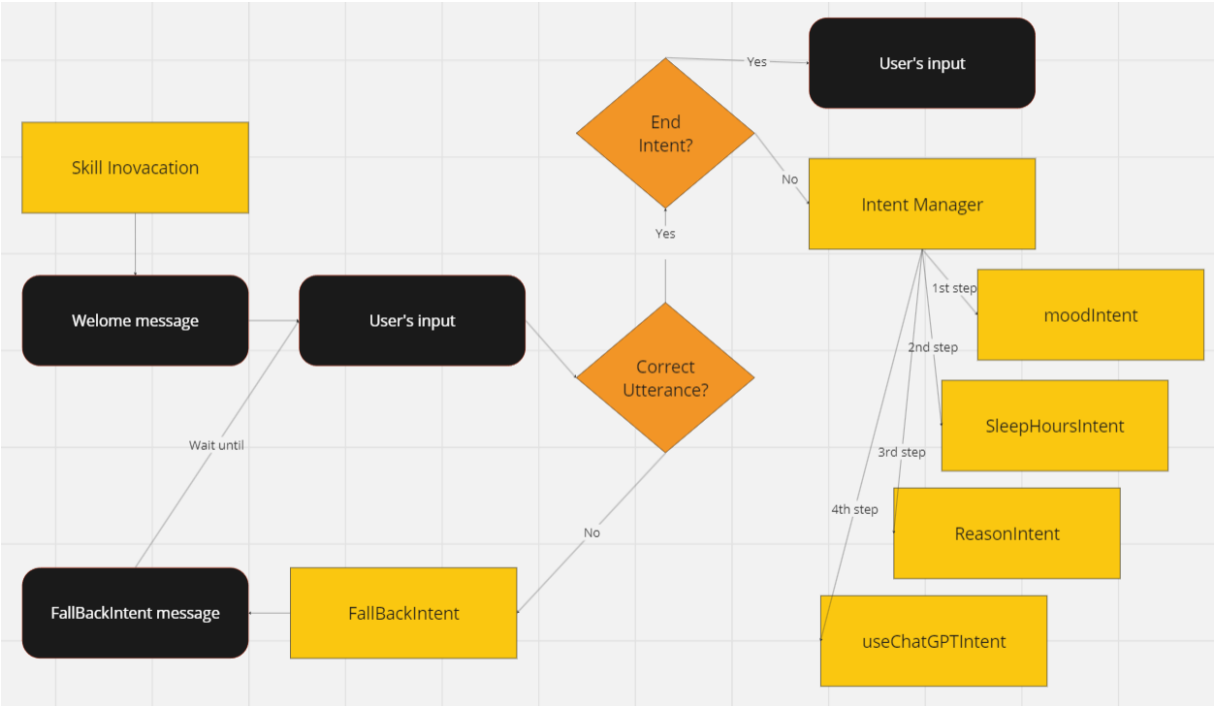


Figura 8. Diagrama de flux de la Skill desenvolupada amb la SDK

3.5.4 Base de dades DynamoDB

Per a la utilització dels resultats diaris, tant de les hores de son, com dels estats d'ànim o els motius de tenir una bona o mala nit de son, es necessita una base de dades que guardi aquesta informació a cada sessió, si no, no es podrien mantenir les dades des d'una crida de la Skill a les següents.

Per aconseguir-ho, es guarda aquests valors en una Base de Dades proporcionada per Amazon anomenada DynamoDB. L'accés a aquesta taula és fàcil, ja que és un servei d'AWS i, per tant, és l'opció més propera que hi ha.

Hi ha dues taules que s'utilitzen en aquest projecte:

- **SleepData**, guarda els resultats diaris dels valors d'hores dormides, sensació en despertar-se i motiu, distingit per un número "id" per a cada dia.
- **TotalSleepData**, guarda els resultats de la suma total d'hores dormides, classificant-les segons la sensació de l'usuari: bona, dolenta, normal. Per ara, aquesta taula és únicament utilitzada per a guardar la suma total dels valors registrats, però de moment no s'està utilitzant.

La forma d'emmagatzemar les dades en la DynamoDB, tal com es pot observar a la figura 9 i 10, és mitjançant una taula on es guarden les dades per camps, es guarda un registre per a cada dia.

id (String)	hours	mood	reason
61	7	normal	nervous for an exam
1	5	happy	i don't know

Figura 9. Taula SleepData

mood (St...	days	hours
neutral	3	19
good	46	394
bad	12	94

Figura 10. Taula TotalSleepData

Per a treballar amb la base de dades DynamoDB d'Amazon des de la lambda s'han de concedir a la funció Lambda els permisos mencionats a l'apartat 3.1.2 AWS Lambda.

3.5.5 Atributs de la sessió

Un servei que es pot utilitzar amb Alexa és que es pot guardar i llegir informació dels atributs usats durant la sessió actual o sessions anteriors (crides diferents), es pot definir atributs i guardar-los durant el transcurs de la sessió actual, per a totes les crides o únicament durant la crida d'una funció. Aquests valors es guarden mitjançant l'ús de claus distintives per a cada valor. Aquests valors que es guarden dins d'un ***mapa***[17] .

17 Mapa: estructura de dades que permet emmagatzemar valors i relacionar-los amb claus.

Per a l'emmagatzematge i utilització de les dades guardades durant la sessió s'ha decidit utilitzar un sistema de mapeig anomenat “sessionAttributes”. Si s’utilitzés variables globals, aquestes es reiniciarien a cada intent. A continuació s’explicarà quins són els valors que es guarden en aquest mapa:

- id: Aquesta és una clau per a diferenciar cada objecte, en aquest cas és el que es coneix com a clau principal.
- hours: aquests és el valor de les hores dormides per l’usuari.
- mood: és l’estat anímic de l’usuari.
- reason: el motiu de l’estat anímic de l’usuari, si n’hi ha.

Aquests valors són guardats un a un, cadascun des del seu corresponent intent. Quan s’arriba al **reasonIntent**, on es pregunta si es vol demanar un consell a la intel·ligència artificial, és on s’agafen tots els valors guardats durant la sessió i s’emmagatzemen a la Base de Bades DynamoDB.

3.5.6 Intents

Els intents de la Skill es poden dividir en dos grups, els requerits i ja implementats, i els afegits i personalitzats. Com ja diu la paraula, amb els intents ja implementats no s’ha de de fer res, són intents per defecte que tenen totes les Skills i serveixen majoritàriament per ajudar a evitar errors. Referència a l'apartat 2.2.

Els intents que s’han personalitzat s'anomenen:

- SleepHoursIntent
- moodIntent
- ReasonIntent
- useChatGPTIntent

Figura 11. Flux per a cridar un intent

La figura 11 fa referència a com seria el començament de la Skill i la seva lògica de funcionament.

A continuació s’explica la funcionalitat de cadascun d'aquests intents:

El **moodIntent** és l'intent que serveix per a preguntar l'estat anímic de l'usuari en despertar-se. Aquesta Skill s'activa automàticament després de respondre correctament el missatge de l'intent de llançament de la Skill (launchIntent) de*:" Welcome! How are you feeling today?"*. D'aquesta forma s'obté l'estat anímic de l'usuari. En la següent conversa es fa referència a com seria el procés de la crida moodIntent:

Alexa:*"Welcome! How are you feeling today?"*

Usuari:*"I feel happy"*

Alexa:*"How many hours did you sleep?*

La forma en la qual, aquest intent actua és preguntant l'estat anímic de l'usuari i llegint el slot que es detecta com a input, comprovant que sigui un adjectiu (estat anímic) vàlid dintre la definició de possibles valors del slot persnalitzat, com per exemple "happy, en cas contrari, es consultarà a la IA si l'adjectiu que l'usuari utilitza per a descriure el seu estat anímic es vàlid en aquesta situació o no, si la resposta de la IA és afirmativa, s'afegirà aquest nou adjectiu a la llista creada per a tenir-lo en compte la propera vegada, en cas contrari, és demanarà a l'usuari que utilitzi un altre adjectiu per a descriure el seu estat anímic.

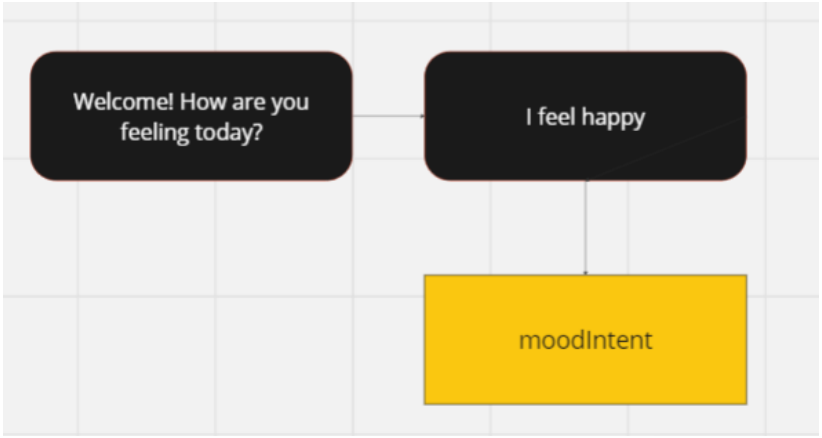


Figura 12. Flux de l'intent moodIntent

El **SleepHoursIntent** és l'intent que serveix per a preguntar les hores de son que ha tingut. Aquesta Skill s'activa automàticament després de respondre el missatge de l'intent moodIntent de: *"feeling: (mood), how many hours did you sleep?"*. D'aquesta forma s'obté l'estat anímic i les hores de son. En la següent conversa es fa referència a com seria el procés de la crida SleepHoursIntent:

Alexa*:"How many hours did you sleep?"*

Usuari*:"10 hours"*

La forma en la qual aquest intent actua és preguntant les hores dormides i llegint el slot que es detecta com a input, observar la següent imatge que fa referència al slot dintre de l'Intent:

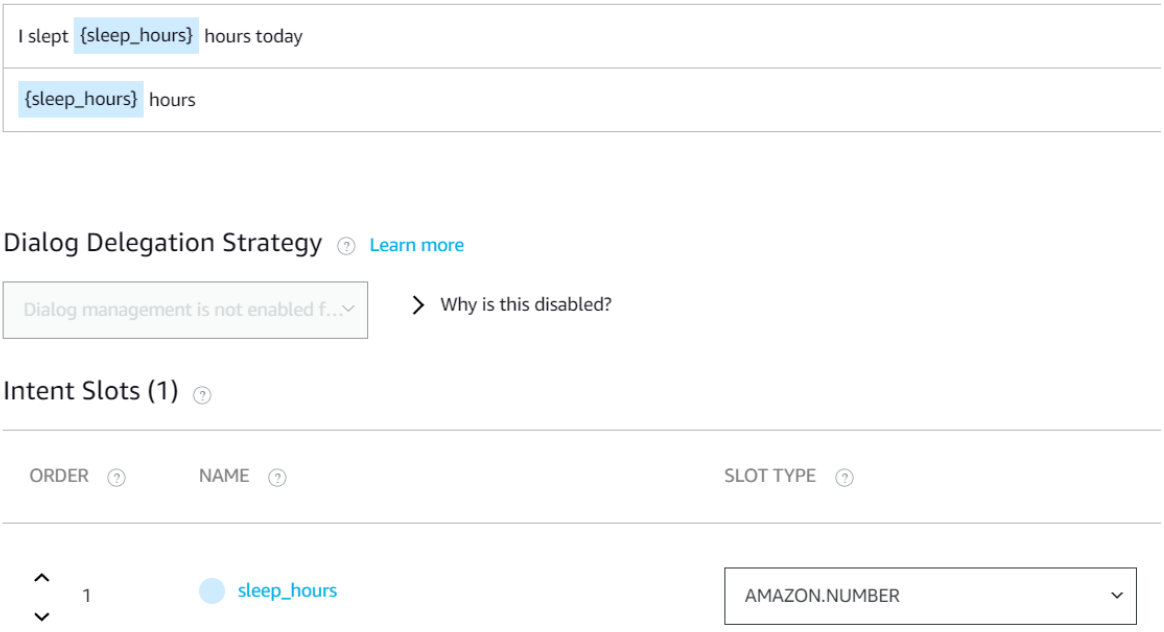


Figura 13. Slot necessari per SleepingHoursIntent

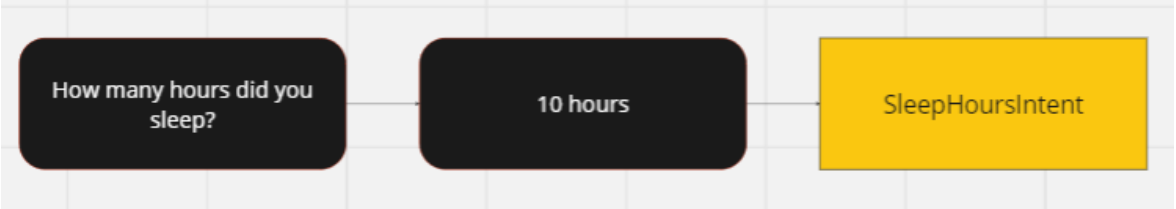


Figura 14. Flux de l'intent SleepHoursIntent

El **ReasonIntent** és l'intent que serveix per a preguntar el motiu de l'estat anímic de l'usuari en despertar-se, si el sap. Aquesta Skill s'activa automàticament després de respondre correctament el missatge de l'intent SleepHoursIntent de: *"And what is the reason for you feeling like good?"*. D'aquesta forma s'obté el motiu de l'estat anímic. En la següent conversa es fa referència a com seria el procés de la crida ReasonIntent:

Alexa: *"And what is the reason for you feeling like good?"*

Usuari: *"Because of my comfort pillow"*

Alexa: *"You feel like happy and slept 10 hours and it was due to comfort pillow, do you want to ask for any advice to the Artificial Intelligence?"*

User: *"sure"*

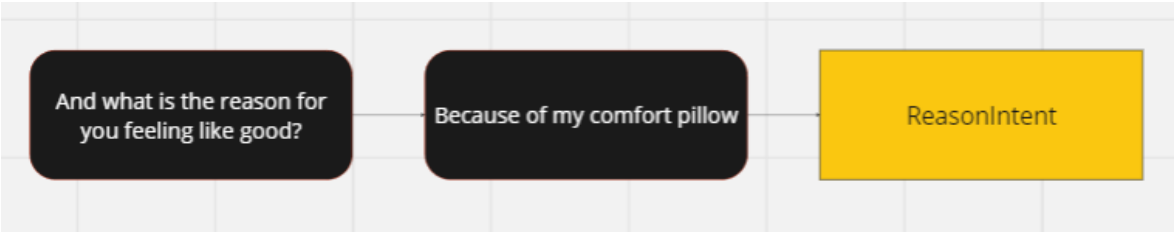


Figura 15. Flux de l'intent ReasonIntent

Es va decidir preguntar a l'usuari directament si sap per què se sent així, tot seguit proposant-li si voldrà preguntar a la intel·ligència artificial algun consell basat en totes les dades guardades anteriorment.

El **useChatGPTIntent**, aquest intent connecta amb una intel·ligència artificial, s'activara quan l'usuari confirmi la resposta a la pregunta de l'intent ReasonIntent de: *"You feel like happy and slept 10 hours and it was due to comfort pillow, do you want to ask for any advice to the Artificial Intelligence?"* En aquest intent es crearà el prompt que Alexa enviarà a la intel·ligència artificial. En la següent conversa es fa referència a com seria el procés de la crida ReasonIntent:

Alexa:*"You feel like happy and slept 10 hours and it was due to comfort pillow, do you want to ask for any advice to the Artificial Intelligence?"*

User:*"sure"*

Alexa:*"(Resposta segons la pregunta (prompt) realitzada)"*

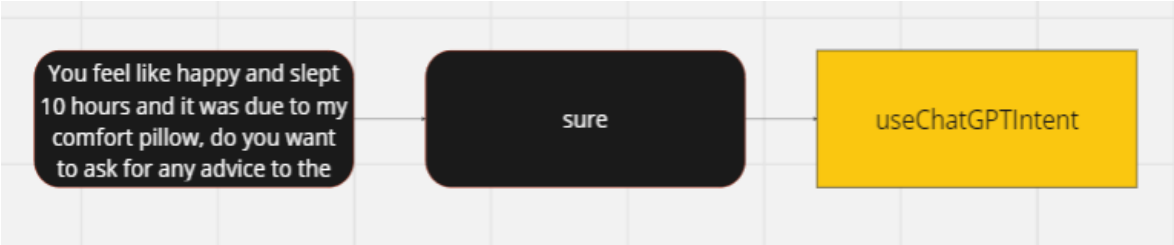


Figura 16. Flux de l'intent useChatGPTIntent

3.5.7 Slots

Aquests són els Slots implementats en la Skill, com es pot observar n'hi ha dos de personalitzades més l'AMAZON.NUMBER, aquesta és la manera de llegir els *inputs*[18] de l'usuari a l'Alexa, per tant, per a inputs no definits o no tan específics com els que es necessitaven, s'ha hagut de crear de personalitzats. Aquests són els següents referits a la **Figura 17**:

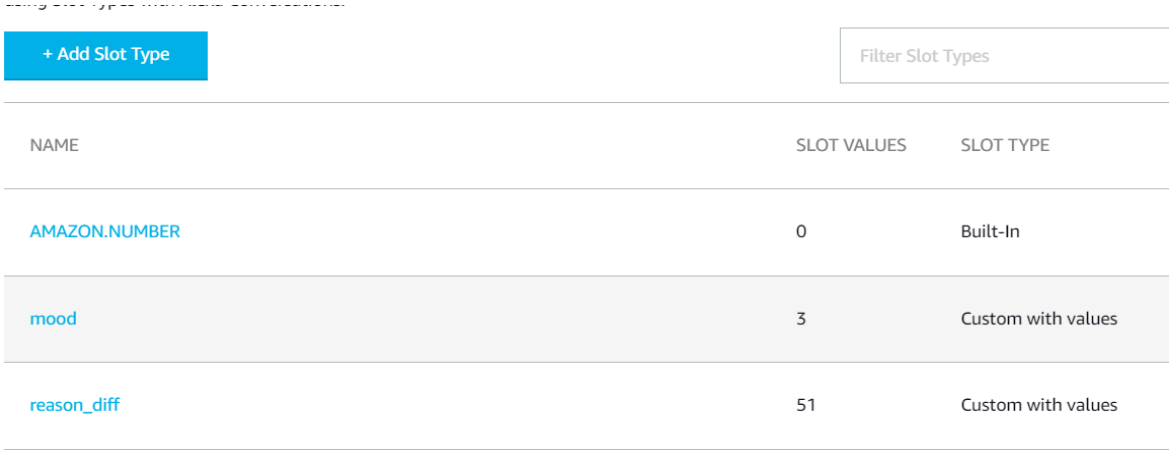


Figura 17. Slots personalitzats del projecte SDK

L' Slot de **"mood"**, referir-se a la Figura 17 i 18, conté valors de tipus *"good"*, *"bad"* i *"neutral"* amb sinònims per a cadascun que es podríen classificar com adjectius.

L' Slot de **"reason_diff"**, conté una llista de possibles valors classificats com "motius" pels quals podria l'usuari tingut una bona o mala nit.

De nou, aquests valors dels slots són "orientatius" per a la Alexa, permetent que es puguin afegir automàticament valors a mesura que s'utilitza la Skill.

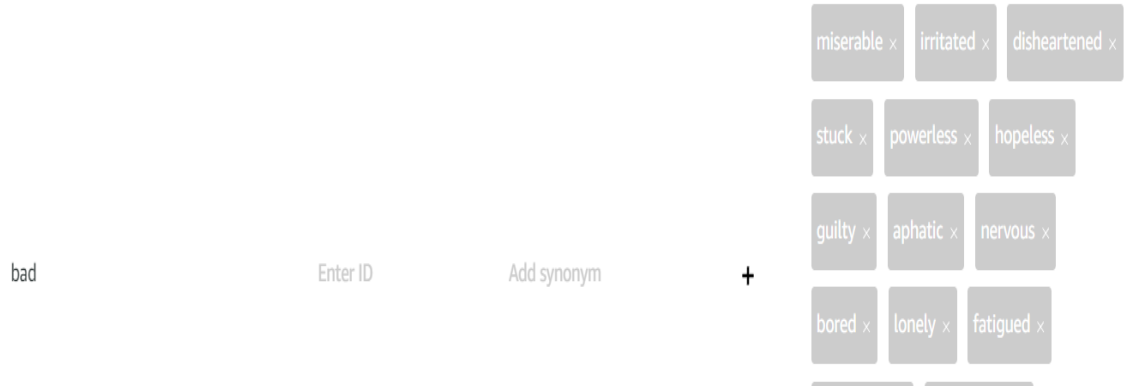


Figura 18. Sinónims del valor bad en el slot mood

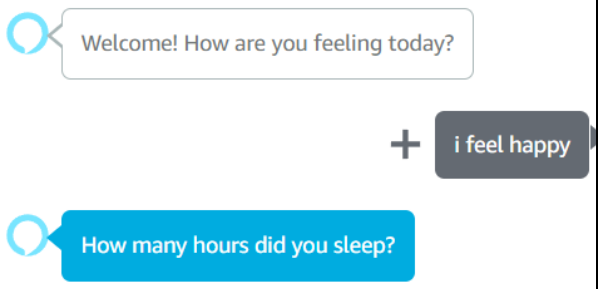
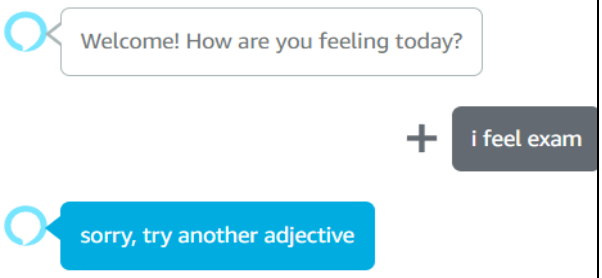
3.5.8 Jocs de proves amb la SDK d'Alexa

En aquest joc de proves es mostra l'exemple de com cridar la Skill amb el seu nom **d'invocació** de forma correcta i incorrecta:

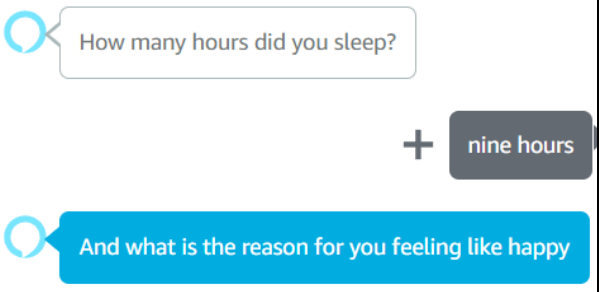
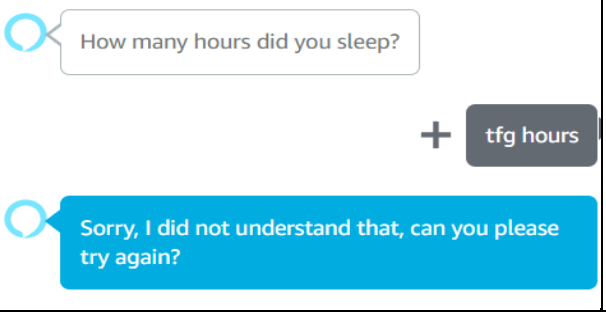
Tests	Iteració	Resultat
Bon input	+ sleeping helper Welcome! How are you feeling today?	OK
Input dolent	+ sleeping Sorry, I don't know that.	OK

18 Inputs: missatges que el programa interpreta o llegeix

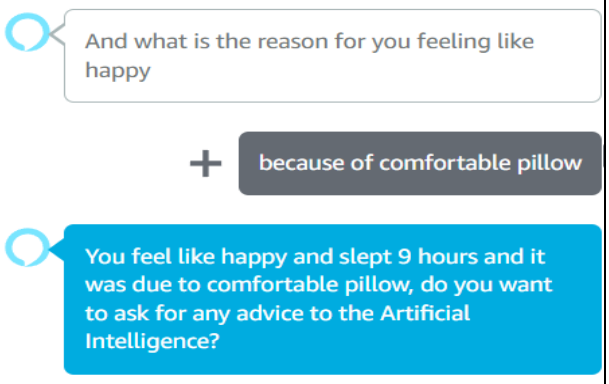
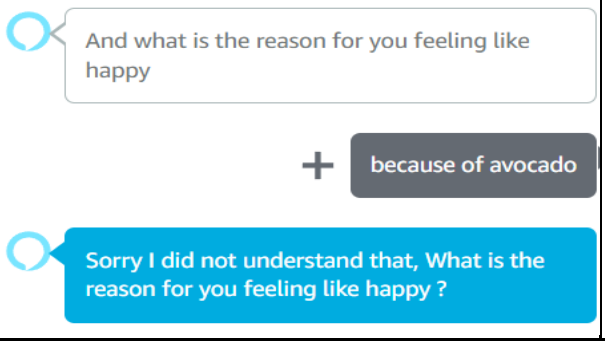
En el següent joc de proves es mostra l'exemple de com cridar l'intent **moodIntent** amb la utterance correcta i incorrecta:

Tests	Iteració	Resultat
Bon input		OK
Input dolent		OK

En el següent joc de proves es mostra l'exemple de com cridar l'intent **SleepHoursIntent** amb la utterance correcta i incorrecta:

Tests	Iteració	Resultat
Bon input		OK
Input dolent		OK

En el següent joc de proves es mostra l'exemple de com cridar l'intent **ReasonIntent** amb la utterance correcta i incorrecta:

Tests	Iteració	Resultat
Bon input		OK
Input dolent		OK

4 Implementació de la Skill amb Voiceflow

4.1 Low code amb Voiceflow

El principal atractiu de Voiceflow és que es fàcil i intuïtiu. No es necessiten coneixements de programació ni coneixements tècnics per crear un propi assistent conversacional. Tot el que es necessita és una interfície *"d'arrossegar i deixa anar*", un conjunt de plantilles per defecte i una mica de creativitat.

4.2 Entorn de desenvolupament Voiceflow

Per a poder treballar a l'entorn de Voiceflow primer s'ha d'entendre la principal estructura d'aquesta plataforma, s'ha dividit en: desenvolupament, disseny, prototipat i, finalment, publicació.

4.2.1 Desenvolupament i Disseny

Al crear nous projectes de Voiceflow, principalment s'ha d'escollit la finalitat d'aquest, per exemple una aplicació de veu per Alexa, Google o qualsevol chatbot per a una web o aplicació mòbil.

La plataforma funciona com un *canvas*[19] on s'utilitzen i es posicionen blocs per a poder crear el flux (*flow*[20]) conversacional. Cada bloc representa una interacció o acció diferent, com parlar a l'usuari, capturar una resposta de l'usuari o executar lògica.

Alguns dels blocs més destacats són:

- **Bloc de parla:** serveix per enviar un missatge a l'usuari, és a dir, parlar a l'usuari.
- **Bloc de captura:** serveix per a capturar el missatge introduït per l'usuari.
- **Bloc d'elecció:** serveix per a crear diversos camins segons el valor introduït.
- **Bloc de configuració:** serveix per assignar valors a variables o per a fer càlculs lògics com per exemple una suma.
- **Bloc d'API:** serveix per a poder integrar APIS externes per obtenir dades o activar accions.

A més, també hi ha:

- **variables** que es poden utilitzar per a guardar valors i manipular-los, com per exemple la resposta de l'usuari, que es pot comparar amb lògica condicional
- **condicional,** permet al programa prendre decisions per a crear converses més dinàmiques que puguin canviar el transcurs del recorregut principal segons les condicions.

19 Canvas: plataforma o sistema digital utilitzat per a diversos propòsits com desenvolupar o enseñar

20 Flow: ordre amb el qual les instruccions o crides de funcions d'un programa són executades.

4.3 Aspectes generals i descripció de la Skill personalitzada

La Skill és la mateixa que ja s'ha explicat anteriorment, però, aquesta Skill és una mica diferent, ja que va ser la primera Skill creada i la metodologia és diferent respecte a una Skill creada mitjançant una SDK. La diferència és que en aquesta Skill no es té en compte els nous adjectius no utilitzats abans, ni es pregunta a l'usuari la raó per tenir l'estat anímic que l'usuari te i finalment no dona l'opció a preguntar directament a la Intel·ligència artificial, sinó que la mateixa Skill en fa una valoració i consulta directament en cas necessari.

Hi ha un únic intent en el projecte elaborat amb Voiceflow, a diferència del projecte elaborat amb la SDK d'Alexa i la Lambda d'AWS. Aquest intent consisteix en la lectura dels valors de son i estat anímic diaris cada cop que s'utilitza la Skill, un cop guardats, la Skill llegeix els últims deu registres guardats a la base de dades, si n'hi ha, i valora sí és necessari demanar consell a la intel·ligència artificial sobre els resultats calculats.

Cal destacar que hi ha petits detalls que no s'han pogut desenvolupar amb la plataforma Voiceflow de la Skill mencionada, que sí que s'han fet amb la funció Lambda d'AWS com ara afegir de forma automàtica nous adjectius o nous motius per haver tingut una bona o mala nit.

4.4 Requeriments de la Skill

Els requeriments de la Skill feta amb Voiceflow són els mateixos que els requisits de la Skill feta amb la SDK d'Alexa.

4.5 Decisions de disseny

Les decisions més importants d'aquest projecte estan relacionades amb la Base de Dades, ja que tot i tenir un bloc API que facilita la feina, la lectura dels valors guardats ha hagut de ser un a un causant que el projecte sigui poc escalable i més limitat. Aquest i els següents han sigut factors importants a tenir en compte:

- **Límit en els adjectius acceptats:** l'única forma d'afegir valors de forma automàtica seria guardant aquests en una base de dades i aquest mètode no és factible perquè es necessita especificar manualment adjectius a comparar de forma manual, no es pot comparar amb una llista o amb una base de dades com a tal.
- **Lectures dels últims deu dies:** A causa de la poca escalabilitat que hi ha en aquest mètode, s'ha hagut de limitar el nombre de registres a deu, ja que no seria factible de calcular un nombre més elevat com ara cent.
- **Ús de components:** Els components de Voiceflow són fluxos reutilitzables d'un o més blocs que poden ser utilitzats durant diferents parts del projecte. Permet crear un flux complex i reutilitzar-lo sense haver de fer-lo de nou a cada ús un mètode eficient per a poder mantenir el codi net i estalviar feina al programador és amb l'ús de components.
- **Guardar a la Base de Dades abans de llegir els deu valors:** En lloc de llegir els últims nou registres i calcular aquests junt amb els valors que estan sent utilitzats durant la sessió s'ha optat per a guardar primer els valors actuals de

la sessió a la Base de Dades abans de continuar amb els càlculs, ja que en cas de tindre algun error, el flux s'atura i no es podria evitar una futura errada.

4.6 Disseny

A Voiceflow no és necessari gestionar els directoris manualment, ja que la plataforma ofereix una solució tot en un, utilitzant una interfície gràfica, eficient i àgil.

4.6.1 Estructura

La plataforma organitza els elements clau del projecte de la següent manera:

Workspace del projecte o **"dessigner"**: en aquesta àrea del projecte és on es desenvolupen i disseny en les interaccions de veu. El flux de conversa es desenvolupa mitjançant una interfície d'arrossegar i deixar anar. Un espai on es pot organitzar i estructurar el propi projecte canvas, afegir diferents blocs per a les interaccions i configurar la seqüència lògica de les respostes de l'assistent de veu.

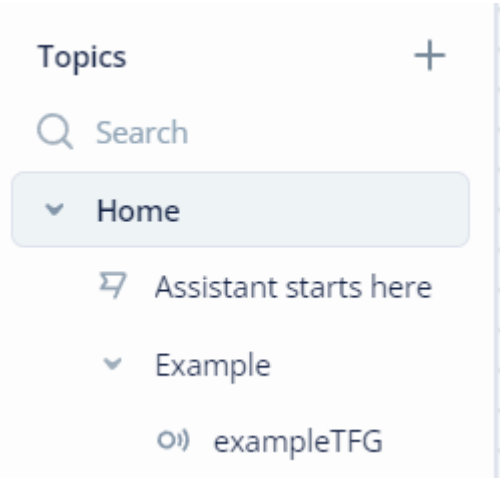


Figura 19. Workflows del projecte de Voiceflow

4.6.2 Flux de la Skill amb Voiceflow

A la figura 20 es pot observar quin seria el flux d'execució de la Skill, començant des de la invocació de la Skill fins a arribar al final de la Skill, passat per l'únic intent que té.

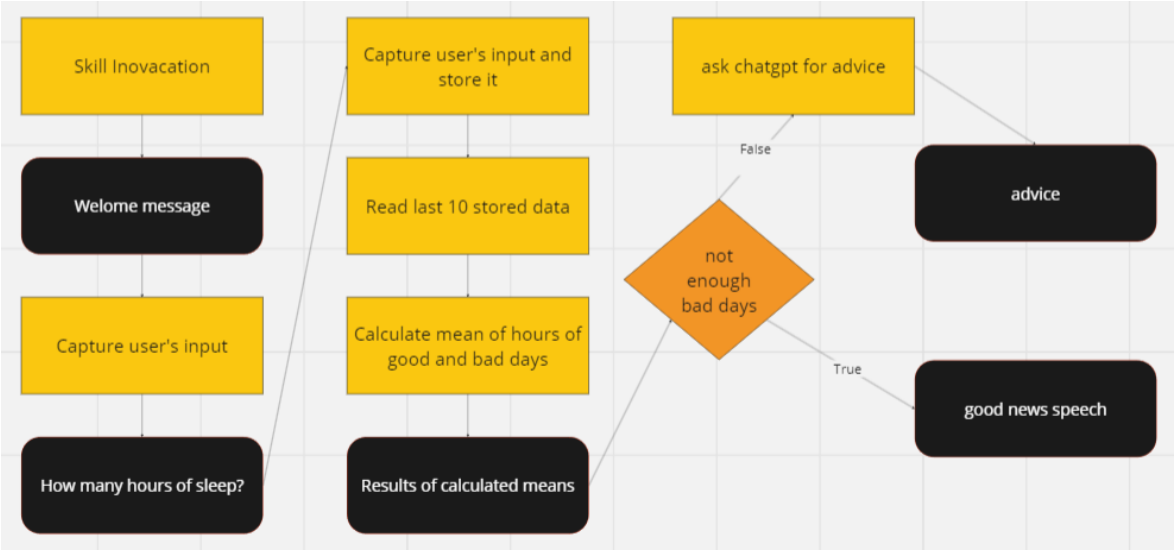
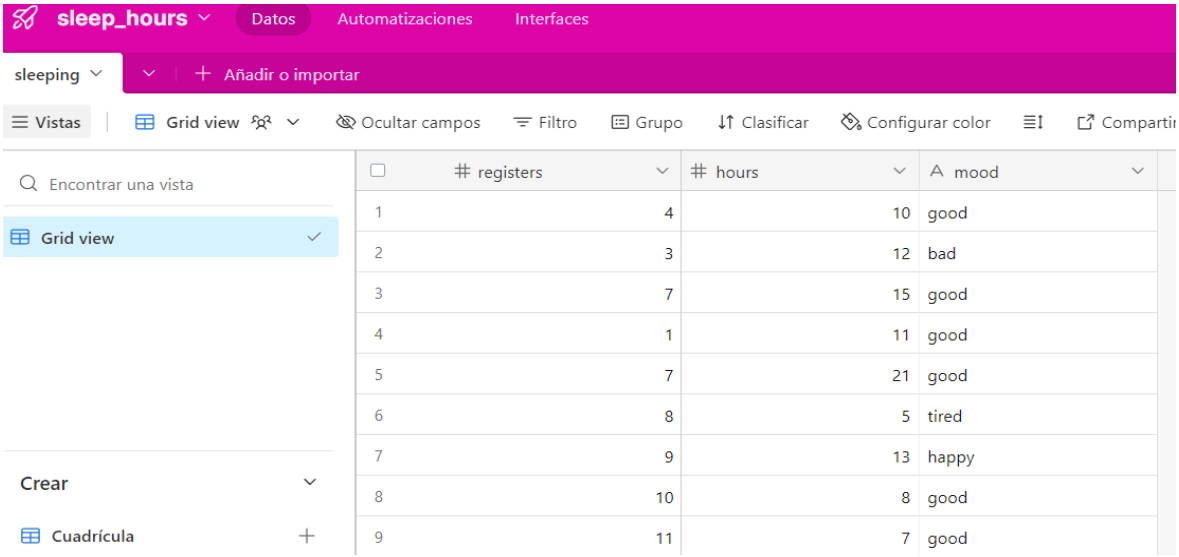


Figura 20. Flux de la Skill de Voiceflow

4.6.3 Base de dades AirTable

Per aquest projecte amb Voiceflow, s'ha utilitzat una Base de Dades relacional, organitza les dades en files i columnes, de la plataforma anomenada AirTable amb la qual es poden fer crides al seu API des de la plataforma Voiceflow i que és relativament fàcil d'entendre i de gestionar.

Airtable és una plataforma col·laborativa basada en el núvol i en el servei de base de dades. Aquesta plataforma permet als usuaris emmagatzemar i organitzar dades, combinant les funcions de base de dades, full de càlcul i programari de col·laboració.



	registers	hours	mood
1	4	10	good
2	3	12	bad
3	7	15	good
4	1	11	good
5	7	21	good
6	8	5	tired
7	9	13	happy
8	10	8	good
9	11	7	good

Figura 21. Taula de la base de dades d'AirTable

Entre d'altres, Airtable té alguns avantatges a destacar:

- **Visualitzacions flexibles:** la plataforma permet visualitzar les dades de diferents formes, deixant lliure elecció per a més comoditat de l'usuari.
- **Camps personalitzables:** es pot escollir molts tipus de variables per a cada camp, text, números, dates, múltiple opció, etc.

- **Integració d'Aplicacions:** Airtable s'integra en moltes altres aplicacions i eines, facilitant la seva automatització i sincronització de dades entre plataformes.
- **Col·laboració en temps real:** de la mateixa forma que es fa en el document de tipus ".dox" del Google Drive, es pot treballar en una mateixa base de dades varies persones de forma concurrent.
- **Historial de canvis:** s'ofereix un historial complet de revisions, on es pot observar qui ha fet quins canvis i quan els ha fet.
- **Aplicacions d'airtable:** la plataforma ofereix "miniaplicacions" que permeten ampliar les funcionalitats de la base de dades, com ara la creació de gràfics i enviar SMS d'entre altres.
- **Plantilles predissenyades:** plantilles ja creades pels casos més comuns com ara la planificació d'esdeveniments i la gestió d'inventari d'entre altres.

- **Filtres, Ordenació i Agrupació:** eines que permeten organitzar i segmentar les dades de forma més eficient, així com visualitzar l'informació que es rellevant.
- **Accessibilitat Mòbil:** Airtable ofereix el servei de la seva plataforma accessible per a aplicacions mòbils iOS i Android, donant encara més flexibilitat per poder accedir i modificar les bases de dades en qualsevol moment.
- **Seguretat i Control:** Es poden definir permisos específics per a cada col·laborador de la base de dades, tant d'edició com de visualització, també ofereixen funcionalitats de seguretat com el doble factor d'autentificació.

La base de dades s'utilitza per a poder gestionar les hores de son i l'estat anímic de l'usuari durant els últims dies, guardant un índex tipus comptador, per a saber els dies que l'usuari porta registrats.

Els elements de la taula són:

L'element anomenat "**Registers**": valor numèric que s'utilitza per a portar un compte dels dies registrats a la base de dades.

L'element anomenat "**Hours**": valor numèric registrat diàriament per a portar el control de les hores dormides a cada registre i que haurà de complir els requisits, major de 0 i menor de 24.

L'element anomenat "**Mood**": cadena de caràcters que s'usa per a classificar l'estat anímic diari de cada registre que s'ha completat.

Per a poder obtenir o guardar els valors de la base de dades hem d'usar blocs *"API"* del tipus *"Dev"*.

Figura 22. Com trobar bloc tipus API

Aquests blocs de la base de dades usen els mètodes "HTTP": "post" per a afegir valors a la base de dades i "get" per a agafar un o diferents valors. Si la trucada ha tingut èxit, el bloc declararà una sortida *"succes"*, en cas de fer una crida errònia o de no poder accedir a l'API, es declararà *"fail"*.

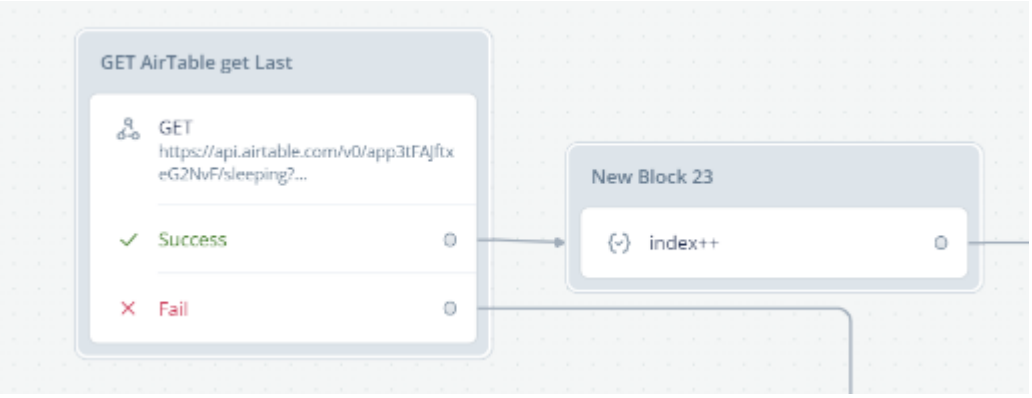


Figura 23. Bloc tipus API

A continuació s'explicarà breument la forma de fer la connexió amb la base de dades per a poder obtenir l'URL necessari per a l'ús d'un bloc tipus API.

Primer s'ha de saber quin és el nostre URL de la base de dades per a poder cridar aquesta correctament, s'ha d'anar a la icona d'ajuda de la part superior dreta de la interfície i seleccionar l'apartat de documentació de l'API. Aquesta documentació és creada de forma automàtica i personalitzada per a la base de dades, és a dir, que els exemples i fórmules que es troben a aquesta documentació són casos d'ús reals per a la Base de Dades sobre la que es treballa per a cada cas. Per aconseguir el URL mencionat anteriorment, s'ha de baixar a l'apartat d'autentificació i observar l'exemple que mostra la documentació, copiar i enganxar aquest URL al bloc al costat d'on es selecciona el tipus de crida "HTTP", posteriorment es podrà observar un exemple amb un bloc de tipus "get".

Figura 24. Com trobar URL de la taula a Voiceflow 1

Per a poder enllaçar la base de dades amb el bloc creat a Voiceflow s'ha d'anar al perfil del compte d'Airtable i seleccionar l'opció del centre de desenvolupadors i crear un *token*[21] personal per a poder autoritzar les crides del bloc d'API del projecte amb "get" i "post".

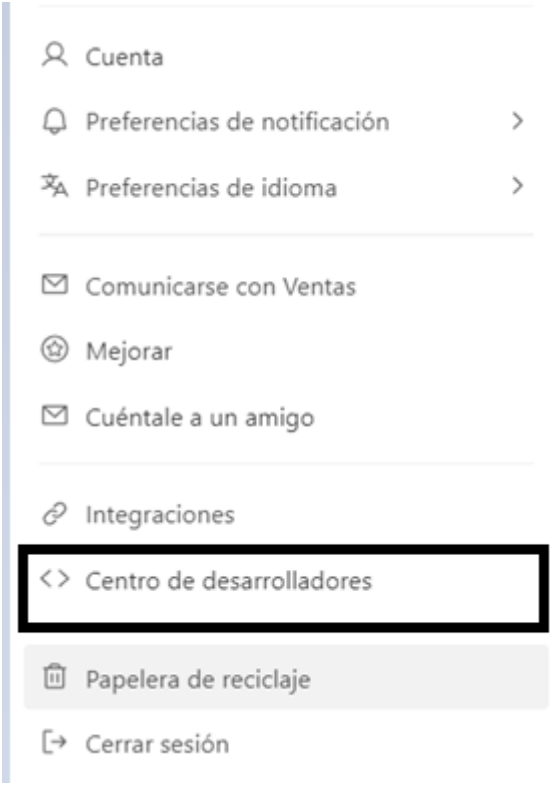


Figura 25. Com trobar URL de la taula a Voiceflow 2

Per a poder fer funcionar el bloc API amb la base de dades mitjançant la clau token personal i l'URL de la base de dades s'ha d'especificar quin tipus de crida es vol fer i el seu URL corresponent, tot seguit s'ha d'afegir a la crida un *header*[22] amb la clau del token personal.

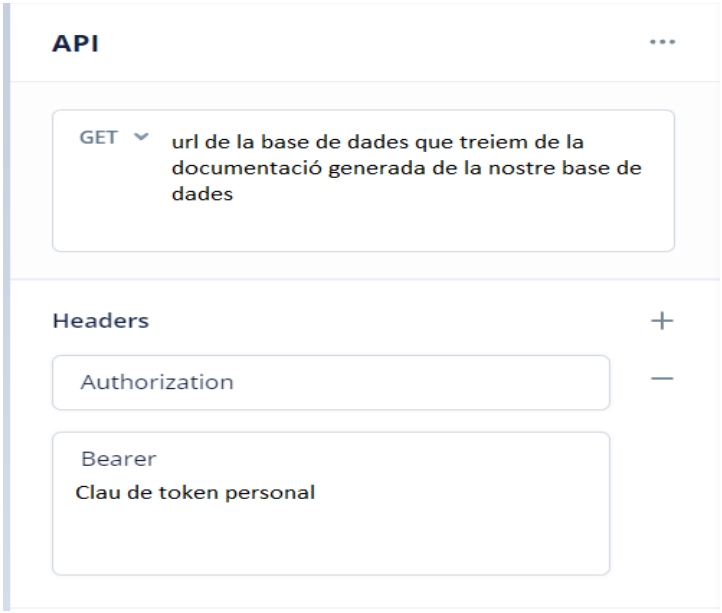


Figura 26. Com configurar un bloc API

21 Token: element únic d'un llenguatge de programació

22 Header: dades complementàries a l'inici d'un bloc de dades emmagatzemades o transmeses

4.6.4 Intent

L'intent en Voiceflow consisteix en la creació i l'ús de blocs i connexions a la interfície visual dintre d'un nou workspace. Amb aquests blocs es pot definir comportaments específics de l'assistent en funció de l'entrada que genera l'usuari.

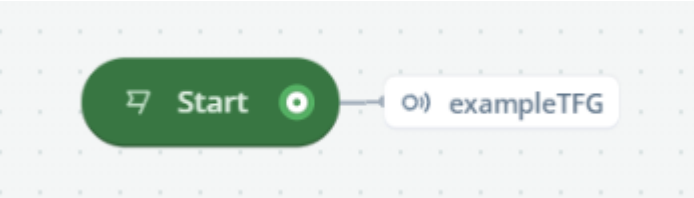


Figura 27. Crida al intent des del workspace principal

L'únic intent creat amb Voiceflow consisteix en: (veure figura 20)

- **Llegir els valors d'entrada** com ara les hores de son i l'estat anímic, comprovant que aquests son vàlids.
- **Guardar aquests valors** a la base de dades **d'Airtable**
- **Llegir els últims valors registrats a la base de dades**, tenint en compte els següents criteris: màxim 10 i mínim 1.
- **Fer una valoració dels resultats** obtinguts al calcular quants registres amb estat anímic positiu i negatiu hi ha.
- **El resultat final dependrà de la valoració** dels resultats anteriors, si el resultat és considerat com negatiu (cinc o més registres negatius sobre els últims deu), es demana a la intel·ligència artificial un consell per a l'usuari basant-nos en la mitjana d'hores dormides durant els registres considerats com a dolents. En cas contrari, Alexa donarà un missatge ja definit i tancarà la sessió.

4.6.5 Components

El projecte segueix el flux creat fins a trobar-se amb un component, aleshores el flux continuarà l'execució seguint els blocs del component fins a acabar-lo i continuarà pel bloc que segueix al component en el flux principal

4.6.6 Variables i Lògica

La gestió de les variables i la lògica de flux es realitza sense codi. Únicament es pot crear variables per emmagatzemar i recuperar dades d'usuari i afegir lògica amb blocs lògics, com el bloc de configuració o el d'elecció per adaptar dinàmicament les respostes de la Skill. Les variables permeten un emmagatzematge únicament temporal durant l'execució de la crida de la Skill. És per això que es requereix una Base de Dades per a guardar els registres diaris.

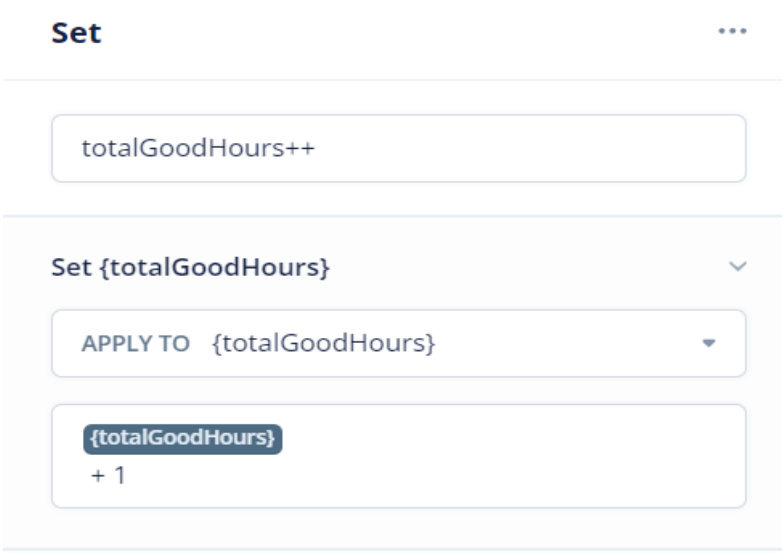


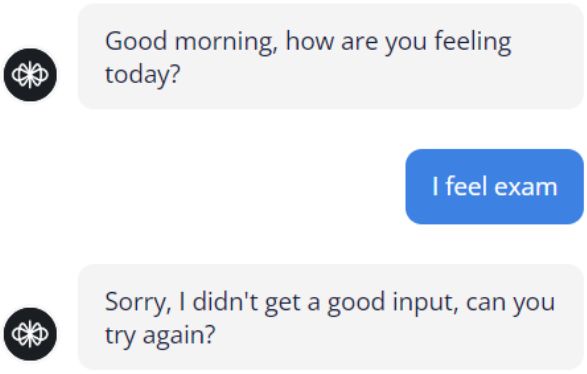
Figura 28. Exemple d'un bloc de configuració

4.6.7 Proves i Prototips

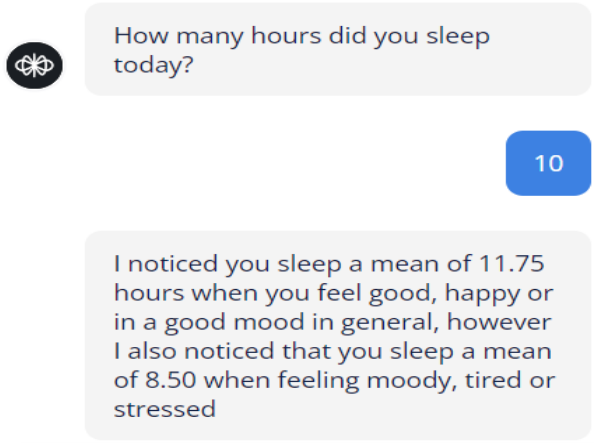
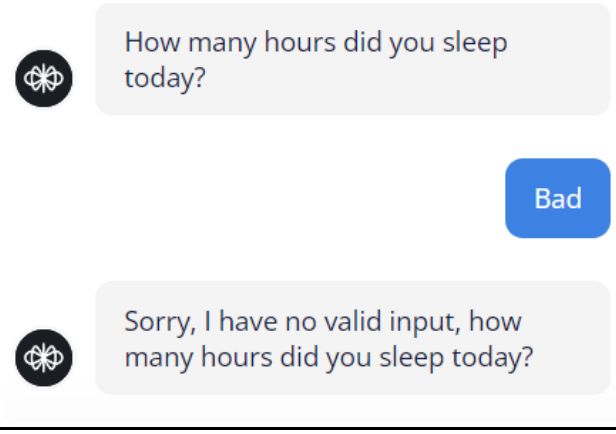
Voiceflow proporciona una eina ja integrada per a poder provar el projecte simulant una conversa, es pot utilitzat com a exemple per fer-se una idea de com funcionaria amb l'assistent amb el qual es vol treballar. Amb aquesta eina també es permet depurar el flux, permetent que es pugui observar els valors de cada variable durant el procés d'implementació.

4.6.8 Jocs de proves amb Voiceflow

Tests	Iteració	Resultat
Bon Input	Good morning, how are you feeling today? Good I'am glad to hear that from you How many hours did you sleep today?	OK

Input dolent		OK

En el següent joc de proves es mostra el següent pas després de respondre correctament l'anterior pregunta sobre l'estat anímic, tant amb la utterance correcta com amb la incorrecta:

Tests	Iteració	Resultat
Bon input		OK
Input dolent		OK

4.6.9 Exportació, Integració i Publicació

Un cop finalitzat el desenvolupament, Voiceflow permet exportar el projecte i integrar-lo amb altres plataformes com Alexa. Aquest procés simplifica la connexió entre la interfície de Voiceflow i els processos tècnics com la funció Lambda i altres configuracions necessàries. Això és possible, ja que Voiceflow *mapeja*[23] en el format ASK totes les converses, intents, utterances i components utilitzats del projecte.

Després de finalitzar el desenvolupament del projecte a Voiceflow, amb l'opció d'exportar el projecte, el projecte que és acceptat per la ASK d'Alexa i es podrà provar a la consola de desenvolupadors d'Alexa.

Finalment, Voiceflow s'integra amb diverses plataformes per agilitzar el desplegament. Per aplicacions de veu com Amazon Alexa i Google Assistant s'integra directament. En el cas d'Alexa, és necessari enllaçar el compte de desenvolupador d'Amazon i configurar les metadades de la Skill.

23 Mapeja: l'acció d'associar diversos elements amb una clau com a valor

5 Diferències entre Lambda/SDK i Voiceflow

5.1 Visió general de la SDK d'Alexa i Lambda

El Kit de Desenvolupament d'Habilitats d'Alexa (ASK) i el Servei de Veu d'Alexa (AVS) formen la base de la SDK d'Alexa. L'ASK permet als desenvolupadors crear habilitats que amplien les capacitats de l'Alexa, mentre que l'AVS proporciona eines per integrar l'Alexa en dispositius. Els desenvolupadors poden aprofitar una varietat de llenguatges de programació, incloent-hi Node.js, Python i Java, oferint flexibilitat en la implementació i la integració en altres sistemes utilitzant la Lambda d'AWS.

5.2 Visió general de Voiceflow

Voiceflow proporciona una interfície visual, adoptant un enfocament sense codi/poc codi per al desenvolupament d'aplicacions de veu. El principal objectiu és simplificar la creació de fluxos conversacionals sense la necessitat de codificació. La plataforma està destinada a desenvolupadors i no desenvolupadors, permet el disseny d'interaccions de veu complexes mitjançant una interfície fàcil d'utilitzar.

5.3 Facilitat d'ús

En canvi, Voiceflow es presenta com una plataforma accessible, permetent als usuaris dissenyar i prototipar aplicacions de veu de manera intuïtiva de forma simple i directe amb menys passes entremig. Aquesta simplicitat pot capacitar als no desenvolupadors per contribuir activament al procés de creació d'aplicacions de veu.

La SDK d'Alexa pot requerir que els desenvolupadors entenguin els matisos del desenvolupament de les habilitats d'Alexa, té una corba d'aprenentatge de programació i de l'ecosistema AWS per a gran part dels desenvolupadors, hi ha un procés de preparació perquè funcioni el projecte com per exemple la connexió de Lambda amb la Skill d'Alexa, els permisos necessaris per a accedir a les BBDD des de la Lambda, etc.

5.4 Flexibilitat i personalització

Voiceflow destaca per la seva aproximació sense codi, que permet als desenvolupadors i no desenvolupadors crear interaccions de veu mitjançant una interfície visual. Mitjançant l'ús de blocs lògics, és possible implementar una lògica personalitzada sense codificar directament. Això ofereix adaptabilitat dinàmica al programar i llegir les entrades de l'usuari, sense la necessitat d'entendre el funcionament d'estructures específiques.

La Lambda i la SDK de l'Alexa adopten un enfocament basat en codi, permetent als desenvolupadors codificar interaccions de veu amb gran detall i personalització. Per exemple, es pot utilitzar la Lambda per integrar-se amb altres serveis externs o bases de dades, oferint una personalització gran en funció de les necessitats específiques del projecte. També es poden fer ús de llistes i estructures, de la mateixa forma que també es poden comparar aquestes.

5.5 Gestió de Slots i variables

A Voiceflow, la gestió de slots es realitza mitjançant l'ús de variables. Per exemple, en una aplicació de reserva de vols, es podria utilitzar una variable per emmagatzemar el destí seleccionat per l'usuari, com "Barcelona" o "Madrid". Aquesta gestió intuïtiva facilita la personalització de les interaccions i la manipulació dinàmica de dades basades en l'entrada de l'usuari. Això no obstant, per a casos com obtenir els valors de son i estat anímic dels últims registres, aquests resultats únicament poden ser obtinguts un per un per a cada valor, el que fa la feina una mica més tediosa, repetitiva i poc escalable. A continuació es pot observar una imatge on es mostra un exemple del mateix cas parlat, on es pot llegir de la base de dades únicament un valor a cada variable, necessitant-se en aquest cas doncs deu variables per a guardar deu resultats.

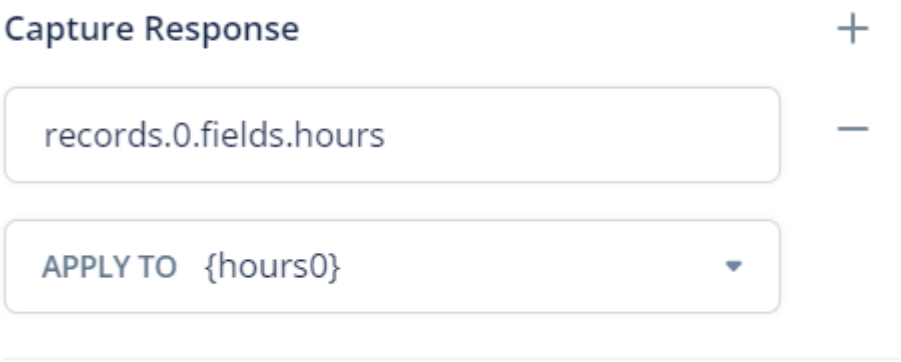


Figura 29. Com capturar un valor de la base de dades AirTable

Amb la SDK de l'Alexa, es pot implementar una gestió de slots més granular i controlada. Per exemple, en una aplicació de consulta de clima, es podria codificar un slot específic per a "ciutats preferides" i usar lògica addicional per gestionar respostes basades en les ciutats guardades. Aquest nivell de control permet una integració més profunda amb altres serveis i una personalització més gran de les interaccions, a més a més, es pot treballar amb més d'un valor a la vegada i utilitzar taules, en comptes d'una variable per a cada valor.

5.6 Disseny i experiència del programador

La interfície visual *"d'arrossegar i deixa anar"* de Voiceflow facilita el disseny d'interaccions de veu, permetent als dissenyadors esquematitzar i iterar sobre les interaccions amb facilitat. Aquest enfocament sense codi és particularment beneficiós per equips multidisciplinaris i projectes que requereixen una aproximació àgil i col·laborativa de forma ràpida.

Amb la SDK de l'Alexa, els dissenyadors podrien implementar interaccions més sofisticades amb una codificació directa i precisa. Per exemple, en una aplicació educativa, es podria utilitzar codi per gestionar l'estat d'una lliçó, guardar progrés de l'usuari i proporcionar retroalimentació detallada i directe. Aquest enfocament ofereix una flexibilitat gran per dissenyar experiències d'usuari més riques i interactives.

5.7 Eines de proves i depuració

Les funcions integrades de proves a Voiceflow permeten als desenvolupadors simular interaccions i identificar problemes directament a la interfície visual. Això agilitza el procés de desenvolupament i facilita la identificació de problemes abans de la implementació. Això no obstant, no hi ha una consola on es pugui veure errors del projecte, com els anomenats logs.

La SDK de l'Alexa també proporciona eines de proves, incloent-hi simuladors i emuladors, que permeten als desenvolupadors validar el comportament de l'aplicació abans de llançar-la al públic. Aquestes eines estan integrades dins de l'entorn de desenvolupament, facilitant la identificació precoç de possibles problemes.

La lambda, com a pàs previ a l'execució a la SDK, té una consola on podem veure els errors del projecte, tant al provar la Skill com en desplegar-la a la SDK.

5.8 Consideracions econòmiques

Els costos associats amb l'ús de Voiceflow en aquest cas han estat nuls, ja que s'ha utilitzat la versió gratuïta d'aquesta plataforma, això no obstant, poden variar segons l'ús i els plans contractats. És important avaluar aquesta estructura de preus per assegurar-se que s'ajusti als requisits econòmics del projecte, amb la possibilitat d'escollir plans que s'adapten a les necessitats específiques, ja que les capacitats de l'eina són limitades en la versió gratuïta.

L'ús de la SDK de l'Alexa és gratuït, no obstant la Lambda pot tenir costos associats amb el consum de recursos computacionals i l'ús de serveis. És essencial avaluar com aquest consum pot impactar els costos operatius del projecte a llarg termini. Alguns exemples d'aquests costos poden ser una àmplia utilització dels serveis de la Base de Dades.

5.9 Seguretat i conformitat

Voiceflow: Mesures de Seguretat i Conformitat

Tot i que Voiceflow no està tan extensament documentat com la SDK de l'Alexa, la plataforma incorpora mesures de seguretat per protegir les interaccions de veu i les dades de l'usuari. Voiceflow es compromet a mantenir la privacitat i la seguretat de les dades de manera efectiva.

Privacitat de Dades: Voiceflow assegura la confidencialitat de les dades de l'usuari, evitant l'accés no autoritzat i garantint que les interaccions de veu estiguin protegides.

Seguretat de la Plataforma: La plataforma implementa mesures de seguretat en els seus servidors per prevenir amenaces cibernètiques i altres vulnerabilitats que puguin comprometre la integritat del sistema.

Conformitat amb Normatives: Tot i que pot no proporcionar la mateixa documentació exhaustiva que l'SDK de l'Alexa, Voiceflow es compromet a mantenir-se actualitzat amb les normatives de privacitat i seguretat rellevants en el desenvolupament de la seva plataforma.

SDK d'Alexa: Rigoroses Mesures de Seguretat

La SDK d'Alexa està sotmesa a estàndards de seguretat rigorosos i conformitat amb normatives de privacitat. Aquesta plataforma ofereix un conjunt detallat de funcionalitats de seguretat, essencial per a aplicacions que gestionen dades delicades de l'usuari.

Gestió de Dades delicades: la SDK d'Alexa inclou protocols estrictes per gestionar dades delicades com la identitat de l'usuari, assegurant-se que aquesta informació estigui adequadament protegida, com per exemple l'autenticació robusta per accedir a les dades de l'usuari així com mecanismes per assegurar que les dades es mantenen privades i segures durant tot el procés de manipulació.

Xifratge de Comunicacions: les comunicacions entre l'aplicació i els servidors d'Alexa estan xifrades utilitzant protocols de seguretat avançats com HTTPS, garantint la seguretat de les interaccions de veu i minimitzant els riscos d'intercepció no autoritzada.

Conformitat amb Estàndards de Privacitat: la SDK d'Alexa es manté actualitzat amb les normatives de privacitat com el Reglament General de Protecció de Dades (RGPD) i altres directives que asseguren el tractament adequat de la informació personal. Això inclou mesures de lectura, emmagatzematge i processament de dades d'acord amb les normatives vigents.

Monitoratge i Auditories: la SDK d'Alexa inclou funcionalitats de monitoratge i registre d'activitats per assegurar que les operacions és realitzen de manera segura i transparent.

AWS Lambda: Rigoroses Mesures de Seguretat

Aquesta plataforma ofereix un conjunt complet de funcionalitats de seguretat, essencial per a aplicacions que processen dades delicades d'usuaris.

Gestió de Dades Delicades: la Lambda implementa protocols estrictes per a gestionar dades delicades com la identitat de l'usuari. També es permet configurar entorns d'execució segurs amb l'ús de "AWS Identity and Acces Management" o (IAM), per a la gestió de permisos i rols, garantint que únicament els recursos i usuaris autoritzats poden accedir a les funcions assignades.

Xifratge de Dades: Les comunicacions entre la Lambda i els altres serveis d'AWS estan xifrades mitjançant HTTPS. També s'ofereix un xifrat per a les dades emmagatzemades, utilitzant claus gestionades per "AWS Key Management Service" o (KMS).

Conformitat amb Estàndards de Privacitat: de la mateixa forma que la SDK, Lambda està dissenyada per complir amb diverses normatives de privacitat, com el Reglament General de Protecció de Dades (RGPD), d'entre altres. AWS proporciona documentació i eines per ajudar els desenvolupadors a complir amb aquests estàndards.

Monitoratge i Auditories: Lambda integra serveis de monitoratge com "AWS CloudTrail" i "AWS CloudWatch", que permeten el seguiment de les activitats i execucions de les funcions. Aquestes eines proporcionen visibilitat i registre detallat de les accions realitzades als serveis ajudant a detectar i respondre de forma ràpida als possibles problemes.

5.10 Problemes trobats durant el desenvolupament del projecte

Alguns inconvenients que hi ha hagut durant el procés d'implementació d'aquest projecte amb la SDK i la Lambda han estat:

- **Cobrament d'AWS**, ja que mai s'havia sobrepassat el percentatge d'us gratuït que et donen, però a causa de certes circumstàncies o la creació de molts *logs*[24] per errors al compilar, es generen alarmes al servei de *"cloudwatch"* d'AWS, i si aquestes es van creant i executant sense que ens adonem, es pot passar l'ús gratuït i començar a tindre cobraments que no esperem.
- **L'ús d'una API key a openai** s'ha de pagar per ús, on el mínim ingrés són cinc euros.
- **Slots personalitzats** són una guia per Alexa i no asseguren que els valors llegits siguin valors que s'han definit als slots, sinó potser semblants. Aquest cas concret es resol en la gran part en canviar la sensibilitat de *"fallbackIntentSensitivity"* a alta perquè quan s'escrigui alguna cosa que no es casi exacte o exacte al que hem definit, aquest intent, fallbackIntent, s'executi.
- **SDK i Lambda complexes per depurar**, ja que amb qualsevol altre editor de codi es poden observar els errors i avisos més directament, per exemple a la lambda hi ha logs dels resultats on es pot veure amb més claredat aquests errors i on si específica la línia que provoca el problema, l'únic mètode de depuració es executant el codi i provar-lo a la Lambda amb un fitxer de tipus json, que es pot extreure cada vegada que s'executi un intent d'Alexa.

Inconvenients amb el Voiceflow:

- **L'ús de blocs condicionals** en comparar amb els valors de l'estat anímic per exemple o en guardar els valors de la base de dades d'Airtable on no es podia guardar els registres de les hores en una taula i per a cada valor s'ha hagut de crear una variable, fent-ho poc escalable i molt tediós.
- **Metode de treball amb les variables:** referenciant al punt 5.6 Gestió de Slots i variables, treballar amb variables que únicament poden guardar un valor es molt tediós i poc escalable.
- **OpenAi gratuïta amb límit:** Voiceflow, en la seva versió gratuïta permet un ús de 100.000 tokens, unes 75.000 paraules, d'un API key a openai.

5.11 Conclusió de la comparativa

En el procés de selecció entre Voiceflow i l'SDK de l'Alexa, és essencial considerar diverses variables que poden influir en la decisió final. Cada plataforma presenta avantatges i desavantatges que s'adeqüen a diferents contextos i objectius de desenvolupament.

Quan Utilitzar Voiceflow:

- **Desenvolupament Ràpid i Intuïtiu:** Voiceflow destaca en situacions en què es requereix un desenvolupament àgil i una interfície intuïtiva amb integracions simples. Amb una aproximació sense codi, és ideal per a equips que prioritzen la rapidesa en la creació de prototips i iteracions ràpides.

24 Logs: es un registre d'esdeveniments o accions que es produeixen dins d'un programa, sistema o aplicació.

- **Col·laboració Multidisciplinària:** La interfície visual de Voiceflow facilita la col·laboració entre membres de l'equip amb diferents habilitats, incloent-hi desenvolupadors, dissenyadors i experts en contingut de veu. Això pot accelerar el procés de desenvolupament amb una comprensió compartida de les interaccions de veu.
- **Projectes Amb Mitjà o Baix Nivell de Complexitat:** En escenaris on els requisits de complexitat són moderats i no es requereix una personalització extensiva mitjançant codi, Voiceflow pot ser una elecció eficient. És especialment apte per a aplicacions de veu estàndard o projectes amb necessitats relativament senzilles.

Quan Utilitzar l'SDK de l'Alexa amb la AWS Lambda:

- **Control Granular i Personalització:** Si el projecte exigeix un control detallat i personalització pel que fa a codi, la Lambda és l'opció preferida. Per a aplicacions amb lògica complexa o integracions específiques, la capacitat de codificar directament que ofereix una flexibilitat significativa com ara l'ús de diferents llenguatges de programació, oferint implementar funcionalitats avançades.
- **Necessitats de Variables i *Arrays*[25] detallades:** Si el projecte requereix comprovació detallada de variables, manipulació de taules i altres operacions de baix nivell, l'SDK de l'Alexa proporciona les eines i la flexibilitat necessàries per gestionar aquestes necessitats amb profunditat.
- **Escalabilitat:** Lambda escala automàticament en funció de la càrrega, gestionant milers de sol·licituds simultànies sense necessitat d'intervenció manual.

Consideracions Addicionals:

Aspectes Ètics i Seguretat dels Usuaris:

És crucial considerar aspectes ètics relacionats amb la seguretat dels usuaris, independentment de la plataforma triada. Garantir una recopilació de dades responsable, transparència en les pràctiques de privacitat i assegurar-se que l'experiència de veu sigui segura per als usuaris és imperatiu.

En resum, la selecció entre Voiceflow i la SDK de l'Alexa dependrà de les necessitats específiques de cada projecte. Voiceflow és ideal per a equips o desenvolupadors que busquen una eina fàcil d'utilitzar amb una corba d'aprenentatge suau, que prioritzen la rapidesa en el desenvolupament mentre que la SDK de l'Alexa i la Lambda d'AWS és la millor opció per a projectes que requereixes alta flexibilitat, funcionalitats avançades i escalabilitat, però requereix un coneixement i experiència prèvia amb programació i l'ecosistema AWS.

25 Arrays: és una estructura de dades que consisteix en una col·lecció d'elements de la mateixa mida

6 Conclusions generals del projecte

He aconseguit desenvolupar l'objectiu que em vaig proposar mitjançant aquests dos projectes on he après a crear una Skill d'Alexa des de zero. He implementat una Skill dedicada a l'horari de son amb assistència de la intel·ligència artificial realitzada mitjançant dos mètodes diferents.

També vull fer èmfasi en la importància que ha tingut crear una Skill de prova mitjançant dues metodologies diferents com ara VoiceFlow abans de fer una Skill més complexa i definitiva. Aquesta és una pràctica recomanada per a qualsevol projecte de desenvolupament de programari. La creació d'una Skill de prova permet identificar, corregir i perfeccionar errors abans que afectin la Skill final.

En finalitzar el treball, he arribat a la conclusió que prefereixo el desenvolupament i resultat de la Skill creada amb la SDK i la AWS Lambda, ja que s'obté una Skill més completa.

- Que he après?
  - En aquest projecte, hi havia àrees en les quals tenia una formació limitada o gairebé nul·la, com l'ús de la AWS Lambda durant el desenvolupament d'una Skill d'Alexa ho he trobat a faltar.
  - Considero que el meu aprenentatge ha sigut molt més notori en el projecte fet amb la SDK i la Lambda, on he après el funcionament de serveis d'AWS com la Lambda i els conceptes bàsics d'Alexa.
  - També he pogut repassar i millorar el meu coneixement de Python.
- En que m'ha ajudat el grau universitari?
  - Aprendre a aprendre, crec que es l'aspecte més important a destacar durant el meu desenvolupament universitari, l'aprenentatge a com cercar documentació, com organitzar-se, com plantejar un projecte, etc.
  - Programació amb Python, gràcies a les classes de Sistemes Distribuits on s'ha treballat el llenguatge Python, he pogut realitzar aquest projecte sense haver d'aprendre a programar des de zero.
  - Coneixements basics de Bases de Dades, gràcies a les nocions bàsiques i exemples teòrics de Bases de Dades donades durant aquest grau, he pogut aprendre i utilitzar la Base de Dades DynamoDB de forma ràpida i adequada.

En resum, aquest projecte és un exemple de com la tecnologia es pot utilitzar per abordar problemes rellevants i millorar la vida de les persones de diferents formes, tant pels més experts com pels que no ho són tant, tots poden aportar el seu petit gra de sorra. El desenvolupament de **sleeping helper** és un prototip ambiciós que té el potencial d'ajudar les persones a millorar la seva qualitat de son i benestar general, no obstant, el desenvolupament d'aquest projecte s'ha centralitzat en diferenciar i explicar el procés de creació d'una petita Skill i documentar les passes a realitzar per a futurs desenvolupadors i per a mi mateix.

Estic segur que amb la meva dedicació i esforç, podré continuar millorant **sleeping helper** i fer que sigui una eina encara més valuosa per a les persones que l'utilitzin.

7 Treball futur per a la Skill

El futur desenvolupament d'aquesta Skill implica diverses etapes crucials. En primer lloc, és essencial fer proves amb diversos usuaris per reforçar la robustesa de la idea de la Skill. Aquestes proves permetran identificar punts febles que actualment existeixen, com la possible resposta inadequada de la Skill a paraules o frases específiques, que podrien provocar errors. L'objectiu és millorar l'experiència de l'usuari mitjançant l'afegiment d'expressions comunes que podrien haver-se obviat inicialment, evitant així una proliferació d'errors durant la implementació.

A més a més, s'ha de fer una anàlisi de dades del son més profund i complex per comprendre millor el comportament dels usuaris i adaptar les respostes de la Skill en conseqüència.

Pel que fa a les millores que es podrien realitzar en el futur, vull destacar les següents:

- Connexió amb altres Skills d'Alexa: La connexió amb altres Skills d'Alexa podria permetre que **sleeping helper** ofereixi una experiència més integrada i completa a l'usuari. Per exemple, es podria connectar la Skill amb una Skill d'alarma perquè **sleeping helper** s'activi automàticament després que l'alarma soni, així també es podria obtenir les hores de son sabent l'hora d'anar a dormir, en comptes de preguntar el nombre d'hores dormides, tot i que aquesta acció podria donar problemes, ja que no tothom es desperta amb la primera alarma, aquest pas només es pot aconseguir un cop la Skill és publicada i aprovada per l'equip d'Amazon segons les seves polítiques, així no s'ha pogut fer encara per a aquesta Skill.
- Millora dels mètodes de seguiment: L'ús de mètodes de seguiment més precisos podria permetre que **sleeping helper** proporcioni a l'usuari informació més precisa sobre els seus hàbits de son. Per exemple, es podria utilitzar dades d'ubicació o incorporar informació relacionada amb les condicions meteorològiques o sonores per a millorar encara més l'experiència de l'usuari.
- Utilitzar més la IA: En un futur pròxim, ja es podrien fer implementacions més potents de la IA i de reconeixement de veu per a poder distingir quan l'usuari ha tingut un bon dia o no ja fins i tot per com respon a les preguntes amb el to de la resposta.

Finalment, es requereix l'elaboració completa de les descripcions, títols i dades adequades per iniciar el procés de publicació de la Skill. Aquesta publicació posteriorment passarà per una revisió minuciosa per part d'Amazon, un procés que, segons les indicacions proporcionades al seu lloc web, implica una avaluació manual i lenta.

8 Referències

[1] Les 60 skills de Alexa mes útils The 60 Most Useful Alexa Skills of 2023 (lifewire.com)

[2] Estudi d'usuaris que utilitzen els assistents virtuals diàriament https://www.marketingdirecto.com/digital-general/mobile-marketing/estudio-revela-22-usuarios-emplean-asistentes-virtuales-todos-los-dias

[3] Aprenent les funcionalitats de Alexa (243) Getting Started With Alexa | Part 2 | Capturing a user's intent - YouTube

[4] Entenent els slots i obtenint els seus valors python - Understanding slots and getting its values in Alexa Skills Kit - Stack Overflow

[5] Aprèn a programar Alexa Learn to Program Alexa | Codecademy

[6] Teoria de capes d'AWS https://dashbird.io/knowledge-base/aws-lambda/lambda-layers/

[7] Teoria de Slots https://medium.com/voiceflow/tips-and-gotchas-using-alexa-custom-slots-b88f97f26b06

[8] Creació d'una lambda https://www.youtube.com/watch?v=CagheYRr9oI&list=PLZmfj7vJb0aFjFHeFIQLsEow7QYAagKeA&index=1

[9] Teoria d'Apis en Voiceflow https://learn.voiceflow.com/hc/en-us/articles/9262787577613-API

[10] Teoria + exemples de DynamoDB y S3 AWS https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/programming-with-python.html

[11] Teoria de roles d'IAM https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html

[12] Com utilitzar DynamoDB amb la Lambda https://www.youtube.com/watch?v=CjVPMocEECM

[13] Exemple d'api en Voiceflow https://www.youtube.com/watch?v=C6-V35uWscM

[14] Preu IA https://gptforwork.com/tools/openai-chatgpt-api-pricing-calculator

[15] Curs DyanmoDb Amazon DynamoDB - A Crash Course (Demos Included) - YouTube

[16] Que es Alexa https://developer.amazon.com/es-ES/alexa/alexa-skills-kit

[17] ALEXA SDK https://developer.amazon.com/es-ES/docs/alexa/sdk/alexa-skills-kit-sdks.html

[18] Com utilitzar les possibilitats que et dona la Skill d'alarma d'Alexa
https://www.digitaltrends.com/home/alexa-alarm-clock/

[19] En aquest treball s'ha fet ús de la Intel·ligència Artificial ChatGPT ChatGPT (openai.com)

9 Codi

9.1 Lambda

9.1.1 Importacions i declaracions globals

```
import openai
import boto3
import logging
import json
openai.api_key = ******************************
s3=boto3.client('s3')
bucket_name='datalambdatfg'
file_name_adjectives='adjectives.json'
file_name_reasons='reasons.json'
logger = logging.getLogger()
logger.setLevel(logging.INFO)
dynamodb = boto3.resource('dynamodb')
table_name = 'SleepData'  # Replace with your DynamoDB table name
table = dynamodb.Table(table_name)
tableTotal_name='TotalSleepData'
tableTotal = dynamodb.Table(tableTotal_name)
```

9.1.2 Lambda Handler:

```
def lambda_handler(event, context):
    session_attributes = event['session']['attributes'] if 'attributes'
in event['session'] else {}
    logger.info('Alexa event received: {}'.format(event))
    if event['request']['type'] == 'LaunchRequest':
        return on_launch(event['request'],
event['session'],event,session_attributes)
    elif event['request']['type'] == 'IntentRequest':
        return on_intent(event['request'],
event['session'],event,session_attributes)
    elif event['request']['type'] == 'SessionEndedRequest':
        return on_session_ended(event['request'], event['session'])

    return build_response("Try again please, I did not undertund the last
utterance", False,session_attributes)
```

9.1.3 Funció de llançament de la Skill:

```
def on_launch(request, session,event,session_attributes):
    return build_response("Welcome! How are you feeling today?", False,
session_attributes)
```

9.1.4 Funció per cercar l'intent que es vol utilitzar:

```
def on_intent(request, session,event,session_attributes):
    intent_name = request['intent']['name']
    if intent_name == 'useChatGPTIntent':
        return handle_ChatGPT_intent(request,event,session_attributes)
    elif intent_name == 'moodIntent':
        return handle_mood_intent(request,event,session_attributes)
```

```
    elif intent_name == 'SleepHoursIntent':#sleep hours // i slept 12
hours today
        return
handle_sleep_hours_intent(request,event,session_attributes)
    elif intent_name == 'ReasonIntent':
        return handle_reasons_intent(request,event,session_attributes)
    elif intent_name == 'AMAZON.StopIntent':
        return build_response("Got it, closing the skill then,Goodbye!",
True,session_attributes)
    elif intent_name == 'AMAZON.CancelIntent':
        return build_response("Cancelled.", True,session_attributes)
    elif intent_name == 'AMAZON.HelpIntent':
        return build_response("You can say hello to me!",
False,session_attributes)
    elif intent_name == 'AMAZON.FallbackIntent':
        return build_response("Sorry, I did not understand that, can you
please try again?", False,session_attributes)
    else:
        # Fallback handling for unrecognized intents
        return handle_fallback_intent(request, event, session_attributes)
```

9.1.5 Funció de fi de sessió:

```
def on_session_ended(request, session):
    logger.info('Session ended with reason:
{}'.format(request['reason']))
    return None
```

9.1.6 SleepHoursIntent:

```
Def handle_sleep_hours_intent(request,event,session_attributes):

    slots = request['intent']['slots']
    # Extract the value of the sleep_hours slot
    sleep_hours_slot = slots.get('sleep_hours', {})
    sleep_hours = sleep_hours_slot.get('value')

    mood = session_attributes.get('mood', 'default')
    #necesitamos el mood aqui  para hacer las operaciones
    if sleep_hours is None:
        return build_response("Sorry I did not understand that, How are
you feeling today?", False,session_attributes)

    try:
        sleep_hours = int(sleep_hours)
    except ValueError:
        return build_response("Hours should be a number.", False)

    session_attributes['hours'] = sleep_hours

    #return build_response("I noticed that in days where you feel good
you slept a mean of {} hours whereas when you feel bad or tired you slept
a mean of {} hours".format(goodMoodMean,badMoodMean), True)
    return build_response("And what is the reason for you feeling like
{}".format(mood), False,session_attributes)
```

9.1.7 moodIntent:

```
def handle_mood_intent(request,event,session_attributes):
    slots = request['intent']['slots']
    mood_slot = slots.get('mood', {})
    mood = mood_slot.get('value')
    # Extract user input from the event

negative_adjectives=get_adjective_list_from_s3('negative_adjectives',file
_name_adjectives)

positive_adjectives=get_adjective_list_from_s3('positive_adjectives',file
_name_adjectives)

neutral_adjectives=get_adjective_list_from_s3('neutral_adjectives',file_n
ame_adjectives)

    if mood not in negative_adjectives and mood not in
positive_adjectives and mood not in neutral_adjectives:
        prompt=generate_gpt_response([],"hi, is this word:{}, an
adjective? replying just by saying yes or no".format(mood))
        if 'Yes' in prompt or 'yes' in prompt:
            prompt=generate_gpt_response([],"do you consider that
adjective: {} as positive, negative or neutral? reply just by saying one
of the 3 options".format(mood))
            if 'positive' in prompt.lower():
                #add in positive list

add_adjective_to_list('positive_adjectives',mood,file_name_adjectives)
            elif 'negative' in prompt.lower():
                #add in negative list

add_adjective_to_list('negative_adjectives',mood,file_name_adjectives)
            else:
                #add in neutral list

add_adjective_to_list('neutral_adjectives',mood,file_name_adjectives)
        else:
            return build_response("sorry, try another adjective",
False,session_attributes)
    session_attributes['mood'] = mood
    return build_response("How many hours did you sleep?",
False,session_attributes)
```

9.1.8 useChatGPTIntent:

```
def handle_ChatGPT_intent(request,event,session_attributes):
    mood = session_attributes.get('mood', 'default')
    hours = session_attributes.get('hours', 'default')
    reason = session_attributes.get('reason', 'default')
    # Extract user input from the event
    if mood =='default' or hours =='default':
        #error
        return build_response("I am sorry, first you need to follow the
format of the skill to fullfill the necessary values to create the promp
for chat GPT, try to say how you feel", False,session_attributes)
```

```
    elif reason is 'default':
        #prompt sin la reason
        prompt ="hi! do u have any advise for a person who has slept like
{} after {} hours of sleep"
    else:
        #prompt con la reason
        prompt ="hi! do you have any advise for a person who has slept {}
after {} hours of sleep and it was due to {}".format(mood,hours,reason)
    # Generate GPT response with an empty chat history


    gpt_response = generate_gpt_response([], prompt)

    return build_response(prompt, False,session_attributes)
```

9.1.9 reasonIntent:

```
def handle_reasons_intent(request,event,session_attributes):
    slots = request['intent']['slots']
    reason_slot = slots.get('reason', {})
    reason = reason_slot.get('value')

    reasons_list=get_adjective_list_from_s3('reasons',file_name_reasons)
    mood = session_attributes.get('mood', 'default')
    if reason is None:
        reason=''
    if reason not in reasons_list:
        prompt=generate_gpt_response([],"hi, is this word:{}, an excuse
or a reason to have a good or bad sleeping night? replying just by saying
yes or no".format(mood))
        if 'Yes' in prompt or 'yes' in prompt:
            add_adjective_to_list('reasons',reason,file_name_reasons)
        else:
            return build_response("Sorry I did not understand that, What
is the reason  for you feeling like {} ?".format(mood),
False,session_attributes)
    postTOTALDDBB(session_attributes);
    hours = session_attributes.get('hours', 'default')
    count_id = session_attributes.get('id', 'default')
    session_attributes['reason'] = reason
    response = table.put_item(
        Item={
            'id': count_id,
            'hours': hours,
            'mood': mood,
            'reason':reason
        }
    )
    if reason ==='':
        return build_response("You feel like {} and slept {} hours, do
you want to ask for any advice to chat gpt?".format(mood, hours),
False,session_attributes)
    else:
        return build_response("You feel like {} and slept {} hours and it
was due to {}, do you want to ask for any advice to the Artificial
Intelligence?".format(mood, hours,reason), False,session_attributes)
```

9.1.10 Funció per a guardar els valors registrats diaris a la base de dades TotalSleepData:

```
def store_data(mood, hours, days):
    try:
        dynamodb = boto3.resource('dynamodb')
        tableTotal_name = 'TotalSleepData'
        tableTotal = dynamodb.Table(tableTotal_name)

        # Perform a put item operation to store the data
        responseTotal = tableTotal.put_item(
            Item={
                'mood': mood,
                'hours': hours,
                'days': days
            }
        )

        if responseTotal['ResponseMetadata']['HTTPStatusCode'] == 200:
            print("Data stored successfully.")
            return True
        else:
            print("Failed to store data:", responseTotal)
            return False
    except Exception as e:
        print("Error storing data:", e)
        return False
```

9.1.11 Funció per a extreure els valors de la base de dades TotalSleepData

```
def retrieve_data_by_mood(mood_category):
    try:
        # Perform a query operation to retrieve the data for the
specified mood category
        response = tableTotal.query(

KeyConditionExpression=boto3.dynamodb.conditions.Key('mood').eq(mood_cate
gory)
        )

        # Extract the data from the response
        if 'Items' in response and response['Items']:
            data = response['Items'][0]  # Assuming only one item is
retrieved
            return data
        else:
            print("Data not found for mood category:", mood_category)
            return None
    except Exception as e:
        print("Error retrieving data:", e)
        return None
```

9.1.12 Funció que connecta la Skill amb la intel·ligència artificial:

```
def generate_gpt_response(chat_history, new_question):
```

```
    try:
        messages = [{"role": "system", "content": "You are a helpful
assistant."}]
        messages.append({"role": "user", "content": new_question})
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=messages,
            max_tokens=100,
            n=1,
            stop=None,
            temperature=0.5
        )
        return response['choices'][0]['message']['content'].strip()
    except Exception as e:
        return f"Error generating response: {str(e)}"
```

9.1.13 Funció per defecte d'Amazon però modificant l'output:

```
def handle_fallback_intent(request, event, session_attributes):
    speech_text = "I'm sorry, I don't understand that intent."
    return build_response(speech_text, False, session_attributes)
```

9.1.14 Funció per a sumar i fer la mitjana dels valors nous, afegint al total d'hores corresponent segons l'estat anímic:

```
def postTOTALDDBB(session_attributes):

negative_adjectives=get_adjective_list_from_s3('negative_adjectives',file
_name_adjectives)

positive_adjectives=get_adjective_list_from_s3('positive_adjectives',file
_name_adjectives)

neutral_adjectives=get_adjective_list_from_s3('neutral_adjectives',file_n
ame_adjectives)
    mood = session_attributes.get('mood', 'default')
    sleep_hours = session_attributes.get('hours', 'default')
    #DDBB of total good sleep data
    goodMoodData=retrieve_data_by_mood('good')
    if goodMoodData:
        goodMoodDataHours=goodMoodData.get('hours')
        goodMoodDataDays=goodMoodData.get('days')
    else:
        goodMoodDataHours=0
        goodMoodDataDays=0
    #DDBB of total bad sleep data
    badMoodData=retrieve_data_by_mood('bad')
    if badMoodData:
        badMoodDataHours=badMoodData.get('hours')
        badMoodDataDays=badMoodData.get('days')
    else:
        badMoodDataHours=0
        badMoodDataDays=0
    #DDBB of total neutral sleep data
    neutralMoodData=retrieve_data_by_mood('neutral')
```

```
    if neutralMoodData:
        neutralMoodDataHours=neutralMoodData.get('hours')
        neutralMoodDataDays=neutralMoodData.get('days')
    else:
        neutralMoodDataHours=0
        neutralMoodDataDays=0

    #compare what mood type is today's mood
    if mood in positive_adjectives:
        goodMoodDataHours=goodMoodDataHours+sleep_hours
        goodMoodDataDays=goodMoodDataDays+1
        store_data('good',goodMoodDataHours,goodMoodDataDays)
    elif mood in negative_adjectives:
        badMoodDataHours=badMoodDataHours+sleep_hours
        badMoodDataDays=badMoodDataDays+1
        store_data('bad',badMoodDataHours,badMoodDataDays)
    elif mood in neutral_adjectives:
        neutralMoodDataHours=neutralMoodDataHours+sleep_hours
        neutralMoodDataDays=neutralMoodDataDays+1
        store_data('neutral',neutralMoodDataHours,neutralMoodDataDays)
    else:
        return build_response("I am sorry, there was an issue while
trying to update the DDBB can you try to use another adjective?",
False,session_attributes)

    ##calcular la mitja
    goodMoodMean=round(goodMoodDataHours/goodMoodDataDays,1)
    badMoodMean=round(badMoodDataHours/badMoodDataDays,1)

    ##try to say again how u feeling
    count_id=str(badMoodDataDays + goodMoodDataDays+neutralMoodDataDays)
    session_attributes['id'] = count_id

    return True;
```

9.1.15 Funció per a obtenir un fitxer d'adjectius guardats a AWS s3:

```
def get_adjective_list_from_s3(adjectives,file_name):
    try:
        # Retrieve the JSON file from S3
        response = s3.get_object(Bucket=bucket_name, Key=file_name)

        # Parse the JSON data
        existing_data = json.loads(response['Body'].read().decode('utf-
8'))

        # Return the list of adjectives
        return existing_data.get(adjectives, [])
    except s3.exceptions.NoSuchKey:
        # If the file doesn't exist, return an empty list
        return []
```

9.1.16 Funció per a actualitzar la llista d'adjectius amb un valor nou:

```
def add_adjective_to_list(key, word,file_name):
    #Retrieve the JSON file from S3
```

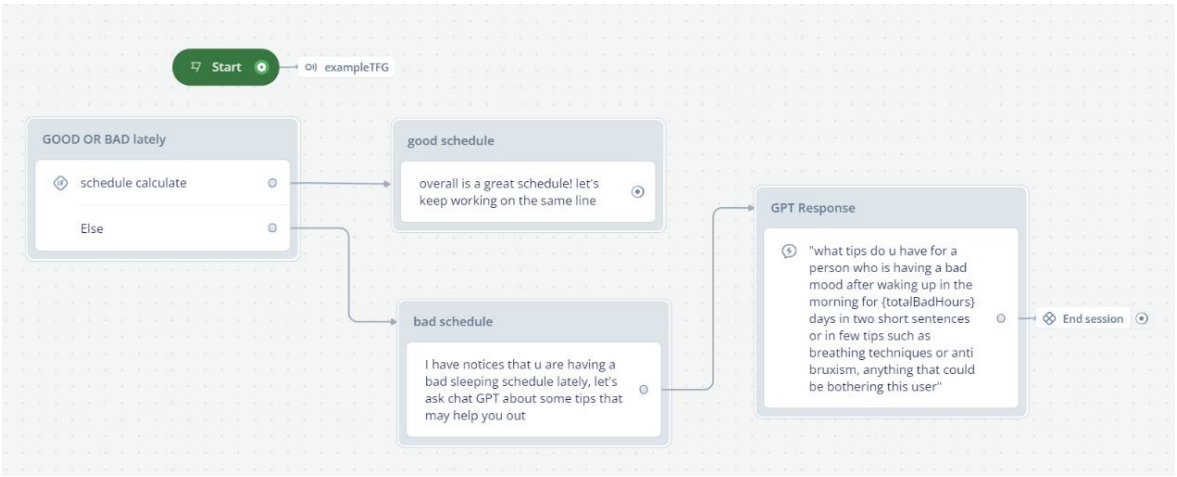
```
    response = s3.get_object(Bucket=bucket_name, Key=file_name)
    # Parse the JSON data
    adjectives = json.loads(response['Body'].read().decode('utf-8'))
    # Add the new word to the list if it doesn't exist
    if word not in adjectives[key]:
        adjectives[key].append(word)

        # Convert the updated data to JSON string
        updated_json_data = json.dumps(adjectives)

        # Upload the updated JSON file to S3, overwriting the existing
one
        s3.put_object(Bucket=bucket_name, Key=file_name,
Body=updated_json_data)
```

9.1.17 Funció que construeix la resposta que dona Alexa a l'usuari:

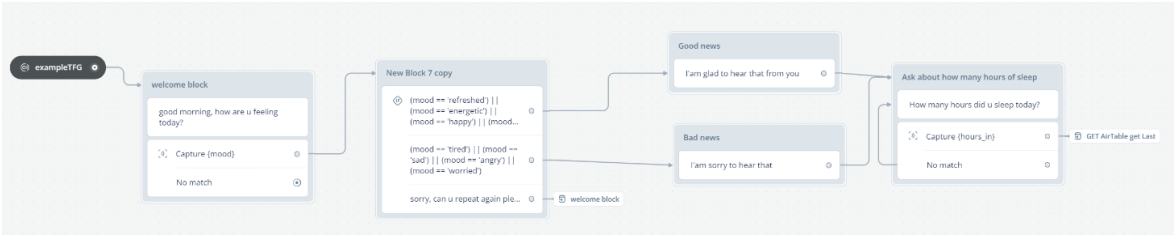
```
def build_response(output, should_end_session,session_attributes):
    return {
        'version': '1.0',
        'sessionAttributes':session_attributes,
        'response': {
            'outputSpeech': {
                'type': 'PlainText',
                'text': output
            },
            'shouldEndSession': should_end_session
        }
    }
```

9.2 Voiceflow

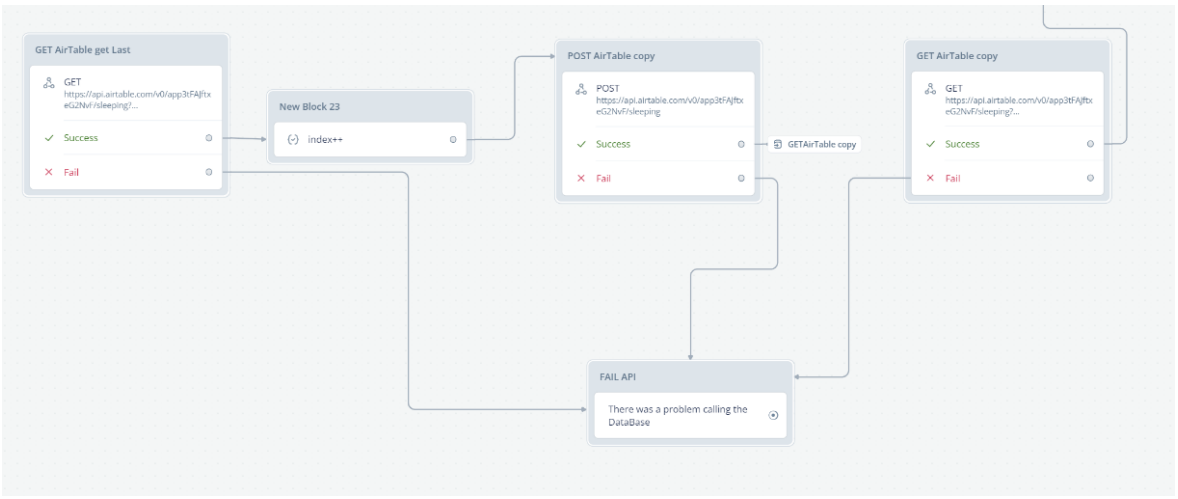
9.2.1 Principi i final del projecte:



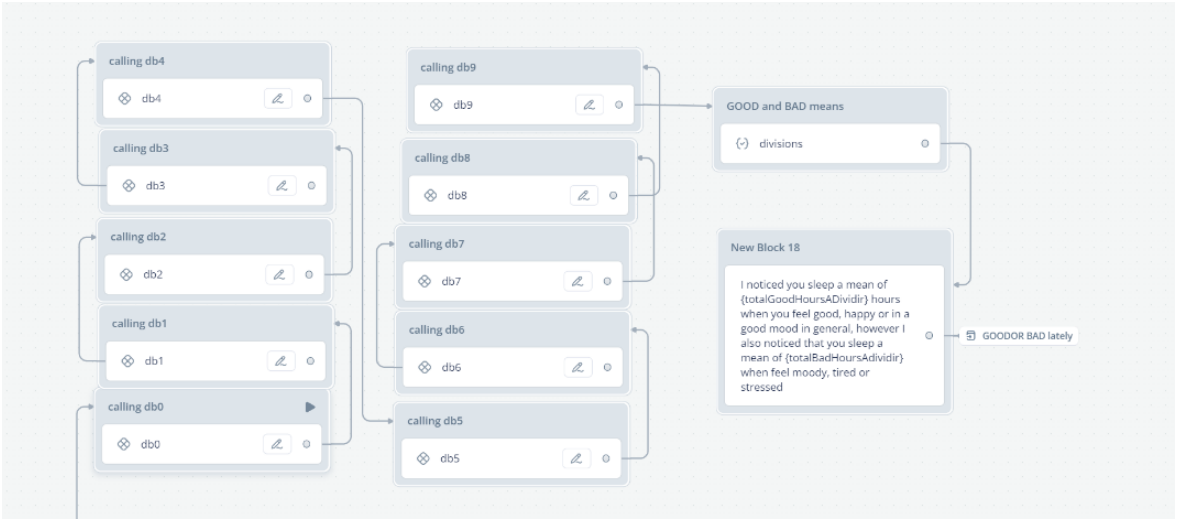
9.2.2 Intent exampleTFG part 1:



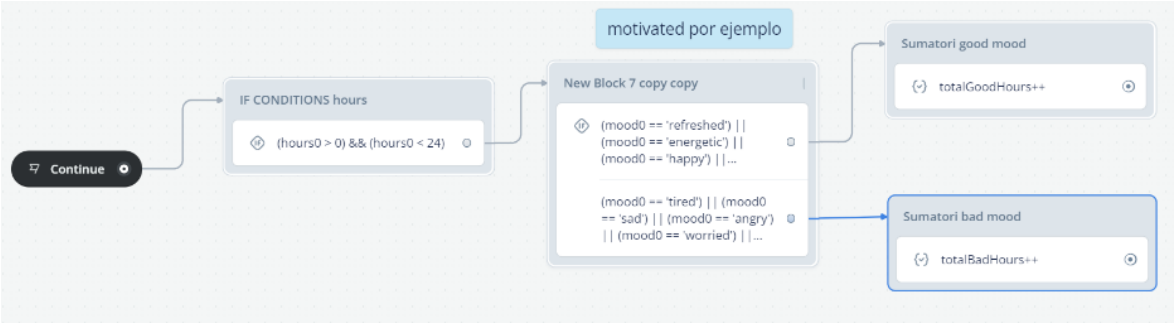
9.2.3 Intent exampleTFG part 2:



9.2.4 Intent exampleTFG part 3:



9.2.5 Component db0 al db9:


motivated por ejemplo
Continue
IF CONDITIONS hours
(hours0 > 0) && (hours0 < 24)
New Block 7 copy copy
(mood0 == 'refreshed') || (mood0 == 'energetic') || (mood0 == 'happy') || ...
(mood0 == 'tired') || (mood0 == 'sad') || (mood0 == 'angry') || (mood0 == 'worried') || ...
Sumatori good mood
totalGoodHours++
Sumatori bad mood
totalBadHours++