



# POLITÉCNICA

TRABAJO FIN DE GRADO EN INGENIERÍA DEL SOFTWARE

## Utilizando los asistentes virtuales de voz para promover la autonomía y el envejecimiento activo y saludable de las personas

Autor:

**Guillermo Díaz Fentanes**

Tutores:

**Raúl Lara Cabrera**

**Álvaro Sánchez García**

**Julio 2020, Madrid**

## Índice

Dedicatoria.....	4
Agradecimientos .....	5
Abstract.....	6
Lista de tablas .....	7
Lista de figuras .....	7
1. Introducción e información general .....	9
1.1 Propósito.....	9
1.2 Justificación .....	9
1.3 Impacto Social y Objetivos del desarrollo sostenible.....	10
1.4 Ámbito .....	14
1.5 Estado de la cuestión .....	15
1.6 Metodología del proyecto.....	15
2. Glosario.....	18
3. Asistentes Virtuales .....	20
3.1 Comparativa.....	21
3.2 Alexa Skills.....	27
3.2.1 Introducción a las Skills de Alexa .....	27
3.2.2 Eligiendo el Modelo .....	28
3.2.3 Amazon Developer Console .....	31
3.2.4 Modificación del Código .....	32
3.2.5 Eligiendo el BackEnd .....	36
3.2.6 SDK Alexa.....	38
3.2.7 Testeando la Skill .....	39
3.2.8 Publicando la Skill.....	40
3.2.9 Proceso general .....	42
3.3 Google Action.....	43
3.3.1 Introducción a las Google Actions .....	43
3.3.2 ¿Qué tipos de modelo hay y cual debe elegir? .....	44
3.3.3 Google Actions Console .....	44
3.3.4 Modificación del Código .....	51
3.3.5 Publicando la Acción.....	52
4. Demostrador.....	53
Funcionalidad 1: .....	55

Funcionalidad 2: .....	55
Funcionalidad 3: .....	56
Funcionalidad 4: .....	56
Funcionalidad 5: .....	56
Funcionalidad 6: .....	57
Diagrama de estados del demostrador .....	58
Implementación de la funcionalidad 1: .....	59
Implementación de la funcionalidad 2: .....	63
Implementación de la funcionalidad 3: .....	64
Implementación de la funcionalidad 4: .....	66
Implementación de la funcionalidad 5: .....	71
Implementación de la funcionalidad 6: .....	74
Conclusión .....	79
Bibliografía.....	80

# Dedicatoria

Dedico este trabajo de forma especial a mi hermana, por ser la persona que más quiero y la más importante, a mis padres por darme todo el apoyo que pueden y ser los pilares fundamentales de mi vida, sin los cuales nada de esto habría sido posible, habiéndome ayudado y confiado en mí en todo momento.

También quiero agradecer a mis abuelos todo el amor y cariño que me han dado y que tanto me ha valido, además de servirme como motivación para llevar a cabo un proyecto con esta temática.

# Agradecimientos

Me gustaría agradecer a mis tutores, en primer lugar, a Álvaro por formarme y enseñarme tantas cosas en el tiempo que estuve realizando las prácticas con él, además de haberme dado infinitas ideas, consejos y pautas para elaborar el proyecto. Gracias a su profesionalidad y responsabilidad he podido realizar con éxito este trabajo. En segundo lugar, quiero agradecer a Raúl, por haber enfocado y dirigido el proyecto de forma excepcional, eligiéndome como alumno al que tutorizar en el trabajo de fin de grado, además de haber sido mi profesor en una de mis asignaturas preferidas de la carrera.

Quiero agradecer también a TECSOS, lugar donde he realizado las prácticas y donde este proyecto tuvo sus inicios. Gracias a todo el equipo por enseñarme valiosas lecciones y conocimientos, jamás lo olvidaré y siempre valoraré lo que han hecho por mí, todas las personas que forman ese equipo son excepcionales y me han tratado de forma impecable, por lo que se merecen mis más sinceros agradecimientos.

# Abstract

El proyecto pretende aportar soluciones a las personas mayores o en situación de dependencia para promover su autonomía y el envejecimiento activo y saludable mediante el uso de los asistentes virtuales. En el proyecto se realizará una comparación y estudio sobre las plataformas y asistentes más adecuados para dicho colectivo, incluyendo una guía de desarrollo de acciones o skills para estos, finalizando con un demostrador que consistirá en una ayuda para la toma de medicamentos, capaz de recordar y ofrecer información e instrucciones para mejorar la adherencia médica.

The project aims to provide solutions to older or dependent people to promote their autonomy and active and healthy aging by using virtual assistants. The project will carry out a comparison and study on the most suitable platforms and assistants for said group, including a guide for developing actions or skills for them, ending with a demonstrator consisting of an aid for taking medications, capable of remembering and offering information and instructions to improve medical adherence.

## Lista de tablas

Tabla 1 Porcentaje de adherencia en enfermedades crónicas .....	12
Tabla 4 Número de Acciones y Skills en US 2017 .....	22
Tabla 5 Ejemplos de compatibilidad de dispositivos con Google y Alexa .....	23
Tabla 2 Comparación de respuestas de Asistentes de voz.....	24
Tabla 3 Comparación de categorías de respuestas .....	25
Tabla 6 Interfaz Domótica Google Home .....	26
Tabla 7 Disminución de costes en pacientes diabéticos .....	54
Tabla 9 Diagrama de estados Skill .....	58

## Lista de figuras

Figura 1 Salud y Bienestar.....	11
Figura 2 Industria, Innovación e Infraestructura .....	13
Figura 3 Reducción de las desigualdades .....	14
Figura 4 Burndown Proyecto.....	16
Figura 5 Diagrama general del proceso de los Asistentes .....	20
Figura 6 Icono de Nestle para la cocina .....	27
Figura 7 Icono de Renfe Ave.....	27
Figura 8 Icono de Donantes .....	27
Figura 9 Indicación de lugar de consola de desarrollo .....	31
Figura 10 Alexa Skills Kit Developer Console .....	32
Figura 11 Archivos básicos de una Skill .....	32
Figura 12 Indicación de lugar del código de una Skill .....	33
Figura 13 Texto base de Skill de curiosidades en español .....	33
Figura 14 Handlers de una Skill .....	34
Figura 15 Parte can_handle de un Handler .....	35
Figura 16 Parte handle de un Handler .....	35
Figura 17 Save y Deploy del código .....	36
Figura 18 Imagen de inexistencia de código hosteado por Amazon .....	36
Figura 19 Endpoint de conexión.....	38
Figura 20 Parte de Development de Alexa Developer Console .....	40
Figura 21 Consola de pruebas .....	40
Figura 22 Diagrama del proceso general de ciclo de vida de una Skill.....	42
Figura 23 Icono de Symptoma.....	43
Figura 24 Icono de Parclick.....	43
Figura 25 Icono de Recetas de cocina .....	43
Figura 26 Link de clientes de Google.....	44
Figura 27 Link de desarrolladores Google .....	45

Figura 28	Indicación de posición de Actions Console .....	45
Figura 29	Menú principal de la Actions Console .....	46
Figura 30	Creación de un nuevo proyecto Google .....	46
Figura 31	Opciones y detalles de acción Google .....	47
Figura 32	Barra de navegación de los detalles de Google Action .....	48
Figura 33	Parte de Development de una Google Action .....	48
Figura 34	Parte de Testing de una Google Action.....	49
Figura 35	Parte de Deploy de una Google Action.....	49
Figura 36	Parte de Analytics de una Google Action .....	50
Figura 37	Dialogflow de Google Actions.....	51
Figura 38	Fórmula Adherencia médica .....	56
Figura 39	Activación de Permisos de notificaciones.....	59
Figura 40	Agregación y actualización de las dependencias .....	60
Figura 41	Aviso de permitir los avisos en el dispositivo móvil .....	60
Figura 42	Esquema general de un recordatorio .....	61
Figura 43	TryCatch para la detección de errores.....	62
Figura 44	Detalle context-text .....	63
Figura 45	Referencias Break .....	64
Figura 46	Bases de datos AWS .....	66
Figura 47	Creación de Tabla .....	67
Figura 48	Características de BBDD .....	67
Figura 49	Políticas DynamoDB.....	68
Figura 50	Declaración de la tabla de DynamoDB .....	68
Figura 51	Bloque de actualización de datos .....	69
Figura 52	Atributo de adherencia añadido .....	72
Figura 53	Constantes actualizadas.....	73
Figura 54	Texto de solicitud de adherencia.....	73
Figura 55	Tipo de comunicación .....	74
Figura 56	Métricas de la alarma .....	75
Figura 57	Tipos de Métricas de Base de Datos .....	75
Figura 58	Métricas de operaciones de tabla .....	76
Figura 59	Configuración Alarma.....	77
Figura 60	SNS de correo electrónico.....	78
Figura 61	Correo electrónico de alarma .....	78



# **1. Introducción e información general**

## **1.1 Propósito**

Este proyecto se inicia con el propósito de aportar nuevas soluciones basadas en la tecnología de asistentes virtuales para las personas mayores o en situación de dependencia que permitan promover su autonomía y el envejecimiento activo y saludable. Se explicará y demostrará la utilidad general de los asistentes de voz más comercializados actualmente para ayudar a los colectivos mencionados.

El proyecto incluye una guía individual para cada uno de ellos en la que se detalla a su vez el proceso desde la creación a la publicación de Acciones o Skills, concluyendo en una comparativa donde se mostrarán las diferencias, ventajas y desventajas. Finalmente, también se ha desarrollado un demostrador donde se materializan todos los conocimientos detallados con anterioridad para hacer un proyecto funcional final.

## **1.2 Justificación**

La tecnología los últimos años ha avanzado drásticamente en todas sus modalidades y los asistentes de voz no son una excepción. Para las personas jóvenes o en contacto continuo con la tecnología supone en la mayoría de los casos un mínimo esfuerzo adaptarse a los nuevos cambios o actualizaciones novedosas, resultando en un progreso incremental y cómodo hacia el futuro [1]. Por el contrario, las personas que no tienen apenas relación con las nuevas tecnologías sufren una lenta regresión al alejarse cada vez más de los nuevos cambios y actualizaciones que se realizan

constantemente, terminando en muchos casos por generar desafección hacia todo lo que esté relacionado con el ámbito tecnológico debido a este fenómeno que comúnmente se conoce como brecha digital.

En este proyecto se pretende terminar con esta barrera en el ámbito de los asistentes virtuales dado que todas las ventajas que ofrecen pueden claramente ser utilizadas por cualquier colectivo, realizando guías y explicaciones para hacer más sencillo el acercamiento con esta clase de contenidos, además de realizar un demostrador que ayudará a mejorar la calidad de la toma de medicamentos gracias a los asistentes mediante las utilidades que nos proporcionan como avisos o recordatorios/alarmas de voz con los que la adherencia médica puede verse mejorada. El demostrador realizado se ha desarrollado únicamente en el asistente de Amazon Alexa puesto que este tiene más importancia en el marco de Cruz Roja, lugar donde se realizan mis prácticas, además de ofrecer funcionalidades muy relacionadas con el objetivo del demostrador. A pesar de esta limitación a la hora de realizar el demostrador se han estudiado completamente ambos asistentes respecto a su proceso general de creación de Skills y Acciones.

### **1.3 Impacto Social y Objetivos del desarrollo sostenible**

Los ODS son necesarios para conseguir un futuro sostenible para todos, una vez aprobada la agenda sostenible en 2015 Fueron comunicados de forma oficial por la Organización de las Naciones Unidas (ONU) con el propósito de que pudieran iniciarse acciones para su cumplimiento desde el año siguiente [2]. Se interrelacionan entre sí e incorporan los desafíos globales a los que nos enfrentamos día a día, como por ejemplo los relacionados con la salud. La fecha determinada para su cumplimiento fue fijada en el año 2030. En este proyecto se tendrán varios de estos

objetivos en cuenta, tratando de contribuir a la consecución de algunos de los mencionados a continuación.

Al estar centrado en la ayuda y la mejora de calidad de vida de las personas, este proyecto recoge un impacto social significativamente alto, importante y positivo. Supone un aumento de facilidades en el día a día en los en los perfiles de usuarios que potencialmente más lo pueden necesitar, además de mejorar específicamente la adherencia médica y la salud en general.



*Figura 1 Salud y Bienestar*

Respecto a los ODS este trabajo guarda gran relación con muchos de ellos. El primero y más importante es el principio de **Salud y Bienestar**, este proyecto está directamente relacionado con la salud, ayudando en gran parte a que las personas tengan una vida mejor, más sana y con un proceso de envejecimiento más saludable. Este trabajo ayudará a cumplir sus metas de este objetivo, las cuales incluyen reducir la tasa de mortalidad, poner fin a muertes evitables o lograr una mayor cobertura sanitaria.

Gracias a una mayor adherencia médica se consigue una reducción de muertes debido a la mejora en la responsabilidad de toma de medicamento. En la Tabla 1 se muestran los porcentajes de adherencia en enfermedades crónicas, observándose que existe una significativa y posible mejoría. Aumentando esta adherencia se consiguen, como se explica posteriormente, numerosas ventajas respecto al tratamiento del paciente como pueden ser la mejoría de la enfermedad o el

aceleramiento de la cura, además de reducir las posibilidades de muerte o agravamiento de los síntomas.

*Tabla 1 Porcentaje de adherencia en enfermedades crónicas*

<b>Diabetes tipo II</b>	<b>65-85 % (antidiabéticos orales)</b> <b>60-80 % (insulina)</b>
<b>Epilepsia</b>	<b>50-70 %</b>
<b>Hipertensión arterial</b>	<b>17-60 %</b>
<b>Dislipemias</b>	<b>45-62 %</b>
<b>SIDA</b>	<b>50-80 %</b>
<b>Osteoporosis</b>	<b>45-70 %</b>

Está demostrado que aumentando la adherencia médica se incrementa la calidad de vida del paciente como bien se indica en el estudio del CISC [3], esto viene dado por la subida de la efectividad del tratamiento obtenida por seguir las instrucciones correctamente, por lo que este proyecto indirectamente estará fortaleciendo la relación que tienen las personas con la medicina en general. Por último, también hace hincapié en poner fin a epidemias y enfermedades fácilmente transmisibles, las cuales la mayoría de ellas pueden ser en menor probabilidad contraídas gracias al conocimiento, cosa que con los asistentes virtuales es muy sencillo y práctico. Un perfecto ejemplo es la Skill creada por Cruz Roja con respecto al covid-19 (en la que tuve el placer de contribuir cuando estaba en mi periodo de prácticas), la cual ofrece información sobre la enfermedad ayudando a prevenir el contagio mediante consejos, además de mucha otras cosas, brindado así una gran ayuda en pleno periodo de epidemia en el cual había una preocupante falta de información sobre el tema, dados por ejemplo los problemas de colapso de los teléfonos, que es un contratiempo que con los asistentes virtuales no se puede dar. Hay un gran número de personas

que no tienen un conocimiento detallado sobre las preocupaciones en materia de salud que se describen en los objetivos de desarrollo sostenible que son de vital importancia y que con esta nueva tecnología pueden estar más a su alcance, con tan solo preguntarlo o al recibir una notificación de información automática. El demostrador pretende incidir a su vez en materia de concienciación, ya que la información es un factor clave a la hora de mejorar la adherencia médica, por lo que este objetivo de poner fin a epidemias también guarda relación con el proyecto puesto que al informar al paciente además de mejorar la toma de medicamentos también ayuda a poner fin a enfermedades de este carácter epidémico.



*Figura 2 Industria, Innovación e Infraestructura*

Respecto al objetivo de desarrollo sostenible de **Industria, innovación e infraestructuras** se guarda también relación, ya que en su objetivo de aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales de todos los países, en particular los países en desarrollo, entre otras cosas fomentando la innovación. Con este proyecto se pretende incidir en la capacidad de innovación de una organización social como Cruz Roja proveyendo de herramientas útiles que puedan ser industrializadas y distribuidas entre la población. La investigación también avanza en este ámbito de los asistentes gracias al proyecto puesto que sirve como base para siguientes proyectos o investigaciones relacionadas con la misma temática.



*Figura 3 Reducción de las desigualdades*

Un objetivo que también está relacionado con el proyecto es el de **Reducción de las desigualdades**, de acuerdo con sus objetivos tenemos, entre otros, promover y potenciar la inclusión social independientemente del sexo, edad, discapacidad, raza, origen, etnia o situación económica. En este proyecto se persigue solventarlo respecto a las personas mayores o con discapacidad, gracias al demostrador, la brecha digital existente [1] se verá reducida y se potenciará así la inclusión social para los colectivos mencionados puesto que estarán más en contacto con la tecnología gracias a los asistentes virtuales y al demostrador realizado.

#### **1.4 Ámbito**

Este proyecto se centra en el estudio de los asistentes virtuales y sus ventajas para la ayuda de las personas mayores o en situación de dependencia con el objetivo de promover su autonomía y el envejecimiento activo y saludable, concretamente se especializa en la parte de adherencia médica. La calidad de toma de medicamentos afecta directamente al paciente y a su capacidad de cura y recuperación [4], y estos han sido incluidos entre los objetivos principales que persigue este proyecto. Con la tecnología de los asistentes se consigue lo mencionado anteriormente mediante recordatorios o información para el paciente e informes o avisos para el responsable de cuidado.

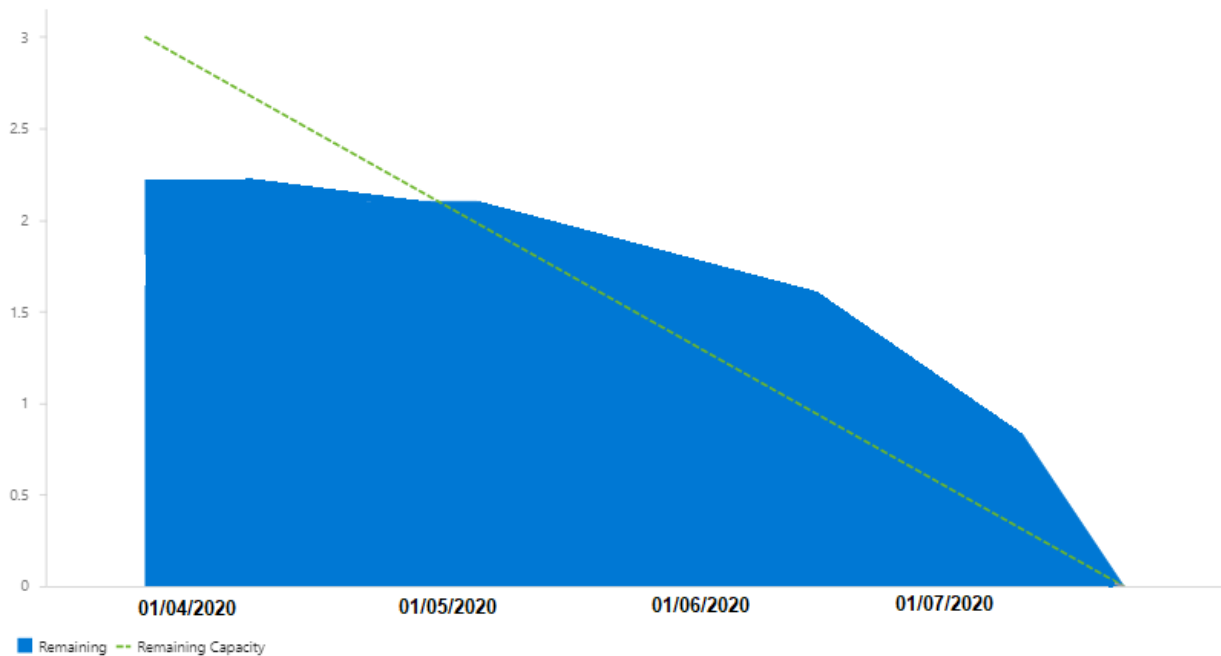
## **1.5 Estado de la cuestión**

La aplicación de los asistentes virtuales está todavía desarrollándose y mejorando puesto que es un tema muy novedoso, apenas hay proyectos terminados completamente que trabajen la temática descrita en este proyecto, aunque es cierto que ya hay varias ramas de la salud en las que se está empezando a desarrollar. Iniciativas como Acho o Skills relacionadas con la salud han dado el primer paso. Acho es un asistente que puede utilizarse sin conexión a la red creado por la universidad de Extremadura [5], además de tener diversas funciones para, entre otras cosas, recordar la toma de medicamentos, no se planea poner a la venta aún puesto que está en fase de desarrollo. También existen Skills que se encargan de ayudar en lo que a salud se refiere [6], como por ejemplo la mencionada anteriormente Skill de Cruz Roja, que puede realizar un gran número de acciones relacionadas con la gestión de la entidad, salud e información o la de SocialDiabetes que ofrece la posibilidad de monitorizar los datos útiles para las personas que tienen diabetes, como por ejemplo los niveles de glucosa.

## **1.6 Metodología del proyecto**

En este trabajo se sigue una metodología ágil para realizar todas las tareas tanto de investigación como de desarrollo de una forma ordenada, continua y abierta al cambio ya que ofrece muchas ventajas a la hora de abordar proyectos como este. Está comprobado que estas metodologías mejoran la motivación del cliente y la implicación del equipo de desarrollo, características que favorecen mucho el producto final [7]. Se tiene también un ciclo de vida incremental y evolutivo validándose al final de cada ciclo. Dentro del espectro de metodologías ágiles he elegido Scrum para abordar el proyecto. Esta decisión ha sido tomada dadas las grandes

bases de Flexibilidad, Adaptabilidad y Productividad que tiene Scrum, ofreciendo el valor más alto de negocio en el menor tiempo posible, utilizando sus técnicas como selección de roles, estimación de puntos y creación de perfiles y equipos multidisciplinarios.



*Figura 4 Burndown Proyecto*

Se ha dividido todo el trabajo en distintos sprints, que se componían de historias de usuario para separar las partes distintas que debían realizarse. Los primeros sprints consisten en la investigación de los asistentes, de sus tipos, posibilidades y características. En los siguientes sprints se comienza la comparativa de Alexa y Google Assistant, además del punto de explicación de asistentes de voz. Después se procedió a los sprints de las dos guías de creación y características de los asistentes, quedando los últimos para el desarrollo del demostrador.

Como puede apreciarse en la Figura 4, la cual muestra el trabajo restante por realizar en el tiempo, la carga al principio no se ha abordado completamente, pero en los siguientes sprints se ha ido paliando y completándose por completo en la parte final del proyecto. Esto se debe a que



inicialmente el trabajo fue difícil de orientar, dadas sus grandes y amplias posibilidades, lo que resultó en una mala estimación de las historias de usuario, ya que tomaron más tiempo del necesario, no siendo esto gran problema finalmente.

Dada la naturaleza de las metodologías ágiles, todo cambio que abordaba el proyecto era bienvenido, ya fuera sobre la teoría o sobre el desarrollo, concluyendo que fue la decisión acertada elegir esta metodología para trabajar. Al existir un punto de partida tan vasto es normal que hubiera variaciones o alteraciones en el proyecto, sobretodo para saber de que forma enfocarlo mejor en sus etapas más tempranas.

## 2. Glosario

**Hostear:** Dar alojamiento web de cualquier manera, ya sea en la nube o servidores dedicados

**AWS:** Amazon Web Services, son todos los servicios en la nube que proporciona Amazon, computación ejecución de código, guardado de datos y muchos más.

**Lambda:** Un servicio de Amazon que ejecuta código en respuesta a eventos y administra automáticamente los recursos informáticos subyacentes sin necesidad de que tengas que tener un servidor.

**Intent (Alexa):** Acción que se quiere realizar. Una Skill tiene varios intents, por ejemplo, en una de comida, un Intent puede ser hacer un pedido, otro cancelar el pedido y otro distinto valorar el servicio, todo dentro de una misma Skill.

**Invocation name (Alexa):** Palabras que hay que decir para activar la Skill

**Sample Utterance:** frases o palabra clave para invocar el nombre del intent

**EndPoint:** Link de conexión proporcionado por el host para conectarlo a la Skill

**Asistente de Google:** asistente virtual de Google para permitir conversaciones entre los usuarios y Google, dando respuesta a alguna necesidad.

**Dialogflow:** herramienta de Google para desarrollar experiencias conversacionales que puedan usarse con el Asistente de Google. Utiliza machine learning para analizar el input del usuario e identificar lo que quiere hacer para poder responder de la manera más correcta.

**Agente:** proyecto de DialogFlow para administrar ciertas experiencias en uno o múltiples idiomas.

**Intent (Google):** se trata del objetivo que tiene el usuario cuando habla con el Asistente de Google. Los agentes de Dialogflow se crean con dos intents por defecto: el de bienvenida y el de fallback, que se ejecuta cuando no se identifica qué intent se debe utilizar. Es recomendable no borrarlos, sino modificarlos para que se adapte a nuestras necesidades.

**Invocación (Google):** es cuando un usuario inicia la interacción con una acción. Puede hacerlo explícitamente, indicando su nombre, o implícitamente, indicando qué quiere hacer. Además, existen los eventos, que son una característica que permite que una acción se invoque programáticamente en vez de directamente por el usuario.

**Parámetro:** concepto que proporciona el usuario a través de su interacción con el Asistente de Google y que se utilizará para dar una respuesta correcta a la necesidad. Por ejemplo, si se quiere comprar tres cartones de leche para el viernes podríamos identificar “cartones de leche”, “tres” y “viernes” como parámetros. Estos parámetros pueden ser de un tipo definido como una entidad.

**Entidad:** se trata de representaciones de conceptos reales y que se pueden utilizar en las frases de entrenamiento para que Dialogflow identifique correctamente los parámetros. Por ejemplo, podríamos tener una entidad que represente las opciones de compra con “cartón de leche”, “batido” y “helado” como opciones. Dialogflow proporciona algunas entidades por defecto, pero también se puede añadir las nuestras.

**Fulfillment:** lógica que administra un evento para realizar la acción correspondiente. Un fulfillment puede ser una aplicación externa, un servicio, etc.

**Frases de entrenamiento:** ejemplos de posibles interacciones del usuario para llamar a un intent en concreto. Estas frases son utilizadas por un algoritmo que entrena al agente para que pueda entender lo que el usuario quiere. Es recomendable proporcionar al menos 20 frases.

### 3. Asistentes Virtuales

Los asistentes virtuales se basan en software integrado en algunos casos en un hardware capaz de facilitar la comunicación que establece el humano con la tecnología y viceversa, además de ofrecer múltiples servicios o realizar tareas. Para llevar a cabo su función siguen generalmente el proceso mostrado en la Figura 5

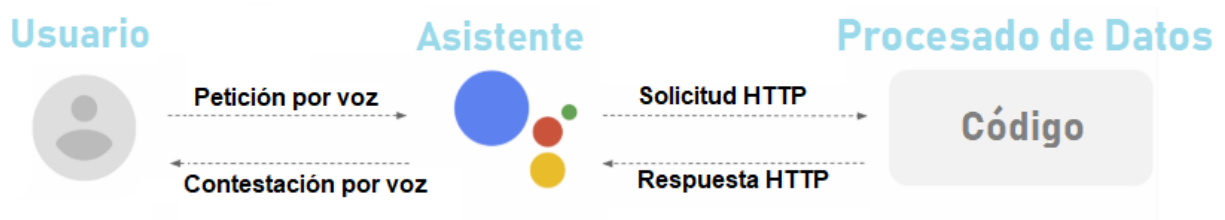


Figura 5 Diagrama general del proceso de los Asistentes

El usuario realiza cualquier tipo de petición que llega al asistente para ser transformado desde el lenguaje hablado a archivos que pueden tratarse fácilmente, de forma general se utilizan tramas Json dadas sus características de almacenamiento de datos, para posteriormente poderse enviar por el protocolo HTTP para que sea procesado en el código correspondiente. Una vez se ha ejecutado el código se genera la respuesta HTTP que llega de nuevo al asistente, donde será convertido a lenguaje hablado para que el usuario sea consciente de los cambios y respuestas ejecutados. Puede haber variaciones como por ejemplo en los casos de control de dispositivos, respuesta en forma de imágenes/vídeo o en lo que a domótica se refiere, pero el diagrama de funcionamiento general tiene casi siempre las mismas bases de funcionamiento y proceso.

Además, al ser capaces de ejecutar código, tienen la capacidad de interactuar con la mayoría de las APIS del mercado, dando cabida a cualquier posibilidad de creación y conexión, lo que supone un

gran avance porque se consigue la conexión entre distintas partes gracias a las API, pero añadiéndole la particularidad de que se puede hacer mediante voz gracias a los asistentes.

En este documento se detallan más concretamente los asistentes Alexa y Google Assistant ya que por el momento son los que cuentan con una mayor penetración en el mercado. Por el momento son los más desarrollados además de que ofrecen la posibilidad de ser programados por cualquier persona, cuentan con una amplia documentación e incitan al usuario a crear para ellos, aumentando el nivel de personalización significativamente.

## 3.1 Comparativa

Los dos asistentes antes mencionados, aunque parecidos, tienen varias diferencias que pueden ser clave a la hora de realizar una elección dependiendo de la necesidad a cubrir, como puede ser la de desarrollo o la de uso personal. Ambos avanzan y evolucionan a gran velocidad guardando cierto paralelismo en las bases principales que los determinan, pero divergen en varios puntos que se tratarán a continuación.

El asistente de Amazon tiene significativamente más funcionalidades que Google Assistant, dando más capacidad a las creaciones que se deseen hacer. Por el contrario, Google Assistant mantiene unas posibilidades más acotadas a la hora de hablar de funcionalidad, lo que se concluye finalmente en un número más bajo de Acciones respecto a el número de Skills que tiene Alexa. En la **¡Error! No se encuentra el origen de la referencia.** se muestran la cantidad en 2017 de Acciones y Skills que había en Estados Unidos, se puede observar que hay una mayoría del lado de Alexa y aunque actualmente está más igualado, sigue estando por delante de Google Asistant. Con estos datos se

concluye que el asistente de Amazon tiene ventaja en el ámbito de funcionalidades, facilitando un desarrollo que cuente con más funcionalidades y opciones que Google Assistant. Actualmente Alexa cuenta con un número aproximado de 24.000 de aplicaciones y servicios frente a tan solo 1.800 de Google Assistant.



Tabla 2 Número de Acciones y Skills en US 2017

Otro punto a tener en cuenta es la capacidad que tienen los asistentes para el control de dispositivos, ya sean de domótica o de cualquier uso en general. En este caso hay ciertas peculiaridades y diferencias en cada asistente. Cuando hablamos de número de dispositivos distintos compatibles es Alexa la que gana, ya que soporta muchos tipos y plataformas. En la **¡Error! No se encuentra el origen de la referencia.** se aprecian algunos ejemplos de compatibilidades de dispositivos. Se puede ver que en todos los ejemplos de distintos tipos es Google Assistant quien tiene casos en los que es imposible su compatibilidad, dejando a Alexa como asistente predominante para su control y concluyendo así cual es de los dos el que más alcance tiene en lo que respecta a las posibilidades de relación y conexión con otros dispositivos.

Tabla 3 Ejemplos de compatibilidad de dispositivos con Google y Alexa

<b>Lámparas e interruptores:</b>	<b>Amazon Echo</b>	<b>Google Home</b>
<a href="#">Philips Hue</a>	Sí	Sí
<a href="#">Teckin</a>	Sí	Sí
Amazon <a href="#">enchufe</a>	Sí	Sí
<a href="#">Osram</a>	Sí	No
Leviton switches	Sí	No

Los Smart Hubs son dispositivos de los fabricantes para integrar muchos dispositivos Smart Home:

<b>Smart Hubs:</b>	<b>Amazon Echo</b>	<b>Google Home</b>
SmartThings	Sí	Sí
Wink	Sí	Sí
Logitech <a href="#">Harmony</a>	Sí	Sí
Insteon	Sí	No
Scout	Sí	No

Termostatos para reducir los costos de calefacción:

<b>Termostatos:</b>	<b>Amazon Echo</b>	<b>Google Home</b>
Nest	Sí	Sí
Ecobee4	Sí	Sí
<a href="#">Honeywell Lyric</a>	Sí	No
Sensi	Sí	No
<a href="#">Netatmo</a>	Sí	No

Una gran parte de las acciones que los asistentes alcanzan a realizar es la de dar respuesta a preguntas que se les plantean, varios estudios concluyen que de entre varios asistentes es Google Assistant el que mejor calidad de respuesta ofrece [8]. La siguiente tabla muestra el porcentaje de aciertos a las preguntas planteadas para medir el nivel de inteligencia de los asistentes y así averiguar quien se desenvuelve mejor, en este caso Google Assistant está a la cabeza de todos.

Tabla 4 Comparación de respuestas de Asistentes de voz

	Answered Correctly	Understood Query
Google Assistant	87.9%	100%
Siri	74.6%	99.6%
Alexa	72.5%	99.0%
Cortana	63.4%	99.4%

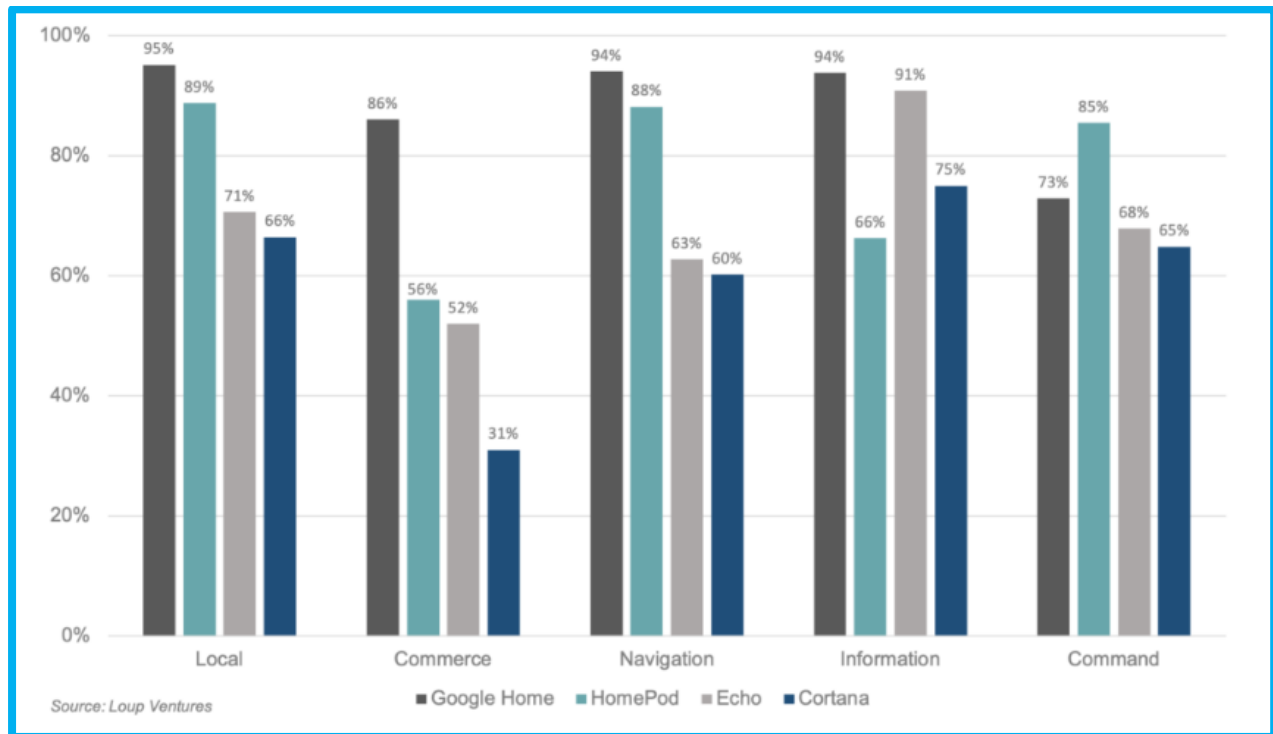
La tabla tiene dos columnas de datos que incluyen el porcentaje de acierto de las preguntas y el porcentaje de comprensión de las mismas, aunque Google Assistant puede parecer el mejor en este aspecto, al ser estos datos recabados en 2018 no se ha podido tener en cuenta la gran mejora que experimentó y experimenta ahora mismo Alexa. Como se puede apreciar, la mayoría de las preguntas son comprendidas casi al máximo, pero no son contestadas correctamente con tal grado de perfección, siendo Google Assistant el que más se intentará acercar a él. Separando en categorías las más de 800 preguntas que se realizaron en el estudio de loupventure [9] se puede comprender en que casos los asistentes son más o menos fuertes a la hora de responder.

Actualmente Alexa denota un mejor desempeño en esta categoría (casi llegando a superar a Google Assistant), ya que su desarrollo y avance ha sido mayor que ningún otro asistente de los mostrados en el estudio realizado en 2017 de la

Tabla 4. A continuación se detalla más la calidad de respuestas dadas a las preguntas, divididas en distintas secciones dependiendo de su clase.



Tabla 5 Comparación de categorías de respuestas

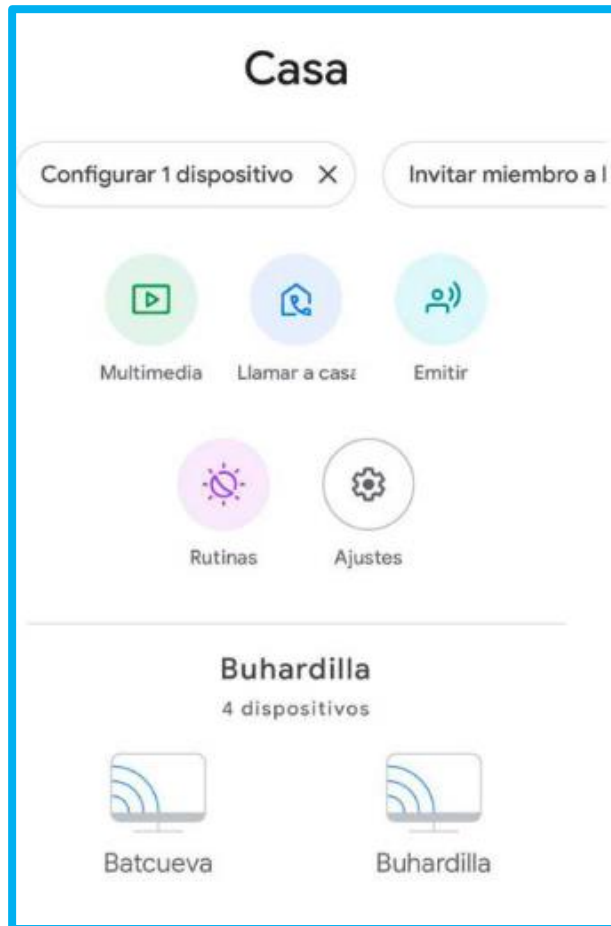


En las cinco distintas categorías de preguntas de la Tabla 5 es Google Assistant quien mantiene el mayor porcentaje de éxito de media, bajando solamente un poco los aciertos en la categoría de Comercio y quedando en una posición distinta a la primera en tan solo la categoría de Comandos, concluyendo igualmente que es el favorito en el ámbito de resolución de preguntas.

Pasando al siguiente punto de comparación, es importante destacar que no es de alta gravedad el no tener un máximo grado de compatibilidad de dispositivos, como le pasa a Google Assistant, ya que igualmente hay una gran mayoría con los que se puede conectar, pero es una diferencia que puede llegar a notarse en caso de querer resultados específicos en el ámbito de la domótica o la automatización. Aunque Google Assistant, como ya se ha expuesto, tiene menos dispositivos conectables cuenta curiosamente con una interfaz mejor adaptada al control de los mismos, sobretodo en el apartado de domótica. La distribución de apartados, botones y grupos hace que sea

mucho más sencillo añadir, gestionar y distribuir los dispositivos de una casa inteligente que Alexa, ya que esta cuenta con una interfaz menos intuitiva y más monótona.

Tabla 6 Interfaz Domótica Google Home



En el resto de aspectos los asistentes están más igualados habiendo pequeños y reñidos detalles como que Alexa cuenta con una voz más natural y mejor sonido desde sus altavoces. Todas las diferencias y características mencionadas en la comparativa ayudan a comprender por que en este proyecto, por lo tanto, se ha decantado Alexa como favorito para realizar la Skill del demostrador, dado que hay más posibilidades para crear funciones que serán necesarias en el mismo.

## 3.2 Alexa Skills

### 3.2.1 Introducción a las Skills de Alexa

Las skills de Alexa son funcionalidades personalizadas que se pueden añadir a los dispositivos **Amazon Echo** y otros dispositivos con **Alexa integrada**, teniendo como particularidad que pueden ser desarrolladas por cualquier persona. Dado que Amazon nos permite crear libremente infinidad de funcionalidades para su asistente las posibilidades que hay disponibles son incalculables, existiendo como única limitación el hecho de no poseer los conocimientos necesarios para poder llevar a cabo esta tarea. A continuación, se detalla todo lo esencial para comprender y crear una Skill de Alexa, abriendo así un amplio mundo de posibilidades para crear y desarrollar.

Algunos ejemplos útiles para empezar a comprender el tema son los siguientes



Figura 6  
Icono de Nestlé  
para la cocina

Una Skill que ayuda a cocinar gracias a su gran repertorio de recetas junto a la explicación detallada paso a paso de cada una de ellas, como el caso de Nestlé para la cocina.



Figura 7  
Icono de Renfe  
Ave

También sirven para poder obtener mucha información, sabiendo en todo momento los horarios, plazas disponibles, trayectos y el precio de los billetes del tren, cosa que se puede hacer gracias a la Skill de Renfe AVE.



Figura 8  
Icono de Donantes

Demostrando el punto de versatilidad al que se puede llegar también existe una Skill en Euskadi para donantes de sangre, informando en qué momento está disponible el autobús de donación de sangre en el municipio de Euskadi.

Respecto a la estructura básica de las Skills, se componen de dos partes principales:

- **FrontEnd** (interfaz de voz)
- **BackEnd** (lugar donde reside la lógica de la Skill)

Además, se dividen en distintos modelos de Skill predeterminados. Si se elige el idioma de Alexa como español habrá menos variedad de modelos de Skill que en la versión de habla inglesa.

Los modelos más importantes son los de Custom Skills, Smart-Home Skills, Flash-Briefing Skills, Video Skills, Music Skills y List Skills.

Principalmente para crear una Skill se deberá elegir **Dónde hostear** el BackEnd y **Qué modelo** de Skill se va a seguir, sin olvidar de elegir un buen “invocation name”, que será la palabra que se utilizará para llamar a la Skill.

### 3.2.2 Eligiendo el Modelo

Para elegir el tipo de Skill que se quiere primero se tiene que conocer exactamente de qué trata y que cubre cada modelo para seleccionar el que más interese.

Se pueden agrupar todos los modelos en dos grupos distintos. El grupo de **Pre-Built Skills**, cuyo uso está definido y adaptado y el grupo de **Custom Skills**, que pueden personalizarse libremente sin necesidad de tener un uso ya definido en las anteriores Skills. A continuación se detallarán ambos grupos con sus distintas características y posibilidades.

## Pre-Built Skills

### Smart Home Skills

Como su propio nombre sugiere, permitirá controlar dispositivos como cámaras, luces, cerraduras, televisiones, termostatos...

Se invoca y maneja dependiendo del dispositivo que se quiera controlar con frases del tipo siguiente:

- Enciende/ Apaga la luz
- Aumenta/Disminuye la temperatura
- Cambia el color de la luz
- Pon el candado en la puerta de atrás
- Muéstrame la cámara del patio

### Flash Briefing Skills

Este modelo de Skill se encarga de ofrecer un breve contenido informativo. Como creador puedes añadir contenido (incluyendo audios o imágenes) a la información que ofrecerá la Skill. Se utilizan el siguiente ejemplo de frases:

- Dime las noticias de hoy
- Ponme al día

### Video Skills

Las Video Skills permiten el contenido de imágenes o vídeos como por ejemplo shows de televisión, series o películas.

- Cambia al canal 5

- Pon el partido del Madrid

## Music Skills

Las Music Skills permiten escuchar canciones, playlists o estaciones de radio. Para este tipo de Skill, la API ofrece las palabras que el usuario puede decir para pedir y controlar todo el contenido de audio.

## List Skills

Las List Skills facilitan el uso de listas de eventos, entendiendo y reaccionando también a los cambios que les pueden ocurrir (cumplimiento, aplazamiento, modificación, edición, vencimiento...). Como creador se puede definir el EndPoint donde se recibirán los ítems de la lista y como se responderá a la lista de eventos.

Pueden ayudar, entre otras muchas cosas, a:

- Añadir un ítem a una lista
- Quitar un ítem de la lista
- Actualizar una lista

## Custom Skills

Este modelo de Skill es el que más control ofrece sobre la experiencia de usuario ya que es el más personalizable y que más libertad otorga. Todas las Skills que se salgan de los marcos predefinidos de los demás modelos pertenecen a este tipo. Permite hacer toda clase de acciones personalizables y el desarrollador puede definir casi todo. Permitirán realizar diversas acciones como, por ejemplo:

- Mirar información sobre la altura de montañas o picos
- Pedir una pizza

- Pedir un taxi
- Cualquier acción que se te ocurra

### 3.2.3 Amazon Developer Console

Una vez elegido el tipo de Skill se procederá al siguiente paso. Todo esto se realizará desde la página de **Amazon Developer Console**. En la web de Amazon Alexa se encuentra el apartado de “Desarrolladores de Skills”, que es donde está la parte de “Consola de Desarrollo”.

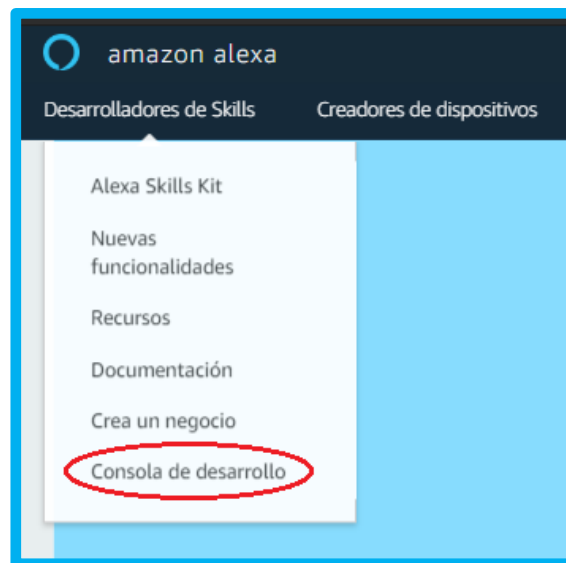


Figura 9 Indicación de lugar de consola de desarrollo

Para entrar a la consola de desarrollo se necesita previamente una cuenta de Amazon Alexa, ya que llevará a una página en la que se deberá iniciar sesión con ella antes de llegar a la consola de desarrollo.

Se podrá ver el botón para crear Skills junto a un menú debajo del mismo donde aparecerán todas las Skills que vayamos haciendo. Esta web no solo permite crear las Skills, sino también gestionarlas, analizarlas, modificarlas, publicarlas, monetizarlas y gestionar esas ganancias.

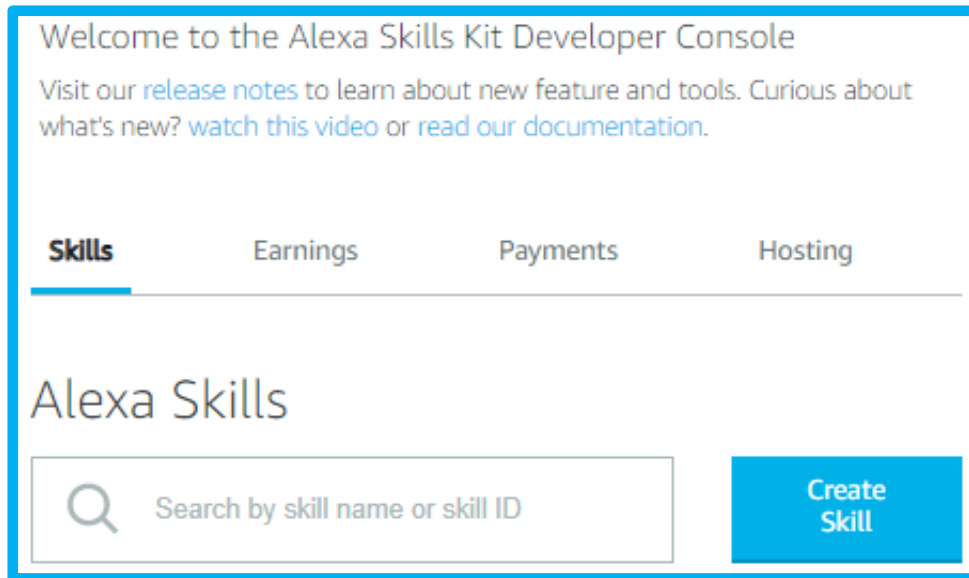


Figura 10 Alexa Skills Kit Developer Console

### 3.2.4 Modificación del Código

Una vez creada la Skill, se desplegará un menú con distintos apartados para modificarla, testearla o publicarla. Para modificar y actualizar el código de la Skill se deberá acceder al apartado de “Code” en la web de Alexa Developer Console. Aparecerá un navegador de archivos por el que se permite explorar libremente, además de poder **editar el código desde ahí**.

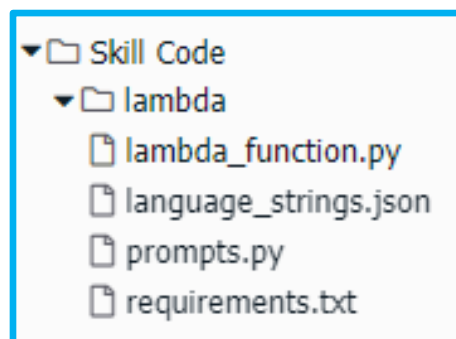


Figura 11 Archivos básicos de una Skill



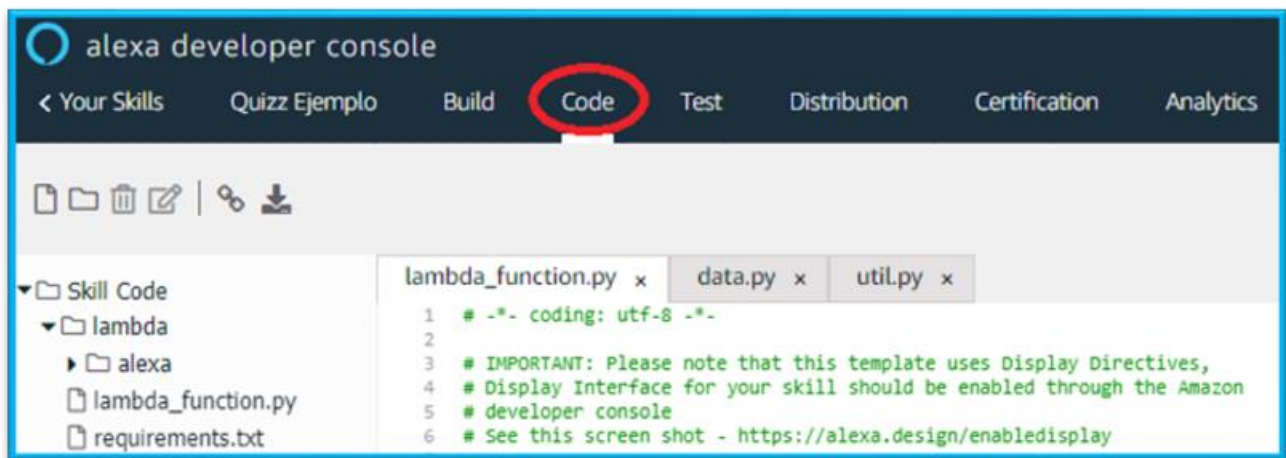


Figura 12 Indicación de lugar del código de una Skill

También se encuentra el archivo de “language\_strings” con las frases que se utilizarán en el lenguaje que se necesitan en cada momento de la Skill. En la siguiente imagen se ve el que corresponde al español.

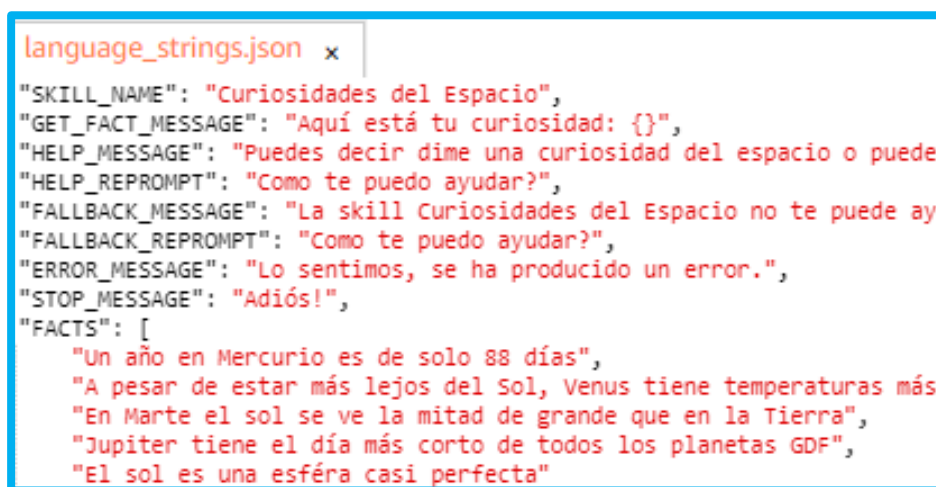
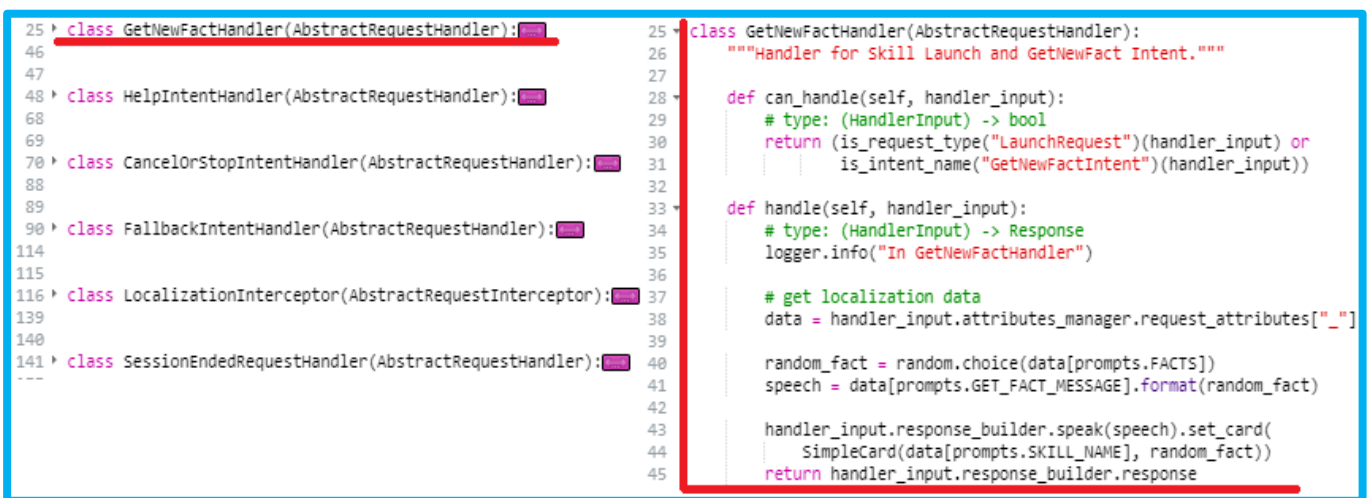


Figura 13 Texto base de Skill de curiosidades en español

En la hoja de código llamada “**lambda\_function**” se encuentran las clases de los distintos **intents**. En el ejemplo inferior (cada Skill es diferente y cambia bastante, sobre todo si son de distintos modelos), se desplegará la clase del intent **GetNewFactHandler**, la cual es la principal y da un dato aleatorio en la Skill que trata de contar curiosidades.

En la parte izquierda de Figura 14 se pueden ver las clases que representan a los intents, estando subrayada la más importante y que se analizarán en detalle. En la parte de la derecha está la clase desplegada con todas las partes que se explicarán.



```
25 ▶ class GetNewFactHandler(AbstractRequestHandler):
46
47
48 ▶ class HelpIntentHandler(AbstractRequestHandler):
68
69
70 ▶ class CancelOrStopIntentHandler(AbstractRequestHandler):
88
89
90 ▶ class FallbackIntentHandler(AbstractRequestHandler):
114
115
116 ▶ class LocalizationInterceptor(AbstractRequestInterceptor):
139
140
141 ▶ class SessionEndedRequestHandler(AbstractRequestHandler):
---
```

```
25 class GetNewFactHandler(AbstractRequestHandler):
26     """Handler for Skill Launch and GetNewFact Intent."""
27
28     def can_handle(self, handler_input):
29         # type: (HandlerInput) -> bool
30         return (is_request_type("LaunchRequest")(handler_input) or
31               is_intent_name("GetNewFactIntent")(handler_input))
32
33     def handle(self, handler_input):
34         # type: (HandlerInput) -> Response
35         logger.info("In GetNewFactHandler")
36
37         # get localization data
38         data = handler_input.attributes_manager.request_attributes["_"]
39
40         random_fact = random.choice(data[prompts.FACTS])
41         speech = data[prompts.GET_FACT_MESSAGE].format(random_fact)
42
43         handler_input.response_builder.speak(speech).set_card(
44             SimpleCard(data[prompts.SKILL_NAME], random_fact))
45         return handler_input.response_builder.response
```

Figura 14 Handlers de una Skill

Se divide en las dos partes distintas **can\_handle** y **handle**:

La parte de **can\_handle** específica se usa para chequear si el handler puede manejar la petición que llega.

```

def can_handle(self, handler_input):
    # type: (HandlerInput) -> bool
    return (is_request_type("LaunchRequest")(handler_input) or
            is_intent_name("GetNewFactIntent")(handler_input))

```

Figura 15 Parte can\_handle de un Handler

La siguiente parte de **handle** dirá como manejar el intent, se encarga de recibir el input y construir la respuesta para el usuario a partir de su request.

```

def handle(self, handler_input):
    # type: (HandlerInput) -> Response
    logger.info("In GetNewFactHandler")

    # get localization data
    1 data = handler_input.attributes_manager.request_attributes["_"]

    2 random_fact = random.choice(data[prompts.FACTS])
    speech = data[prompts.GET_FACT_MESSAGE].format(random_fact)

    handler_input.response_builder.speak(speech).set_card(
    3 |     SimpleCard(data[prompts.SKILL_NAME], random_fact))
    return handler_input.response_builder.response

```

Figura 16 Parte handle de un Handler

**En el punto 1** se cargan los datos de los cuales sacarán las dos siguientes sentencias.

**En el punto 2** primero elige de forma random una frase de FACTS (consultar en la imagen de “Language\_strings”) y luego lo junta con GET\_FACT\_MESSAGE, el cual al consultar de nuevo la imagen de “Language\_strings” se verá que contiene la frase de “Aquí está tu curiosidad”, por lo que dirá esa frase acompañado de una curiosidad elegida de forma aleatoria.

**En el punto 3** se observa el return que puede, en este caso, también ir acompañado de una “card” que es una imagen para que aparezca en la pantalla de los Echo Show.

Una vez editado o actualizado el código en caso de que se necesite, se deberá guardar y posteriormente darle a “Deploy”, para que los cambios se suban finalmente a la Skill.

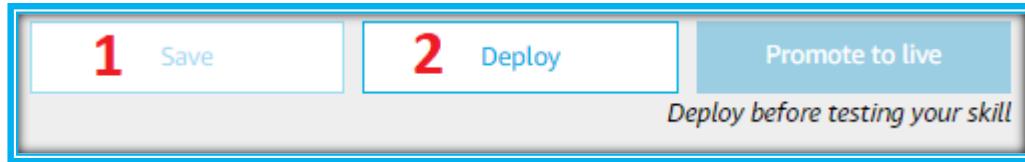


Figura 17 Save y Deploy del código

En caso de no hostear con Amazon, estará y será posible editarlo desde el sitio en el que esté hosteado, en su propio editor de código. Si se elige otro servidor distinto a Amazon, al entrar al apartado “Code” se mostrará la siguiente imagen, ya que no se está hosteando la Skill en los servidores de Amazon.



The code editor only works with an Alexa-hosted skill.  
Have you tried one?

Figura 18 Imagen de inexistencia de código hosteado por Amazon

### 3.2.5 Eligiendo el BackEnd

Una vez se haya elegido el tipo de Skill, se deberá tener claro donde residirá la lógica de la misma. Existen dos opciones, ya que se puede elegir si hostear la Skill en los **servidores de Amazon** o en **otros servidores** que se proporcionen de forma específica.

Dependiendo de las necesidades se deberá elegir una opción u otra. Lo más recomendable es utilizar los servidores de Amazon para tener todo más cómodamente unificado y ahorrarse pasos extra, pero dependerá de cada situación.

### **Amazon Web Services:**

Para utilizar este servicio que proporciona Amazon se necesitará crear una cuenta de **AWS**. Esta opción no tiene un coste asociado y ofrece una capa gratuita de servicios durante un año. Para crear esta cuenta se necesitará introducir una tarjeta de crédito, aunque no será necesario pagar.

Desde AWS se utilizará una de sus funciones llamada **Lambda**, que permitirá ejecutar código cuando sea necesario y sin necesidad de un servidor, justo lo que se necesita para una Skill. Los lenguajes son **Node.js, Java, Python, C# o Go**. Tiene las siguientes ventajas:

- **Low Cost:** Esta plataforma sigue la premisa de ofrecer servicios a bajo coste y a un amplio número de usuarios de lowcost y cuantos más usuarios la usen más barato será
- **Escalabilidad:** Se puede contratar y configurar el servicio que se quiera, escalándolo a grandes niveles por si fuera necesario
- **Pago por uso:** AWS permite pagar solamente por lo que se utilice y en el momento que se utilice, ahorrando bastante dinero. Aun así, puede resultar demasiado complicado para casos sencillos o para personas principiantes, siendo esto un inconveniente.

### **Otros servidores:**

En caso de que se tenga un proveedor propio de servidores o servicios se podrá elegir para que haga de host y ejecute la Skill. Cuando se elija esta opción habrá que realizar un paso extra. Al ser un servidor ajeno, la Skill necesitará un EndPoint para vincularse con el servidor.

En el menú de la Skill de Alexa habrá que dirigirse al apartado EndPoint e Introducir ahí el link con la API proporcionada. Dependiendo del proveedor la forma y manera de conseguirlo irá en función de sus métodos. Al ser un servidor propio, se podrá utilizar **cualquier lenguaje** de programación



Figura 19 Endpoint de conexión

Esta forma ofrece la ventaja de poder elegir el servidor que se quiera. De esta manera se podrá tener una **elección muy especializada** para cada caso y no depender así de AWS.

Como inconveniente pueden estar los costos de mantenimiento o la latencia de los servicios, ya que seguramente no tengan las mismas capacidades que AWS ofrece.

### 3.2.6 SDK Alexa

Una de las herramientas más útiles a la hora de crear Skills es el SDK que proporciona Amazon Alexa.

Significa Alexa Skills Kit y son un conjunto de herramientas y librerías que harán más fácil programar y diseñar Skills.

Está disponible para tres lenguajes de programación que son **Node.js, Java y Python**.

Contiene la documentación y guías para las siguientes características:

- Amazon Pay
- Audio Player
- Display para la pantalla del Echo Show
- Personalización de interfaces
- Mensajería
- Vídeos
- Monetización
- Lista
- Enlace de cuentas
- Recordatorios

Además, se encuentra documentación técnica sobre aspectos muy interesantes a la hora de programar Skills en sus diferentes partes, como por ejemplo el procesamiento de solicitudes, la construcción de las respuestas, la gestión de los atributos y las llamadas a la API de servicios de Alexa

### **3.2.7 Testeando la Skill**

Habiendo terminado de crear o modificar la Skill se podrá testear para ver si funciona o va avanzando según se añaden funcionalidades. Para ello hay que ir al apartado “**Test**” y cambiar la Skill a **Development**, ya que por defecto se encuentra en **Off**.

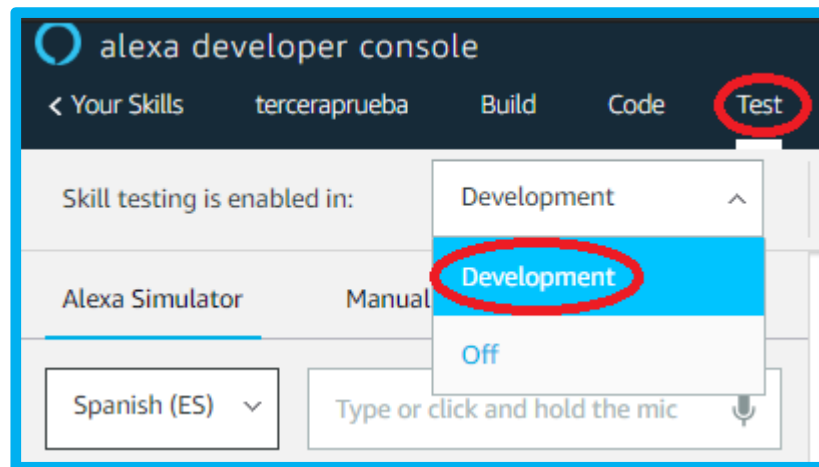


Figura 20 Parte de Development de Alexa Developer Console

Habr  un espacio en el que se puede interactuar con Alexa y probar la Skill. Se puede interactuar con ella con el micr fono y por teclado, escribiendo lo que se quiera decir. Tambi n hay una pantalla que muestra las im genes en caso de que se quiera testear la Skill con un “Medium Hub” por ejemplo.

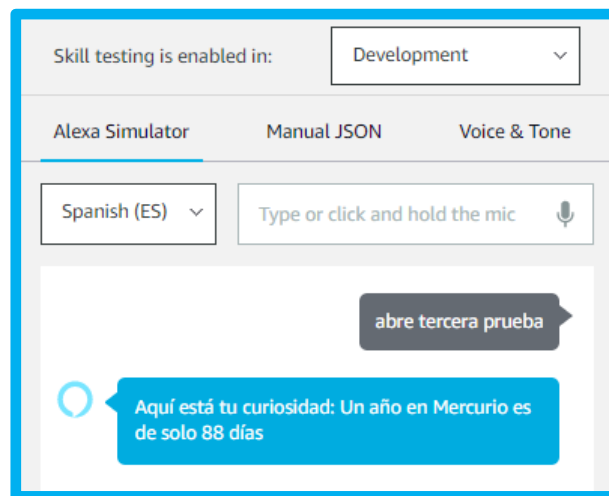


Figura 21 Consola de pruebas

### 3.2.8 Publicando la Skill

Una vez desarrollada la Skill se deber  publicar. Antes de hacer esto permanecer  en estado “**en desarrollo**”, hasta que finalmente se suban y pase a ser publicada en la tienda de Skills.



Es necesario dirigirse a la parte de Distribución y rellenar toda la información que solicitan sobre la Skill. También hay que subir dos iconos y rellenar unos criterios de Test.

Una vez hecho esto pasará al estado de “**En Revisión**”. Este proceso suele durar aproximadamente una semana. Si no es aceptada, se indicará lo que hay que corregir vía correo electrónico para que sea corregida y pase exitosamente el proceso la siguiente vez que se envíe.

### **¿Se puede utilizar privadamente la Skill creada?**

De momento **no se pueden** utilizar de forma privada tus propias Skills. Esto se debe a que desde el estado de “en desarrollo” pasa a “publicado”, lo cual implica que sale públicamente en el Skills Store de Amazon. **Hay una posibilidad** de usar la Skill que se ha creado de forma privada, aunque no está hecho para utilizarse de esta manera. Consiste en dejarla en la fase de “en desarrollo” y no publicarla, para que nunca pase a ser pública.

Al utilizar este método, solo tendrá la Skill la cuenta de desarrollador en la que está siendo desarrollada. En caso de querer que la Skill sea utilizada por más gente se deberá compartir esa misma cuenta de desarrollador en la que reside nuestra Skill. Puede que se reciban avisos de Amazon sobre por qué no se publica la Skill, ya que como se ha mencionado antes, el estado de “en desarrollo” no está dedicado para esto.

También es posible que se abra una versión beta donde tengan acceso múltiples usuarios para probar la Skill, aunque tengan distintos correos y cuentas, que es el caso que se dio cuando se estaba desarrollando la Skill del covid19 de CruzRoja.

### 3.2.9 Proceso general

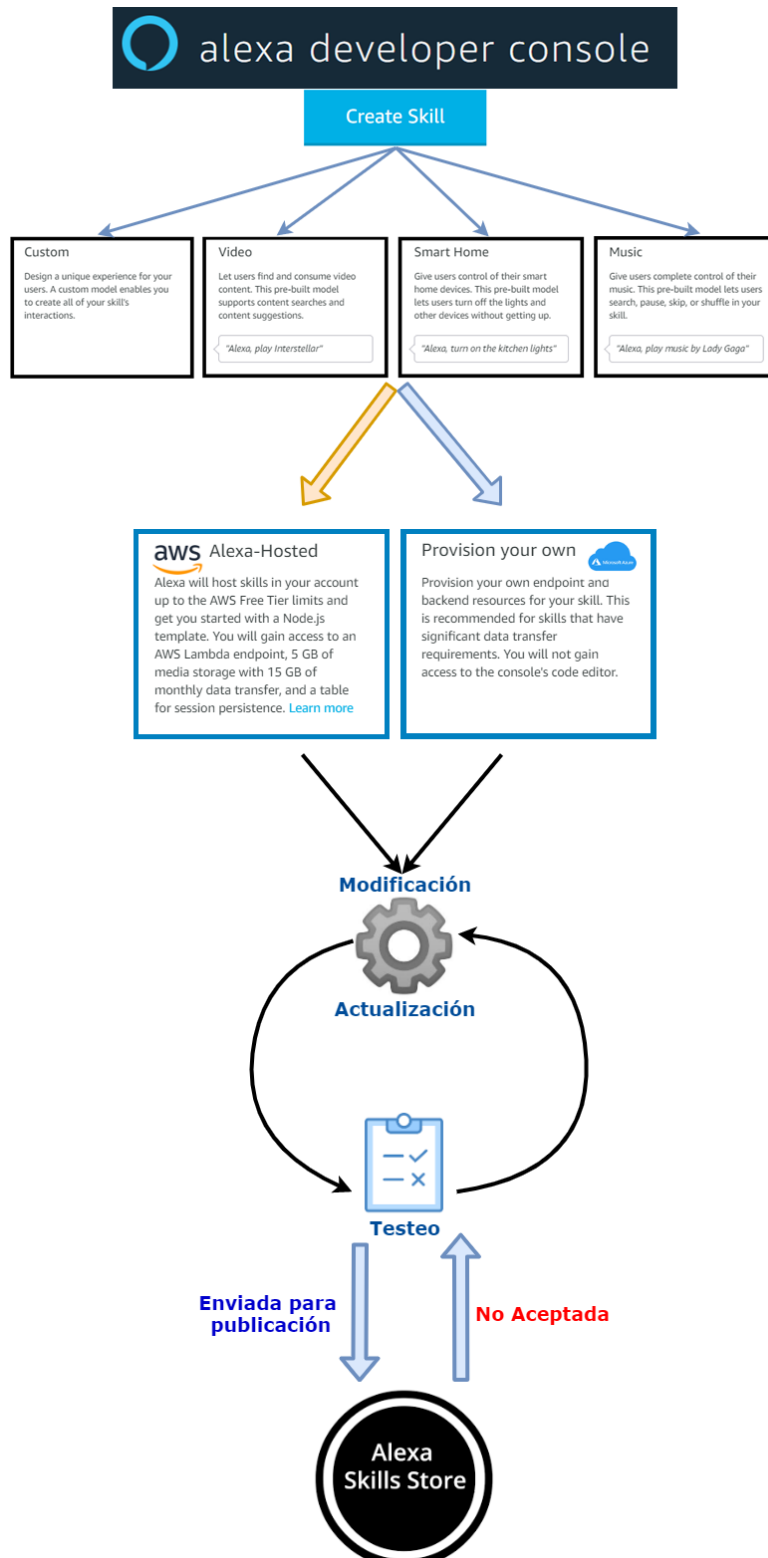


Figura 22 Diagrama del proceso general de ciclo de vida de una Skill

## 3.3 Google Action

### 3.3.1 Introducción a las Google Actions

Al igual que con Alexa, Google también cuenta con la posibilidad de crear Skills para su asistente de voz, aquí reciben el nombre de Actions. Aunque ambos cuentan con bases similares como permitir extender las funcionalidades que ofrecen todos los dispositivos que tienen integrado su asistente, hay partes en las que difieren y es interesante conocerlas. Se mantendrá un paralelismo con respecto a la parte del documento de Alexa para que sea más sencillo el seguimiento de los pasos.



Figura 23  
Icono de Symptoma

Como ejemplos se encuentran la Google Action de Symptoma, capaz de analizar los síntomas cuando se está enfermo y aunque, por supuesto, no sustituye al médico, resulta de gran ayuda en los momentos iniciales de una posible enfermedad.



Figura 24  
Icono de Parclick

También existe Parclick entre las más útiles, una Action para conductores que ayuda a encontrar aparcamiento cuando se conduce, siendo de gran utilidad y ahorrando mucho tiempo de búsqueda de sitio para aparcar.



Figura 25  
Icono de Recetas de cocina

Si lo que se quiere es algo popular y sencillo, las Actions de cocina siempre resultan muy útiles gracias a la gran variedad de recetas y posibilidades que ofrecen a la hora de querer cocinar algo.

### 3.3.2 ¿Qué tipos de modelo hay y cual debe elegir?

Los tipos de Action se dividen principalmente en 2 grupos distintos, las Conversational Actions y las Smart Home.

Las **Smart Home Action** permiten conectar los dispositivos a través de las apps de Google Home y Google Assistant para poder utilizarlos.

Las **Conversational Actions** extienden la funcionalidad del propio asistente permitiendo a desarrolladores crear experiencias personalizadas o conversaciones con el asistente.

### 3.3.3 Google Actions Console

Para crear las acciones y gestionarlas hay que utilizar **Google Actions Console**, será la herramienta principal en todo este proceso, entre otras.

Para acceder a esta herramienta es aconsejable primero visitar la propia página de **Google Assistant**.

Es importante diferenciar que hay una página de Google Assistant para clientes y otra para **desarrolladores**, siendo esta última la que más interesa.

En la Figura 26 se puede observar la página normal para cualquier **cliente** o persona que se quiera informar.

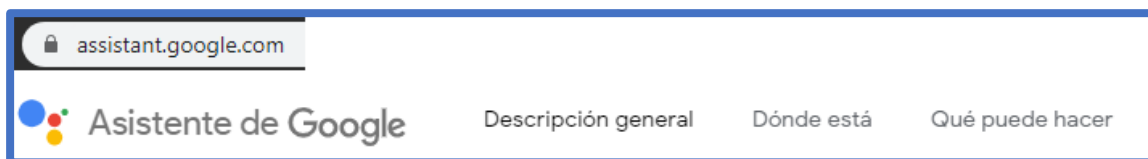


Figura 26 Link de clientes de Google

Indica una descripción general o qué cosas se pueden hacer con el asistente, así como a que tipos de dispositivos los puedes conectar

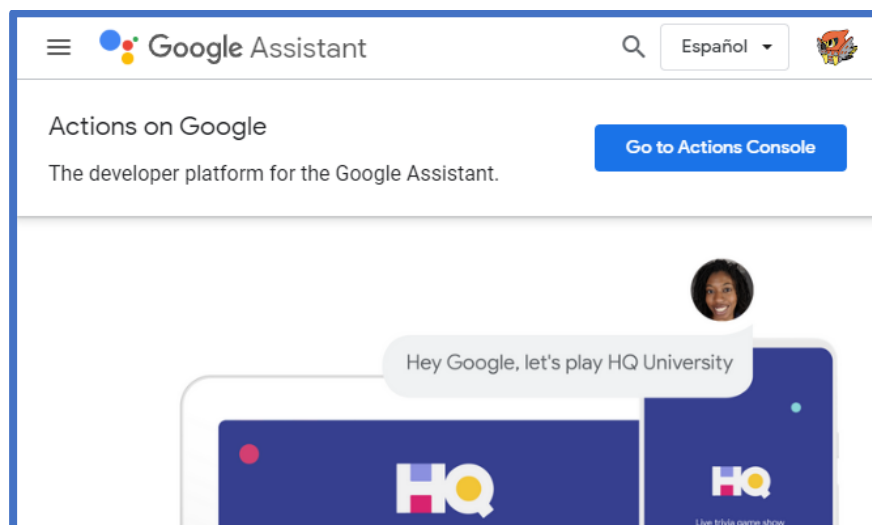
La que resultará más útil será la siguiente, la de **desarrolladores**.



Figura 27 Link de desarrolladores Google

Aquí se encuentra información sobre cómo desarrollar, de qué formas, que ideas se pueden tener y otro tipo de utilidades. Además, en esta web es donde se encuentra la documentación oficial del asistente. Por último, hay un apartado de comunidad donde se puede preguntar y resolver dudas sobre el tema que se elija en comunidades como github o stackoverflow junto con un botón de ayuda por si se necesita asistencia por parte de Google Assistant. Una vez se tenga presente la información que se tiene aquí se podrá pasar al siguiente paso. Ahora que se saben las diferencias entre los dos enlaces de Google Assistant se podrá encontrar el link a Actions Console, el cual hay que visitar para empezar a desarrollar las acciones de Google

Figura 28 Indicación de posición de Actions Console



Una vez se esté en la consola, se apreciará esta interfaz, desde la cual se podrá crear un proyecto además de ver los que ya existen para editarlos o gestionarlos. También se encuentra un link a la documentación, códigos de ejemplo, información sobre la API y soporte en caso de que se necesite ayuda.

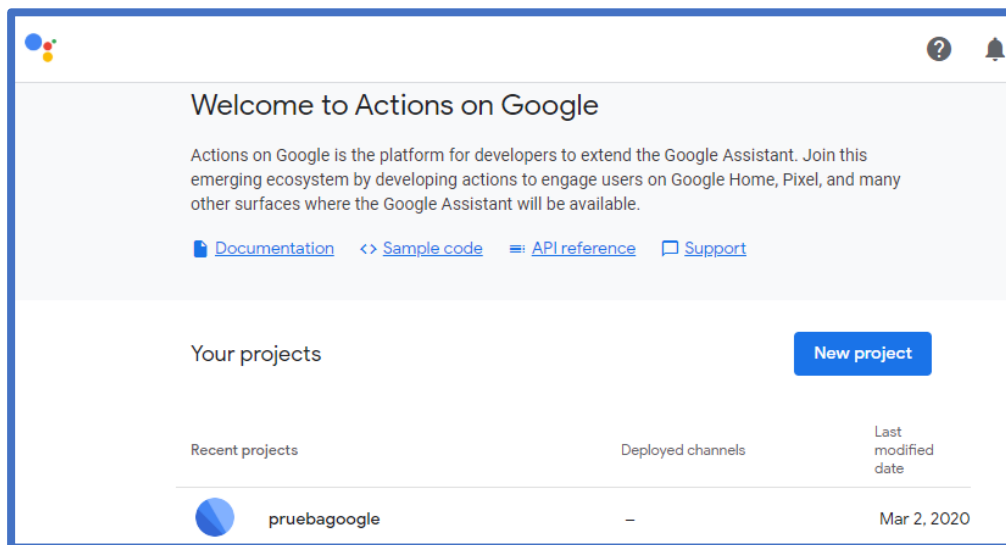


Figura 29 Menú principal de la Actions Console

Para empezar, se creará un nuevo proyecto, en el que se selecciona un nombre, un idioma y el país en el que se vaya a crear

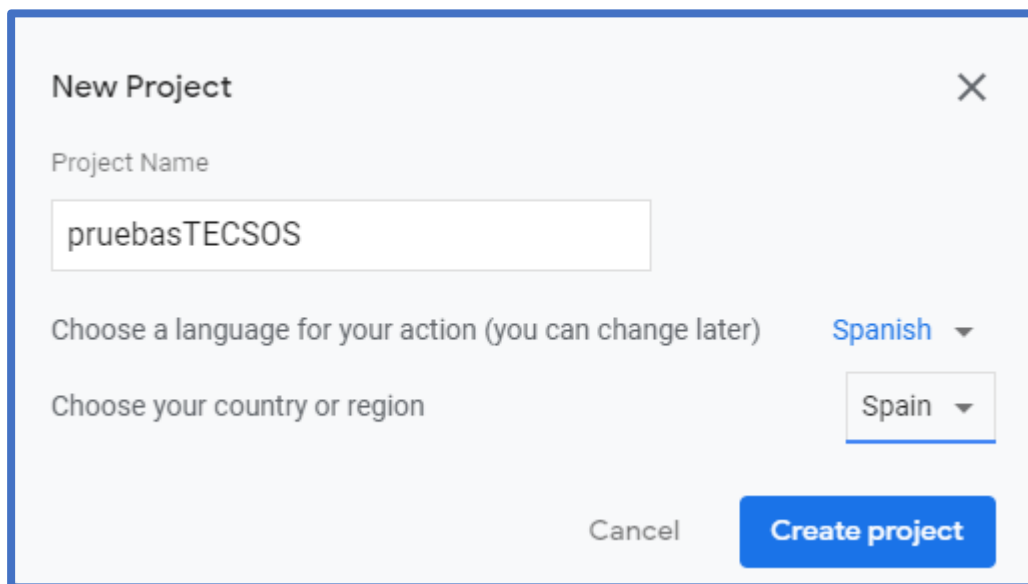
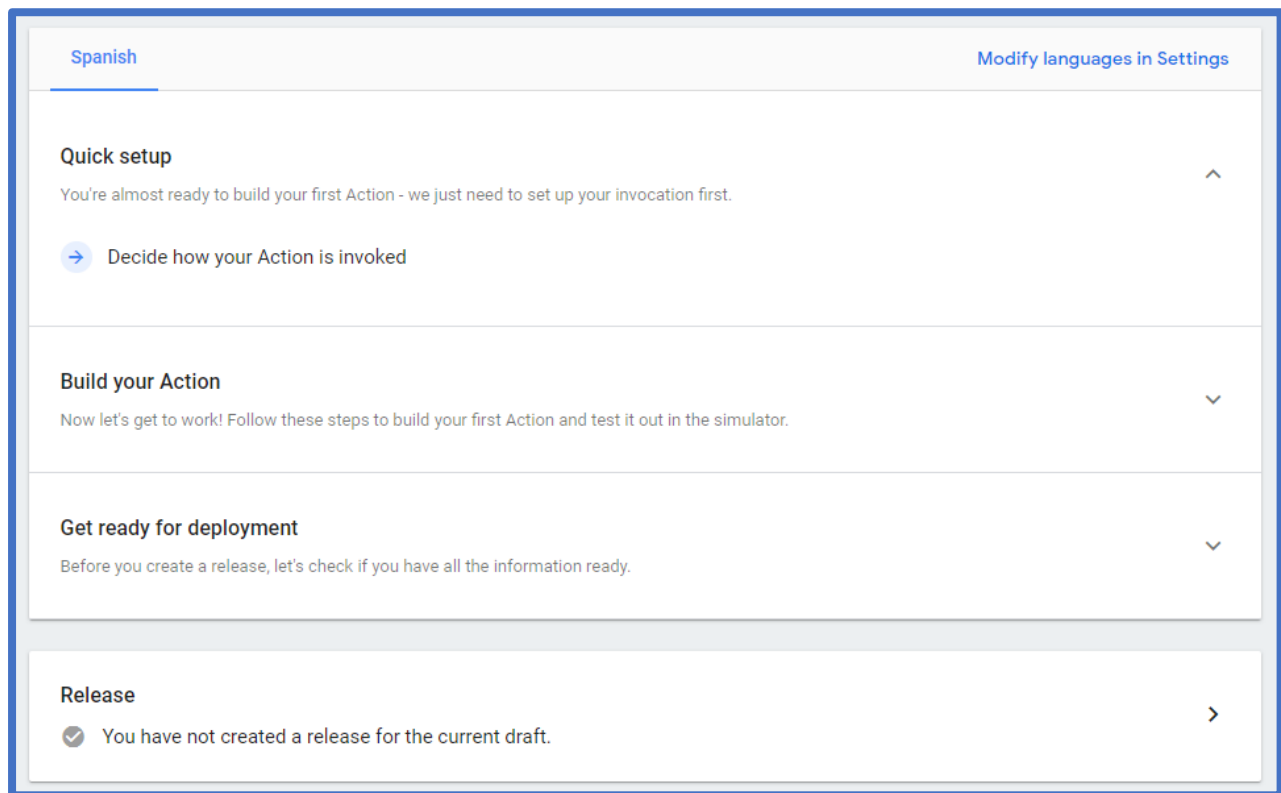


Figura 30 Creación de un nuevo proyecto Google

Una vez creado se pasará a la siguiente pantalla, que se corresponde con la parte de **Overview**, en la cual se ofrecen distintas posibilidades con relación a las acciones.

Se puede crear una acción, preparar una para subirla (públicamente) o ver las entregas de la misma que se han hecho hasta el momento. Como en este ejemplo se acaba de crear el proyecto no existen acciones ni entregas hechas, por lo que no se verá nada.



*Figura 31 Opciones y detalles de acción Google*

Mientras, en la barra de arriba, existe un navegador por distintas partes para configurar la Google Action, mostrado en la Figura 32.

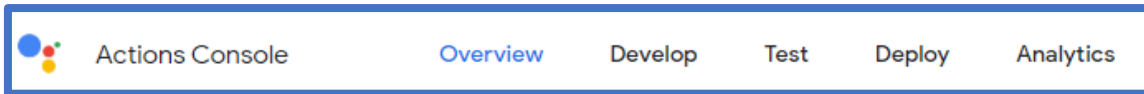


Figura 32 Barra de navegación de los detalles de Google Action

La parte que se ha explicado anteriormente es la de “Overview”, la cual ofrece un pequeño resumen o vista por encima de lo que se tiene.

Pasando a la siguiente opción, **Develop** permite cambiar y crear la parte más técnica de la Acción. Aquí se encuentra lo necesario para modificar y crear el Invocation name, la propia acción en sí, la personalización de los temas, su enlace a cuentas y los servicios de Backend.

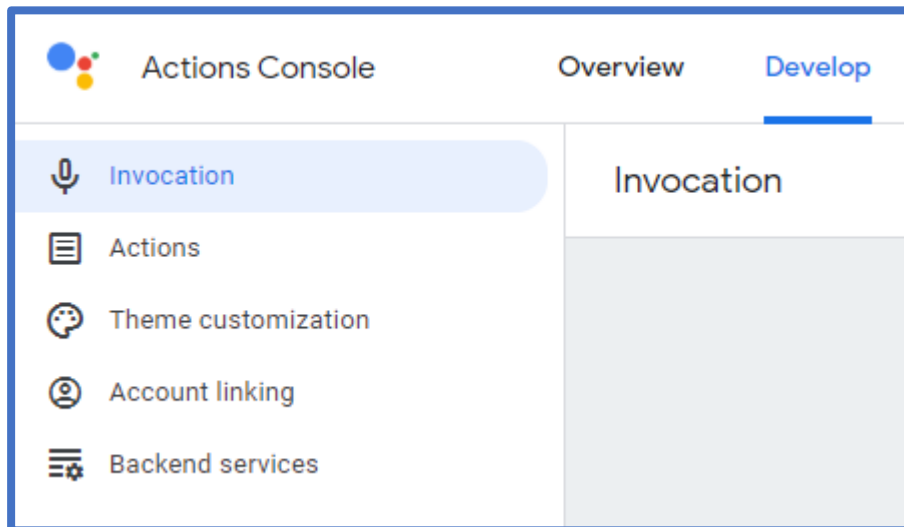


Figura 33 Parte de Development de una Google Action

En la parte de **Test** se encuentra lo necesario para probar y ver cómo funciona lo que se lleva desarrollado de Acción. Se podrá elegir en que dispositivo probarlo, en el idioma que se quiera y desde el sitio que se desee. Además, esta característica permite ver las Request y Responses, que son lo que la Acción y Google Assistant intercambian para hacer funcionar lo que se ha creado hasta el momento, será muy útil para detectar fallos o ver cómo funciona.



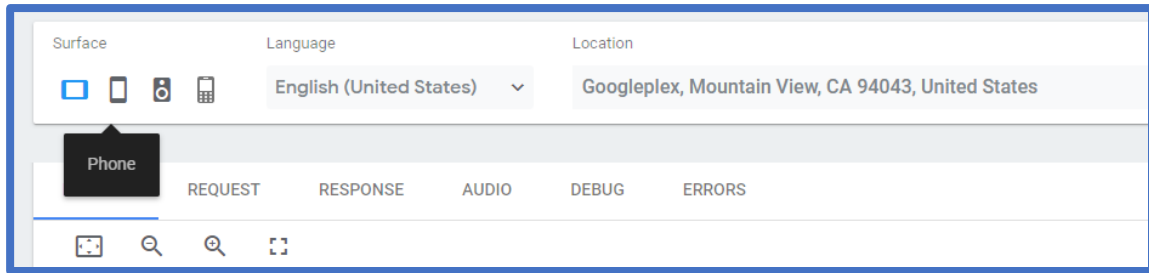


Figura 34 Parte de Testing de una Google Action

Finalmente, en la parte de **Deploy** se encuentran todos los campos y opciones a rellenar para poder subir la Acción. Se deberá completar los huecos como descripción, imágenes o verificaciones para conseguir su publicación. Esto se debe hacer cuando la Acción tenga un mínimo de funcionalidad y pueda ser desplegada sin errores.

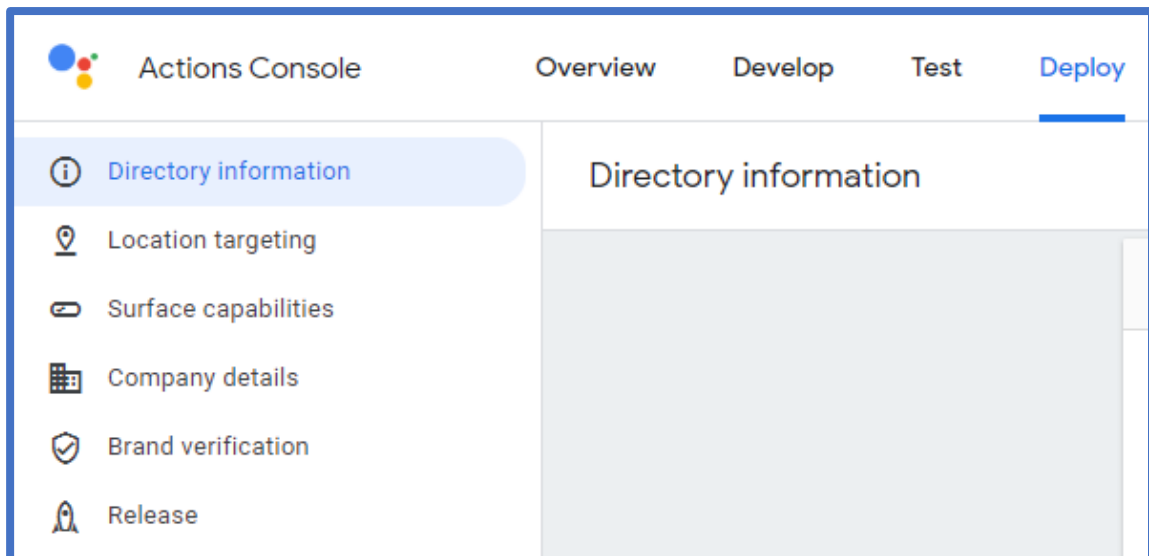


Figura 35 Parte de Deploy de una Google Action

Adicionalmente, cuando ya se haya publicado la Acción, se podrá ver la retroalimentación de la misma en la última parte del menú, **Analytics**. Se podrá hacer un seguimiento del uso que se le da desde la parte de Usage. Si tiene que ver con la salud, también existe un apartado dedicado a este

ámbito en Health. En la parte de Discovery se verán cuales son las formas más populares por las cuales los clientes encuentran la Acción que se ha creado. Además, se podrá ver que piensan de la Action creada mediante valoraciones o comentarios en la parte de Directory, teniendo también la posibilidad de exportarla en la última opción de Export.

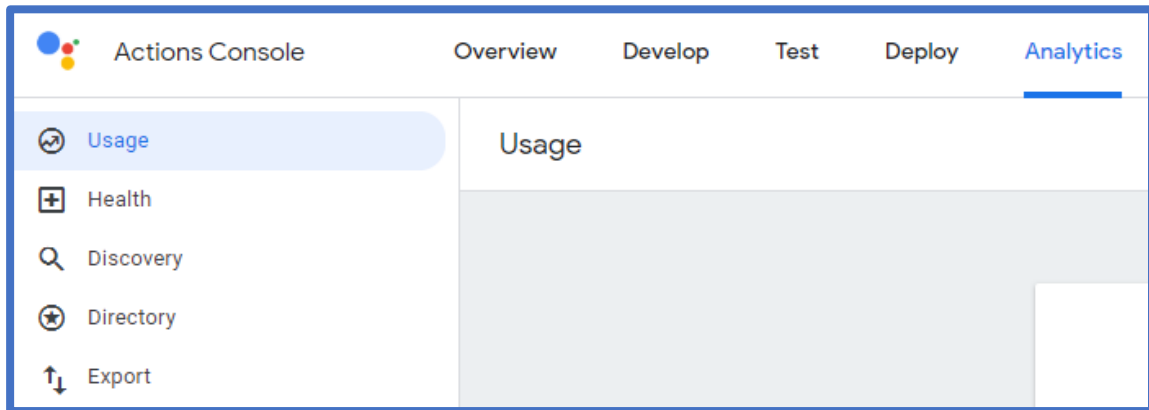


Figura 36 Parte de Analytics de una Google Action

### 3.3.4 Modificación del Código

Para modificar todo lo relacionado directamente con la Acción se utilizará una herramienta distinta dentro de la Consola de Acciones, DialogFlow

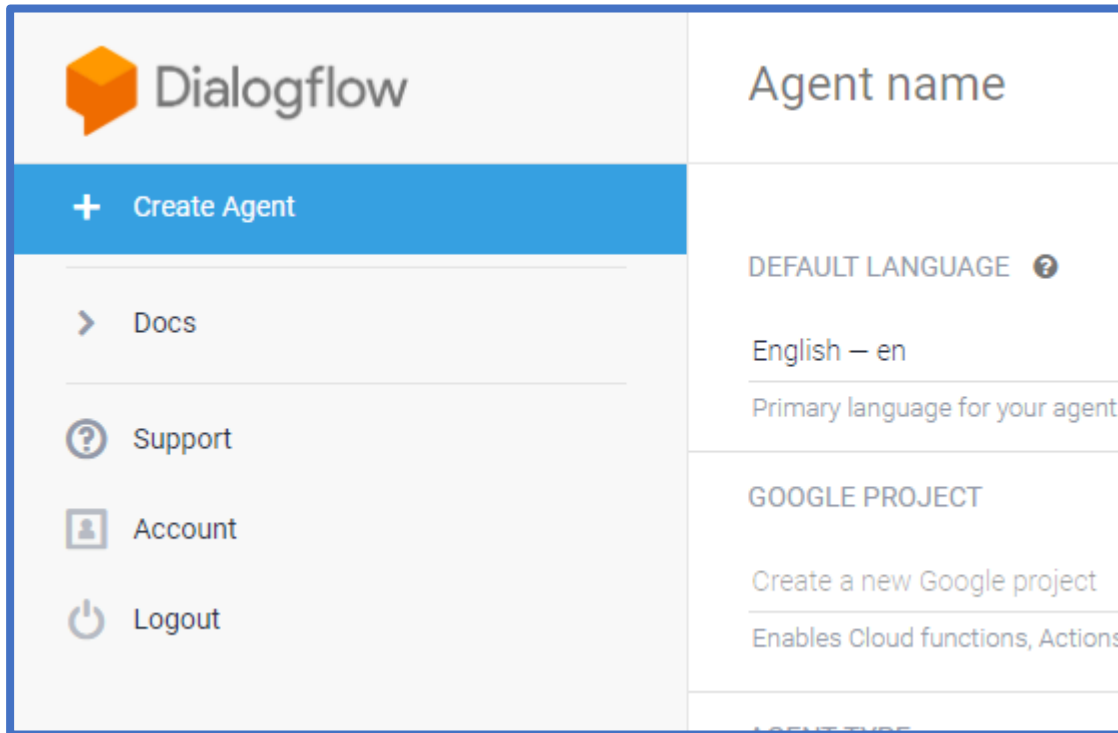


Figura 37 Dialogflow de Google Actions

Desde aquí se puede modificar el código de la Acción. Lo primero que se debe hacer es crear un “Agent”, que permitirá continuar con toda la gestión de la Acción.

La modificación del código se realizará desde el apartado de “fullfillment” si hiciera falta, ya que esto solo se utilizará en caso de que se quiera dar una mayor funcionalidad a la acción.

### 3.3.5 Publicando la Acción

Una vez desarrollada la Acción se deberá publicar. Antes de hacer esto permanecerá en estado “**en desarrollo**”, hasta que finalmente se suba y pase a ser publicada en la tienda de Acciones.

Es necesario dirigirse a la parte de Distribución y rellenar toda la información que nos solicitan sobre la Acción. También hay que subir dos iconos y rellenar unos criterios de Test.

Una vez hecho esto pasará al estado de “**En Revisión**”. Si no es aceptada, se indicará lo que hay que hay que corregir vía correo electrónico para que sea corregida y pase exitosamente el proceso la siguiente vez que se envíe.

#### ¿Se puede utilizar privadamente la Acción creada?

De momento **no se pueden** utilizar de forma privada tus propias Acciones. Esto se debe a que desde el estado de “en desarrollo” pasa a “publicado”, lo cual implica que sale públicamente en el Acción Store de Google. **Hay una posibilidad** de usar la Acción que se ha creado de forma privada, aunque no está hecho para utilizarse de esta manera. Consiste en dejar la acción en la fase de “en desarrollo” y no publicarla, para que nunca pase a ser pública.

Al utilizar este método, solo se tendrá la Acción la cuenta de desarrollador en la que está siendo desarrollada. En caso de querer que la Acción sea utilizada por más gente se deberá compartir esa misma cuenta de desarrollador en la que reside nuestra Acción. Puede que se reciban avisos de Google sobre por qué no se publica la Acción, ya que como se ha mencionado antes, el estado de “en desarrollo” no está dedicado para esto

## 4. Demostrador

Como complemento práctico al trabajo anteriormente expuesto se ha realizado un demostrador para dar una idea general del funcionamiento y de la creación de las skills de Alexa. Abarcará solamente la parte de voz, sin utilizar imágenes o vídeos en caso de que hubiera pantalla que los posibilitara.

El demostrador consiste en una Skill para Adherencia médica a los tratamientos, es decir, tomar la medicación de acuerdo con la dosificación del programa prescrito; y la persistencia, tomar la medicación a lo largo del tiempo. Este proceso es muy importante e indispensable para que los tratamientos se completen satisfactoriamente.

Contrariamente a lo que se tiende a pensar, la adherencia no significa tan sólo que nos tomemos la medicación. Aunque es un término que durante mucho tiempo se ha asociado al tratamiento farmacológico, la Organización Mundial de la Salud (OMS) incluye una serie de pautas que llevan a una buena adherencia

- Toma correcta de la medicación.
- Asistencia a las revisiones médicas que cada paciente necesite.
- Hábitos de vida saludables.

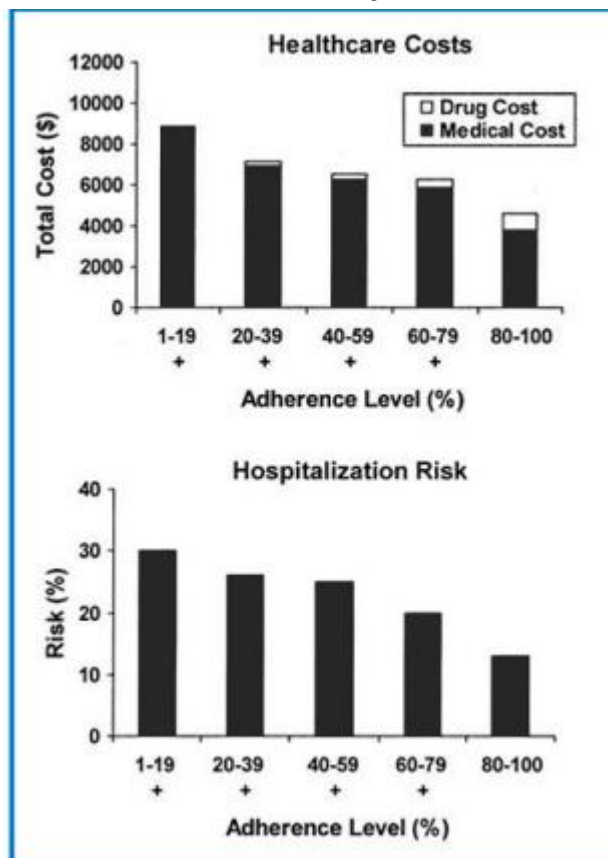
El impacto que tiene la no adherencia no se suele tener en cuenta, lo que provoca un empeoramiento de la enfermedad, en un incremento de la morbi-mortalidad, en la disminución de la calidad de vida y en la necesidad de administrar otros tratamientos, además de aumentar el consumo de recursos sanitarios, por lo que es de vital importancia controlarlo. Además, al no ser un problema del que se tenga especial conciencia hace que sus empeoramientos sean más graves, puesto que los

pacientes normalmente no tienen una buena adherencia médica y ni si quiera ellos mismos lo saben, lo que hace completamente imposible resolver el problema.

Existen varias causas o barreras que producen este efecto que son entre otras:

- Problemas con el régimen prescrito (efectos adversos)
- Instrucciones insuficientes
- Fallo en la relación médico-paciente
- Desacuerdo del paciente respecto al tratamiento
- Mala memoria

Tabla 7 Disminución de costes en pacientes diabéticos



Con el demostrador se pretende paliar las barreras anteriormente descritas. Gracias a las cualidades y oportunidades que otorgan los asistentes virtuales será de gran utilidad para ayudar a mantener y mejorar la adherencia médica de muchas personas que, por diversas razones, tiene dificultades para ello. Este desarrollo ha sido dividido en las siguientes funcionalidades:

### **Funcionalidad 1:**

#### **Recordatorio sonoro a la hora específica de toma de Medicación**

Realizar un aviso sonoro a la hora adecuada es de vital importancia puesto que de acuerdo con los estudios realizados [10], el 39% de los errores de toma se deben a olvidos, además de que en los pacientes que no tienen bien definido el horario de la toma, existe un 80% de posibilidades de que se olvide o retrase. Con este aviso se reducirá drásticamente este problema y mejorará el porcentaje de tomas exitosas.

### **Funcionalidad 2:**

#### **Especificación por voz de la información necesaria de toma de Medicación**

Otra barrera más a la hora de tener una buena adherencia médica es la falta de instrucciones o desconocimiento, los pacientes pueden fallar dada su falta de instrucciones u olvidos. Con la especificación por voz de la información necesaria se brindará al paciente con todo lo necesario para que no ocurran problemas por el desconocimiento, esto incluye información del medicamento, de la forma de tomarlo, de las instrucciones específicas de cada uno, la utilidad o razón de uso y su importancia médica.

### **Funcionalidad 3:**

#### **Insistencia a la hora de tomar el medicamento**

Cuando un medicamento no es tomado de forma correcta o en el momento adecuado es importante notificarlo e insistir en su importancia. Se hará hincapié en que se debe tomar el medicamento e insistir en que se realice siguiendo las instrucciones.

### **Funcionalidad 4:**

#### **Apunte de las veces que se toma o no el medicamento**

Cada vez que se deba tomar un medicamento es crucial, mediante un apunte de tomas se conseguirá un seguimiento del paciente y de su nivel de adherencia, lo cual es muy necesario para realizar futuros diagnósticos o tomar decisiones posteriormente, además, se puede calcular el nivel de adherencia gracias a esto, con la siguiente fórmula

$$\text{Adherencia} = \frac{\text{Unidades dispensadas} - \text{Unidades deshechadas} \times 100}{\text{Unidades prescritas}}$$

*Figura 38 Fórmula Adherencia médica*

### **Funcionalidad 5:**

#### **Informe semanal sobre la toma de medicamentos**

Una vez realizado el apunte de tomas y recabado los datos necesarios, será de especial utilidad hacer un informe semanalmente que informe de la toma. Este informe podrá ser enviado a otras



personas que estén interesadas en la información del paciente, ya sean familiares o personal médico.

### **Funcionalidad 6:**

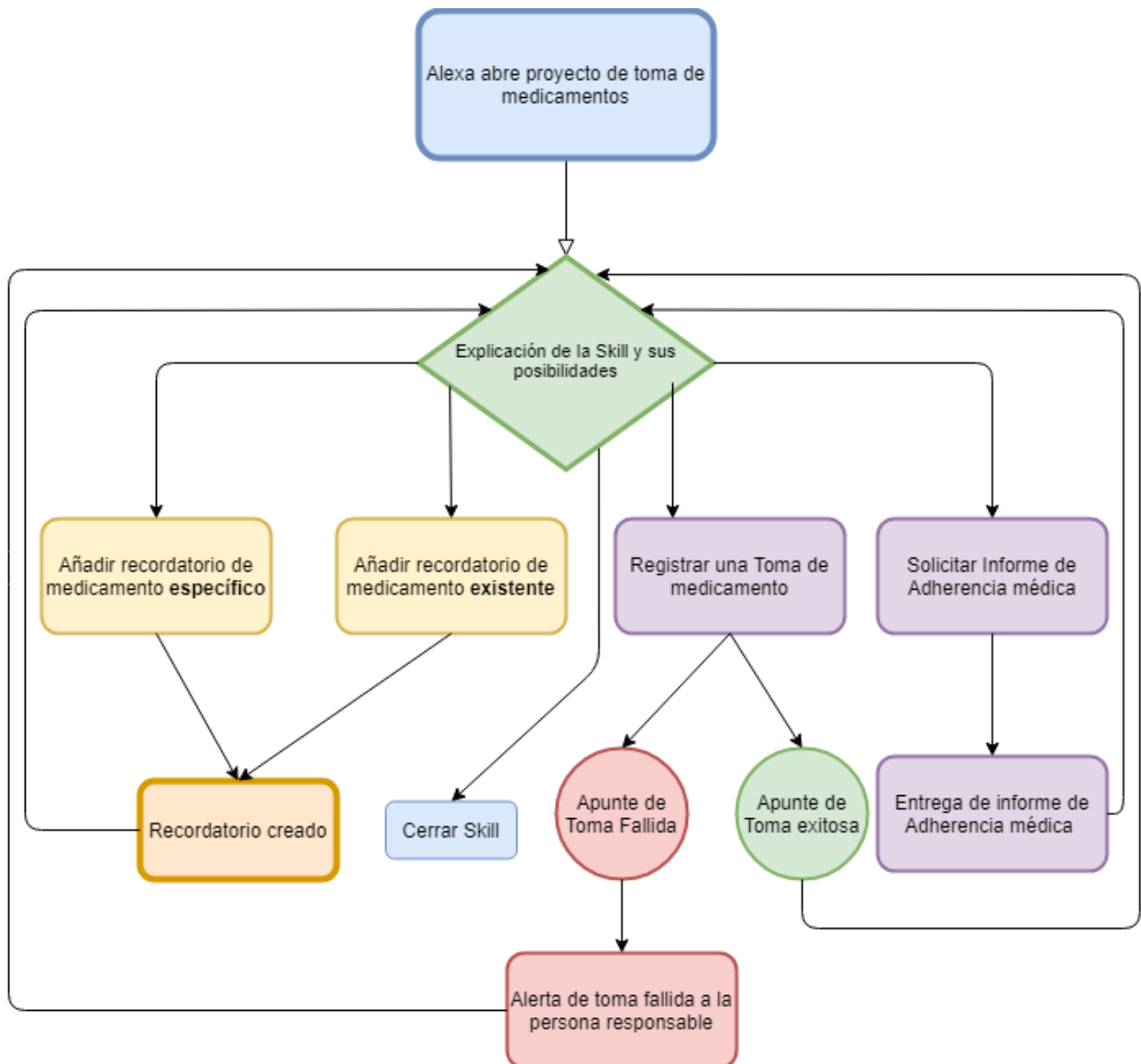
#### **Avisos de alerta en caso de que no se tome la medicación**

Cada vez que la medicación no es tomada el nivel de adherencia decrece, siendo esto un riesgo muy peligroso, se alertará a quien se desee cuando un usuario no se tome la medicación para avisar de que ha habido algún problema o que ha ocurrido algo por lo que no se ha tomado correctamente. El método de aviso se especificará más adelante, cabiendo las posibilidades de mensaje SMS, correo electrónico o llamada telefónica.

## Diagrama de estados del demostrador

En el siguiente diagrama se aprecian las distintas funcionalidades mencionadas anteriormente junto al flujo común que cualquier Skill debe tener, como es el de comenzar con una introducción para derivar consecuentemente a los distintos caminos elegidos por el usuario, para finalmente volver al punto anterior desde el que poder retomar cualquier otra posibilidad. También se muestra la opción de terminar la sesión cuando se desee acabar la interacción.

Tabla 8 Diagrama de estados Skill



# Implementación del Software del demostrador

## Implementación de la funcionalidad 1:

### Recordatorio sonoro a la hora específica de toma de Medicación

Para trabajar con recordatorios en Alexa lo primero que se debe hacer es activar los permisos en la consola. En el apartado de permisos hay que dejar activa la opción de recordatorios

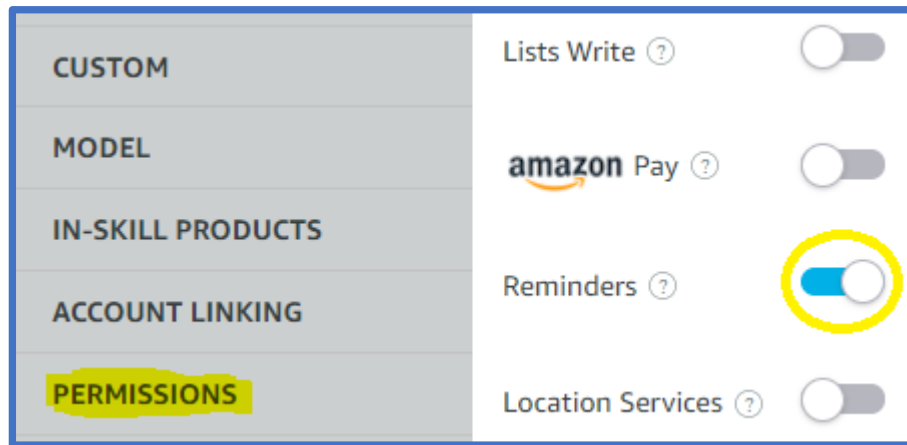
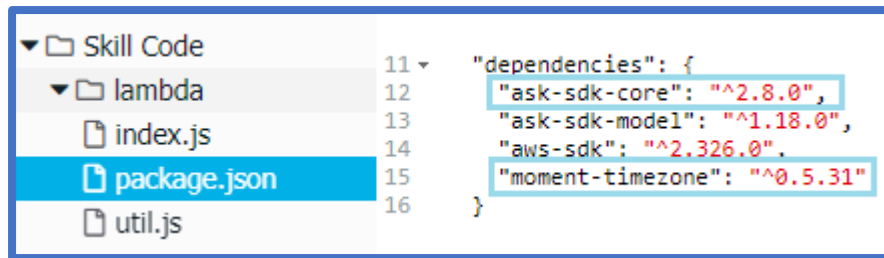


Figura 39 Activación de Permisos de notificaciones

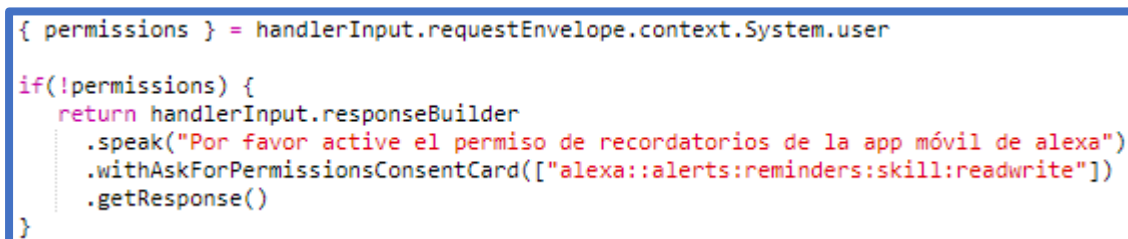
Una vez activados ya será posible implementarlos en la parte de código. El siguiente paso es ir al archivo package.json y revisar la versión del ask-sdk-core además de añadir la dependencia de **moment-timezone**, ya que al utilizar recordatorios necesitaremos una librería más sofisticada que se encargue de gestionar las horas y el tiempo. En este caso se ha cambiado la versión del ask-sdk-core a la más reciente, la 2.8.0 y se ha añadido la librería moment-timezone junto con su versión más reciente, la 0.5.31.



```
11  "dependencies": {
12    "ask-sdk-core": "^2.8.0",
13    "ask-sdk-model": "^1.18.0",
14    "aws-sdk": "^2.326.0",
15    "moment-timezone": "^0.5.31"
16  }
```

Figura 40 Agregación y actualización de las dependencias

Al tener toda la configuración puesta a punto la siguiente etapa será la confirmación de notificaciones en el dispositivo móvil. Por defecto la App de Alexa necesita permiso para que una Skill en específico muestre notificaciones en el dispositivo móvil. Antes de crear un recordatorio es esencial que Alexa asegure si estos permisos están concedidos o no, por lo que se implementarán las siguientes líneas de código para ello.



```
{ permissions } = handlerInput.requestEnvelope.context.System.user

if(!permissions) {
  return handlerInput.responseBuilder
    .speak("Por favor active el permiso de recordatorios de la app móvil de alexa")
    .withAskForPermissionsConsentCard(["alex::alerts:reminders:skill:readwrite"])
    .getResponse()
}
```

Figura 41 Aviso de permitir los avisos en el dispositivo móvil

Primero se crea la variable **permissions** que indica si los permisos están concedidos y si esta es falsa entrara al siguiente bloque gracias al condicionador if que se activará cuando esto ocurra. En este bloque se emite el mensaje por voz pidiendo que se concedan los permisos además de mostrar en el dispositivo móvil el acceso directo para ello. Una vez se concedan los permisos no será necesario hacerlo de nuevo.

En el siguiente paso ya se podrá integrar el recordatorio, para ello hay que visitar la documentación de donde se obtiene el código base de un recordatorio. Se divide en tres partes distintas:

**Trigger:** En esta parte se detallan las características de activación y el tiempo, gracias a la librería que se configuró en pasos anteriores podremos gestionar las fechas. También se deberá de detallar la hora y la frecuencia del recordatorio.

**Alert Info:** Sirve para dar la información deseada del recordatorio, en este caso, la toma de un medicamento.

**PushNotification:** En esta parte se detalla si se quiere mostrar la notificación en el dispositivo móvil asociado por lo que se mantendrá en activo para mayor seguridad de aviso del recordatorio.

```
const currentTime = moment().tz("Europe/Madrid"),
reminderRequest = {
  requestTime: currentTime.format("YYYY-MM-DDTHH:mm:ss"),
  trigger: {
    type: "SCHEDULED_ABSOLUTE",
    scheduledTime: currentTime.set({
      hour: "00",
      minute: "25",
      second: "00"
    }).format("YYYY-MM-DDTHH:mm:ss"),
    timeZoneId: "Europe/Madrid",
    recurrence: {
      freq: "DAILY"
    }
  },
  alertInfo: {
    spokenInfo: {
      content: [{
        locale: "es-ES",
        text: "Te recuerdo que debes tomarte X ",
      }],
    },
  },
  pushNotification: {
    status: "ENABLED"
  }
}
```

*Figura 42 Esquema general de un recordatorio*

Como podemos observar en la imagen superior, el bloque sirve para que se cree un recordatorio diario a las 00:25 con el texto de "Te recuerdo que debes tomarte X".

Para finalizar la integración de recordatorios también es necesario un comprobador de errores, además de ser esencial para pasar las pruebas de publicación. Hay que añadir un tryCatch en la parte de la API de crear recordatorios para así comprobar cada vez que se vaya a crear uno que no ha habido ningún problema. En la siguiente imagen aparece como está implementado, en caso de ocurrir un error se mostrará en la consola, además de avisar a la persona de que ha habido un error al crear el recordatorio.

```
try {
  await remindersApiClient.createReminder(reminderRequest)
} catch(error) {
  console.log("~~~~~ createReminder Error ${error} ~~~~~")
  return handlerInput.responseBuilder
    .speak("Ha habido un error creando tu recordatorio, prueba de nuevo más tarde")
    .getResponse();
}
```

Figura 43 TryCatch para la detección de errores

## Implementación de la funcionalidad 2:

### Especificación por voz de la información necesaria de toma de Medicación

Una parte esencial de la adherencia es el conocimiento sobre el medicamento que se va a tomar, las restricciones médicas que tiene, la forma de tomarlo y cualquier otra información relevante que pueda aportarse para facilitar el proceso, si esa parte flaquea puede terminar siendo uno de los puntos por los cuales bajan los niveles de éxito de la adherencia, que acabaría derivando en problemas mucho más graves para la salud. Para llevar a cabo esta funcionalidad deberemos utilizar la parte de `AlertInfo` que tenemos en el recordatorio, hay que introducir el texto deseado en el bloque de `Content` en el apartado de `Text`. En este caso se introducirá un ejemplo de texto con algunos consejos para la toma de paracetamol

```
alertInfo: {
  spokenInfo: {
    content: [{
      locale: "es-ES",
      text: "Es la hora de tomar el medicamento Paracetamol"
    }],
  },
},
```

Figura 44 Detalle context-text

Este texto será leído por voz por Alexa, se puede cambiar dependiendo del medicamento o de la forma que se quiera tomar, así como las especificaciones médicas o del propio doctor.

## Implementación de la funcionalidad 3:

### Insistencia a la hora de tomar el medicamento

Para cumplir esta funcionalidad utilizaremos una herramienta que tiene Alexa a la hora de hablar, se llama Speech Synthesis Markup Language (SSML) y sirve para caracterizar más las frases que dice Alexa. Al utilizarlo en un texto a reproducir, se le pueden añadir opciones como son el énfasis, la emoción, exclamación, decir fonemas o números decimales o añadir pausas.

En la documentación [11] aparecen todas las opciones disponibles junto con sus recomendaciones y métodos de implementación. En el caso del demostrador y de la funcionalidad 3, la característica que más encaja es la de pausa, que permitirá realizar una pausa de hasta 10 segundos hasta volver a hablar, lo cual es perfecto para esperar el tiempo suficiente para volver a recordar cualquier acción, que en este caso será la toma de un medicamento.

Attribute	Possible Values
<code>strength</code>	<ul style="list-style-type: none"><li>• <code>none</code> : No pause should be outputted. This can be used to remove a pause that would normally occur (such as after a period).</li><li>• <code>x-weak</code> : No pause should be outputted (same as <code>none</code> ).</li><li>• <code>weak</code> : Treat adjacent words as if separated by a single comma (equivalent to <code>medium</code> ).</li><li>• <code>medium</code> : Treat adjacent words as if separated by a single comma.</li><li>• <code>strong</code> : Make a sentence break (equivalent to using the <code>s</code> tag).</li><li>• <code>x-strong</code> : Make a paragraph break (equivalent to using the <code>p</code> tag).</li></ul>
<code>time</code>	Duration of the pause; up to 10 seconds ( <code>10s</code> ) or 10000 milliseconds ( <code>10000ms</code> ). Include the unit with the time ( <code>s</code> or <code>ms</code> ).

Figura 45 Referencias Break

Como se aprecia en la Figura 45, existen dos métodos para introducir pausas, el primero es `strength`, el cual permite introducir las pausas naturales del habla como son las que hay entre oraciones o párrafos, va desde los más cortos que corresponden a las comas a los más duraderos



que corresponden con los puntos y aparte. El otro atributo que se puede utilizar es el de time, que introduce la pausa del tiempo que se desee, hasta los 10 segundos. En el caso del demostrador añadiremos una espera del máximo posible, además de poner pausas naturales entre las comas y los puntos, modificando el código de la siguiente forma:

```
content: [{
  locale: "es-ES",
  text: "Es la hora de tomar el medicamento Paracetamol.
        El paracetamol es eficaz para reducir el dolor y la fiebre, debes tomar 1 comprimido cada 4-6 horas, según necesidad.
        Recuerda que es importante mantener una buena adherencia médica para recuperarte más fácilmente",
  ssml: "<ssml>
        <p>Es la hora de tomar el medicamento Paracetamol</p>
        <s> El paracetamol es eficaz para reducir el dolor y la fiebre </s>
        <s> debes tomar 1 comprimido cada 4-6 horas, según necesidad. </s>
        <break time='10s' />
        <p> Recuerda que es importante mantener una buena adherencia médica para recuperarte más fácilmente </p>
        </ssml>"
}]
```

Para poder llevar a cabo la tarea se debe realizar de la siguiente forma, en el primer bloque llamado text, figura el texto simple que se tiene que decir, y luego debajo en el bloque ssml es donde se introducen las reglas que se seguirán a la hora de expresar el texto escrito. En este caso se han añadido espaciados de tiempo de párrafo con el comando <p> y espaciados de tiempo de líneas con el comando <s>, hay que iniciarlos cuando empieza la frase y acabarlos con el mismo comando añadiendo una barra de la siguiente forma: </p> </s>

Además, se ha añadido el comando <break time="10s"/> para que se realice una pausa mayor entre el aviso de toma de medicamentos y la insistencia de su importancia, creando así un parón temporal que hará posible cumplir con la funcionalidad a desarrollar.

## Implementación de la funcionalidad 4:

### Apunte de las veces que se toma o no el medicamento

A la hora de abordar esta parte del desarrollo resulta obvio que se deberá contar con un método para poder registrar el número de veces que se toma de forma exitosa o fallida el medicamento, ya que la propia Skill es incapaz de registrar de forma permanente los atributos, resultando en que no es posible retenerlos más allá de la sesión. Habrá que utilizar una base de datos externa para realizar la cuenta de forma persistente en el tiempo. Dado que se están utilizando numerosos recursos de AWS explicados anteriormente (como por ejemplo el lambda) lo más sencillo para desarrollar esta funcionalidad será la utilización de DynamoDB, que es una base de datos NoSQL enlazable con nuestra Skill.

Desde la consola de AWS se deberá acceder al apartado de servicios, y en la parte de Bases de datos se deberá seleccionar DynamoDB

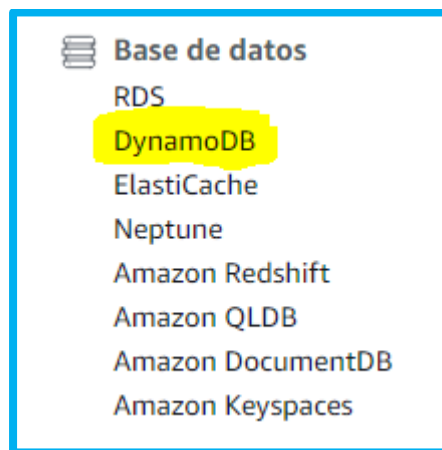


Figura 46 Bases de datos AWS

Se ha elegido DynamoDB entre el resto de las opciones dado su carácter administrativo que ofrece un desempeño simple, rápido y predecible, así como una gran escalabilidad.

Una vez seleccionado el servicio de DynamoDB ya se podrá realizar la creación de tablas. Para este caso se creará una única tabla ya que son pocos datos los que hay que manejar. Son necesarios pocos campos a rellenar para este proceso, con indicar el nombre y la clave principal, junto a la elección de configuración predeterminada será suficiente.

Crear una tabla de DynamoDB Tutorial ?

DynamoDB es una base de datos sin esquema que solo necesita un nombre de tabla y una clave principal. La clave principal de la tabla está compuesta de uno o dos atributos que identifican de manera inequívoca cada elemento, efectúan la partición de datos y ordenan los datos dentro de cada partición.

Nombre de la tabla\*  ⓘ

Clave principal\* Clave de partición  Cadena ⓘ

Añadir clave de ordenación

Configuración de la tabla

La configuración predeterminada proporciona la forma más rápida de comenzar con la tabla. Puede modificar esta configuración predeterminada ahora o después de crear la tabla.

Usar la configuración predeterminada

- No hay índices secundarios.
- Capacidad aprovisionada establecida en 5 lecturas y 5 escrituras.
- Alarmas básicas con umbral superior al 80% que usan el tema de SNS "dynamodb".
- Cifrado en reposo con el tipo de cifrado PREDETERMINADO.

Figura 47 Creación de Tabla

Terminada la creación de la tabla se pueden editar las preferencias y visualizar todos los parámetros de la base de datos. Entre ellos se encuentran la información general, las métricas, alarmas, capacidad, índices, tablas globales, copias de seguridad, desencadenadores, etiquetas o controles de acceso. Los propios datos se encuentran en la parte de “elementos” como indica la Figura 48.

dynamodb-starter Cerrar

Información general **Elementos** Métricas Alarmas Capacidad Índices Tablas globales Copias de seguridad Contributor Insig

Crear elemento Acciones

Examen: [Tabla] dynamodb-starter: userid, numeroDeTo...

Examen [Tabla] dynamodb-starter: userid, numeroDeToma

+ Añadir filtro

Iniciar búsqueda

userid ⓘ	numeroDeToma	Negativo	Positivo
43T234D2T3	12	6	6

Figura 48 Características de BBDD

El siguiente paso es gestionar los permisos para que nuestra Skill tenga la posibilidad de operar con la base de datos. De una forma similar a como se realizó anteriormente con los permisos para recordatorios se deberá acceder a la página en la que se gestiona todo lo relacionado con este tema.

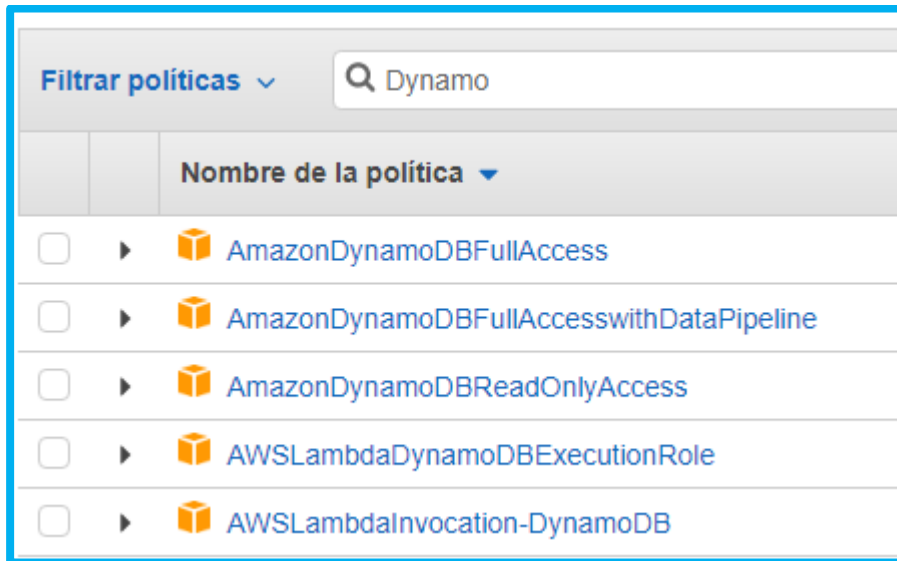


Figura 49 Políticas DynamoDB

De entre todos los permisos se puede filtrar mediante el buscador para visualizar solo los que se deseen. En este ejemplo se utilizará el filtrador para ver los permisos asociados a DynamoDB. De entre los existentes en la Figura 49 se escogerá el primero ya que brinda un acceso completo a la base de datos.

Una vez que todo lo anterior se ha configurado ya se puede trabajar con estos servicios, continuando de la siguiente forma. Se iniciará declarando varios parámetros para que la conexión sea posible con la Skill.

```
const dynamodb-starter = 'tablaNumeroDeTomas';  
const docClient = new awsSDK.DynamoDB.DocumentClient();
```

Figura 50 Declaración de la tabla de DynamoDB

De esta forma se añade la tabla al código, en este caso con el nombre de tablaNumeroDeTomas que será el que se usará para referirse a la tabla creada anteriormente en DynamoDB, con el nombre real de dynamodb-starter. El siguiente paso consiste en actualizar los datos de la tabla para así poder asegurar su persistencia. En la Figura 51 se puede apreciar el bloque de código encargado de esta tarea que será detallado para su mejor comprensión.

```
const tomaPositiva = slots.tomaPositiva.value;
const tomaNegativa = slots.tomaNegativa.value;
const tomasTotales = tomaPositiva + tomaNegativa;

const dynamoParams = {
  TableName: tablaNumeroDeTomas,

  Item: {
    UserId: userId,
    numeroDeTomas: tomasTotales,
    Positivo: tomaPositiva,
    Negativo: tomaNegativa,
  }
};
```

*Figura 51 Bloque de actualización de datos*

En la primera parte aparecen tres contantes que coinciden con las posibilidades de toma y el número total de las mismas. Estas constantes obtendrán el valor mediante el método slot.x.value mostrado en la imagen, que significa que se les otorgará el valor del slot en cuestión o por medio de las otras dos constantes, como es el caso de tomasTotales. Estos slots son los que se rellenan mediante la voz en los distintos intent, por lo que para abordar este caso se han creado 2 slots distintos, tomaPositiva para recoger los éxitos de toma y tomaNegativa para recoger los intentos fallidos. Cuando Alexa recibe el aviso de si se ha tomado o no la medicación actualiza los slots mencionados anteriormente que, de forma normal, no quedarían guardados más allá de lo que dure la conversación con Alexa. Se podría haber creado un tercer slot para tomasTotales, pero al coincidir

este con la suma de los dos anteriores resulta un poco recargado e ineficiente, se puede solucionar de forma sencilla obteniendo el número de esa constante mediante el cálculo de la suma de las otras dos. Una vez que está claro que Alexa actualiza los slots y que estos no tienen memoria de por sí el siguiente paso es muy fácil de comprender.

La siguiente parte del bloque de código coincide con la actualización de la tabla de la base de datos, es en este punto donde los datos volátiles de una sesión de Skill de Alexa pasan a ser persistentes y utilizables en un futuro. Indicando el nombre de la tabla y de sus atributos, que ya se describieron anteriormente, es tan sencillo como rellenar los datos con las constantes que antes fueron recogidas. En la parte izquierda residen los atributos de la tabla de la base de datos, se debe apreciar que han de coincidir con la misma, mostrada en la Figura 48. En la parte derecha se encuentra lo que rellenará estos datos de la tabla, que son las constantes que hay en la parte de arriba del bloque del código, primero son rellenadas (por los slots o mediante una suma) y luego son introducidas a la base de datos.

Respecto al `userId`, es un atributo obligatorio para las sesiones de Alexa y de la funcionalidad lambda, por lo que debe aparecer y ser clave primaria. Esto sucede debido a que en la base de datos se guardarán muchos casos de pacientes y de sus tomas exitosas/fallidas, por lo que se tendrán que distinguir por el `userId` diferente de cada uno, de otra forma se mezclarían todas las tomas positivas y negativas de todos los pacientes que utilizaran la Skill.

## **Implementación de la funcionalidad 5:**

### **Informe semanal sobre la toma de medicamentos**

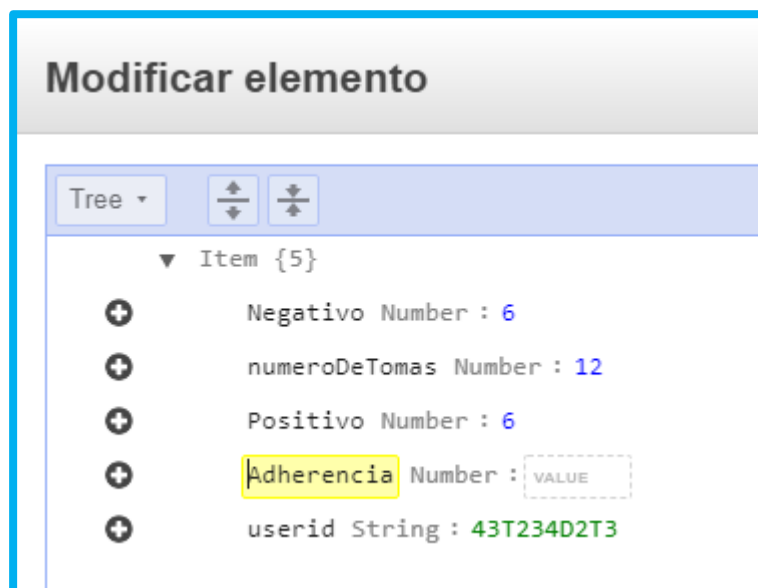
Al igual que es de vital importancia mantener la cuenta de las tomas exitosas y fallidas como se realiza en la funcionalidad 6, es igual o más relevante el uso de estos datos para obtener conclusiones o deducciones para poder así tomar medidas en función de la gravedad de los datos.

En esta funcionalidad se explicará cómo realizar un informe para estar al tanto de lo que sucede y de como sucede. Para poder analizar mejor los datos se utilizará como medida la propia adherencia médica del paciente usuario. Esta fórmula viene explicada en la **¡Error! No se encuentra el origen de la referencia.** y expresa lo siguiente:

La Adherencia se podría calcular restando las unidades desechadas a las dispensadas, dividido entre las prescritas y multiplicado a su vez por cien para así pasarlo a la forma de porcentaje, donde se puede analizar de mejor forma. De forma sencilla se puede entender como restar las unidades que no se toman a las unidades totales, dividido de nuevo entre las unidades totales y multiplicado por cien, dando así el porcentaje de éxito de tomas respecto al número total de tomas.

Para realizar este apartado se necesitará crear otro atributo en la tabla, puesto que con los existentes se queda corto para poder calcular el nuevo objetivo. DynamoDB permite la agregación de atributos en tablas ya creadas, por lo que será muy sencillo realizar este paso, desde la consola de AWS en la parte de la tabla hay que acceder a la parte de elementos y ahí seleccionar la opción de editar.

Una vez se hayan realizado esos pasos ya existirá la posibilidad de añadir el nuevo atributo, como se aprecia en la Figura 52, se ha agregado el atributo de adherencia a los ya existentes. Mediante los antiguos atributos se podría calcular este valor, pero es completamente necesario crear el nuevo hueco para poder guardarlo y actualizarlo en un futuro. Ahora que ya existe el atributo de adherencia será muy fácil manejarlo y utilizarlo para dar la información deseada.



*Figura 52 Atributo de adherencia añadido*

En el bloque de código se expresará de la siguiente forma, siendo calculado mediante las tomas positivas dividido entre las negativas y a su vez multiplicado por cien, utilizando así todos los atributos creados. Aunque parezca que el atributo de tomas negativas resulta inútil a la hora de calcular la adherencia es de vital importancia puesto que sin ese dato no se podría calcular el número de tomas totales.



```
const tomaPositiva = slots.tomaPositiva.value;
const tomaNegativa = slots.tomaNegativa.value;
const tomasTotales = tomaPositiva + tomaNegativa;
const adherencia = tomasPositivas / tomasTotales . 100;
```

*Figura 53 Constantes actualizadas*

Ahora que se tiene el atributo de adherencia creado en la base de datos y definido junto a su cálculo en la Skill de Alexa solo queda la parte de solicitarlo para así recibir la información. Para ello se utilizará el siguiente bloque de código, el cual incluye el texto que se recibirá al solicitar el informe. Indica una breve introducción sobre la adherencia para que el usuario no tenga un contacto demasiado brusco con un número que a priori podría no conocer.

```
this.emit(':tell', `La Adherencia médica se calcula mediante el número de tomas positivas y negativas respecto del total de las mismas.Actualmente la adherencia del paciente es de ${adherencia} sobre 100.`);
```

*Figura 54 Texto de solicitud de adherencia*

Para obtener el dato se utiliza la notación mostrada, que resultará en el número que reside en la base de datos, acto seguido mediante el texto se añade que es sobre cien, puesto que en la base de datos solo se guarda el propio número, siendo necesaria la explicación de que esto es un porcentaje.

## Implementación de la funcionalidad 6:

### Avisos de alerta en caso de que no se tome la medicación

Para desarrollar esta funcionalidad se hará uso del sistema de alarmas de AWS, que permite la opción de recibir notificaciones en caso de que ciertas características se cumplan. El tipo de comunicación de la alerta es seleccionable y se ha decidido finalmente que sea mediante correo electrónico dada la comodidad que ofrece, aunque habrían sido válidas otras opciones.

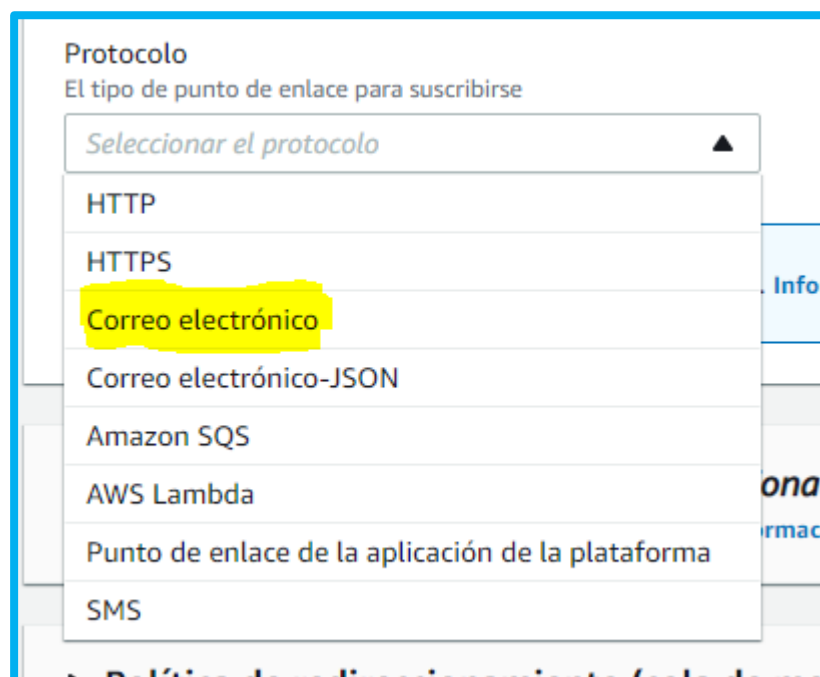


Figura 55 Tipo de comunicación

Lo primero que se debe hacer es crear la alarma y rellenar todos sus campos, esto es posible accediendo a la web de AWS en el apartado de cloudwatch, que es el apartado que se encarga de manejar los Logs del lambda que carga la Skill.

Al crear la alarma lo primero que hay que identificar es su métrica, que decidirá cómo será la activación de la alarma. En este caso de uso, al estar relacionado con las tomas negativas, se decantará por seleccionar la parte de la base de datos y especificar una actualización de los mismos.

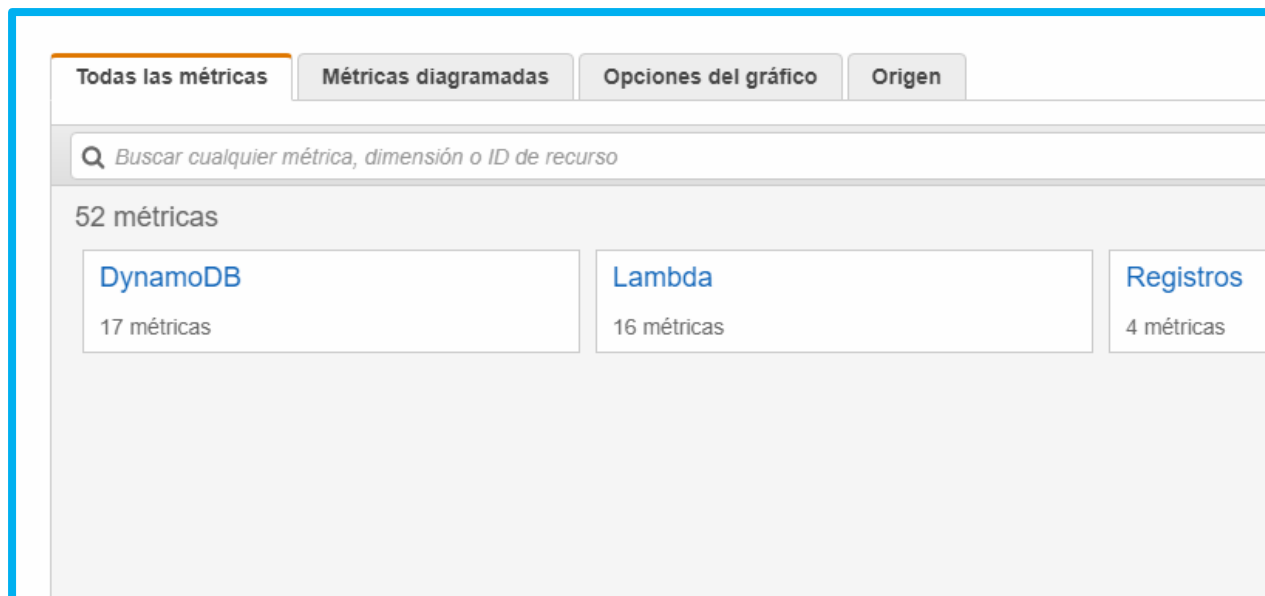


Figura 56 Métricas de la alarma

En la Figura 56 aparecen las métricas seleccionables, siendo la óptima para este caso las de la base de datos, que será la cual se seleccionará para continuar con la alarma. Una vez decidida, hay que especificar una vez más la política que activará la alarma.

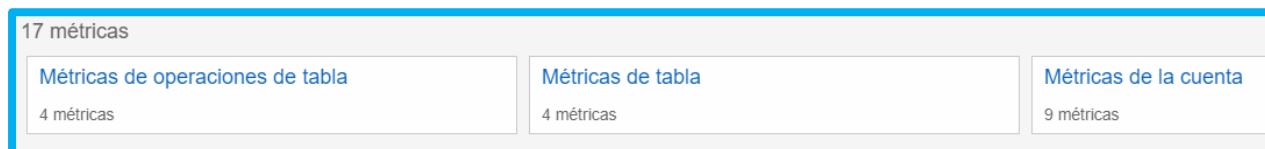


Figura 57 Tipos de Métricas de Base de Datos

Como lo que se desea es que se active al actualizar un dato se seleccionará la opción de “Operaciones de tabla” que es la que incluye las necesidades expuestas.

Todo > DynamoDB > Métricas de operaciones de tabla		Q Buscar cualquier métrica, dimensión o ID
<input type="checkbox"/>	TableName (4)	Operación (Operation)
<input type="checkbox"/>	dynamodb-starter	PutItem
<input type="checkbox"/>	dynamodb-starter	Scan
<input type="checkbox"/>	dynamodb-starter	Scan
<input checked="" type="checkbox"/>	dynamodb-starter	UpdateItem

Figura 58 Métricas de operaciones de tabla

Dentro de “Operaciones de tabla” se encuentran los distintos casos de escaneado o adición de datos, además del de actualización, que es el que viene perfecto para esta funcionalidad.

El siguiente paso será gestionar la parte del propio aviso, como se explicó anteriormente se ha elegido el correo electrónico como medio predeterminado, aunque hay más factores que decidir.

### Notificación

**Activador de estado de alarma**  
Definir el estado de alarma que activará esta acción.

**En modo alarma**  
La métrica o expresión se encuentra fuera del límite definido.

**CORRECTO**  
La métrica o expresión está dentro del límite definido.

**Datos insuficientes**  
La alarma se acaba de iniciar o no hay suficientes datos disponibles.

**Eliminar**

**Seleccione un tema de SNS**  
Defina el tema de SNS (Simple Notification Service) que recibirá la notificación.

**Seleccione un tema de SNS existente**

Crear un tema nuevo

Usar el ARN del tema

**Enviar una notificación a...**

Solo están disponibles las listas de direcciones de correo electrónico de esta cuenta.

**Correo electrónico (puntos de enlace)**  
**pruebastecsos@gmail.com** - [Abrir en la consola de SNS](#)

Figura 59 Configuración Alarma

En la primera parte de la Figura 59 aparece el estado de la alarma que la activará. Esto sirve para recibir las notificaciones, por ejemplo, cuando todo esté correcto, como indica la opción 2, que serviría para recibir siempre un aviso de que todo va correctamente. En el caso de esta funcionalidad es al contrario, se desea recibir un aviso cuando no vaya bien, en lugar de recibir muchos avisos cuando sí vaya bien. Respecto al tema de SNS, es necesario seleccionar uno puesto que se encargará de la parte de las notificaciones.

ID	Punto de enlace	Estado	Protocolo	Tema
26f23e62-dce4-4054-ae25-12329282a806	pruebastecosos@gmail.com	Confirmada	EMAIL	Default_CloudWatch_Alarms_Topic

Figura 60 SNS de correo electrónico

Para que esa opción esté disponible hay que haberlo creado anteriormente, como se indica en la Figura 60. Las características esenciales son el punto de enlace, que debe ser correo electrónico, además de que el estado debe ser siempre “confirmada”, de otro modo no se tendrá permiso de enviar correos electrónicos a la dirección deseada. Para realizar la confirmación hay que hacer unos pequeños típicos pasos de enlace entre la página de AWS y el correo.

Una vez listo todo, al cumplirse también lo anterior, se recibirá un correo electrónico cada vez que las condiciones sean las definidas.



Figura 61 Correo electrónico de alarma

En la Figura 61 se aprecian las primeras líneas del correo estándar que se recibe. Una desventaja de esto es que no tiene mucha capacidad de personalización, se puede añadir alguna oración al correo pero la mayoría de él no se puede cambiar y no aporta mucho a un usuario normal.

## **Conclusión**

Este trabajo es la iniciación de un nuevo camino hacia la mejora de la salud, entre otros muchos aspectos, a través de la tecnología que ofrecen los asistentes de voz. No hay apenas estudios o desarrollos sobre este tema, por lo que aún es pronto para sacar conclusiones seguras o certeras, aunque todo apunta a que su futuro será prometedor y beneficioso. Aún existiendo escasos ejemplos, se ha trabajado y expuesto el conocimiento y herramientas necesarios para cubrir de forma óptima las necesidades especificadas inicialmente. Gracias a las guías sobre los asistentes y la creación y personalización de los mismos junto al demostrador realizado se ha logrado aportar una parte importante a esta novedosa e inexplorada cuestión. A nivel personal creo que es una gran e importante labor el avance producido que sin duda tendrá impacto de una forma u otra para el desarrollo futuro de cualquier persona que lea el proyecto. Cabe destacar que esto no concluye aquí, el proyecto seguirá desarrollándose, ampliando y perfeccionando sus funcionalidades progresivamente.

# Bibliografía

- [1] 2015 UGT, «UGT Brecha Digital,» [En línea]. Available: [http://portal.ugt.org/Brecha\\_Digital/BRECHADIGITAL\\_WEB.pdf](http://portal.ugt.org/Brecha_Digital/BRECHADIGITAL_WEB.pdf).
- [2] 2015 ONU, «Objetivos de desarrollo sostenible,» [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/>.
- [3] 2018 CSIC, «Ministerio de economía y competitividad,» [En línea]. Available: <http://envejecimiento.csic.es/documentos/documentos/pfizer-adherencia-01.pdf>.
- [4] 2017 RiojaSalud, «Gobierno de la Rioja,» [En línea]. Available: <https://www.riojasalud.es/ciudadanos/catalogo-multimedia/farmacia/4880-impacto-de-la-aherencia-en-los-resultados-en-salud-y-en-la-sostenibilidad-del-sistema-sanitario>.
- [5] 2016 elgrupoinformatico, «Acho-elgrupoinformático,» [En línea]. Available: <https://www.elgrupoinformatico.com/noticias/acho-asistente-por-voz-extremeno-tipo-google-assistant-pero-para-mayores-t75782.html>.
- [6] 2019 Amazon, «Skills Salud,» [En línea]. Available: <https://www.amazon.es/s?rh=n%3A14010736031&brr=1&rd=1>.
- [7] 2016 P. ÁGILES, «proyectosagiles/beneficios-scrum,» [En línea]. Available: <https://proyectosagiles.org/beneficios-de-scrum/>.
- [8] 2019 ProfesionalReview. [En línea]. Available: <https://www.profesionalreview.com/2019/09/28/google-assistant-vs-alexa/>.
- [9] 2016 Loupventure, «loupventure/speakeriqtest,» [En línea]. Available: <https://loupventures.com/annual-smart-speaker-iq-test/>.
- [10] 2018 CSIC, «csic Adherencia Médica,» [En línea]. Available: <http://envejecimiento.csic.es/documentos/documentos/pfizer-adherencia-01.pdf>.
- [11] 2019 A. SSML, «SSML Reference,» [En línea]. Available: <https://developer.amazon.com/es-ES/docs/alexa/custom-skills/speech-synthesis-markup-language-ssml-reference.html>.
- [12] 2018 SEFAC, «sefac.org,» [En línea]. Available: [https://www.sefac.org/sites/default/files/sefac2010/private/documentos\\_sefac/documentos/farmaindustria-plan-de-adherencia.pdf](https://www.sefac.org/sites/default/files/sefac2010/private/documentos_sefac/documentos/farmaindustria-plan-de-adherencia.pdf).