

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA MECÁNICA



"DESARROLLO DE UNA SKILL EDUCATIVA SOBRE EL
ASISTENTE VIRTUAL ALEXA:
ESTUDIA CONMIGO"

TRABAJO DE FIN DE GRADO

Febrero -2022

AUTOR: Javier Martín Domínguez

DIRECTOR: María Asunción Vicente Ripoll

ABSTRACT

After the passage of the last decades, we have seen how digitization is a pending issue in almost all sectors. In this final degree project, we will present a solution within the field of education. The purpose of the project is to be able to design an ability or skill in the Alexa virtual assistant development platform. We will seek to develop a skill which serves as a study tool for the user, where they can receive the lesson as many times as they want and be evaluated by Alexa. Getting in this way an interactive, educational and productive skill.

Our project will be divided into three large blocks. In the first block we will begin by presenting virtual assistants and their applications, deeping into the educational sector. Then we will go into a global vision of the Alexa visual assistant, where we will talk about its functionalities and its physical devices.

In the second block, we will explain how Alexa behaves internally and her work environment. Then we will focus on explaining how a skill works and what its structures are like, that is, we will go into its architecture. Finally, we will talk about its development platform and explain how a skill can be created.

In the third and last block, we will present the skill that we want to develop, giving a vision of what need we want to cover thanks to the application of our skill and we will talk about its functionalities. Then we will go into its development explaining its front-end and back-end. Finally, we will test the developed skill and provide possible future work.

ÍNDICE

ABSTRACT	1
ÍNDICE DE FIGURAS.....	3
GLOSARIO	5
CAPÍTULO 1. INTRODUCCIÓN	7
1.1 Resumen	7
1.2 Objetivos del TFG.....	8
1.3 Motivaciones y justificación de asignaturas.....	9
CAPÍTULO 2. ANTECEDENTES	10
2.1 Asistentes virtuales.....	10
2.1.1 Antecedentes y evolución de los asistentes virtuales.....	10
2.1.2 Asistentes virtuales en la actualidad	15
2.1.3 Campos de aplicación de los asistentes virtuales	18
2.1.4 Tecnología y digitalización del sector educativo	20
2.1.4.1 Aplicación de los asistentes virtuales en el sector educativo	21
2.2 Alexa y sus dispositivos físicos.....	22
2.2.1 Aparición y evolución del asistente virtual de Alexa.....	22
2.2.2 Principales funciones	23
2.2.3 Dispositivos físicos	24
CAPÍTULO 3. ARQUITECTURA DEL ASISTENTE VIRTUAL ALEXA	26
3.1 Comportamiento de Alexa	26
3.2 Presentación del entorno de trabajo de Alexa.....	28
3.3 ¿Qué es una skill?	30
3.3.1 Tipos de skills	32
3.3.2 Arquitectura de software	37
3.3.2.1 Front-end.....	38
3.3.2.2 Back-end	41
3.4 Plataforma de desarrollo	42
CAPÍTULO 4. PRESENTACIÓN Y DESARROLLO DE LA SKILL “ESTUDIA CONMIGO”	52
4.1 Exposición del problema	52
4.2 Presentación y funcionamiento de la skill “ESTUDIA CONMIGO”	53
4.3 Desarrollo de la skill.....	57
4.4 Pruebas y evaluación	70
4.5 Trabajos futuros.....	75
4.6 Anexos	76
4.6.1 Front-end	76
4.6.2 Back-end	81
CONCLUSIONES.....	88
BIBLIOGRAFÍA	89
BIBLIOGRAFÍA FIGURAS	93

ÍNDICE DE FIGURAS

Figura 1. Esquema del Test de Turing.....	11
Figura 2. Ilustración del logotipo de Siri	13
Figura 3. Ilustración del logotipo de Google Now	13
Figura 4. Ilustración del logotipo de Alexa	14
Figura 5. Ilustración del logotipo de Cortana	14
Figura 6. Diagrama de las tareas más realizadas por asistentes virtuales	15
Figura 7. Diagrama del uso de los asistentes virtuales.....	16
Figura 8. Gráfica del uso de los asistentes de voz por años	16
Figura 9. Diagrama con los principales desafíos a sortear para el desarrollo del aprendizaje en remoto	21
Figura 10. Tabla de ejemplos de solicitudes del asistente virtual Alexa	23
Figura 11. Tabla de las empresas sincronizadas con el asistente virtual Alexa	24
Figura 12. Esquema de la arquitectura del asistente virtual Alexa	27
Figura 13. Esquema del entorno de trabajo del asistente virtual Alexa	29
Figura 14. Ilustración del acceso a una skill.....	30
Figura 15. Diagrama del proceso de invocación de una skill.....	31
Figura 16. Tabla de comparación de interfaces	32
Figura 17. Tabla de tipos de skills	33
Figura 18. Esquema de funcionamiento de Smart Home Skill	35
Figura 19. Esquema de la interacción con la skill Flash Briefing Skill	36
Figura 20. Estructura de la arquitectura de una skill.....	38
Figura 21. Gráfico de la estructura del front-end.....	39
Figura 22. Ilustración de un tipo de slots	40
Figura 23. Ilustración de la lista con los intents predeterminados	40
Figura 24. Tabla de lenguajes de programación compatibles con AWS Lambda	41
Figura 25. Recorte de la plataforma de desarrollo de Alexa	43
Figura 26. Recorte de la plataforma de desarrollo de Alexa	43

Figura 27. Recorte de la plataforma de desarrollo de Alexa	44
Figura 28. Recorte de la plataforma de desarrollo de Alexa	44
Figura 29. Recorte de la plataforma de desarrollo de Alexa	45
Figura 30. Recorte de la plataforma de desarrollo de Alexa	46
Figura 31. Recorte de la plataforma de desarrollo de Alexa	46
Figura 32. Recorte de la plataforma de desarrollo de Alexa	47
Figura 33. Recorte de la plataforma de desarrollo de Alexa	47
Figura 34. Recorte de la plataforma de desarrollo de Alexa	48
Figura 35. Recorte de la plataforma de desarrollo de Alexa	48
Figura 36. Recorte de la plataforma de desarrollo de Alexa	49
Figura 37. Recorte de la plataforma de desarrollo de Alexa	50
Figura 38. Recorte de la plataforma de desarrollo de Alexa	50
Figura 39. Recorte de la plataforma de desarrollo de Alexa	51
Figura 40. Recorte de la plataforma de desarrollo de Alexa	54
Figura 41. Recorte de la plataforma de desarrollo de Alexa	55
Figura 42. Recorte de la plataforma de desarrollo de Alexa	56
Figura 43. Recorte de la plataforma de desarrollo de Alexa	56
Figura 44. Recorte de la plataforma de desarrollo de Alexa	57
Figura 45. Recorte de la plataforma de desarrollo de Alexa	70
Figura 46. Recorte de la plataforma de desarrollo de Alexa	70
Figura 47. Recorte de la plataforma de desarrollo de Alexa	71
Figura 48. Recorte de la plataforma de desarrollo de Alexa	71
Figura 49. Recorte de la plataforma de desarrollo de Alexa	72
Figura 50. Recorte de la plataforma de desarrollo de Alexa	72
Figura 51. Recorte de la plataforma de desarrollo de Alexa	73
Figura 52. Recorte de la plataforma de desarrollo de Alexa	73
Figura 53. Recorte de la plataforma de desarrollo de Alexa	74

GLOSARIO

A lo largo de este Trabajo de Fin de Grado se utilizarán términos y abreviaciones técnicas que puede ser que el lector desconozca, es por ello que podrán ser consultadas en el siguiente listado.

Amazon: Es una de las 500 mayores empresas de EE.UU. La compañía, con sede en Seattle (Washington), es un líder global en el comercio electrónico.

Back-end: Es el lado del servidor del sitio web. Almacena y organiza datos, y también se asegura de que todo en el lado del cliente del sitio web funcione bien. Es la parte del sitio web que no puedes ver ni interactuar. Es la parte del software que no entra en contacto directo con los usuarios.

Braina: Es un software de IA multifuncional que le permite interactuar con su computadora utilizando un comando de voz con una gran disponibilidad de idiomas.

Chatbot: Es un programa de computadora que simula una conversación humana a través de comandos de voz, chats de texto o ambos.

Cloud: "La nube" hace referencia a los servidores a los que se accede a través de Internet, y al software y bases de datos que se ejecutan en esos servidores. Los servidores de la nube están ubicados en centros de datos por todo el mundo. Con la informática en la nube, no es necesario que los usuarios y las empresas gestionen los servidores físicos ni que ejecuten aplicaciones de software en sus propios ordenadores.

Electrodomésticos inteligentes: Son aquellos que utilizan en sus sistemas procesos de domótica, es decir, la automatización inteligente del electrodoméstico. Eso hace que sea posible la comunicación entre el usuario y el electrodoméstico, gracias a la conexión a internet.

Front-end: Es la parte de un sitio web con la que el usuario interactúa directamente. Incluye todo lo que los usuarios experimentan directamente: colores y estilos de texto, imágenes, gráficos y tablas, botones, colores y menú de navegación.

IA: (Inteligencia Artificial) es la combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano.

Interfaz: Es la conexión física y funcional que se establece entre dos aparatos, dispositivos o sistemas que funcionan independientemente uno del otro. En este sentido, la comunicación entre un ser humano y una computadora se realiza por medio de una interfaz.

JAVA: Es un lenguaje de programación y una plataforma informática lanzado por primera vez por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta potenciar una gran parte del mundo digital actual, proporcionando la plataforma confiable sobre la que se construyen muchos servicios y aplicaciones.

JSON: Es un formato ligero de intercambio de datos, que resulta sencillo de leer y escribir para los programadores y simple de interpretar y generar para las máquinas.

MS-DOS: (Microsoft Disk Operating System) es un sistema operativo de línea de comandos no gráfico derivado de 86-DOS que se creó para computadoras compatibles con IBM.

PowerShell: Es una solución de automatización de tareas multiplataforma compuesta por un shell de línea de comandos, un lenguaje de secuencias de comandos y un marco de gestión de configuración. PowerShell se ejecuta en Windows, Linux y macOS.

Procesamiento del lenguaje natural: Se refiere a la rama de la informática, y más específicamente, la rama de la inteligencia artificial que se ocupa de brindar a los ordenadores la capacidad de comprender textos y palabras habladas de la misma manera que los seres humanos.

Servidor: Los servidores son computadoras de alta potencia creadas para almacenar, procesar y administrar datos, dispositivos y sistemas de red. Los servidores son los motores que impulsan a las organizaciones al proporcionar dispositivos y sistemas de red con los recursos adecuados. Para las empresas, los servidores ofrecen capacidades críticas de escalabilidad, eficiencia y continuidad comercial.

Smartphone: El término smartphone (teléfono inteligente) se refiere a un teléfono móvil multifuncional que incluye todo, desde una cámara y un navegador web hasta una pantalla de alta densidad.

Skill: Las skills son como aplicaciones para Alexa. Con una interfaz interactiva, Alexa brinda a los usuarios una forma de manos libres para interactuar con su habilidad. Los usuarios pueden usar su voz para realizar tareas cotidianas como consultar las noticias escuchar música o jugar a un juego.

Subwoofer: Es un altavoz diseñado para reproducir frecuencias de audio de tono bajo conocidas como graves y subgraves, de frecuencia más baja que las que puede generar un woofer.

CAPÍTULO 1. INTRODUCCIÓN

1.1 Resumen

Tras el paso de las últimas décadas, hemos visto como la digitalización es una asignatura pendiente en casi todos los sectores. En este Trabajo de Fin de Grado, presentaremos una solución dentro del campo de la educación.

El proyecto tiene como finalidad el ser capaces de diseñar una habilidad o skill en la plataforma de desarrollo del asistente virtual Alexa. Buscaremos desarrollar una skill la cual sirva como herramienta de estudio para el usuario, donde pueda recibir la lección tantas veces como quiera y ser evaluado por Alexa. Consiguiendo de esta manera una skill interactiva, educativa y productiva.

Pero ello, nuestro proyecto estará dividido en tres grandes bloques.

- En el primer bloque empezaremos presentando a los asistentes virtuales y sus aplicaciones, profundizando en el sector educativo. Después entraremos a dar una visión global del asistente visual Alexa, donde hablaremos de sus funcionalidades y de sus dispositivos físicos.
- En el segundo bloque, entraremos a explicar cómo es el comportamiento de Alexa a nivel interno y su entorno de trabajo. Después nos centraremos en explicar cómo funciona una skill y como son sus estructuras, es decir, entraremos en su arquitectura. Finalmente, hablaremos de su plataforma de desarrollo y explicaremos cómo se puede crear una skill.
- En el tercer bloque y último, presentaremos la skill que queremos desarrollar, dando una visión de qué necesidad queremos cubrir gracias a la aplicación de nuestra skill y hablaremos de sus funcionalidades. Después entraremos en su desarrollo explicando su *front-end* y *back-end*. Por último, testaremos la skill desarrollada y aportaremos posibles futuros trabajos.

1.2 Objetivos del TFG

En este apartado hablaremos de la finalidad de este Trabajo de Fin de Grado. Para ello hablaremos de los objetivos preestablecidos a la hora de desarrollar este proyecto.

Nuestro objetivo principal en este trabajo será el de conseguir desarrollar una skill que sirva como herramienta de estudio para los usuarios. Para alcanzar este objetivo, estableceremos unos objetivos más elementales los cuales nos ayuden a alcanzar nuestro objetivo principal.

A continuación, hablaremos de los diferentes objetivos.

- Alcanzar un conocimiento general de los asistentes virtuales, consiguiendo que entienda su funcionamiento y sus diferentes campos de aplicación. Este objetivo es de lo más importante ya que me capacitará para visualizar y distinguir posibles campos de aplicación.
- Familiarizarme con el asistente virtual Alexa, consiguiendo entender su funcionamiento interno. Para ello deberé conocer las estructuras utilizadas para el desarrollo de sus skills.
- Alcanzar un nivel básico del lenguaje de programación JavaScript, partiendo de un nivel básico de C++ y Matlab. Este objetivo es muy importante ya que el código de nuestra skill se escribirá en JavaScript.
- Crear una skill con una interfaz simple para que el usuario no necesite un gran grado de conocimiento para utilizarla. También debemos conseguir que la experiencia de uso sea interactiva para mantener la atención del usuario. Y educativa, ya que la finalidad de la skill es ser utilizada como herramienta de estudio.
- Conseguir crear una experiencia totalmente dinámica, para ello deberemos establecer unos tiempos de acceso, tiempos de movimiento dentro de la skill y cierre de la misma totalmente optimizados.

1.3 Motivaciones y justificación de asignaturas

Las motivaciones que me llevan a realizar este proyecto vienen impulsadas por el interés hacia lo desconocido y lo tecnológico. El grado de conocimiento en el campo de los asistentes virtuales que presento es totalmente mínima por lo que este proyecto se presenta como un gran reto.

La motivación que me lleva a elegir el sector educativo como campo de aplicación se basa en mi pensamiento crítico sobre el nivel de digitalización de dicho sector. Siempre me he preguntado cual es papel que tiene que tomar la tecnología en el sector educativo. Preguntándome el por qué si hoy en día ya existen herramientas que complementan las experiencias educativas por qué los usuarios no las utilizan.

A raíz de la pandemia, la necesidad de digitalizar el aprendizaje se ha vuelto una obligación es por ello que los asistentes virtuales jugarán un papel muy importante en la próxima década.

Por otro lado, desarrollaremos la justificación de las asignaturas. La asignatura que más conocimientos y competencias me ha aportado para el desarrollo de este proyecto ha sido Fundamentos de Informática.

En esta asignatura me presentaron y explicaron los fundamentos de la programación los cuales me servirían para poder desplazarme hacia otros lenguajes de programación. A lo largo de la carrera, ha habido diferentes asignaturas que han ido pincelando mi nivel programación como pueden ser las asignaturas de Álgebra y Ecuaciones diferenciales. Por otro lado, el conocimiento más teórico de los asistentes virtuales viene justificada por la asignatura de Teoría de sistemas. Esta asignatura me ha brindado un conocimiento general de su funcionamiento y de su entorno de trabajo.

CAPÍTULO 2. ANTECEDENTES

2.1 Asistentes virtuales

Para poder abordar este estudio primero debemos remontarnos a sus antecedentes, creando de esta manera una base bien estructurada en la que poder desarrollar este proyecto de fin de grado.

Fijaremos el punto de partida en la aparición de los asistentes virtuales los cuales marcaron un antes y un después en el sector tecnológico. Para ello, empezaremos hablando de qué se entiende por asistente virtual.

Un asistente virtual es un agente de software que ayuda a usuarios de sistemas computacionales, automatizando y realizando tareas con la mínima interacción “hombre-máquina”. La interacción que se da entre un asistente virtual y una persona debe ser natural.

El esquema de la interacción queda resumido en: un usuario se comunica usando la voz y el asistente virtual lo procesa, interpreta y responde de la misma manera. Algunos de los asistentes virtuales más conocidos en la actualidad son Braina, Google Assistant, Google Now, Siri de Apple y Alexa de Amazon.

Será en este último asistente virtual en el cual desarrollaremos nuestra skill donde más adelante explicaremos el porqué de la elección de este asistente virtual, para el desarrollo del mismo.

2.1.1 Antecedentes y evolución de los asistentes virtuales

Pero para poder llegar a lo que hoy en día conocemos, en la historia supuso un gran recorrido tecnológico lleno de estudios y proyectos. Estos estudios empujaron a los asistentes virtuales hasta llegar a las máquinas precisas que podemos observar en la actualidad.

El punto de partida de este recorrido tecnológico se le otorga al matemático y padre de la informática teórica Adam Turing. Para ello nos tenemos que remontar a 1950, cuando ni siquiera existían los ordenadores personales y la informática estaba aún dando sus primeros pasos.

Turing se adentró en el estudio de la pregunta de si las máquinas podían pensar, pero lo que propuso una prueba en la que el interrogador determinaba qué jugador era un ser humano y cuál era una máquina, mediante un conjunto de preguntas seleccionadas.

Como todo descubrimiento no estuvo exento de polémica, sin embargo, en la actualidad se sigue utilizando para identificar qué programas informáticos tienen un razonamiento más parecido al de un ser humano.

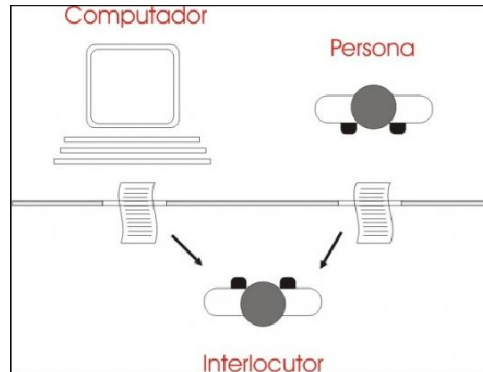


Figura 1. Esquema del Test de Turing. [1]

ELIZA [1964]

El sonado artículo de Alan Turing *Computer Machinery and Computation*, actuó como punto de partida en la creación del primer chatbot por el profesor del MIT Joseph Weizenbaum en la década de 1960. *ELIZA* fue uno de los primeros programas de procesamiento de lenguaje natural diseñado y desarrollado. *ELIZA* fue retada a la prueba de Turing citada anteriormente, consiguiendo un resultado negativo ya que consiguió simular una conversación mediante la técnica de coincidencia de patrones mostrando un aura de comprensión por parte del programa. Sin embargo, no tenía un marco integrado para contextualizar eventos, por lo que el resultado de la prueba fue negativo.

PARRY [1972]

Kenneth Colby fue un psiquiatra estadounidense que trabajó en la teoría y aplicación de la informática y la inteligencia artificial a la psicología, el que llevó a cabo el desarrollo del programa informático *PARRY*. El programa implementó un modelo burdo del comportamiento de una persona con esquizofrenia basado en conceptualizaciones y creencias. Era mucho más avanzado que *ELIZA*, ya que poseía las mismas funciones que *ELIZA* pero proyectaba una actitud propia.

En una encuesta realizada entre sus compañeros psiquiatra, solo el 48% de los encuestados pudo identificar que *PARRY* no era un humano. Pero sin embargo, el prototipo *PARRY* tampoco pudo pasar la prueba de Turing.

RACTER [1983]

RACTER fue un programa de computación desarrollado por William Chamberlain y Thomas Etter. Revelaron su potencial a partir del libro llamado "*The Policeman's Beard is half constructed*" el cual fue desarrollado íntegramente por el programa, consiguiendo el logro de ser la primera inteligencia artificial en escribir un libro.

JABBERWACKY [1988]

Después de la aceptación de los chatbots basados en textos, *JABBERWACKY* chatbot fue un programa capaz de aprender a partir de sonido y otras entradas sensoriales. Fue diseñado y desarrollado por Roller Carpenter en 1988. Tenía como finalidad la de simular una conversación humana natural de una manera interesante y divertida. Carpenter difundía la visión de incorporar este tipo de máquinas en nuestro hogar, basadas en la figura de una mascota con inteligencia artificial que pudiera hacer compañía a los seres humanos. Y siguiendo la tendencia de los modelos anteriores, *JABBERWACKY* chatbot tampoco consiguió pasar la prueba de Turing.

DR.SBAITSO [1992]

Creative Labs desarrolló *DR.SBAITSO* como un programa de síntesis de voz de IA para MS-DOS en 1992. Fue uno de los primeros esfuerzos de integración de IA en un chatbot operado por voz. Se distribuyeron en varias tarjetas de sonido.

El programa conseguía tener una conversación con los usuarios como si fuera un verdadero psicólogo. Las respuestas del modelo eran bastantes simples por la que la interacción no era complicada. Si se enfrentaba a una interacción que no era capaz de entender, el modelo te devolvía una línea parecida a “*jese no es mi problema!*”. Sin embargo, tampoco pudo pasar la prueba de Turing.

ALICE [1995]

ALICE (Artificial Linguistic Internet Computer Entity) es un chatbot de procesamiento de lenguaje universal que utiliza la coincidencia de patrones heurísticos para llevar conversaciones, desarrollado por Richard Wallace en 1995.

El programa consigue recrear la experiencia de chat con una persona real en Internet. El programa funciona en un esquema XML conocido como un lenguaje de marcado de inteligencia artificial (AIML), que ayuda a especificar las reglas de conversación.

SMARTER CHILD [2001]

Sé distribuyó ampliamente a través de SMS y redes de mensajería instantánea. *SMARTER CHILD* atrajo a más de 30 millones de usuarios en AIM, MSN y Yahoo Messenger. Permitió una interfaz de mensajería divertida con acceso rápido a los datos.

SIRI [2010]

El primer asistente virtual digital instalado en un teléfono inteligente fue *SIRI*, que fue introducido como una característica del iPhone 4S el 14 de octubre de 2011. Apple Inc. desarrolló Siri tras su adquisición en 2010 de Siri Inc. un “spin-off” de SRI International, que es un instituto de investigación financiado por DARPA y el Departamento de Defensa de los Estados Unidos.

El software se posiciona a los usos, búsquedas y preferencias del idioma de los usuarios, con un uso continuo. Aparte consigue aprendizaje continuo con la ayuda del aprendizaje automático y la inteligencia artificial.

Actualmente *SIRI* es capaz de hacer llamadas y mandar mensajes, aparte de sugerirte enviar mensajes entendiendo el evento que está aconteciendo.

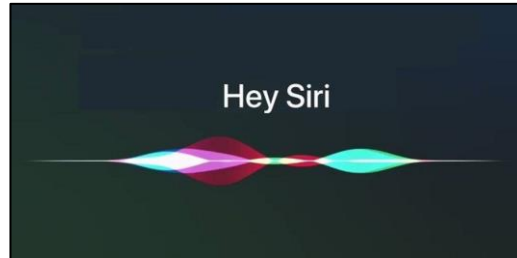


Figura 2. Ilustración del logotipo de Siri. [2]

GOOGLE NOW [2012]

Fue lanzado en 2012 por Google con la finalidad de poder competir contra el asistente virtual de Apple. Como *SIRI*, puede responder consultas, llevar a cabo acciones mediante accesos de web y hacer recomendaciones. Se puede utilizar en smartphones y se ha actualizado para adaptarse a varias funciones. Su función es identificar cosas como las búsquedas más frecuentes en Google, lugares que sueles visitar o citas repetitivas en la aplicación Calendario para facilitarte información antes de buscarla. *GOOGLE NOW* se alimenta de toda la información que encuentra en el smartphone para facilitarte el contenido acorde a tus necesidades y preferencias.

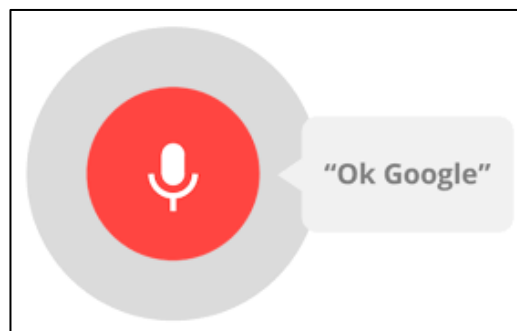


Figura 3. Ilustración del logotipo de Google Now [3]

ALEXA [2014]

En noviembre de 2014, Amazon anunciaba Alexa junto a Echo. Para crear a Alexa se inspiraron en la voz del ordenador y el sistema de conversación a bordo del Starship Enterprise de las películas de Star Trek. El nombre de Alexa se escogió por que el hecho que la “x” es una consonante complicada y de esta manera fácil de reconocer por el asistente virtual. En este asistente virtual no nos pararemos demasiado tiempo, ya que más adelante profundizaremos más sobre sus detalles y funcionalidades.

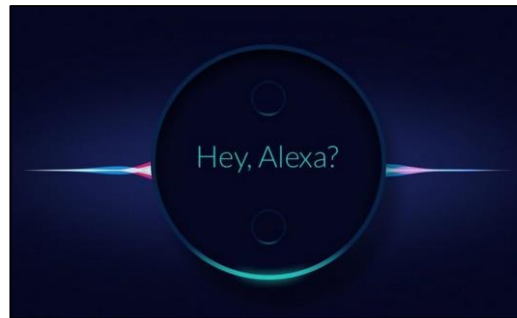


Figura 4. Ilustración del logotipo de Alexa. [4]

CORTANA [2014]

CORTANA es el chatbot de Microsoft implementado en Windows phone y PC en Windows 10. Consigue obtener datos a través de Bing y proporciona servicios basados en la web. Sigue los comandos de voz y texto. Para poder empezar una interacción, el usuario tiene varias opciones: preguntar en el cuadro de búsqueda, seleccionar el micrófono o hablar con *CORTANA*.



Figura 5. Ilustración del logotipo de Cortana. [5]

2.1.2 Asistentes virtuales en la actualidad

En la actualidad, los asistentes virtuales se han ido transformando cada vez más en un asistente más personal. Son capaces de reconocer nuestro tono de voz y grabar nuestras interacciones.

La inteligencia artificial está teniendo un papel crucial en la integración de los asistentes virtuales a nuestro estilo de vida. Empiezan a realizar nuestras tareas personales más repetitivas y a ser una fuente de información de una manera más dinámica y eficaz.

Las tareas más realizadas por asistentes virtuales se pueden observar en el siguiente diagrama.

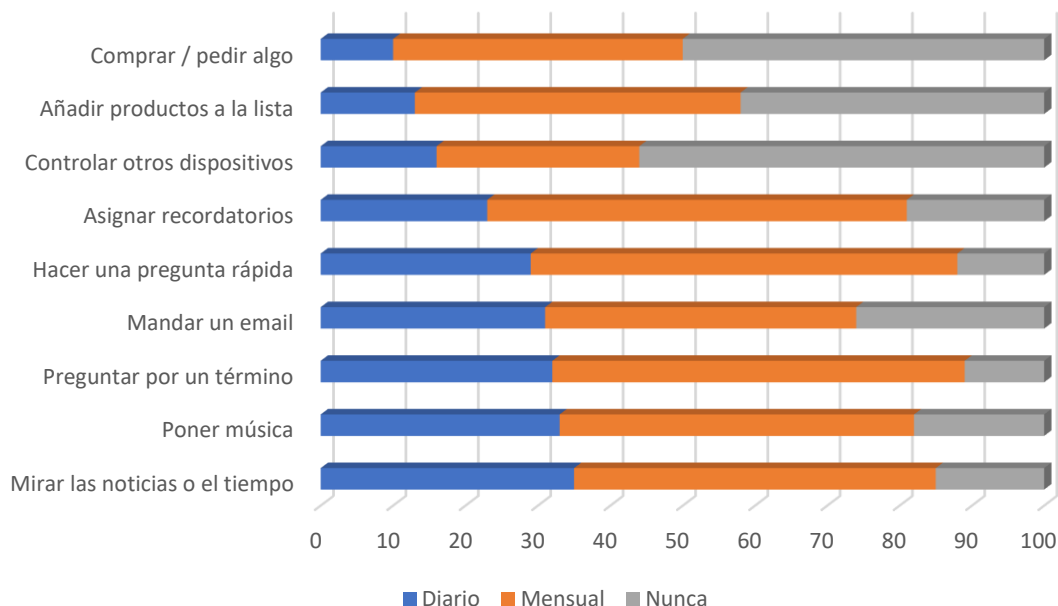


Figura 6. Diagrama de las tareas más realizadas por asistentes virtuales [6]

Todos los datos utilizados en el diagrama de la *Figura 6* fueron recogidos por la consultora PWC. La consultora intentaba brindar una visión de cómo estaban posicionados los asistentes virtuales en el mercado.

Aparte notificaron que el 10% de las personas no estaban familiarizados con los asistentes virtuales por voz y del 90% que, si estaban familiarizados, el 72% han utilizado alguna vez un asistente virtual.

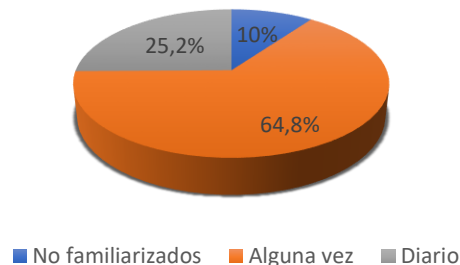


Figura 7. Diagrama del uso de los asistentes virtuales.

Si nos centramos en la línea histórica de la integración de los asistentes virtuales a nuestras vidas podemos observar cómo en 2016 hubo un aumento notario lo que haría despegar el desarrollo e innovación de nuevos asistentes virtuales como se muestra en la *Figura 8*.

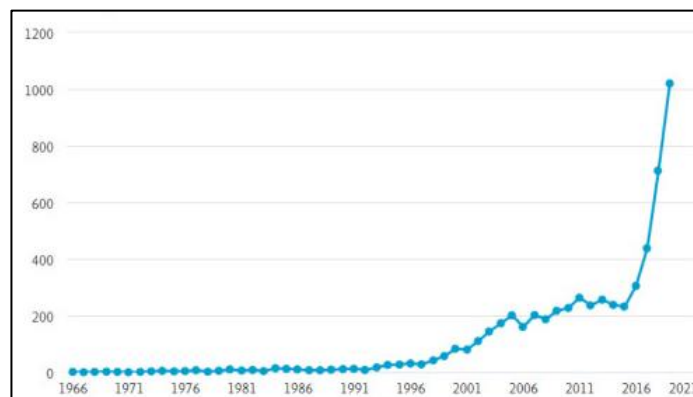


Figura 8. Gráfica del uso de los asistentes de voz por años. [7]

Tenemos que admitir que los asistentes virtuales por voz están situados en la tendencia del futuro ya que pueden llegar a cubrir muchas necesidades requeridas en un futuro. Sin embargo, en la actualidad los asistentes virtuales tienen que hacer frente a diferentes problemas que aún no han sido resueltos por la tecnología.

Los cuatro problemas más notorios son:

Coste - valor: En la actualidad ya existen tecnologías que realizan las mismas funciones que un asistente virtual, por lo que la variable diferenciadora no es tan grande como para que los usuarios quieran gastar su dinero en un asistente virtual. La mayoría de los usuarios que utilizan un asistente virtual lo hacen por deseo y no por necesidad.

Confianza: Como podemos observar en el diagrama anterior, que a la hora de hacer acciones que requieran dinero como por ejemplo hacer una compra, los usuarios aún no confían en los asistentes virtuales para realizar dicha acción. Por lo que tienen que trabajar en seguridad para que los usuarios perciban un entorno más fiable.

Conciencia y complejidad: Las capacidades de los asistentes por voz se están desarrollando vertiginosamente y por lo tanto su complejidad, pero para poder ganar valor los usuarios tienen que ser capaces de entender que capacidades tienen los asistentes virtuales y cómo se tienen que utilizar.

Intimidad: Una de las barreras que tienen que hacer frente los asistentes virtuales es la intimidad del usuario. Cuando el usuario está en público, aun se hace raro en la sociedad actual que un usuario este dialogando con un asistente virtual, aparte de muchas más acciones las cuales requieren intimidad.

La intimidad es una de las barreras por las que muchos usuarios no utilizan los asistentes virtuales.

Una de las noticias más sonoras y que más daño han hecho a todos los dispositivos que utilizan inteligencia artificial ha sido el caso de Mark Zuckerberg, el creador de Facebook. El empresario ha sido juzgado por traficar con los datos personales de sus usuarios sin el permiso de estos.

Es por ello que se ha generado una gran corriente negativa en contra de los dispositivos virtuales. Generando un miedo al consumidor que le aleja indirectamente de la utilización de estos.

En la actualidad coexisten varios asistentes virtuales controlados por voz, entre ellos destacan Alexa, Siri, Cortana y Google Assistant.

El objetivo de esta sección es comparar los diferentes asistentes virtuales enfocándonos en las funciones más utilizadas.

	Alexa	Cortana	Google	Siri
Asignar recordatorios	✓		✓	✓
Responder preguntas rápidas	✓	✓	✓	✓
Poner alarmas	✓		✓	✓
Mirar las noticias o el tiempo	✓	✓	✓	✓
Realizar llamadas	✓	✓	✓	✓
Poner música	✓	✓	✓	✓
Mostrar imágenes	✓		✓	✓
Consultar eventos del calendario	✓	✓	✓	✓
Responder emails				✓
Enviar y leer mensajes	✓	✓	✓	✓
Actividades lúdicas	✓	✓	✓	✓
Consultar términos	✓	✓	✓	✓

Como podemos observar en la tabla, todos los asistentes virtuales cubren casi las mismas funcionalidades por lo que me lleva a preguntarme por qué Alexa. Pero para poder conocer la respuesta a esta pregunta debemos esperarnos al capítulo siguiente, ya que no es un motivo de funcionalidades sino de desarrollo.

2.1.3 Campo de aplicación de los asistentes virtuales

Los asistentes virtuales son denominados tecnologías emergentes, que jugaran un papel importante en un nuestro estilo de vida como usuarios, este impacto se espera que sea positivo y genere bienestar. En la actualidad, los asistentes virtuales pueden aplicarse en distintos campos.

Servicio al cliente: El desarrollo de las nuevas tecnologías ha hecho que las personas interactúen entre sí de manera diferente, al igual que su interacción con las empresas. El comercio electrónico ha evolucionado y ha cambiado por completo la forma en que las empresas venden sus productos, pero existen algunos problemas con la calidad del servicio al cliente. Especialmente en los chats en vivo, el tiempo de espera para que un

empleado comercial responda puede ser largo y las respuestas pueden no ser siempre precisas.

En la actualidad, muchas empresas utilizan chatbots para ayudar a sus clientes. La atención al cliente está disponible las 24 horas del día a través del chatbot, lo que permite a los consumidores resolver su consulta independientemente del horario de atención estándar, lo que mejora la satisfacción del cliente. Se espera que el uso de asistentes virtuales domine pronto el campo de la industria del servicio al cliente.

Salud: En el cuidado de la salud, los asistentes virtuales están diseñados para proporcionar a los pacientes información personalizada sobre salud, productos y servicios relacionados con el paciente, y ofrecer diagnósticos y sugerir tratamientos basados en los síntomas del paciente.

Si nos centramos en las ventajas de usar asistentes virtuales, debemos enfatizar en que ayudan a la toma de decisiones médicas, apoyan la terapia cognitiva que brindan un tratamiento de salud eficiente con una precisión igual a la de los médicos humanos.

Por otro lado, los médicos creen que los chatbots son más efectivos en actividades administrativas como programar citas, encontrando hospitales o entregando recordatorios de recetas.

Sin embargo, los médicos no confían en los asistentes virtuales para remplazar tareas complicadas de toma de decisiones que requieren asesoramiento médico profesional.

Un asistente virtual puede conseguir que el sistema de atención de la salud mental sea más accesible y exitoso, ya que podrían tratar a las personas que se resisten a hablar con un médico porque se sienten incómodas al revelar sus sentimientos o a hablar de su situación actual.

A parte los asistentes virtuales al no ser personas, ni se agotan ni se enferman por lo tanto son más rentable y pueden funcionar de forma más interrumpida durante todo el día, lo que es especialmente útil para las personas que pueden tener problemas de salud más allá del horario laboral de sus médicos.

Una de las organizaciones que se ha lanzado a esta tendencia ha sido *Hera Technology*. Una innovadora *start up* que trata de democratizar el acceso a la salud mental mediante el uso de asistentes virtuales. Esta *start up* no ha pasado por desapercibido y ha conseguido ser financiada por grandes entidades como Pfizer, UDecides, Politécnica de Madrid, entre otras muchas entidades. Este impulso le ha ayudado a asentarse en el sector sanitario como una de las organizaciones con más proyección para el futuro.

Robótica: El área más crítica a la hora de desarrollar un robot físico es la parte de desarrollo de la interfaz del lenguaje natural. Por tanto, en el campo de los robots físicos encontramos abundantes aplicaciones de los asistentes virtuales. Por ejemplo, se está desarrollando una nueva interfaz de lenguaje natural para el robot autónomo llamado KAMRO. El usuario puede realizar preguntas sobre el comportamiento del vehículo. Cuando hablamos sobre robots desarrollados para el sector de la educación podemos hacer hincapié en *CONTI*, un humanoide capaz de narrar historias y reducir la ansiedad de los estudiantes extranjeros los cuales aún no manejan el idioma del país actuando como un tutor físico.

Sector bancario: En el sector bancario, los asistentes virtuales ya son capaces de hablar con los clientes informándoles sobre los saldos de sus cuentas bancarias y facilitando el pago de sus facturas. También son capaces de sugerir formas para ahorrar más y capaces de ayudar a activar tarjetas. Al mismo tiempo, son capaces de recopilar las opiniones de los clientes. Ejemplos de tales asistentes virtuales son *Erika* de Bank of America, *EVA* de HDFC y *Ceba* de Bank of Australia.

Logística y cadena de suministro: Los asistentes virtuales ya son capaces de aceptar y rastrear pedidos, organizar detalles de entrega de los pedidos, hacer reservas y son capaces de contactar con los clientes informándoles sobre ofertas y descuentos, incluso responder las preguntas de los clientes. Por lo que está suponiendo un gran ahorro de personal y eficiencia, ya que estas tareas en la mayoría de las empresas vienen siendo realizadas por personas y con el uso de asistentes virtuales los clientes tienen un flujo de información ilimitado.

Educación: Los asistentes virtuales dentro del sector educativo tienen un potencial determinante por su capacidad comunicativa mediante el procesamiento de lenguajes naturales. Sin embargo, su incorporación educativa debe ir precedida de una reflexión previa que garantice organización funcionalidad y viabilidad dentro de la institución. En este campo de aplicación no nos centraremos más ya que como nuestra skill está enfocada al sector educativo, por lo desarrollaremos más en profundidad este campo en el punto siguiente.

Sin embargo, la utilidad de los asistentes virtuales no solo se limita a las áreas mencionadas. Puede que sea una afirmación un poco soñadora decir que en la mayoría de las ocasiones donde la comunicación se lleva a cabo a través del lenguaje natural es una verdadera oportunidad de aplicación para un asistente virtual.

2.1.4 Tecnología y digitalización del sector educativo

La tecnología y la digitalización del sistema educativo son los principales retos entre los diferentes países y especialmente en España. Muchos centros de enseñanza se ven obligados a trasladar la forma de impartir las clases de un entorno presencial a un entorno de enseñanza remota.

La pandemia ha generado una demanda acelerada de necesidad de digitalizar el sistema educativo para que la enseñanza de los estudiantes no se vea afectada por los obstáculos generados por la pandemia, como la imposibilidad de asistir a los centros educativos cuando el estudiante se ve afectado tanto directamente como indirectamente por el COVID.

Centrándonos en los principales desafíos u obstáculos a sortear por los estudiantes para el desarrollo del aprendizaje a remoto, podemos observar en el diagrama siguiente como el mayor desafío fue la incertidumbre, seguido de la modalidad de las evaluaciones.

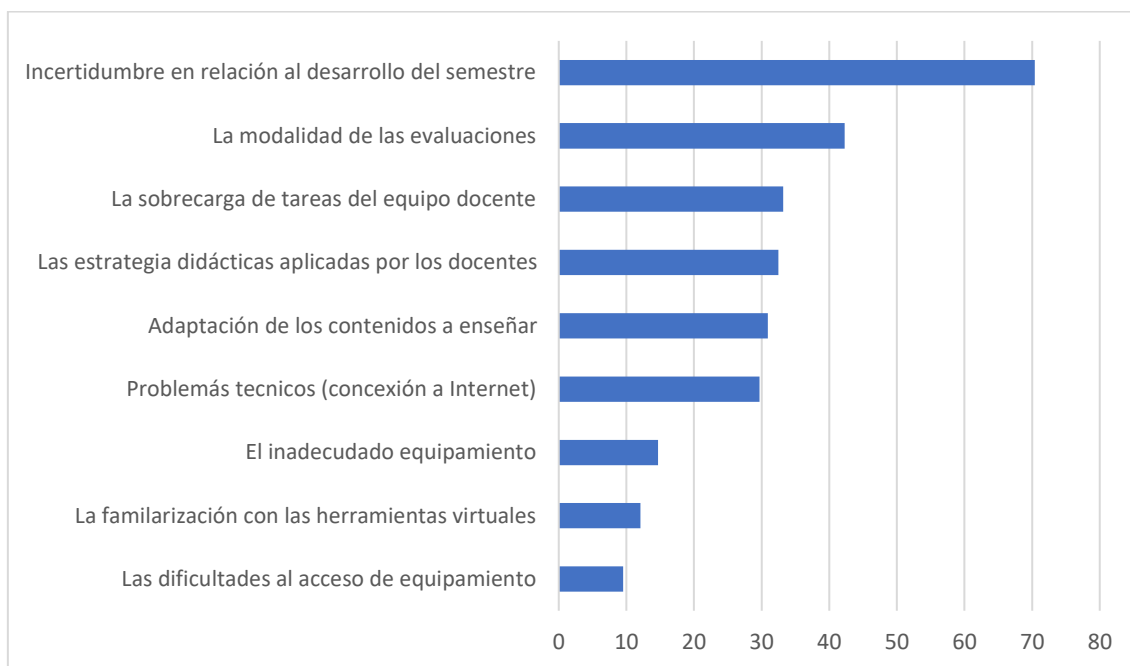


Figura 9. Diagrama con los principales desafíos a sortear para el desarrollo del aprendizaje en remoto. [8]

Los datos que se encuentran en la *Figura 9* fueron recogidos por la Universidad de SciELO en Uruguay durante el primer semestre de 2020. Los datos muestran un sistema educativo totalmente ineficaz a la hora de poder adaptarse al aprendizaje en remoto. Es por ello por lo que, si ya los asistentes virtuales tenían un gran alcance en este sector, con los estragos de la pandemia se ha visto multiplicado dicho potencial.

2.1.4.1 Aplicación de los asistentes virtuales en el sector educativo

La creciente demanda de aprendizaje en remoto ha conducido a una alta competencia entre las instituciones para resolver dicha necesidad. Una de las principales razones viene potenciada por el desmedido número de alumnos que se han visto obligados a recibir las clases en remoto, donde la ayuda que reciben del profesor se ve reducida. Los asistentes virtuales vienen para cubrir dichas necesidades ya que son capaces de brindar contenido educativo y asistencia personal, de manera ilimitada y continuada.

De los roles y del tipo de relación que existen entre los docentes, los asistentes virtuales y los estudiantes, todo se reduce al siguiente esquema. El profesor diseña y configura estos programas con el fin de que le complementen y le ayuden en la realización de tareas sencillas. Por lo tanto, el docente se encarga la tutorización más personalizada e interviene en situaciones de conflicto o de alegaciones de evaluación y dedica el tiempo a cuestiones más creativas y de alto nivel cognitivo.

Los asistentes virtuales actúan como mediadores interactivos que proporcionan atención ininterrumpida, resuelven preguntas frecuentes y guían en tareas mecánicas y repetitivas de bajo nivel cognitivo. Con ellos, el estudiante accede fácilmente a los contenidos.

Para las familias también tienen un gran impacto ya que son una herramienta cómoda y productiva. Los asistentes les ayudan a obtener respuestas fáciles sobre el centro y el plan docente, proporcionando información sobre distintos procesos administrativos.

2.2 Alexa y sus dispositivos físicos.

En este apartado profundizaremos en el asistente virtual Alexa, que como previamente hemos comentado, será en este asistente virtual donde desarrollaremos nuestra skill. Para ello empezaremos hablando sobre su aparición y evolución en el mercado, luego profundizaremos en sus funcionalidades y finalmente hablaremos sobre sus dispositivos físicos.

2.2.1 Aparición y evolución del asistente virtual de Alexa.

Alexa se desarrolló a partir de un procesador llamado *Ivona* que fue desarrollado por A Space Odysee en Polonia en 2001. Posteriormente sería comprado por Amazon. En noviembre de 2014, Amazon anunció Alexa junto con Echo. Alexa se inspiró en la voz de la computadora y el sistema conversacional a bordo del Starship Enterprise en series de televisión y películas de ciencia ficción, comenzando con Star Trek: The Original Series y Star Trek: The Next Generation.

En 2014, cuando Amazon estaba lanzando Echo (primer dispositivo físico lanzado al mercado por Amazon) al mercado, la compañía lo presentó como un altavoz inteligente para controlar la música y poco más. Pero cuando Alexa impulsó a Echo y lo hizo inteligente, el altavoz inteligente evolucionó rápidamente de un práctico dispositivo para reproducir música a un centro de hogar inteligente que multiplicaba en funcionalidades.

La clave de este éxito tan abrumador fue que tanto los fabricantes como los usuarios buscaban un método simple y polivalente que pudiera controlar diferentes dispositivos a la vez. La idea del comando por voz parecía brillante en su simplicidad, ya que uno podía obtener la información necesaria o encender la música sin siquiera moverse. Lo bueno de eso fue la capacidad del altavoz para captar la voz, procesarla y de esta forma suprimir la necesidad de tener que transportar la voz. Fue entonces cuando los fabricantes de electrodomésticos inteligentes aceleraron ya que se podían controlar con Alexa.

Después de este éxito, Google, Microsoft y Apple también comenzaron a impulsar los asistentes virtuales que funcionaran como centros inteligentes. Sin embargo, en ese momento el mercado ya estaba inundado con dispositivos Echo, y los fabricantes de electrodomésticos inteligentes ya estaban sincronizados con él.

La segunda gran idea que diferenció a Alexa de sus competidores nació del deseo de entender a sus clientes. Por eso Amazon decidió abrir su ecosistema a los fabricantes de dispositivos y desarrolladores externos. Como resultado, las habilidades de Alexa se llenaron de nuevo contenido útil y sus capacidades se ampliaron notoriamente. Cuando los asistentes inteligentes comenzaron a despegar, el primer lugar se suponía que estaba

designado para Google, ya que tenía un motor de búsqueda y gran cantidad de información de los usuarios. Pero como pasó un tiempo antes de que la empresa pudiera encontrar un buen uso para sus capacidades, Amazon se permitió obtener una ventaja inicial y ganarse la reputación de ser el mejor asistente a los ojos de los clientes.

2.2.2 Principales funcionalidades

Alexa tiene muchas habilidades las cuales es capaz de ejecutar a la perfección. La aplicación de Alexa te permite personalizar el dispositivo Echo con funciones adaptadas a tus preferencias.

Algunas funcionalidades tienen como requisito sincronizarse con una cuenta existente o una suscripción específica para su uso. Como por ejemplo para utilizar Uber, debes primero haberte creado una cuenta en su plataforma y tenerla activa.

A continuación, podrás observar algunos ejemplos de solicitudes que se le puede pedir a Alexa.

Solicitudes
Pedir que te despierte a una hora concreta
Pedir que te reserve un vuelo en la aplicación de Skyscanner
Preguntarle por las noticias actuales
Preguntarle el tiempo en una ciudad en concreto
Poner música
Preguntarle sobre el tráfico que hay en un trayecto en concreto
Subir el volumen de la música
Preguntarle si lloverá mañana
Pedirle que te lea un libro
Pedirle a Uber que te solicite un viaje
Pedir el último pedido de Just Eat
Pedirle que encienda la cafetera
Pedirle que apague las luces
Pedirle que regule el termostato
Pedir una receta a Jamie Oliver
Preguntarle sobre un término en concret

Figura 10. Tabla de ejemplos de solicitudes del asistente virtual Alexa. [9]

Como hemos podido observar diferentes organizaciones ofrecen una sincronización con el asistente virtual Alexa. A continuación, las empresas más comunes a las cuales se pueden acceder desde un dispositivo de Alexa.

Empresa	Habilidad
Just Eat	Pedir comida a domicilio
Met Office	Proporcionar información climatológica
Uber	Pedir un transporte a la localización que desees
National Rail	Proporcionar información de horarios y trenes disponibles
The guardian	Leer noticias
Jamie Oliver	Pedir una receta de comida
Fitbit	Solicita información relacionada con tú salud
Skyscanner	Reservar vuelos
Easyjet	Reservar vuelos
Spotify	Escuchar música.
Philips Hue	Controlar las luces
Neato	Controlar un robot aspirador
BMW	Notificar los niveles de gasolina y batería de su coche
Laundrapp	Encargar para que te limpien prendas concretas
Nido	Controlar el termostato

Figura 11. Tabla de las empresas sincronizadas con el asistente virtual Alexa. [9]

Como podemos observar el control del hogar y de las funciones diarias se han transformado en los campos de actuación de Alexa. Ya que no solo se trata de consultar el tiempo o de poner música, sino de controlar otros dispositivos incluso fuera del hogar.

2.2.3 Dispositivos físicos

En la actualidad, Amazon presenta una gran variedad de dispositivos físicos. Los altavoces Echo, se adaptan a una amplia gama de precios y usos. Los asistentes virtuales varían desde los altavoces más compactos, hasta los más grandes para una experiencia musical más específica. Aparten también cubren los dispositivos con pantallas.

A continuación, hablaremos de los diferentes asistentes citando sus mejores funcionalidades.

Amazon Echo Dot (3ª generación, 2018): El Echo Dot ha sido increíblemente famoso entre los usuarios. Es un excelente dispositivo independiente de Alexa, bien equipado y aunque no es un excelente dispositivo de música, pero te proporcionará unas buenas prestaciones.

Amazon Echo Plus (2ª generación, 2018): Ofrece las mismas funcionalidades de Alexa que todos los demás dispositivos, pero tiene un controlador Zigbee integrado, lo que significa que puede conectarse directamente y controlar dispositivos inteligentes compatibles.

Amazon Echo Studio (2019): Este modelo es el que mejor prestación de sonido consigue, gracias a que es el modelo más grande. Es compatible con el servicio Amazon Music HD, y con otras muchas plataformas de música. Aparte es capaz de manejar el audio de la televisión y es compatible con conexiones inalámbricas a dispositivos Fire TV.

Amazon Echo Flex: Este es el modelo más pequeño con el que puedes tener Alexa en tu casa. Tiene un pequeño altavoz y micrófono para brindar todas las funcionalidades de Alexa. También tiene una conexión USB que se puede usar para cargar el móvil. Es un pequeño dispositivo versátil y fácil de agregar a cualquier habitación.

Amazon Echo Sub: Este modelo mejora su configuración de Echo al agregar un *subwoofer* dedicado. Se puede emparejar con un Echo, pero es ideal para un conjunto de Echos emparejados en estéreo.

Amazon Echo Dot (4ª generación, 2020): Este modelo cambia radicalmente la forma, y se pasa a una forma esférica, sin embargo es un cambio totalmente estético ya que el altavoz no mejora y sus prestaciones se mantienen.

Amazon Echo (4ª generación, 2020): El último lanzamiento de Amazon, también cambia la forma a una esfera y deja de lado el diseño cilíndrico. La nueva forma mejora las credenciales de audio consiguiendo una gran mejora comparado con los modelos anteriores.

A continuación, tocaría hablar de los dispositivos que contienen pantalla física, sin embargo, no nos adentraremos ya que la skill que vamos a desarrollar no necesita de esas prestaciones. Pero aun así debemos enfatizar que Amazon presenta una gran variedad de dispositivos con pantalla con altas prestaciones.

CAPÍTULO 3. ARQUITECTURA DEL ASISTENTE VIRTUAL ALEXA

En este capítulo, nos centraremos en desarrollar la estructura interna de Alexa. Empezando por su comportamiento y terminando, explicando cómo se puede crear una skill desde su plataforma de desarrollo. Este capítulo nos servirá para entender el porqué de desarrollar mi habilidad en Alexa y entender las facilidades que nos da su plataforma de desarrollo. Aparte nos aportará información vital para desarrollar nuestra skill de una manera satisfactoria.

3.1 Comportamiento de Alexa

Para poder adentrarnos en los entresijos de Alexa, primero debemos de entender cómo se comporta Alexa. Escenifiquemos que el cliente quiere hacer una consulta simple, en este caso el usuario pregunta, “¿Alexa, qué hora es?”. Para entender que partes interactúan en esta consulta, iremos explicando cada punto y lo podremos ir visualizando en el esquema que aparece en la siguiente página.

- I. Conversión del audio a un flujo de audio digital: El usuario empieza la acción preguntándole a Alexa, “¿Alexa, qué hora es?”. Alexa capta el audio gracias a su micrófono integrado, una vez captado lo transforma en un flujo de audio digital y lo envía al soporte en la nube de procesamiento de flujos de audio.
- II. El flujo de audio se convierte en texto: Una vez llega al procesador de audio, el audio es transformado en texto.
- III. Activación de la skill: Una vez interpretado el audio, Alexa sincroniza la consulta con la skill que quiere activar el usuario.
- IV. Ejecución de la skill: La skill procesa el texto utilizando su interfaz y apoyándose en Lambda, el cual es el servicio de la skill. La habilidad elabora una respuesta teniendo en cuenta los *intents* interpretados. La skill calcularía la hora qué es donde la respuesta sería de tipo textual, *la una en punto*.
- V. La respuesta textual se convierte en flujo de audio digital: En este punto invertimos el proceso anterior, por lo que la respuesta de carácter textual es transformada a flujo de audio.
- VI. El flujo de audio digital es transformado en audio: Finalmente Alexa le responde al usuario, *la una en punto*. Para transmitir la respuesta Alexa hace uso de los altavoces integrados del dispositivo físico.

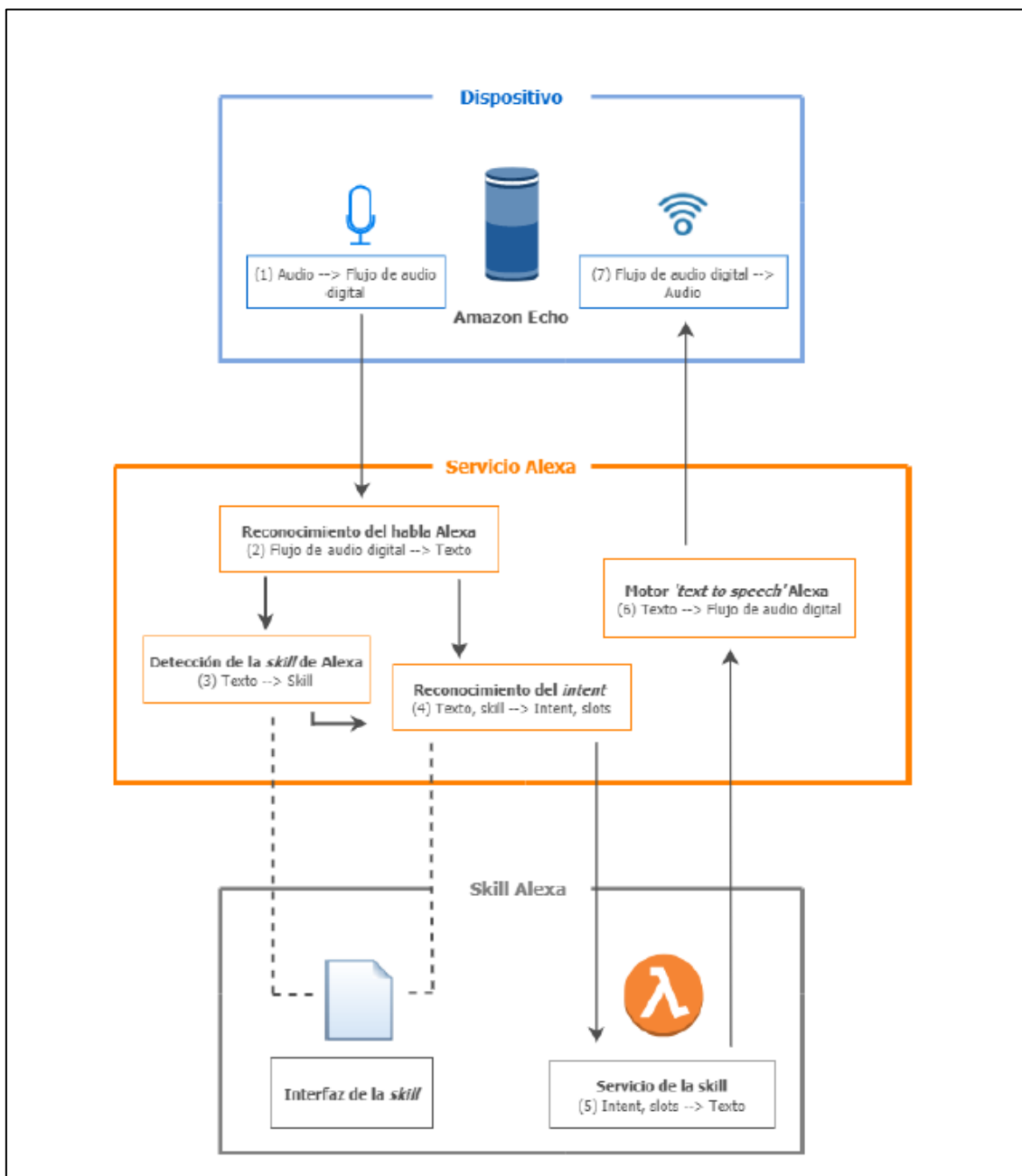


Figura 12. Esquema de la arquitectura del asistente virtual Alexa.

3.2 Presentación del entorno de trabajo de Alexa

Una vez entendido el comportamiento de Alexa, debemos de incidir en su entorno de trabajo. Para comprender este concepto, debemos imaginarnos un matriz perfecta donde en el medio podemos encontrar el servicio de Alexa, capaz de interpretar audio y sincronizar consultas.

Todas las operaciones tienen lugar en el *cloud* es decir en la nube, por lo que no necesita dispositivos grandes para procesar grandes consultas ya que se procesan en servidores externos de Amazon.

Las principales partes del servicio de Alexa son:

- Automatic Speech Recognition (ASR): Es una tecnología que convierte las palabras habladas en texto. Gracias a ASR, la tecnología de voz puede detectar sonidos hablados y reconocerlos como palabras. ASR es la piedra angular de toda la experiencia de voz, lo que permite que los ordenadores finalmente nos entiendan a través de nuestra forma natural de comunicación: el habla.
- Natural Language Understanding (NLU): Con la comprensión del lenguaje del natural, los ordenadores pueden deducir lo que realmente quiere decir un usuario y no solo las palabras que dice. Las tecnologías de voz en la actualidad se construyen con NLU, que es inteligencia artificial centrada en reconocer patrones y significado dentro del lenguaje humano.
- Skill Alexa Management API (SMAPI): Se encarga de coordinar las skills con las intenciones de los usuarios.
- Text to Speech: Se encarga de transformar el texto generado por la skill en sonido.

Todas estas partes conforman el servicio de Alexa, es decir el corazón del entorno de trabajo.

Sin embargo, la plataforma de trabajo de Alexa necesita estar sincronizado con las skills y los dispositivos físicos para poder funcionar ya que todas las partes conforman dicha matriz. Esta está compuesta por la interfaz, el servicio de Alexa y los dispositivos físicos como se muestra en la *Figura 13*.

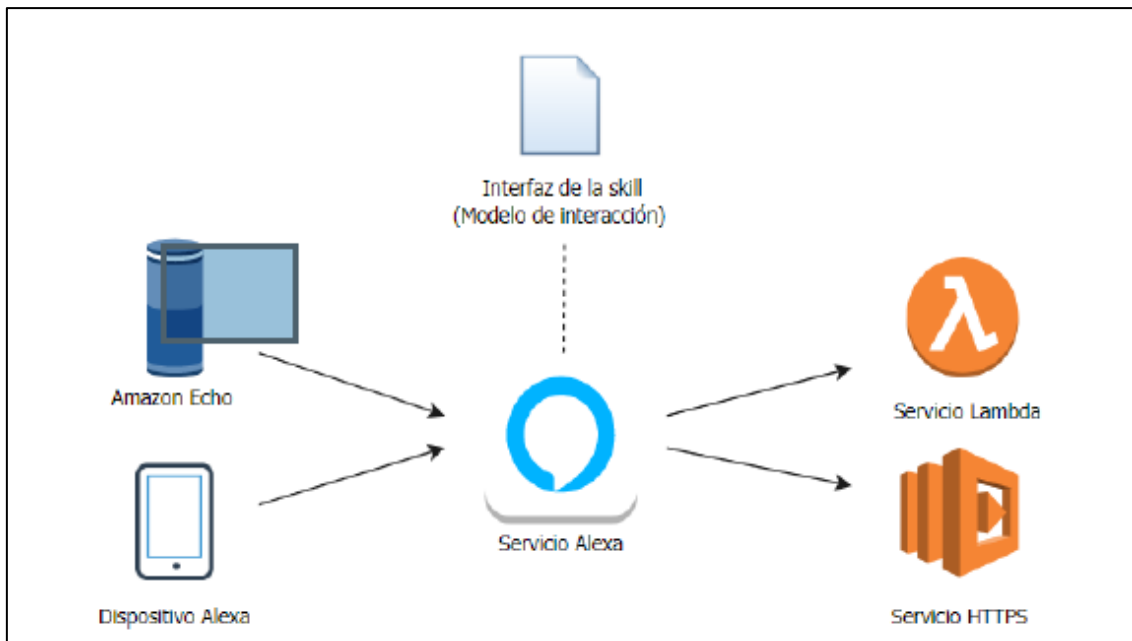


Figura 13. Esquema del entorno de trabajo del asistente virtual Alexa.

(1) La interfaz de usuario de voz determina que es lo que tu skill tiene que comprender cuando el usuario hable. Es un código que permite que el usuario y la skill se conecten entre sí.

(2) El servicio Alexa se encarga de actuar como coordinador entre las consultas. El entorno de Alexa lo reclama cuando se origina una similitud de intención en el lenguaje del usuario. El servicio se direcciona a un servicio al AWS Lambda o en un servidor HTTPS.

(3) Los dispositivos físicos los cuales son los citados en el capítulo anterior como los altavoces Echo. A diferencia de la arquitectura que nos podemos encontrar en la mayoría de los ordenadores, en los dispositivos Alexa no corre ningún tipo de código, ya que todas las operaciones quedan externalizadas en el *cloud* como bien hemos explicado antes. Eso quiere decir que la función de los dispositivos queda relegada a captar y transmitir sonido.

Una vez explicado el entorno de trabajo de Alexa nos adentraremos primeramente en lo que entendemos por skill y como es su arquitectura por dentro. Y después nos adentraremos en cómo desarrollar una.

3.2 ¿Qué es una skill?

Las Skills de Alexa son programas que pueden activarse en línea y que sirven para ampliar la gama de funciones del servicio de habla de Alexa. Con una interfaz interactiva, Alexa brinda a los usuarios una forma de manos libres para interactuar con su habilidad. Los usuarios pueden usar su voz para realizar tareas cotidianas como consultar las noticias, escuchar música o jugar a un juego.

Los usuarios también pueden usar su voz para controlar los dispositivos conectados en la nube. Por ejemplo, los usuarios pueden pedirle a Alexa que encienda las luces o cambie el termostato mediante una skill. La plataforma de desarrollo de software donde se diseñan las skill es conocido como *Alexa Skills Kit (ASK)*.

Dentro de una skill podemos encontrar diferentes etapas es por ello que empezaremos hablando de cómo se accede al contenido de una skill.

Acceso al contenido de una skill:

Un usuario accede al contenido de una skill pidiéndole a Alexa que invoque la skill. Cuando hablamos de invocar nos referimos al acto de comenzar una interacción con una skill particular de Alexa.



Figura 14. Ilustración del acceso a una skill.

Como podemos observar en la imagen anterior, primero debes de decir la palabra de activación que suele ser *Alexa* pero es configurable, le puedes poner *Echo* o *Amazon*. Luego viene la palabra de lanzamiento donde le decimos *abre* y en este caso podemos utilizar muchas posibilidades como por ejemplo: *inicia*, *empieza*, *comienza*, *lanza*, *pídele a*, *pregunta*. Luego viene el *invocation name*, es como los usuarios van a llamar a la skill, es decir, es el nombre de la skill. En esta parte es muy importante que el desarrollador decida cuidadosamente cuál va a ser el nombre de su habilidad.

El usuario puede transmitir más información además del *invocation name* como un dato o una frase específica llamado *utterance*.

Debes tener en cuenta que el usuario nos está transmitiendo una acción concreta que quiere realizar por eso lo denominamos intención o *intent*.

Una vez hemos determinado el *invocation name*, Alexa se comunica con la skill mediante un mecanismo de solicitud y respuesta a través de la interfaz HTTPS. Cuando un usuario invoca una habilidad recibe una solicitud que contiene un JSON. El JSON contiene los parámetros necesarios para que su habilidad comprenda la solicitud, realice su lógica y luego genere una respuesta.

El siguiente diagrama muestra el flujo de procesamiento activado por voz para invocar una skill con el servicio de Alexa.



Figura 15. Diagrama del proceso de invocación de una skill [10]

Interacción entre el usuario y la skill:

Cada habilidad de Alexa tiene un modelo de interacción de voz que define las palabras y frases que los usuarios pueden decir para que la habilidad haga lo que quieren. Este modelo determina cómo los usuarios se comunican y controlan su habilidad. Una interfaz de usuario de voz es similar a una interfaz gráfica de usuario en una aplicación tradicional.

En lugar de hacer clic en los botones y seleccionar opciones de los cuadros de diálogo, los usuarios realizan sus solicitudes y responden a las preguntas por voz. A menudo, la interacción de voz tiene una duración mucho más corta que la interacción con una aplicación.

Cuando un usuario hace preguntas y solicitudes, Alexa emplea el modelo de interacción para interpretar y traducir las palabras en una solicitud específica para la habilidad identificada.

La siguiente tabla compara una skill de interfaz de usuario de voz con una aplicación de interfaz de usuario gráfica para hacer reservas de vuelos.

Acción	Interfaz de usuario de voz	Interfaz de usuario gráfica
Haz una solicitud	El usuario dice: <i>"Alexa quiero volar a Madrid"</i> .	El usuario hace clic en la aplicación y luego selecciona los aeropuertos de origen y destino. El usuario se desplaza por la lista de aeropuertos para encontrar Alicante y luego se desplaza para encontrar Madrid.
Recopilar más información del usuario	Alexa responde: <i>"¿Cuándo te gustaría viajar?"</i> y luego espera una respuesta.	La aplicación muestra un calendario y luego espera a que el usuario seleccione una fecha.
Proporcione la información necesaria	El usuario responde: <i>"El uno de febrero"</i> . La habilidad hace la reserva y espera la confirmación.	El usuario abre el calendario, selecciona el 1 de febrero y luego elige Aceptar. El usuario hace clic en un botón para completar la solicitud y luego espera la confirmación.
Solicitud completa	Alexa responde: <i>"Tu reserva de Alicante a Madrid el lunes 1 de febrero está lista"</i> .	La aplicación muestra el resultado de la solicitud. El usuario cierra la aplicación.

Figura 16. Tabla de comparación de interfaces.

Alexa admite dos tipos de modelos de interacción de voz:

- **Modelo de interacción de voz preconstruido:** En este modelo, ASK define el conjunto de palabras que los usuarios dicen para invocar una habilidad. Por ejemplo, un usuario puede decir: *"Alexa, enciende la luz"* o *"Alexa, apaga la televisión"*.
- **Modelo de interacción de voz personalizado:** El modelo personalizado brinda la mayor flexibilidad, pero es el más complejo. Tú diseñas toda la interacción de voz. Con el modelo personalizado, normalmente debes definir todas las formas en que un usuario puede comunicar la misma solicitud a su habilidad. Por ejemplo, *"Alexa, planifica un viaje de Alicante a Madrid"*, *"Alexa, quiero ir de viaje a Madrid desde Alicante"* y *"Alexa, planifica un viaje a Madrid"*.

3.2.1 Tipos de skills

ASK clasifica las habilidades en tipos según su modelo de interacción de voz y sus capacidades o características.

La siguiente tabla muestra los tipos de skills existentes en orden alfabético.

Tipo de skill	Descripción
Automotive	Desarrollar skills adaptadas al entorno del automóvil.
Business	Desarrollar skills que brinden a los usuarios acceso de voz a sus reuniones y calendarios comerciales, o permitirles buscar y reservar salas de reuniones.
Cooking	Desarrollar habilidades para los aparatos de cocina.
Custom	Desarrollar skills con un modelo de interacción de voz personalizado.
Entertainment Device	Desarrollar skills que permitan a los usuarios controlar el entretenimiento doméstico inteligente y los dispositivos audiovisuales (AV), como televisores y altavoces inteligentes.
Flash Briefing	Desarrollar skills de Flash Briefing para proporcionar a los clientes de Alexa titulares de noticias y otros contenidos breves.
Games	Desarrollar skills de juego impulsadas por interacciones de voz.
Knowledge	Desarrollar skills que permitan a los usuarios hacer preguntas a Alexa sobre los datos de la hoja de cálculo de una organización sin invocar un nombre de habilidad.
List	Desarrollar skills que lean y actualicen las listas de Alexa de un usuario.
Music, Radio, and Podcast	Desarrollar skills que permitan a los usuarios controlar el contenido de audio transmitido a través de dispositivos habilitados para Alexa.
Networking and Wi-Fi	Desarrollar skills para modelar una red Wi-Fi doméstica y los dispositivos conectados a ella.
Smart Home	Desarrollar skills que permitan a los usuarios controlar sus dispositivos domésticos inteligentes.
Smart Home Security	Desarrollar skills que permitan a los usuarios controlar dispositivos de seguridad.

Figura 17. Tabla de tipos de skills.

A continuación, hablaremos de los tipos de skills más utilizadas en la actualidad ya que tienen mucho impacto en nuestro estilo de vida.

Smart Home Skill:

Este tipo de skill es de las más conocidas entre los consumidores, ya que tiene un *marketing* muy potente al ser capaz de controlar tu propia casa de una manera simple y eficaz. Para poder hacer un uso correcto, primero debemos entender los pasos a seguir.

(1) Primero el usuario debe configurar una Smart Home Skill en su cuenta de Alexa, sincronizando de esta manera todos sus dispositivos.

(2) Una vez realizado el paso anterior, el cliente ya puede argumentar su primera solicitud. El cliente dice, *“Alexa, reduce la luz del salón al 50 por ciento”*.

(3) Alexa interpreta el final del mensaje y la conexión con la skill a la que el usuario está haciendo referencia. Alexa consigue transformar este contenido para enviar este informe llamado directiva. La directiva está compuesta por el contenido de autenticación del usuario.

(4) Alexa envía la información al *cloud* donde se encuentra Smart Home Skill, que es la skill que se responsabiliza de la correcta coordinación de los diferentes dispositivos sincronizados.

(5) Smart Home Skill introduce el contenido de la directiva en su código y lo ejecuta. De esta manera genera una respuesta que va enviada al dispositivo físico correspondiente con toda la información necesaria. La skill también genera otro mensaje, esta vez a Alexa llamado evento, en el cual le comunica que la acción ha sido exitosa.

Para realizar esta acción tiene dos opciones o transmitir el evento de forma síncrona desde la función Lambda o de forma asíncrona desde la nube del dispositivo.

(6) Cuando Alexa recibe el evento, Alexa le comunica al usuario que la acción ha sido exitosa. Este mensaje podría ser por ejemplo, *Ok*.

En la ilustración siguiente podréis observar el esquema de funcionamiento para complementar la información.

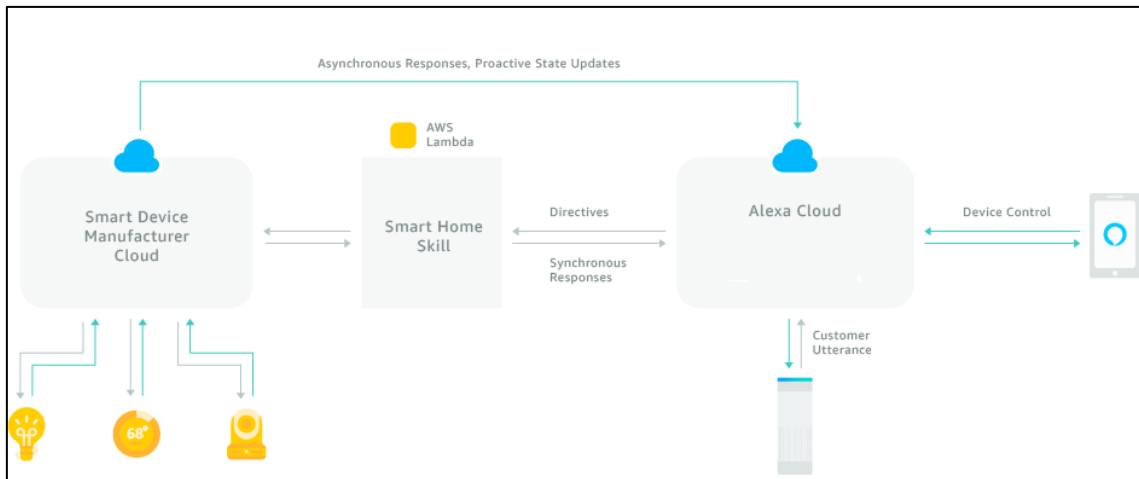


Figura 18. Esquema de funcionamiento de Smart Home Skill.

En resumen, para poder crear habilidades para Smart Home Skill se pueden diferenciar dos grandes grupos. El grupo más mayoritario está conformado por trabajadores que pertenecen a empresas dedicadas a la fabricación de electrodomésticos que buscan incorporarse o asentarse en la vertiente de domótica para así atraer a más consumidores.

Al contrario que el segundo grupo que está formado por desarrolladores totalmente anexos a empresas que buscan sincronizar sus dispositivos de una manera más eficaz y eficiente.

Flash Briefing Skill:

Este tipo de skill transmite una visión general de lo que está pasando en la actualidad al usuario. Para ello se nutre de titulares y noticias, y así consigue transmitirle al usuario un contenido general pero comprimido.

Este tipo de skills se pueden configurar basándose en nuestros gustos y necesidades ya que para que sea útil deber ser una experiencia muy personalizada para el usuario. Es por ello por lo que la skill necesita estar nutriéndose de información constantemente para que así el contenido sea lo más actual posible. A menudo también se pueden añadir juegos de palabras al final del mensaje, para que sea algo más dinámico.

A continuación, podéis observar cómo sería la interacción con esta skill.

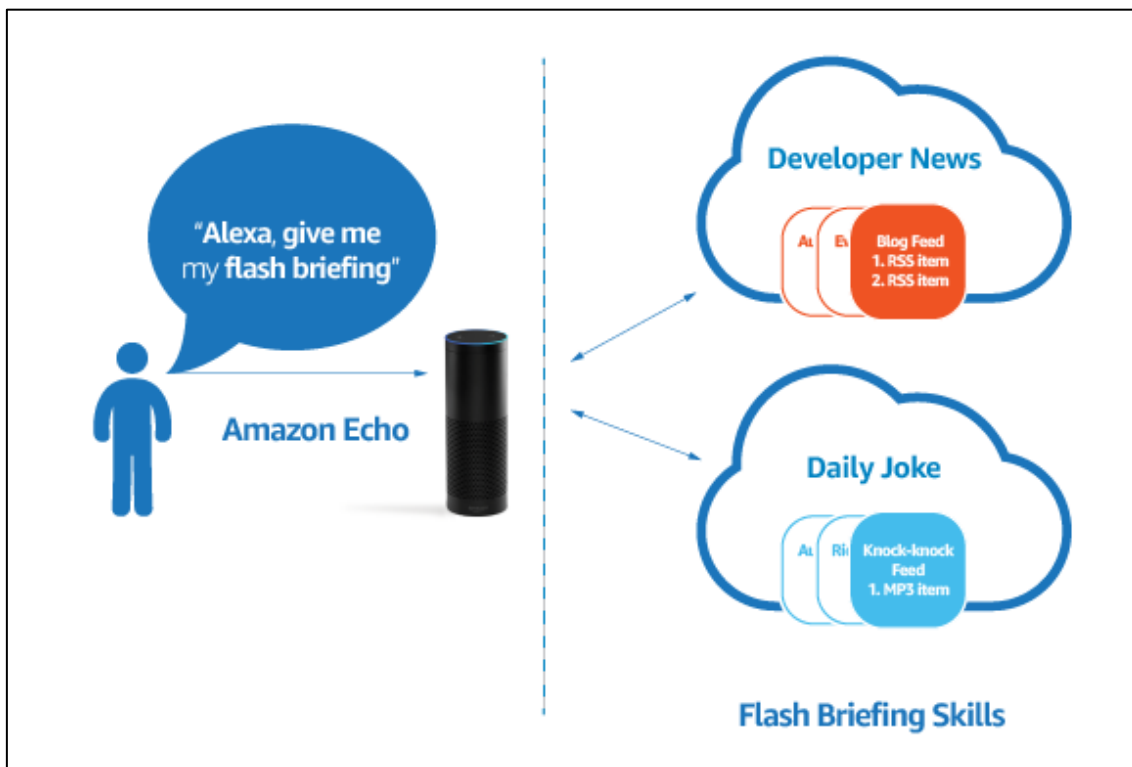


Figura 19. Esquema de la interacción con la skill Flash Briefing Skill.

Este tipo de habilidades suelen ser muy utilizadas por los usuarios en su día a día, ya que informarnos de lo que pasa a nuestro alrededor suele ser una de las primeras acciones que hacemos al empezar el día.

Custom Skill:

No es casualidad que hayamos dejado para el final este tipo de skill, ya que nuestra skill pertenecerá a este grupo. Para poder entender y posteriormente diseñar una skill debemos saber cuáles son sus componentes. En este apartado, presentaremos los diferentes componentes de una manera más tímida ya que el siguiente punto profundizaremos en cada una de las partes para así tener un conocimiento preciso para desarrollar nuestra skill.

- **Intents:** representan acciones que los usuarios pueden realizar con su habilidad. Estas intenciones representan la funcionalidad principal de la skill.
- **Sample utterances:** especifican las palabras y frases que los usuarios pueden decir para invocar esas intenciones. Mapean estas expresiones a sus intentos. Este mapeo forma el modelo de interacción para la skill.

- **Invocation name:** identifica la skill. El usuario incluye este nombre cuando inicia una conversación con su skill.
- **Imágenes, sonidos, archivos:** estos deben almacenarse en un sitio de acceso público para que cada elemento sea accesible mediante un directorio, en nuestro caso no utilizaremos ningún tipo de archivo.
- **Alexa Service:** acepta estas intenciones como solicitudes estructuradas y luego actúa sobre ellas. Este servicio debe ser accesible a través de Internet.
- **Configuración:** reúne todo lo anterior para que Alexa pueda enrutar solicitudes al servicio para su habilidad. Esta configuración se crea en la consola del desarrollador.

Una vez explicado este punto podemos decir que ya tenemos un conocimiento básico de lo que es una skill, su funcionamiento y sus diferentes tipos. Es por ello por lo que en el siguiente punto entraremos en el conocimiento más técnico, es decir, en su arquitectura.

3.3.2 Arquitectura de software.

La arquitectura de software de una skill de Alexa sigue la misma estructura de cualquier otro programa de programación excepto con algunas peculiaridades. La estructura esta dividida en dos grandes partes.

Por un lado, podemos encontrar el front-end que es la parte que se encarga de interactuar con el usuario. También conocida como la interfaz de la skill, la interfaz de usuario de voz determina que es lo que tu skill tiene que comprender cuando el usuario hable. En esta parte podremos encontrar estructuras como los *intents*, los *utterances* y los *slot types*.

Por otro lado, se encuentra el back-end que es la parte que se encarga de toda la lógica de la skill, también conocida como el servicio de la skill. Por ejemplo imaginémonos que la skill necesita nutrirse de datos a partir de una base de datos y devolver información al usuario tendrá que haber un *back-end*. La forma más común para desarrollar el *back-end* es mediante Lambda. Lambda es una tecnología que es capaz de correr código bajo demanda y no necesita nutrirse de ningún servidor.

A continuación, podremos observar el esquema de la arquitectura de software de una skill antes de profundizar en sus componentes.

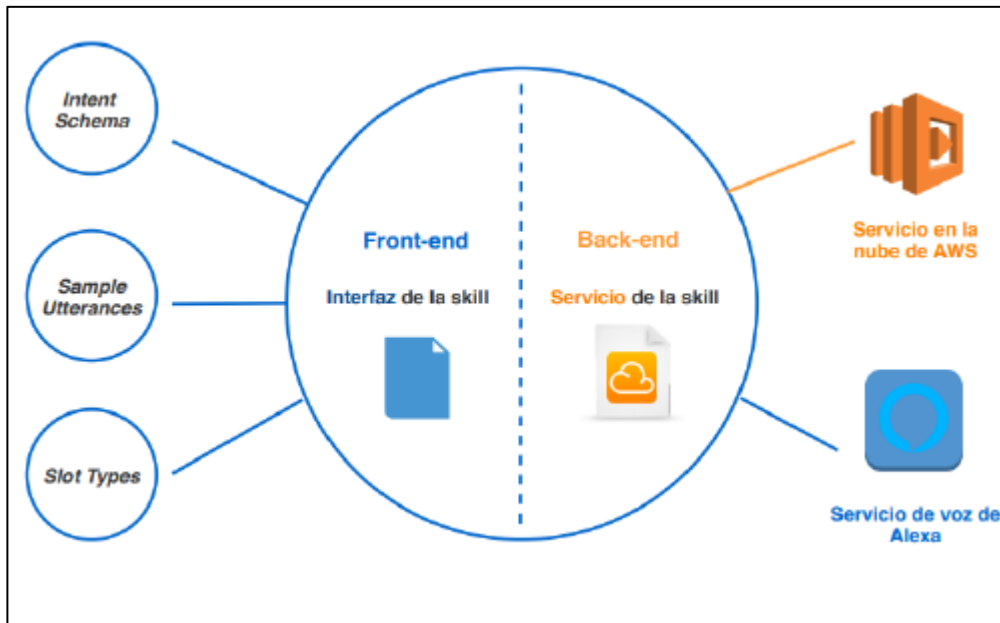


Figura 20. Estructura de la arquitectura de una skill.

3.3.2.1 Front-end

Para crear un modelo de interacción personalizado, debemos definir las siguientes estructuras: los *intents*, las *utterances* y los *slots*.

Un *intent* representa una acción que cumple con la solicitud hablada de un usuario. Los *intents* pueden tener opcionalmente argumentos llamados *slots*. Las *utterances* en cambio son un conjunto de posibles frases habladas asignadas a *intents*.

Cuando creamos skills tenemos que definir distintas *utterances* y tendremos que definir un conjunto específico de *utterances* que mapean a un *intent* específico, como se muestra en la Figura 21.

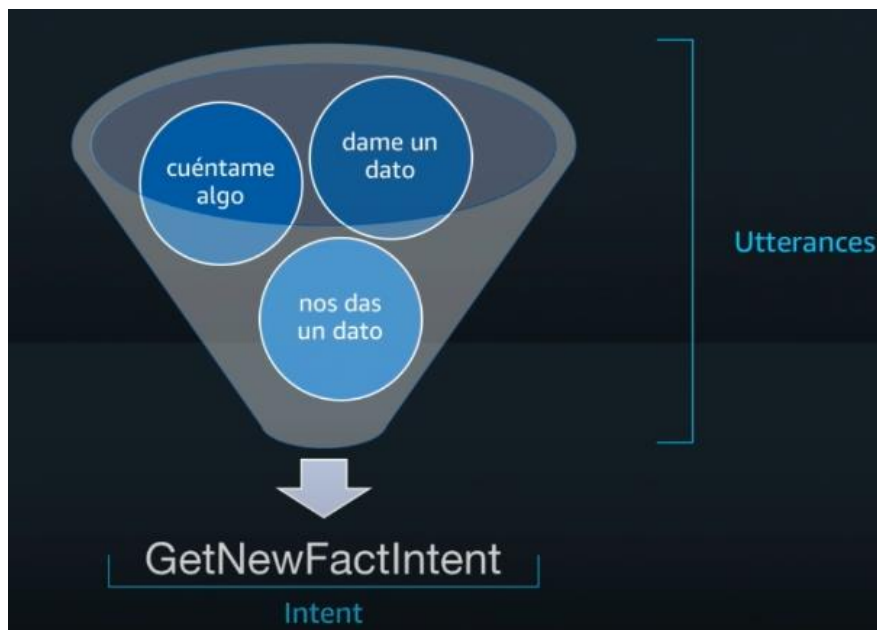


Figura 21. Gráfico de la estructura del front-end.

Debes tener en cuenta que cada vez que definamos una serie de *utterances* o enunciados para un determinado *intent* estamos aportando material de entrenamiento. Por lo que quiere decir que no es necesario que contemples todas las posibilidades que el usuario podría decir con respecto a ese *intent*.

Supongamos que decimos: *“Alexa, pide a curiosidades del espacio un dato de marte”*. Como podemos observar, nuestra intención ha cambiado ya no queremos cualquier dato sino uno específico de un planeta.

Por ello esta intención la vamos a mapear a un *intent* específico que podemos llamar *GetFactByPlanetIntent*.

Si queremos que nuestra skill pueda capturar cualquier planeta al que se refiera el usuario tendremos que utilizar *slots* o variables. Los *slots* pueden tener un tipo predefinido o puedes crearlo tú si el tipo no existe.

En este caso el tipo de planeta no existe es por ello por lo que tocaría crear un tipo de *slot* que tenga toda la lista de planetas posible. Posteriormente podríamos calcular el valor simplemente preguntando por el valor del planeta, como podemos observar en la Figura 22.

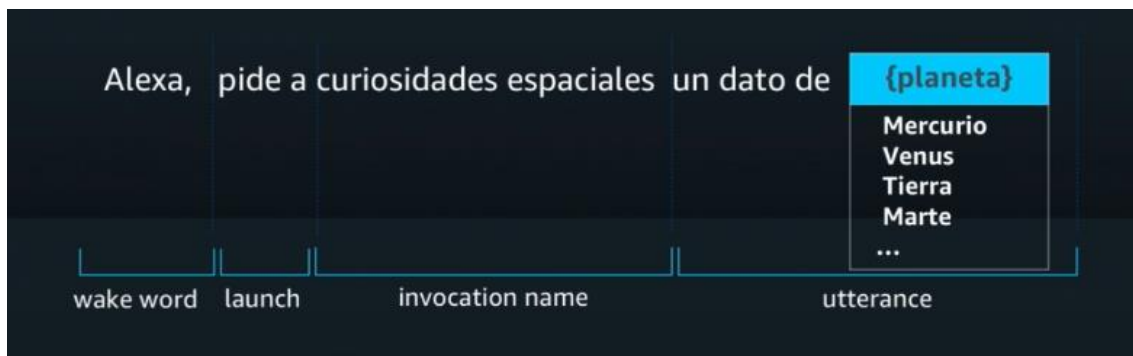


Figura 22. Ilustración de un tipo de slots.

Si bien cuando hacemos una skill definimos nuestros propios *intents* en base a lo que la habilidad tiene que hacer, también existen *intents* predeterminados. Estos *intents* están incorporados en todas nuestras skills. En este caso los tres *intents* más importantes son el de cancelar, el de ayuda y el de parar. Es importante que la habilidad tenga incorporado estos *intents* para que todas las skills respondan de la misma manera ante estas intenciones y así la experiencia del usuario sea homogénea.

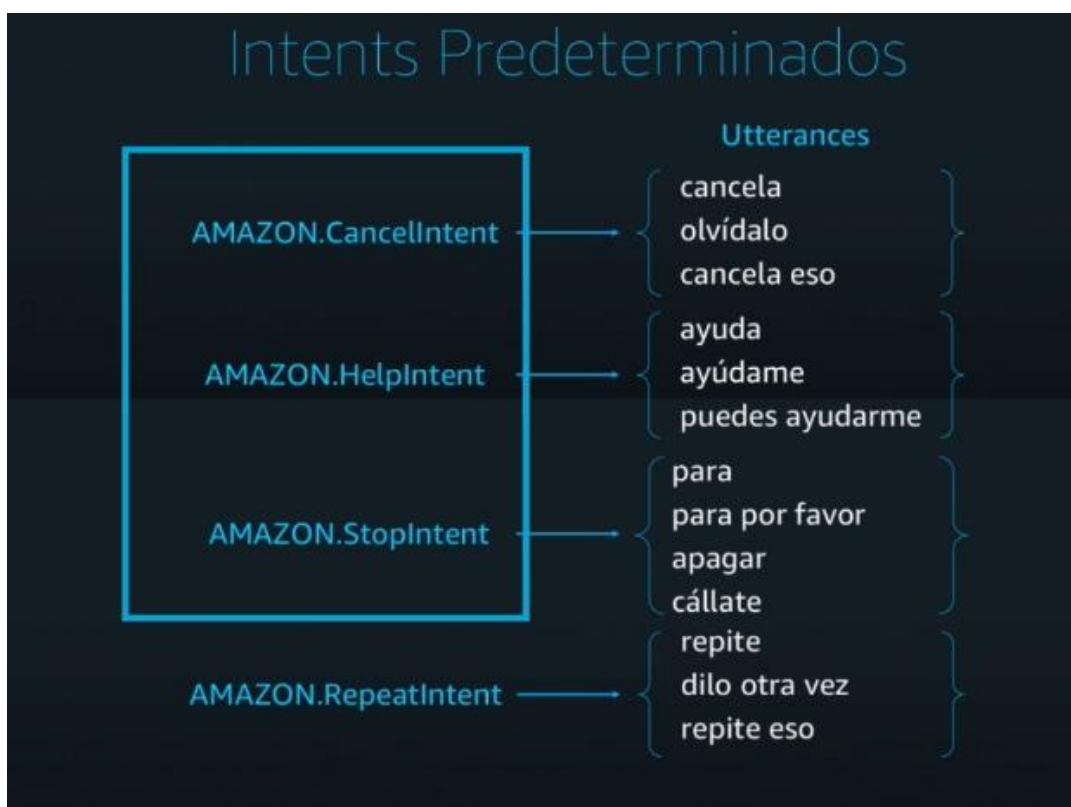


Figura 23. Ilustración de la lista con los intents predeterminados.

Un punto importante en el que hay que enfatizar es que los *intents* predeterminados ya cuentan con frases de entrenamiento para comprender muchas variaciones de como los

usuarios pueden pedir ayuda o pedirle a Alexa que pare. Sin embargo, puedes extender estas frases si lo ves necesario.

3.3.2.2 Back-end

En esta parte de la estructura no nos pararemos mucho tiempo ya que en el siguiente capítulo desarrollaremos toda la estructura del *back-end* e iremos explicando punto por punto cada una de las estructuras utilizadas en nuestra skill.

Pero si hablaremos de ASK, que es el entorno donde tiene lugar todo el back-end.

El Alexa Skills Kit (ASK) es un conjunto de herramientas, documentación, muestras de código y API en *self-service* con el que puedes añadir skills a Alexa de forma rápida y sencilla. El ASK permite a diseñadores, desarrolladores y marcas crear skills atractivas para poder llegar a los consumidores. Con este kit, puedes aprovechar el conocimiento y la innovación de Amazon en el sector del diseño de voz.

Por otro lado, podemos encontrar el AWS Lambda, que es un servicio informático sin servidor y basado en eventos que le permite ejecutar código para prácticamente cualquier tipo de aplicación o servicio de *back-end* sin necesidad de aprovisionar o administrar servidores. Es por ello que todo el código se ejecuta en Lambda.

AWS Lambda es compatible con una gran variedad de lenguajes de programación entre ellos destacan los que aparecen en la tabla de la *Figura 24*.

Lenguajes de programación
Java
JavaScript
Go
PowerShell
Node.js
C#
Python
Ruby

Figura 24. Tabla de lenguajes de programación compatibles con AWS Lambda [11]

Visualizando la gran variedad de lenguajes de programación, es difícil escoger el ideal para desarrollar nuestra skill. Sin embargo, en este estudio nos hemos decantado por el lenguaje de programación de JavaScript.

A continuación, explicaremos las razones que nos hacen decantarnos por este lenguaje de programación.

JavaScript es un lenguaje de programación ligero orientado a objetos para crear secuencias de comandos en las skills desarrolladas en ASK. Es un lenguaje de programación interpretado y completo. En la siguiente tabla, podremos observar las características que nos hicieron decantarnos por él.

JavaScript
Lenguaje multiplataforma
Lenguaje rápido de ejecutar.
Lenguaje sencillo y versátil
Brinda la posibilidad de trabajar en modo <i>Full Stack</i> .

Para terminar este punto hablaremos de los archivos que representan una skill, es decir, si quisiésemos descargarnos una skill que elementos se adjuntarían.

- Por un lado, tendríamos una carpeta llamada */models* donde encontraríamos los *voice interaction models* o interfaces de usuario de voz. En esta carpeta tendríamos que haber definido los lenguajes que queremos que soporte nuestra skill. Esta carpeta pertenecería al *front-end*. El documento se llamaría */es-ES.json*.
- Por otro lado, tendríamos una carpeta que se llama */lambda/custom* donde estaría alojado nuestro código del *back-end*. Dependiendo del lenguaje los archivos en la carpeta de *lambda* pueden tener diferentes nombres, en nuestro caso se llamará */index.js*. Porque nuestro futuro *back-end* estará escrito en *Node.js*, es decir, en JavaScript. Nuestro ASK de *Node.js* es el más popular, pero también tenemos ASK para otros lenguajes como hemos podido observar anteriormente.

Una vez escenificado la arquitectura de una skill de Alexa, ya podremos adentrarnos en la explicación de la plataforma de desarrollo, la cual nos sorprenderá la sencillez y el dinamismo que nos proporciona a los desarrolladores de skills.

3.4 Plataforma de desarrollo

Como bien hemos explicado anteriormente, todo el desarrollo de la skill tiene lugar en el ASK. La plataforma de desarrollo nos facilita una gran accesibilidad y simplicidad a la hora de desarrollar skills como podremos observar a continuación.

Instrucciones para el desarrollo de una skill en ASK:

Para poder entrar a la plataforma de desarrollo lo primero que debemos hacer es entrar en *Amazon Developer*. Una vez entrado en la página deberán clicar en *Amazon Alexa* como se muestra en la *Figura 26*.

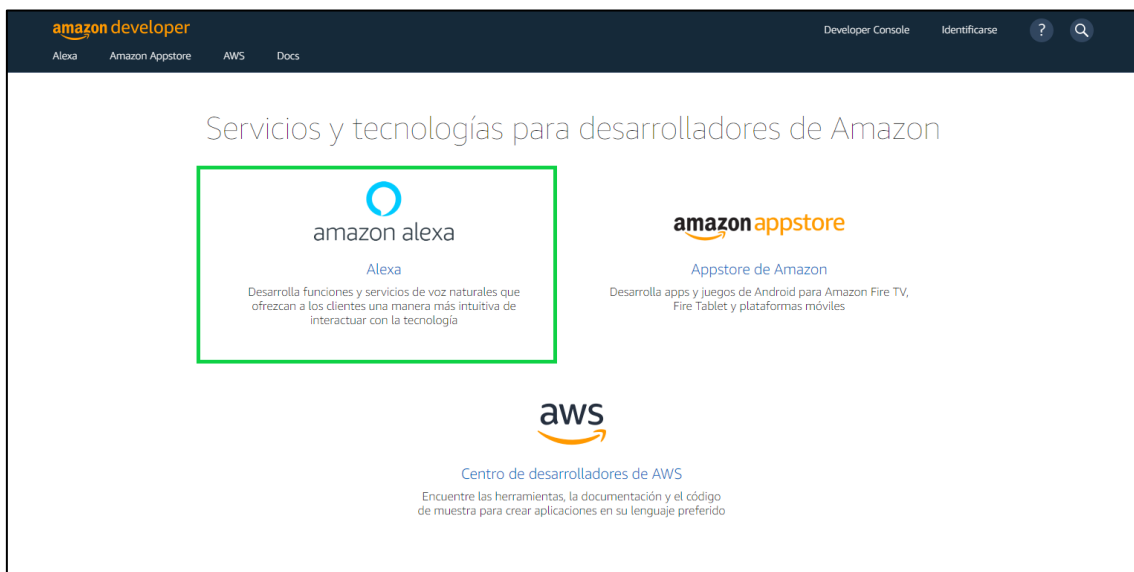


Figura 25. Recorte de la plataforma de desarrollo de Alexa.

Tendremos que crearnos una cuenta de desarrollador, la cual es totalmente gratuita. Para crear la cuenta solo se necesita un email y una contraseña, en ningún caso se necesita ningún tipo de licencia.



Figura 26. Recorte de la plataforma de desarrollo de Alexa.

Una vez creada nuestra cuenta nos dirigiremos a la pestaña de arriba a la izquierda y clicaremos en Desarrolladores de Skill > Consola de desarrollo como se muestra en la Figura 27.

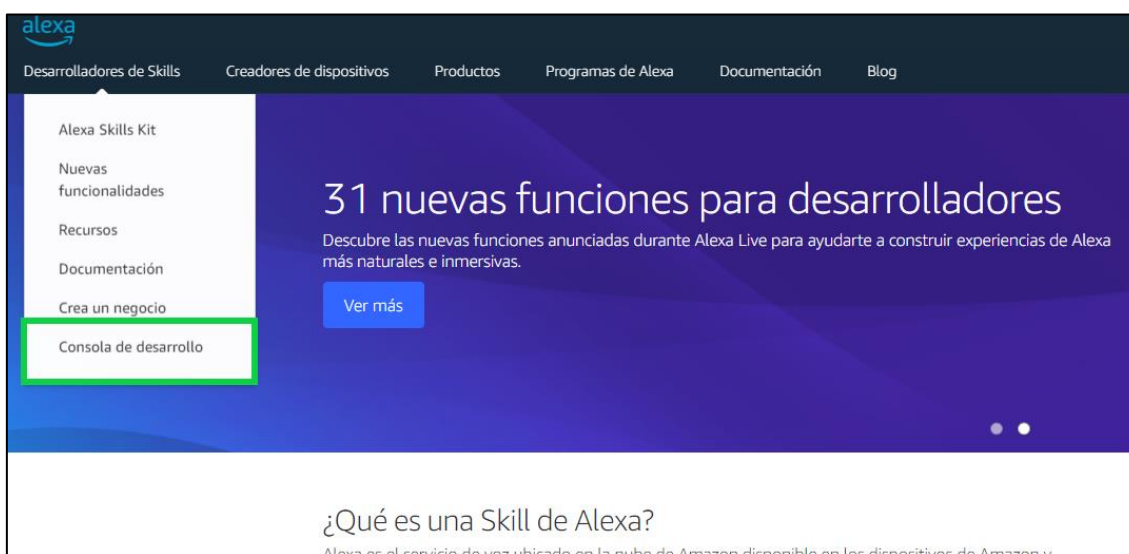


Figura 27. Recorte de la plataforma de desarrollo de Alexa.

Como podéis observar, en mi portal ya hay diferentes skills creadas, si es la primera vez que entráis no os aparecerá ninguna skill como es normal. Una vez entrado clicaremos en lo botón de “crear skill” para poder empezar nuestra habilidad como se muestra en la *Figura 28*.

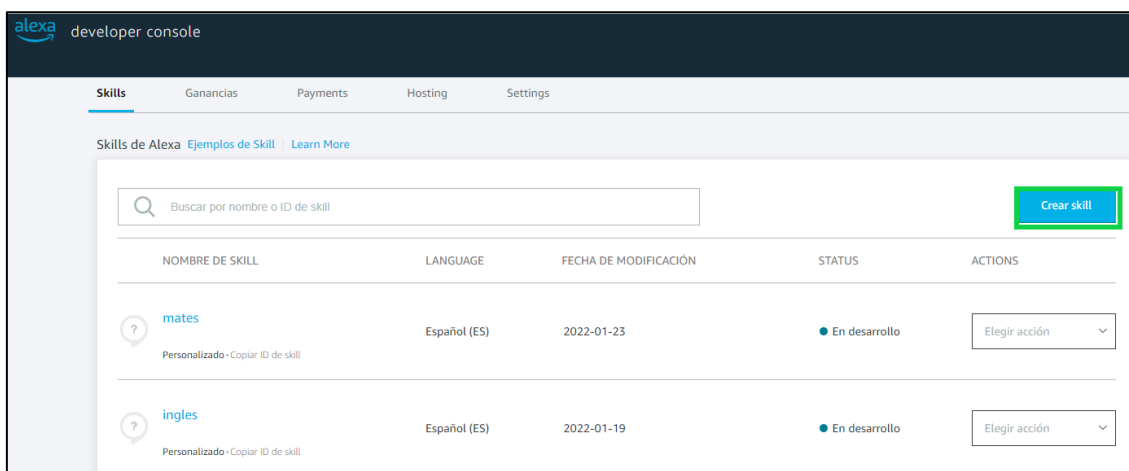
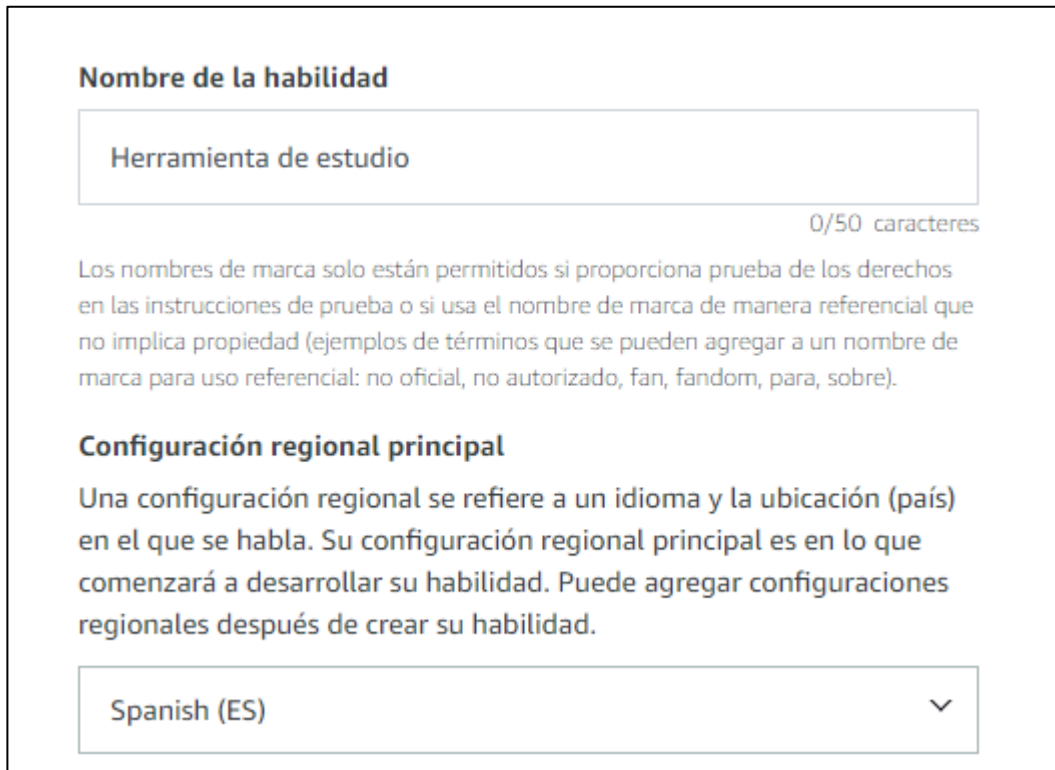


Figura 28. Recorte de la plataforma de desarrollo de Alexa.

Lo primero que debemos aportar es el nombre de la skill, que no es lo mismo que el *invocation name*. El nombre de la skill corresponde a la descripción básica con la que el usuario podrá encontrar tu habilidad en la tienda de skill.

En este caso vamos a ponerle “Herramienta de estudio”. En nuestro caso, como la skill va a ser desarrollada en español, debemos que indicar el idioma de la habilidad.



Nombre de la habilidad

Herramienta de estudio

0/50 caracteres

Los nombres de marca solo están permitidos si proporciona prueba de los derechos en las instrucciones de prueba o si usa el nombre de marca de manera referencial que no implica propiedad (ejemplos de términos que se pueden agregar a un nombre de marca para uso referencial: no oficial, no autorizado, fan, fandom, para, sobre).

Configuración regional principal

Una configuración regional se refiere a un idioma y la ubicación (país) en el que se habla. Su configuración regional principal es en lo que comenzará a desarrollar su habilidad. Puede agregar configuraciones regionales después de crear su habilidad.

Spanish (ES) ▼

Figura 29. Recorte de la plataforma de desarrollo de Alexa.

A continuación, seleccionamos el tipo de skill. En nuestro caso elegiremos *Custom*, ya es un modelo que nos proporciona un mayor grado de libertad a la hora de desarrollar nuestra skill.

Además, seleccionaremos donde queremos alojar nuestro *back-end*. Como bien hemos explicado en el capítulo anterior, nuestro código estará situado en el *Node.js* como se muestra en la *Figura 30*.

1. Elija un modelo para agregar a su habilidad

Hay muchas maneras de empezar a desarrollar una habilidad. Puede diseñar su propio modelo personalizado o comenzar con un modelo preconstruido. Los modelos preconstruidos son modelos de interacción que contienen un paquete de intenciones y declaraciones que puede agregar a su habilidad.

Personalizado

Diseña una experiencia única para tus usuarios. Un modelo personalizado le permite crear todas las interacciones de su habilidad.

Informe relámpago

Ofrezca a los usuarios el control de sus noticias. Este modelo preconstruido permite a los usuarios controlar qué actualizaciones escuchan.

"Alexa, pon el resumen de noticias."

Casa inteligente

Ofrezca a los usuarios el control de sus dispositivos domésticos inteligentes. Este modelo prefabricado permite a los usuarios apagar las luces y otros dispositivos sin levantarse.

"Alexa, enciende las luces de la cocina"

Video

Permita que los usuarios encuentren y consuman contenido de video. Este modelo prediseñado admite búsquedas de contenido y sugerencias de contenido.

"Alexa, pon Interstellar"

2. Elija un método para alojar los recursos de back-end de su habilidad

Puede aprovisionar sus propios recursos de back-end o puede hacer que Alexa los aloje por usted. Si decide que Alexa aloje su habilidad, obtendrá acceso a nuestro editor de código, que le permitirá implementar el código directamente en AWS Lambda desde la consola del desarrollador.

Alojado por Alexa (Node.js)

Alexa alojará habilidades en su cuenta y lo ayudará a comenzar con una plantilla de Node.js. Obtendrá acceso a puntos finales de AWS Lambda en todas las regiones de servicio de Alexa, una tabla de DynamoDB para la persistencia de datos y S3 para almacenamiento de medios. [Aprende más](#)

Alojado por Alexa (Python)

Alexa alojará habilidades en su cuenta y lo ayudará a comenzar con una plantilla de Python. Obtendrá acceso a puntos finales de AWS Lambda en todas las regiones de servicio de Alexa, una tabla de DynamoDB para la persistencia de datos y S3 para almacenamiento de medios. [Aprende más](#)

Aprovisiona tu propio

Aprovisione su propio punto final y recursos de back-end para su habilidad. Esto se recomienda para habilidades que tienen requisitos significativos de transferencia de datos. No obtendrá acceso al editor de código de la consola.

Figura 30. Recorte de la plataforma de desarrollo de Alexa.

Una vez seleccionado todas estas opciones, podremos clicar en el botón de arriba a la derecha donde pone “Crear habilidad”.

En este punto nos preguntarán con que plantilla queremos empezar a desarrollar en nuestro caso, elegiremos “Empezar desde el principio”. Esta plantilla te genera el conocido programa, que todos los que hayamos programado alguna vez fue nuestro primer programa. En efecto, “¡Hola mundo!”. Nos será muy útil porque habrá estructuras que seguro que podremos reutilizar.

Empezar desde el principio

Esta habilidad lo ayuda a comenzar con las intenciones requeridas y con el código que demuestra la funcionalidad “Hello World” si está creando una habilidad alojada en Alexa. [Aprende más](#)

Por Alexa

Habilidad de hechos

Desarrolle una habilidad de hecho atractiva sobre cualquier tema. Alexa seleccionará un hecho al azar y lo compartirá con el usuario cuando se invoque la habilidad. [Aprende más](#)

Incluye: intenciones personalizadas, Personalización

Por Alexa

Habilidad de programación

Desarrolle una habilidad para permitir a los usuarios programar citas en su calendario, recibir confirmaciones y recordatorios por correo electrónico. [Aprende más](#)

Incluye: permisos de voz, recordatorios, llamadas API, persistencia de sesión

Por Laboratorio Dabble

Habilidad de encuesta

Desarrolle una habilidad de stand-up o encuesta que use códigos de acceso para permitir que solo los usuarios autorizados proporcionen actualizaciones y respondan preguntas. [Aprende más](#)

Incluye: uso de códigos de acceso, llamadas API, persistencia de sesión

Por Laboratorio Dabble

Introducción a las conversaciones

Esta habilidad lo introduce a las conversaciones proporcionar la funcionalidad básica de “color una respuesta de voz de Alexa. [Aprende más](#)

Incluye: vista previa de conversaciones de # para audio, persistencia de sesión

Por Alexa

Figura 31. Recorte de la plataforma de desarrollo de Alexa.

Una vez seleccionado el paso anterior, se nos creará la plantilla para poder empezar a desarrollar nuestra skill. En la página principal, veremos cómo hay diferentes pestañas donde el desarrollador puede interactuar para crear su skill como se observa en la *Figura 32*.

Empezaremos declarando cuál queremos que sea el *invocation name* de nuestra skill. Para ello seleccionaremos donde pone “*Invocation name*”.

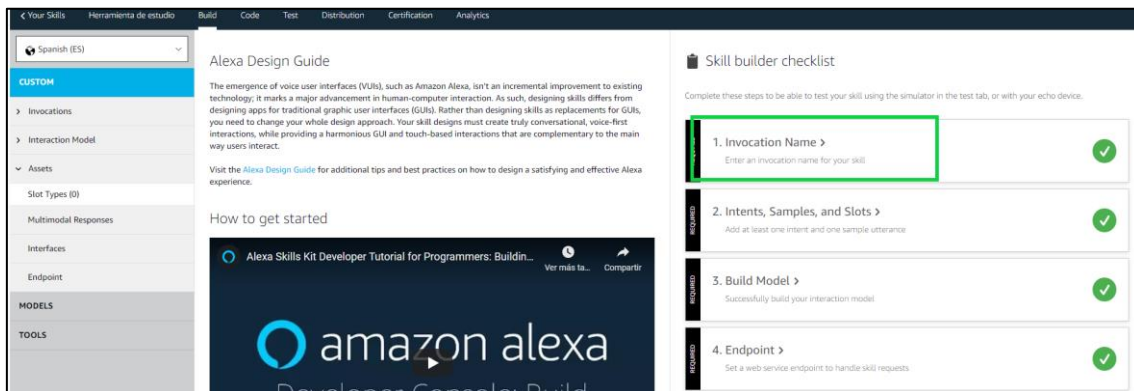


Figura 32. Recorte de la plataforma de desarrollo de Alexa.

Una vez hemos puesto el *invocation name* que queremos para nuestra skill es muy importante que le demos a “*save*” y después a “*build model*”. Nuestro modelo de la interfaz se actualizará sólo sin tener que escribir ningún código.

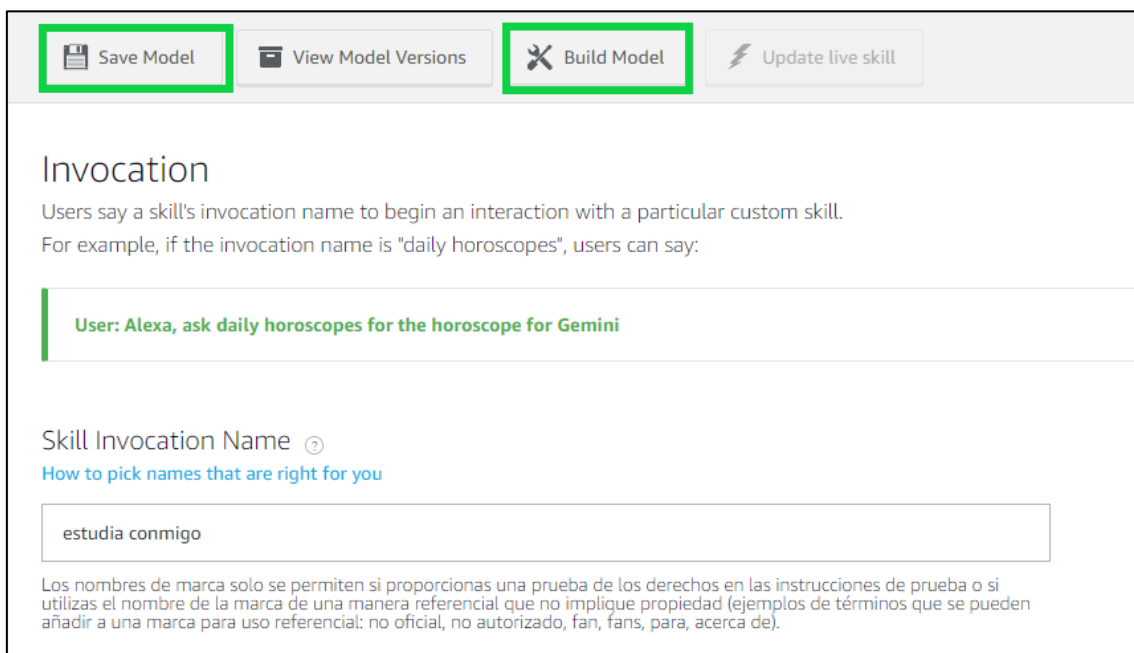


Figura 33. Recorte de la plataforma de desarrollo de Alexa.

Si nos movemos a la pestaña de *intents* podemos observar cómo solo tenemos *HelloWorldIntent* con muchas *sample utterances*. Es decir, esta skill esta esperando a que el usuario diga cualquiera de las *sample utterances* para activar *HelloWorldIntent*.

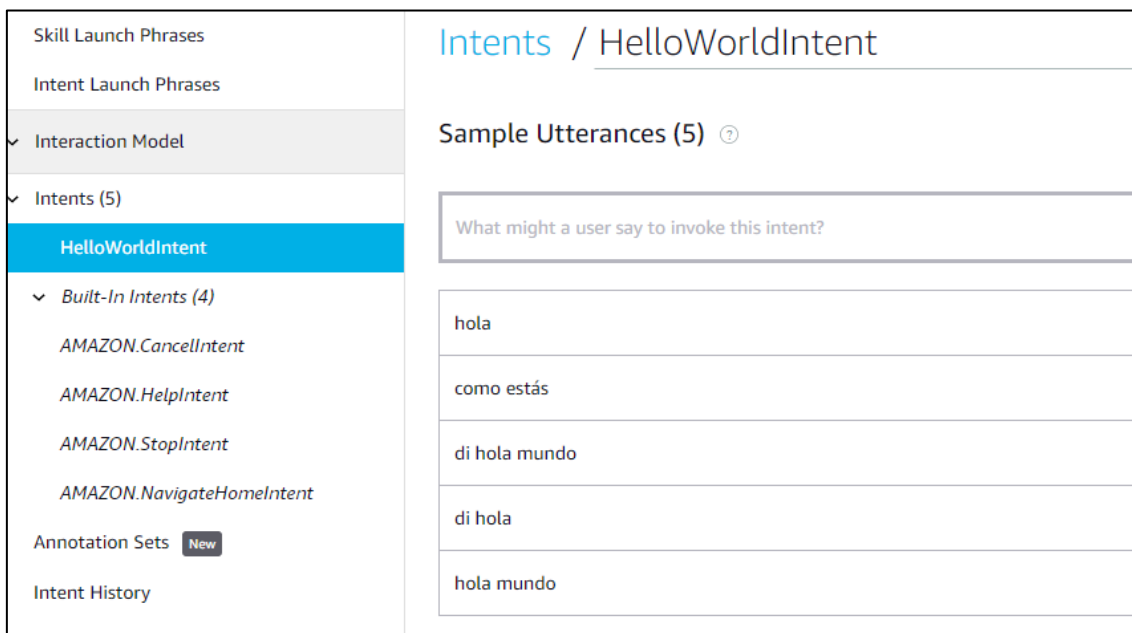


Figura 34. Recorte de la plataforma de desarrollo de Alexa.

Una vez explorado estas pestañas, indagaremos sobre el archivo que se ha generado por debajo, es decir, el JSON. Como podemos observar el código se ha actualizado sin tener que nosotros tocar nada de él.

También podemos observar cómo están los tres intents predeterminados que debían estar en todas nuestras skills como son: *CancelIntent*, *HelpIntent* y *StopIntent*.

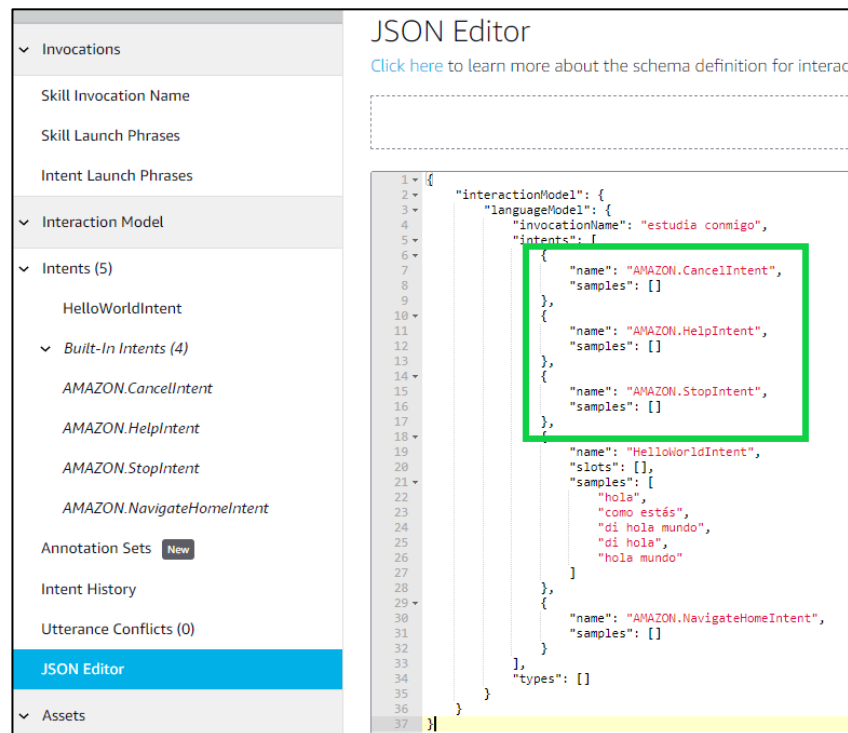


Figura 35. Recorte de la plataforma de desarrollo de Alexa.

Una vez citado lo que es el *front-end*, nos moveremos al *back-end*. Para ello, debemos de clicar en la pestaña de “Code” como se muestra en la *Figura 36*.

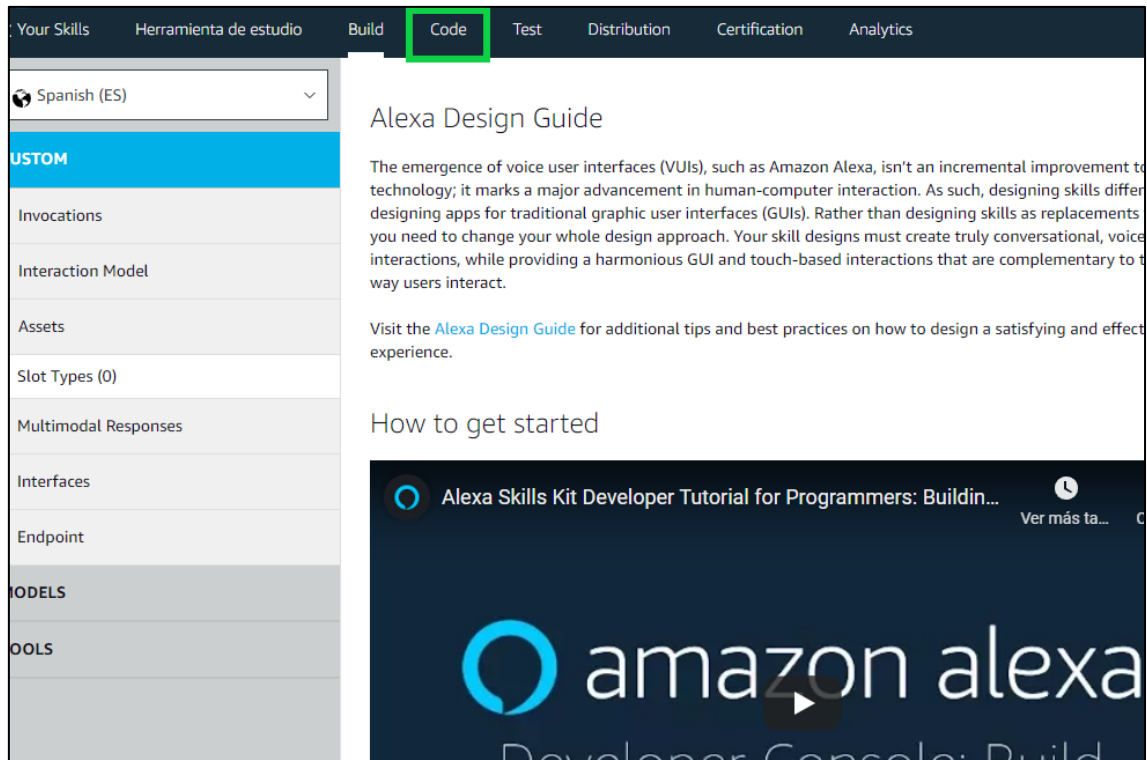


Figura 36. Recorte de la plataforma de desarrollo de Alexa.

En esta pestaña podremos encontrar todo nuestro *back-end*, en esta parte no profundizaremos en este capítulo ya que en el siguiente lo explicaremos punto por punto.

Sin embargo, podéis observar que estilo tendrá mirando la *Figura 37*.

```

index.js x
1  /*
2   * This sample demonstrates handling intents from an Alexa skill using the Alexa Skills Kit SDK (v2).
3   * Please visit https://alexa.design/cookbook for additional examples on implementing slots, dialog management,
4   * session persistence, api calls, and more.
5   */
6  const Alexa = require('ask-sdk-core');
7
8  const LaunchRequestHandler = {
9    canHandle(handlerInput) {
10      return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchRequest';
11    },
12    handle(handlerInput) {
13      const speakOutput = 'Welcome, you can say Hello or Help. Which would you like to try?';
14
15      return handlerInput.responseBuilder
16        .speak(speakOutput)
17        .reprompt(speakOutput)
18        .getResponse();
19    }
20  };
21
22  const HelloWorldIntentHandler = {
23    canHandle(handlerInput) {
24      return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
25        && Alexa.getIntentName(handlerInput.requestEnvelope) === 'HelloWorldIntent';
26    },
27    handle(handlerInput) {
28      const speakOutput = 'Hello World!';
29
30      return handlerInput.responseBuilder
31        .speak(speakOutput)
32        //.reprompt('add a reprompt if you want to keep the session open for the user to respond')
33        .getResponse();
34    }
35  };
36
37  const HelpIntentHandler = {
38    canHandle(handlerInput) {
39      return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
40        && Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.HelpIntent';
41    },
42    handle(handlerInput) {
43      const speakOutput = 'You can say hello to me! How can I help?';
44
45      return handlerInput.responseBuilder
46        .speak(speakOutput)
47        .reprompt(speakOutput)
48        .getResponse();
49    }
50  };
51
52  const Handlers = [LaunchRequestHandler, HelloWorldIntentHandler, HelpIntentHandler];
53  const app = new ASKSdk({
54    handlers: Handlers,
55    onSessionStarted: () => {},
56    onSessionEnded: () => {}
57  });
58
59  export default app;

```

Figura 37. Recorte de la plataforma de desarrollo de Alexa.

En cambio, si que profundizaremos en la forma de como correr nuestro código para después probarlo. Para ello, una vez hemos hecho todas las modificaciones a nuestro *back-end*. Debemos darle al botón de “Save” y después al de “Deploy”. Este paso es muy importante, ya que se suele olvidar. Los botones se encuentran arriba a la derecha.

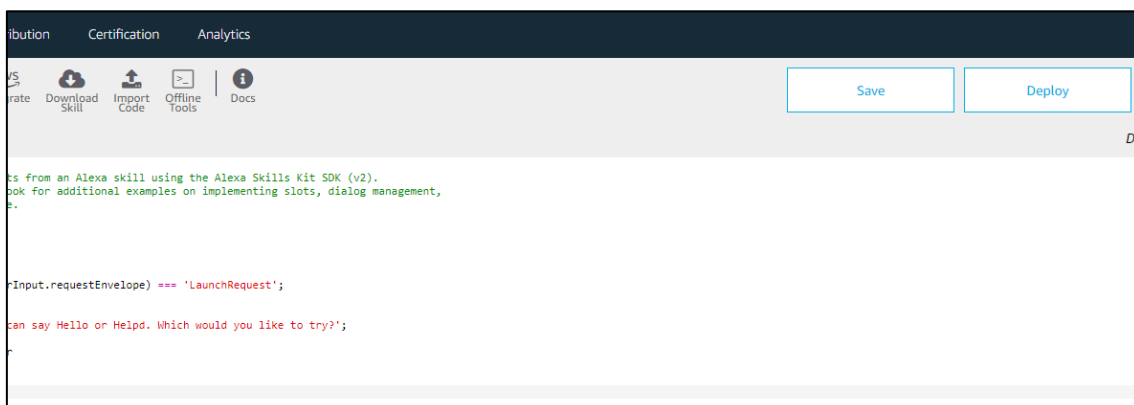


Figura 38. Recorte de la plataforma de desarrollo de Alexa.

Una vez hecho el *deploy* ya podremos probar nuestra skill en la pestaña de “Test”. Dentro de la pestaña, debemos seleccionar la opción de “Development” como se muestra en la *Figura 39*. Finalmente, ya estarían todos los pasos explicados para poder testear la skill que hemos desarrollado.

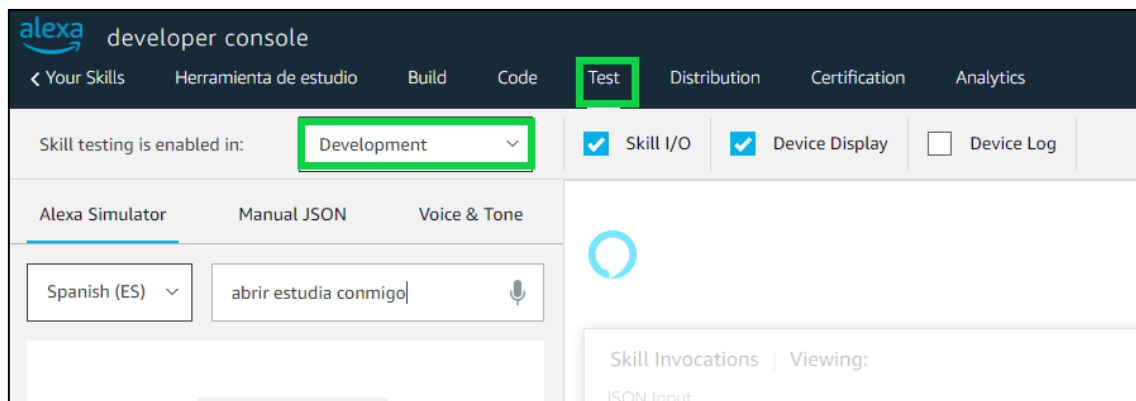


Figura 39. Recorte de la plataforma de desarrollo de Alexa.

CAPÍTULO 4. PRESENTACIÓN Y DESARROLLO DE LA SKILL “ESTUDIA CONMIGO”

En este capítulo, pondremos en práctica todos los conocimientos adquiridos a lo largo de este proyecto para poder desarrollar satisfactoriamente nuestra skill. Empezaremos exponiendo el problema que queremos solventar con el desarrollo de nuestra skill. Continuaremos presentando la skill, definiendo como será su funcionamiento y su aplicación. Entraremos en profundidad en su desarrollo y finalmente testaremos su funcionamiento.

4.1 Exposición del problema

Esta fue una de las cuestiones que más me hizo dilatar en el tiempo. Sabía que quería realizar una skill en la plataforma virtual de Alexa pero quería que mi skill tuviera una aplicación muy directa. Es por ello que invertí mucho tiempo en pensar que problema podía ser resuelto mediante una skill. Otras de las variables a estudiar, era conseguir que yo le viera una posibilidad de integrarlo en mi estilo de vida y no lo viera como algo demasiado superficial e impuesto.

Fue entonces cuando empecé a dictaminar en qué sector quería que tuviese aplicación mi skill. Llegué a la conclusión que debía desarrollar mi skill dentro del sector educativo, ya que el mundo estudiantil había estado presente a lo largo de toda mi vida. Es por ello que veía que mi skill tenía una gran posibilidad de que los usuarios fueran capaces de integrarla a lo largo de su carrera educativa.

Llegado a este punto, solo me hacía falta identificar qué problema educativo quería atacar. Es entonces cuando empecé a pensar en toda mi experiencia educativa, e intentar visualizar donde había podido tener obstáculos los cuales se hubieran podido solucionar con un asistente virtual. Fue entonces cuando me acordé cuando iba a educación primaria, que para mí una de las grandes barreras eran las asignaturas que se basaban en memorizar información más que de entender.

Recuerdo que me costaba mucho el tener la iniciativa de sentarme sólo a estudiarme la lección después de haber estado ocho horas en el colegio. Algunas veces tenía la suerte de que mi madre tuviera tiempo y me fuera preguntando la lección conforme yo me la iba aprendiendo, de esta manera las horas de estudio de por la tarde se me hacían más dinámicas y productivas.

Este problema que yo tenía con diez años, es un obstáculo que gran parte del alumnado con esas edades ha tenido que lidiar alguna vez. Cuando somos más pequeños, se nos hace más difícil visualizar cual es la finalidad de estudiar por lo que el muro que se nos crea es más grande. Una parte del cuerpo estudiantil tiene la fortuna de tener una tercera persona, con la cual puede ir interactuando para dinamizar las horas de estudio.

4.2 Presentación de la skill “ESTUDIA CONMIGO”

Una vez fijado el problema, la siguiente pregunta era cómo abordarlo. A lo largo de este estudio, hemos observado como la mayor aplicación que tienen los asistentes virtuales es en la de interactuar con las personas. Por lo que parece que el problema y la solución parecen estar sincronizadas.

Si tuviéramos que fijar una visión del proyecto, sería la de desarrollar una skill la cual sea capaz de interactuar con el estudiante, de una manera que consiga dinamizar y aumentar la productividad de su tiempo de estudio.

Una vez fijada la visión, estableceremos que objetivos debe de cumplir nuestra skill

1. El primero objetivo que vamos a establecer es que nuestra skill tenga una interfaz muy simple y dinámica. Como hemos expuesto en el apartado anterior, nuestra skill entra en acción cuando el estudiante vuelve de pasar muchas horas en el colegio por lo que debemos ser una herramienta dinámica a los ojos del estudiante.
2. El segundo de ellos se basa en que nuestra skill sea una fuente de información, por lo que necesitamos que el asistente virtual pueda recitar la lección que el estudiante se tiene que aprender. De esta forma estaremos rompiendo otra barrera, ya que le daríamos la posibilidad de volver a asistir a la lección desarrollada en la clase.
3. El tercer objetivo se basa en desarrollar una experiencia educativa la cual consiga testear los conocimientos del estudiante, de una forma interactiva y resolutive. Aportándole no solo una experiencia dinámica sino también aportándole un *feedback* del grado de conocimiento de la lección.

Estos tres puntos, intentarán ser cubiertos mediante el desarrollo de mi propia skill llamada “Estudia conmigo”.

La estructura de la skill la podemos presentar en dos escenarios bien diferenciados. Se buscará un acceso a cada uno de los escenarios de la forma más rápida y efectiva. También buscaremos que la skill pueda establecer puentes entre los dos escenarios, ya que queremos que el usuario o en este caso el estudiante tenga una gran independencia de movimiento.

Antes de presentar cada uno de los escenarios, introduciremos la interacción inicial donde el alumno podrá elegir con qué escenario quiere interactuar.

Estudiante: *Alexa, abrir estudia conmigo.*

Alexa: *Hola Javier, veo que quieres ponerte a estudiar la lección de geografía. Si quieres que te explique la lección, di lección. Y si quieres que te la pregunte, di preguntas.*

Figura 40. Cuadro de diálogo de la skill “Estudia conmigo”.

Como podemos observar, Alexa nos proporcionará dos opciones. En este punto para que el estudiante reciba la lección, deberá decir “lección” como se muestra en la *Figura 41*.

A continuación, podremos visualizar el escenario uno.

Estudiante: *Alexa, abrir estudia conmigo.*

Alexa: *Hola Javier, veo que quieres ponerte a estudiar la lección de geografía. Si quieres que te explique la lección, di lección. Y si quieres que te la pregunte, di preguntas.*

Estudiante: *Lección.*

Alexa: *Lección del tema cuatro. Los ríos de España. ¿De qué dependen las características de los ríos? La longitud de nuestros ríos, su caudal y su régimen a lo largo del año, dependen del clima y del relieve. Los ríos son corrientes continuas de agua que nacen en las montañas y desembocan en el mar, en un lago o en otro río. En un río podemos distinguir tres elementos: el curso, el caudal y el régimen. El curso. Es el recorrido que realiza un río desde su nacimiento hasta su desembocadura. Se distinguen tres tramos: curso alto, curso medio y curso bajo. El caudal. Es la cantidad de agua que lleva un río. El régimen. Es la variación que sufre el caudal de un río a lo largo de año. El régimen es regular, cuando la variación de desembocan en tres vertientes: Cantábrica, Atlántica y Mediterránea. La vertiente cantábrica. Está al norte de España. La vertiente mediterránea. Situada al este y sur de España. La vertiente atlántica. Está al oeste de España. A continuación, hablaremos de los ríos más importantes de España. El río Miño*

Alexa: Por la derecha los ríos Jalón y Guadalope. Por la izquierda los ríos Aragón, Gállego, Cinca y Segre. Hola Javier, ya hemos terminado la lección del tema cuatro, que prefieres que te la pregunte o que te la vuelva a explicar.

Figura 41. Cuadro de diálogo de la skill “Estudia conmigo”.

Como podemos observar, hasta aquí sería nuestro escenario uno. Tenemos que intentar que Alexa sea capaz de recitar la lección de una forma continuada y articulada para que el estudiante pueda retener los conocimientos.

Para ambientar esta skill y darle un aire de realismo, nuestra habilidad estará aplicada a una lección de geografía, es por ello que la temática predominante serán los ríos de España.

Por otro lado, presentaremos el segundo escenario, en el cual se desarrollará una interacción con el usuario. Esta interacción se basará en que Alexa sea capaz de hacerle preguntas al estudiante y este responda. Una vez el estudiante haya respondido, Alexa tendrá que verificar si la respuesta que le ha proporcionado es correcta o no y comunicárselo. Alexa le formulará todas las preguntas, y al final le generará un *re5port* donde le transmitirá cuantas preguntas ha acertado. De esta manera el usuario tendrá una visión global del grado de conocimiento que tiene de la lección.

A continuación, visualizaremos el escenario dos.

Estudiante: Alexa, abrir estudia conmigo.

Alexa: Hola Javier, veo que quieres ponerte a estudiar la lección de geografía. Si quieres que te explique la lección, di lección. Y si quieres que te la pregunte, di preguntas.

Estudiante: Preguntas.

Alexa: Muy bien Javier, ahí va tu 1ª pregunta. ¿Dónde nace el río Guadiana? En la sierra de:

Estudiante: Albarracín.

Alexa: ¡Perfecto! Albarracín es correcto. Aquí tienes tu 2ª pregunta. ¿En qué picos nace el río Duero? En los Picos de:

Estudiante: Urbión.

Alexa: ¡Perfecto! Urbión es correcto. Aquí tienes tu 3ª pregunta. ¿Cómo se denomina al recorrido que realiza un río desde su nacimiento hasta su desembocadura?

Estudiante: Caudal.

Alexa: ¡Mal! Caudal es incorrecto. La respuesta correcta era Miño. Aquí tienes tu 4ª pregunta. ¿Qué río tiene como afluente más importante el río Sil? El río:

Estudiante: Miño.

Figura 42. Cuadro de diálogo de la skill “Estudia conmigo”.

En esta interacción buscamos que sea una conversación fluida, con preguntas y respuestas concisas. De esta manera mantendremos la atención del estudiante, ya que aparte el asistente no parará de darle *feedback* de sus respuestas. Ahora nos adentramos en la parte final del escenario dos, como podremos observar en la Figura 43.

Alexa: ¡Perfecto! Pisuerga y Esla es correcto. Hasta aquí las preguntas. Tu puntuación final ha sido 12 aciertos de 20 preguntas. Pero recuerda Javier, la clave del éxito está en la constancia. Ya hemos terminado con el cuestionario, ¿qué prefieres que te la vuelva a preguntar o que te explique la lección?

Figura 43. Cuadro de diálogo de la skill “Estudia conmigo”.

Podemos ver como la skill nos proporcionará un informe con la cantidad de respuestas acertadas al final del cuestionario. Aparte nos preguntará si queremos que nos vuelva a preguntar la lección o que nos la explique.

A lo largo de toda la interacción, crearemos la opción de que el usuario pueda cerrar la skill cuando quiera, para ello tendremos que generar el siguiente diálogo como se muestra en Figura 44.

El usuario solo tendrá que decir “cancela” o “para”, y la skill se cerrará.

Alexa: ¡Mal! Caudal es incorrecto. Aquí tienes tu 4ª pregunta. ¿Qué río tiene como afluente más importante el río Sil? El río:

Estudiante: Cancela.

Alexa: Perfecto Javier. Espero que repasemos esta lección en otro momento. Y recuerda, cuanto más duramente trabajo, más suerte tengo.

Figura 44. Cuadro de diálogo de la skill “Estudia conmigo”.

Una vez presentado los diferentes escenarios, ya podemos adentrarnos en el desarrollo de la skill.

4.3 Desarrollo de la skill

Para explicar nuestro desarrollo, iremos saltando del *back-end* al *front-end*. Iremos explicando cada estructura utilizada de una manera instructiva y sofisticada. El desarrollo lo dividiremos en diferentes pasos, para así facilitar la explicación al lector.

Paso 1: Incluir el ASK SDK

Lo primero que haremos será usar la función de JavaScript para incluir el módulo de ASK SDK en nuestro código de Lambda. El módulo Core ASK SDK nos brinda todas las herramientas necesarias para comenzar nuestra skill.

```
const Alexa = require('ask-sdk-core');
```

Paso 2: Establecer el primer handler llamado LaunchRequest

Cuando el usuario inicia la habilidad diciendo “*abrir estudia conmigo*”, la skill manda una solicitud de tipo “*LaunchRequest*”. Para manejar esta solicitud, crearemos el primer *handler*.

La estructura del *handler* internamente se pregunta, “¿*puedo manejar esto?*” por lo que significa que cada *handler* establece las condiciones específicas que es capaz de manejar. Esto se hace dentro de la función *canHandle()* definida dentro del controlador. En esta parte, se establece los *intents* con los que se acciona el *canHandle()*. Si las condiciones establecidas en el *canHandle()* te devuelven “verdadero”. Se ejecuta el *handle()*, en cambio si te devuelve “falso” la solicitud continua la búsqueda del controlador adecuado.

```
const LaunchRequestHandler = {
  canHandle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    return request.type === "LaunchRequest";
  },
  handle(handlerInput) {
    const speechOutput = BIENVENIDA_MENSAJE;
    const repromptSpeechOutput = AYUDA_MENSAJE;
    return handlerInput.responseBuilder
      .speak(speechOutput)
      .reprompt(repromptSpeechOutput)
      .getResponse();
  }
};
```

Cuando el usuario interactúa con una skill de Alexa, la función Lambda recibe una solicitud en forma de JSON. Este archivo contiene la información necesaria para mapear el controlador que es capaz de manejar esa solicitud.

Por lo tanto, cuando el *LaunchRequest* sea igual a nuestro *invocation name* el cual es “estudia conmigo”, el *canHandle()* devolverá “verdadero” por lo que se ejecutará el *handle()*. En nuestro caso solo recitará el mensaje que está dentro de la variable “BIENVENIDA_MENSAJE” a partir de la función de *.speak*.

```
const BIENVENIDA_MENSAJE = "Hola Javier, veo que quieres ponerte a estudiar la lección de geografía. Si quieres que te explique la lección, dí lección. Y si quieres que te la pregunte, dí preguntas.";
```

También destacar la estructura de *.reprompt()*. Esta estructura se encarga de una vez que se ha ejecutado el *handle()* mantener la sesión de la skill abierta y esperar una respuesta del usuario.

Paso 3: Controlador de funciones de AWS Lambda

Para garantizar que Lambda conozca la entrada de todos nuestros controladores de solicitudes, crearemos la estructura *Alexa.SkillBuilders.custom()*. Si algún controlador o *handler* no aparece en esta lista para Lambda será como si no existiera.

```
const skillBuilder = Alexa.SkillBuilders.custom();
exports.handler = skillBuilder
    .addRequestHandlers(
        CancelAndStopIntentHandler,
        HelpIntentHandler,
        LaunchRequestHandler,
        LeccionHandler,
        PreguntaHandler,
        RespuestaHandler,
        ResultadoFinalHandler
    )
```

Paso 4: Establecer el segundo handler llamado LeccionHandler

Para entrar en el escenario uno, debemos de crear un *handler* el cual será mapeado cuando el usuario aporte el *utterance* que mapea nuestro *handler*. El *utterance* será “lección” el cual activará el intent del *canhandle()*. A continuación, podremos observar cómo queda recogida esta información en nuestro *front-end*.

```
{
  "name": "LeccionIntent",
  "slots": [],
  "samples": [
    "lección"
  ]
},
```

Como podemos observar en el *name* aparece el *intent* al que acciona y en *samples* aparecen todas las posibles *utterances*. En este caso, simplemente existe un solo *utterance*. Llegados a este punto nuestro *back-end* entrará en acción.

```
const LeccionHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'LeccionIntent';
  },
  handle(handlerInput) {
    const speakOutput = 'Lección del tema cuatro. Los ríos de España. ¿De que dependen las características de los ríos? La longitud de nuestros ríos, su caudal y su régimen a lo largo del año, \n dependen del clima y del relieve. Los ríos son corrientes continuas de agua que nacen en las montañas y desembocan en el mar, en un lago o en otro río. En un río podemos distinguir tres elementos:\n el curso, el caudal y el régimen. El curso. Es el recorrido que realiza un río desde su nacimiento hasta su desembocadura. Se distinguen tres tramos: curso alto, curso medio y curso bajo. El caudal.\n Es la cantidad de agua que lleva un río. El régimen. Es la variación que sufre el caudal de un río a lo largo de año. El régimen es regular, cuando la variación de caudal es pequeña; y es irregular, \n cuando la variación de caudal es grande. Las vertientes de España. Un río principal es un río que desemboca directamente en el mar. Un afluente es un río que va a parar a otro río. Una cuenca fluvial \n es el territorio recorrido por un río y sus afluentes. Una vertiente hidrográfica es el conjunto de tierras cuyas aguas van a parar al mismo mar. Los ríos españoles desembocan en tres vertientes: Cantábrica, \n Atlántica y Mediterránea. La vertiente cantábrica. Está al norte de España. La vertiente mediterránea. Situada al este y sur de España. La vertiente atlántica. Está al oeste de España. A continuación, hablaremos \n de los ríos más importantes de España. El río Miño. El río Miño nace en Fuente Miña , provincia de Lugo, pasa por Lugo y Orense y desemboca en el océano Atlántico por La Guardia, provincia de Pontevedra. Sirve de \n frontera entre España y Portugal. Su afluente más importante es el Sil. El río Duero. El río Duero nace en los Picos de Urbión, provincia de Soria, pasa por Soria y Zamora y desemboca en el océano Atlántico por Oporto \n (Portugal). Sus afluentes más importantes son: Por la derecha los ríos Pisuerga y Esla. Por la izquierda los ríos Eresma, Adaja y Tormes. El río Tago. El río Tago nace en la Sierra de Albarracín, provincia de Teruel, \n pasa por Toledo y Aranjuez y desemboca en el océano Atlántico por Lisboa (Portugal). Sus afluentes más importantes son: Por la derecha los ríos Jarama, Alberche, Tiétar y Alagón. Por la izquierda los ríos Almonte y Salor. \n El río Guadiana. El río Guadiana nace en las lagunas de Ruidera, provincia de Ciudad Real, pasa por Mérida y Badajoz y desemboca en el océano Atlántico por Ayamonte, provincia de Huelva. Sus afluentes principales son: \n Por la derecha los ríos Záncara y Cigüela. Por la izquierda los ríos Jabalón, Zújar; Matagorda y Ardila. El río Guadalquivir. El río Guadalquivir nace en la sierra de Cazorla, provincia de Jaén, pasa por Córdoba y Sevilla \n y desemboca en el océano Atlántico por Sanlúcar de Barrameda, provincia de Cádiz. Sus afluentes más importantes son: Por la derecha los ríos Guadalquivir y Guadiato. Por la izquierda los ríos Guadiana Menor y Genil. \n El río Ebro. El río Ebro nace en Fontibre (Cantabria), pasa por Logroño y Zaragoza y desemboca en el mar Mediterráneo por Amposta, provincia de Tarragona. Sus principales afluentes son: Por la derecha los ríos \n Jalón y Guadalope. Por la izquierda los ríos Aragón, Gállego, Cinca y Segre. Hola Javier, ya hemos terminado la lección del tema cuatro, que prefieres que te la pregunte o que te la vuelva a explicar.';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
};
```

Está seria toda la estructura de *Leccionhandler*, donde la estructura es igual a la del anterior *handler*. Sin embargo el *intent* que acciona el *canhandle()* es *LeccionIntent* como bien hemos explicado en el apartado anterior. En el cuerpo del mensaje que recitará el *handle()* es donde hemos incorporado la lección. Al final del mensaje, Alexa le preguntará si quiere volver a escuchar la lección o si quiere pasar al escenario dos, es decir, a las preguntas. En este caso también está incorporada la estructura *.reprompt(speakOutput)* para que Alexa no cierre la sesión.

En este punto, ya habríamos explicado el desarrollo del escenario uno. Ahora nos centraremos en explicar el escenario dos, donde tiene que ser capaz de formularle preguntas al usuario, y comprobar si el resultado es correcto.

Paso 5: Establecer el tercer handler llamado PrimeraPreguntaHandler

La primera acción que deberá hacer nuestro código será la de generar la primera pregunta, para ello primero hablaremos del repositorio donde se guardan las preguntas y las respuestas.

```
const stories = [
  {
    "question": "¿Qué tres elementos podemos distinguir en un
    río?", "answer": ["curso caudal y régimen", "caudal curso y régimen", "curso régimen
    y caudal", "caudal régimen y curso", "régimen caudal y curso", "régimen curso y
    caudal"]
  },

```

La estructura está un poco desordenada al pasarlo al formato de Word, pero suficiente para poder explicarlo. En este caso, nuestro repositorio de preguntas y respuestas estará guardado en la variable *stories*, la cual viene definida por *const* ya que su valor será constante. Tiene un formato de lista donde cada posición presenta dos estructuras.

La primera estructura lo conforma la pregunta ("*question*") y la segunda la respuesta ("*answer*").

Como podemos observar en el ejemplo que se muestra, la respuesta puede tener diferentes opciones. Esto es debido aquí si en una pregunta la solución son tres términos debemos de contemplar todas las combinaciones que pueda aportar el usuario en distinto orden.

Una vez explicado el repositorio, empezaremos a explicar cómo funciona el *handler* que genera la primera pregunta.

```
const PrimeraPreguntaHandler = {
  canHandle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    return request.type === "IntentRequest" &&
      request.intent.name === "PrimeraPreguntaIntent"
  },
  handle(handlerInput) {
    const story = SiguientePregunta(handlerInput);
    const speechOutput = story.question;
    return handlerInput.responseBuilder
      .speak('Muy bien Javier, ahí va tu 1ª pregunta. ' +
speechOutput)
      .reprompt(speechOutput)
      .getResponse();
  }
}
```

Nuestro *canhandle()* será activado cuando el usuario diga “preguntas”, el cual será un *utterance* del *intent* “PrimeraPreguntaIntent”. Nuestro *handle* tendrá que hacer uso de la función llamada “SiguientePregunta”.

A continuación, explicaremos su funcionamiento.

```
function SiguientePregunta(handlerInput){
  const attributes = handlerInput.attributesManager.getSessionAttributes();
  var storiesDeck = [];

  if (attributes.counter == 20 || !attributes.counter ){
    storiesDeck = aleatorio(stories);
    attributes.storiesDeck = storiesDeck;
    attributes.counter = 0;
    attributes.correctCount = 0;
    attributes.wrongCount = 0;
  }
  else{
    storiesDeck = attributes.storiesDeck;
  }

  const story = storiesDeck[attributes.counter];
  attributes.lastQuestion = story;
  handlerInput.attributesManager.setSessionAttributes(attributes);
  return story;
}
```

Nuestra función trabaja de la siguiente manera. Primero define un administrador de atributos. Sirve para realizar un seguimiento de los datos a nivel de la sesión, como los contadores que podemos observar, número de respuestas correctas, también para guardar la pregunta realizada, etc.

Para ello usaremos el administrador de atributos proporcionado por el SDK para obtener y configurar los atributos de la sesión usando *getSessionAttributes*, y *setSessionAttributes*.

Una vez explicado esto, podemos volver a la función. La función se encarga de chequear si es la primera vez que ha sido llamada por ello pregunta por el contador de atributos *"attributes.counter"*. Si es la primera vez que ha sido usada esta función, quiere decir que viene llamada desde el handler de *"PrimeraPregunta"*.

Por lo tanto, no tiene los contadores definidos por lo que entraría en la primera parte del condicional. En ella se generarían todos los contadores. También sirve para resetear los contadores cuando el usuario ha realizado todas las preguntas, de esta forma el usuario podría realizar el cuestionario tantas veces como quisiera.

Aparte se generaría la variable *storiesDeck* donde estaría todo el repositorio de preguntas y respuestas, pero con un orden totalmente aleatorio gracias a la función “aleatorio”.

```
function aleatorio(arr) {  
  var ctr = arr.length, temp, index;  
  while (ctr > 0) {  
    index = Math.floor(Math.random() * ctr);  
    ctr--;  
    temp = arr[ctr];  
    arr[ctr] = arr[index];  
    arr[index] = temp;  
  }  
  return arr;  
}
```

Esto nos generaría que cada vez el usuario se enfrente a las preguntas pueda tener una experiencia totalmente distinta.

Por lo tanto, cada vez que llamemos a esta función nos devolverá la línea de dentro del repositorio dependiendo del valor del “attributes.counter”. En este caso, sería [0] por lo que nos devolvería en la variable *story*, con la primera posición.

Por lo que el *handler* ejecutaría la pregunta de la variable *story* como podemos observar en el siguiente cuadro.

```
const speechOutput = story.question;  
  return handlerInput.responseBuilder  
    .speak(speechOutput)  
    .reprompt(speechOutput)  
    .getResponse();  
}
```

Paso 6: Establecer el cuarto handler llamado RespuestaHandler

Una vez hemos generado la primera pregunta, nuestra skill esperaría una respuesta la cual sería sincronizada con nuestro nuevo *handler* llamado RespuestaHandler.

```
const RespuestaHandler = {
  canHandle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    return request.type === "IntentRequest" &&
    request.intent.name === "AnswerIntent" &&
    attributes.counter < attributes.storiesDeck.length - 1;
  },
  handle(handlerInput) {
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    const answerSlot =
handlerInput.requestEnvelope.request.intent.slots.answer.value;
    const result = checkrespuesta(handlerInput, answerSlot);
    const story = SiguientePregunta(handlerInput);
    const speechOutput = result.message + "Aquí tienes tu " + (attributes.counter + 1
) + "ª pregunta. " + story.question;

    attributes.lastResult = result.message;
    handlerInput.attributesManager.setSessionAttributes(attributes);

    return handlerInput.responseBuilder
      .speak(speechOutput)
      .reprompt(speechOutput)
      .getResponse();
  }
};
```

En este caso el *handler()* será activado con cualquier palabra que no active otro *handler*. para ello nos moveremos al *front-end*.

```
{
  "name": "AnswerIntent",
  "slots": [
    {
      "name": "answer",
      "type": "ANSWER_SLOT"
    }
  ],
  "samples": [
    "{answer}"
  ]
},
```

Como podemos observar, *front-end* nos recogerá el *utterance* y servirá para activar nuestro *handler*. Sin embargo, todos los ejemplos anteriores se activaban simplemente con un *intent* en este caso también tiene que cumplir la desigualdad que aparece. El contador de atributos debe ser menor que el contador de líneas disponibles de nuestro repositorio menos uno. Esta desigualdad está establecida para que nuestra skill sepa cuando es la última pregunta y de esta manera salte a otro *handler*. Como podemos observar dentro del *handler* se ejecuta la función que determina si la respuesta que ha dado el usuario es correcta o incorrecta. Esta función la hemos denominado “checkrespuesta”.

```
function checkrespuesta(handlerInput,answerSlot){
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    var status = "";
    var message ="";

    if (attributes.lastQuestion.answer.includes(answerSlot)){
        console.log("correct");
        message = "¡Perfecto! " + PrimeraLetraMayuscula(answerSlot) + " es
correcto. ";
        attributes.correctCount += 1;
        status =true;
    }
    else{
        console.log("wrong");
        message = "¡Mal! " + PrimeraLetraMayuscula(answerSlot) + " es
incorrecto. La respuesta correcta era " + attributes.lastQuestion.answer + ". ";
        attributes.wrongCount += 1;
        status = false;
    }
    attributes.counter += 1;
    handlerInput.attributesManager.setSessionAttributes(attributes);
    return {"status":status,"message":message};
}
```

Esta función como podemos observar nos genera el mensaje de si ha sido correcta o no la respuesta y también activa los contadores. También es capaz de aportar la respuesta correcta cuando el usuario se ha equivocado. Cuando el contador de atributos pase por esta función diecinueve veces que es el número de preguntas menos una, la desigualdad no se cumplirá y no volverá al *handler* llamado “Respuesta”.

Aparte también hemos generado una función para que la respuesta del usuario en el mensaje aparezca en mayúscula, mediante la función “PrimeraLetraMayuscula”.

```
function PrimeraLetraMayuscula(string) {  
  return string.charAt(0).toUpperCase() + string.slice(1);  
}
```

Continuando con el *handler*, este se encarga de guardar la información de la función de “checkrespuesta” en la variable *result*. Acciona otra vez la función “SiguientePregunta”, para que le genere la información de la siguiente pregunta. Y finalmente comunica el resultado de la respuesta y genera la nueva pregunta.

Paso 7: Establecer el quinto handler llamado ResultadoFinalHandler

En este *handler()* se encargará de revisar la última respuesta y generar la puntuación final.

```
const ResultadoFinalHandler = {  
  canHandle(handlerInput) {  
    const request = handlerInput.requestEnvelope.request;  
    const attributes = handlerInput.attributesManager.getSessionAttributes();  
    return request.type === "IntentRequest" &&  
      request.intent.name === "AnswerIntent" &&  
      attributes.counter == attributes.storiesDeck.length - 1;  
  },  
  handle(handlerInput) {  
    const attributes = handlerInput.attributesManager.getSessionAttributes();  
    const answerSlot = handlerInput.requestEnvelope.request.intent.slots.answer.value;  
    const result = checkrespuesta(handlerInput, answerSlot);  
  
    attributes.lastResult = result.message;  
    handlerInput.attributesManager.setSessionAttributes(attributes);  
    return handlerInput.responseBuilder  
      .speak(attributes.lastResult + " Hasta aquí las preguntas. Tu puntuación  
final ha sido " + attributes.correctCount + ' aciertos de ' + (attributes.counter) + ' preguntas. Pero  
recuerda Javier, la clave del éxito está en la constancia. Ya hemos terminado con el cuestionario,  
¿qué prefieres que te la vuelva a preguntar o que te explique la lección?' )  
      .reprompt(attributes.lastResult + " Hasta aquí las preguntas. Tu  
puntuación final ha sido " + attributes.correctCount + ' aciertos de ' + (attributes.counter) + '  
preguntas. Pero recuerda Javier, la clave del éxito está en la constancia. Ya hemos terminado con el  
cuestionario, ¿qué prefieres que te la vuelva a preguntar o que te explique la lección?' )  
      .getResponse();  
  }  
};
```

Este *handler* se accionará cuando se genere la última pregunta del usuario. Cumpliendo de esta manera con la igualdad que podemos observar, donde el contador de atributos es igual a la longitud del repositorio menos uno. En este caso, realizará las mismas funciones es decir revisará la respuesta del usuario y finalmente reportará los valores de los contadores de respuestas correctas y del contador de atributos el cual será igual al número de preguntas.

Paso 8: Establecer el sexto handler llamado HelpIntentHandler

Este *handler* se activará cuando el usuario diga “ayuda”. En ese momento el *handler* lanzará el mensaje donde explica las funcionalidades de la skill. Y se activa cuando el usuario dice “ayuda”.

```
const HelpIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) ===
    'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) ===
    'AMAZON.HelpIntent';
  },
  handle(handlerInput) {
    const speakOutput = AYUDA_MENSAJE;

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
};
```

Y recitará el siguiente mensaje.

```
const AYUDA_MENSAJE = "Hola Javier, veo que necesitas ayuda si quieres ponerte a estudiar la lección de geografía, dí lección. Y si quieres que te la pregunte, dí preguntas";
```

Paso 9: Establecer el séptimo handler llamado CancelAndStopIntentHandler

En este *handler* definiremos cómo el usuario puede cancelar la skill. Este *handler* será activado cuando el usuario diga “cancela” o “para”.

```
const CancelAndStopIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) ===
    'IntentRequest'
      && (Alexa.getIntentName(handlerInput.requestEnvelope) ===
    'AMAZON.CancelIntent'
      || Alexa.getIntentName(handlerInput.requestEnvelope) ===
    'AMAZON.StopIntent');
  },
  handle(handlerInput) {
    const speakOutput = 'Perfecto Javier. Espero que repasemos esta lección en
    otro momento. Y recuerda, cuanto más duramente trabajo, más suerte tengo.';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .getResponse();
  }
};
```

En ese momento, Alexa se despedirá como podemos observar en el cuadro anterior. Si nos fijamos más detenidamente, veremos como no está la estructura de `.reprompt(speakOutput)` por lo que Alexa no esperará ninguna respuesta del usuario y cerrará la skill.

4.4 Pruebas y evaluación

En este punto testaremos todas las funciones de la skill que hemos desarrollado. Para ello haremos uso del apartado de test que se encuentra dentro de la plataforma de desarrollo.

Prueba 1: Activación de la skill “Estudia conmigo”

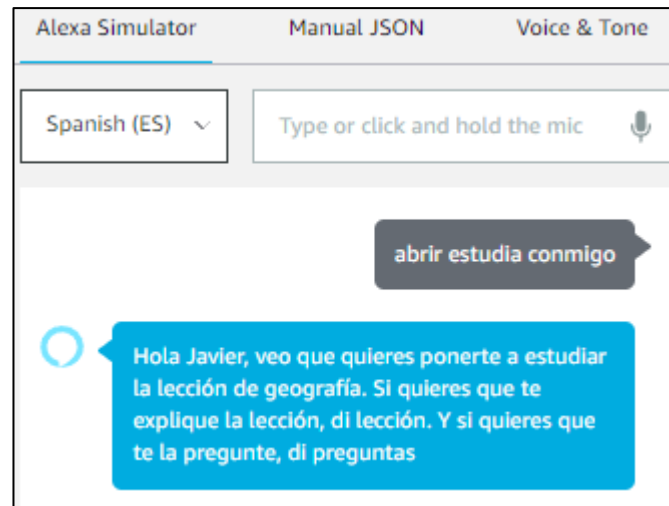


Figura 45. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. La skill es capaz de activarse mediante las palabras de activación “abrir estudia conmigo”.

Prueba 2: El usuario quiere recibir la lección

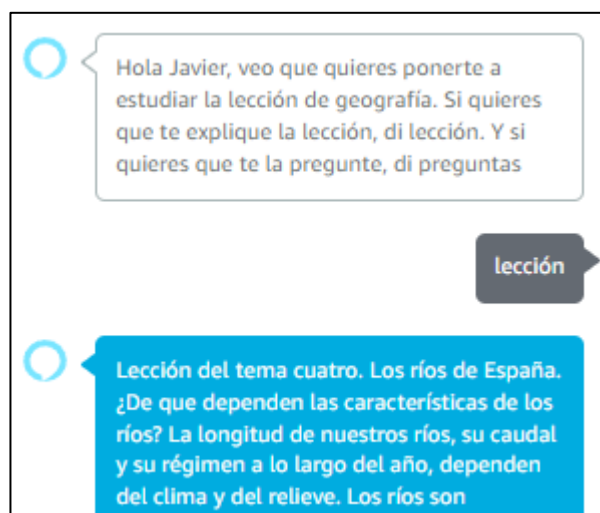


Figura 46. Recorte de la plataforma de desarrollo de Alexa.

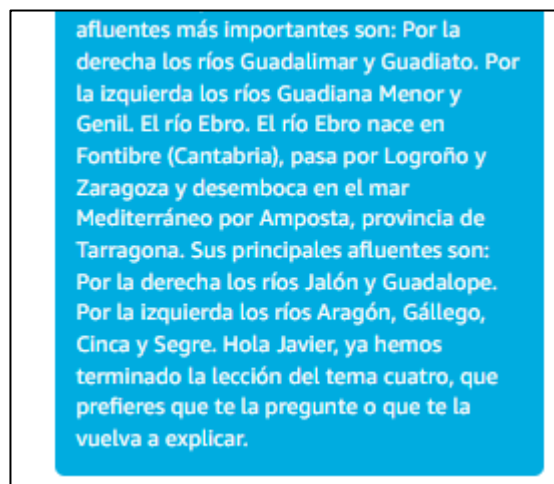


Figura 47. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. La skill es capaz de interpretar que el usuario quiere entrar en el escenario uno y por tanto recibir la lección.

Prueba 3: El usuario quiere volver a recibir la lección

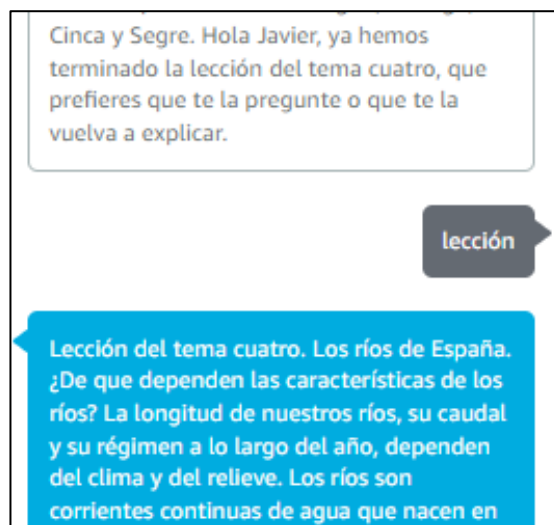


Figura 48. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. Una vez el usuario ha recibido la lección la skill proporciona la posibilidad de recibir la lección tantas veces como quiera el usuario.

Prueba 4: El usuario quiere que le pregunten la lección

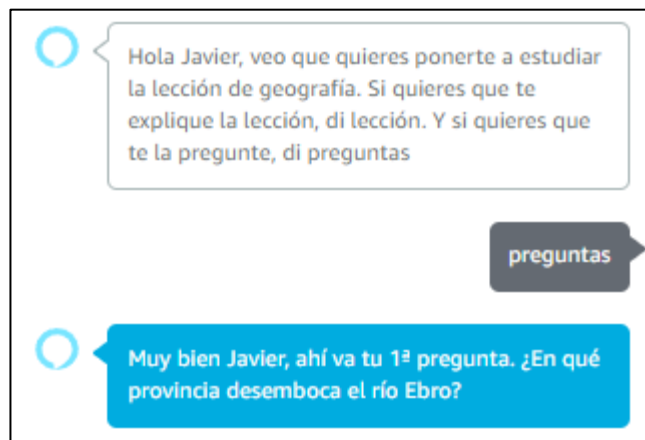


Figura 49. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. La skill es capaz de interpretar que el usuario quiere entrar en el escenario dos.

Prueba 5: El usuario responde a las preguntas y Alexa es capaz de verificar el resultado

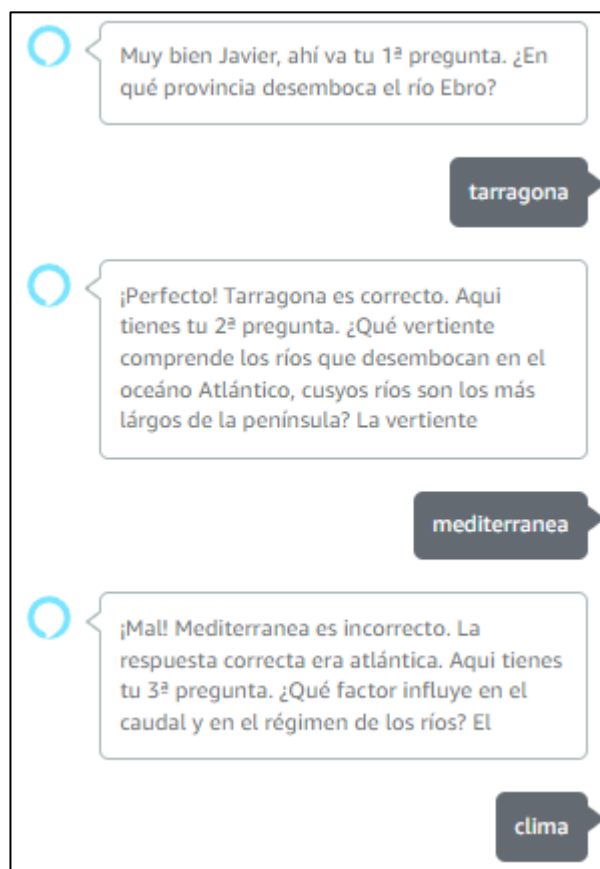


Figura 50. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. La skill es capaz de verificar la respuesta, proporcionándole el resultado y generándole una nueva pregunta. Aparte es capaz de proporcionarle cual era la respuesta correcta cuando se equivoca.

Prueba 6: Proporcionar el total de preguntas acertadas al usuario

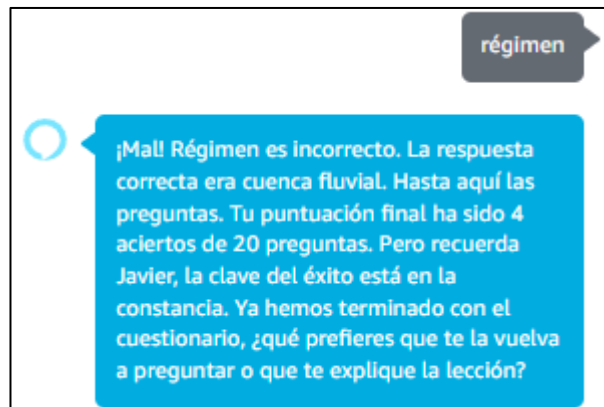


Figura 51. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. La skill es capaz de ir contando el número de aciertos y al final transmitírselo al usuario.

Prueba 7: El usuario se siente perdido y quiere accionar un mensaje de ayuda

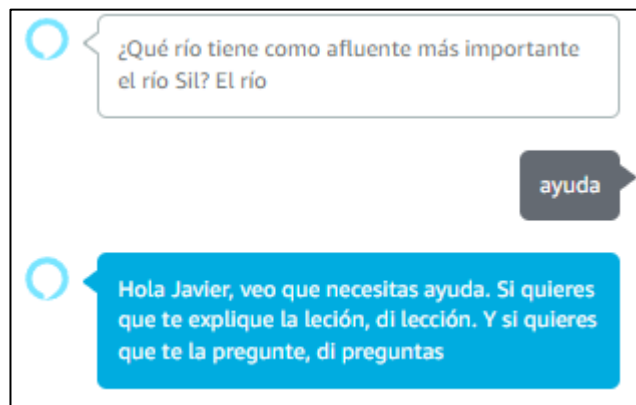


Figura 52. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. La skill es capaz de responder ante la consulta "ayuda".

Prueba 8: El usuario quiere cerrar la aplicación

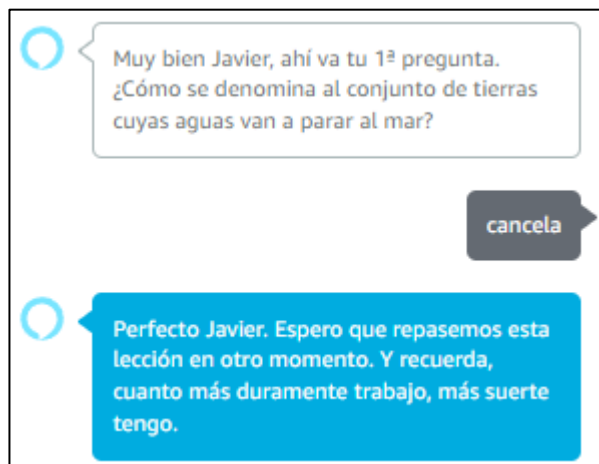


Figura 53. Recorte de la plataforma de desarrollo de Alexa.

Evaluación: Satisfactoria. La skill es capaz de despedirse y cerrarse cuando el usuario lo solicita.

Como hemos podido observar, todas las pruebas han sido satisfactorias. Hemos cumplido con todas las funcionalidades que habíamos preestablecido antes de crear nuestra skill. Por lo que podemos concluir afirmando que la evaluación ha sido totalmente exitosa.

4.5 Trabajos futuros

En este apartado hablaremos de los posibles futuros trabajos a raíz de este proyecto. Para ello nos centraremos en aportar mejoras a la skill desarrollada.

La primera mejora sería la de aumentar los utterances de cada intent, donde conseguiríamos que el usuario se pudiera comunicar con Alexa de una manera más articulada y natural. De esta manera la experiencia de uso mejoraría en un porcentaje considerable.

La segunda mejora sería crear un tercer escenario, donde el usuario pudiera preguntar por conceptos específicos y no tener que escuchar la lección entera. Esta actualización enriquecería la skill ya que aumentaríamos la productividad del usuario.

La tercera mejora sería al contenido, esta skill sólo contiene una lección por lo que la diversidad de contenido es nula. Mejoraría la skill si dependiendo de la lección que el usuario quisiera estudiar se actualizase tanto el escenario uno como el dos.

Para ello conectaría la función Lambda con una base de datos relacionar para que se pudiera nutrir de los contenidos educativos, consiguiendo una liberación del código y ganando en escalabilidad.

Elegiría una base de datos relacionar ya que solo haría consultas bastantes generales y no necesitaría sustituir ningún valor de la base de datos por lo que sería una buena elección.

4.6 Anexos

4.6.1 Front-end

```
{
  "interactionModel": {
    "languageModel": {
      "invocationName": "estudia conmigo",
      "intents": [
        {
          "name": "AMAZON.FallbackIntent",
          "samples": []
        },
        {
          "name": "AMAZON.HelpIntent",
          "samples": []
        },
        {
          "name": "PrimeraPreguntaIntent",
          "slots": [],
          "samples": [
            "preguntas"
          ]
        },
        {
          "name": "LeccionIntent",
          "slots": [],
          "samples": [
            "lección"
          ]
        },
        {
          "name": "AnswerIntent",
          "slots": [
            {
              "name": "answer",
              "type": "ANSWER_SLOT"
            }
          ],
          "samples": [
            "{answer}"
          ]
        },
        {
          "name": "AMAZON.CancelIntent",
          "samples": [
            "cancela"
          ]
        }
      ]
    }
  }
}
```

```

}
],
"types": [
{
"name": "ANSWER_SLOT",
"values": [
{
"name": {
"value": "curso caudal y régimen"
}
},
{
"name": {
"value": "caudal curso y régimen"
}
},
{
"name": {
"value": "régimen curso y caudal"
}
},
{
"name": {
"value": "régimen caudal y curso"
}
},
{
"name": {
"value": "caudal régimen y curso"
}
},
{
"name": {
"value": "curso régimen y caudal"
}
},
{
"name": {
"value": "zamora y soria"
}
},
{
"name": {
"value": "soria y zamora"
}
},
{
"name": {
"value": "pisuerga y esla"
}
},
],

```

```
{
  "name": {
    "value": "esla y pisuerga"
  }
},
{
  "name": {
    "value": "tajo"
  }
},
{
  "name": {
    "value": "duero"
  }
},
{
  "name": {
    "value": "miño"
  }
},
{
  "name": {
    "value": "urbión"
  }
},
{
  "name": {
    "value": "vertiente"
  }
},
{
  "name": {
    "value": "atlántica"
  }
},
{
  "name": {
    "value": "caudal"
  }
},
{
  "name": {
    "value": "régimen"
  }
},
{
  "name": {
    "value": "curso"
  }
},
{
  "name": {
    "value": "clima"
  }
}
```

```
},  
  {  
    "name": {  
      "value": "albarracín"  
    }  
  },  
  {  
    "name": {  
      "value": "ruidera"  
    }  
  },  
  {  
    "name": {  
      "value": "relieve"  
    }  
  },  
  {  
    "name": {  
      "value": "amposta"  
    }  
  },  
  {  
    "name": {  
      "value": "aragón"  
    }  
  },  
  {  
    "name": {  
      "value": "cinca"  
    }  
  },  
  {  
    "name": {  
      "value": "gállego"  
    }  
  },  
  {  
    "name": {  
      "value": "segre"  
    }  
  },  
  {  
    "name": {  
      "value": "guadalope"  
    }  
  },  
  {  
    "name": {  
      "value": "cuenca fluvial"  
    }  
  },  
}
```



```
{  
  "name": {  
    "value": "cazorla"  
  }  
}  
]  
}  
]  
}  
]  
}  
}
```

4.6.2 Back-end

```
const Alexa = require('ask-sdk-core');

const BIENVENIDA_MENSAJE = "Hola Javier, veo que quieres ponerte a estudiar la lección de geografía.\n    Si quieres que te explique la lección, di lección. Y si quieres que te la pregunte, di preguntas";
const AYUDA_MENSAJE = "Hola Javier, veo que necesitas ayuda. Si quieres que te explique la lección, di lección.\n    Y si quieres que te la pregunte, di preguntas";

const LaunchRequestHandler = {
    canHandle(handlerInput) {
        const request = handlerInput.requestEnvelope.request;
        return request.type === "LaunchRequest";
    },
    handle(handlerInput) {
        const speechOutput = BIENVENIDA_MENSAJE;
        const repromptSpeechOutput = AYUDA_MENSAJE;
        return handlerInput.responseBuilder
            .speak(speechOutput)
            .reprompt(repromptSpeechOutput)
            .getResponse();
    }
};

const LeccionHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'LeccionIntent';
    },
    handle(handlerInput) {
        const speakOutput = 'Lección del tema cuatro. Los ríos de España. ¿De que dependen las características de los ríos? La longitud de nuestros ríos, su caudal y su régimen a lo largo del año, \n    dependen del clima y del relieve. Los ríos son corrientes continuas de agua que nacen en las montañas y desembocan en el mar, en un lago o en otro río. En un río podemos distinguir tres elementos:\n    el curso, el caudal y el régimen. El curso. Es el recorrido que realiza un río desde su nacimiento hasta su desembocadura. Se distinguen tres tramos: curso alto, curso medio y curso bajo. El caudal.\n    Es la cantidad de agua que lleva un río. El régimen. Es la variación que sufre el caudal de un río a lo largo de año. El régimen es regular, cuando la variación de caudal es pequeña; y es irregular, \n    cuando la variación de caudal es grande. Las vertientes de España. Un río principal es un río que desemboca directamente en el mar. Un afluente es un río que va a parar a otro río. Una cuenca fluvial \n    es el territorio recorrido por un río y sus afluentes. Una vertiente hidrográfica es el conjunto de tierras cuyas aguas van a parar al mismo mar. Los ríos españoles desembocan en tres vertientes: Cantábrica, \n    Atlántica y Mediterránea. La vertiente cantábrica. Está al norte de España. La vertiente mediterránea. Situada al este y sur de España. La vertiente atlántica. Está al oeste de España. A continuación, hablaremos \n';
    }
};
```

de los ríos más importantes de España. El río Miño. El río Miño nace en Fuente Miña , provincia de Lugo, pasa por Lugo y Orense y desemboca en el océano Atlántico por La Guardia, provincia de Pontevedra. Sirve de \

frontera entre España y Portugal. Su afluente más importante es el Sil. El río Duero. El río Duero nace en los Picos de Urbión, provincia de Soria, pasa por Soria y Zamora y desemboca en el océano Atlántico por Oporto\

(Portugal). Sus afluentes más importantes son: Por la derecha los ríos Pisuerga y Esla. Por la izquierda los ríos Eresma, Adaja y Tormes. El río Tago. El río Tago nace en la Sierra de Albarracín, provincia de Teruel, \

pasa por Toledo y Aranjuez y desemboca en el océano Atlántico por Lisboa (Portugal). Sus afluentes más importantes son: Por la derecha los ríos Jarama, Alberche, Tiétar y Alagón. Por la izquierda los ríos Almonte y Salor. \

El río Guadiana. El río Guadiana nace en las lagunas de Ruidera, provincia de Ciudad Real, pasa por Mérida y Badajoz y desemboca en el océano Atlántico por Ayamonte, provincia de Huelva. Sus afluentes principales son: \

Por la derecha los ríos Záncara y Cigüela. Por la izquierda los ríos Jabalón, Zújar; Matachel y Ardila. El río Guadalquivir. El río Guadalquivir nace en la sierra de Cazorla, provincia de Jaén, pasa por Córdoba y Sevilla \

y desemboca en el océano Atlántico por Sanlúcar de Barrameda, provincia de Cádiz. Sus afluentes más importantes son: Por la derecha los ríos Guadalimar y Guadiato. Por la izquierda los ríos Guadiana Menor y Genil. \

El río Ebro. El río Ebro nace en Fontibre (Cantabria), pasa por Logroño y Zaragoza y desemboca en el mar Mediterráneo por Amposta, provincia de Tarragona. Sus principales afluentes son: Por la derecha los ríos \

Jalón y Guadalope. Por la izquierda los ríos Aragón, Gállego, Cinca y Segre. Hola Javier, ya hemos terminado la lección del tema cuatro, que prefieres que te la pregunte o que te la vuelva a explicar.';

```
return handlerInput.responseBuilder
    .speak(speakOutput)
    .reprompt(speakOutput)
    .getResponse();
```

```
}
```

```
};
```

```
const PrimeraPreguntaHandler = {
  canHandle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    return request.type === "IntentRequest" &&
    request.intent.name === "PrimeraPreguntaIntent"
  },
  handle(handlerInput) {
    const story = SiguientePregunta(handlerInput);
    const speechOutput = story.question;
    return handlerInput.responseBuilder
      .speak('Muy bien Javier, ahí va tu 1ª pregunta. ' + speechOutput)
      .reprompt(speechOutput)
      .getResponse();
  }
};
```

```

const RespuestaHandler = {
  canHandle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    return request.type === "IntentRequest" &&
    request.intent.name === "AnswerIntent" &&
    attributes.counter < attributes.storiesDeck.length - 1;
  },
  handle(handlerInput) {
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    const answerSlot =
handlerInput.requestEnvelope.request.intent.slots.answer.value;
    const result = checkrespuesta(handlerInput, answerSlot);
    const story = SiguientePregunta(handlerInput);
    const speechOutput = result.message + "Aquí tienes tu " + (attributes.counter + 1 )
+ "ª pregunta. " + story.question;

    attributes.lastResult = result.message;
    handlerInput.attributesManager.setSessionAttributes(attributes);

    return handlerInput.responseBuilder
      .speak(speechOutput)
      .reprompt(speechOutput)
      .getResponse();
  }
};

const ResultadoFinalHandler = {
  canHandle(handlerInput) {
    const request = handlerInput.requestEnvelope.request;
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    return request.type === "IntentRequest" &&
    request.intent.name === "AnswerIntent" &&
    attributes.counter == attributes.storiesDeck.length - 1;
  },
  handle(handlerInput) {
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    const answerSlot = handlerInput.requestEnvelope.request.intent.slots.answer.value;
    const result = checkrespuesta(handlerInput, answerSlot);

    attributes.lastResult = result.message;
    handlerInput.attributesManager.setSessionAttributes(attributes);
    return handlerInput.responseBuilder
      .speak(attributes.lastResult + " Hasta aquí las preguntas. Tu puntuación
final ha sido " + attributes.correctCount + ' aciertos de ' + (attributes.counter) + ' preguntas. Pero
recuerda Javier, la clave del éxito está en la constancia. Ya hemos terminado con el cuestionario,
¿qué prefieres que te la vuelva a preguntar o que te explique la lección?' )
      .reprompt(attributes.lastResult + " Hasta aquí las preguntas. Tu
puntuación final ha sido " + attributes.correctCount + ' aciertos de ' + (attributes.counter) + '
preguntas. Pero recuerda Javier, la clave del éxito está en la constancia. Ya hemos terminado con el
cuestionario, ¿qué prefieres que te la vuelva a preguntar o que te explique la lección?' )
      .getResponse();
  }
};

```

```

function SiguientePregunta(handlerInput){
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    var storiesDeck = [];

    if (attributes.counter == 20 || !attributes.counter ){
        storiesDeck = aleatorio(stories);
        attributes.storiesDeck = storiesDeck;
        attributes.counter = 0;
        attributes.correctCount = 0;
        attributes.wrongCount = 0;
    }
    else{
        storiesDeck = attributes.storiesDeck;
    }

    const story = storiesDeck[attributes.counter];
    attributes.lastQuestion = story;
    handlerInput.attributesManager.setSessionAttributes(attributes);
    return story;
}
function checkrespuesta(handlerInput,answerSlot){
    const attributes = handlerInput.attributesManager.getSessionAttributes();
    var status = "";
    var message = "";

    if (attributes.lastQuestion.answer.includes(answerSlot)){
        console.log("correct");
        message = "¡Perfecto! " + PrimeraLetraMayuscula(answerSlot) + " es correcto. ";
        attributes.correctCount += 1;
        status =true;
    }
    else{
        console.log("wrong");
        message = "¡Mal! " + PrimeraLetraMayuscula(answerSlot) + " es incorrecto. La
respuesta correcta era " + attributes.lastQuestion.answer + ". ";
        attributes.wrongCount += 1;
        status = false;
    }
    attributes.counter += 1;
    handlerInput.attributesManager.setSessionAttributes(attributes);
    return {"status":status,"message":message};
}
function aleatorio(arr) {
    var ctr = arr.length, temp, index;
    while (ctr > 0) {
        index = Math.floor(Math.random() * ctr);
        ctr--;
        temp = arr[ctr];
        arr[ctr] = arr[index];
        arr[index] = temp;
    }
    return arr;
}

```

```

}

function PrimeraLetraMayuscula(string) {
    return string.charAt(0).toUpperCase() + string.slice(1);
}

const stories = [
    {
        "question": "¿Qué tres elementos podemos distinguir en un río?", "answer": ['curso
caudal y régimen', 'caudal curso y régimen', 'curso régimen y caudal', 'caudal régimen y
curso', 'régimen caudal y curso', 'régimen curso y caudal']
    },
    {
        "question": "¿Por qué provincias pasa el río Duero?", "answer": ['zamora y
soria', 'soria y zamora']
    },
    {
        "question": "¿Qué río tiene como afluentes por la derecha los ríos Jarama,
Alberche, Tiétar y Alargón? El río", "answer": 'tajo'
    },
    {
        "question": "¿Dime un afluente del río Ebro? El río", "answer": ['jalón', 'guadalope',
'aragón', 'gállego', 'cinca', 'segre']
    },
    {
        "question": "¿En qué picos nace el río Duero? En los Picos de", "answer": 'urbión'
    },
    {
        "question": "¿En qué sierra nace el río Guadalquivir? En la sierra de
", "answer": 'cazorla'
    },
    {
        "question": "¿Qué río tiene como afluente más importante el río Sil? El río
", "answer": 'miño'
    },
    {
        "question": "¿Qué vertiente comprende los ríos que desembocan en el océano
Atlántico, cuyos ríos son los más largos de la península? La vertiente ", "answer": 'atlántica'
    },
    {
        "question": "¿Cómo se denomina al territorio recorrido por un río y sus afluentes?
", "answer": 'cuenca fluvial'
    },
    {
        "question": "¿Cómo se denomina al conjunto de tierras cuyas aguas van a parar al
mar? ", "answer": 'vertiente'
    },
    {
        "question": "¿Cómo se denomina a la cantidad de agua que lleva un río?
", "answer": 'caudal'
    },

```

```
{
    "question": "¿Cómo se denomina al recorrido que realiza un río desde su
nacimiento hasta su desembocadura? ", "answer": 'curso'
},
{
    "question": "¿Cómo se denomina a la variación que sufre el caudal de un río a lo
largo del año? ", "answer": 'régimen'
},
{
    "question": "¿En qué provincia desemboca el río Ebro? ", "answer": 'tarragona'
},
{
    "question": "¿Qué factor influye en la longitud y en la velocidad de las aguas de los
ríos? ", "answer": 'relieve'
},
{
    "question": "¿Dónde desemboca el río Ebro? En ", "answer": 'amposta'
},
{
    "question": "¿Dónde nace el río Guadiana? En las lagunas de ", "answer": 'ruidera'
},
{
    "question": "¿Dónde nace el río Tajo? En la sierra de ", "answer": 'albarracín'
},
{
    "question": "¿Qué factor influye en el caudal y en el régimen de los ríos? El
", "answer": 'clima'
},
{
    "question": "¿Qué río tiene como afluentes por la derecha los ríos Pisuerga y Esla?
El río ", "answer": 'duero'
}
};

const HelpIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.HelpIntent';
    },
    handle(handlerInput) {
        const speakOutput = AYUDA_MENSAJE;

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .reprompt(speakOutput)
            .getResponse();
    }
};
```

```
const CancelAndStopIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && (Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.CancelIntent'
        || Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.StopIntent');
  },
  handle(handlerInput) {
    const speakOutput = 'Perfecto Javier. Espero que repasemos esta lección en otro momento. Y recuerda, cuanto más duramente trabajo, más suerte tengo.';

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .getResponse();
  }
};

const skillBuilder = Alexa.SkillBuilders.custom();
exports.handler = skillBuilder
  .addRequestHandlers(
    CancelAndStopIntentHandler,
    HelpIntentHandler,
    LaunchRequestHandler,
    LeccionHandler,
    PrimeraPreguntaHandler,
    RespuestaHandler,
    ResultadoFinalHandler
  )
  .lambda();
```


CONCLUSIONES

Podemos concluir afirmando que el desarrollo de este proyecto ha sido totalmente satisfactorio. Ya que hemos conseguido alcanzar nuestro principal objetivo el cual se basaba en el desarrollo de una skill educativa que tuviera una interacción totalmente dinámica cuya finalidad fuera aumentar la productividad del usuario. De todos modos, se repasarán los diferentes objetivos alcanzados a lo largo de este proyecto.

- Durante todo el proyecto se ha trabajado recorriendo por todas las fases de desarrollo de un proyecto de estas dimensiones (definición del problema, diseño de la solución y su posterior evaluación).
- Para llevar a cabo el desarrollo de esta skill se ha utilizado el framework de Amazon Alexa (ASK y AWS Lambda) y se han creado ficheros.js en JavaScript. Esto ha implicado el estudio previo y continuo de dichas herramientas, que ha conllevado una considerable inversión de tiempo de aprendizaje, ya que no se había trabajado con ellas anteriormente. Por tanto, la implementación de la idea que se había planteado al inicio del proyecto ha sido la parte más complicada y a la que se le ha dedicado más tiempo.
- Como resultado, se ha obtenido una aplicación en el ámbito de la educación. Consiguiendo de esta manera una skill con una interfaz rápida y dinámica la cual tendrá un impacto más que positivo ante los ojos de los usuarios. También hemos conseguido que la aplicación fuese una fuente de información, ya que el asistente virtual es capaz de recitar la lección que el estudiante se tiene que aprender como habíamos preestablecido. Finalmente, también hemos conseguido desarrollar el core de nuestra aplicación el cual se basa en el cuestionario en el cual el asistente le pregunta la lección de una manera interactiva. Además, hemos conseguido darle la opción al usuario de repetir el cuestionario tantas veces como quiera, donde cada experiencia es totalmente distinta a la anterior ya que el orden de las preguntas va cambiando aleatoriamente.
- Por último, cabe mencionar que se han completado los objetivos planteados al inicio del proyecto, pero que evidentemente aún queda espacio para la mejora como se ha establecido en el apartado de “Trabajos futuros”.

BIBLIOGRAFÍA

Asistentes virtuales. Antecedentes y evolución de los asistentes virtuales

[1] SEIDOR, *¿qué es un asistente virtual y cómo puede ayudar a las empresas?*

URL: <https://www.seidor.es/content/seidorweb/es/blog/asistente-virtual-empresas.html>

[Noviembre 2021]

[2] YELLOW.AI, *the evolution of intelligent virtual assistants.*

URL: <https://blog.yellow.ai/resources/articles/the-evolution-of-virtual-assistants>

[Noviembre 2021]

[3] THINKBIG, *la evolución de los asistentes a virtuales en 8 años y para qué los utilizan los usuarios.*

URL: <https://blogthinkbig.com/asistentes-virtuales-usuarios>

[Noviembre 2021]

[4] INVESTOPEDIA, *chatbot definition.*

URL: <https://www.investopedia.com/terms/c/chatbot.asp>

[Diciembre 2021]

[5] COMPUTER HOPE, *MS-DOS.*

URL: <https://www.computerhope.com/jargon/m/msdos.htm>

[Diciembre 2021]

[6] APPLE, *antes de que digas nada, SIRI hace más que nunca.*

URL: <https://www.apple.com/es/siri/>

[Diciembre 2021]

[7] USWITCH, *what are smartphones?*

URL: <https://www.uswitch.com/mobiles/guides/what-are-smartphones/>

[Diciembre 2021]

Asistentes virtuales. Asistentes virtuales en la actualidad.

[8] SILICON PLACE, *¿cuál es el mejor asistente de voz?*

URL: <https://www.siliconplace.es/altavoces-inteligentes/siri-vs-google-vs-alexa-vs-cortana-cual-es-el-mejor-asistente-de-voz/>

[Noviembre 2021]

[9] SCIENCE DIRECT, *historia, tecnología y aplicaciones*.

URL: <https://www.sciencedirect.com/science/article/pii/S2666827020300062>

[Noviembre 2021]

[10] ELCOFIDENCIAL, *el escándalo del robo de datos en Facebook llega a Washington: Zuckerberg testificará*.

URL: https://www.elconfidencial.com/tecnologia/2018-04-04/facebook-cambridge-analytica-mark-zuckerberg_1544818/

Asistentes virtuales. Campos de aplicación de los asistentes virtuales

[11] IBM, *what is natural language processing?*

URL: <https://www.ibm.com/cloud/learn/natural-language-processing>

[Diciembre 2021]

Asistentes virtuales. Tecnología y digitalización del sector educativo

[12] ELECONOMISTA, *la tecnología y la digitalización del sistema educativo*.

URL: <https://www.eleconomista.es/ecoaula/noticias/11015024/01/21/La-tecnologia-y-la-digitalizacion-del-sistema-educativo-principales-retos-del-sistema-educativo-de-nuestro-pais.html>

[Diciembre 2021]

[13] WIKIPEDIA, *asistente virtual inteligente*.

URL: https://ca.wikipedia.org/wiki/Assistent_personal_intel%C2%B7ligent

[Diciembre 2021]

[14] SCIELO, *enseñanza y aprendizaje virtual en contexto de pandemia. Experiencias y vivencias de docentes y estudiantes de la Facultad de Psicología en el primer semestre del año 2020*.

URL: http://www.scielo.edu.uy/scielo.php?pid=S230101262020000200150&script=sci_arttext

[Noviembre 2021]

[15] HERA TECHNOLOGY, *tecnología para la salud mental*.

URL: <https://www.heratechnology.org/>

[Diciembre 2021]

Asistentes virtuales. Alexa y sus dispositivos físicos.

[16] TOWARDS DATASCIENCE, *how amazon Alexa works? Your guide to Natural Language Processing*.

URL: <https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3>

[Noviembre 2021]

[17] POCKET, *what is uber and how does it work?*

URL: <https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3>

[Noviembre 2021]

Arquitectura del asistente virtual Alexa

[18] DEVELOPER AMAZON, *what is automatic speech recognition?*

URL: <https://developer.amazon.com/es-ES/alexa/alexa-skills-kit/asr>

[Noviembre 2021]

[19] AMAZON, *acerca de Amazon*.

URL: <https://www.amazon.es/Acerca-Amazon-Descubre-Nuestra-Empresa-Nuestra-Tecnologia/b?ie=UTF8&node=1323175031>

[Noviembre 2021]

[20] NEXTU, *JSON, ¿qué es y para que sirve?*

URL: <https://www.amazon.es/Acerca-Amazon-Descubre-Nuestra-Empresa-Nuestra-Tecnologia/b?ie=UTF8&node=1323175031>

[Noviembre 2021]

[21] IONOS, *programar skills de Alexa es muy fácil: te contamos cómo*.

URL: <https://www.ionos.es/digitalguide/online-marketing/vender-en-internet/programar-skills-de-alexa/>

[Noviembre 2021]

[22] DEVELOPER AMAZON, *what is Alexa Skills Kit?*

URL: <https://developer.amazon.com/en-US/docs/alexa/ask-overviews/what-is-the-alexa-skills-kit.html>

[Noviembre 2021]

[23] JAVA, *what is Java technology and why do I need it?*

URL: https://www.java.com/en/download/help/whatis_java.html

[Diciembre 2021]

[24] DOCS MICROSOFT, *what is PowerShell?*

URL: <https://docs.microsoft.com/enus/powershell/scripting/overview?view=powershell-7.2>

[Diciembre 2021]

[25] CLOUD FLARE, *what is the cloud?*

URL: <https://www.cloudflare.com/es-es/learning/cloud/what-is-the-cloud/>

[Diciembre 2021]

[26] SERVER WATCH, *what Is a Server and What Do Servers Do?*

URL: <https://www.serverwatch.com/guides/what-is-a-server/>

[Diciembre 2021]

BIBLIOGRAFÍA FIGURAS

Asistentes virtuales. Antecedentes y evolución de los asistentes virtuales.

[1] Figura 1, *esquema del Test de Turing.*

URL: <https://www.researchgate.net/figure/Figura-21-Esquema-del-Test-de-Turing>

[Noviembre 2021]

[2] Figura 2, *ilustración del logotipo de Siri.*

URL: <https://oaxaka.net/como-evitar-que-siri-se-abra-cuando-mantienes-presionado-el-boton-lateral-de-inicio-en-el-iphone>

[Noviembre 2021]

[3] Figura 3, *ilustración del logotipo de Google Now.*

URL: <https://www.tuexpertomovil.com/2016/03/21/google-now-como-decirle-a-google-que-medios-no-te-interesan/>

[Noviembre 2021]

[4] Figura 4, *ilustración del logotipo de Alexa.*

URL: <https://www.businessinsider.es/todo-puede-hacer-alexa-740195>

[Noviembre 2021]

[5] Figura 5, *ilustración del logotipo de Cortana.*

URL: <https://www.xataka.com/aplicaciones/cortana-empequenece-desaparece-lanzador-android-reduce-su-protagonismo-windows-10>

[Noviembre 2021]

Asistentes virtuales. Asistentes virtuales en la actualidad.

[6] Figura 6, *diagrama de las tareas más realizadas por asistentes virtuales.*

URL: <https://www.siliconplace.es/altavoces-inteligentes/siri-vs-google-vs-alexa-vs-cortana-cual-es-el-mejor-asistente-de-voz/>

[Noviembre 2021]

[7] Figura 8, *chatbots: historia, tecnología y aplicaciones.*

URL: <https://www.sciencedirect.com/science/article/pii/S2666827020300062>

[Noviembre 2021]

Asistentes virtuales. Tecnología y digitalización del sector educativo

[8] Figura 9, *enseñanza y aprendizaje virtual en contexto de pandemia. Experiencias y vivencias de docentes y estudiantes de la Facultad de Psicología en el primer semestre del año 2020.*

URL: http://www.scielo.edu.uy/scielo.php?pid=S230101262020000200150&script=sci_arttext

[Noviembre 2021]

Alexa y sus dispositivos físicos.

[9] Figura 10, *¿qué es Alexa y qué puede hacer Amazon Echo?*

URL: <https://www.pocket-lint.com/es-es/hogar-inteligente/noticias/amazon/138846-que-es-alexa-como-funciona-y-que-puede-hacer-amazons-alexa>

[Noviembre 2021]

Arquitectura del asistente virtual Alexa

[10] Figura 15, *what is Alexa Skills Kit?*

URL: <https://developer.amazon.com/es-ES/docs/alexa/ask-overviews/what-is-the-alexa-skills-kit.html>

[Noviembre 2021]

[11] Figura 24, *AWS Lambda.*

URL: <https://aws.amazon.com/es/lambda/faqs/>

[Noviembre 2021]

[12] Figura 25, *what are the uses of JavaScript?*

URL: <https://www.javatpoint.com/what-are-the-uses-of-javascript>

[Noviembre 2021]