



Universidad
Rey Juan Carlos

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Curso Académico 2019/2020

Trabajo Fin de Grado

**APRENDIZAJE DEL ALGORITMO DE DIJKSTRA A TRAVÉS
DEL ASISTENTE VIRTUAL ALEXA**

Autor: Alberto Barragán Costo

Director: Maximiliano Paredes Velasco

A mis padres y hermano, por confiar siempre en mí. Por darme la oportunidad de estudiar la carrera con la que soñaba. Por todo el apoyo que día a día me han dado e incansablemente me dan. Por hacerme la vida más fácil. Por permitir cuantas locuras se me ocurren. Por luchar por mis estudios y por confiar cuando ni yo lo hacía. Por insistir tanto para la realización de este proyecto. Aquí está. .

A mi pareja, por aguantar tantas y tantas horas de confinamiento mientras realizaba este proyecto. Gracias por tu paciencia y por tu comprensión. Gracias por estar siempre a mi lado. Gracias por apoyarme en todo lo que hago. Gracias por hacer todo esto posible.

A mis compañeros de universidad. Sin ellos nada de esto sería posible. Sin su apoyo, la universidad y mi vida, serían totalmente distintas. Por esas tres estaciones de biblioteca y una de festivales. Mención especial a Álvaro, juntos desde la primera hasta la última clase, sin él la universidad habría sido mucho más difícil.

A mis abuelas. Por enterarme, justo antes de comenzar uno de los exámenes más difíciles de la carrera, que una de ellas había comenzado a ayudarme de otro modo. Por regalarme aquel día, una matrícula de honor en la tierra y una ayuda para siempre desde el cielo.

Resumen

En esta memoria se explicará cómo se ha realizado el desarrollo de una aplicación, que permite al usuario de la misma, aprender a resolver un problema mediante un algoritmo voraz, concretamente el problema de los caminos mínimos de un grafo.

Actualmente, la tecnología avanza a un ritmo altísimo, por lo que la educación, como cualquier otro ámbito de la vida, debe adaptarse a estos cambios lo más rápido posible. Para hacer esto posible, surgen proyectos como el descrito en las páginas posteriores. Proyectos capaces de darle a la educación una nueva manera de hacer las cosas y de enseñar a los alumnos a través de métodos novedosos.

En las siguientes páginas se describirá una aplicación, accesible a través de un dispositivo utilizado por el asistente virtual Alexa, capaz de ir enseñando a un usuario, ya sea técnico o no, cómo resolver el problema de los caminos mínimos. Dicha aplicación será capaz, tanto de enseñar mediante un ejemplo al usuario a cómo resolver dicho problema, como a poner a prueba al usuario mediante la realización de preguntas que tendrán que ser respondidas correctamente. La aplicación desarrollada para el asistente virtual Alexa será capaz de comunicarse con una aplicación desarrollada para Android como apoyo para la enseñanza de los conocimientos al alumno.

Para la realización del proyecto se ha utilizado NodeJs para desarrollar la parte servidor de la skill para Alexa, el lenguaje de programación Kotlin para desarrollar la aplicación Android y para comunicar ambas aplicaciones se ha utilizado una librería de Amazon Web Service, la cual utiliza Lambdas como funciones ejecutadas en un servidor en la nube, DynamoDB como herramienta de base de datos alojada en la nube y un API Gateway de Websocket para la disponibilidad de un WebSocket alojado también en la nube.

Índice

1 INTRODUCCIÓN	9
1.1 MOTIVACIÓN	9
1.2 OBJETIVOS	9
2 FUNDAMENTOS	11
2.1 AMAZON ALEXA	11
2.1.1 Skills	11
2.1.2 Intenciones	12
2.1.3 Alexa-Hosted Skill	13
2.1.4 Alexa Developer Console	13
2.1.5 Display	17
2.1.6 NodeJS	17
2.1.7 Git	18
2.1.8 Webstorm	19
2.2 ANDROID	19
2.2.1 Android Studio	20
2.2.2 Kotlin	20
2.3 AMAZON WEB SERVICE	21
2.3.1 Amazon API Gateway	21
2.3.2 AWS Lambda	22
2.3.3 AWS DynamoDB	22
3. DESCRIPCIÓN INFORMÁTICA.	24
3.1 METODOLOGÍA DE TRABAJO.	24
3.2 ANÁLISIS	25
3.2.1 ¿Por qué Alexa?	25
3.2.2 ¿Por qué Android?	27
3.2.3 ¿Por qué kotlin?	28
3.3 DISEÑO	29
3.3.1 Flujo de ejecución	29
3.3.2 Ejemplos de la interfaz	32
3.4 IMPLEMENTACIÓN	33
3.4.1 Interfaz de voz de usuario.	33
3.4.2 Estructura de directorios del backend	35
3.4.3 Implementación del backend.	36
3.4.5 Imagen mostrada por pantalla	38
3.4.6 Implementación de la aplicación Android	38
3.4.7 Simple-websockets-chat-app.	39
3.4.8 Conexión entre skill y aplicación.	40
4 PRUEBAS	42
4.1 PRUEBAS MANUALES	42
4.2 PRUEBAS CON USUARIOS REALES	44
5 CONCLUSIONES	47
5.1 CONCLUSIONES GENERALES	47
5.2 CONCLUSIONES PERSONALES.	48
5.3 TRABAJOS FUTUROS	50
6 BIBLIOGRAFÍA	51

Tabla de Ilustraciones

ILUSTRACIÓN 1. INVOCACIÓN DE UNA SKILL.	12
ILUSTRACIÓN 2. ALEXA DEVELOPER CONSOLE. CREATE SKILL.	14
ILUSTRACIÓN 3. ALEXA DEVELOPER CONSOLE. BUILD.	14
ILUSTRACIÓN 4. ALEXA DEVELOPER CONSOLE. CODE.	15
ILUSTRACIÓN 5. ALEXA DEVELOPER CONSOLE. TEST.	15
ILUSTRACIÓN 6. ALEXA DEVELOPER CONSOLE. DISTRIBUTION.	16
ILUSTRACIÓN 7. ALEXA DEVELOPER CONSOLE. CERTIFICATION.	16
ILUSTRACIÓN 8. ALEXA DEVELOPER CONSOLE. ANALYTICS	17
ILUSTRACIÓN 9. WEBSTORM	19
ILUSTRACIÓN 10. AMAZON API GATEWAY	21
ILUSTRACIÓN 11. ASISTENTES VIRTUALES MÁS POPULARES.	26
ILUSTRACIÓN 12. USO DE LOS ASISTENTES VIRTUALES.	26
ILUSTRACIÓN 13. CUOTA DE MERCADO DE ASISTENTES VIRTUALES.	27
ILUSTRACIÓN 14. CUOTA DE MERCADO DISPOSITIVOS MÓVILES	28
ILUSTRACIÓN 15. FLUJO EJECUCIÓN SKILL	30
ILUSTRACIÓN 16. INTERFAZ PANTALLA 2	32
ILUSTRACIÓN 17. INTERFAZ PANTALLA 18	33
ILUSTRACIÓN 18. INTENCIÓN DE PRUEBA	34
ILUSTRACIÓN 19. TABLA DE INTENCIONES	35
ILUSTRACIÓN 20: ESTRUCTURA DE DIRECTORIOS DEL BACKEND	36
ILUSTRACIÓN 21. DISEÑO COMUNICACIÓN SKILL-APP	41

1 Introducción

En estos primeros apartados dentro de la introducción, se busca que el lector tenga una idea clara del porqué se ha realizado el proyecto descrito en estas páginas y qué se busca con la realización del mismo. Todo ello se describirá en un primer apartado de cuál ha sido la motivación para la realización del proyecto, así como otro apartado donde se explicará el objetivo que se busca con su realización.

1.1 Motivación

La motivación para realizar el proyecto descrito en estas páginas nace del boom que supusieron los asistentes virtuales, sobre todo en las navidades de 2019 y 2020. Las ventas de este tipo de dispositivos fueron tal, que muchos medios de comunicación lo describieron como “el regalo estrella de esta navidad” en 2019 y utilizando esas mismas palabras, en las navidades del 2020. Fue tal su venta en esas fechas, que durante la mañana del 26 de diciembre del año 2018 y del año 2019, Alexa, el asistente virtual propiedad de Amazon, sufrió una caída de sus sistemas en Europa, entre las 09:00 y las 11:29 de la mañana, coincidiendo con la apertura de los regalos en las casas y su posterior configuración.

Por otro lado, tras observar las tiendas de aplicaciones de Amazon y Google, se vio que hay una amplia variedad de aplicaciones en el apartado educación, pero son aplicaciones que se centran en informar más que en proponer un problema y hacer pensar al usuario. Entre las aplicaciones que se encuentran en este apartado pueden verse skills que nos informan sobre ciudades, leyes, deportes, brindan curiosidades de países, deportes, elementos de la naturaleza, dan consejos sobre el día a día como la conducción o el lavado de manos... e incluso hay alguna skill que ayuda a aprender un idioma...

Si se une la cantidad de dispositivos vendidos, y por tanto, la cantidad de usuarios que poseen estos dispositivos, con la carencia de una aplicación que proponga un problema para solucionarlo, surge la idea de realizar el proyecto que se describirá las siguientes líneas.

1.2 Objetivos

El objetivo de este proyecto es poder realizar una aplicación, que, a través de la ayuda de un asistente virtual, enseñe paso a paso a un usuario, ya sea este un usuario con conocimientos sobre algoritmos o nuevo en la materia, el algoritmo voraz Dijkstra. Para ello debe:

- Tener un apartado, en el cual se presente un grafo a modo de ejemplo y se detalle al usuario cómo se resuelve el problema mediante explicaciones paso a paso, para que este comprenda cómo llegar a la solución del problema planteado. En este apartado, el usuario no debe resolver nada ni responder a ninguna pregunta, si no que únicamente deberá ir navegando por las distintas explicaciones.
- Tener un apartado donde se exponga un grafo y se formulen preguntas para que el usuario pueda llegar a resolverlo poniendo así a prueba sus conocimientos. Para este apartado, en caso de que el usuario no responda correctamente a una pregunta formulada por la aplicación, se le ha de explicar detalladamente que fallo ha podido cometer y cómo se resolvería la pregunta previamente formulada. El usuario irá pasando por las distintas preguntas en caso de acierto hasta su resolución final.
- La aplicación deberá, en todo momento, mostrar al usuario el estado de resolución del problema propuesto tanto para la el apartado de explicación como para el apartado del problema propuesto como ejercicio. De esta forma el usuario sabrá que parte del problema tiene resuelto y cuál le queda por resolver.
- Tener un apartado, en donde se muestre el algoritmo voraz Dijkstra en pseudocódigo, para que el usuario pueda consultarlo en cualquier momento y así reforzar su aprendizaje con dicho algoritmo.
- La aplicación deberá almacenar el estado del usuario tanto para la progresión llevada en el apartado de la explicación del problema como en el apartado de la resolución del problema. De esta manera si el usuario desea cambiar de apartado y posteriormente volver al apartado en el que se encontraba anteriormente, pueda hacerlo sin perder la progresión.
- La aplicación se apoyará en un dispositivo con pantalla para que de esta manera sea más fácil la comprensión del problema por parte del usuario. En el caso en el que el dispositivo que ejecute la aplicación no disponga de una pantalla para mostrar la información deseada, este debe apoyarse en otro dispositivo.
- La aplicación deberá utilizar un lenguaje perfectamente entendible para la mayoría de usuarios, siendo así un lenguaje inclusivo tanto con las personas técnicas como con las personas no conocedoras de otros algoritmos voraces ni familiarizadas con algún lenguaje de programación.
- El usuario podrá, en cualquier momento, silenciar la explicación que esté dando la aplicación quedándose esta, en silencio.

2 Fundamentos

En este apartado se hablará acerca de las tecnologías, herramientas y lenguajes de programación utilizados para la realización de las distintas partes del proyecto. Se hablará sobre qué son y porqué han sido seleccionadas cada una de ellas.

2.1 Amazon Alexa

Alexa es un asistente virtual controlado por voz y lanzado al mercado por Amazon en noviembre de 2014 integrado en su línea de altavoces inteligentes Amazon Echo. Como curiosidad, comentar que su nombre es debido a que sus desarrolladores querían que su nombre contuviese una vocal fuerte al comienzo y una x en algún lugar para que así fuese más difícil activar. Además, con este nombre se le rinde homenaje a la biblioteca de Alejandría, el centro del saber del mundo antiguo.

Actualmente, Alexa está disponible en español, inglés, francés, alemán, italiano, japonés y portugués.

Alexa, está disponible en una gran cantidad de dispositivos como puede ser desde una nevera hasta un coche. Cualquier compañía puede incorporar Alexa a sus dispositivos gracias a la herramienta Alexa Voice Service. No entraremos en detalle, ya que en el contexto de estas páginas utilizaremos un Amazon Echo con Alexa ya integrado.

Para comenzar a conversar con Alexa, un usuario tiene que decir su nombre en voz alta cerca de un dispositivo donde esté configurada y se activará. A continuación, el usuario debe decir lo que se necesite saber o hacer y Alexa realizará las operaciones necesarias y le contestará.

Alexa trae integradas ciertas funcionalidades desarrolladas por Amazon como son, conocer el estado del tiempo, poner una alarma, hacer llamadas entre dispositivos Amazon Echo, conectarse con otros dispositivos inteligentes de la casa como pueden ser bombillas o cámaras de vigilancia... Además, Alexa permite aplicaciones de terceros, llamadas skills, y aquí es donde entra el desarrollo descrito.

2.1.1 Skills

Una skill es una aplicación capaz de añadirle funcionalidad extra a cualquier dispositivo que tenga integrado Alexa. Esta conexión se realiza a través de la herramienta Alexa Skills Kit. Las skills se apoyan en un servicio de voz de Alexa que vive alojado en la nube y desarrollado por Amazon. En dicho servicio se realizan dos etapas de machine learning cuando los usuarios interactúan con una skill. Una primera etapa llamada “Reconocimiento de voz automatizado” donde transforma en texto lo recibido por audio por parte del usuario, y una segunda etapa, llamada “Comprensión del lenguaje natural” donde se le asigna un significado a la frase.

Para invocar una skill:

- Lo primero debe ser la palabra de accionado del dispositivo. Por defecto es Alexa, pero puede ser configurada y tener los nombres de Echo o Amazon. Esta palabra hará que el dispositivo “despierte” y comience a procesar todo lo que el usuario le indica a continuación.
- Seguido, viene la palabra de lanzamiento. Puede tener diferentes valores, como por ejemplo, lanza, abre, inicia, empieza, comienza, pregunta... Esta palabra es necesaria para que el dispositivo utilizado por Alexa conozca que un usuario va a llamar a una skill.
- Después, viene el nombre de invocación. Este será el nombre por el que los usuarios llamarán a una skill determinada y provocará que dicha skill comience su procesamiento.
- De manera adicional, en el momento de invocar una skill, se le puede pasar una frase. Dicha frase se llamará enunciado. Este enunciado será pasado a la skill invocada, que lo procesará y convertirá en una intención. Los enunciados, además pueden llevar variables, que son elementos que pueden tener diferentes valores, pero que la skill tratará y transformará en una misma intención. Estas variables pueden tener un tipo predefinido, como pueden ser números, países, colores... o puedes crearte tu propio tipo de variable.

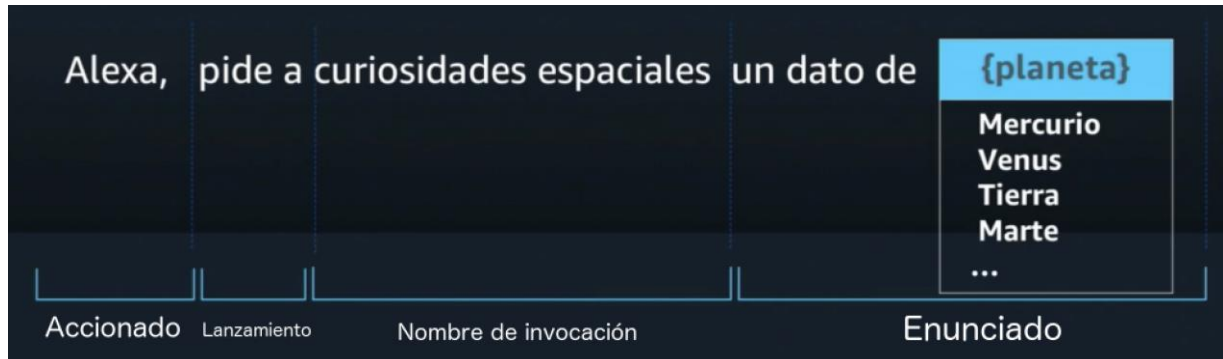


ILUSTRACIÓN 1. INVOCACIÓN DE UNA SKILL.

2.1.2 Intenciones

Una intención es una acción que cumple con una petición realizada por un usuario. Cada intención puede ser activada por uno o más enunciados. Las intenciones pueden ser creadas por el desarrollador de la skill, aunque, tiene que haber una serie de interacciones por defecto que tienen que comportarse de la misma manera para todas las skill para que se comporten de la misma manera. Si no cumples con estas intenciones por defecto, no pasarás el proceso de validación de la skill. Estas interacciones por defecto son:

- AMAZON.CancelIntent. Se acciona con los enunciados “cancela”, “olvidalo”... Sirve para parar la acción que se esté realizando dentro de la skill sin cerrarla.
- AMAZON.StopIntent. Se acciona con los enunciados “para”, “apagar”... Sirve para parar la acción que está realizando la skill y salir de ella.
- AMAZON.HelpIntent. Se acciona con los enunciados “ayuda”, “ayúdame”... Sirve para que la skill responda con el mensaje de ayuda que tenga configurado.

2.1.3 Alexa-Hosted Skill

Hasta ahora se ha descrito la interfaz de voz de usuario, que es lo que la skill tiene que comprender cuando un usuario habla con ella. A esta parte de la skill se le llama Front-end.

Pero esto no es únicamente lo que forma una skill, sino que tiene que haber un script de programación para que la skill pueda realizar las operaciones que se deseen. A esto se le llama back-end. El back-end es un script de programación que se ejecuta en un servidor y realiza una serie de acciones.

A la hora de realizar un back-end para una skill puede alojarse en cualquier servidor que se desee, lo único que se debe tener en cuenta es que el servidor seleccionado permita peticiones HTTPS.

Lo más popular a la hora de desarrollar una skill es tener el back-end en una Lambda.

Amazon, además, provee la herramienta Alexa-Hosted skill. Esta herramienta aprovisiona y administra automáticamente el conjunto de servicios de AWS que necesita la skill, como por ejemplo las lambdas necesarias. De esta forma, desde la consola de desarrollador de Alexa se puede construir, editar y publicar una skill sin tener que preocuparse por configurar el back-end.

Alexa-Hosted skill será la herramienta que se utilizará para crear la skill, ya que, como se ha hablado anteriormente, abstrae de tener que configurar las lambdas para el back-end de la skill.

2.1.4 Alexa Developer Console

Amazon, provee de una consola para desarrolladores llamada Alexa Developer Console. En esta consola se puede crear, administrar, editar y desplegar una skill a través de una interfaz de usuario de una manera muy clara y sencilla.

Se divide en las principales tareas:

- Crear una skill. Permite configurar el nombre, el idioma por defecto, si se quiere partir de una estructura ya creada o completamente nueva y si se quiere que el back-end esté en un servidor propio o que esté configurado por Alexa-Hosted skill. Si se quiere utilizar Alexa-Hosted skill puede hacerse en los lenguajes de programación Node.js y Python.

Create a new skill Cancel Create skill

Model: Custom
Host: Provision your own

Skill name

Enter skill name

Skill name must have at least 2 characters. 0/50 characters

Default language

Spanish (ES)

More languages can be added to your skill after creation

1. Choose a model to add to your skill

There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.

<p>Custom</p> <p>Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.</p>	<p>Flash Briefing</p> <p>Give users control of their news feed. This pre-built model lets users control what updates they listen to.</p> <p>"Alexa, pon el resumen de noticias."</p>	<p>Smart Home</p> <p>Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.</p> <p>"Alexa, enciende las luces de la cocina"</p>	<p>Video</p> <p>Let users find and consume video content. This pre-built model supports content searches and content suggestions.</p> <p>"Alexa, pon Interstellar"</p>
---	---	---	---

2. Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

<p>Provision your own</p> <p>Provision your own endpoint and backend resources for your skill. This is recommended for skills that have significant data transfer requirements. You will not gain access to the console code editor.</p>	<p>Alexa-Hosted (Node.js)</p> <p>Alexa will host skills in your account up to the AWS Free Tier limits and get you started with a Node.js template. You will gain access to an AWS Lambda endpoint, 5 GB of media storage with 15 GB of...</p>	<p>Alexa-Hosted (Python)</p> <p>Alexa will host skills in your account up to the AWS Free Tier limits and get you started with a Python template. You will gain access to an AWS Lambda endpoint, 5 GB of media storage with 15 GB of...</p>
---	---	---

ILUSTRACIÓN 2. ALEXA DEVELOPER CONSOLE. CREATE SKILL.

- Pestaña de Build. Permite configurar una skill tanto en la parte front-end, configurando una interfaz de voz de usuario, como en la parte back-end, indicando cuales son los endpoint.

← Your Skills **Build** Code Test Distribution Certification Analytics Provide feedback

Spanish (ES)

CUSTOM

Interaction Model

Utterance Conflicts (3)

Invocation

Intents (24) + Add

Slot Types (0) + Add

JSON Editor

Interfaces

Endpoint

Intent History

Annotation Sets

Display

IN-SKILL PRODUCTS

ACCOUNT LINKING

PERMISSIONS

How to get started

Alexa Skills Kit Developer Tutorial for Progra... Ver más ta... Compartir

amazon alexa

Developer Console: Build

Resources

[Update your live skill instantly](#)

Qualify for live updates to your skill if changes are made to slot values and/or sample utterances. [Learn more](#) about live updates to your skill.

[Catalog Management](#)

Use Catalog management for managing slot types with large, constantly changing slot values. Catalog management is currently available only on Alexa Skill Management API (SMAP) and the Alexa Skills Kit Command Line Interface (ASK CLI).

[Feature Updates & Releases](#)

New Alexa Skills Kit Features and Tools.

[Getting Started: A Comprehensive Course \(Cake Walk\)](#)

Cake Walk: Build an Engaging Alexa Skill. This course is for software developers, voice developers, ...

Skill builder checklist

Complete these steps to be able to test your skill using the simulator in the test tab, or with your echo device.

REQUIRED	1. Invocation Name >	Enter an invocation name for your skill	✓
REQUIRED	2. Intents, Samples, and Slots >	Add at least one intent and one sample utterance	✓
REQUIRED	3. Build Model >	Successfully build your interaction model	✓
REQUIRED	4. Endpoint >	Set a web service endpoint to handle skill requests	✓

ILUSTRACIÓN 3. ALEXA DEVELOPER CONSOLE. BUILD.

- Pestaña de Code. Si se ha configurado una skill con Alexa-Hosted skill aparecerá esta pestaña para que poder desarrollar el código en el lenguaje de programación elegido. Si se ha decidido alojar el código en un servidor externo, ya sea una lambda o cualquier otro servidor, esta pestaña no aparecerá.

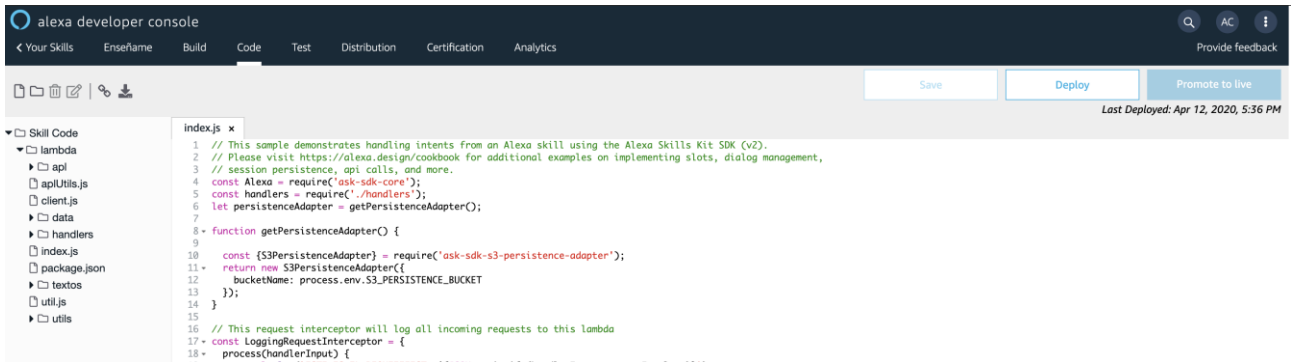


ILUSTRACIÓN 4. ALEXA DEVELOPER CONSOLE. CODE.

- Pestaña Test. En esta pestaña se pueden hacer pruebas sobre la skill ya sea por voz o pasándole las órdenes mediante texto.

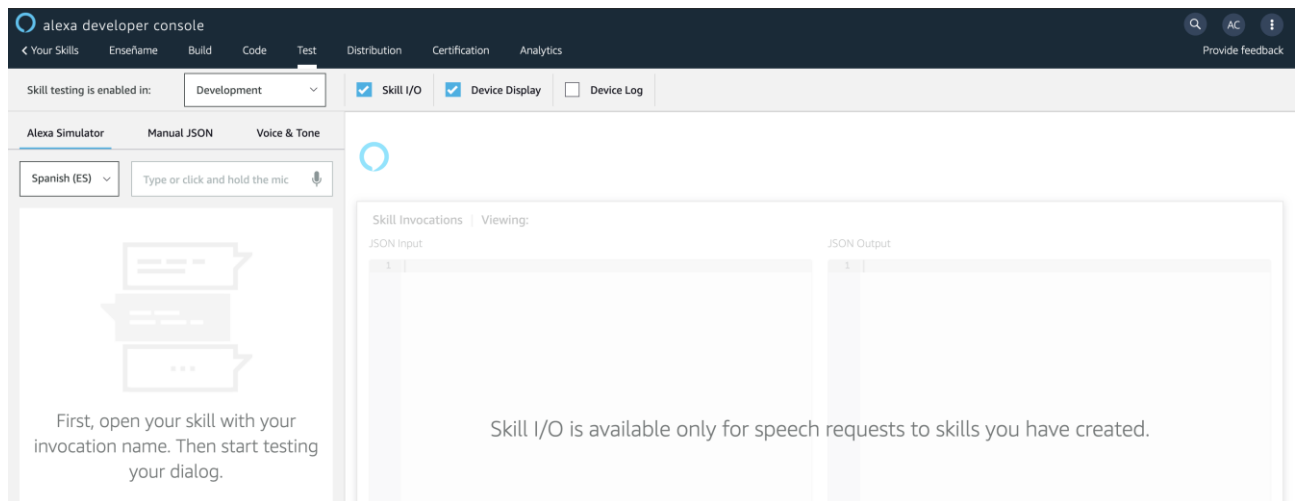


ILUSTRACIÓN 5. ALEXA DEVELOPER CONSOLE. TEST.

- Pestaña Distribution. En esta pestaña puede verse cómo aparecerá la skill dentro de la tienda de skills de Amazon.

The screenshot shows the 'Distribution' tab in the Alexa Developer Console. The left sidebar has a 'Skill Preview' dropdown menu with 'Spanish' selected. Below it are links for 'Privacy & Compliance' and 'Availability'. The main content area is titled 'Spanish Store Preview' and includes a sub-header 'Tell us how your skill should appear to Spanish customers.' There is a 'Copy from an existing locale' button. Below this are three text input fields: 'Public Name *' (containing 'Enseñame' with a 8/50 characters count), 'One Sentence Description *' (with a placeholder 'Enter a short description (160 character maximum) about your skill...' and a 0/160 characters count, showing a 'Value is too short' error), and 'Detailed Description *' (with a placeholder 'Enter an engaging description of the skill's purpose, features, and how it works...' and a 0/4,000 characters count, also showing a 'Value is too short' error).

ILUSTRACIÓN 6. ALEXA DEVELOPER CONSOLE. DISTRIBUTION.

- Pestaña Certification. En esta pestaña se puede validar la skill, ejecutar pruebas de pre-certificación y enviar la skill al proceso de certificación.

The screenshot shows the 'Certification' tab in the Alexa Developer Console. The left sidebar has a 'Validation' tab selected, with a 'Submission' link below it. The main content area is titled 'Validation' and features a blue 'Run' button in the top right corner. Below the title, there is a red error message: '7 Fixes Required'.

ILUSTRACIÓN 7. ALEXA DEVELOPER CONSOLE. CERTIFICATION.

- Pestaña de Analytics. En esta pestaña el desarrollador dispone de un resumen del análisis de los datos de la skill como por ejemplo cuantas veces ha sido usada, por cuantos usuarios, que intenciones han sido las más usadas...

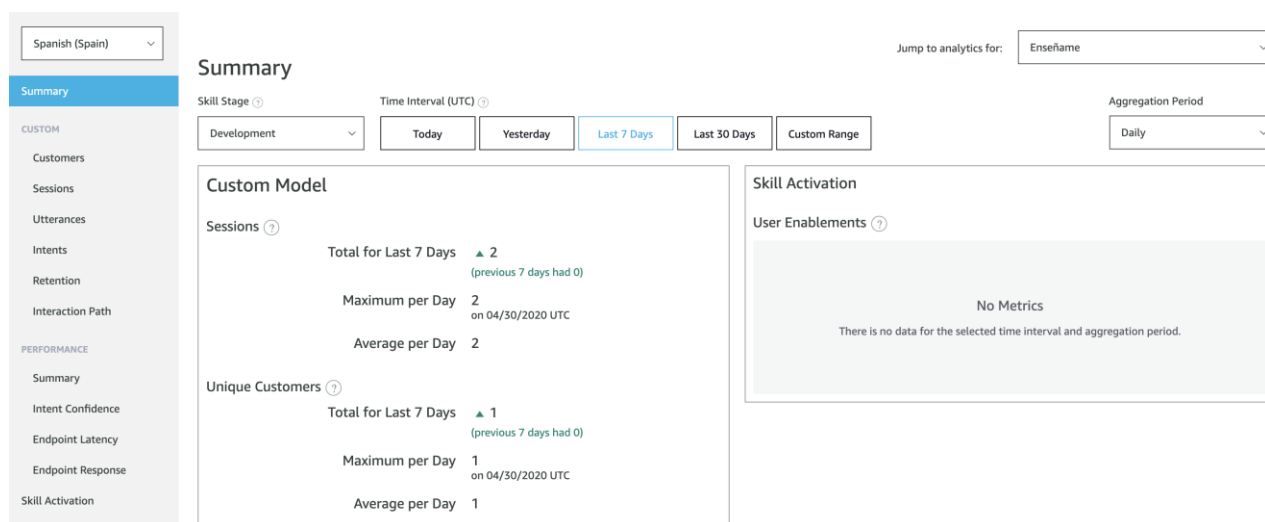


ILUSTRACIÓN 8. ALEXA DEVELOPER CONSOLE. ANALYTICS

2.1.5 Display

Una de las herramientas a las que se tiene acceso desde Alexa Developer Console, es “Display”, la cual permite crear imágenes para poder mostrarlas al usuario desde la pantalla del dispositivo que tenga integrado a nuestro asistente virtual Alexa.

Esta herramienta permite crear una plantilla y poder añadirle imágenes, videos, textos, elementos táctiles... darles un formato y ver cómo quedaría en los distintos dispositivos que permiten la integración con Alexa y contengan una pantalla.

Esta es la herramienta elegida para realizar las imágenes que saldrán en la skill desarrollada y explicada en estas páginas.

2.1.6 NodeJS

Javascript es un lenguaje de programación que se ejecuta en el lado del cliente, es decir, un navegador web interpreta el código.

Para poder ser ejecutado en el lado del servidor, es necesario un entorno en tiempo de ejecución capaz de interpretar el código y ejecutarlo. Ahí nace Node.js, que es una máquina virtual que utiliza un modelo asíncrono y dirigido por eventos, capaz de transformar el código que originalmente se creó para ser ejecutado en navegadores, en código que puede ser ejecutado en un ordenador como si de una aplicación independiente se tratara.

En comparación con las técnicas tradicionales de servicio web donde cada conexión (que crea una solicitud) genera un nuevo subproceso, ocupando la RAM del sistema y regularmente maximizando la cantidad de RAM disponible, Node.js opera en un solo subproceso, utilizando el modelo entrada y salida sin bloqueo de la salida, lo que le permite soportar decenas de miles de conexiones al mismo tiempo mantenidas en el bucle de eventos.

Cuando hay una nueva solicitud se genera un tipo de evento. El servidor empieza a procesarlo y, cuando hay una operación de bloqueo de entrada y salida, no espera hasta que se complete y en su lugar crea una función de devolución de llamada. El servidor comienza en el acto a procesar otro evento (tal vez otra solicitud) y cuando finaliza la operación de entrada y salida, continuará trabajando en la solicitud ejecutando la devolución de llamada tan pronto como tenga tiempo.

Por lo tanto, el servidor nunca necesita crear más subprocesos o cambiar entre subprocesos, lo que significa que tiene muy poca sobrecarga. Por tanto, Node es capaz de manejar una gran cantidad de conexiones simultáneas con un alto nivel de rendimiento, lo que equivale a una alta escalabilidad.

Debido a esto, utilizaremos Node.js como lenguaje para el desarrollo de la skill.

2.1.7 Git

Como se ha visto anteriormente, Alexa Developer Console, brinda la oportunidad de editar el código desde la pestaña “Code”, por lo que se podría desarrollar los scripts desde ahí sin ningún problema.

En este caso, se cree que lo más recomendable es tener una herramienta de control de versiones, ya que si se editara el código directamente en Alexa Developer Console no se tendría un historial de cambios y no se tendría la opción de volver atrás. Para ello, se utilizará Git.

Git es un software de código abierto de control de versiones y pensado para la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Git, nos provee de muchas herramientas para hacer posible esto. Nosotros en este proyecto utilizaremos:

- Creación de ramas por funcionalidad. De esta manera podemos ver de una forma rápida el desarrollo de cada nueva característica
- Realizar commits por cada pequeño desarrollo creado para así tener un control y poder volver atrás en caso de error.

Git se utilizará para trabajar con el código de desarrollo de manera local, mediante ramas de desarrollo y así tener un control de versiones y un historial de cambios.

2.1.8 Webstorm

Webstorm es un entorno de desarrollo integrado (IDE) especialmente diseñado para Javascript por la empresa JetBrains. Al ser este su propósito, no se necesitará configurar ningún complemento adicional para poder empezar a utilizarlo para el desarrollo de la skill. Webstorm ofrece la detección de errores mientras se escribe código de programación, permite depurar, lanzar test unitarios con las principales librerías de los mismos, permite una refactorización del código con el mínimo esfuerzo posible, y además, ofrece una integración con git sin necesidad de una configuración adicional, por lo que es la herramienta más potente que se utilizar para realizar la skill.

Tiene un coste de 129€ al año, pero por ser estudiante, se dispone de la licencia de una manera gratuita.

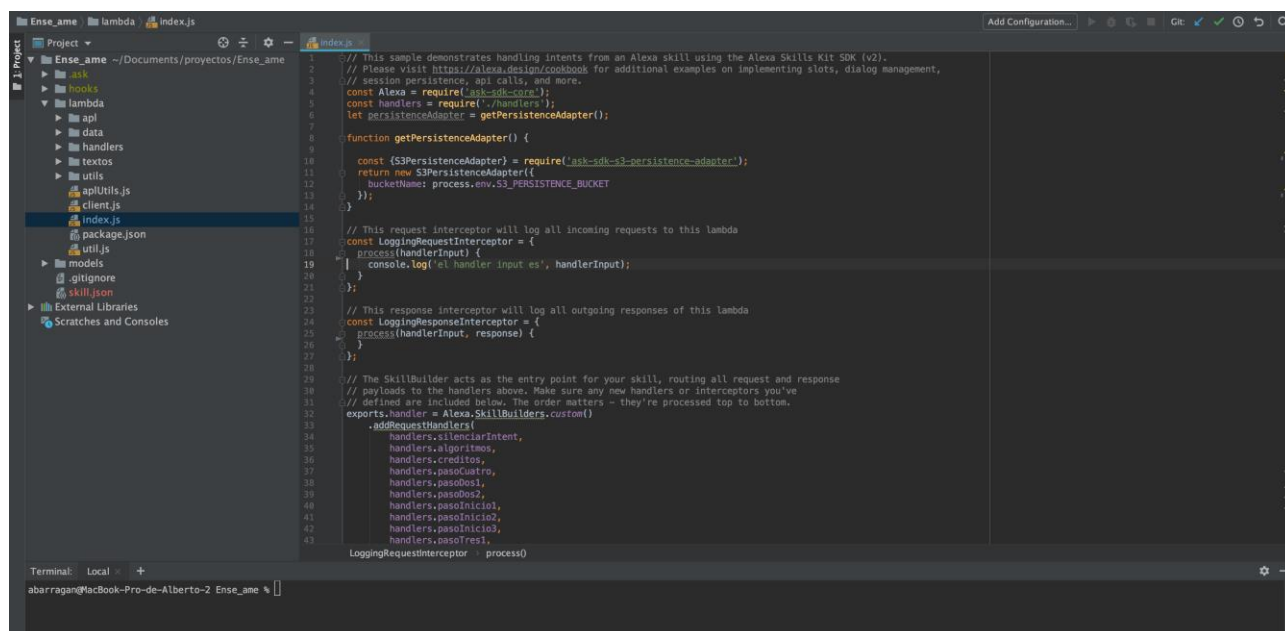


ILUSTRACIÓN 9. WEBSTORM

2.2 Android

Android es un sistema operativo para dispositivos móviles inicialmente desarrollado por la empresa Android Inc y más tarde, antes de su lanzamiento en el año 2007, adquirida por Google en el año 2005. Es un sistema operativo de código abierto basado en el kernel de Linux. Principalmente se encuentra en teléfonos móviles, pero puede encontrarse en distintos dispositivos como pueden ser desde relojes hasta coches. Actualmente, Google tiene a nivel mundial una cuota de mercado en teléfonos móviles del 72.26% (a marzo de 2020) a nivel mundial y una cuota del 78,34% a nivel nacional.

2.2.1 Android Studio

Android estudio es un entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en los IDEs de JetBrains (como WebStorm, visto en el apartado 2.1.8). Además de las funcionalidades descritas en Webstorm, Android Studio tiene una serie de funcionalidades específicas para el desarrollo de apps como son:

- Un sistema de compilación flexible basado en Gradle
- Un emulador rápido y cargado de funciones
- Un entorno unificado donde poder desarrollar para todos los dispositivos Android
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación
- Integración con GitHub y plantillas de código para ayudar a compilar funciones de apps comunes y también importar código de muestra
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros

Además de ser la herramienta oficial para el desarrollo de apps de Android, es la herramienta más potente actualmente en el mercado y no tiene ningún coste, por lo que es el IDE seleccionado para desarrollar la aplicación.

2.2.2 Kotlin

Kotlin es un lenguaje de programación fuertemente tipado desarrollado por JetBrains (la misma empresa que Android Studio y WebStorm) anunciado por primera vez en el verano de 2011 y publicado en febrero de 2012. Kotlin está disponible con licencia de software libre Apache 2, por lo que cualquiera puede usarlo a su antojo, así que sigue las mismas premisas de código abierto por las que lucha Android. Kotlin está pensado para correr en la Máquina Virtual de Java e incluso ser compilado a código fuente de Javascript, por lo que le provee de mucha versatilidad. El código desarrollado en Kotlin puede utilizar los frameworks y librerías Java aunque su sintaxis tiene un parecido a Scala. Según las palabras de Andrey Breslav, líder de desarrollo de Kotlin, este lenguaje ha sido diseñado para superar a Java pero sin dejar de ser interoperable con el código Java.

2.3 Amazon Web Service

Amazon Web Service es una plataforma desarrollada y comercializada por Amazon, que ofrece más de 175 servicios alojados en la nube que van desde tecnologías de infraestructura como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas. Para este proyecto se utilizarán tres de estos servicios: Amazon API Gateway, AWS Lambda y AWS DynamoDB

2.3.1 Amazon API Gateway

Amazon API Gateway es un servicio de Amazon Web Service para la creación, publicación, mantenimiento, monitorización y protección de API REST, HTTP y WebSocket a cualquier escala. Entre sus características pueden destacarse:

- Compatibilidad con las API con estado (WebSocket) y las API (HTTP y REST).
- Mecanismos de autenticación eficaces y flexibles.
- Portal para desarrolladores para la publicación de sus API.
- Implementaciones de la versión Canary para el despliegue de cambios de forma segura.
- Registro de CloudTrail y monitorización del uso y de los cambios en las API.
- Registro de acceso y registro de ejecución de CloudWatch, que incluye la posibilidad de establecer alarmas.
- Posibilidad de utilizar plantillas de AWS CloudFormation para habilitar la creación de las API.
- Soporte para los nombres de dominio personalizados.
- Integración con AWS WAF para la protección de sus API frente a ataques web comunes.
- Integración con AWS X-Ray para comprender y cribar latencias de rendimiento.

Una foto de su arquitectura sería:

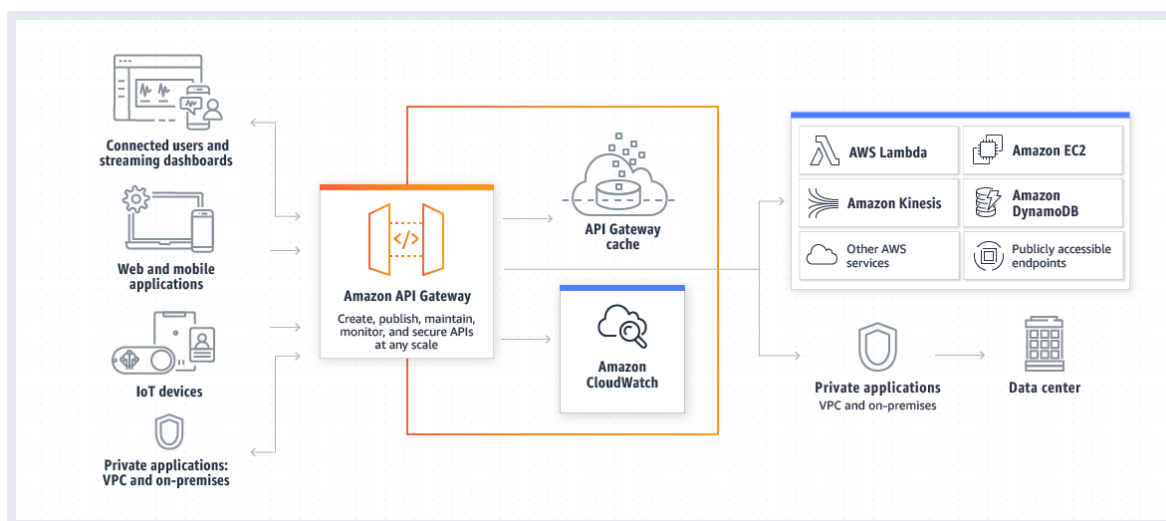


ILUSTRACIÓN 10. AMAZON API GATEWAY

En ella, puede verse como Amazon API Gateway actúa como enlace entre cualquier herramienta que se desee y un servidor, olvidándose así de tener que montar toda la infraestructura.

Como se ha dicho, una de las herramientas que permite Amazon API Gateway es crear un API de WebSocket. En una API de WebSocket, el cliente y el servidor pueden enviarse mensajes entre sí en cualquier momento. Gracias al API de WebSocket de API Gateway pueden crearse aplicaciones seguras de comunicación en tiempo real sin tener que aprovisionar ni administrar servidores para administrar las conexiones o los intercambios de datos a gran escala, por lo que es un sistema perfecto para permitir la comunicación entre la skill de Alexa y la app Android.

Por otro lado, el API de WebSocket permite 1 millón de mensajes enviados y 750.000 minutos de conexión al mes de forma gratuita, por lo que para la finalidad deseada en este proyecto es suficiente.

2.3.2 AWS Lambda

AWS Lambda es un servicio prestado por Amazon Web Service que ejecuta código en respuesta a eventos y administra automáticamente los recursos informáticos requeridos por cada evento a nivel de disponibilidad, seguridad, rendimiento y escalabilidad. (Amazon - <https://aws.amazon.com/es/lambda/features/>).

La ventaja de utilizar Lambda es que hay que preocuparse por mantener un servidor ejecutándose todo el día, con todo lo que ello conlleva en temas de requerimientos mencionados anteriormente, si no que AWS lo provee de manera ilimitada y hasta cierto nivel gratuita, un millón de solicitudes gratuitas al mes y 400 000 GB/segundos de tiempo de cómputo al mes por lo que hace el sistema perfecto para desarrollar ciertas funciones que se verán más adelante en cuanto a la comunicación entre la skill y la aplicación.

2.3.3 AWS DynamoDB

Amazon DynamoDB es un servicio de bases de datos NoSQL totalmente administrado para que haya que preocuparse del aprovisionamiento, la instalación y la configuración del hardware, ni tampoco de las tareas de replicación, revisión del software o escalado de clústeres, ni tampoco en temas de seguridad y protección de información ya que ofrece un cifrado en reposo. Además, permite:

- Crear tablas de base de datos capaces de almacenar y recuperar cualquier cantidad de datos con una escalabilidad óptima y sin tiempos de inactividad ni reducción del rendimiento.
- Realizar copias de seguridad y su posterior recuperación.
- Eliminar automáticamente los elementos vencidos de las tablas para así reducir el consumo de almacenamiento

- Distribuir los datos en distintos servidores para así tener una alta disponibilidad y durabilidad.

Como en los productos vistos anteriormente, tenemos una parte gratuita de AWS DynamoDB. En este caso disponemos 25 GB de almacenamiento de datos y 2,5 millones de solicitudes de lectura de streams de DynamoDB Streams, por lo que para el uso que se le va a dar en este proyecto, bastará con la capa gratuita.

3. Descripción informática.

En el siguiente apartado se describirá el proceso de realización del proyecto, comenzando por la metodología de trabajo utilizada, siguiendo con el análisis realizado en dónde se verá por qué se ha utilizado cada una de las herramientas, continuando por el diseño de la aplicación y finalizando con la descripción de la implementación de las distintas herramientas

3.1 Metodología de trabajo.

Para el desarrollo de este proyecto, el método utilizado es el método de desarrollo iterativo e incremental, el cual se basa en la idea de desarrollar una implementación inicial, mostrárselo al cliente (en este caso el tutor) y desarrollar sucesivas versiones hasta obtener el sistema requerido. Esta metodología se asemeja mucho a la forma en la que resolvemos los problemas. Normalmente elaboramos una solución en varios pasos y no de una vez, volviendo a un paso anterior si nos hemos equivocado. Cada incremento o versión del sistema incluye una nueva funcionalidad. Una de las principales ventajas que tiene este método es que es más fácil obtener las impresiones del cliente sobre el software que se está desarrollado ya que es más fácil para los clientes juzgar el progreso a través de entregas periódicas de software funcional que a través de documentos. Además, en cada entrega se va aprendiendo sobre el trabajo ya realizado pudiendo planificar mejor los nuevos desarrollos.

Siguiendo esta metodología de trabajo, se realizaron una serie de entregables al cliente para la revisión del estado de la aplicación y propuesta de mejoras de la parte ya desarrollada o nuevas funcionalidades que pudiesen aportar valor. A continuación se describen:

Para el primer entregable se buscó que el usuario de la skill pudiese aprender a resolver un problema propuesto. Para este primer entregable, la skill debería ir explicando al usuario cómo se resuelve el problema de los caminos mínimos y el usuario únicamente interactuaría con la skill diciendo siguiente una vez hubiese escuchado la explicación. En este primer entregable se consiguió lo acordado, pero se observó que el usuario no podía poner en práctica lo aprendido, por lo que para la segunda entrega la skill debería tener el enunciado de un problema como ejemplo y el usuario intentase resolverlo mientras era guiado y corregido por la skill de Alexa.

En el segundo entregable se comprobó cómo la skill podía guiar al usuario por la explicación y comprobar si el usuario respondía correctamente o erróneamente a las preguntas planteadas por la misma. En esta segunda entrega, el grafo tanto de la explicación como del problema propuesto se dictaba mediante voz por parte de la skill de Alexa, por lo que se hacía muy complejo de entender.

Por tanto, para el tercer entregable se determinó que la skill debería mostrar por pantalla los grafos propuestos y de esta forma mejorar el entendimiento del problema.

En la tercera entrega, se observó el correcto funcionamiento del problema planteado para dicha entrega observándose así una mejora en la comprensión del enunciado. El problema observado en esta entrega fue ver que no todos los dispositivos en los que se ejecuta Alexa tienen pantalla, por lo que para la cuarta entrega se debería crear una aplicación móvil para poder mostrarles a todos los usuarios la imagen del grafo independientemente del dispositivo de Alexa que utilice.

Para la cuarta entrega se desarrolló un websocket para poder comunicar la skill una aplicación y una aplicación en Android para así disponer de un dispositivo con pantalla en el que mostrar lo que se deseara.

3.2 Análisis

Al comienzo del proyecto, hubo que elegir sobre qué tecnologías trabajar para el desarrollo de la aplicación propuesta. En las siguientes líneas se describirá el porqué de cada herramienta seleccionada.

3.2.1 ¿Por qué Alexa?

Como se ha comentado en la introducción, la aplicación quiere ser desarrollada para los asistentes virtuales, por lo que la primera acción que se tomó, fue ir al mercado de los asistentes virtuales para ver la variedad que hay y qué podían ofrecer cada uno de ellos al proyecto. En la tabla siguiente puede verse un análisis de los más populares del mercado.

Asistente virtual	Marca propietaria	Idiomas
Alice	Yandex	Ruso
AliGenie	Alibaba Group	Chino
Alexa	Amazon	Español, Inglés, Francés, Alemán, Italiano, Japonés y Portugués
Siri	Apple	Español, Chino, Árabe, Danés, Alemán, Inglés, Francés...
DuerOS	Baidu	Chino
Clova	Naver Corp. And Line Corp	Japonés y coreano
Google Assistant	Google	Español, Danés, Holandés, Inglés, Francés...
Cortana	Microsoft	Español, Inglés, chino, francés, alemán, italiano, japonés y portugués

Asistente virtual	Marca propietaria	Idiomas
Sirona	Safety Labs	Inglés
Xiowei	Tencent	Chino
Bixby	Samsung	Español, Inglés, italiano, alemán, coreano y chino.
Magenta	Deutsche Telekom	Alemán.

ILUSTRACIÓN 11. ASISTENTES VIRTUALES MÁS POPULARES.

Tras observar todos ellos y ver la gran variedad se comenzó a descartar por idiomas soportados. La idea, en un primer momento, es hacer una aplicación en español, por lo que los asistentes virtuales que soportan este idioma se reducen a 5: Alexa, Bixby, Cortana, Google Assistant y Siri.

El número ya se había reducido a 5, por lo que era el momento de buscar una comparativa entre esos 5 asistentes virtuales. En el estudio realizado por Telefónica y presentado en la “32nd Human Computer Interaction Conference” en Julio de 2018, llamado “Todo el mundo habla de los asistentes virtuales, pero ¿cómo los utilizan realmente los usuarios?” se analizan 4 de los 5 asistentes virtuales (a excepción de Bixby que por aquella época aún no estaba disponible en español) por lo que se eligió este estudio como referencia. En este estudio, se puede comprobar que, apoyados por la gráfica mostrada a continuación, Alexa es el dispositivo más utilizado a diario por los usuarios de estos asistentes con un 67% de usuarios que lo usan a diario frente a sus competidores, los cuales disfrutan de un uso con márgenes que van desde el 32% en el caso de Cortana, pasando por el 30% en el caso de Google assistance y llegando hasta el 21% en el caso de Siri.

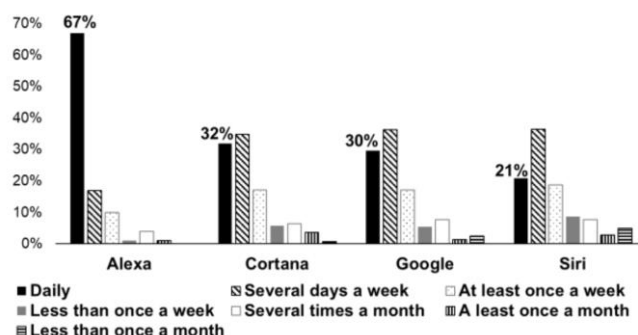


ILUSTRACIÓN 12. USO DE LOS ASISTENTES VIRTUALES.

No se quiso utilizar únicamente ese estudio, sino que además, la elección de Alexa como asistente virtual viene apoyada por el estudio publicado por Canalys, empresa líder mundial en análisis del mercado tecnológico, en donde analizan el mercado de los asistentes virtuales a nivel mundial, y en donde se puede observar que Amazon es el líder en ventas tanto en el año 2018 como en el año 2019,

teniendo en este último, una cuota de mercado mundial del 31.7% como se puede comprobar en la siguiente tabla.

Vendor	Q4 2019 shipments (million)	Q4 2019 Market share	Q4 2018 shipments (million)	Q4 2018 Market share	Annual growth
Amazon	15.6	31.7%	11.3	34.9%	+38%
Google	12.5	25.3%	8.8	27.2%	+41%
Baidu	5.7	11.6%	2.5	7.7%	+129%
Alibaba	5.6	11.4%	2.7	8.3%	+109%
Xiaomi	4.6	9.4%	2.5	7.8%	+82%
Others	5.2	10.5%	4.6	14.2%	+13%
Total	49.2	100.0%	32.5	100.0%	+52%



Note: percentages may not add up to 100% due to rounding

**Source: Canalys Smart Speaker Analysis (sell-in shipments),
February 2020**

ILUSTRACIÓN 13. CUOTA DE MERCADO DE ASISTENTES VIRTUALES.

Por lo tanto, observando que Alexa es el dispositivo más usado a diario por los usuarios de los mismos y viendo que es el que más cuota de mercado tiene, es el asistente virtual seleccionado para este proyecto ya que uno de los objetivos es llegar al mayor número de personas posible.

3.2.2 ¿Por qué Android?

Para la elección del sistema operativo de dispositivos móviles se hizo la misma operación que para el asistente virtual, es decir, investigar el mercado y realizar una comparativa entre ellos. En el mercado de sistemas operativos para dispositivos móviles, se puede observar que también hay una extensa variedad, pero que la mayoría del mercado está ocupado por Android (en sus diferentes versiones) e iOS. Una explicación más detallada, con datos en porcentaje de cuota de mercado, la podemos encontrar en el estudio de mercado realizado por la empresa statcounter en donde se muestra el estado del mercado en España desde Marzo de 2019 a marzo de 2020.

Date	Android	iOS	Samsung	Windows	Unknown	Other
2020-03	78.34	21.39	0.21	0.02	0.03	0.02
2020-02	80.38	19.41	0.14	0.03	0.02	0.01
2020-01	80.11	19.65	0.15	0.04	0.03	0.02
2019-12	80.83	18.92	0.15	0.05	0.03	0.02
2019-11	80.86	18.89	0.14	0.08	0.02	0.02
2019-10	80.77	18.96	0.15	0.08	0.02	0.02
2019-09	79.41	20.31	0.14	0.09	0.02	0.03
2019-08	78.35	21.36	0.13	0.1	0.02	0.03
2019-07	77.55	22.11	0.15	0.11	0.04	0.04
2019-06	77.97	21.71	0.14	0.12	0.02	0.04
2019-05	79.28	20.41	0.12	0.13	0.01	0.04
2019-04	80.26	19.41	0.14	0.15	0.01	0.04
2019-03	80.05	19.62	0.13	0.15	0	0.04

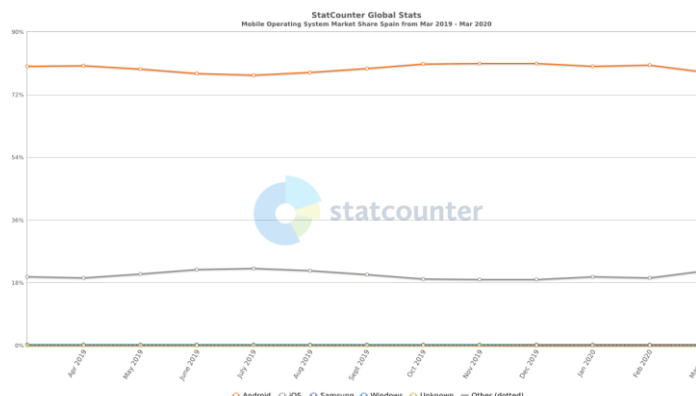


ILUSTRACIÓN 14. CUOTA DE MERCADO DISPOSITIVOS MÓVILES

Como se puede apreciar en estos datos, Android posee, a marzo de 2020, un 78,34 % del mercado de dispositivos móviles en España, por lo que si se desea llegar al mayor número posible de usuarios, esta debe ser la plataforma principal utilizada.

Otro de los motivos por los que se ha elegido Android como sistema operativo en el que se ejecutará la aplicación desarrollada, es el motivo económico. En Android, para subir a la tienda de aplicaciones, se puede hacer con un correo de gmail, el cual es gratuito. Sin embargo, para poder subir aplicaciones a la tienda de iOS es necesario una cuenta de desarrollador de aplicaciones, la cual tiene una tarifa anual de 99 USD.

Por lo tanto, si se une la cuota de mercado, con la diferencia de coste para desarrollar entre Android e iOS, se concluye que Android será el sistema operativo que mejor se adapta a nuestras necesidades.

3.2.3 ¿Por qué Kotlin?

Para el desarrollo de la aplicación móvil se debía elegir entre desarrollar en Java o en Kotlin, ya que son los dos lenguajes más utilizados para el desarrollo de este tipo de aplicaciones.

En la Google I/O Conference, la conferencia anual de desarrolladores organizada por Google y celebrada en mayo de 2019, Google anunció a Kotlin como su lenguaje preferido para el desarrollo de aplicaciones móviles. De hecho, anunció que “Muchas nuevas APIs y características serán ofrecidas primero para Kotlin. Si quieres empezar un nuevo proyecto, deberías escribirlo en Kotlin. El código escrito en Kotlin a menudo significa mucho menos código que escribir, probar y mantener.”. En solo 2 años, Kotlin ha pasado de ser presentado en la Google I/O Conference de 2017 como un lenguaje al que Google le daba soporte en Android Studio, a ser presentado como lenguaje principal a la hora de desarrollar aplicaciones móviles siendo utilizado por más del 50% de los

desarrolladores. De hecho, en la página web de referencia para la formación de desarrollo de aplicaciones Android, existe un apartado dedicado exclusivamente a Kotlin.

Entonces, ¿Por qué Kotlin en vez de Java?

- Kotlin permite escribir menos código. Menos código implica menos errores. Lo bueno es que con Kotlin no se pierde en legibilidad, sino que es un código mucho más entendible ya que JetBrains intentó hacer un lenguaje lo más conciso posible.
- Nulabilidad de tipos. Kotlin ofrece herramientas para que lidiar los posibles objetos nullables sea más sencillo que en Java. Por ejemplo, cuenta con el operador *safe call* (?) para evitar *NullPointerException* al acceder a un opcional, o con el operador *safe cast* para proteger al desarrollador en caso de querer realizar un casting. El compilador de Kotlin, además, fuerza a controlar los tipos que podrían tener valor nulo, e incluso introduce comprobaciones en tiempo de ejecución en caso de compatibilización con código Java.
- Programación funcional: En Kotlin se dispone de soporte nativo para trabajar con colecciones y sets de datos como Streams. Se puede llamar directamente a *.flatMap { }* en una colección, así como *.filter{ }*, *.map{ }*, y un largo etcétera.
- Opcionales: Están integrados en el lenguaje en Kotlin, basta con declarar el tipo de una variable terminando en un símbolo de interrogación ?.
- Es más seguro. Los desarrolladores de Kotlin integraron en su semántica principios que evitan diversos errores comunes en la ejecución del programa. Permite la implementación de las mismas tareas con menor tiempo y esfuerzo. Cuando el código es más sencillo y fácil de entender es mucho más difícil cometer errores y son más fáciles de detectar. así que disminuyen considerablemente.

3.3 DISEÑO

En los siguientes apartados se detallará el recorrido que un usuario puede realizar dentro de la aplicación y una muestra de algunas pantallas tanto para la skill de Alexa como para la aplicación de Android.

3.3.1 Flujo de ejecución

Basándose en los objetivos, la aplicación deberá tener dos caminos muy bien definidos. Un camino será la utilización de un grafo para la explicación del problema de los caminos mínimos y otro camino será todo el recorrido que ha de hacer el usuario respondiendo preguntas para comprobar que ha entendido cómo se resuelve el problema y lo resuelve correctamente. Además, el usuario debe en

todo momento poder consultar el algoritmo en pseudocódigo y ver la pantalla de créditos, por lo que estas dos pantallas estarán fuera de ambos flujos pero accesible en cualquier momento. También, el usuario puede comenzar la explicación en cualquier momento y en caso de fallo en cualquier paso se le comunicará un mensaje personalizado para la pantalla en la que se encuentre.

Con todo esto, el diagrama del flujo de ejecución simplificado quedaría de la siguiente manera:

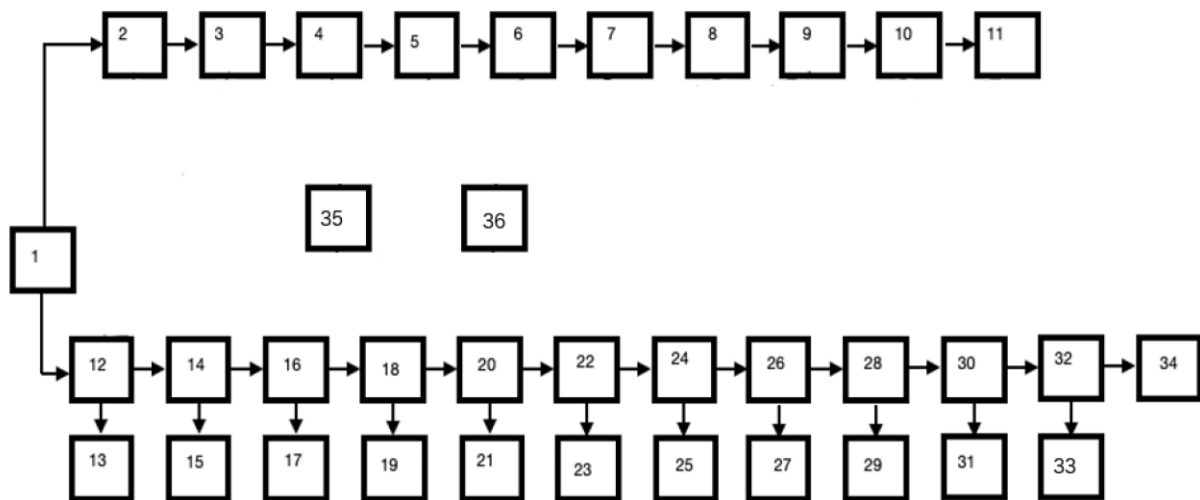


ILUSTRACIÓN 15. FLUJO EJECUCIÓN SKILL

El diagrama presentado en la ilustración 20 es un diagrama simplificado del flujo completo de navegación del usuario dentro de la aplicación, por lo que a continuación se detalla cada una de las pantallas y el flujo completamente.

- Pantalla 1. Es la pantalla de inicio. En ella se describe el funcionamiento de la skill y se detalla qué debe decir el usuario para acceder a las distintas pantallas accesibles desde la misma. A esta pantalla el usuario únicamente puede acceder cuando inicia la skill.
- Pantallas de la 2 a la 11. Estas pantallas forman parte del camino en el que la aplicación le explica al usuario cómo se resuelve el problema de los caminos mínimos mediante un ejemplo. Para acceder a la pantalla número 2, el usuario puede encontrarse en cualquier lugar de la aplicación y decir “comenzar explicación”, una vez haya comenzado la skill a explicar cómo resolver el problema, el usuario puede navegar por las distintas explicaciones diciendo “siguiente” o “continuar”. Si el usuario se encuentra dentro de alguna de estas pantallas y decide ir a alguna otra fuera de este flujo, su progreso se guardará para cuando el usuario desee volver a este camino volver a la pantalla en donde se encontraba anteriormente.

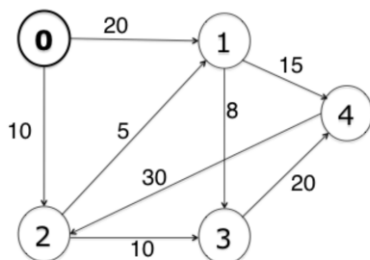
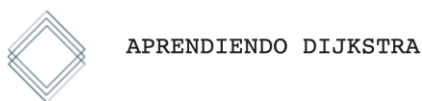
- Pantallas de la 12 a la 34. Estas pantallas comprenden la parte de la aplicación correspondiente al camino en el cual el usuario tiene que poner a pruebas sus conocimientos mediante la correcta contestación a preguntas dadas por la skill. Para todas estas pantallas, la progresión del usuario estará almacenada y Se puede dividir en 4 partes:
 - Pantalla 12. Se describe el problema a resolver. En ella se le presenta al usuario el grafo que debe resolver y se le formula la primera pregunta. A ella se puede acceder desde cualquier pantalla cuando el usuario dice “resolver problema”.
 - Pantallas pares de la 14 a la 32. Estas pantallas corresponden al caso de acierto a la pregunta formulada en la anterior pantalla par. En estas pantallas se le indica al usuario que ha resuelto correctamente la pregunta formulada y se le formula una nueva pregunta correspondiente al momento en el que se encuentre dentro del camino de resolución de problemas. A estas pantallas se puede acceder estando en las 2 pantallas anteriores a ella y respondiendo correctamente a la pregunta formulada por la anterior pantalla par.
 - Pantallas impares de la 13 a la 33. Estas pantallas corresponden a las pantallas para los casos en los que el usuario responda erróneamente a la pregunta formulada por la pantalla inmediatamente anterior a ellas. Estas pantallas le comunicarán al usuario que no ha respondido correctamente a la pregunta formulada y además les dará algún consejo sobre cómo responder correctamente de nuevo. Se accede a ellas estando el usuario en la pantalla inmediatamente anterior a ellas y dando una respuesta fallida a la pregunta formulada.
 - Pantalla 34. Esta pantalla corresponde a la última pantalla dentro de la resolución del problema. En ella, se le comunica al usuario que ha resuelto correctamente todo el problema y se le da las gracias por utilizar la skill. A esta pantalla se puede acceder estando en las 2 pantallas anteriores a ella y respondiendo correctamente a la pregunta formulada en la pantalla 32.
- Pantalla 35. Esta pantalla muestra el algoritmo Dijkstra en pseudocódigo. Se puede acceder a ella desde cualquier pantalla diciendo “ ver algoritmo”. Para salir, el usuario tiene que decir la palabra “volver” y regresará a la pantalla en la que se encontraba antes de acceder a la pantalla 35.
- Pantalla 36. Esta pantalla muestra los datos del desarrollador de la aplicación. Se puede acceder a ella diciendo “ver créditos” estando en cualquier pantalla de la skill. Para salir de ella, el usuario tiene que decir la palabra “volver” y volverá a la pantalla en la que se encontraba anteriormente.

3.3.2 Ejemplos de la interfaz

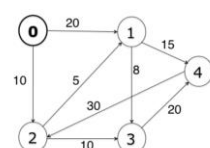
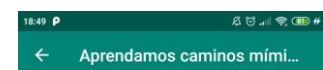
En las siguientes líneas se expondrán algunos ejemplos de pantallas descritos detalladamente. De cada pantalla se explicará a qué pantalla corresponde, qué frases puede decir el usuario para activarlas, el mensaje que dirá Alexa y cómo quedará en las pantallas del dispositivo que utilizará Alexa y en un dispositivo con sistema operativo Android.

Pantalla 2:

- **Se inicia** desde cualquier pantalla y cuando el usuario pronuncia una de las siguientes frases:
 - “comenzar explicación”
 - “explicación”
 - “dame la explicación”
 - “quiero la explicación”
 - “oír explicación”
- **Mensaje que dice Alexa:** “El problema consiste en calcular el camino mínimo de un nodo al resto de nodos del grafo. En este caso de ejemplo, el nodo del que partimos es el nodo 0, así que debemos calcular las distancias entre el nodo 0 y el resto de nodos. En cada paso debemos ir calculando los caminos mínimos parcialmente. Para ello vamos marcando el conjunto de nodos candidatos, que son aquellos nodos de los que aún no hemos partido, y las distancias desde el nodo 0 al resto de nodos. Si desea volver a escuchar esta explicación diga, repetir, y en caso de querer continuar con la explicación, diga, siguiente...”
- **Pantallas:**



PASO	NODO SELECCIONADO (N)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	0		



PASO	NODO SELECCIONADO (N)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	0		

- En un dispositivo con Alexa.

En la app para Android

ILUSTRACIÓN 16. INTERFAZ PANTALLA 2

Pantalla 18. Esta pantalla corresponde al caso en el cual el usuario responde correctamente a la pregunta sobre cuál es el nodo seleccionado para el paso 1.

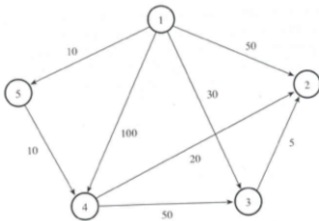
Se inicia cuando el usuario se encuentra en la pantalla 16 o 17 y diciendo una de las siguientes frases:

- “la solución es cinco”
- “la respuesta es cinco”
- **Mensaje que dice Alexa:** “Muy bien, Partimos del nodo 5 para el paso uno. ¿Cuáles serían los nodos candidatos que nos quedan disponibles? ”
- **Pantallas**

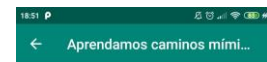


APRENDIENDO DIJKSTRA

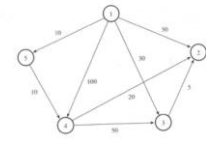
Calcular el camino mínimo de un nodo al resto de nodos



PASO	NODO SELECCIONADO (N)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	1	{2, 3, 4, 5}	{50, 30, 100, 10}
1	5		



Aprendiendo Dijkstra



PASO	NODO SELECCIONADO (N)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	1	{2, 3, 4, 5}	{50, 30, 100, 10}
1	5		

- En un dispositivo con Alexa.

En la app para Android

ILUSTRACIÓN 17. INTERFAZ PANTALLA 18

3.4 Implementación

En los siguientes apartados se describirá la implementación de cada una de las partes de la aplicación. Primero se describirán las partes pertenecientes a la skill, a continuación se describirá el desarrollo de la aplicación Android y por último se detallará como se ha realizado la unión entre ambas aplicaciones.

3.4.1 Interfaz de voz de usuario.

La interfaz de voz de usuario es la parte encargada de interpretar las palabras del usuario y transformarlas en intenciones las cuales ejecutarán código en la aplicación. Cada intención se activa mediante una serie de frases. Una intención puede tener varias frases que la activen, pero una frase no puede pertenecer a varias intenciones puesto que la aplicación únicamente puede ejecutar una única intención en cada interacción del usuario con la skill.

Para desarrollar una skill, Amazon, a través de Amazon Developer Console da la oportunidad de crear intenciones con dos herramientas distintas. A través de un editor gráfico o mediante la edición de un objeto JSON. El desarrollo de este proyecto, se ha realizado mediante el editor gráfico.

Para crear una intención mediante Amazon Developer Console, basta con indicar un nombre para identificar la intención y unas frases para activar dicha intención. Un ejemplo de cómo crearlas puede observarse en la siguiente imagen:

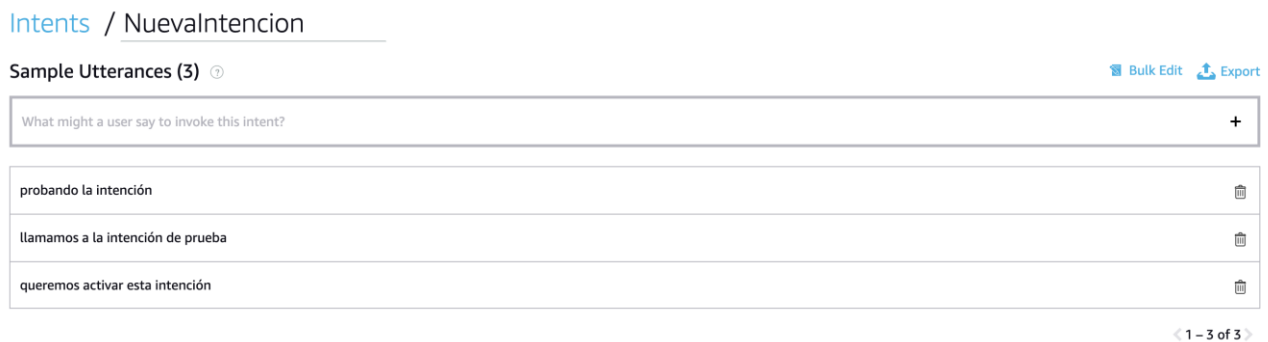


ILUSTRACIÓN 18. INTENCIÓN DE PRUEBA

Para esta skill se han desarrollado un total de 23 intenciones que pasan a ser descritas en la siguiente tabla:

Tabla Intenciones

Nombre	Se activa con	Sirve para
ExplicacionIntent	Comenzar explicación Oír explicación ...	Acceder a la pantalla dónde se enuncia el problema que se enseñará al usuario
ContinuarIntent	Siguiente Continúa ...	Continuar la navegación dentro de las pantallas en las que se explica cómo resolver el algoritmo
AlgoritmoIntent	Muestra el algoritmo Ver algoritmo ...	Acceder a la pantalla en la que se muestra el algoritmo Dijkstra en pseudocódigo.
CreditosIntent	Muestra los créditos Quiero ver los créditos ...	Acceder a la pantalla en dónde se muestran los créditos de la skill
volverIntent	Volver Quiero volver ...	Volver desde las pantallas de algoritmo y créditos a la que se encontrase el usuario anteriormente
silenciarIntent	Para Silencio ...	Silenciar la explicación que esté dando la skill de Alexa sin salir de la misma

Tabla Intenciones

Nombre	Se activa con	Sirve para
ResolverIntent	Resolver problema Resuelvo ...	Acceder a la pantalla en la que comienza la descripción el problema propuesto
PasoInicioDistanciasIntent PasoDosNodoIntent ...	La solución es ... La respuesta es ...	Detectar que el usuario ha dicho una respuesta correcta para el momento en el que se encuentra dentro de la resolución del problema

ILUSTRACIÓN 19. TABLA DE INTENCIONES

3.4.2 Estructura de directorios del back-end

En el back-end se concentra todo el código que manejará la aplicación a excepción de la interfaz de voz de usuario, por lo que es muy importante tener una buena estructura de cara a que la aplicación sea entendible y mantenible en cuanto a código se refiere. Para que esto sea posible se ha dividido el proyecto en varios directorios explicados a continuación:

- **Handlers.** En este directorio se encuentran todos los manejadores. Cada manejador tiene que contener el código necesario para su ejecución.
- **Textos:** Dentro de este directorio se encuentran todos los mensajes que Alexa comunicará al usuario. Se han desarrollado en un directorio distinto al manejador para que su acceso y posible edición sea más fácil de cara a nuevos desarrollos. Se han dividido en 2 funciones. Por un lado está la función “firstTime”, que se llamará en el caso de que se acceda a la pantalla por primera vez y “volverIntent” que se llamará en el caso de que se haya accedido a la pantalla a través de la intención “volver”.
- **Apl y data.** En estos directorios se encuentra el código que necesitará cada manejador para poder mostrar por pantalla la imagen deseada por cada uno de ellos.

- Utils. Directorio que contiene código utilizado por todos los manejadores, de esta manera se evita la repetición de código.

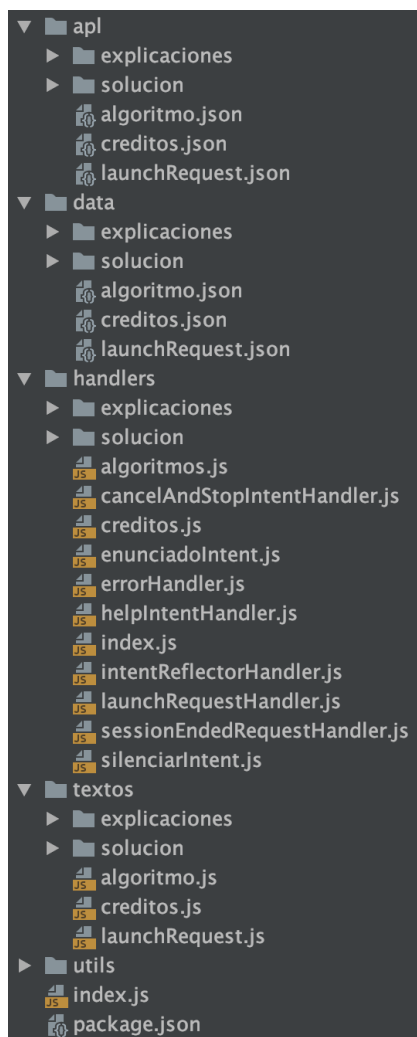


ILUSTRACIÓN 20: ESTRUCTURA DE DIRECTORIOS DEL BACKEND

3.4.3 Implementación del back-end.

Una vez que la aplicación ha interpretado las palabras del usuario y las ha transformado a una intención, debe saber qué manejador utilizará y por tanto qué acciones debe realizar. Para ello, cada manejador tiene que exportar dos funciones cuya descripción es:

- CanHandle. En esta función se realizarán las comprobaciones necesarias para saber si el manejador es el adecuado para la intención o no. Si cumple las condiciones realizará la siguiente función y en caso contrario seguirá con las comprobaciones de los manejadores restantes.
- Handler. Esta función se ejecutará en el caso de que las comprobaciones de la función anterior hayan sido correctas y contendrá todo el código que se desee ejecutar.

Para que el desarrollo de los manejadores sea lo más sencillo y mantenible posible se han realizado una serie de funciones comunes para todos ellos. Estas funciones se encuentran dentro del directorio “Utils” y su funcionalidad en detalle se podría describir como:

- `getSpeakOutput`: Devolverá el mensaje que transmitirá Alexa al usuario, correspondiente a la pantalla que se le haya indicado por parámetro. Además, también recibirá por parámetro el “handlerInput” para que la función compruebe si se trata de la primera vez que se accede a la intención indicada o si se accede a través de la intención “volver”.
- `setScreenAttributeWithValue`. Alexa permite almacenar información en unos atributos persistentes en cada sesión de la aplicación. Con esta función se le añade el valor recibido como parámetro en un atributo llamado “screen”, el cual se usará para identificar en qué pantalla se encuentra en cada momento el usuario
- `attributeScreenIs`. Esta función devolverá verdadero en el caso de que el valor recibido por parámetro sea igual al valor que se encuentre en el atributo de sesión “screen” y falso en el caso de que no lo cumpla.
- `intentNameIs`. Esta función devolverá verdadero en caso de que el nombre recibido por parámetros sea igual que el nombre de la intención de la petición actual y falso en caso contrario.
- `requestTypeIs`. Esta función devolverá verdadero en caso de que el tipo de la petición de la skill sea igual al recibido por parámetro y falso en caso contrario. Al igual que las otras dos funciones anteriores de comparación, también tiene que recibir el objeto “handlerInput” para poder acceder al tipo de petición realizada.
- `showAPLWithScreen`. Esta función comprueba si el dispositivo que está utilizando Alexa dispone de una pantalla. En caso afirmativo, muestra en la pantalla el valor recibido por parámetro en la función.
- `sendMessageToWebsocket`. Esta función permite enviar el mensaje recibido por parámetro a través del websocket configurado. Levanta el websocket, envía el mensaje y cierra la conexión.

Para el desarrollo de la skill se han implementado tantos manejadores como número de pantallas disponibles. Además de estas, también se han implementado manejadores para las respuestas erróneas dadas por un usuario y manejadores para las intenciones por defecto. En total se han implementado 43 manejadores.

3.4.5 Imagen mostrada por pantalla

Para crear las imágenes mostradas en el dispositivo utilizado por Alexa, se ha utilizado la herramienta llamada Display contenida dentro de Alexa Developer Console. Para todos los diseños se ha seguido el mismo patrón:

- En la parte superior se colocará el título de la aplicación.
- En el cuerpo se colocarán dos imágenes:
 - A la izquierda el grafo utilizado.
 - A la derecha la tabla de contenidos.

El diseño es adaptativo para cualquier dispositivo. Cuando la pantalla de un dispositivo es demasiado pequeña para albergar dos imágenes, se mostrará únicamente la imagen del grafo utilizado.

3.4.6 Implementación de la aplicación Android

Para este proyecto, se desea que toda la lógica la contenga la skill desarrollada, quedando de esta manera la aplicación como una aplicación basada en la muestra de una serie de imágenes pero sin una posible interacción por parte del usuario.

La aplicación Android, por un lado, implementará un webSocket. Este webSocket será el encargado de la comunicación entre la skill y la aplicación móvil. Este webSocket permitirá recibir un mensaje por parte de la skill y la aplicación será capaz de procesarlo y con ello decidir qué pantalla a de mostrarle al usuario. Este webSocket se iniciará al arrancar la aplicación y se mantendrá a la espera de recibir mensajes hasta que la aplicación se cierre, momento en el que se cerrará.

En cuanto a las pantallas desarrolladas en la aplicación móvil, se ha utilizado la herramienta que provee Android Studio llamada “preview”. Gracias a esta herramienta se puede crear pantallas de aplicaciones de una manera gráfica e insertando imágenes y textos a un panel en blanco. Estas pantallas, gráficamente, se corresponderán con las pantallas que mostraría la skill si el dispositivo sobre el que corre Alexa dispusiese de una pantalla, dándole de esta manera al usuario una misma experiencia de uso. Estas pantallas están formadas por el título en la parte superior de la misma, seguidas en su parte inferior por dos imágenes correspondientes al grafo utilizado y a la tabla de contenidos.

Se han desarrollado tantas pantallas como las que contiene la skill, además de una pantalla de bienvenida y una pantalla para el caso de que la aplicación obtenga un error y no sea capaz de recibir

correctamente el mensaje a través del webSocket. Todas las pantallas de la aplicación disponen de un botón para volver a la pantalla principal, siendo esta la única interacción posible con la aplicación.

3.4.7 Simple-websockets-chat-app.

Para realizar la unión entre la skill utilizada por Alexa y la aplicación Android, se ha utilizado una librería disponibilizada por Amazon, a través del repositorio de aplicaciones de AWS, llamada “simple-websockets-chat-app”. Esta librería contiene una serie de funcionalidades alojadas en AWS que pasarán a ser descritas en los siguientes apartados.

La primera funcionalidad que se describe es AWS API Gateway. Será el encargado de disponibilizar un API de WebSocket. A este API llamarán tanto la aplicación como la skill. Dicho API, expondrá 3 rutas:

- **“connect”**: Ruta obligatoria. Será llamada cuando se realice una conexión con webSocket. Esta ruta llamará a una función Lambda llamada “onConnectFunction”
- **“disconnect”**: Segunda y última ruta obligatoria. Será llamada cuando se realice una desconexión con webSocket. Cuando esta ruta sea accionada, llamará a una función nombrada como “onDisconnectFunction”
- **“sendMessage”**: Esta ruta será llamada cuando la skill desee comunicarse con la aplicación. Esta ruta puede recibir un mensaje que será enviado a una función lambda llamada “sendMessageFunction”.

Además de AWS API Gateway, la librería “simple-websockets-chat-app” disponibiliza 3 funciones lambda las cuales pasarán a ser descritas a continuación:

- **“onConnectFunction”**. Esta función se encargará de guardar en una base de datos alojada en DynamoDB un identificador del cliente que ha llamado a la ruta “connect” del API. De este modo, se tendrá un registro de qué clientes han realizado una conexión con el webSocket para, de este modo, crear una conversación entre ellos.
- **“onDisconnectFunction”**. Esta función será llamada cuando un cliente llame a la ruta “disconnect” del API. Se encargará de obtener el identificador del cliente y eliminarlo de la base de datos alojada en DynamoDB. De este modo, el cliente dejará de estar presente en la comunicación que mantenía.

- **“sendMessageFunction”**. Cuando esta función sea llamada desde la ruta “sendmessage” del API, obtendrá el mensaje recibido, accederá a la base de datos, consultará todos los clientes conectados y enviará el mensaje a todos ellos.

Por último, la librería “simple-websockets-chat-app” disponibiliza una tabla de base de datos alojada en DynamoDB cuya función es ir almacenando cada uno de los clientes que se han conectado al WebSocket. Gracias a esta funcionalidad, se puede mantener un registro de qué usuarios hay en ese momento en la conversación de WebSocket.

3.4.8 Conexión entre skill y aplicación.

Tanto la skill como la aplicación se conectan como clientes al API de Websocket. Cada una de ellas tiene una serie de particularidades que se describen a continuación.

Cuando un usuario abre la aplicación Android, esta realiza una llamada a la ruta “connect” del API que a su vez llamará a la función lambda “onConnectFunction” que realizará el almacenamiento en la base de datos de la información necesaria para identificar un nuevo cliente en la conversación de WebSocket. Mientras la aplicación esté abierta por el usuario, su estado será el de escucha. Cuando se reciba un nuevo mensaje, es decir, cuando un cliente del API llame a la ruta “sendMessage” y este a su vez llame a la función lambda “sendMessageFunction” que obtendrá de la base de datos la sesión de la aplicación y le enviará el mensaje recibido, la aplicación llamará a una función que se encargará de comprobar el mensaje recibido y mostrar la pantalla correspondiente a dicho mensaje. Por último, cuando un usuario cierra la aplicación, se realizará una llamada a la ruta “disconnect” del API, la cual realizará una llamada a la función “onDisconnectFunction” que eliminará de la base de datos todo lo relacionado con la sesión de la aplicación.

Por otro lado, se encuentra la skill. A diferencia de la aplicación, la skill no puede mantener sesiones abierta entre diferentes interacciones, por lo que la solución acordada ha sido, para cada interacción por parte del usuario, llamar a la ruta “connect” del API, realizar el envío del mensaje correspondiente con la llamada a “sendMessage” y cerrar la sesión establecida con el WebSocket mediante la llamada a la ruta “disconnect”.

El esquema del funcionamiento entre los distintos componentes que intervienen en el proyecto puede verse en la siguiente ilustración:

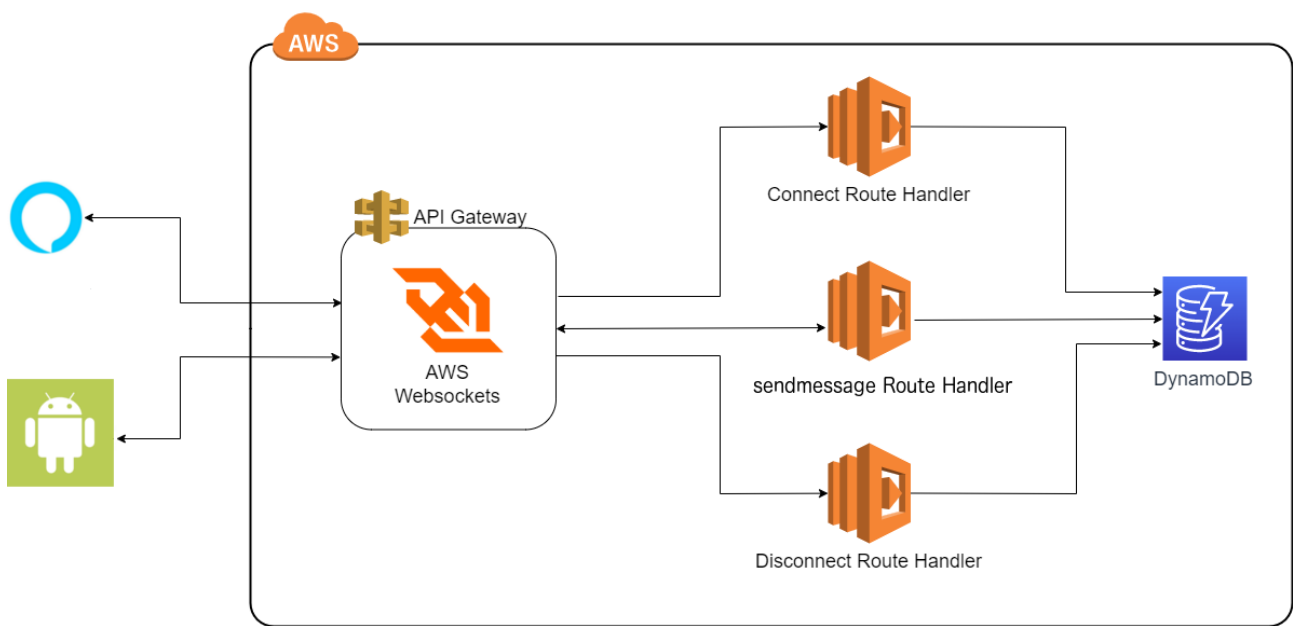


ILUSTRACIÓN 21. DISEÑO COMUNICACIÓN SKILL-APP

4 Pruebas

En este apartado, se describirán las pruebas realizadas para la comprobación del correcto funcionamiento de la aplicación desarrollada. Se dividirá en dos apartados. Por un lado se describirán las pruebas manuales realizadas por parte del desarrollador y por otro lado se describirán las pruebas realizadas por distintos usuarios de la aplicación, así como el cuestionario realizado posteriormente y los resultados obtenidos a través del mismo.

4.1 Pruebas manuales

Para comprobar el correcto funcionamiento de la aplicación, se han realizado una serie de pruebas relacionadas con los objetivos marcados en el proyecto. Cada tabla de prueba contendrá el objetivo que tiene cada prueba en su realización, pantalla desde la cual parte la prueba, frase formulada por el usuario, hacia que pantalla debería ir y el resultado de la prueba realizada.

Prueba 1

Objetivo	Acceder al apartado de explicación
Pantalla actual	1
Frase formulada por el usuario	Comenzar explicación
Pantalla esperada	2
Resultado	OK

Prueba 2

Objetivo	Acceder al apartado de resolución del problema
Pantalla actual	1
Frase formulada por el usuario	Resolver problema
Pantalla esperada	12
Resultado	OK

Prueba 3

Objetivo	Acceder al pseudocódigo
Pantalla actual	7
Frase formulada por el usuario	Ver algoritmo
Pantalla esperada	35
Resultado	OK

Prueba 4

Objetivo	Acceder a los créditos
Pantalla actual	34
Frase formulada por el usuario	Ver créditos
Pantalla esperada	36
Resultado	OK

Prueba 5

Objetivo	Comprobar la acción de volver
Pantalla actual	18
Frase formulada por el usuario	Ver algoritmo
Pantalla esperada	35
Frase formulada por el usuario	Volver
Pantalla esperada	18
Resultado	OK

Prueba 6

Objetivo	Comprobar el acceso a la explicación desde la solución
Pantalla actual	24
Frase formulada por el usuario	Comenzar explicación
Pantalla esperada	2
Resultado	OK

Prueba 7

Objetivo	Silenciar la explicación
Pantalla actual	17
Frase formulada por el usuario	Silencio
Pantalla esperada	17
Resultado	OK

Prueba 8

Objetivo	Comprobar la detección de una respuesta correcta
Pantalla actual	16
Frase formulada por el usuario	La solución es cinco
Pantalla esperada	18
Resultado	OK

Prueba 9

Objetivo	Comprobar la detección de una respuesta incorrecta
Pantalla actual	16
Frase formulada por el usuario	La solución es cuatro
Pantalla esperada	17
Resultado	OK

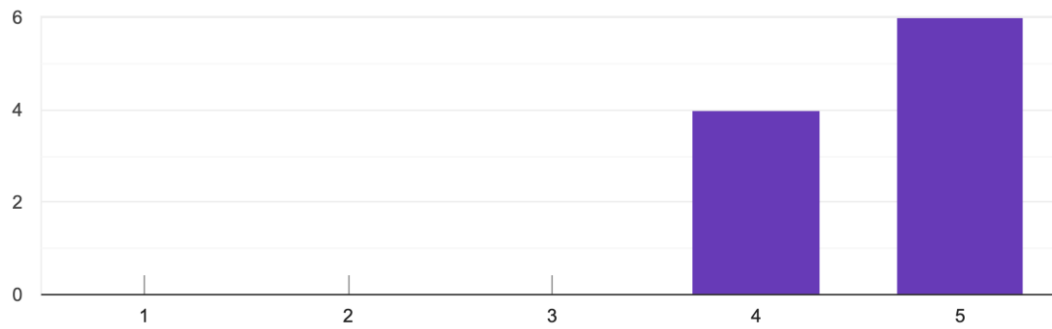
Prueba 10

Objetivo	Comprobar la navegación dentro de la explicación
Pantalla actual	2
Frase formulada por el usuario	Continuar
Pantalla esperada	3
Resultado	OK

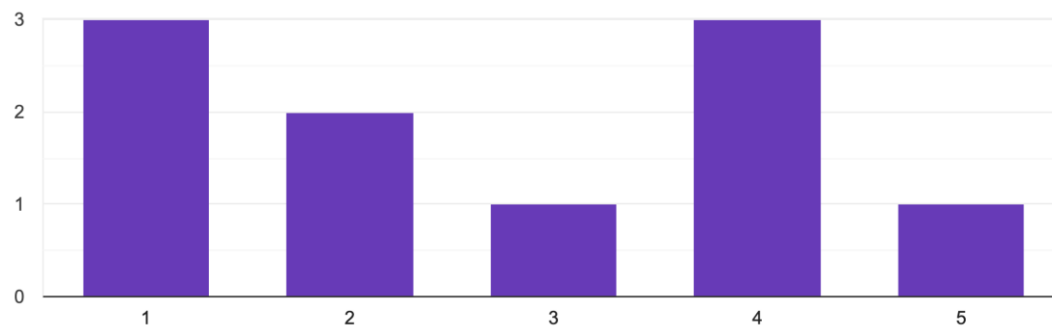
4.2 Pruebas con usuarios reales

Para comprobar el correcto funcionamiento de la aplicación, se han realizado una serie de pruebas a un total de 10 usuarios. Para realizar estas pruebas, se les ha dado un dispositivo Amazon Echo y un dispositivo Android, ambos con la skill y la aplicación correctamente configuradas. La única pauta que se les ha dado a los usuarios, es que debían utilizar la skill durante un máximo de 20 minutos y al finalizar realizar un cuestionario. Las respuestas al cuestionario y las conclusiones a las que se ha llegado se describen a continuación.

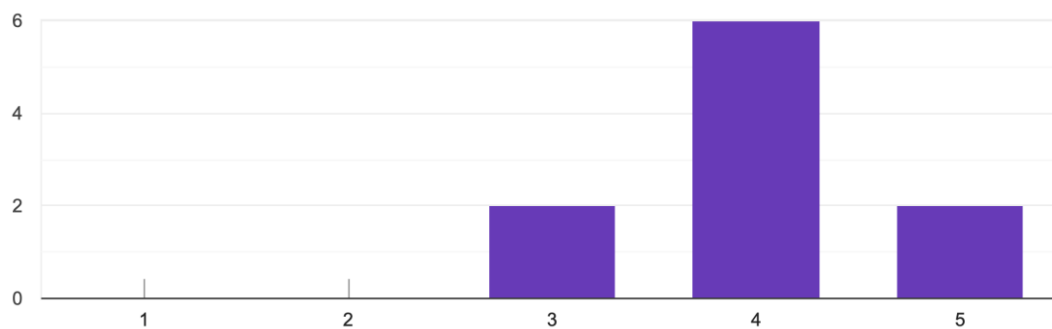
¿Cómo valorarías la skill?
10 respuestas



¿Qué nivel de conocimiento tenías antes del uso de la skill con respecto a un algoritmo voraz?
10 respuestas

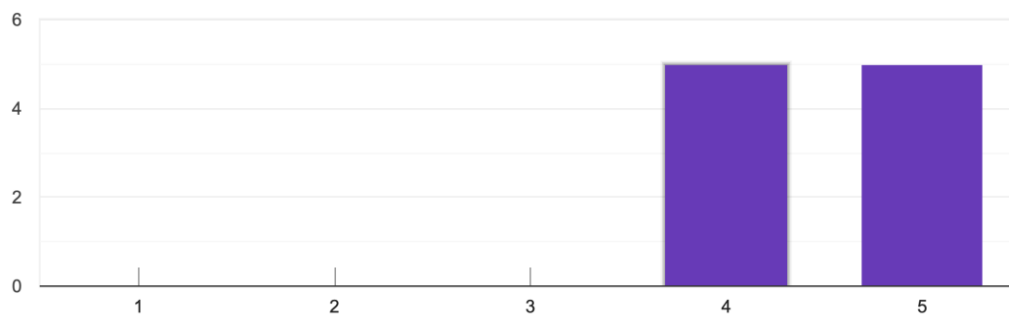


¿Cuál crees que es tu nivel de conocimiento actual?
10 respuestas



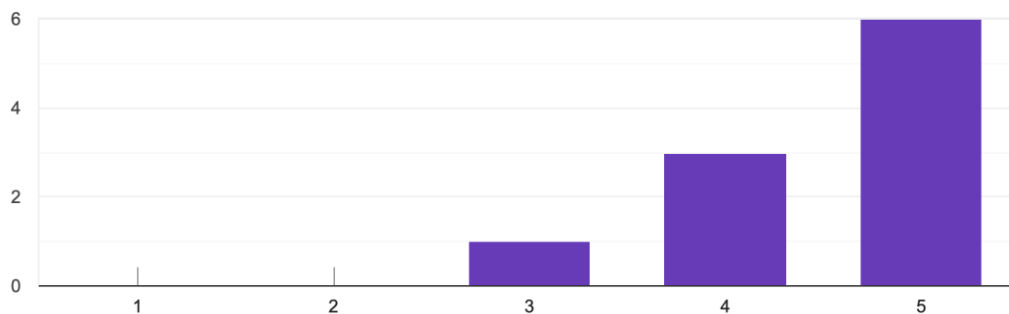
¿Los conceptos explicados en la skill se entienden correctamente?

10 respuestas



¿Te ha parecido divertida esta manera de resolver un problema?

10 respuestas



Como se puede observar, tras la prueba realizada a estos diez usuarios se puede llegar a las siguientes conclusiones:

- El 100% de los usuarios ha valorado la skill con unos 4 o 5 puntos, por lo que quiere decir, que en líneas generales tiene una gran aceptación por parte de los usuarios.
- Se puede observar que había una gran variedad de conocimiento acerca de la resolución de un algoritmo voraz y que, después de la utilización de la skill, todos los usuarios consideran haber mejorado sus conocimientos previos, por lo que resulta una buena herramienta de aprendizaje.
- Todos los usuarios han puntuado con un 4 o un 5 que han entendido correctamente las explicaciones recibidas, por lo que se puede deducir que la skill comunica bien al usuario los conceptos dados.
- El 90% de los usuarios le ha parecido divertida el uso de la aplicación, por lo que cumple con el objetivo de atraer a usuarios a la utilización de la skill.

5 Conclusiones

En este apartado se expondrán las conclusiones a las que se ha llegado tras la realización del proyecto. Comenzará exponiéndose unas conclusiones generales acerca del proyecto, continuando con unas conclusiones personales y por último se describirán unos posibles trabajos futuros.

5.1 Conclusiones generales

Este proyecto nace con el objetivo principal de poder enseñar, de una manera novedosa, un algoritmo voraz al mayor número de personas posibles. Para ello se basa en una serie de objetivos específicos que se han cumplido satisfactoriamente como se expone a continuación:

- **Crear y utilizar un grafo a modo de ejemplo para enseñar al usuario a resolver el problema del algoritmo voraz Dijkstra.** Este objetivo se ha cumplido realizando las pantallas de la número 2 a la número 11.
- **Crear y utilizar un grafo mediante el cual el usuario pueda ir resolviendo el problema del algoritmo voraz Dijkstra mediante la respuesta a preguntas formuladas por la aplicación.** Este objetivo se ha cumplido realizando las pantallas de la número 12 a la pantalla número 34
- **Presentar al usuario el estado actual del grafo en cada momento.** Este objetivo se ha cumplido colocando una imagen del estado del grafo en cada pantalla mostrada al usuario.
- **Visualización por parte del usuario del algoritmo Dijkstra en pseudocódigo.** Para este objetivo se ha realizado la pantalla número 35
- **Almacenar el estado de la resolución del problema para el momento en el que el usuario desee realizar alguna consulta y cambiar de pantalla y posteriormente volver al momento en el que se encontraba.** Para este objetivo se ha ido almacenando la progresión llevada por el usuario y se ha habilitado una intención para que el usuario pueda volver a la pantalla en la que se encontraba.
- **Poder mostrar información por pantalla.** Para este objetivo se ha realizado una aplicación para el sistema operativo Android, de tal manera que si un usuario está utilizando Alexa en un dispositivo que no posee una pantalla, la información será enviada a dicha aplicación

- **Silenciar la explicación en cualquier momento.** Para la realización de este objetivo se ha desarrollado una intención mediante la cual, el usuario únicamente tiene que decir la palabra “silencio” y la skill entenderá que debe parar la explicación que estaba exponiendo.

Observando la realización de todos los objetivos, así como el feedback dado por los pocos usuarios que, hasta ahora, han utilizado la aplicación, puede concluirse que cumple con su cometido de manera satisfactoria por lo que es totalmente apto para que tanto alumnos de informática como usuarios no técnicos puedan utilizar la aplicación y aprender de una manera novedosa un algoritmo voraz.

Como conclusión, creo que este proyecto tal y como está planteado puede aportar mucho al tema educativo, que es realmente el propósito del mismo, pero además, creo que como tiene futuro es para abrir el camino para nuevos proyectos similares. El poder comunicar una skill de un asistente virtual con un dispositivo móvil es un camino que abre muchas puertas. No sé si en la universidad se podrá aplicar, creo que en cierta medida sí, pero sobre todo lo veo para una educación en edades más tempranas. Se podrían hacer preguntas de suma a un niño y no tener que esperar hasta el día siguiente para que el profesor le responda si está bien o mal, se podría enseñar gramática y si ciertas palabras tienen o no acento a un alumno de primaria, se podrían practicar definiciones con un alumno de secundaria... Le veo futuro, como ese compañero, madre o padre, que hemos tenido todos y al que le decíamos “pregúntame, a ver si me lo sé”.

Los asistentes virtuales, son uno de los productos estrella de nuestros días. El problema que tienen, es que, actualmente, no hay grandes aplicaciones para un uso continuado, por lo que corren el peligro de que acaben siendo un dispositivo más que cae en el olvido como le ha pasado a tantos otros. Para solventar este problema, hacen falta aplicaciones del estilo a la descrita en este proyecto, aplicaciones innovadoras, aplicaciones capaces de salvar las limitaciones de la pantalla y hacerlas aún más potentes al comunicarse con el móvil.

5.2 Conclusiones personales.

Este proyecto, para mí, a nivel profesional, ha sido muy enriquecedor. Gracias a él, he podido unir lo aprendido en la universidad junto con lo aprendido en mi etapa laboral, además de aprender una gran cantidad de nuevas herramientas que sin duda servirán para desarrollar mi carrera profesional en un futuro. Dicho esto, me gustaría desgranar lo aprendido por partes.

En cuanto a la metodología de trabajo iterativa e incremental me ha aportado muchísimo el ir viendo cómo no tiene por qué estar la aplicación completamente desarrollada al comienzo del proyecto, si no que se puede partir de una idea e ir ampliándola conforme se van realizando pequeños entregables. Al comienzo del proyecto no podría imaginarme que acabaría realizando un WebSocket que

permitiese comunicar la skill con la aplicación, pero gracias al ir entregando poco a poco los desarrollos realizados, se observó la necesidad de crearla.

Desarrollar una skill de Alexa ha sido muy enriquecedor ya que, para mí, Alexa era únicamente ese asistente virtual al cual le preguntaba qué tiempo hacía o qué hora era, pero nunca me planteé que podría aprender a desarrollar una skill. Además, tras haber realizado el ciclo completo del desarrollo, he podido ver lo importante que son cada uno de los perfiles en cualquier aplicación, ya que por ejemplo, en la aplicación desarrollada podrían intervenir desarrolladores de back-end, desarrolladores de front-end e incluso lingüistas para la parte de entendimiento y respuesta por parte de la skill.

En cuanto a la realización de la aplicación Android me ha aportado muchísimo el investigar y estudiar un lenguaje, para mi desconocido, como es Kotlin. Hasta ahora, había realizado aplicación para Android en Java, pero decidí que era el momento de utilizar Kotlin y gracias a ello he podido aprender un nuevo lenguaje de programación.

Dicho todo lo anterior, creo, que la parte que más me ha podido aportar a nivel personal en este proyecto, ha sido todo el ecosistema de AWS. Gracias a este proyecto he podido aprender muchas de las herramientas que ofrece AWS, algunas las he utilizado para este proyecto y otras no. Este proyecto me ha ayudado para ver que cualquier aplicación puede ser desplegada en AWS con unos costes bajísimos y con un potencial enorme.

A un nivel más personal y alejándome de la parte más profesional, he de reconocer que ha sido un proyecto muy duradero en el tiempo. Al encontrarme trabajando, sacar tiempo para este proyecto ha sido complicado, por lo que la realización del mismo se ha ido haciendo durante muchos días un corto periodo de tiempo cada día. Además, he ido realizando mucha refactorización al código. Al comienzo del proyecto y debido a la metodología utilizada, se desarrolló el proyecto de una manera, pero a medida que iba creciendo veía como debía sacar factor común de muchas partes de código ya que un 80% del código era similar para todas las pantallas. Otra parte del proyecto que me ha llevado mucho tiempo y por tanto ha retrasado la entrega del mismo, ha sido la realización de las imágenes para las pantallas, ya que al haber diversos tamaños de pantallas tienes que adaptar muy bien las imágenes para cada una de ellas. Nunca ha sido mi fuerte el dibujo y la interfaz gráfica, por lo que ha sido una de las partes que más me ha costado desarrollar. En definitiva, pese al esfuerzo y al tiempo invertido en el proyecto, estoy muy contento con el resultado, con su posible utilidad futura y con todo lo que me ha aportado como persona y como profesional.

5.3 Trabajos futuros

Es verdad que se ha avanzado mucho en este proyecto y se han conseguido grandes pasos, pero para que sea un proyecto el cual pueda utilizar un profesor de universidad y que el conjunto de sus alumnos lo utilicen en clase para aprender a desarrollar un algoritmo Dijkstra, faltan una serie de pasos:

- **Publicación en las tiendas de aplicaciones.** Habría que publicar tanto la skill de Alexa como la app de Android en sus respectivas tiendas de aplicaciones. De esta forma, los usuarios podrían acceder de una manera sencilla a dichas aplicaciones.
- **Desarrollar la aplicación en iOS.** Si bien se ha dicho que la gran cuota de mercado la tiene Android, no es el 100% de los dispositivos móviles, por lo que muchos alumnos en clase no tendrían la posibilidad de poder utilizarlo.
- **Desarrollar la skill para Google Assistance.** Es verdad que Amazon tiene la mayor cuota de mercado en cuanto asistentes virtuales se refiere, pero no posee el 100% de la cuota de mercado. Si se desarrolla para Google Assistance, se tendría una mayor cuota de mercado y así más usuarios podrían utilizar la skill desarrollada.
- **Conseguir establecer conexión skill-app uno a uno.** Actualmente, como se ha visto en la arquitectura de la comunicación, se establece una conexión de una skill con todas las aplicaciones Android que estén conectadas. Como trabajo futuro se tendría que conseguir que una sesión de una skill se comunicase únicamente con una aplicación Android.
- **Mantener persistencia de progreso.** En el estado actual, cuando un usuario cierra la skill pierde su progreso. Habría que almacenar el estado del progreso para que así un usuario no tuviese que empezar de nuevo en la resolución del problema cada vez que abra la skill.

6 Bibliografía

Amazon API Gateway. 2020. Amazon -

<https://docs.aws.amazon.com/es-es/apigateway/latest/developerguide/apigateway-dg.pdf#welcome>

Amazon DynamoDB. 2020. Amazon -

<https://docs.aws.amazon.com/es-es/amazondynamodb/latest/developerguide/dynamodb-dg.pdf>

Simple-websockets-chat-app. 2020. Amazon -

<https://serverlessrepo.aws.amazon.com/applications/arn:aws:serverlessrepo:us-east-1:729047367331:applications~simple-websockets-chat-app>

“Todo el mundo habla de los asistentes virtuales, pero ¿cómo los utilizan realmente los usuarios?”.

2019. Telefónica - <https://aura.telefonica.com/public/papers/04-todo-el-mundo-habla-sobre-asistentes-virtuales-es.pdf>

Cuota de mercado asistentes virtuales. 2020. Canalys - <https://www.canalys.com/newsroom/-global-smart-speaker-market-Q4-2019-forecasts-2020>

Cuota de mercado sistemas operativos en dispositivos móviles. 2020. StatCounter -

<https://gs.statcounter.com/os-market-share/mobile/spain>

Amazon Alexa Developer Console. 2020. Amazon - [https://developer.amazon.com/es-](https://developer.amazon.com/es-ES/docs/alexa/devconsole/about-the-developer-console.html)

[ES/docs/alexa/devconsole/about-the-developer-console.html](https://developer.amazon.com/es-ES/docs/alexa/devconsole/about-the-developer-console.html)

Android Studio. 2020. Android - <https://developer.android.com/studio/intro>

Coste WebStorm. 2020. Webstorm - <https://www.jetbrains.com/es-es/webstorm/>

Tabla asistentes virtuales. 2020. Wikipedia - https://en.wikipedia.org/wiki/Smart_speaker

Anexo

En los siguientes apartados se describirá una guía para la instalación y la utilización de la aplicación descrita en estas páginas por parte de quien lo desee.

Guía de instalación

Actualmente, la skill no puede ser probada debido a que no se encuentra desplegada en la tienda de skills de Amazon y Amazon no permite sacar versiones de prueba de una skill, a menos que esta vaya a ser desplegada para Estados Unidos, Reino Unido o Alemania. Aún no se ha considerado desplegar la skill a la espera de realizar las tareas descritas en el apartado “trabajos futuros”, momento en el cual sería funcionalmente completa. Hasta el momento en el que la skill sea desplegada en la tienda de skills, puede consultarse el código de la misma en el siguiente enlace: <https://github.com/abarragancosto/aprendamosAlgoritmos/tree/master/aprendamosAlgoritmos>

La aplicación desarrollada para el sistema operativo Android, sí puede ser probada, pero como se ha descrito en los apartados anteriores, es funcionalmente dependiente de la skill, por lo que únicamente se podría acceder a su pantalla de inicio. Para su instalación, bastaría con descargar el apk desde este, enlace:

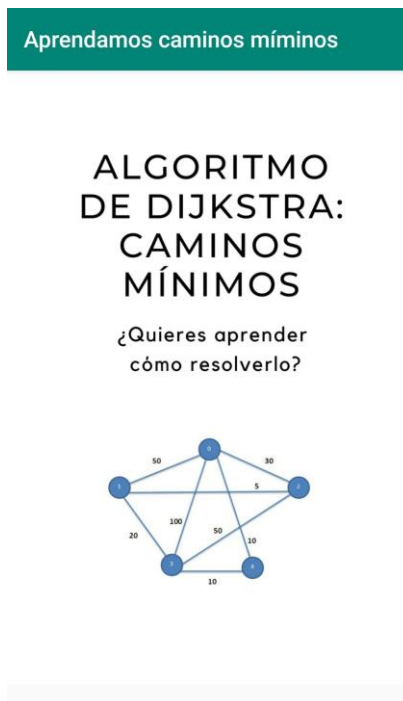
<https://github.com/abarragancosto/aprendamosAlgoritmos/blob/master/aprendamosAlgoritmos.apk>

e instalarla en un dispositivo Android.

Guía de uso

En este apartado se describirá qué pasos debe realizar un usuario para poder probar la skill e ir navegando por las distintas pantallas disponibles.

Para comenzar con la utilización de la skill y la aplicación, primero se debe abrir la aplicación desarrollada para Android. Una vez abierta, tiene que aparecer una pantalla como la que se puede observar en la siguiente imagen:



Para comenzar a utilizar la skill, el usuario debe decir “Alexa, abre aprendamos algoritmos”. Una vez dicho esto, Alexa responderá diciendo “Hola y bienvenido a Aprendamos algoritmos, el asistente que te ayudará a cómo se resuelve un algoritmo voraz Dijkstra. Si aún no sabes en qué consiste y te gustaría aprender a resolverlo prueba a decir, Comenzar explicación. En cambio, si ya eres conocedor de este algoritmo y te gustaría ponerte a prueba e intentar resolver un ejemplo, prueba a decir. Resolver problema. En cualquier momento puedes decir: muestra el algoritmo y podrás ver el algoritmo en pseudocódigo. Además, puedes decir "ver créditos" y aparecerá el autor de la skill.” Además, tanto en la pantalla del dispositivo Android como en la pantalla del dispositivo con Alexa integrada, deberán aparecer las siguientes imágenes:

← Aprendamos caminos mími...

Aprendiendo Dijkstra

PASO	NODO ORIGEN	CANDIDATOS	DISTANCIAS
Paso inicialización	1		
1			
2			
3			

APRENDIENDO DIJKSTRA

Si el usuario desea acceder al grafo de explicación del algoritmo deberá decir “comenzar explicación” a lo que Alexa contestará: “El problema consiste en calcular el camino mínimo de un nodo al resto de nodos del grafo. En este caso de ejemplo, el nodo del que partimos es el nodo 0, así que debemos calcular las distancias entre el nodo 0 y el resto de nodos. En cada paso debemos ir calculando los caminos mínimos parcialmente. Para ello vamos marcando el conjunto de nodos candidatos, que son aquellos nodos de los que aún no hemos partido, y las distancias desde el nodo 0 al resto de nodos. Si desea volver a escuchar esta explicación diga, repetir, y en caso de querer continuar con la explicación, diga, siguiente.”. Además de exponer la explicación, en las pantallas de ambos dispositivos aparecerán las siguientes imágenes:

← Aprendamos caminos mími...

Aprendiendo Dijkstra

PASO	NODO SELECCIONADO (V)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	0		

APRENDIENDO DIJKSTRA

PASO	NODO SELECCIONADO (V)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	0		

54

Para continuar con la explicación de la resolución del grafo, el usuario debe ir diciendo “siguiente”. Cada vez que diga la palabra “siguiente” aparecerá una nueva pantalla y Alexa dictará una nueva explicación.

Si el usuario desea resolver el problema propuesto por la skill, debe decir “resolver problema”, a lo que Alexa contestará “Comencemos con la resolución del problema propuesto. Estamos en el paso de inicialización y partimos del nodo 1. ¿Cuál es el conjunto de posibles candidatos? Una vez sepas la solución, debes decir: la solución es, seguido de tu respuesta. Por ejemplo, si tenemos que decir que el nodo dos es el nodo seleccionado, debemos decir: "la solución es dos". Si vamos a decir el conjunto de candidatos y creemos que los candidatos son los nodos dos, tres cuatro y cinco, debemos decir. La solución es dos tres cuatro cinco. Si queremos decir el conjuntos de distancias, y creemos que las distancias son veinte treinta y cuarenta, debemos decir, La solución es veinte treinta cuarenta. Si deseas que repita algo, prueba a decir repetir”. Además de esto, las pantallas mostradas serán:

← Aprendamos caminos mími...

Aprendiendo Dijkstra

PASO	NODO ORIGEN	CANDIDATOS	DISTANCIAS
Paso inicialización	1		
1			
2			
3			

APRENDIENDO DIJKSTRA

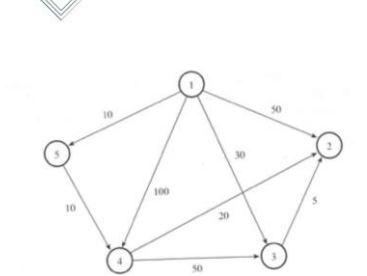
Calcular el camino mínimo de un nodo al resto de nodos

PASO	NODO SELECCIONADO (N)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	1		

En este momento el usuario puede responder correctamente diciendo “la solución es dos tres cuatro cinco” a lo que Alexa le responderá: “Muy bien, este es el conjunto de los candidatos correcto para el paso de inicialización. ¿Sabrías decirme cuales son las distancias correctas para el paso de inicialización?” Y mostrará por las pantallas:

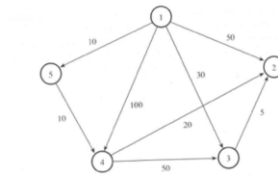


Aprendiendo Dijkstra



PASO	NODO SELECCIONADO (N)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	1	{2, 3, 4, 5}	

Calcular el camino mínimo de un nodo al resto de nodos



PASO	NODO SELECCIONADO (N)	CANDIDATOS (C)	DISTANCIAS (D)
Paso inicialización	1	{2, 3, 4, 5}	

Para la navegación por el grafo de resolución del problema, el usuario deberá ir respondiendo a las preguntas planteadas por Alexa. En el caso en el que el usuario responda correctamente Alexa responderá con un mensaje satisfactorio y cambiará las imágenes y en el caso en el que el usuario responda erróneamente a la pregunta formulada, Alexa le indicará que su respuesta no es correcta, le indicará alguna ayuda y las pantallas mostradas no cambiarán.

Si el usuario desea ver el algoritmo en pseudocódigo, lo que debe decir a Alexa es “ver algoritmo”, a lo que Alexa le contestará con “En la pantalla se muestra el algoritmo de Dijkstra en pseudocódigo” y mostrará por pantalla las siguientes imágenes:



Aprendiendo Dijkstra

Algoritmo de dijkstra:

```
función Dijkstra(L[1..n, 1..n]): matriz [2..n]
    matriz D[2..n]
    {Iniciación}
    C ← {2, 3..n}
    para i ← 2 hasta n hacer D[i] ← L[1, i]
    {bucle voraz}
    repetir n -2 veces
        v ← algún elemento de C que minimiza D[v]
        C ← C \ {v} {e implícitamente S ← S U {v}}
        para cada w ∈ C hacer
            D[w] ← min(D[w], D[v] + L[v, w])
    devolver D
```

```
función Dijkstra(L[1..n, 1..n]): matriz [2..n]
    matriz D[2..n]
    {Iniciación}
    C ← {2, 3..n}
    para i ← 2 hasta n hacer D[i] ← L[1, i]
    {bucle voraz}
    repetir n -2 veces
        v ← algún elemento de C que minimiza D[v]
        C ← C \ {v} {e implícitamente S ← S U {v}}
        para cada w ∈ C hacer
            D[w] ← min(D[w], D[v] + L[v, w])
    devolver D
```


Si el usuario desea ver la pantalla de crédito, debe decir “ver créditos” a lo que Alexa le responderá con el mensaje de “Esta skill ha sido desarrollada por Alberto Barragán Costo” y mostrará las pantallas expuestas a continuación:

Aprendamos caminos mínimos



Aprendiendo Dijkstra

Creado por:
Alberto Barragán Costo
abarragancosto@gmail.com



CREDITOS



Creado por:
Alberto Barragán Costo
abarragancosto@gmail.com