

UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

ARQUITECTURA PARA EL DISEÑO DE APLICACIONES DE
RESERVA DE CITAS UTILIZANDO ALEXA

Trabajo final de investigación aplicada sometido a la consideración de
la Comisión del Programa de Estudios de Posgrado en Computación e
Informática para optar al grado y título de Maestría Profesional en
Computación e Informática

ANDRES MEZA FERNANDEZ

Ciudad Universitaria Rodrigo Facio, Costa Rica

2019

DEDICATORIA

Nadie ha sido más importante para mí durante la elaboración de este proyecto que los miembros de mi familia. Quisiera dedicar esta investigación a mis padres, abuela, hermano y novia, cuyo amor y guía me acompañan en todo lo que persigo. Ellos son los pilares de mi vida y proporcionan una inspiración interminable.

AGRADECIMIENTOS

Este trabajo no hubiera sido posible sin el apoyo del grupo de investigadores del User Interaction Group de la Universidad de Costa Rica.

Quisiera expresar mi profunda gratitud al Dr. Gustavo López, por su guía, orientación, y críticas útiles durante la elaboración este trabajo de investigación. También me gustaría agradecer al Dr. Luis Quesada, al Dr. Luis Guerrero y al M. Sc Ignacio Díaz, por sus consejos y asistencia en las revisiones de mi proyecto.

También quisiera extender mi agradecimiento a las personas que colaboraron en las diversas evaluaciones del sistema, su aporte ha sido muy valioso para esta investigación.

Agradezco a todos los profesores de los cursos de maestría por ayudarme a extender mi conocimiento y enseñarme mucho sobre la investigación científica.

Agradezco a Fabiana Solano y su familia por siempre apoyarme y mantenerme motivado a continuar.

Finalmente, deseo agradecer a Janet Meza, Christian Arias y María Meza, por su apoyo y aliento a lo largo de mi carrera.

“Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Computación e Informática de la Universidad de Costa Rica, como requisito para optar por el grado y título de Maestría Profesional en Computación e Informática.”

Dr. Luis Guerrero Blanco
**Representante del Decano
Sistema de Estudios de Posgrado**

Dr. Gustavo López Herrera
Profesor guía

Dr. Luis Quesada Quirós
Asesor

M.Sc. Ignacio Díaz Oreiro
Asesor

Dra. Gabriela Marín Raventós
**Directora Programa de Posgrado en Computación e
Informática**

Andrés Meza Fernández
Candidato

CONTENIDO

DEDICATORIA	ii
AGRADECIMIENTOS	ii
RESUMEN.....	vi
LISTA DE FIGURAS.....	vii
LISTA DE TABLAS.....	vii
INTRODUCCIÓN.....	9
Pregunta de investigación.....	10
Objetivos	10
Estructura.....	10
TRABAJO RELACIONADO.....	11
PROBLEMA Y JUSTIFICACION	14
METODOLOGÍA.....	18
4.1 Diseño de arquitectura.....	18
4.2 Diseño de prototipo.....	18
4.3 Evaluación de prototipo	19
Architecture to Design Booking Appointment Applications for the Smart Personal Assistant Alexa.....	21
5.1 Elementos de la arquitectura	21
5.1.1 Estructura de una cita	21
5.1.2 Elementos de una cita	21
5.1.3 Tipos de configuración de citas	23
5.1.4 Combinaciones de configuración.....	24
5.1.5 Composición a través del diálogo	24

5.1.6	Almacenamiento de datos y panel de control empresarial	25
5.1.7	Abordar errores comunes	25
5.1.8	Múltiples tipos de negocios.....	26
5.1.9	Interacción multimodal.....	26
5.1.10	Confirmación y cambios.....	27
5.1.11	Casos de uso de las triadas.....	27
5.2	Diseño del prototipo.....	27
5.2.1	Configuración.....	28
5.2.2	Descripción de la arquitectura	28
5.2.3	Flujo de interacción.....	29
5.3	Evaluación	31
CONCLUSIONES Y TRABAJO FUTURO		33
ANEXOS.....		35
Anexo A – Tríadas		35
Anexo B – Funcionalidades deseadas en el prototipo.....		36
Anexo C – Materiales Complementarios		37
Anexo D – Architecture to Design Booking Appointment Applications for the Smart Personal Assistant Alexa		38
REFERENCIAS		53

RESUMEN

Los asistentes inteligentes son cada vez más interactivos y útiles para las tareas cotidianas. Amazon Echo es una herramienta que permite utilizar interacciones de voz avanzadas para ejecutar tareas complejas. Los desarrolladores no están aprovechando al máximo el potencial de Amazon Echo en el área de reserva de citas. Se propuso una arquitectura flexible para desarrollar aplicaciones de reserva de citas utilizando Amazon Echo. La arquitectura abstrae la complejidad de las interacciones de voz y sirve como guía para que los desarrolladores aún sin experiencia puedan construir interfaces de usuario por voz para reserva de citas. Se desarrolló un *skill* prototipo siguiendo los principios de la arquitectura y fue evaluada por un grupo de usuarios. El *skill* define con éxito cómo debe ser un *skill* de reserva de citas para Amazon Echo. El artículo producto de esta investigación fue aceptado en la decimotercera Conferencia *Ubiquitous Computing and Ambient Intelligence UCAmI 2019*, celebrada en Toledo, España, del 2 al 5 de diciembre de 2019 (Anexo D).

LISTA DE FIGURAS

Figura 1. Componentes principales del skill prototipo	30
Figura 2. Panel de administración de citas.....	31
Figura 3. Resultados del cuestionario de escala de usabilidad (SUS).....	32
Figura 4. Modelo de voz para la intención de agendado.....	37

LISTA DE TABLAS

Tabla 1. Casos de uso comunes para las tríadas.	27
Tabla 2. Tríadas validas	35

CAPÍTULO 1

INTRODUCCIÓN

El área de esta investigación se desenvuelve en el ámbito de las interfaces de usuario por voz (VUI) o asistentes personales inteligentes, conceptos que se utilizarán de forma intercambiable en este documento. Las interfaces de usuario por voz se están volviendo más interactivas y más fáciles de utilizar para las tareas diarias, especialmente las que son controladas utilizando lenguaje natural [1]. Para los efectos de este estudio se utiliza el asistente personal inteligente controlado por voz “Alexa Voice Service”, que responde al nombre “Alexa” [2].

Amazon Echo es una marca de altavoces inteligentes desarrollados por Amazon.com. Estos dispositivos se conectan mediante internet al servicio “Alexa Voice Service”. Las aplicaciones de Amazon Echo se llaman *skills* o habilidades [3]. Estas permiten a los usuarios interactuar con la voz, reproducir música, hacer listas de tareas, configurar alarmas, transmitir podcasts, reproducir audiolibros y proporcionar información sobre el clima, el tráfico, los deportes y otra información en tiempo real, como noticias [4]. Normalmente se hace referencia también a los dispositivos físicos utilizando el nombre “Alexa”. La creación de aplicaciones de interacción por voz para Amazon Echo o cualquier otro asistente inteligente requiere al menos cierto grado de conocimiento de diseño de VUI (Interfaces de usuario de voz), por ejemplo, estar familiarizado con conceptos tales como *utterances*, *intents* y *slots* [5] [6].

A pesar de que Amazon proporciona herramientas como el *Alexa Skills Kit* [3] que permite desarrollar experiencias avanzadas de voz, lograrlo generalmente no es un proceso sencillo para los nuevos desarrolladores. Esto lleva a la publicación de VUI mal diseñadas que eventualmente resultan en frustración para los usuarios finales [8].

Esta investigación se da en el contexto de la reserva de citas médicas para un consultorio dental utilizando aplicaciones de interacción por voz. Para los objetivos de esta investigación, una cita consiste en un acuerdo entre un usuario y una compañía, para

recibir un servicio por parte de esta en un momento y lugar. Opcionalmente se considera un asesor que es la persona que brinda el servicio durante la cita. A continuación, se presenta la pregunta de investigación propuesta.

Pregunta de investigación

La pregunta de investigación propuesta para el problema planteado es la siguiente: ¿Es posible desarrollar *skills* de reserva de citas para “*Alexa*” usables y adaptables a diversos tipos de negocio? La siguiente sección describe los objetivos planteados para la investigación.

Objetivos

El objetivo general de esta investigación es: Diseñar una arquitectura para implementación de interfaces de voz utilizando “*Alexa*” en el contexto de reserva de citas.

Los objetivos específicos de la presente investigación incluyen:

- 1. Modelar el problema de la reserva de citas de manera que sea automatizable mediante una interfaz por voz.**
- 2. Implementar un prototipo de skill de “*Alexa*” para agendar citas que siga los principios de diseño de interfaces por voz aplicables.**
- 3. Evaluar la usabilidad del prototipo en diferentes escenarios de reserva de citas.**

Estructura

Esta sección detalla la estructura del presente documento. El capítulo 1 realiza una introducción a VUI y los conceptos de Amazon Echo, adicionalmente se presentan los objetivos generales y específicos de la investigación. El capítulo 2 detalla el trabajo relacionado, se mencionan algunos sistemas y skills enfocados en la realización de reservaciones utilizando interfaces de usuario por voz. En el capítulo 3, se realiza una crítica sobre problemas encontrados en las soluciones existentes de reserva de citas y la importancia de solucionar dichos problemas. El capítulo 4 describe la metodología a seguir para completar cada objetivo planteado. El capítulo 5 contiene una descripción de la arquitectura desarrollada, detalles de la implementación del prototipo y los resultados

de evaluar dicho prototipo. Finalmente, el capítulo 6 menciona brevemente las conclusiones y trabajo futuro.

CAPÍTULO 2

TRABAJO RELACIONADO

Durante el proceso de esta investigación se realizó una revisión de literatura académica, sin embargo, la literatura no muestra mucha evidencia sobre estudios relacionados con la implementación de *skills* en repositorios académicos, esto potencialmente debido a que la mayoría de los desarrolladores están enfocados en la programación de *skills*, por lo tanto, esta sección se enfoca principalmente en trabajos relacionados que están en el ámbito industrial.

Un dispositivo experimental diseñado por IBM en 1961, el Shoebox fue un esfuerzo temprano para dominar el reconocimiento de voz. A principios de la década de 2000, los VUI eran sistemas interactivos de respuesta de voz (IVR) que entendían el habla humana por teléfono para realizar tareas. En la actualidad, hay muchas cosas que un asistente digital puede hacer bien por voz, pero todavía hay muchas cosas que simplemente no puede hacer. Las tecnologías como la interacción de voz no son nuevas, actualmente son interesantes porque finalmente están ampliamente disponibles comercialmente [7].

Los asistentes digitales de hoy de la mayoría de las empresas, como Apple (Siri), Amazon (Alexa), Google Now y Microsoft Cortana, han realizado mejoras serias a pasos agigantados para hacer las interacciones mucho más alegres y divertidas, pero aún no han utilizado por completo toda la funcionalidad que podría ayudarlos a convertirse en los asistentes inteligentes del futuro para casas y oficinas [7]. Hacer que los agentes de voz sean lo suficientemente amplios como para cubrir casi todos los contextos de uso de una historia de usuario puede ser un desafío [8]. En el mundo actual de asistentes personales, la mayoría de los usuarios a menudo tienen interacciones limitadas con ellos, como preguntar sobre el clima, hacer preguntas básicas de trivia, ayudar a establecer un

recordatorio o reproducir una canción. Esto generalmente se percibe como una falta en el uso de la capacidad del habla como un método de interacción [7]

Existen algunos desarrollos recientes en el área de reserva de citas con asistentes personales inteligentes. En la conferencia Google I / O de 2018, se presentó una demostración de un sistema llamado *Duplex AI*, que permite reservar citas y realizar otras tareas mediante llamadas telefónicas reales [9]. La tecnología aún está en desarrollo y no está claro si estará disponible para el público o los desarrolladores.

En el momento de elaboración de este documento, mayo de 2019 se exploró el sitio web Amazon.com utilizando “*Alexa Skills*” como categoría. Esta búsqueda retornó pocos resultados relacionados al utilizar términos como: “*reservation*”, “*booking*”, “*booking appointments*” o “*scheduling appointments*”. Se obtuvieron 24 skills como resultado, estas aplicaciones fueron instaladas y verificadas manualmente. Entre los resultados más relevantes se encontraron dos *skills* de compañías de viajes reconocidas: Expedia [10] y Kayak [11]. Ambas tienen como objetivo el alquiler de vehículos, y la reserva de hoteles y vuelos utilizando “*Alexa*”. Se encontraron también, dos *skills* pertenecientes a instituciones relacionadas al campo de la salud: el *skill Delta Dental from Kansas* [12] y el *skill Novant Health* [13]. Adicionalmente, se encontraron dos *skills* que ofrecen en su descripción la funcionalidad de reserva de citas. El primer *skill* se llama *Make Appointment* y pertenece a un centro de atención oftalmológica ubicado en Las Vegas llamado *Enhanced Eye Care* [14]. El *skill*, permite proporcionar aprobación para ser contactado por correo electrónico para coordinar una cita. El segundo *skill* se llama Pingup [15], pertenece a una compañía del mismo nombre que brinda servicios de reserva de citas a diferentes compañías. Pingup permite reservar una cita en cualquier compañía registrada en su plataforma. Las compañías se registran en Pingup proporcionando el tipo de negocio, el servicio y el nombre con el que desean operar. Finalmente, se encontró el *skill* de la plataforma Nimblr [16], la cual está enfocada exclusivamente en citas médicas y además, brinda el servicio completo de administración de citas, recordatorios y agenda. El resto de *skills* encontrados ofrecen únicamente la función de brindar información acerca de cómo realizar reservaciones. Durante la exploración de los *skills* mencionados se identificaron varios espacios de mejora para el uso y construcción de estas aplicaciones,

especialmente cuando se trata de facilidad, personalización, costo y completitud de la solicitud o reserva sin utilizar otro medio que no sea un asistente personal inteligente. La siguiente sección expone los problemas identificados en mayor detalle y el propósito con el que se abordan.

CAPÍTULO 3

PROBLEMA Y JUSTIFICACION

No todos los asistentes personales son capaces de proporcionar mecanismos para que sus usuarios puedan concretar negocios o transacciones de principio a fin, por ejemplo, “Alexa” es el único asistente personal capaz de realizar compras en línea [17].

La exploración de *skills* realizada en la sección anterior permitió determinar que no es posible concretar el agendado de citas o reservaciones de forma completa utilizando únicamente el asistente personal. Los *skills* existentes, permiten únicamente obtener información de disponibilidad o bien utilizar un método alternativo como correo electrónico para completar el proceso.

Esto indica que en general los skills no van más allá de listar información y no permiten una interacción completa con las compañías que los diseñan, con algunas excepciones puntuales que se mencionan más adelante.

Al diseñar aplicaciones para asistentes personales inteligentes relacionadas a citas o reservas pueden surgir historias de usuario con escenarios complejos, por ejemplo: programas con exceso de validaciones o flujos condicionales extensos. Cubrir estos contextos utilizando las funcionalidades disponibles en los servicios de voz puede ser un desafío para los programadores [8].

Amazon proporciona herramientas como el “Alexa” Skill Kit [3], el cual permite desarrollar experiencias de voz complejas, incluso con esta herramienta el proceso de desarrollo de skills no es sencillo para los desarrolladores. Esto ocasiona la publicación de aplicaciones con VUI mal diseñadas que eventualmente resultan en frustración para los usuarios finales [18]. Debido a lo anterior, hay una curva de aprendizaje para los desarrolladores cuando se trata de crear experiencias de voz complejas para asistentes inteligentes.

A continuación, se detallan los aspectos más relevantes obtenidos al probar y verificar los *skills* con potencial de reserva de citas que fueron identificados en la exploración conducida en la sección anterior. Se identificaron y probaron los siguientes skills:

- **Delta Dental from Kansas [12] y Novant Health [13]:**
 - Ninguno de estos dos skills ofrece en realidad la funcionalidad de concretar una reserva en su descripción
 - Únicamente brindan información de contacto para hacer la reserva mediante otro medio.
- **Make Appointment [14] y Pingup [15]:**
 - Ambos afirman ofrecer la funcionalidad de permitir la reserva de citas en su descripción, sin embargo, lo hacen parcialmente.
 - En Make Appointment [14] el proceso de reserva no se completa por medio de la interfaz por voz, sino que se hace vía correo electrónico.
 - Pingup si permite concretar la reserva de una cita, pero requiere que el negocio se encuentre previamente registrado en su sitio web, adicionalmente este skill combina todos los negocios bajo una misma aplicación, lo cual genera confusión al buscar negocios para reservar ya que el usuario debe conocerlos de antemano y saber que son parte de Pingup.
- **Kayak [11] y Expedia [10]:**
 - Ambos afirman ofrecer la funcionalidad de permitir la reserva de hoteles y autos en su descripción, sin embargo, lo hacen parcialmente.
 - A pesar de funcionar para reservar productos de ambas compañías, tienen el inconveniente de que no es posible para un desarrollador acceder a sus modelos de voz, esto debido a que su código no es público,
 - Permiten completar el proceso para un rango limitado de productos solamente, y en ocasiones se limitan a dirigir el usuario a un sitio web.
- **Nimblr [16]:**
 - Permite hacer reservas de citas utilizando la plataforma del mismo nombre como mediador.

- Al ser Nimblr una plataforma que ofrece el servicio completo de administración de citas, es necesario adquirir y utilizar la plataforma para todos los procesos involucrados, incluso aquellos fuera de Alexa. Esto significa que un comercio no podría integrar sus sistemas existentes con Nimblr.
- Nimblr está exclusivamente enfocado en citas médicas, y es poco adaptable cuando se trata de otro tipo de negocio.
- Nimblr no es gratuito, se debe adquirir un paquete para utilizar sus servicios.
- Nimblr al igual que pingup agrupa muchos comercios bajo un mismo skill, esto evita que el comercio tenga control sobre el skill.

No fue posible encontrar un skill totalmente funcional que permitiera reservar citas en un comercio particular sin la mediación de un tercero. Es importante resaltar que no todos los tipos de citas son iguales, existen variaciones en los requerimientos de estas que añaden aún más complejidad a este problema.

Se ha identificado un vacío en el mundo de los asistentes de voz inteligentes en lo que respecta a la reserva de citas. Los skills existentes no permiten completar una reserva de inicio a fin, no están disponibles para desarrollo aún o bien tienen un costo elevado para los negocios pequeños y medianos. Abordar estos temas es la principal motivación de este estudio.

Debido a lo anterior, resulta necesario buscar un mecanismo adaptable que pueda proveer los siguientes beneficios:

- Brindar a los usuarios un servicio funcional de agendado de citas, con una experiencia sencilla pero funcional.
- Abstractar a los desarrolladores de la complejidad que implica crear skills de reserva de citas utilizando interfaces de usuario por voz.
- Desarrollar skills con una interacción real entre los usuarios y los comercios de forma sencilla y sin mediadores.

- Permitir crear skills que puedan ser fácilmente integradas con sistemas existentes.
- Facilitar la integración de los negocios con “Alexa” de forma económica y en el menor tiempo posible.
- Convertir a los negocios en los dueños de sus skills potenciando una futura monetización y mercadeo de estos.

Este mecanismo tendrá como finalidad principal resolver los problemas antes mencionados de forma sencilla y accesible. En la siguiente sección se detalla la metodología utilizada en la investigación.

CAPÍTULO 4

METODOLOGÍA

Esta sección describe la metodología a ser implementada para alcanzar cada uno de los objetivos específicos planteados.

4.1 Diseño de arquitectura

Construir una arquitectura que permitirá diseñar *skills* de “*Alexa*” adaptables a diversos escenarios de reserva de citas. La arquitectura definirá las interacciones y comandos posibles entre el usuario y “*Alexa*”, adicionalmente, indicará la forma en la que la cita debe ser compuesta según su configuración. La configuración de las citas está definida en forma de tríadas (Ver Anexo A con la definición de lo que representa una tríada y el detalle de las potenciales tríadas que se pueden utilizar para configurar el *skill* según el tipo de cita). El desarrollo de la arquitectura se realizará tomando como base mejores prácticas para la construcción de interfaces por voz obtenidas de artículos académicos [18].

4.2 Diseño de prototipo

Diseñar un prototipo de *skill* de agendado de citas para “*Alexa*”, utilizando como guía la arquitectura previamente propuesta y como compañía un consultorio dental ficticio con varias sucursales. Para el diseño del prototipo serán considerados los principios de diseño de interfaces por voz [18], principalmente el manejo de errores, las recomendaciones para conversaciones por turno, el manejo de conversaciones para usuarios novatos o expertos y la simplicidad de los diálogos. Este prototipo será configurable para todas las combinaciones de elementos de cita soportadas por la arquitectura (Ver Anexo A con los elementos soportados), con el fin de poder evaluarlos. El prototipo definirá los flujos posibles que el usuario puede seguir al utilizarlo (Ver Anexo B con las funcionalidades deseadas en el prototipo).

Construir el prototipo mediante el uso de las tecnologías NodeJS [19], Typescript [20] y MySQL [21]. El prototipo será construido utilizando el *framework* “*Alexa Skills Kit SDK for Node.js*” proporcionado por Amazon.

Desplegar el prototipo en una función *Lambda* [22] de la plataforma AWS (Amazon Web Services) [23] para su utilización en dispositivos Echo reales.

Diseñar y desarrollar un sitio web tipo panel de control que permitirá visualizar y comprobar la correctitud de las citas reservadas mediante el prototipo de *skill* desarrollado, con el fin de simular la visualización final del comercio en su sistema *back office*. Este sitio web se construirá utilizando ASP.net [24] y MySQL [21]. Cabe destacar, que este sitio web y el prototipo compartirían la misma base de datos para almacenamiento de los datos de citas.

4.3 Evaluación de prototipo

Diseñar un instrumento de evaluación que permita medir la usabilidad del prototipo elaborado al ejecutar tareas de reserva de citas. El instrumento estará basado principalmente en el sistema SUS (*System Usability Scale*) adaptado para la evaluación de interfaces de usuario por voz [25]. Adicionalmente, se incluirán preguntas abiertas que permitirán obtener la opinión del usuario según los siguientes elementos: aspectos positivos y negativos del *skill*, caracterización, recomendación de uso, emoción, innovación, motivación, nivel de atractivo, y evaluación global. El instrumento también registrará el éxito o fracaso en la ejecución de las tareas realizadas, la configuración utilizada y las razones de fracaso de las tareas en caso de existir.

Evaluar el prototipo diseñado utilizando el instrumento anterior con un grupo de 24 personas mayores de 18 años de cualquier género, ubicados en Costa Rica, que tengan un nivel de inglés intermedio o avanzado. La evaluación consistirá en la ejecución de dos tareas de reserva de citas utilizando el *skill* de oficina dental.

La primera tarea consistirá en la reserva de una cita permitiendo que el usuario seleccione las preferencias de su elección, en este caso se solicitará al usuario aceptar la reserva de la cita en el último paso.

La segunda tarea consistirá en la reserva de una cita permitiendo que el usuario seleccione las preferencias de su elección con la variante de que en este caso se solicitará al usuario no aceptar la reserva en el último paso y realizar algunos cambios de su preferencia para finalmente aceptarla.

La evaluación se realizará en sesiones privadas individuales con un evaluador, utilizando un dispositivo Amazon Echo. Para cada persona se cambiará la configuración de tríadas en el *skill* de forma aleatoria, con el fin de contar con al menos una evaluación para todos los escenarios de reserva de citas soportados por la arquitectura. Una vez completadas por el usuario ambas tareas, el instrumento es proporcionado a cada usuario. La configuración utilizada y el éxito o fracaso de cada tarea también serán registrados en el instrumento en una sección exclusiva para el evaluador.

CAPÍTULO 5

Architecture to Design Booking Appointment Applications for the Smart Personal Assistant Alexa

Esta sección presenta en detalle la arquitectura diseñada, el prototipo construido a partir de la arquitectura y los resultados obtenidos de la evaluación de este.

5.1 Elementos de la arquitectura

En este segmento se describen los elementos principales de la arquitectura, y se proporciona un desglose de las estructuras de citas y diálogo. Se describen también algunas de las ventajas que ofrece la arquitectura en términos de usabilidad y personalización. Finalmente se discuten los casos de uso y las razones detrás de las decisiones de diseño.

5.1.1 Estructura de una cita

Una cita consiste en un acuerdo entre dos partes para un servicio en un momento y lugar determinados, extendiendo aún más este concepto, las citas pueden tener varios elementos adicionales que pueden ser opcionales según el tipo de negocio. Esta arquitectura se centra en las citas entre una persona y un negocio, no se admitirán otras variaciones en esta investigación. Los asistentes de voz como Amazon Echo no pueden recopilar grandes cantidades de información en una sola interacción de voz, esto significa que se necesitan varias interacciones de voz para intercambiar la suficiente información necesaria para construir un conjunto de información útil.

5.1.2 Elementos de una cita

Pensando en términos de programación orientada a objetos y con la idea anterior en mente, el objeto de cita se dividió en pequeñas partes, estas partes se llamarán elementos de cita de ahora en adelante. Cada elemento de cita debe recopilarse en una única interacción de voz, esto siguiendo el principio de simplicidad para los diseños de voz. Algunos elementos requerirán que los usuarios elijan entre largas listas de opciones,

solo algunos de los elementos de estas listas se leerán en voz alta por el dispositivo como ayuda para el usuario, ya que hacerlo para todos ellos podría dar como resultado una experiencia de voz deficiente. Las interacciones de voz deben ser breves y simples, la Sección 5.1.5 describe la solución propuesta para este problema en particular. Ciertos elementos se pueden configurar para necesidades particulares, esto se explorará en detalle en la sección Tipos de configuración de citas. La siguiente es una descripción de los elementos de cita válidos en la arquitectura propuesta.

- **Establecimiento / *Establishment* (E):** este elemento identifica la empresa o negocio donde tienen lugar las citas, este es un elemento inmutable y la raíz de la arquitectura, una aplicación debe tener un solo establecimiento. El nombre de la empresa generalmente se asigna a este elemento. Este elemento representa la aplicación en sí, lo que significa que cada establecimiento debe tener su propia aplicación. Manejar múltiples compañías en la misma aplicación puede ser problemático como se discutió anteriormente.
- **Sucursal / *Branch* (B):** este elemento identifica la oficina física o el lugar donde se realiza la cita. Es seleccionado por el usuario de una lista de sucursales válidas proporcionadas por la aplicación. La aplicación proporciona al usuario una lista de las sucursales más cercanas obtenidas en función de la ubicación configurada en el dispositivo del asistente inteligente. Este elemento no se puede configurar con la configuración de la arquitectura. El elemento sucursal siempre es necesario para crear un objeto de cita, incluso si el negocio tiene solo una ubicación física.
- **Servicio / *Service* (S):** este elemento describe los servicios disponibles que la empresa tiene para reservar. El usuario puede seleccionar un solo servicio de una lista de servicios válidos proporcionados por la aplicación, los servicios pueden ser globales para el *skill* o programarse en función de su disponibilidad en cada sucursal, el desarrollador está a cargo de tomar esta decisión. Este elemento no se puede configurar con la configuración de la arquitectura y siempre es necesario para crear un objeto de cita.
- **Asesor / *Assessor* (A):** algunas citas tienen una persona dedicada a proporcionar el servicio deseado, por ejemplo, el médico en el caso de una cita médica. El elemento asesor permite solicitar y almacenar información de esta persona. Este

elemento se puede configurar completamente utilizando la configuración de la arquitectura y es opcional cuando se trata de construir un objeto de cita.

- **Fecha / Date (D):** este elemento indica la fecha en que se realiza la cita. El usuario proporciona su preferencia de fecha para la cita, el valor ingresado se valida con las reglas comerciales establecidas por el programador, si la fecha ingresada es válida, la interacción finaliza. En una fecha no válida, la aplicación proporciona al cliente sugerencias de fechas disponibles o válidas. Los usuarios pueden indicar la fecha de varias maneras, por ejemplo, diciendo mañana, el próximo lunes o el 19 de noviembre. La lógica para soportar este tipo de entradas generalmente la proporciona el servicio de voz. Este elemento se puede configurar parcialmente utilizando la configuración de la arquitectura y siempre se requiere cuando se trata de construir un objeto de cita.
- **Hora / Time (T):** igual que la fecha, este elemento indica la hora en que se agenda la cita. El usuario proporciona su preferencia de tiempo para la cita y luego el valor ingresado se valida con las reglas de negocio, una vez más establecidas por el programador, si la entrada proporcionada es válida, la interacción finaliza, de lo contrario, la aplicación proporciona al cliente sugerencias de tiempo disponibles. Este elemento se puede configurar completamente usando la configuración de la arquitectura y es opcional cuando se trata de construir un objeto de cita.

5.1.3 Tipos de configuración de citas

Como se describió anteriormente, algunos elementos del objeto cita se pueden personalizar utilizando la configuración de la arquitectura. El propósito de esta configuración es proporcionar una estructura flexible que se pueda ajustar fácilmente a diferentes tipos de negocios. La siguiente es una descripción de las opciones de configuración disponibles en la arquitectura.

- **Dado / Given (G):** un elemento se configura como dado cuando se requiere su presencia en el objeto de cita, pero el usuario no puede seleccionarlo de una lista. Cuando se utiliza esta configuración, la aplicación se encarga de asignar un valor al elemento, ya sea siguiendo una regla comercial o de forma aleatoria, esto

dependerá de la configuración de la aplicación. Los elementos Asesor, Fecha y Hora son posibles candidatos para esta configuración.

- **Seleccionable / *Selectable* (S):** un elemento se configura como seleccionable cuando se requiere su presencia en el objeto de cita y el usuario puede seleccionarlo de una lista. Cuando se utiliza esta configuración, el usuario se encarga de asignar un valor al elemento. Los elementos Asesor, Fecha y Hora pueden configurarse para ser Seleccionables.
- **No requerido / *Not required* (N):** un elemento se configura como no requerido cuando no se requiere su presencia en el objeto de cita. Cuando se utiliza esta configuración, la aplicación omitirá todas las menciones a este elemento. Los elementos Asesor y Tiempo se pueden configurar como no requeridos.

5.1.4 Combinaciones de configuración

Para simplificar la comprensión de la arquitectura, cada elemento de cita y tipo de configuración se ha identificado con una letra según su nombre en el idioma inglés. Establecimiento (E), Sucursal (B) y Servicio (S) no son configurables, por lo que Asesor (A), Fecha (D) y Hora (T) son los elementos válidos para hacer combinaciones con los diferentes tipos de configuración: Dado (G), Seleccionable (S), y No requerido (N). Las combinaciones se realizan en forma de tríadas, por ejemplo, DG, TS, AN, para Fecha dada, Hora seleccionable y Asesor No requerido. Se diseñó una lista de 18 posibles tríadas que representan combinaciones válidas; consulte el Anexo A como referencia.

Una aplicación solo puede usar una tríada de configuración a la vez, el desarrollador a cargo de construir la aplicación debe identificar el tipo de cita que ofrece el negocio y seleccionar la tríada más apropiada para esa cita. Algunos ejemplos del uso de las tríadas se explican al final de esta sección.

5.1.5 Composición a través del diálogo

Las experiencias de voz pueden comenzar de diferentes maneras, el usuario debe tener la libertad de poder comenzar la interacción de solicitar una cita ya sea eligiendo un asesor en particular, un lugar en particular, un servicio específico, o simplemente una cita sin ningún otro parámetro. Tener el objeto de cita dividido en elementos más pequeños

tiene la ventaja de permitir obtener un conjunto de datos válido independientemente del orden en que se recopilen los elementos.

El contexto es importante cuando se trata de interacciones por voz, una mejor experiencia para el usuario significa proporcionarle opciones basadas en las elecciones que ya ha hecho. Si la aplicación ya tiene un valor para el elemento sucursal, debería ser posible enumerar solo los servicios disponibles en esa sucursal, también si la aplicación ya tiene un valor para el servicio, debería ser posible enumerar solo las sucursales que ofrecen ese servicio. La arquitectura propuesta admite esta característica al permitir a los desarrolladores retornar estos datos desde un repositorio de datos basado en los elementos conocidos.

5.1.6 Almacenamiento de datos y panel de control empresarial

Los datos de las citas deben almacenarse y recuperarse fácilmente, una recomendación inicial para el desarrollador es utilizar una base de datos relacional, pero también puede ser utilizar un API (*Application Programming Interface*) o cualquier tipo de sistema de almacenamiento, esta elección depende de las necesidades del negocio. Sin embargo, hay tres funcionalidades básicas que el sistema de almacenamiento debe proporcionar para tener una aplicación funcional: (1) Almacenamiento y recuperación de elementos y datos de citas, (2) Tiempo de respuesta rápido y (3) Acceso desde el tablero de negocios.

El establecimiento o la empresa deben tener un tablero para ver y administrar las citas próximas y existentes, se recomienda tener un sitio web para este propósito, una vez más, esta opción se delega al desarrollador ya que estas características también se pueden integrar el *skill* con una aplicación o sistema existente. Idealmente, este panel extraerá la información de la cita del sistema de almacenamiento.

5.1.7 Abordar errores comunes

Como se mencionó anteriormente, se encontraron problemas críticos en algunas de los *skills* existentes diseñadas para reservar citas, los esfuerzos de esta investigación se centraron en idear formas de lidiar con los más críticos en función de las pruebas

realizadas. La siguiente es una lista de los problemas que la arquitectura propuesta pretende cubrir.

5.1.8 Múltiples tipos de negocios

La arquitectura propuesta se enfoca en crear una aplicación individual para cada negocio, algunas de los *skills* existentes en el mercado de Alexa le preguntan al usuario el tipo de negocio que busca, por ejemplo, "¿le gustaría buscar negocios de belleza o de finanzas?", Esto podría confundir al usuario debido a que en una interfaz de voz es confuso saber que opción se ha seleccionado. Potencialmente, esto podría llevar a reservar una cita en un negocio no deseado, y previene la comercialización adecuada y la monetización del *skill* debido a la combinación de negocios involucrados.

5.1.9 Interacción multimodal

Como usuario, es fácil perderse en aplicaciones que dependen únicamente de la voz, especialmente cuando se trata de elegir un valor de una lista, algunos de los *skills* evaluados le solicitan al usuario proporcionar una opción, sin embargo, no le brindan las opciones disponibles para elegir. Los usuarios no saben que decir y se frustran por la falta de ayuda. La arquitectura propuesta hace uso de la aplicación complementaria para teléfono inteligente proporcionada por Amazon Echo llamada Alexa App, en esta aplicación se le presentan al usuario las opciones disponibles para cada interacción en forma de tarjetas de texto, esto sirve como una guía visual para el usuario y evita la necesidad de leer todas las opciones en voz alta, esta información también se muestra en dispositivos Amazon Echo que tienen una pantalla. En algunos casos, no es posible tener acceso a la aplicación complementaria, la arquitectura permite marcar algunos elementos de las listas a elegir como destacados. Estos elementos serán mencionados al usuario si no se proporciona una respuesta en un tiempo dado. Los elementos destacados son totalmente configurables en el mecanismo de almacenamiento, pueden ser aleatorios o estar basados en una regla de negocio, por ejemplo, destacar los servicios más comunes o las sucursales más populares. Se recomienda no tener más de cinco elementos marcados como destacados en las listas, con el fin de mantener cortas las interacciones

de voz. Tener estas ayudas en el *skill* evita que los usuarios se pierdan y les permite recibir ayuda de dos fuentes.

5.1.10 Confirmación y cambios

La arquitectura obliga en el *skill* una confirmación de la información que se recopila en cada interacción de voz, esto se hace para evitar la condición en la que un usuario responde muchas preguntas solo para darse cuenta de que el dispositivo ha recopilado los datos incorrectos desde el inicio, adicionalmente existe una confirmación final que resumen los detalles de la cita. Si el usuario indica que un elemento en el resumen es incorrecto, el dispositivo solicitará que se corrija el elemento específico de la cita y redirigirá al usuario a esa interacción.

5.1.11 Casos de uso de las tríadas

La Tabla 1 describe algunos casos de uso comunes para diferentes tipos de tríadas. Hay más casos de uso según el tipo de negocio, la arquitectura pretende ser lo suficientemente flexible como para cubrir la mayoría de ellos. Consulte el Anexo A para obtener una lista completa de las tríadas.

Tabla 1. Casos de uso comunes para las tríadas.

Tríada	Caso de uso
DS, TS, AS	Consultorio dental privado donde los usuarios pueden reservar una cita eligiendo la fecha, la hora y el dentista.
DS, TS, AG	Agencia de automóviles donde el usuario puede reservar una cita para una prueba de manejo, eligiendo la fecha y la hora, pero se le asigna un asociado a cargo de la demostración del vehículo.
DG, TG, AG	Hospital público donde los usuarios pueden reservar una cita para un servicio en particular, pero no pueden elegir la fecha, la hora o el médico.
DS, TG, AN	Oficina gubernamental donde los usuarios pueden reservar una cita para una fecha seleccionada, se les asigna la hora y no se requiere un asesor.

D: Fecha, T: Hora, A: Asesor, S: Seleccionable, G: Dado, N: No Requerido.

5.2 Diseño del prototipo

Basado en la arquitectura descrita en la sección anterior, se implementó un *skill* para Alexa llamado *Dental Office* con el propósito de evaluar su usabilidad. El desarrollo del *skill* ha alcanzado una fase de evaluación inicial; sin embargo, vale la pena mencionar que todavía es un trabajo en progreso, por esa razón no se implementaron todas las

características deseadas en la arquitectura, otros aspectos a evaluar posteriormente se mencionan en la sección de trabajo futuro.

5.2.1 Configuración

El *skill* implementado intenta proporcionar un servicio de reserva de citas para una clínica dental ficticia utilizando Alexa como canal de comunicación. Este caso fue seleccionado ya que puede ajustarse fácilmente para representar diferentes tipos de citas y negocios durante el proceso de evaluación del *skill*. La clínica dental tiene oficinas en 6 hospitales diferentes, estos hospitales representan las sucursales del negocio en la arquitectura, una lista de 31 servicios provistos por la clínica se incluyó en el *skill*, también se incluyó una lista de 6 médicos para servir como asesores. Las listas de sucursales, servicios y asesores se incluyeron en el modelo de voz y también en la base de datos, también se incluyeron sinónimos de todos los elementos en el modelo de voz para mejorar la experiencia del usuario (ver Anexo C, modelo de voz). Las sucursales se configuraron abiertas para recibir citas de lunes a viernes de 8 a.m. a 6 p.m. Cada cita se estableció para tener un intervalo de tiempo fijo de 30 min.

5.2.2 Descripción de la arquitectura

El *skill* de la clínica dental se desarrolló utilizando el *Alexa Skills Kit SDK for Node.js* versión 2 [26] siendo este el último SDK disponible para Node.js al momento de escribir este documento. Node.js versión 6.11.0 [19] fue seleccionado como plataforma de desarrollo principalmente debido a su modelo de entrada/salida sin bloqueo controlado por eventos, muy adecuado para aplicaciones de Alexa.

Una función lambda de AWS es un plataforma proporcionada por Amazon que provee la ejecución de código como servicio [22], para este *skill* particular se ha creado una función lambda de AWS cuya función es ejecutar código de back-end en respuesta a eventos originados por el servicio Alexa. El código fue creado usando el Typescript como lenguaje de programación [20] y desplegado en la función lambda (ver Anexo C, plantilla de código). En esta implementación particular, MySQL [21] se está utilizando como mecanismo de almacenamiento, esta base de datos está alojada en un servidor

independiente. Como se mencionó antes, esta implementación se puede reemplazar fácilmente con un API u otro mecanismo de almacenamiento.

5.2.3 Flujo de interacción

El usuario inicia el *skill* diciendo una de las frases de invocación admitidas por Alexa, seguido del nombre del *skill* y una intención opcional [27], por ejemplo, Alexa solicita a la clínica dental que reserve una cita / *Alexa ask Dental Office to book an appointment*.

Las expresiones del modelo de voz se asignan a métodos de intención. Cada vez que el usuario habla, la entrada del lenguaje natural se traduce a texto a través del *Alexa Service*, si el texto resultante coincide con una permutación de frases válida para una intención, el *Alexa Service* ejecuta el método mapeado para esa intención en la función lambda utilizando una solicitud de intención. En el código de función lambda, los métodos de intención se definen como funciones de middleware en los controladores de la aplicación, también hay servicios encargados de proporcionar a los controladores datos externos y funciones auxiliares.

Cada vez que se ejecuta un método de intención, el control se devuelve al *Alexa Service* y, potencialmente, al usuario. La función lambda envía comandos de voz para información y comportamiento al *Alexa Service* utilizando el formato SSML [28]. La función lambda también se comunica con el servidor de bases de datos para almacenar los datos de citas y recuperar las listas de elementos de citas. Esta base de datos también sirve como almacenamiento para el sitio web del panel de citas. La Figura 1 describe los componentes principales del *skill* prototipo y el flujo de una interacción por voz (TTS: texto a voz, STT: voz a texto). La Figura 2 muestra el sitio web utilizado como panel de administración de citas y configuración de tríadas.

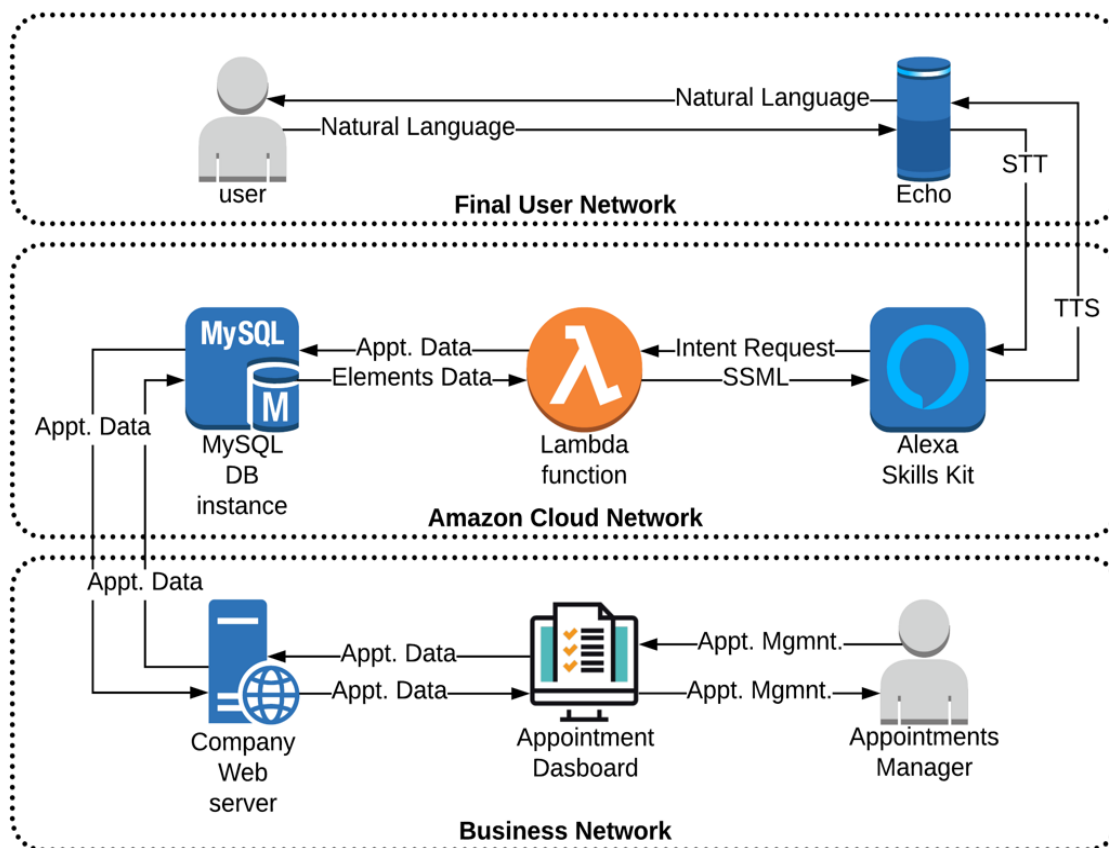


Figura 1. Componentes principales del skill prototipo

Dental Office

Model Configuration

Select a new config:
 
 Current Config:
 Date: G Time: G Assessor: G

Appointments list

Id	Branch	Service	Assessor	Date	StartTime	EndTime
147	Catolica Hospital	Crowns	Dr. Oz	2018-11-26	12:00	12:00
146	Zima Hospital	GENERAL	N/A	2018-12-11	N/A	N/A
145	Zima Hospital	Cleaning	N/A	2018-11-28	N/A	N/A
144	Biblica Hospital	Cleaning	N/A	2018-11-20	10:00	10:00
143	Zima Hospital	Dental Implants	N/A	2018-11-20	09:00	09:00
142	Biblica Hospital	Cleaning	Dr. Miranda	2018-11-20	08:00	08:00
141	Santa Rita Clinic	Oral Cancer Screening	Dr. Meza	2018-12-04	N/A	N/A
140	Catolica Hospital	Crowns	Dr. Oz	2018-11-27	N/A	N/A
139	Catolica Hospital	Root Canal Therapy	Dr. Oz	2018-11-07	08:00	08:00

Figura 2. Panel de administración de citas

En este punto del proceso de desarrollo, no todas las características del modelo se incluyeron en el prototipo. El Anexo B describe las características incluidas y excluidas de el *skill* prototipo.

5.3 Evaluación

En esta sección se presentan los resultados de la evaluación del *skill* prototipo, se incluyen datos de los evaluadores, porcentaje de éxito de las tareas y la evaluación de usabilidad (SUS).

La evaluación fue conducida en 24 participantes cuya edad se ubicó entre 20 y 40 años, siendo el 37.5% de ellos mujeres y el 62.5% hombres.

La mayoría de los usuarios pudieron completar ambas tareas con éxito, el 83.3% de los usuarios pudieron completar la primera tarea y el 75% de los usuarios pudieron completar la segunda tarea. El 75% y el 33% de los fallos para la primera y segunda tarea respectivamente, fueron causadas por una pronunciación incorrecta de los comandos de voz por parte de los usuarios. El 25% y el 16,7% de los fallos en la primera y segunda

tarea respectivamente, fueron causadas por usuarios que le proporcionaron comandos de ruta inválidos al *skill*, un comando de ruta inválida termina la ejecución de un *skill* en Alexa.

Se pidió a los usuarios que caracterizaran el *skill* usando tres palabras. Las palabras más relevantes encontradas fueron fácil, útil y rápida.

Las respuestas al cuestionario de escala de usabilidad (SUS) del sistema fueron en general positivas con un puntaje promedio de 80. Los usuarios están muy de acuerdo con la rapidez con que se puede aprender el uso del *skill*, la mayoría de los usuarios está de acuerdo con la integración adecuada de las funciones del *skill*. La Figura 3 presenta los principales resultados del cuestionario de escala de usabilidad (SUS).

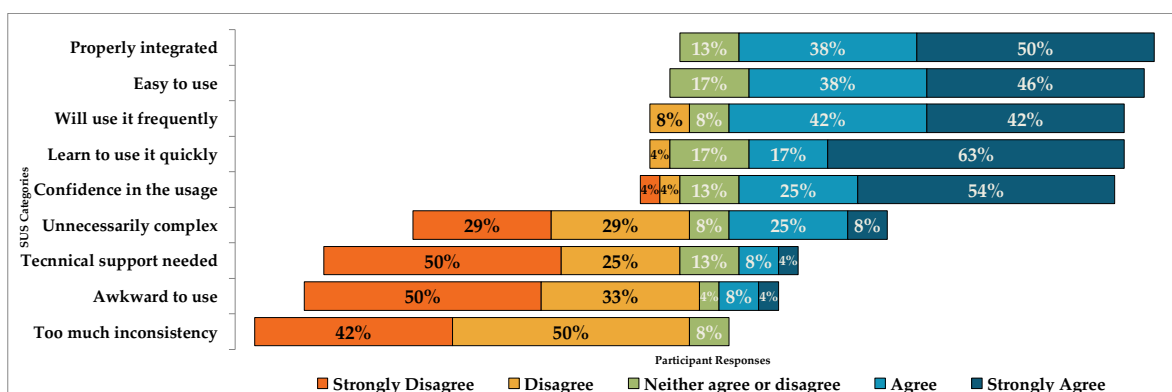


Figura 3. Resultados del cuestionario de escala de usabilidad (SUS)

Se pidió a los usuarios que evaluaran el *skill* en términos generales usando una escala del 1 al 10, siendo 10 la mejor puntuación posible. 83.3% de los usuarios clasificaron el *skill* entre 8 y 10. Finalmente, se consultó a los usuarios si recomendarían el *skill* a otros, el 91.7% de los usuarios respondieron que sí a esta pregunta. En la siguiente sección se discuten las conclusiones y el trabajo futuro.

CAPÍTULO 6

CONCLUSIONES Y TRABAJO FUTURO

Pocos *skills* de Alexa proporcionan la funcionalidad de reserva de citas, la mayoría dependen de un tercero para lograrlo, crear dichos *skills* desde cero plantea un desafío para los nuevos desarrolladores. Se diseñó una arquitectura que sirve como base para crear *skills* reserva de citas en varios escenarios comerciales. Se creó un *skill* prototipo siguiendo los principios de la arquitectura y finalmente el *skill* fue evaluado por un grupo de personas utilizando un instrumento basado en la escala de usabilidad del sistema (SUS).

Los resultados de caracterización y percepción encontrados nos permitieron concluir que el sistema fue recibido positivamente por los usuarios y la mayoría de los usuarios lo encontraron fácil, útil y rápido. Los usuarios creen que la aplicación puede ser fácilmente parte de su vida diaria y podría ayudarlos a ahorrar tiempo.

Los resultados del SUS revelaron un alto puntaje en términos de funciones integradas adecuadamente, siendo el que más le gustó el hecho de que las citas se reservan correctamente y no se requiere ninguna acción adicional por parte del usuario. La confianza del usuario y la facilidad de uso también obtuvieron un puntaje alto en la evaluación del SUS, asociamos este resultado a la guía constante incluida en toda la aplicación, los usuarios rara vez no saben qué decir. A pesar de que la mayoría de los usuarios encuentran que el *skill* es fácil de usar, el 33.33% está de acuerdo en que la aplicación es innecesariamente compleja, principalmente debido al exceso de confirmaciones, las cuales son necesarias.

Los resultados de la primera tarea son un claro indicador de lo fácil que fue para los usuarios utilizar el *skill* por primera vez sin experiencia previa.

Los resultados de la segunda tarea indican que la mayoría de los usuarios pudieron resolver el obstáculo de hacer cambios en los datos de la cita. El hecho de que la

segunda tarea fue también la segunda vez que los usuarios escuchan las instrucciones, sirve como un ejemplo de la intuición del *skill*.

Hay mucho trabajo por delante en el área de asistentes inteligentes, el trabajo futuro para este proyecto implica una evaluación de la experiencia empresarial durante el proceso de reserva de citas utilizando el *skill* prototipo y mejorando los diálogos y respuestas de VUI. Una evaluación adicional de la arquitectura con un grupo de desarrolladores podría proporcionar una idea de cuánto tiempo se puede ahorrar al usarla. Además, en este punto, la arquitectura es lo suficientemente genérica como para implementarse en otras VUI que siguen el concepto de expresiones, intentos y sesiones. La arquitectura podría adaptarse para ser utilizada en otros asistentes inteligentes y también modificada para ser utilizada en diferentes idiomas para Amazon Echo. Finalmente, el modelo podría extenderse para que funcione para otros modelos de negocios.

ANEXOS

Anexo A – Tríadas

Una cita está compuesta por los elementos: Ubicación, Servicio, Asesor, Fecha y Hora, siendo los primeros dos siempre requeridos. Las tríadas representan las posibles combinaciones de los elementos variables: Fecha / *Date* (D), Hora / *Time* (T) y Asesor / *Assessor* (A) para las diferentes necesidades de reserva de citas. Seleccionable / *Selectable* (S) se utiliza cuando el usuario puede elegir el elemento, Dado / *Given* (G) cuando al usuario se le asigna el elemento, y No Requerido / *Not Required* (N) cuando el elemento no es necesario para la cita.

Tabla 2. Tríadas validas

Tríadas
DS, TS, AS
DS, TS, AG
DS, TS, AN
DS, TG, AS
DS, TG, AG
DS, TG, AN
DS, TN, AS
DS, TN, AG
DS, TN, AN
DG, TS, AS
DG, TS, AG
DG, TS, AN
DG, TG, AS
DG, TG, AG
DG, TG, AN
DG, TN, AS
DG, TN, AG
DG, TN, AN

D: Date, T: Time, A: Assessor, S: Selectable, G: Given, N: Not Required

Anexo B – Funcionalidades deseadas en el prototipo

- Reserva de citas seleccionando ubicación, servicio, asesor, fecha y hora
- *Skill* configurable para soportar todas las tríadas (combinaciones de elementos) descritas en la arquitectura.
- Confirmación inmediata de los elementos en cada interacción de voz.
- Funcionalidad para realizar cambios en elementos individuales antes de confirmar el resumen de la cita.
- Recomendación de elementos destacados en los silencios del usuario.
- Visualización de las opciones disponibles en la aplicación móvil “*Alexa*”.
- Capacidad de utilizar datos de elementos extraídos de una base de datos MySQL o mecanismo de almacenamiento alternativo.
- Sinónimos en los modelos de voz para los nombres y ubicaciones del asesor.
- Soporte para diferentes tipos de entrada de fecha y hora (sin admitir formatos incompletos. Por ejemplo, la próxima semana, tarde)
- Panel de control del sitio web para gestionar las citas realizadas.
- Solicitar la lista de opciones disponibles en un momento dado de la interacción.
- Notificación vía email con resumen de cita. *
- Solicitar la fecha disponible más cercana para una cita. *
- Solicitar el tiempo disponible más cercano en un día determinado. *
- Solicitar la oficina más popular para un servicio dado.
- Solicitar la próxima fecha disponible para un servicio dado.
- Cancelación de cita utilizando voz. *
- Indicación de conflictos de citas en términos de fecha y hora.
- Recomendaciones válidas en términos de fecha y hora.
- Solicitar ayuda según el contexto en un momento dado de la interacción.
- Uso de la ubicación del dispositivo para recomendar sucursales. *

* No esenciales

Anexo C – Materiales Complementarios

- Modelo de voz:
 - El modelo de voz representa el conjunto de posibles elementos que Alexa utiliza para realizar permutaciones y potencialmente generar un modelo de interacción por con miles de interpretaciones de una misma intención.
 - Configuración de modelo de voz en bruto:
 - <https://s3.amazonaws.com/cdn.andresmeza.com/Alexa/AppointmentsModel/en-US.json>
 - La figura 2 presenta un extracto visual del modelo de voz para la intención de agendado, cada color representa un elemento variable en una oración para ejecutar la intención. La intención de agendado tiene 54 permutaciones de 13 slots, cada slot puede contener n valores. El *skill* contiene 12 intentos con diferentes cantidades permutaciones y slots. El objetivo de estas combinaciones es comprender la mayor cantidad de variantes del lenguaje natural.

Figura 4. Modelo de voz para la intención de agendado

- Plantilla de código:
 - Código utilizado como plantilla para el *skill Dental Office*:
 - <https://github.com/ameza/alex-appointments-blueprint>

Anexo D – Architecture to Design Booking Appointment Applications for the Smart Personal Assistant Alexa

Este anexo incluye el texto completo del artículo publicado en la conferencia *Ubiquitous Computing and Ambient Intelligence UCAmI 2019*.



Proceedings

Architecture to Design Booking Appointment Applications for the Smart Personal Assistant Alexa †

Andrés Meza *, Gustavo López, Luis Quesada and Luis A. Guerrero

Programa de Posgrado en Computación e Informática, Ciudad Universitaria Rodrigo Facio, University of Costa Rica, 11501-2060 San José, Costa Rica; gustavo.lopez_h@ucr.ac.cr (G.L.); luis.quesada@ecci.ucr.ac.cr (L.Q.); luis.guerrero@ecci.ucr.ac.cr (L.A.G.)

* Correspondence: andres.meza@ucr.ac.cr (A.M.); Tel.: +506-89710216

† Presented at the 13th International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2019, Toledo, Spain, 2–5 December 2019.

Published: 20 November 2019

Abstract: The intelligent smart assistants are becoming more interactive and helpful for everyday tasks. The Amazon Echo has potential for advanced voice interactions and as a tool for conducting complex tasks. The potential of the Amazon Echo in the area of booking appointments is not being fully exploited by developers. A flexible architecture for developing appointment booking applications for the Amazon Echo was proposed. The architecture serves as guide for developers without experience working with Voice User Interfaces and saves development time by abstracting the complexity of voice interactions. A prototype skill was developed following the architecture principles and evaluated by a group of users. The skill successfully defines how an appointment booking skill should be.

Keywords: Alexa; appointments; Amazon Echo; skill; AWS; VUI; architecture

1. Introduction

Intelligent personal assistants are devices that are becoming more interactive and easier to use for everyday tasks, especially the ones controlled by natural language [1]. Most of the applications available in these devices do not provide a mechanism for businesses to engage with their users. There is also a learning curve for developers when it comes to create complex voice experiences. Addressing these issues is the main motivation for this work.

Amazon Echo is a brand of smart speakers developed by Amazon.com. The devices connect via internet to the voice-controlled intelligent personal assistant service called “Alexa Voice Service”, which responds to the name “Alexa” [2]. Amazon Echo applications are called skills [3], the users send commands to the skills using voice only, the skills allow to play music, make to-do lists, set alarms, broadcast podcasts, play audiobooks and get information on weather, traffic, sports, news and other information in real time [4]. The name “Alexa” is often used to identify the physical Amazon Echo devices. Creating voice interaction applications for the Amazon Echo or any other smart assistant requires at least some degree of VUI (Voice User Interfaces) design knowledge, e.g., being familiar with concepts such as utterances, intents, and slots [5,6]. Making the voice agents vast enough to cover almost every usage context of a user story can also be challenging [7].

Despite the fact that amazon provides tools like the Alexa Skill Kit [3] to develop vast experiences, achieving this is usually not a straightforward process for new developers. This leads to the publication of poorly designed VUIs that eventually result in frustration for the final users [8].

The context of this research is limited to appointments in the dentistry field, where an appointment consists of an agreement between a user and a company, so the user can receive a service at a given time and place, optionally an assessor who is the person who provides the service is also

considered. Most of the appointment skills available for Alexa at the time of creation of this study cannot be used as channel of interaction between customers and the companies that provide such skills, sometimes they fail to perform the reservation, offer confusing user experiences and have limitations for developers. These drawbacks will be discussed in detail in the background section. It is important to highlight that not all types of appointments are the same, there are variations in their requirements that add even more complexity to this problem. For the reasons exposed, it is necessary to look for an adaptable mechanism that allows developers to be abstracted from the complexity involved in creating booking skills using voice user interfaces, and at the same time allow real interactions between users and businesses.

In this paper, a voice interaction architecture for the Amazon Echo is presented, focused on abstracting Amazon Alexa developers from the complexity of building a VUI for appointment booking in most of its variations. The purpose of the proposed architecture is to serve as template and guide for developers. This approach allows developers to simply fill the gaps in the provided architecture and configure it for their specific needs, Alexa blueprints works in a similar fashion for simpler experiences [9]. This work provides the interaction architecture documentation, the backend code for the prototype skill, and also the code for a business facing back office website that allows one to verify the appointments and configure the prototype. The elements described above are in charge of providing the connection between the final user and the business where he wants to book. Following the guidelines in the proposed architecture, a prototype skill focused on booking appointments for a dental clinic was designed, the usability of the prototype was evaluated by a group of users.

This document is structured as follows: Section 2 describes the background, Section 3 presents the proposed architecture, Section 4 shows the architecture implementation and configurations, Section 5 contains a brief description of the conducted evaluation, finally Section 6 covers the conclusion and future work.

2. Background

During the process of this research, a review of academic literature was conducted, however, the literature does not show much evidence about studies related to the implementation of skills in academic repositories, this is potentially due to the fact that most of the developers are focused on construction, therefore, this section focuses mainly on related works that belong to the industrial field. There have been some recent developments in the area of booking appointments using digital assistants, in the Google I/O conference of 2018 a demo of a system called Duplex AI was presented, it allows one to book appointments and accomplish other tasks by making real phone calls [10]. The technology is still in development and it is not clear if it will be available to the public or developers.

At the time of writing this document, May 2018, a search was conducted on the amazon.com website using "Alexa Skills" as a category, this search returned 24 skills using terms such as: "booking", "booking appointments" or "scheduling appointments". The 24 skills found were installed tested and verified manually. Most of them were focused on booking tables in restaurants or booking other services like taxis and hotel rooms. Two skills from recognized travel companies Expedia [11] and Kayak [12] were found. Both aim to rent vehicles, and reserve hotels and flights using Alexa. Skills associated with companies that have booking potential tend to provide contact information or interesting facts rather than providing the booking functionality themselves. Some examples are: the Delta Dental of Kansas [13] skill and the Novant Health [14] skill, both belonging to healthcare institutions.

None of the skills in the category of healthcare offer the appointment booking functionality in their description. Three skills were found which advertise in their description the functionality of allowing to book appointments. The first skill it is called "Make Appointment", it belongs to an ophthalmological center located in Las Vegas called Enhanced Eye Care [15], the skill allows to sign up for being contacted via email to coordinate an appointment. The second skill is called Pingup [16], it belongs to a company of the same name that provides booking services to other companies, Pingup allows to book an appointment on any company registered in their platform. This is achieved by

providing the type of business, the service and the name of the business, however, this data must be known prior to its use. The last skill belongs to the Nimblr platform [17] and goes by the same name, it allows to book medical appointments, and is not very adaptable when it comes to another type of appointment or reservation, also it does not allow integration with existing systems. The Nimblr platform has associated costs for its usage.

The booking appointment skills found in the market were enabled and partially tested, the main goals were: testing the booking appointment functionality and explore the skill structure as an application. This quick assessment of the skills revealed the following issues:

- Multiple types of businesses: branding and monetization of the individual companies in the amazon platform is affected as all the businesses are behind a single skill name and brand.
- Lack of help: the user is required to know the type of businesses before using the skill, they do not provide any kind of help mechanisms when it comes to making choices.
- Incomplete experience: the appointment functionality is delegated to email and turned into a manual process, leaving the voice assistant behind.
- Associated fees for businesses: businesses are charged monthly fees for the booking platform service.
- Lack of control: the business is not in control of the Alexa Skill, therefore, there is a limit on the changes that can be made.
- Security risk: having multiple businesses under the same skill means that companies data is combined to certain degree.
- Proprietary voice models: some skills despite working properly belong to businesses and for that reason they do not disclose their voice models to the public.

3. Appointments Architecture

In this section, the main elements of the architecture are described, and a breakdown of the appointment and dialog structures is provided, some of the advantages offered by the architecture in terms of usability and customization are described, finally, the use cases and the reasons behind some of the design choices are discussed.

3.1. Appointment Structure

An appointment is defined as an agreement between two parts at a given time and place, extending this concept further, appointments can have several additional elements that can be optional depending on the business type. This architecture is focused on appointments between a single person and a single business only, no other variations on this area will be supported by this study. Voice assistants like the Amazon Echo are unable to collect big chunks of information in a single voice interaction, this means that several voice interactions are needed in order to exchange enough information to build a useful information set.

3.1.1. Appointment Elements

Thinking in terms of Object-Oriented Programming and with the previous idea in mind, the appointment object was divided in small pieces, these pieces are going to be called appointment elements from now on. Each appointment element must be collected on a single voice interaction following the principle of simplicity for voice designs. Some elements will require the users to choose from long lists of items, only a few of the items in these lists will be read aloud to the user, doing it for all of them might result in a poor voice experience. Voice interactions must be kept short and simple, Section 3.2 describes the proposed solution to this particular issue. Certain elements can be configured for particular needs, this will be explored in detail in the Appointment Configuration Types section. The following is a description of the valid appointment elements in the proposed architecture.

- Establishment (E): This element identifies the company or business where the appointments take place, this is an immutable element and the root of the architecture, an application must have just one establishment. The company name is usually assigned to this element. This element represents the application itself, meaning that each establishment must have its own application. Handling multiple companies in the same application can be problematic as discussed earlier, the proposed solution to this problem will be discussed further in this section.
- Branch (B): This element identifies the physical office or location where the appointment takes place. It is selected by the user from a list of valid branches provided by the application. The application provides the user with a list of the nearest branches obtained based on the location configured in the smart assistant device. This element cannot be configured using the architecture settings. The Branch element is always required to build an appointment object, even if the business has only one physical location.
- Service (S): This element describes the available services the business has for booking. A single service can be selected by the user from a list of valid services provided by the application, services can be global for the skill or programmed based on their availability on each branch, the developer is in charge of making this decision. This element cannot be configured using the architecture settings and is always required to build an appointment object.
- Assessor (A): Some appointments have a person dedicated to providing the desired service, e.g., the doctor in the case of a medical appointment, the assessor element allows to request and store this person information. This element can be fully configured using the architecture settings and its optional when it comes to build an appointment object.
- Date (D): This element indicates the date when the appointment takes place. The user provides his preference of date for the appointment, the input gets validated against the business rules set by the programmer, if the input is valid the interaction finishes. On invalid input the application provides the customer with available date suggestions. Users are allowed to indicate the date in several ways, e.g., by saying tomorrow, next Monday or November 19th. The logic to support this kind of inputs is usually provided by the voice service. This element can be partially configured using the architecture settings and is always required when it comes to build an appointment object.
- Time (T): Same as the date, this element indicates the time when the appointment takes place. The user provides his preference of time for the appointment and then the input gets validated against the business rules, once again set by the programmer, if the provided input is valid the interaction finishes, otherwise, the application provides the customer with available time suggestions. This element can be fully configured using the architecture settings and is optional when it comes to build an appointment object.

3.1.2. Appointment Configuration Types

As described above, some elements of the appointment object can be configured using the architecture settings. The purpose of this configuration is to provide a flexible structure that can be easily adjusted to different types of businesses. The following is a description of the architecture settings available.

- Given (G): An element is configured as given when its presence is required in the appointment object, but the user is not allowed to select it from a list. When this configuration is used the application is in charge of assigning a value to the element, either by following a business rule or randomly, this will depend on the application setup. The Assessor, Date and Time elements can be configured to be Given.
- Selectable (S): An element is configured as selectable when its presence is required in the appointment object, and the user is allowed to select it from a list. When this configuration is used the user is in charge of assigning a value to the element. The Assessor, Date and Time elements can be configured to be Selectable.

- Not Required (N): An element is configured as not required when its presence is not required in the appointment object. When this configuration is used the application will skip all the mentions to this element. The Assessor and Time elements can be configured to be Not required.

3.1.3. Configuration Combinations

In order to simplify the understanding of the architecture each appointment element and configuration type has been identified with a letter. Establishment (E), Branch (B) and Service (S) are not configurable, this leaves Assessor (A), Date (D) and Time (T) to make valid combinations with the different configuration types: Given (G), Selectable (S), and Not Required (N). The combinations are made in form of triads e.g., DG, TS, AN, for Date Given, Time Selectable and Assessor Not required. A list of 18 possible triads that stand for valid combinations was devised, see Appendix A for reference.

An application can only use one configuration triad at the time, the developer in charge of building the application must identify the type of appointment the business offers and select the most appropriate triad for it. Some examples of the usage of triads are explained at the end of this section.

3.2. Composition through Dialog

Voice experiences can start in different ways, the user should have the freedom to start the interaction by asking for booking an appointment with a particular assessor, in a particular location, with a specific service, or just an appointment without any other parameter. Having the appointment object divided into smaller elements has the benefit of allowing to obtain a valid dataset regardless of the order in which the elements were collected.

Context is important when it comes to voice interactions, a better experience for the user means providing the user with choices based on the choices he already made. If the application has already a value for the branch element, it should be possible to list only services available in that branch, also if the application already has a value for the service, it should be possible to list only branches that offer that service. The proposed architecture supports this feature by allowing developers to return this data from a data repository based on the known elements.

3.3. Data Storage and Business Dashboard

The appointments data should be easily stored and retrieved, a relational database is the initial recommendation, but this can be also replaced by an API or any kind of storage system, this choice depends on the business needs. However, there are three basic functionalities that the storage system must provide in order to have a functional application: (1) Storage and retrieval of elements and appointments data, (2) Fast response time, and (3) Access from the business dashboard.

The establishment or business must have a dashboard to see and manage the upcoming and existing appointments, having a website for this purpose is recommended, once again this choice is delegated to the developer as these features can also be integrated in an existing application or system. Ideally, this dashboard will extract the appointment information from the storage system.

3.4. Addressing Common Mistakes

As it was mentioned in the background, critical issues in some of the existing skills designed for booking appointments were found, the efforts were focused on devising ways to deal with the most critical ones based on the conducted testing. The following is a list of the issues that the proposed architecture aims to cover.

3.4.1. Multiple Types of Businesses

The proposed architecture focuses on creating an individual application for each single business, some of the existing Alexa skills ask the user for the desired type of business, e.g., "would you like beauty or finance?", this might confuse the user because it makes easier to lose track of the business where the user is trying book. This could potentially lead to booking an appointment in an undesired

business, and also prevents the proper marketing and monetization of the skill due the mix of businesses involved.

3.4.2. Multimodal Interaction

As a user it is easy to get lost in applications that rely solely on voice, especially when it comes to make a choice, some of the skills tested ask the user to provide an option but fail on providing the available choices to pick from, users do not know what to say and get frustrated by the lack of help. The proposed architecture makes use of the companion app provided by the Amazon Echo called Alexa App, in this smart phone application the user is presented with the available options for every choice available in form of text cards, this serves as a visual guide to the user and avoids the need of reading all the options aloud, this information is also displayed in Amazon Echo devices that have a screen. In some cases, it is not possible to have access to the companion app, for this reason, the option to have some items in the lists identified as starred was included in the proposed architecture. These items will be voiced to the user if no answer is provided after a short time. The starred items are fully configurable in the storage mechanism, they can be random or based on a business rule. e.g., most common services or most popular branches. It is recommended to have no more than five items as starred in the lists to keep the voice interactions short. Having these helpers in place prevents the users from getting lost and allows them to receive output from two sources.

3.4.3. Confirmation and Changes

The user is requested to provide a prompt confirmation for each piece of information that gets collected on each voice interaction, this is done in order to prevent the condition where a user answers a lot of questions just to realize the device has collected the wrong data, there is also a final confirmation for the appointment summary. If the user indicates an item in the summary is incorrect, the device will ask for the specific element on the appointment to be corrected and redirect the user to that interaction.

3.5. Customization Requirements

This model structure aims to be simple and easily customizable for the developers in charge of the application implementation, aside from the basic skill requirements, the following items are the minimum conditions needed to start the development process: (1) A valid configuration triad, (2) An establishment or company name, (3) A list of branches, services, and assessors, each one with their respective starred items, (4) A storage mechanism, and (5) A business dashboard. Branches, Services, and Assessors can go by different names, it is recommended to include synonyms in the voice model, this provides more flexibility in terms of user input.

3.6. Triads Use Cases

Table 1 describes some common use cases for different types of triads. There are more use cases depending on the business type, the architecture aims to be flexible enough to cover most of them. See Appendix A for a full list of triads.

Table 1. Common Use Cases for Triads.

Triad	Use Case
DS, TS, AS	Private dental office where users can book an appointment choosing the date, the time and the dentist.
DS, TS, AG	Car dealership where users can book an appointment for a test drive, choosing the date and time but are given an associative in charge of demoing the vehicle.
DG, TG, AG	Public hospital where users can book an appointment for a particular service but do not get to choose the date, time or Doctor.
DS, TG, AN	Government office where users can book an appointment for a selected date, but the time is given to them and no assessor is required.

D: Date, T: Time, A: Assessor, S: Selectable, G: Given, N: Not Required.

3.7. Features

Appendix B describes a list of the application features desired by the architecture. Most of these features aim to enhance the user experience. The book appointment intent is the most critical piece of functionality for this architecture as it encompasses the logic early described. Figure 1 describes the appointment intent flow and the structure of the appointment object that is being built.

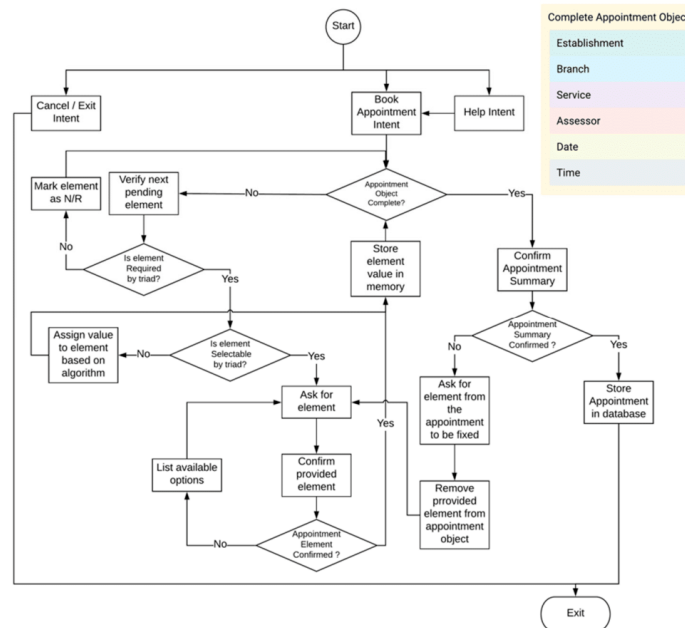


Figure 1. Appointments Architecture Main Structure and Flow.

In the next section, the implementation of the testing prototype and its parts will be described.

4. Architecture Implementation and Configurations

Based on the architecture described in the previous section, an Amazon Echo skill called dental clinic was implemented with the purpose of evaluating its usability. The development of the skill has reached an initial evaluation phase; however, it is worth to mention that this is still a work in progress, for that reason not all the desired features in the architecture were implemented, and more improvements will be added based on the evaluation results.

4.1. Setup

The implemented skill attempts to provide a booking service for a dental clinic using Alexa as a communication channel. This case was selected as it can be easily adjusted to represent different types of appointments during the evaluation process of the skill. The dental clinic has offices in 6 different hospitals, these hospitals served as branches for the skill, a list of 31 services provided by the clinic were included in the skill, also a list of 6 doctors were included to serve as assessors. The lists of branches, services and assessors were included in the voice model and also in the database, also synonyms for all the items were included in the voice model to improve the user experience (see supplementary materials). The branches were set to be open to receive appointments from Monday to Friday from 8 a.m. to 6 p.m. Each appointment was set to have a fixed timespan of 30 min.

4.2. Architecture Overview

The dental clinic skill was developed using the Alexa Skills Kit SDK for Node.js version 2 [18] being this the latest available SDK for Node.js at the time of writing this paper. Node.js version 6.11.0 [19] was selected as development framework mainly due its event-driven, non-blocking I/O model, well-suited for Alexa Skills.

An AWS lambda function is a serverless platform provided by Amazon [20], for this particular skill an AWS lambda function in charge of running backend code in response to events. The code was created using typescript as programming language [21] and deployed to the lambda function. In this particular implementation MySQL is being used as storage mechanism, this database is hosted in an independent server. Like it was mentioned before this implementation can be easily replaced with another database or API.

4.3. Interaction Flow

The user starts the skill by calling one of the invocation phrases supported by Alexa including the skill name and an optional intent [22] e.g., Alexa ask dental clinic to book an appointment.

Voice model utterances are mapped to intent methods. Each time the user speaks, the natural language input is translated to text via the Alexa Skills Kit, if the resulting text matches a valid utterance, the Alexa Skills Kit calls the mapped method for that utterance in the lambda function using an intent request. In the lambda function code, intent methods are defined as middleware functions in the application controllers, there are also services in charge of providing the controllers with external data and helper functions.

Each time an intent method is executed, the control is returned to the Alexa Skills Kit and potentially to the user. The lambda function returns information and behavior commands to the Alexa Skill Kit using the SSML format [23]. The lambda function also communicates with the database server to store the appointments data and retrieve the lists of appointment elements. This database also serves as storage for the appointments dashboard web site. Figure 2 describes the main components of the prototype skill.

At this point of the development process, not all the model features were included in the prototype. Appendix B describes the features included and excluded from the prototype skill.

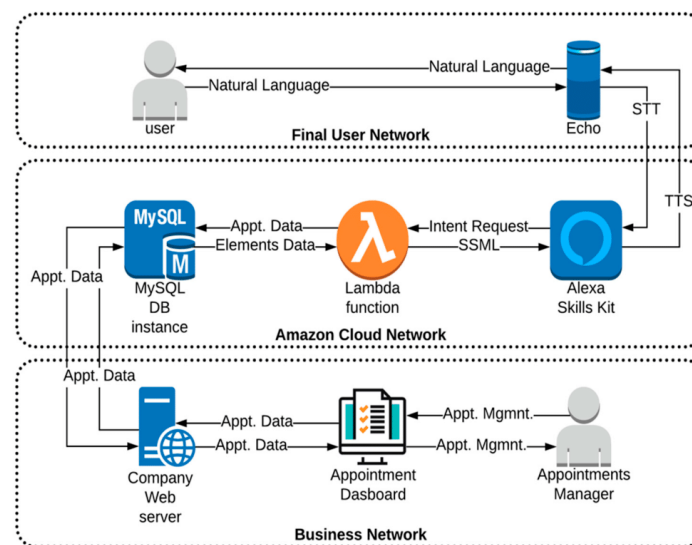


Figure 2. Main components of the prototype skill. TTS: Text-To-Speech, STT: Speech-To-Text.

5. Evaluation Procedure

This section describes the methodology and instrument used to evaluate the prototype skill. An evaluation instrument was created in order to measure the usability of the elaborated prototype when its being used to conduct booking appointment tasks. The instrument uses the SUS system (System Usability Scale) as base adapted for the evaluation of user interfaces by voice, as this system has been proved to be appropriate for evaluating VUIs, offering both convergent and concurrent validity [24]. In addition to the SUS system, open questions were included in the instrument to collect the opinion of the user according to the following elements: positive and negative aspects of the skill, characterization, recommendation of use, emotion, innovation, motivation, level of attractiveness, and global evaluation. The instrument recorded the success or failure in the execution of the tasks performed by the users, the configuration used by each user and the causes of failed tasks, if any.

The prototype skill was evaluated by a group of 24 users selected using the following criteria: being over 18 years old, being of any gender, located in Costa Rica, and having an intermediate or advanced English level. At the time of conducting the evaluation Alexa was not available in Spanish, the language should not be a challenge for the normal use of the skill. The evaluation consists of each user executing two booking tasks using the dental office skill. The first task requested the users to book an appointment using the preferences of their choice, in this case, the users were asked to confirm the appointment reservation in the last step. The second task requested the users to book an appointment using the preferences of their choice, however, in this case, the users were asked to reject the appointment reservation in the last step and then make one or more changes in their appointment, once the changes are made, the users were asked to accept the appointment reservation in the last step. The evaluation was conducted by a single evaluator on private sessions, an Amazon Echo device with a screen was used. Prior to the evaluation, the users were provided with basic instructions on how to use an Alexa device, the purpose of the skill and the two tasks. For each user, the configuration of triads in the skill was changed randomly, this in order to have at least one evaluation for all the reserved booking scenarios supported by the architecture. Once both tasks were completed, the instrument was provided to each user in order to collect their input. The configuration used for each user and the success or failure of each task were also documented in the instrument by the evaluator, an exclusive section in the instrument was devised for this purpose.

6. Results

This section presents the results of the evaluation performed. The results include responses to the System Usability Scale questionnaire and the responses to the open questions. The users age was located between 20 and 40 years old, being 37.5% of them female and 62.5% male.

Most of the users were able to complete both tasks successfully, 83.3% of the users were able to complete the first task, and 75% of the users were able to complete the second task. In terms of failure 75% and 33% of the failures for the first and the second task respectively were caused by an incorrect pronunciation of the voice commands by the users, also 25% and 16.7% of the failures in the first and second task respectively were caused by users providing invalid path commands to the skill, an invalid path command takes Alexa out of the program.

The users were asked what they like the most and the least from the skill, the users liked the most the easiness and speed of the skill, and also not needing to use a laptop or interact with a real person to book the appointment. e.g., *"How easy to learn and simple it was"*, *"How easy and fast it is to use it"*, *"Not having to make a call or turn on the laptop"*. Alexa failing to understand their voice commands and the complexity of providing a date using their voice only were the things users liked the least. e.g., *"It did not catch my accent"*, *"I was confused by the date format expected by Alexa"*. The users were asked to characterize the skill using three words, Figure 3 presents a cloud of the words used by the users. The most relevant words found were easy, useful and fast.

7. Conclusions and Future Work

Few Alexa skills provide the booking appointment functionality, the ones that do rely on a third party to achieve it, creating such skills from scratch pose a challenge for new developers. We created an architecture that serves as base to create booking appointment skills in several business scenarios. A prototype skill was created following the architecture principles and eventually evaluated by a group of people using an instrument based on the System Usability Scale.

The characterization and perception results found allowed us to conclude that the system was positively received by the users and most of the users found it easy, useful and fast. Users believe the application can be easily part of their daily lives and could potentially help them to save time.

The SUS results revealed a high score in terms of properly integrated functions, being the most liked one the fact that appointments get properly booked and no further action is required from the user's part. The user confidence and easiness of use got also a high score in the SUS evaluation, we associate this result to the constant guidance included all over the application, users rarely do not know what to say. Despite that most users find the skill easy to use, 33.33% agree the application is unnecessary complex, further exploration on the user's comments shed some light in this result. A group of users experienced a barrier trying to indicate the appointment's date, no help was provided on how the format for the expected date should be provided. Moreover, the date field is interpreted by Amazon and there is little to no control on its interpretation, we associate the complexity score with the complains about the date input.

The results of the first task are a clear indicator of how easy was for users to pick up the skill for the first time with no previous experience.

The results of the second task indicate that most of the users were able to sort the roadblock of making changes in the appointment data. The fact that the second task was also the second time the users hear the instructions, serves as an example of the skill intuitiveness.

Innovation and attractiveness got the highest score in terms of perception, according to the comments people see the smart assistants as a novelty and this creates an interest in having more useful solutions available in such devices.

The skill got high scores in terms of usage frequency (83.34%), recommendation (91.7%) and overall score (8.20/10), we conclude the architecture serves as good base for creating usable booking skills that can be adapted to different business requirements.

It is the expectation of this work to encourage developers into creating not only better voice experiences for the Amazon Echo, but also applications that help connecting businesses with their customers in a more natural way.

There is plenty of work ahead in the area of smart assistants, future work for this project involves an evaluation of the business experience during the process of booking appointments using the prototype skill and improving the VUI dialogs and responses. An additional evaluation of the architecture with a group of developers could provide some insight on how much time can be saved by using it. Furthermore, at this point, the architecture is generic enough to be implemented in other VUIs that follow the concept of utterances, intents, and sessions. The architecture could be adapted to be used in other smart assistants and also modified to be used in different languages for the Amazon Echo. Finally, the model could be extended to work for other business models.

Supplementary Materials: The following are available online: Voice Model: <https://s3.amazonaws.com/cdn.andresmeza.com/Alexa/AppointmentsModel/en-US.json>, Boilerplate Code: <https://github.com/ameza/alexa-appointments-blueprint>.

Author Contributions: Conceptualization, A.M. and G.L.; Methodology, G.L.; Software, A.M.; Validation, L.Q., and L.A.G.; Data Curation, A.M.; Writing-Original Draft Preparation, A.M.; Writing-Review & Editing, G.L. and L.Q.; Supervision, G.L. and L.A.G.

Funding: This work was funded by Centro de Investigaciones en Tecnologías de la Información y Comunicación (CITIC), Escuela de Ciencias de la Computación e Informática (ECCI) both at Universidad de Costa Rica. Grant No. 834-B6-178. And the APC received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Valid Triads.

Triad
DS, TS, AS
DS, TS, AG
DS, TS, AN
DS, TG, AS
DS, TG, AG
DS, TG, AN
DS, TN, AS
DS, TN, AG
DS, TN, AN
DG, TS, AS
DG, TS, AG
DG, TS, AN
DG, TG, AS
DG, TG, AG
DG, TG, AN
DG, TN, AS
DG, TN, AG
DG, TN, AN

D: Date, T: Time, A: Assessor, S: Selectable, G: Given, N: Not Required.

Appendix B

Model Desired Features

Features included in the prototype skill

- Booking of appointments selecting location, service, assessor, date and time
- Configurable skill to support all the triads described in the model
- Promptly confirmation of elements on each voice interaction
- Functionality to make changes on individual elements before confirming the appointment summary
- Recommendation of starred items on user silences
- Displaying of available options in the Alexa companion App
- Elements data pulled from a MySQL database or alternative storage mechanism
- Synonyms on the voice models for the assessor names, and locations
- Support for different types of date and time input, (incomplete formats are not supported yet. e.g., next week, afternoon)
- Dashboard website to manage the appointments
- Requesting the list of available options at any given time
- Notification via email with summary of appointment.

Features not included in the prototype skill

- Requesting the nearest available date for an appointment
- Requesting the nearest available time in a given day
- Requesting the most popular office for a given service
- Requesting next available date for a given service
- Appointment cancelation using voice
- Appointment conflicts in terms of date and time
- Valid recommendations in terms of date and time
- Requesting help based on context at any given time
- Using the device location for branches

References

1. Wobcke, W.; Nguyen, A.; Ho, V.; Krzywicki, A. The Smart Personal Assistant : An Overview. *Scenario* **2007**, 2005–2006.
2. Amazon Team Amazon.com Help: Set Up Your Echo Available online: <https://www.amazon.com/gp/help/customer/display.html?nodeId=201601770> (accessed on 22 May 2018).
3. Alexa Skills Kit—Build for Voice with Amazon Available online: <https://developer.amazon.com/alexa-skills-kit> (accessed on 22 May 2018).
4. Alexa Voice Service Overview (v20160207) | Alexa Voice Service Available online: <https://developer.amazon.com/docs/alexa-voice-service/api-overview.html> (accessed on 16 December 2018).
5. Amazon.de Amazon Alexa Voice Design Guide Available online: <https://developer.amazon.com/designing-for-voice/design-process/> (accessed on 14 May 2018).
6. Create Intents, Utterances, and Slots | Custom Skills Available online: <https://developer.amazon.com/docs/custom-skills/create-intents-utterances-and-slots.html> (accessed on 22 May 2018).
7. Ghosh, S.; Pherwani, J. Designing of a natural voice assistants for mobile through user centered design approach. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer, Cham, 2015; Volume 9169, pp. 320–331.
8. Abbott, K. VUI Design Principles and Techniques. In *Voice Enabling Web Applications: VoiceXML and Beyond*; Apress: Berkeley, CA, USA, 2002; pp. 87–103, ISBN 978-1-4302-0850-1.
9. Alexa Blueprints Available online: <https://blueprints.amazon.com/help> (accessed on 14 May 2018).
10. Google AI Blog: Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone Available online: <https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html> (accessed on 23 May 2018).
11. Amazon.com: Expedia: Alexa Skills Available online: <https://www.amazon.com/Expedia-Inc/dp/B01M7TT1O8> (accessed on 23 May 2018).
12. Book Hotels, Track Flights, Explore & Search with Amazon Alexa + KAYAK Available online: <https://www.kayak.com/alexa> (accessed on 22 May 2018).
13. Amazon.com: Delta Dental of Kansas: Alexa Skills Available online: <https://www.amazon.com/Delta-Dental-of-Kansas/dp/B07BDRKK5K/> (accessed on 23 May 2018).
14. Amazon.com: Novant Health: Alexa Skills Available online: <https://www.amazon.com/Novant-Health/dp/B0773HPGMX> (accessed on 23 May 2018).
15. Amazon.com: Make Appointment: Alexa Skills Available online: https://www.amazon.com/TsaTsaTzu-Make-Appointment/dp/B073F1541N/ref=sr_1_1?s=digital-skills&ie=UTF8&qid=1527066184&sr=1-1&keywords=book+appointment (accessed on 23 May 2018).
16. Pingup | Be Bookable. Anytime. Everywhere. Available online: <https://pingup.com/> (accessed on 23 May 2018).
17. Nimblr | Much more than appointments, reminders and scheduling Available online: <https://nimblr.ai/home> (accessed on 27 November 2018).
18. Alexa Skills Kit SDK for Node.js | SMAPI Available online: <https://developer.amazon.com/docs/sdk/alexa-skills-kit-sdk-for-nodejs.html> (accessed on 22 May 2018).
19. About | Node.js Available online: <https://nodejs.org/en/about/> (accessed on 25 November 2017).
20. AWS Lambda—Serverless Compute—Amazon Web Services Available online: <https://aws.amazon.com/lambda/> (accessed on 22 May 2018).
21. TypeScript—JavaScript that scales. Available online: <https://www.typescriptlang.org/> (accessed on May 22, 2018).
22. Understanding How Users Invoke Custom Skills | Custom Skills Available online: <https://developer.amazon.com/docs/custom-skills/understanding-how-users-invoke-custom-skills.html> (accessed on 22 May 2018).

23. W3C Speech Synthesis Markup Language (SSML) Version 1.1 Available online: <https://www.w3.org/TR/speech-synthesis11/> (accessed on 22 May 2018).
24. Ghosh, D.; Foong, P. S.; Zhang, S.; Zhao, S. Assessing the Utility of the System Usability Scale for Evaluating Voice-based User Interfaces. In Proceedings of the Sixth International Symposium of Chinese CHI on—ChineseCHI '18, Montreal, QC, Canada, 21–22 April 2018; ACM Press: New York, New York, USA, 2018; pp. 11–15.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

REFERENCIAS

- [1] B. Azvine, D. Djan, K. C. Tsui, and W. Wobcke, "The Intelligent Assistant: An Overview," Springer, Berlin, Heidelberg, 2000, pp. 215–238.
- [2] Amazon Team, "Amazon.com Help: Set Up Your Echo," 2015. [Online]. Available: <https://www.amazon.com/gp/help/customer/display.html?nodeId=201601770>. [Accessed: 22-May-2018].
- [3] "Alexa Skills Kit - Build for Voice with Amazon." [Online]. Available: <https://developer.amazon.com/alexa-skills-kit>. [Accessed: 22-May-2018].
- [4] "Alexa Voice Service Overview (v20160207) | Alexa Voice Service." [Online]. Available: <https://developer.amazon.com/docs/alexa-voice-service/api-overview.html>. [Accessed: 16-Dec-2018].
- [5] Amazon.de, "Amazon Alexa Voice Design Guide." [Online]. Available: <https://developer.amazon.com/designing-for-voice/design-process/>. [Accessed: 14-May-2018].
- [6] "Create Intents, Utterances, and Slots | Custom Skills." [Online]. Available: <https://developer.amazon.com/docs/custom-skills/create-intents-utterances-and-slots.html>. [Accessed: 22-May-2018].
- [7] "Voice User Interface Design Moving from GUI to Mixed Modal Interaction-Ritwik Dasgupta," 2018.
- [8] S. Ghosh and J. Pherwani, "Designing of a natural voice assistants for mobile through user centered design approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9169, pp. 320–331.
- [9] "Google AI Blog: Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone." [Online]. Available: <https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html>. [Accessed: 23-May-2018].
- [10] "Amazon.com: Expedia: Alexa Skills." [Online]. Available: <https://www.amazon.com/Expedia-Inc/dp/B01M7TT1O8>. [Accessed: 23-May-2018].
- [11] "Book Hotels, Track Flights, Explore & Search with Amazon Alexa + KAYAK." [Online]. Available: <https://www.kayak.com/alexa>. [Accessed: 22-May-2018].

- [12] “Amazon.com: Delta Dental of Kansas: Alexa Skills.” [Online]. Available: <https://www.amazon.com/Delta-Dental-of-Kansas/dp/B07BDRKK5K/>. [Accessed: 23-May-2018].
- [13] “Amazon.com: Novant Health: Alexa Skills.” [Online]. Available: <https://www.amazon.com/Novant-Health/dp/B0773HPGMX>. [Accessed: 23-May-2018].
- [14] “Amazon.com: Make Appointment: Alexa Skills.” [Online]. Available: https://www.amazon.com/TsaTsaTzu-Make-Appointment/dp/B073F1541N/ref=sr_1_1?s=digital-skills&ie=UTF8&qid=1527066184&sr=1-1&keywords=book+appointment. [Accessed: 23-May-2018].
- [15] “Pingup | Be Bookable. Anytime. Everywhere.” [Online]. Available: <https://pingup.com/>. [Accessed: 23-May-2018].
- [16] “Nimblr | Much more than appointments, reminders and scheduling.” [Online]. Available: <https://nimblr.ai/home>. [Accessed: 27-Nov-2018].
- [17] G. López, L. Quesada, and L. A. Guerrero, “Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces,” Springer, Cham, 2018, pp. 241–250.
- [18] K. Abbott, “VUI Design Principles and Techniques,” in *Voice Enabling Web Applications: VoiceXML and Beyond*, Berkeley, CA: Apress, 2002, pp. 87–103.
- [19] “About | Node.js.” [Online]. Available: <https://nodejs.org/en/about/>. [Accessed: 25-Nov-2017].
- [20] “TypeScript - JavaScript that scales.” [Online]. Available: <https://www.typescriptlang.org/>. [Accessed: 22-May-2018].
- [21] “MySQL.” [Online]. Available: <https://www.mysql.com/>. [Accessed: 27-Nov-2018].
- [22] “AWS Lambda – Serverless Compute - Amazon Web Services.” [Online]. Available: https://aws.amazon.com/lambda/?sc_channel=PS&sc_campaign=acquisition_CR&sc_publisher=google&sc_medium=english_lambda_b&sc_content=lambda_e&sc_detail=aws_lambda&sc_category=lambda&sc_segment=222658714551&sc_matchtype=e&sc_country=CR&sc_kwid=AL!4422!3!222. [Accessed: 22-May-2018].

- [23] “Amazon Web Services (AWS) - Cloud Computing Services.” [Online]. Available: <https://aws.amazon.com/>. [Accessed: 27-Nov-2018].
- [24] “ASP.NET | The ASP.NET Site.” [Online]. Available: <https://www.asp.net/>. [Accessed: 27-Nov-2018].
- [25] D. Ghosh, P. S. Foong, S. Zhang, and S. Zhao, “Assessing the Utility of the System Usability Scale for Evaluating Voice-based User Interfaces,” in *Proceedings of the Sixth International Symposium of Chinese CHI on - ChineseCHI '18*, 2018, pp. 11–15.
- [26] “Alexa Skills Kit SDK for Node.js | SMAPI.” [Online]. Available: <https://developer.amazon.com/docs/sdk/alexa-skills-kit-sdk-for-nodejs.html>. [Accessed: 22-May-2018].
- [27] “Understanding How Users Invoke Custom Skills | Custom Skills.” [Online]. Available: <https://developer.amazon.com/docs/custom-skills/understanding-how-users-invoke-custom-skills.html>. [Accessed: 22-May-2018].
- [28] W3C, “Speech Synthesis Markup Language (SSML) Version 1.1,” 2010. [Online]. Available: <https://www.w3.org/TR/speech-synthesis11/>. [Accessed: 22-May-2018].