



**BUAP**

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

Tesis para obtener el grado de:  
LICENCIADO EN INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

“Aplicación de Procesamiento de Lenguaje Natural en Centro de  
Visitantes del Centro de Diseño en Guadalajara de Intel”

Presenta:  
Carlos Yael Canizales Castro

Director:  
M.C. Nicolás Quiroz Hernández

Puebla Pue., Octubre 2021



## Índice general

Índice general.....	III
Índice de figuras .....	VII
Índice de tablas .....	IX
Glosario de términos .....	XI
Agradecimientos .....	XIII
Resumen.....	XV
Capítulo 1 Introducción .....	1
1.1    Introducción .....	2
1.2    Objetivos.....	4
1.2.1    Objetivo general .....	4
1.2.2    Objetivos específicos: .....	4
1.3    Justificación .....	5
1.4    Descripción del proyecto de Tesis.....	5
Capítulo 2    Marco teórico .....	9
2.1    Procesamiento del Lenguaje Natural .....	10
2.1.1    Procesamiento de voz.....	10
2.1.2    Reconocimiento de voz.....	10
2.1.3    Reconocimiento de frase clave .....	12
2.1.4    Síntesis de voz .....	13
2.1.5    Aplicaciones .....	13
2.2    Asistentes de voz.....	14
2.2.1    Problemas de privacidad sobre los asistentes de voz .....	16
2.3    Sistemas del Centro de Visitantes.....	16
2.3.1    Sistema Hochob.....	17
2.3.2    Amikoo.....	20
2.3.3    Juum .....	22
2.4    Operación.....	22
2.5    Comparativa de minicomputadoras.....	23

2.6	Comparativa de Micrófonos.....	25
2.7	Selección del sistema para desarrollar interfaz de voz.....	26
Capítulo 3	Desarrollo del sistema Uéej.....	29
3.1	Introducción .....	30
3.2	Análisis del sistema Uéej.....	30
3.3	Diseño del sistema Uéej.....	32
3.4	Diseño de la <i>Persona</i> e interacciones.....	35
3.5	Configuración inicial del hardware .....	40
3.6	Integración del software .....	41
3.6.1	Integración del software en Amikoo .....	41
3.6.2	Configuración del contenedor Rhasspy.....	43
3.6.3	Configuración del contenedor Hochob .....	44
3.6.4	Integración de Rhasspy con Hochob .....	46
3.7	Conclusiones .....	47
Capítulo 4	Resultados del desarrollo del sistema Uéej.....	49
4.1	Metodología de pruebas.....	50
4.2	Pruebas del sistema Uéej.....	52
4.2.1	Primera etapa de pruebas: hardware .....	52
4.2.2	Segunda etapa de pruebas: software.....	53
4.2.3	Tercera etapa de pruebas: integración .....	55
4.2.4	Etapa final de pruebas: comportamiento espacial.....	56
4.3	Resultados de las pruebas realizadas .....	58
4.3.1	Resultados de las pruebas de hardware.....	58
4.3.2	Resultados de las pruebas de software.....	58
4.3.3	Resultados de las pruebas de integración .....	61
4.3.4	Resultados de las pruebas espaciales .....	63
4.4	Análisis de los resultados .....	70
Capítulo 5	Conclusiones y Trabajo Futuro .....	73
5.1	Conclusiones .....	74
5.2	Trabajo futuro .....	75

Apéndice I: Archivo con frases de entrenamiento para Rhasspy .....	77
Apéndice II: Reparación de los componentes actuales .....	80
Audio y video .....	82
Transmisión de audio a Juum .....	83
Bibliografía .....	85



## Índice de figuras

Figura 1: Vestíbulo del Centro de Visitantes de Intel Guadalajara .....	3
Figura 2: Parte del primer pasillo del Centro de Visitantes, Intel GDC .....	4
Figura 3: Zona de Talleres, Intel GDC .....	4
Figura 4: Diagrama de bloques del sistema Uéej .....	5
Figura 5: Diagrama a bloques de los subsistemas de un Sistema de Reconocimiento de Voz Automático .....	10
Figura 6: Diagrama de bloques de los componentes de un Asistente de voz convencional	16
Figura 7: Tarjeta Intel Edison en el robot Amikoo.....	21
Figura 8: Robot Amikoo Lupe .....	22
Figura 9: Diagrama de bloques del proceso para operar los sistemas del Centro de Visitantes. .....	23
Figura 10: Vestíbulo del Centro de Visitantes.....	33
Figura 11: Diagrama a bloques del sistema Uéej, indicando los elementos modificados. ...	34
Figura 12: Diagrama a bloques del Sistema Uéej con los nuevos componentes.....	35
Figura 13: Robot Amikoo con los componentes de hardware .....	40
Figura 14: Contenido del archivo Docker de la imagen HeyAmikoo .....	42
Figura 15: Contenido del archivo Docker Compose del sistema HeyAmikoo.....	42
Figura 16: Contenido del archivo shell encargado del servicio cURL .....	44
Figura 17: Ejemplo de automatización en Home Assistant .....	45
Figura 18: Archivo shell con las acciones de la automatización .....	45
Figura 19: Ejemplo de la declaración de una automatización en la interfaz de Home Assistant .....	46
Figura 20: Interfaz del manejador de intentos en Rhasspy .....	46
Figura 21: Diagrama de bloques de la interacción del Centro de Visitantes con el proyecto Uéej.....	47
Figura 22: Diagrama metodológico de pruebas .....	50
Figura 23: Interfaz de Rhasspy con una petición detectada .....	55

Figura 24: Diagrama de los puntos seleccionados en el vestíbulo del Centro de Visitantes donde se realizarán las pruebas de la detección de audio .....	57
Figura 25: Error de incompatibilidad al construir los contenedores del sistema Hochob ...	80
Figura 26: Fragmento del nuevo conjunto de instrucciones para inicializar Hochob .....	80
Figura 27: Modo seguro de Home Assistant.....	81
Figura 28: Errores de menor prioridad mostrados al iniciar el sistema .....	82
Figura 29: Errores por incompatibilidad de software para desplegar audio y video dentro de Hochob .....	82
Figura 30: Contenido de los archivos de servicios multimedia en Hochob .....	83
Figura 31: Archivo shell para invocar funcionalidades de streaming .....	84
Figura 32: Fragmento del archivo shell del servicio de voz .....	84

## Índice de tablas

Tabla 1: Comparativa de minicomputadoras Upboard, Upboard Squared y Raspberry Pi 3B .....	23
Tabla 2: Comparativa entre los micrófonos ReSpeaker v2 y UMA-8.....	25
Tabla 3: Comparativa de sistemas para crear aplicaciones de voz.....	26
Tabla 4: Estado de los componentes del Centro de Visitantes .....	31
Tabla 5: Estado de las funcionalidades del Centro de Visitantes .....	31
Tabla 6: Interacciones relacionadas a tecnologías de Intel e Intel Guadalajara .....	36
Tabla 7: Interacciones relacionadas al compromiso ambiental .....	37
Tabla 8: Interacciones relacionadas a las áreas de STEM .....	37
Tabla 9: Interacciones conversacionales .....	39
Tabla 10: Pruebas realizadas a los servicios audiovisuales de Hochob y sus respuestas esperadas.....	53
Tabla 11: Parámetros de las pruebas del servicio de movimiento .....	54
Tabla 12: Tabla con eventos y ejemplos de peticiones del sistema Uéej .....	56
Tabla 13: Resultados de las pruebas a nivel hardware.....	58
Tabla 14: Resultados de las pruebas a nivel sistema del software: servicios audiovisuales	59
Tabla 15: Resultados de las pruebas a nivel sistema del software: servicios motrices.....	60
Tabla 16: Resultados de la detección de frase clave en las pruebas de software .....	60
Tabla 17: Resultados de la detección de eventos en las pruebas de software .....	61
Tabla 18: Resultados de la detección de frase clave en las pruebas de integración .....	62
Tabla 19: Resultados de la detección de eventos en las pruebas de integración.....	63
Tabla 20: Resultados de las pruebas de detección de frase clave con sensibilidad 0.5 .....	64
Tabla 21: Resultados de las pruebas de detección de frase clave con sensibilidad 0.559 ...	64
Tabla 22: Resultados de la detección de eventos en la zona A .....	65
Tabla 23: Resultados de la detección de eventos en la zona B .....	66
Tabla 24: Resultados de la detección de eventos en la zona C .....	67
Tabla 25: Resultados de la detección de eventos en la zona D .....	68

Tabla 26: Resultados de la detección de eventos en la zona E .....	69
Tabla 27: Resultados de la detección de eventos en la zona F .....	70
Tabla 28: Porcentajes de detección de palabra clave con valores de sensibilidad en las pruebas espaciales .....	71
Tabla 29: Resumen de los porcentajes obtenidos en las pruebas espaciales de eventos....	71

## Glosario de términos

**4K:** Resolución de pantalla con aproximadamente 4000 píxeles horizontales.

**8K:** Resolución de pantalla con aproximadamente 8000 píxeles horizontales.

**API:** Interfaz de programación de aplicaciones, utilizada para realizar comunicación entre sistemas.

**Amikoo:** Palabra maya que significa “Amigo”. Sistema robótico multimedia parte del Centro de Visitantes en Intel Guadalajara

**ARM:** Arquitectura de computadoras con instrucciones reducidas y de bajo consumo energético.

**DDR3L:** Memoria de acceso aleatorio de tercera generación con voltajes de funcionamiento 1.5 V y 1.35 V, utilizada en dispositivos móviles.

**eMMC:** Sistema de almacenamiento embebido compuesta por memoria *flash*.

**GHz:** Mil millones de ciclos por segundos, esta magnitud es utilizada para describir ondas en aplicaciones relacionadas a radio, audio o la velocidad de operaciones en microprocesadores.

**GPIO:** Entradas y salidas de propósito general.

**HDMI:** Interfaz multimedia de alta definición.

**Hochob:** Palabra maya que significa “lugar donde se almacena el maíz”. Sistema automatizado para control de contenido en museos.

**HTTP:** Protocolo de transmisión de hipertexto.

**I2C:** Protocolo de comunicación serial síncrono multimaestro, multiesclavo de baja velocidad.

**I2S:** Protocolo de comunicación serial síncrono para comunicación de dispositivos de audio.

**MacOS:** Sistema operativo para computadoras de la marca Apple.

**KHz:** Mil ciclos por segundos, esta magnitud es utilizada para describir ondas en aplicaciones relacionadas a radio, audio o la velocidad de operaciones en microprocesadores.

**KWS:** siglas de *keyword spotting*, aplicación del procesamiento de voz para detectar la presencia de una cierta frase en un flujo de audio.

**LPDDR4:** Memoria de acceso aleatorio de cuarta generación de bajo consumo 1.1V

**M.2:** Especificación para tarjetas de expansión en computadoras.

**Mini PCIe:** También conocido como PCI Express Mini Card, es una versión reducida del conector PCI Express para dispositivos móviles.

**NLP:** Procesamiento del Lenguaje Natural.

**NLU:** Comprensión del Lenguaje Natural.

**PWM:** Modulación por ancho de pulsos. Método para transmisión de información a través de un canal de comunicación modificando el ciclo de trabajo de una señal periódica.

**RAM:** Memoria de Acceso Aleatorio.

**SPI:** Protocolo de comunicación serial de cuatro conductores.

**STEM:** Conjunto de áreas de estudio relacionadas a Ciencia, Tecnología, Ingeniería y Matemáticas.

**STT:** Proceso en el que a partir de un flujo de voz, se genera su transcripción.

**TLS:** Seguridad de la capa de transporte. Es un protocolo criptográfico para asegurar comunicaciones en redes de computadoras.

**TTS:** Proceso en el que a partir de una transcripción, se genera un flujo de voz.

**UART:** Dispositivo para proveer comunicación asíncrona serial.

**URL:** Referencia de elementos web que especifica su ubicación en una red de computadoras.

**USB:** Bus serial de comunicación.

**YAML:** Formato para archivos de texto que utiliza indentación para expresar jerarquía.

## Agradecimientos

Al Centro de Diseño de Intel en Guadalajara (Intel GDC) por el préstamo de equipo e instalaciones para la implementación de este proyecto, en particular al Dr. Ramon Morales, quien lideró y apoyó el esfuerzo del Centro de Visitantes de Intel Guadalajara. Además al Ingeniero Abraham Arce Moreno y a la Oficina de Innovación por inspirar, motivar e impulsar la realización de este proyecto.

A la Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación y el Laboratorio de Innovación BUAP-Intel por los apoyos brindados para desarrollar el proyecto en Intel GDC.

A mi asesor, el M. C. Nicolás Quiroz Hernández por su paciencia, conocimientos, apoyo y experiencia que me hicieron crecer como persona y profesionalista.

Y en general a mi familia por el apoyo incondicional para concluir mis estudios profesionales.



## Resumen

El propósito del presente trabajo es desarrollar un sistema interactivo con interfaz de voz para el Centro de Visitantes de Intel GDC. Se dividirá en 4 etapas. Análisis de los sistemas existentes en el Centro de Visitantes y revisión de código; investigación y selección de una plataforma para hacer reconocimiento de voz y de los dispositivos de hardware para ejecutarlo; desarrollo del sistema con base en el software y hardware seleccionado previamente; y por último la integración de todos los sistemas.



## Capítulo 1

### Introducción

Se da contexto del trabajo de tesis a realizar, así como, a la importancia que tiene este desarrollo dentro de la visión de Intel, en su búsqueda por motivar a los niños y jóvenes a estudiar carreras de STEM. Además, se revisan los objetivos del trabajo de desarrollar, se muestra un diagrama de bloques del proyecto completo y se describen las etapas de este.

## 1.1 Introducción

En los últimos años ha cobrado gran relevancia el uso del lenguaje natural para interactuar con sistemas computacionales, ejemplo de ello es el incremento en los sistemas de dialogo, como los asistentes de voz (Alexa, Google Assistant o Siri). El Procesamiento del Lenguaje Natural (NLP por sus siglas en inglés) es una rama de las Ciencias Computacionales, enfocada en hacer que las computadoras puedan emular la capacidad humana de comunicación. Estos sistemas se encuentran instalados en teléfonos inteligentes o bocinas localizadas en espacios comunes de viviendas y son utilizados para realizar tareas concretas a partir de instrucciones de voz de los usuarios (Hirschberg, 2015). Ejemplos de estas tareas son entretenimiento, robótica, educación (Terzopoulos & Satratzemi, 2020), la atención a clientes (Moriuchi, 2019) (Houndify, s.f.), etc. Estas tareas estan generalmente relacionadas con el concepto de Internet de las Cosas (Polyakov E.V., 2018).

Los esfuerzos dentro de la robótica, han generado diversos sistemas como Asimo y Pepper:

- Asimo es un robot humanoide desarrollado por la compañía Honda, tiene capacidades de reconocimiento del entorno exterior, comportamiento autonomo, y movimiento en dos piernas; tiene como finalidad mostrar el potencial de los robots como sistemas de asistencia en tareas que pongan en riesgo a las personas, ademas de, inspirar a jovenes a estudiar carreras relacionadas con STEM (Honda Robotics, s.f.).
- Pepper es un robot comercializado por la empresa japonesa Softbank con capacidades de reconocimiento de sentimientos y de agente conversacional; esta enfocado en areas caseras, empresariales y de educacion. En el ambito comercial funciona como un sistema de atencion a clientes por mas de 250 empresas (Softbank, s.f.).

Ademas, diversos estudios han demostrado que los sistemas robóticos son un mecanismo efectivo para incentivar estudiantes a incorporarse en carreras del area STEM (Ciencia, Tecnologia, Ingenieria y Matematica por sus siglas en inglés) (Plaza, y otros, 2018). Disciplinas que han tomado relevancia a nivel mundial en los ultimos años.

Según previsiones (Alianza para la Promoción de STEM, 2019), el campo laboral relacionado con estas carreras se incrementará año con año, en un ritmo mayor al que las escuelas egresan estudiantes. Dada esta situacion, diversas empresas del ramo tecnologico estan dedicando esfuerzos para incentivar a los estudiantes estudiar carreras del area de STEM, una de ellas es Intel.

Intel y en particular el Centro de Diseño en Guadalajara (Guadalajara Design Center, GDC por sus siglas en inglés), tienen diversas iniciativas cómo lo son, brindar pláticas y talleres a estudiantes y profesores de educación media superior; además de recibirlos en las instalaciones. Un lugar importante en esas visitas, es su Centro de Visitantes.

El Centro de Visitantes, es un espacio enfocado en hablar de la historia de Intel e Intel Guadalajara, las actividades que se desarrollan en sus instalaciones, y explicar diversos conceptos relacionados tecnologías actuales y futuras. Para recorrer este espacio es necesario estar acompañado de al menos un guía, el cual explica cada uno de estos temas.

El espacio del Centro de Visitantes de Intel GDC está conformado por las siguientes áreas: Vestíbulo, Pasillos Temáticos, Zona de Talleres. El vestíbulo es el espacio destinado para dar una introducción al recorrido, y se utiliza un robot (Lupe) para desplegar contenido multimedia en la pantalla y de esta manera reforzar los temas. Dicho robot es controlado por un guía a través de botones en un teléfono inteligente. Además, tiene asientos, una pantalla y un robot Amikoo llamado Lupe, mostrado al fondo en la Figura 1.



*Figura 1: Vestíbulo del Centro de Visitantes de Intel Guadalajara*

Los Pasillos Temáticos son 4, el primero es “Antes del Inicio y el Inicio”, refiere a temas de la historia de Intel e Intel Guadalajara y la evolución de sus procesadores, mostrado en la Figura 2. El segundo pasillo es relacionado con “Los Labs y las Fabs”. Aquí se muestran temas de los procesos de Fabricación (Fabs) y validación (Labs) de productos de Intel. El tercer pasillo es “La ciencia es divertida”. Este pasillo está enfocado en explicar conceptos tecnológicos de una manera sencilla, haciendo uso de aplicaciones interactivas. El ultimo pasillo del recorrido es “Ecosistema”. Este pasillo está enfocado en la relación que tiene Intel Guadalajara con la industria, academia y gobierno.

En cada pasillo existen un conjunto de sistemas multimedia que muestran información mediante aplicaciones interactivas para enriquecer la experiencia los visitantes.



Figura 2: Parte del primer pasillo del Centro de Visitantes, Intel GDC

Zona de talleres, es el espacio donde se llevan a cabo diversas actividades lúdicas o se brindan charlas. Se encuentran 4 pantallas que muestran contenido audiovisual e interactivo, además de un proyector para que el guía brinde explicaciones (Figura 3).



Figura 3: Zona de Talleres, Intel GDC

## 1.2 Objetivos

### 1.2.1 Objetivo general

Se busca desarrollar un sistema interactivo con interfaz de voz para el Centro de Visitantes de GDC con procesamiento de lenguaje natural.

### 1.2.2 Objetivos específicos:

- Investigar y seleccionar una plataforma de software para reconocimiento de voz.
- Investigar las tarjetas de desarrollo para reconocimiento y reproducción de voz.
- Analizar el código actual del Robot Lupe para identificar los cambios requeridos y controlarlo por comandos de voz.

- Implementar un programa para reconocer voz y emitir mensajes de voz con la plataforma de software y tarjeta seleccionada.
- Integrar al robot Lupe los comandos de voz
- Probar el sistema completo por comandos de voz.

### 1.3 Justificación

Intel Guadalajara visualizó la importancia de acercar a los jóvenes las áreas de tecnología, por ello, en el año 2004 inauguró su Centro de Visitantes, el cual recibe anualmente alrededor de 4,000 personas con distintos perfiles: alumnos de educación básica, media, pregrado y posgrado; académicos, investigadores, empresarios, emprendedores, diplomáticos, por mencionar algunos.

En Intel GDC, se utiliza el sistema robótico Amikoo para presentar los temas de Ciencia y Tecnología a los visitantes. Sin embargo, algunos de sus componentes están desactualizados o no funcionan correctamente. Por ello, se vio la oportunidad de agregar un sistema para realizar Procesamiento del Lenguaje Natural, una tecnología de actualidad. Esto permitirá que el robot tenga una interacción más natural con los visitantes, al entender comandos de voz y poder responder a sus preguntas. Esto con la finalidad de poder inspirar a los estudiantes a que estudien una carrera relacionada con las áreas de STEM.

### 1.4 Descripción del proyecto de Tesis.

Como se muestra en la Figura 4, el sistema que se desarrollará en el Centro de Visitantes se llama Uéej y significa “explorar un lugar con interés”. Está compuesto por tres subsistemas: Hochob, Amikoo y Juum

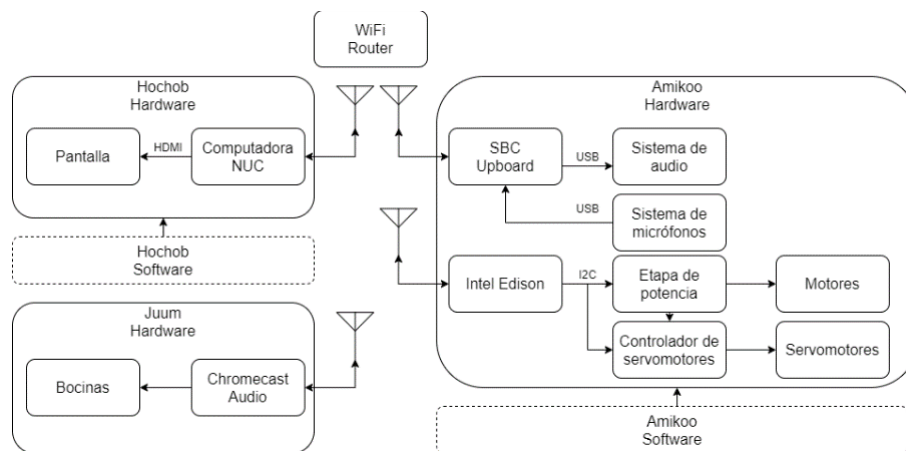


Figura 4: Diagrama de bloques del sistema Uéej

Hochob es un sistema multimedia de código abierto (Organización Hochob, s.f.). Es el encargado de desplegar imágenes, videos y audios en dispositivos como: Pantallas, Bocinas, y Google Chromecast (Audio y Video). Además, es capaz de mandar imágenes, audio, video y voz al robot Amikoo para que este los reproduzca. Otra funcionalidad que tiene Hochob, es mandar instrucciones de movimiento al robot Amikoo. Hochob está instalado en una computadora Intel NUC con sistema operativo Linux. Está conectada a una pantalla, en la cual despliega multimedia.

Amikoo es un sistema robótico de código abierto, desarrollado para motivar en los niños estudiar carreras del área de STEM (Reyes Montero, s.f.). Brinda explicaciones de varios temas mediante voz y son acompañadas con contenido audiovisual. Para llevar a cabo estas funcionalidades, es necesario que un operador (conocido como guía) haga peticiones a través de una aplicación móvil. Esta compuesto por dos minicomputadoras, una Upboard y una Intel Edison. La tarjeta Upboard es la encargada de recibir y desplegar voz y multimedia que recibe de Hochob. La Intel Edison recibe las instrucciones de movimiento, las procesa y haciendo uso del protocolo I2C, mueve los motores necesarios.

Juum es un sistema compuesto por un Chromecast audio y un par de bocinas. Está localizado en los pasillos del Centro de Visitantes. Su función es reproducir voz o audio que recibe del sistema Hochob.

Considerando lo anterior, se busca realizar en este trabajo de tesis lo siguiente:

- Revisar el estado actual de los sistemas de hardware y software de los sistemas del Centro de Visitantes.
- Reparar los sistemas que componen actualmente el Centro de Visitantes
- Investigar alternativas de sistemas de Procesamiento del Lenguaje Natural
- Investigar nuevos componentes de hardware y software para mejorar la interacción del sistema robótico con los visitantes.
- Actualizar los que presenta el robot, además de agregar nuevos.
- Mejorar interacción del guía con el robot
- Integrar los sistemas multimedia del robot con el Centro de Visitantes

En el segundo capítulo se revisa el código y documentación existente. También, se investiga con el personal encargado de realizar los recorridos del Centro de Visitantes las funcionalidades o temas que pueden aportar a la experiencia de las visitas. Una vez realizado esto, se hace una investigación de las plataformas existentes para crear aplicaciones de procesamiento de voz, comparándolas entre sí, para luego enunciar las

ventajas y desventajas que tienen. También, se hará una investigación de diversas computadoras o tarjetas de desarrollo para ejecutar el sistema seleccionado, mencionando sus características, ventajas y desventajas, para luego compararlas entre sí y de esta manera elegir la adecuada. En siguiente capítulo, se desarrollará el sistema de acuerdo con las aportaciones del personal del Centro de Visitantes. Esto con base en el software y hardware elegidos. También se documentarán estos nuevos componentes para la consulta de futuros colaboradores. En el cuarto capítulo, se realizarán diversas pruebas para evaluar la capacidad de respuesta del sistema en diversas circunstancias.



## Capítulo 2

### Marco teórico

Se presenta la definición una investigación de lo que representa un asistente de voz, además de los sistemas Hochob y Amikoo. También se realizará una comparación de las plataformas existentes para crear aplicaciones de procesamiento de voz, enunciando las ventajas y desventajas que tienen. Para después comparar diversas tarjetas de desarrollo para ejecutar el sistema seleccionado. Por último, se agrega la retroalimentación que tienen los empleados del sistema actual.

## 2.1 Procesamiento del Lenguaje Natural

Es un área de investigación y aplicación que explora técnicas para entender y manipular habla o texto proveniente de lenguaje natural utilizando computadoras. Se fundamenta en diferentes disciplinas como Ciencias Computacionales, Lingüística, Matemáticas, Ingeniería Eléctrica y Electrónica, Robótica, Psicología, por mencionar algunas. Algunas aplicaciones del Procesamiento del Lenguaje Natural (NLP por sus siglas en inglés) son: procesamiento y reconocimiento de voz, traducción, obtención de información a partir de varios idiomas, etcétera. (Chowdhury, 2003).

### 2.1.1 Procesamiento de voz

Conjunto de técnicas para el procesamiento digital de señales, aplicadas al procesamiento y análisis de las señales del habla. Estas técnicas toman como base el complejo sistema auditivo, neuronal y respiratorio para diversas aplicaciones. Siendo algunas de ellas el reconocimiento automático de voz, el reconocimiento de frases y la síntesis de voz. (Soumya, Anjan, & Nilanjan, 2019).

### 2.1.2 Reconocimiento de voz

Metodología en la cual una serie de palabras provenientes de un hablante son convertidas a texto, con el fin de reconocer y procesar su contenido. Una parte importante para considerar es que este proceso no lleva a cabo una comprensión del audio (Soumya, Anjan, & Nilanjan, 2019).

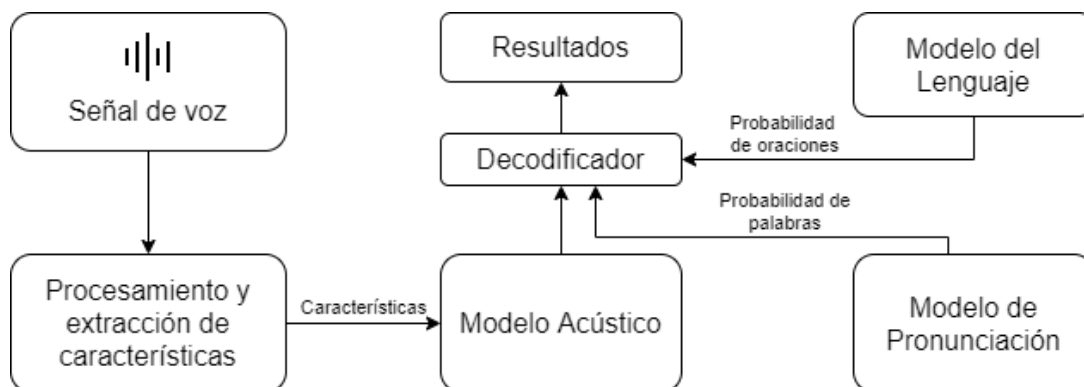


Figura 5: Diagrama a bloques de los subsistemas de un Sistema de Reconocimiento de Voz Automático

Para conseguir esta conversión, es necesario considerar los pasos mostrados en la Figura 5, que se describen a continuación:

- Análisis acústico

La voz generada por los seres humanos es de carácter analógico, esto requiere una conversión analógico-digital para su procesamiento en sistemas digitales como las computadoras. Una vez realizado esto, se aplican filtros para reducir o eliminar el ruido y se divide en fragmentos de 10 a 40 milisegundos, cuyo dominio con respecto al tiempo será convertido a uno con respecto a frecuencias. A este resultado, le serán generados vectores de características, para utilizarse en el modelo acústico (Yu & Deng, 2015).

- Modelo acústico

En este bloque se conserva una relación estadística entre señales de audio y fonemas (unidades básicas del habla). Esta relación se genera a partir de un entrenamiento con una base de datos de voz. Los fonemas en los sistemas de reconocimiento de voz son utilizados, debido a que permiten al sistema ser más flexible ante palabras desconocidas. (Soumya, Anjan, & Nilanjan, 2019)

Dada la salida del análisis acústico, el modelo acústico genera un conjunto de fonemas con sus probabilidades, de acuerdo con la relación estadística que este modelo contiene. Para realizar el cálculo de estas probabilidades existen principalmente dos metodologías: modelos ocultos de Márkov (por sus siglas en inglés HMM.) y redes neuronales.

- Modelo de pronunciación

El modelo de pronunciación se encarga de organizar los fonemas para formar palabras con un significado correcto, con base en un diccionario con un gran número de palabras y su correspondiente secuencia de fonemas. Este diccionario es construido por lingüistas. En esta fase, no se realiza un entrenamiento. (Soumya, Anjan, & Nilanjan, 2019)

- Modelo de lenguaje

El modelo de lenguaje ayuda a organizar palabras para formar una oración con significado, tomando en cuenta un modelo estadístico que indica que tan probable es que una secuencia de palabras pueda aparecer junta, también solucionando ambigüedades entre acústicas similares. (Soumya, Anjan, & Nilanjan, 2019)

Una de las metodologías más utilizadas en este modelo es N-Gramas, donde se usa para predecir la n-ésima palabra tomando en cuenta las “n-ésima menos un” palabra (Soumya, Anjan, & Nilanjan, 2019).

- Decodificador

De acuerdo con, el vector de características del modelo acústico y la secuencia de palabras propuesta del modelo del lenguaje, el decodificador genera una secuencia de palabras con la mayor probabilidad (Yu & Deng, 2015).

### 2.1.3 Reconocimiento de frase clave

El reconocimiento de frase clave (KWS por sus siglas en inglés), se conoce como el conjunto de técnicas para identificar palabras específicas en un flujo de habla continua (voz, multimedia, entre otros). Existen diferentes técnicas para realizar KWS, entre los que resaltan:

- LVCSR

El mecanismo de reconocimiento de habla continua de dominio extenso (LVCSR por sus siglas en inglés) produce una completa transcripción del material y el reconocimiento de palabra clave se realiza en el texto resultante. (Moyal, Aharonson, Tetariy, & Gishri, 2013)

- Acústico

En el enfoque acústico, no se realiza una transcripción del material entrante, en cambio se hace la búsqueda directo en el audio. Es usado principalmente para reconocimiento de un conjunto pequeño de palabras clave (Moyal, Aharonson, Tetariy, & Gishri, 2013).

- Búsqueda fonética

La búsqueda fonética consiste en convertir flujo entrante de audio a un conjunto fonemas y después buscar el conjunto de fonemas que coincidan con el generado a partir de la palabra clave (Moyal, Aharonson, Tetariy, & Gishri, 2013) .

#### 2.1.4 Síntesis de voz

Los sistemas de síntesis de voz son los encargados de tomar un determinado texto para producir señales de voz y comunicar algo, por lo anterior, también se les conoce como sistemas de texto a voz (TTS por sus siglas en inglés). Para realizar la comunicación de manera efectiva, es necesario que la voz resultante sea natural e inteligible. Estos sistemas pueden tener gran importancia en aplicaciones donde se requiera interacción no visual del usuario, como en el caso de dispositivos de navegación, ayudas auditivas, y asistentes inteligentes (Sreenivasa Rao & Narendra, 2019). Los sistemas TTS están compuestos por dos módulos:

- Análisis de texto y lingüística

En este módulo, se realiza la identificación de abreviaciones, signos de puntuación y acrónimos; para luego interpretarlos y removerlos del texto. Cada palabra es transformada a fonemas, para después, generar información que contenga entonación, duración y pausas, las cuales serán utilizadas por el siguiente modulo: el generador de ondas (Sreenivasa Rao & Narendra, 2019).

- Generación de ondas

De acuerdo con la información generada por el análisis de texto y lingüística, este módulo genera el habla resultante. Es el encargado de proveer la naturalidad e inteligibilidad al habla (Sreenivasa Rao & Narendra, 2019).

#### 2.1.5 Aplicaciones

Si bien, las aplicaciones de procesamiento de voz han aparecido desde los años 60 (Soumya, Anjan, & Nilanjan, 2019), a partir de finales de los años 90 se ha generado un aumento en la investigación en esta área. Cuatro de los factores clave para este aumento han sido, el incremento en la capacidad computacional, la disponibilidad de cantidades ingentes de información lingüística, el desarrollo de algoritmos de aprendizaje de máquina y un mejor entendimiento de la estructura del lenguaje humano y su uso en contextos sociales (Hirschberg, 2015).

También, los factores antes mencionados, han influido en la democratización de estas aplicaciones, pues actualmente, en la vida diaria de las personas se pueden encontrar aplicaciones como sistemas de autenticación de voz para seguridad, traducción automática

(Hirschberg, 2015), transcritores de voz, subtítulos automáticos en videos (Soumya, Anjan, & Nilanjan, 2019), asistentes de voz (Hoy, 2018), etcétera.

Una de las aplicaciones que más ha ganado popularidad en los últimos 10 años son los asistentes inteligentes de voz (también conocidos simplemente como asistentes de voz), que se encuentran en los teléfonos inteligentes. Estos asistentes de voz tienen la función de comunicarse con dispositivos y servicios para realizar tareas. Algunos ejemplos de esas tareas son: enviar o recibir mensajes, responder peticiones acerca del clima, controlar servicios como Spotify o Netflix y controlar dispositivos de Internet de las Cosas (Hoy, 2018).

## 2.2 Asistentes de voz

Un asistente de voz es un agente de software programado para reconocer peticiones de voz de los usuarios. Estos se encuentran en dispositivos especializados como bocinas inteligentes, o bien en teléfonos inteligentes, por mencionar algunos (Hoy, 2018). El uso de asistentes de voz se ha incrementado debido a su utilización con productos de Internet de las Cosas (focos inteligentes, televisiones inteligentes, sistemas de seguridad inteligentes, etc.).

Debido a que estos sistemas interactúan con personas, se recomienda que los agentes tengan una persona y una personalidad, con la finalidad de tener una conversación más natural y mejorar la experiencia del usuario (Google, s.f.). Un ejemplo de ello es el asistente de voz Alexa, cuya persona es la de una mujer joven, que incluso llega a tener preferencias en ciertos temas (Amazon, s.f.).

Como se muestra en la Figura 6, los asistentes de voz como Google Assistant o Alexa están conformados por los siguientes subsistemas:

- Palabra o frase de activación

Aplicación de KWS que mantiene al dispositivo en un estado de escucha constante, al detectar una coincidencia con la frase o palabra clave, pasa al modo de escucha de acciones.

Con motivo de mejorar la experiencia de usuario, el dispositivo emite una alerta sonora o visual para indicar que fue activado.

La detección de la frase clave se realiza de manera local en los dispositivos.

- Escucha de acciones

El modo para la escucha de acciones realiza una grabación de audio, donde el usuario realiza peticiones. En el caso de los asistentes asistidos por nube como Google o Alexa, envía esas grabaciones a sus respectivos servicios de nube para realizar el procesamiento. Este procesamiento incluye filtros y algoritmos para el reconocimiento automático de voz o de comprensión del lenguaje natural (NLU por sus siglas en inglés).

- Manejador de intentos

Las muestras anteriormente recolectadas y procesadas son evaluadas, de acuerdo con las posibles acciones que pueda realizar el asistente de voz, y la que tenga mayor probabilidad de ser la que el usuario pidió, se intentará realizar esa acción.

Se conocen como intentos, debido a que existe la posibilidad que el sistema determine que la acción no existe o no puede realizarse.

- Acciones.

Las acciones para realizar pueden ser servicios de la compañía desarrolladora del asistente, o servicios de terceros. Algunos ejemplos de acciones que realizan los asistentes de voz son el seguimiento de paquetes, activación de dispositivos IoT, lectura de noticias, peticiones a servicios de música o video o transacciones.

- Texto a voz

Para mejorar la experiencia del usuario, los asistentes de voz utilizan TTS como respuesta a la petición del usuario. Las respuestas pueden ser que la acción: se realizó correctamente, no se pudo realizar, no se encontró, que el sistema requiere información adicional para realizarla o confirmación para su realización.

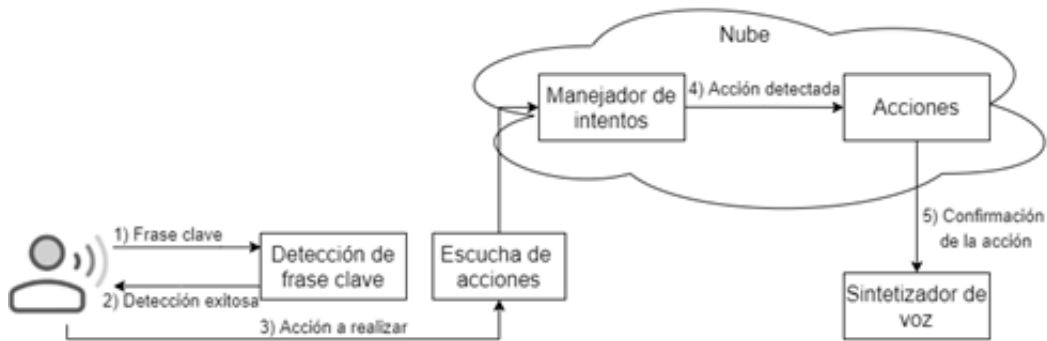


Figura 6: Diagrama de bloques de los componentes de un Asistente de voz convencional

### 2.2.1 Problemas de privacidad sobre los asistentes de voz

Diversos medios de comunicación como *CNET* (NG, 2019) y *The Guardian* (Lynskey, 2019), además de, instituciones e investigadores (Chung, Iorga, Voas, & Lee, 2017) (Hoy, 2018) han documentado y estudiado diversos incidentes provocados por problemas de privacidad relacionados con el manejo de la información que recopilan los asistentes de voz. Uno de estos problemas, es la publicación de información sensible de los usuarios, entre la que se incluye nombre, dirección, etcétera. Otro problema relevante, es la grabación de conversaciones en las cuales el asistente no fue activado.

Una de las acciones realizadas por las compañías desarrolladoras para resolver estas problemáticas, ha sido cambiar sus lineamientos, para que la información sea borrada después de un determinado tiempo o que el usuario borre su información cuando lo decida. Sin embargo, también se ha demostrado que las empresas conservan las muestras de voz, incluso tiempo después de ser borradas (Makena & Statt, 2019).

Esta incertidumbre ha causado que cada vez haya más opciones de código abierto a sistemas para desarrollar asistentes de voz. Algunos de ellos realizando el procesamiento de manera remota y otros de manera local. Para fines de este trabajo, se utilizará la opción que realiza el procesamiento de manera local.

### 2.3 Sistemas del Centro de Visitantes

Como se muestra en la Figura 4, actualmente en el centro de visitantes existen diversos sistemas para ejecutar las diferentes aplicaciones. Siendo estos sistemas:

### 2.3.1 Sistema Hochob

Hochob es un sistema multimedia de código abierto para sistemas Linux. Su finalidad es ser un recurso narrativo inspirador para estudiar carreras relacionadas con áreas de STEM (Organización Hochob, s.f.).

El sistema Hochob está conformado por los siguientes dispositivos:

- Intel NUC

Computadora de tamaño reducido (aproximadamente 12cm x 11cm x 5cm) fabricada por Intel. Su principal característica es que puede ser colocada en espacios reducidos (por ejemplo, aplicaciones tipo quiosco). Algunos modelos cuentan con procesadores utilizados en computadoras de escritorio convencionales. Otra característica es que algunos modelos cuentan con puertos USB 3.0, HDMI y DisplayPort para conectar periféricos (Intel Corporation, s.f.). Este dispositivo es el encargado de manejar las aplicaciones del sistema Hochob.

- Chromecast audio

Dispositivo fabricado por Google para la transmisión inalámbrica de audio. Es compatible con sistemas Android, iOS, Windows, Chrome OS y Linux. Para su funcionamiento es necesario conectarlo a dispositivos de audio con conector Jack 3.5 milímetros, debido a que el Chromecast audio no tiene altavoces (Google, s.f.).

- Chromecast video

Dispositivo fabricado por Google para transmisión de inalámbrica video o imágenes. Es compatible con sistemas Android, iOS, Windows, Chrome OS y Linux. Tiene un puerto HDMI para conectarse en pantallas. Proporciona una resolución máxima 1920 x 1080 píxeles (1080p) a 60 cuadros por segundo (Google, s.f.).

- Cámara web USB

El sistema Hochob también cuenta con diversos servicios como:

- Transmisión de audio a Chromecast
- Transmisión de video a Chromecast

- Despliegue de video local
- Despliegue de imágenes locales
- Despliegue de sonidos locales
- Integraciones con Amikoo
- Sintetizador de voz

Los servicios de Hochob se ejecutan a través de contenedores. Estos contenedores son unidades de software donde se aíslan programas y sus dependencias, permitiendo una ejecución autónoma en el sistema operativo; siendo posible ejecutar múltiples contenedores en un mismo sistema operativo compartiendo el *kernel*. (Docker, 2020)

Una herramienta para realizar desarrollos con la tecnología de contenedores es Docker, la cual permite crear y manejar estos contenedores a través de una consola interactiva.

- Docker

Plataforma para el desarrollo, entrega y ejecución de aplicaciones. Permite empaclar y ejecutar aplicaciones aisladas en espacios llamados contenedores. Estos contenedores son instancias ejecutables con software específico, tienen la posibilidad de comunicarse con otros contenedores y tener almacenamiento. Para crear contenedores, es necesaria una imagen, la cual son instrucciones para crear un contenedor. Estas imágenes representan software concreto, ejecutado internamente en un sistema operativo. Para la creación de imágenes personalizadas es requerido crear un archivo *Dockerfile*, el cual tiene una sintaxis que indica en que imagen se basará, que software se agregará y que acciones se realizaran al inicializarse (Docker, 2020).

Sin embargo, cuando existe más de un contenedor en un mismo desarrollo, se recomienda el uso de Docker Compose, una herramienta para definir y ejecutar aplicaciones con múltiples contenedores de Docker. Permite configurar los servicios de las aplicaciones haciendo uso de un archivo YAML. Y una vez que el archivo es generado, se utiliza una interfaz en línea de comandos para construir entornos, iniciarlos o detenerlos (Docker, 2020).

El sistema Hochob está conformado por las siguientes imágenes de Docker:

- Mosquitto

Aplicación de código abierto para envío y recepción mensajes. Esta aplicación es comúnmente utilizada en aplicaciones con bajo consumo. (Eclipse Foundation, s.f.).

Mosquitto hace uso de MQTT, el cual es un protocolo de comunicación enfocado a desarrollos de Internet de las Cosas. Provee un método para el envío de mensajes a través de un modelo publicación/suscripción haciendo uso de tópicos, los cuales funcionan como identificadores. Es considerado un protocolo ligero, ya que los clientes pueden ser desde computadoras hasta microcontroladores. Provee mecanismos para Calidad de Servicio, y encriptación por TLS (MQTT, 2020).

- Grafana

Aplicación de código abierto utilizada para consultar, visualizar y generar alertas a partir de métricas y registros. Es compatible con bases de datos de datos como Graphite, e InfluxDB. También es posible utilizar este software para monitorizar servicios de nube (Grafana Labs, 2020).

- Home Assistant

Software de código abierto para automatización enfocado en casas, está enfocado en proveer control local para ayudar a la privacidad de los usuarios. Cuenta con aproximadamente 2000 integraciones, que en gran parte son creadas por su comunidad. Estas integraciones pueden conectarse tanto a servicios remotos, como a servicios locales. Tiene soporte para diversos protocolos de comunicación de red como HTTP, MQTT; Zigbee, Z-wave; y dispositivos comerciales como Google Chromecast (Home Assistant, s.f.).

- Influxdb

Base de datos de código abierto para almacenar datos con series de tiempo. Está diseñada para soportar alto volumen de peticiones de lectura y escritura (InfluxData, 2020).

- Node-RED

Herramienta de programación basada en flujos originalmente desarrollada por IBM. Se utiliza a través de un navegador web. Debido a su abstracción, permite acercar la programación a más personas, gracias a su representación visual de los componentes y funcionalidades (OpenJS Foundation, s.f.).

- Glances

Software de código abierto para la monitorización. Se enfoca en presentar la información de manera accesible, puede ser consultada a través de terminal o interfaz web. (Hennion, s.f.).

### 2.3.2 Amikoo

Robot de código abierto, desarrollado para incentivar el estudio de carreras relacionadas con el área de STEM (Reyes Montero, s.f.). Está formado por dos subsistemas: el motriz y el multimedia.

El sistema motriz está conformado por un cuerpo de aluminio donde se alojan: 2 baterías de 12V 7Ah para alimentar el sistema, 2 motores DC conectados a un puente H para el desplazamiento del robot, un convertor de voltaje de 12V 3A a 6V 3A para alimentar los 7 servomotores MG995 y de esta manera generar el movimiento de las articulaciones. El sistema es capaz de tener movimiento en 5 articulaciones (cabeza, hombros y brazos) que son controlados por una tarjeta Intel Edison. El sistema multimedia es capaz de, sintetizar voz, desplegar video y audio, además de tener comunicación con el sistema Hochob. Está conformado por una cámara USB con micrófono, altavoces USB y una pantalla táctil. El sistema está controlado por una tarjeta Upboard.

Intel Edison es una tarjeta de desarrollo diseñada por Intel, enfocada en prototipado y desarrollo de aplicaciones de Internet de las Cosas y productos vestibles. La tarjeta es a la vez una computadora con un procesador Intel Atom de 32 bits a 500 MHz. y un microcontrolador Intel Quark a 100 MHz. Tiene 40 GPIO, conectividad wifi y bluetooth integrada, también soporta protocolos de comunicación UART, I2C, SPI, I2S. Para realizar prototipados, es necesario conectar la tarjeta Intel Edison a placas llamadas *breakouts*, que permiten acceder a los GPIO y conexiones de manera sencilla. Con respecto a su programación, se realiza a través de su sistema operativo Yocto, el IDE de Arduino, o en el entorno de desarrollo XDK. (Intel Corporation, 2015).

En el sistema Amikoo, como se muestra en la Figura 7, la tarjeta Intel Edison es la encargada de controlar la parte motriz. La tarjeta recibe mensajes en protocolo MQTT, donde cada mensaje especifica la acción a realizar y la tarjeta indica que motores deben moverse.

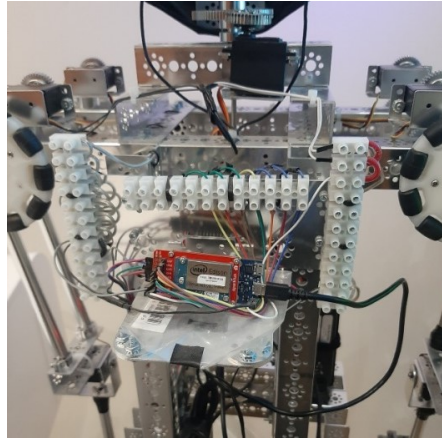


Figura 7: Tarjeta Intel Edison en el robot Amikoo

Para mover los diferentes motores, la Intel Edison está conectada a un *breakout* para tener conectividad PWM de hasta 8 servomotores y un puente H para controlar los dos motores encargados del desplazamiento del robot.

La tarjeta de desarrollo Upboard, es una minicomputadora de alto desempeño y bajo consumo. Utiliza un procesador Intel Atom x5-Z8350 de 64 bits y hasta 1.92 GHz, además, tiene una GPU integrada para ejecutar gráficos en 3 Dimensiones. Con respecto al apartado de conectividad, incluye 1 puerto Gigabit Ethernet, 4 puertos USB tipo A 2.0, 1 USB OTG 3.0, 1 puerto HDMI, 1 puerto MIPI-CSI, 1 puerto DSI, así como, 40 pines de propósito general con la misma distribución de una Raspberry Pi. (AAEON Europa, 2020)

En el sistema Amikoo, la tarjeta Upboard es la encargada de tomar fotografías, desplegar imágenes y video en su pantalla táctil. Además, en este dispositivo se reproducen archivos de audio y voz. La voz se genera a partir de mensajes con el mensaje a reproducir, enviados con el protocolo de comunicación MQTT.

En el Centro de Visitantes se encuentran dos sistemas Amikoo con los nombres Lupe y Norman, estos robots son sistemas similares, por lo que el trabajo se enfocará en solamente uno de ellos

### 2.3.2.1 Lupe y Norman

Lupe y Norman son dos sistemas Amikoo que se encuentran en el Centro de Visitantes. Cuentan con las mismas características, sin embargo, difieren en que Lupe (Figura 8), cuenta con una *Persona* establecida por los desarrolladores. Esta *Persona*, es la de un robot estudiante, sin género definido, el cual es tímido y vive apasionado por aprender; también tiene pasatiempos como jugar y leer.

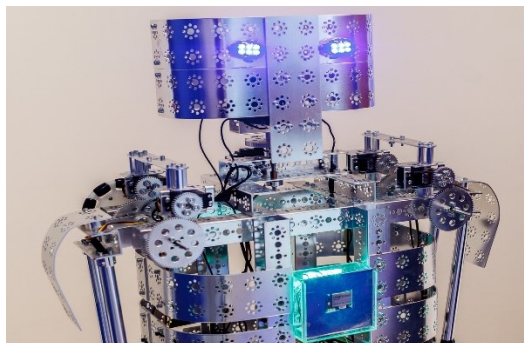


Figura 8: Robot Amikoo Lupe

### 2.3.3 Juum

Juum es un sistema de audio compuesto por un Chromecast audio y un par de bocinas. Está localizado en los pasillos del Centro de Visitantes. Su función es reproducir voz o audio que contiene la descripción de los elementos mostrados en el Centro de Visitantes. El audio es mandado por el sistema Hochob a través de la red inalámbrica. Juum es la palabra maya para sonido.

## 2.4 Operación

Como se muestra en la Figura 9, la manera de operar estos sistemas es haciendo uso de un teléfono celular con la aplicación de Home Assistant. Esta aplicación, se comunica con el sistema Home Assistant de Hochob; donde se encuentran listadas las acciones de los sistemas Amikoo y Hochob. Por este motivo, siempre es necesario un operador para ejecutar las funciones. Estos operadores son los empleados encargados del Centro de Visitantes.

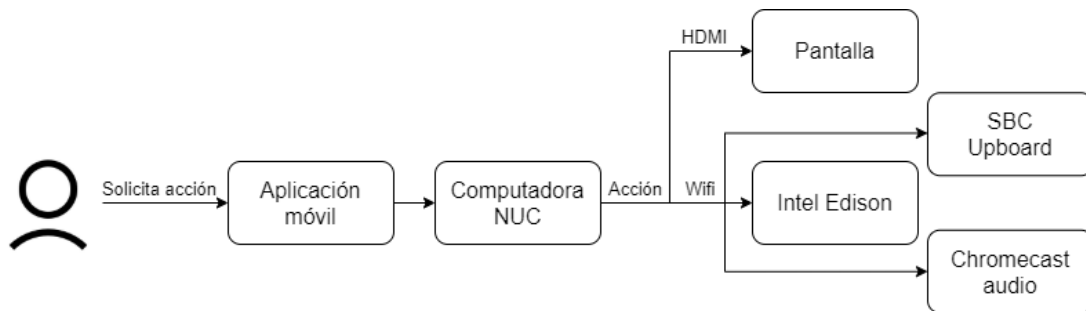


Figura 9: Diagrama de bloques del proceso para operar los sistemas del Centro de Visitantes.

## 2.5 Comparativa de minicomputadoras

Existen diversos modelos de minicomputadora, con características similares a la tarjeta Upboard utilizada por el sistema Amikoo. En la Tabla 1, se comparan las características de la tarjeta Upboard, su versión actualizada Upboard Squared, y la Raspberry Pi 3B.

Tabla 1: Comparativa de minicomputadoras Upboard, Upboard Squared y Raspberry Pi 3B

Comparativa de minicomputadoras			
Nombre	Upboard	Upboard Squared	Raspberry Pi 3 B
<b>Procesador</b>	Intel Atom x5 Z8350 Cuatro núcleos a 1.44 GHz.	Atom X7-E3950 Cuatro núcleos a 1.6 GHz.	Broadcom BCM2837. cuatro Núcleos a 1.5 GHz. Tipo ARM.
<b>Memoria</b>	1 GB a 4 GB de RAM DDR3L	1 GB a 4 GB de RAM LPDDR4	1 GB RAM LPDDR2
<b>Almacenamiento</b>	16 GB a 64 GB tipo eMMC	32 GB hasta 128 GB eMMC	No tiene, se necesita una tarjeta microSD
<b>GPIO</b>	40 pines	40 pines	40 pines
<b>RTC</b>	Si	Si	No
<b>Consumo</b>	5 V a 4 A	5 V a 4-6 A	5 V a 2.5 A
<b>Conectividad</b>	RJ45, HDMI, 4 x USB 2.0 tipo A, 1 x USB 3.0 OTG Micro B	RJ45, HDMI 1.4, DP 1.2, 3 x USB 3.0 tipo A, 1x mPCIe, 1 x USB 3.0 OTG MicroB, 1 x M.2	RJ45, HDMI, 4 x USB 2.0, Wifi, Bluetooth, Jack 3.5 mm para salida de audio
<b>Sistemas Operativos Soportados</b>	Windows 10, Ubinlinux, Ubuntu, Yocto, Android	Windows 10, Ubinlinux, Ubuntu, Yocto, Android	Raspberry OS, Ubuntu, LibreElec
<b>Conectividad (dispositivos)</b>	Cámara (MIPI-CSI), Pantalla táctil (MIPI-DSI)	2 x Cámara (MIPI-CSI2), RealSense Serie 435i, VPU Myriad X vía mPCIe	Cámara (CSI), Pantalla táctil (DSI)

Nota: Información de las minicomputadoras Upboard obtenida de (AAEON Europa, 2020), Upboard Squared obtenida de (AAEON Europa, 2020) y Raspberry obtenida de (Fundación Raspberry Pi, 2020)

Algunas de las características que tienen estas tarjetas en la conectividad MIPI CSI, MIPI-DSI, mPCIe para dispositivos VPU y puertos de alta velocidad para dispositivos como cámaras infrarrojas como la línea RealSense.

CSI (Camera Serial Interface) es un protocolo desarrollado por la alianza MIPI, una organización colaborativa encargada de desarrollar dispositivos móviles y sus relacionados. Su finalidad es comunicar cámaras con dispositivos (celulares, automóviles inteligentes, dispositivos del Internet de las Cosas, equipo médico, etcétera) manteniendo un alto desempeño y bajo consumo energético. De acuerdo con la especificación, permite resoluciones de video 1080p, 4K, 8K y más, además de, fotografía de alta resolución. (Alianza MIPI, 2019).

DSI (Display Serial Interface) es un protocolo desarrollado por la alianza MIPI. Define una interfaz serial de alta velocidad entre el procesador del dispositivo y una pantalla, las características de esta interfaz son: bajo consumo, baja interferencia electromagnética y manteniendo la compatibilidad de dispositivos entre diferentes marcas. Este protocolo es usado en relojes inteligentes, video juegos, y dispositivos empujados (Alianza MIPI, 2019).

Las VPU o unidades de procesamiento para visión, son aceleradores de instrucciones para aplicaciones con inteligencia artificial, son utilizados junto a procesadores de propósito general de bajo consumo (Rivas-Gomez, Pena, Moloney, Laure, & Markidis, 2018) para aplicaciones como cámaras inteligentes o dispositivos de Internet de las Cosas. Existen diversas VPU enfocadas en *edge computing*, un paradigma en que los dispositivos realizan un procesamiento y análisis a la información que obtienen o manejan, en vez de enviar esta información completa a sus respectivos servidores (IBM, s.f.). Un ejemplo de estas VPU son los dispositivos Myriad X desarrollados por Intel, los cuales ofrecen un rendimiento de más de 4 TOPS (*Trillion operations per second* o billón de operaciones por segundo en español), además de, tener una alta eficiencia energética (Movidius e Intel, 2018).

RealSense es una línea de cámaras desarrolladas por Intel para aplicaciones de visión por computadora, estas cámaras tienen un lente de alta resolución y un lente infrarrojo que puede ser utilizado para medir profundidad. Estas cámaras pueden ser utilizadas tanto en entornos cerrados y entornos en exterior. Son utilizadas para desarrollos de autos autónomos, robótica y en algunos casos para desarrollo de interfaces de usuario. (Intel, s.f.). Una de estas cámaras es la modelo D435i, que además de los lentes de alta resolución e infrarrojo tiene una unidad IMU (Unidad de Medida Inercial). Esta unidad permite hacer un refinamiento en las mediciones de profundidad, está enfocada a aplicaciones de robótica y

drones. Permite una resolución máxima de 1920 x 1080, con video hasta 30 cuadros por segundo; usa una conexión USB 3.1 tipo C de generación 1. (Intel, s.f.)

## 2.6 Comparativa de Micrófonos

Tomando en cuenta que el sistema Amikoo se encuentra en un espacio semiabierto, en el que puede existir ruido externo y eco, es necesario comparar opciones que ayude con estas problemáticas. En la Tabla 2, se realiza la comparativa de sistemas con arreglo de micrófonos.

Tabla 2: Comparativa entre los micrófonos ReSpeaker v2 y UMA-8

Comparativa de arreglo de micrófonos		
Nombre	ReSpeaker v2	UMA-8
Procesador	XMOS XVF-3000	XMOS XVF-3000
Tipo de micrófonos	Microelectromecánico	Microelectromecánico
Numero de micrófonos	4	7, permitiendo expandir el numero
Conectividad	Puerto micro USB para alimentación y comunicación, conector Jack 3.5 mm para salida o entrada de audio	Puerto mini USB para alimentación y comunicación. Soporta protocolo I2S
Algoritmos implementados	<i>Beamforming</i> , dirección de llegada, cancelación acústica de eco, dereverberación	<i>Beamforming</i> , dirección de llegada, cancelación acústica de eco, detección activa de voz, dereverberación
Consumo energético	5 V 180 mA	5 V
Tamaño	Diámetro de 70 mm	Diámetro de 90 mm
Tasa de muestreo	Máximo 16 kHz	11/16/32/44.1/48 kHz

Nota: Información del micrófono ReSpeaker obtenida de (Seed Studio, 2020) y del micrófono UMA-8 de (miniDSP, 2020)

Una de las ventajas de usar arreglos de micrófonos, es la posibilidad de obtener mejor calidad en la grabación de voz, debido a que en las tarjetas se aplican diferentes técnicas para reducir el ruido, el eco y la reverberación. Además, es posible obtener más información como la localización y numero de las fuentes de audio, entre otros (Dey & Ashour, 2018). Existen diferentes configuraciones en los arreglos de micrófonos, como lo son lineal, circular, concéntrica, esfera, por mencionar algunos. Una de las configuraciones más utilizadas es la circular, la cual permite tener una resolución espacial para todo el azimut (Lai, Nordholm, & Leung, 2017).

Una de las técnicas aplicables en arreglos de micrófonos es el *beamforming*. El *beamforming* es un filtro espacial que consiste en reconstruir la señal de habla, a partir de una grabación multicanal. Esta reconstrucción muestra ser efectiva en un número entre 4 y 8 micrófonos (Dey & Ashour, 2018, págs. 15-17). El también *beamforming* puede ser utilizado para otros propósitos, como lo son la detección de señales, estimación de la dirección de llegada, y mejora de una señal deseada a partir de una señal corrompida por ruido, concurrencia de fuentes, y reverberación (colección de sonidos reflejados en las superficies de un espacio cerrado) (Benesty, Jingdong, & Huang, 2008, pág. 39).

## 2.7 Selección del sistema para desarrollar interfaz de voz

Para crear aplicaciones de voz, existen alternativas comerciales y de código abierto. Las alternativas comerciales más conocidas son Alexa y Google Assistant, las cuales tienen servicios de Nube donde procesan la voz de los usuarios para convertirla en texto y realizar la comunicación con las diversas acciones que se intentan realizar. Por otra parte, existen alternativas de código abierto que realizan una parte de todo el procesamiento de voz o incluso todo de manera local, dos de estas alternativas son Mycroft y Rhasspy.

En la Tabla 3 se comparan los anteriormente mencionados sistemas para desarrollar aplicaciones de voz.

Tabla 3: Comparativa de sistemas para crear aplicaciones de voz

<b>Comparativa de sistemas para crear aplicaciones de voz</b>				
<b>Nombre</b>	Alexa	Google Assistant	Mycroft	Rhasspy
<b>Desarrollador</b>	Amazon	Google	Mycroft AI Inc.	Comunidad de Rhasspy
<b>Idiomas</b>	8	20	Más de 68	16
<b>Procesamiento</b>	Nube	Nube	Local y Nube	Local o Nube
<b>Código Abierto</b>	No	No	Parcial	Si
<b>Manejo de Privacidad</b>	Los usuarios pueden borrar su información	Los usuarios pueden borrar su información	Los usuarios pueden borrar su información	Los usuarios pueden borrar su información
<b>Algoritmos Utilizados (KWS)</b>	Redes Neuronales Convolucionales	Redes Neuronales Profundas	Redes Neuronales Recurrentes	Depende de la plataforma o software a utilizar

<b>Algoritmos Utilizados (STT)</b>	Redes Neuronales Recurrentes	Redes Neuronales Profundas	Depende de la plataforma a utilizar	Depende de la plataforma o software a utilizar
<b>Dispositivos donde se puede ejecutar</b>	Dispositivos compatibles con Alexa, Raspberry Pi, dispositivos con sistema Windows, Mac, Ubuntu, Android.	Computadoras con sistema operativo Windows, Mac y Linux	Computadoras Linux, Windows y Mac. Se incluyen dispositivos como RaspberryPi	Computadoras Linux, Mac y Windows. Incluida RaspberryPi

Nota: Información de las características de los asistentes de voz obtenida de Alexa (Amazon Alexa, 2020), (Amazon, 2020), (Amazon Science, 2020), (Amazon Alexa, 2020), (Raju, Filimonov, Tiwari, Lan, & Rastrow, 2019); Google Assistant (Google Assistant, 2021) (Google Assistant, 2020), (Kěpuska & Bohouta, 2018); Mycroft (Mycroft, 2020) y Rhasspy (Rhasspy, 2021).

\* Investigaciones han demostrado que en algunos casos la información ha sido usada para otros propósitos que los descritos, a pesar de las cláusulas de manejo de privacidad.

Para el caso de Alexa y Google Assistant, son servicios de voz totalmente en la Nube, los cuales pueden ser instalados en dispositivos como la computadora Raspberry Pi, únicamente con fines de prototipado. Estos servicios requieren una cuenta vinculada para su funcionamiento.

Mycroft permite realizar la instalación de su *software* en computadoras con pocos recursos como la Raspberry Pi, con la finalidad de crear clientes similares a una bocina inteligente, donde el reconocimiento de frase clave es realizado de manera local (KWS), mientras que la conversión de habla a texto es realizada por servicios externos como Google Cloud, IBM Watson y wit.ai.

Rhasspy funciona de manera modular, donde cada parte del proceso de un asistente de voz es realizado por una aplicación diferente, permitiendo tener diversas opciones, en su mayoría ejecutándose de manera local. Las opciones de servicios disponibles para el idioma español son las siguientes:

- Detección de frase clave

Las aplicaciones para realizar detección de frase clave son las siguientes: Raven, Porcupine, Snowboy, Precise, Snowboy, Pocketsphinx. Algunas de las características que tienen estas aplicaciones es rendimiento, personalización y la posibilidad realizarlo totalmente en el dispositivo. En el apartado del rendimiento las aplicaciones

mencionadas anteriormente presentan un buen rendimiento, exceptuando Pocketsphinx, esto de acuerdo con la documentación (Rhasspy, 2021). Acerca de la personalización, Porcupine únicamente permiten detectar de ciertas frases clave, y si se requiere alguna otra solo se puede hacer a nivel personal (Picovoice, 2021); en el caso de Precise y Raven, se puede hacer la personalización sin ningún costo, sin embargo, el proceso de Precise es más complejo.

- Voz a texto:

Las aplicaciones para convertir voz a texto son Pocketsphinx y Kaldi. Pocketsphinx y Kaldi hacen el reconocimiento de voz en el dispositivo, sin requerir servicios externos. En algunos casos pueden presentarse problemas de rendimientos en dispositivos Raspberry Pi. Pocketsphinx en este sistema está configurado para utilizar un modelo entrenado, permitiendo a los usuarios crear sus propios modelos. La aplicación de Kaldi está formada por una imagen Docker generada de una copia preconstruida. (Rhasspy, 2021)

- Reconocimiento de intentos:

Las opciones de aplicaciones para el reconocimiento de intentos son Fsticuffs, Fuzzywuzzy, Snips NLU, RasaNLU, Mycroft Adapt, Flair. De estas aplicaciones, las que presentan un mejor rendimiento (velocidad de entrenamiento y velocidad de detección) son Fsticuffs, Fuzzywuzzy y Snips NLU. Fsticuffs tiene como característica ignorar palabras desconocidas, Fuzzywuzzy relaciona cadenas con el conjunto de entrenamiento de manera difusa, y Snips NLU maneja relaciones con palabras o entidades desconocidas.

- Manejo de Intentos

Para este servicio, únicamente se puede utilizar Home Assistant. Home Assistant es capaz de interactuar con los eventos o con los intentos generados por Rhasspy, para integrarlo es requerida una clave de API.

Cabe mencionar, que Rhasspy únicamente reconoce frases que se encuentren en su conjunto de entrenamiento. Este conjunto utiliza un archivo con extensión *ini*, y una sintaxis flexible para valores opcionales, múltiples, grupales, por mencionar algunas.

## Capítulo 3

### Desarrollo del sistema Uéej

Se presenta la propuesta de diseño del sistema Uéej con los componentes elegidos de hardware y software. Además, se muestra la implementación del sistema Uéej con nuevos contenidos didácticos

### 3.1 Introducción

Como paso previo al desarrollo del sistema Uéej, se realizará un análisis del estado actual de los sistemas del Centro de Visitantes. A partir de esto se realizará la propuesta de diseño, siendo la principal modificación, la adición de una interfaz de voz con elementos de Procesamiento de Lenguaje Natural. Estos elementos incluirán reconocimiento de voz para invocar acciones de los sistemas Amikoo, Hochob y Juum, permitiendo una interacción más natural y menos dependiente de una aplicación móvil; para el software de reconocimiento de voz, se valorarán diferentes alternativas. Además de la interfaz de voz, la propuesta de diseño incluirá la creación de peticiones, acciones y la *Persona*. Otro punto por considerar en la propuesta será la adición de un arreglo de micrófonos (presentando diferentes alternativas) y la sustitución de componentes no funcionales en los sistemas actuales.

### 3.2 Análisis del sistema Uéej

Como primer paso para el desarrollo del sistema Uéej, se realizó un análisis del estado actual de los sistemas que componen el Centro de Visitantes y de sus funcionalidades. El análisis de las funcionalidades y subsistemas consistió en invocar las diferentes funcionalidades, inspeccionar el código fuente y utilizar software para verificar el envío de información a través del protocolo MQTT.

El sistema del Centro de Visitantes está compuesto por 3 subsistemas: Amikoo, Hochob y Juum (como se explica en la sección 2.3 de este trabajo). Hochob es el sistema responsable de la interfaz de usuario, procesar peticiones de la interfaz, además de, enviar estas peticiones y el respectivo contenido multimedia a los diferentes subsistemas o a la pantalla principal. Los subsistemas que interactúan con Hochob son Amikoo y Juum, estos funcionan como dispositivos de Hochob. Juum recibe archivos de audio a través de la API de Chromecast y Amikoo recibe multimedia o instrucciones de movimiento por MQTT. Los audios que recibe Juum tienen la finalidad de crear un recorrido automatizado por el Centro de Visitantes, mientras que Amikoo (Robot Lupe) recibe frases para convertir a voz, audio para reproducir en su sistema de audio, así como, imágenes y videos para presentar en una pantalla. En la tabla 4 se presenta el estado de los componentes del Centro de Visitantes y en la tabla 5 las funcionalidades de los sistemas con su respectivo estado.

Tabla 4: Estado de los componentes del Centro de Visitantes

<b>Estado de los componentes en el Centro de Visitantes</b>		
<b>Componentes</b>	<b>Sistema</b>	<b>Estado</b>
NUC	Hochob	Correcto
Pantalla	Hochob	Correcto
Intel Edison	Amikoo	Correcto
Sistema motor	Amikoo	Correcto
Upboard	Amikoo	No funcional
Sistema de audio	Amikoo	Correcto
Chromecast audio	Juum	Correcto
Sistema de audio	Juum	Correcto

Tabla 5: Estado de las funcionalidades del Centro de Visitantes

<b>Estado de las funcionalidades en el Centro de Visitantes</b>		
<b>Funcionalidades</b>	<b>Sistema</b>	<b>Estado</b>
Interfaz de usuario	Hochob	Correcto
Despliegue multimedia en pantalla	Hochob	Correcto
Generación de audio para Juum	Hochob	Textos faltantes
Envío de audios a Juum	Hochob	No presente en el sistema
Envío de textos para conversión a voz en Amikoo	Hochob	Correcto
Envío de acciones motoras	Hochob	Correcto
Reproducción de audio (Hardware)	Juum	Correcto
Conversión de texto a voz	Amikoo	Presente en el sistema
Acciones motrices (Hardware)	Amikoo	Correcto

Sin embargo, como se muestra en la

Tabla 4 y Tabla 5, algunos de los estos componentes y funcionalidades no son operantes, no están presentes en el sistema funcional o solo están presentes en el código en componentes no funcionales. En el caso de Hochob, si bien el software es funcional en la NUC donde está instalado, el repositorio presenta diversos errores causados por incompatibilidades. Con respecto al sistema Amikoo, la tarjeta Upboard donde se ejecuta parte del sistema no funciona, y realizando observaciones en el código fuente de Hochob y Amikoo, no hay presencia del material didáctico (audio, videos, imágenes), además, existen únicamente dos interacciones de voz. De las funcionalidades relacionadas con Juum, no se efectúa la comunicación entre el sistema Hochob y Juum, debido a que no se encuentran los textos que generan los audios a enviar y tampoco el software que realiza el envío de estos.

Una vez realizado el análisis de los componentes y sistemas existentes, se consultó a los empleados del Centro de Visitantes a través de entrevistas, cuáles son las mejoras que ellos consideran deben desarrollarse en los sistemas, a lo que ellos respondieron que podría mejorarse la interacción de los usuarios con la aplicación móvil o buscar alguna otra interfaz, reparar los sistemas para realizar recorridos automatizados en un futuro y agregar diversos contenidos didácticos relacionados con STEM, compromiso ambiental y tecnologías relacionadas a Intel e Intel Guadalajara.

### 3.3 Diseño del sistema Uéej

A partir del análisis realizado se determinó integrar una interfaz de voz, con la finalidad de realizar procesamiento y reconocimiento de voz, además de, remplazar la tarjeta Upboard. Otras acciones que se realizaran son la reparación del repositorio de Hochob, la reparación de las funcionalidades encargadas de generar y enviar audio a Juum, así como, integrar nuevos contenidos didácticos para Amikoo y la pantalla de Hochob.

Las motivaciones detrás de la elección de la interfaz de voz y la búsqueda de incluir procesamiento y reconocimiento de voz son que actualmente los sistemas de voz permiten una interacción más natural con los usuarios, un ejemplo de ello son los asistentes de voz (aunque deben tomarse en cuenta sus problemas de privacidad). Otra motivación es la distribución del vestíbulo, como se muestra en la Figura 10, el espacio es semicerrado y cuenta con mobiliario, por lo que una interfaz de voz permitiría la misma interacción entre los visitantes desde cualquier parte del vestíbulo, contrastando con la interacción generada por una aplicación móvil o una interfaz táctil. El lugar donde se localizará el sistema de micrófonos es en el robot, esto con la finalidad de generar una interacción directa con el robot.



Figura 10: Vestíbulo del Centro de Visitantes

Algo a considerar en la integración de una interfaz de voz y reconocimiento de voz, son las diversas interacciones, ya que estas serán las que provoquen naturalidad en la comunicación. Para crear estas interacciones entre el sistema y los usuarios se deben diseñar peticiones, acciones y la *Persona*; todo esto a partir de los contenidos didácticos. Los contenidos didácticos incorporarán temas relacionados a STEM, compromiso ambiental y tecnologías relacionadas a Intel e Intel Guadalajara. Las peticiones serán las diferentes maneras en que los usuarios piden al robot que realice acciones. Las acciones serán las diferentes respuestas a estas peticiones, pudiendo ser con voz o multimedia a través de los diversos dispositivos que componen los sistemas Hochob, Amikoo o Juum. La *Persona* indicará la manera de las respuestas de voz, teniendo en cuenta el personaje y personalidad.

Para el desarrollo del sistema Uéej, es necesario seleccionar un sistema de procesamiento de voz teniendo en cuenta las opciones de la sección 2.7. Las dos primeras opciones son Alexa y Google Assistant, las cuales como se explica en la sección 2.2.1, mantienen información de las conversaciones de los usuarios, generando preocupación por la información que pudiera ser grabada. Por este motivo no serán consideradas para este desarrollo y se compararán las otras opciones: Mycroft y Rhasspy. Mycroft es un sistema de código abierto, aunque también envía información de las consultas a diversos servicios de Nube, provocando la misma preocupación mencionada anteriormente, a diferencia de esto, Rhasspy permite realizar sistemas de procesamiento y reconocimiento de voz de manera local, por lo que, se decide utilizar Rhasspy para este desarrollo.

De acuerdo con la sección 2.3 y la Figura 11 de este trabajo, los sistemas Hochob y Amikoo contienen componentes de software del Centro de Visitantes, los cuales es necesario reparar y modificar (señaladas de color verde en la Figura 11). Para conseguir la integración de esta interfaz de voz se necesita sustituir la minicomputadora Upboard (señalada de color azul en la Figura 11) y agregar el arreglo micrófonos (señalado de color naranja en la Figura 11).

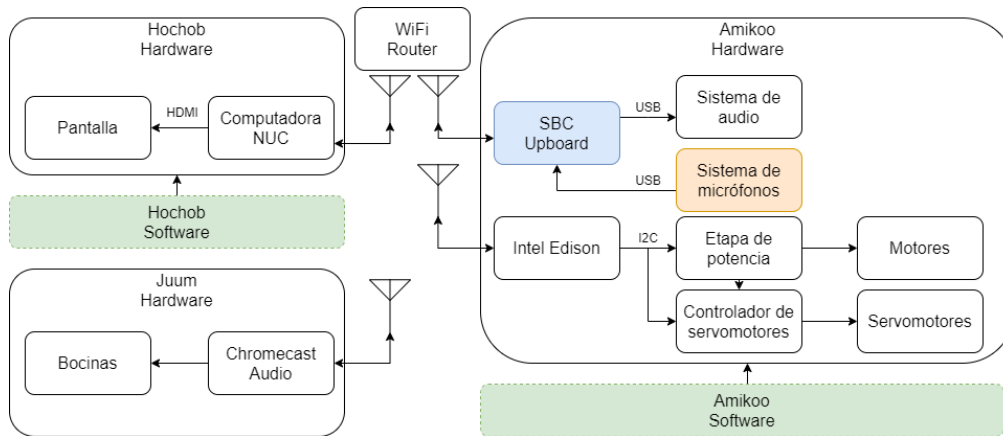


Figura 11: Diagrama a bloques del sistema Uéej, indicando los elementos modificados.

Como se muestra en la sección 2.5, las opciones comparadas para sustituir la minicomputadora existente son Upboard, Upboard Squared y Raspberry Pi 3. El primer punto para considerar, de acuerdo con la documentación de Rhasspy, pueden presentarse problemas de rendimiento al ejecutar algunos de sus subsistemas en las minicomputadoras Raspberry Pi 3 (Rhasspy, 2021), por lo tanto, será necesario elegir una opción con mejores características para escalar las funcionalidades del proyecto en futuras versiones. Las alternativas restantes son Upboard y Upboard Squared, siendo esta última una actualización de la Upboard con memoria RAM LPDDR4, puertos USB 3.0, un procesador con mayores frecuencias, y conectores mini PCIe y M.2 para periféricos. Esta selección de tecnologías, hacen posible la actualización del desarrollo de este trabajo para futuras versiones, permitiendo integrar otras tecnologías como la visión por computadora. Por estos motivos la Upboard Squared será la minicomputadora elegida para este desarrollo. El sistema operativo que se instalará en la minicomputadora será Ubuntu 16.04, debido a que es el utilizado anteriormente y debido a que la documentación de la Upboard Squared está enfocada en él.

Una vez elegido la minicomputadora y el software de procesamiento de voz, se elige el arreglo de micrófonos que realizara la adquisición de voz. Entre las opciones de arreglos de micrófonos se encuentran ReSpeaker v2 y UMA-8. Como se muestra en la sección 2.6, estos dos arreglos realizan procesamiento del audio en las tarjetas para reducir la complejidad en el proceso de adquirir voz, se diferencian en la cantidad de micrófonos, algoritmos implementados y tasa de muestreo. El ReSpeaker v2 tiene 4 micrófonos integrados mientras que el UMA-8 tiene 7 micrófonos integrados, además, este último soporta diversas tasas de muestreo (mayores y menores al ReSpeaker v2), algo que aportará flexibilidad al desarrollo de futuras versiones de este trabajo, por lo que se elige el arreglo UMA-8 para este proyecto.

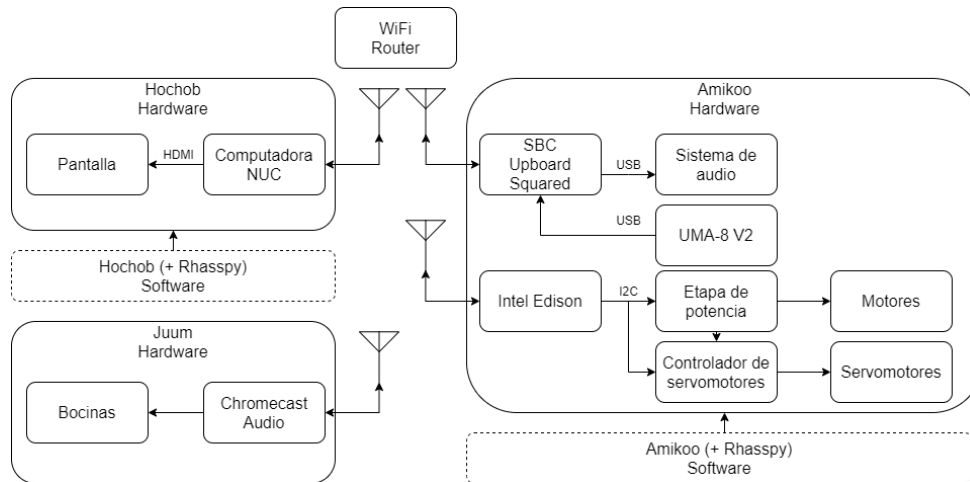


Figura 12: Diagrama a bloques del Sistema Uéej con los nuevos componentes

Con los dispositivos y sistema de procesamiento de voz elegidos, el sistema Uéej estará conformado como indica la Figura 12. La minicomputadora Upboard Squared tendrá conectados a través de USB el sistema de audio y el arreglo de micrófonos UMA-8, estos dispositivos se alimentan a través de esta misma conexión, y en el caso del UMA-8, realiza el procesamiento de la voz adquirida en el dispositivo para mejorar la calidad de voz. La minicomputadora Upboard Squared ejecutará una versión modificada del sistema Amikoo con el software Rhasspy, conjuntándolos en un sistema de contenedores de Docker. Para lograr este sistema de contenedores, será necesario migrar el sistema Amikoo a un contenedor; Rhasspy no lo requiere, ya que es la manera en que se distribuye.

El sistema Hochob tendrá modificaciones en su repositorio para hacerlo funcional, además, se realizará la reparación de las comunicaciones de este sistema con Juum, integrando Stream2Chromecast. También, se realizarán modificaciones en Hochob para hacer el manejo de las acciones que serán invocadas por el sistema de voz y procesadas por Amikoo, recibiendo únicamente la acción activada. Además, integrarán y actualizarán servicios de comunicación HTTP para comunicarse con la interfaz de voz de Rhasspy.

Amikoo y Hochob serán los encargados de recibir las peticiones de voz de los usuarios, procesarlas y efectuar las interacciones de acuerdo con las temáticas didácticas y la *Persona* diseñada.

### 3.4 Diseño de la *Persona* e interacciones.

A partir de la consulta realizada a los empleados acerca de los temas didácticos que deben incluirse en este desarrollo, se obtuvieron los siguientes: STEM, compromiso ambiental y

tecnologías relacionadas a Intel (global) e Intel Guadalajara. Estos temas didácticos serán utilizados para diseñar la *Persona* y las interacciones.

La versión actual de *Persona* del robot es un estudiante de educación básica, tímido, con conocimientos en las áreas de STEM, sin género y guía del Centro de Visitantes. Sin embargo, para este sistema el robot Lupe será visto como un ingeniero joven del área de validación, sin género, alegre, deportista, interesado en cuidar el medio ambiente y apasionada por aprender cosas nuevas y compartir su conocimiento. Esta *Persona* refleja de mejor manera los temas didácticos y facilitara la creación de las acciones.

Las interacciones que tendrá el sistema Uéej se componen de las peticiones y las acciones. Las peticiones serán las diferentes maneras en que un usuario puede solicitar información a Lupe. Las acciones serán diseñadas de acuerdo con la *Persona* diseñada. Las interacciones quedaran de la siguiente manera. En la temática de tecnologías relacionadas a Intel e Intel Guadalajara, se explicará que es un microprocesador, el rol que tiene en una computadora, se hablará acerca de los procesos de fabricación y validación de productos, del Centro de Visitantes y los elementos que se encuentran ahí (Tabla 6). Del tema del compromiso ambiental, se explicarán las medidas que tiene Intel Guadalajara con respecto a su compromiso ambiental, como lo son el reciclaje, uso de energías renovables y tratamiento de agua (Tabla 7). Acerca de la temática de STEM, se busca inspirar a los visitantes menores de 18 años a estudiar una carrera afín a estas áreas, hablando de nuevas tecnologías como la Inteligencia Artificial y algunas de sus aplicaciones (Tabla 8). Además de estas temáticas, es necesario incluir interacciones relacionadas a la presentación del robot con los visitantes (de acuerdo con la *Persona* diseñada), interacciones conversacionales de apoyo, y el inicio del tour a los visitantes (Tabla 9). Las peticiones respectivas a las interacciones se muestran en el Apéndice I; al contener cada interacción diversas combinaciones de peticiones, se elegirá una para mostrarse en las Tabla 6 a la Tabla 9, correspondientes a las interacciones. En conjunto la *Persona*, las peticiones, y acciones permitirá tener a los visitantes una interacción más natural con el sistema.

Tabla 6: Interacciones relacionadas a tecnologías de Intel e Intel Guadalajara

Interacciones relacionadas a tecnologías de Intel e Intel Guadalajara		
Peticiones	Respuesta	Contenido auxiliar
¿A qué se dedica Intel?	Intel es conocida por hacer procesadores para las computadoras que usamos diario, pero eso es una pequeña parte. Intel en su búsqueda por mejorar la vida de las personas, crea y busca como mejorar componentes para computadoras y servidores como	Ninguno

	procesadores y memorias. También desarrolla equipo especializado para mejorar aplicaciones con Inteligencia Artificial, un ejemplo son los dispositivos inteligentes y los autos autónomos	
¿Qué se hace en GDC?	En Guadalajara los ingenieros se encargan de revisar que los procesadores y demás productos de Intel funcionen adecuadamente. Intel Guadalajara es un centro de validación, no es una fábrica	Imagen de Intel GDC
¿Sabes qué es un micro-procesador?	Es el cerebro que controla una computadora, por ejemplo la que me hace funcionar. ¿Sabían que Intel hace los procesadores más avanzados del mundo? Y ni siquiera son más grandes que las uñas de sus dedos	Imagen mostrando un microprocesador con un cerebro. Girar cabeza, señalarla con su brazo izquierdo y volver a estado inicial
¿Qué hay en el Centro de Visitantes?	Aquí conocerán el pasado, el presente y el futuro de Intel. Al final encontrarán una sección donde podrán jugar y maravillarse con la ciencia. También encontrarán un piano cósmico	Video con imágenes del Centro de Visitantes
¿Qué es un Piano cósmico?	El piano cósmico es un instrumento muy especial. Lo tocan partículas que caen del espacio llamadas Muones	Imágenes del piano cósmico
¿Qué es un muon?	Los muones son partículas elementales que se encuentran en los rayos cósmicos, tienen un tiempo de vida de 2.2 microsegundos. Han sido utilizados para conocer a detalle volcanes o cámaras ocultas en pirámides egipcias	Video simulando muones cayendo del espacio

Tabla 7: Interacciones relacionadas al compromiso ambiental

#### Interacciones relacionadas al compromiso ambiental

Peticiones	Respuesta	Contenido Auxiliar
Qué cosas hacemos para cuidar el medio ambiente	Estos edificios tienen una certificación LEED Gold. Lo que significa que tomamos medidas para cuidar el ambiente. Por ejemplo, tenemos una planta de tratamiento de agua, la basura es separada reciclada. Además, apoyamos transportes alternativos como el transporte público y las bicicletas. También se están instalando paneles solares	Imágenes de la planta de tratamiento de agua, contenedores con basura separada y paneles solares

Tabla 8: Interacciones relacionadas a las áreas de STEM

#### Interacciones relacionadas a las áreas de STEM

<b>Peticiones</b>	<b>Respuesta</b>	<b>Contenido Auxiliar</b>
Qué es la inteligencia artificial	Es llevar un pedazo de la inteligencia de las personas a las computadoras, usando matemáticas. Y en algunos casos es más fácil de lo que creen	Ninguno
Para que sirve la inteligencia artificial	Con la inteligencia artificial, las computadoras ayudan a las personas a hacer cosas más rápido y mejor. También hace a las computadoras ver, hablar, o escuchar, así como lo hago yo	Ninguno

Tabla 9: Interacciones conversacionales

Interacciones conversacionales		
Peticiones	Respuesta	Contenido Auxiliar
Saluda a los visitantes	Bienvenidos a mi casa, su casa. Es un gusto conocerlas y conocerlos. Soy Lupe, soy guía en el emocionante Centro de Visitantes de Intel. Estoy lista para mostrarles lo interesante que es el mundo de las Ciencias, la Tecnología, las Ingenierías y las matemáticas.	Imagen con bienvenida. Levantar brazo izquierdo simulando saludo y volver a estado inicial
Preséntate	Soy guía en el emocionante Centro de Visitantes de Intel. Estoy lista para mostrarles lo interesante que es el mundo de las Ciencias, la Tecnología, las Ingenierías y las Matemáticas. Estoy programada para resolver las dudas que tengas, y hago café	Ninguno
Prepara café	Brrrrrrrr... ¿Acaso parezco una cafetera?	Imagen de café
Quien es tu creador?	Soy producto del esfuerzo de ingenieras e ingenieros que han trabajado aquí. A veces veo a uno de ellos... pero olvidé su nombre	Imágenes de un equipo de personas trabajando
Que te gusta hacer en tu tiempo libre	En mi tiempo libre me gusta hacer Yoga, ¿Sabían que aquí dan clases de Yoga? También me gusta correr, aunque más bien ruedo	Imagen de paisajes con el robot
Cuenta un chiste	Chistes	Imagen de un escenario
Nos tenemos que ir	Frase de despedida	Imagen mostrando "gracias por visitarnos"
Inicia el recorrido	Ahora empezaremos con el recorrido, les pido de favor que todos permanezcan juntos para que puedan escuchar la explicación y si tienen alguna duda con confianza pueden preguntar, pues una de las cosas que Intel valora mucho es la curiosidad por aprender y saber más. La curiosidad ha hecho que ingenieros hagan invenciones para resolver problemas y hacer nuestra vida más práctica y sencilla. Por favor pasen, mi voz los guiará por el recorrido	Audios del recorrido.

Una vez creadas las interacciones, se abordará la configuración de los componentes para integrarlas en los sistemas.

### 3.5 Configuración inicial del hardware

Los componentes de hardware que serán instalados en el robot son la minicomputadora Upboard Squared y el arreglo de micrófonos UMA-8. En la Figura 13a se muestra la ubicación donde será colocada la minicomputadora en el robot Amikoo. Una vez colocada, la minicomputadora tendrá conectados el arreglo de micrófonos, sistema de audio USB y una tarjeta Wifi. La tarjeta Wifi tendrá conector M.2 y dos antenas (Figura 13b en rojo).

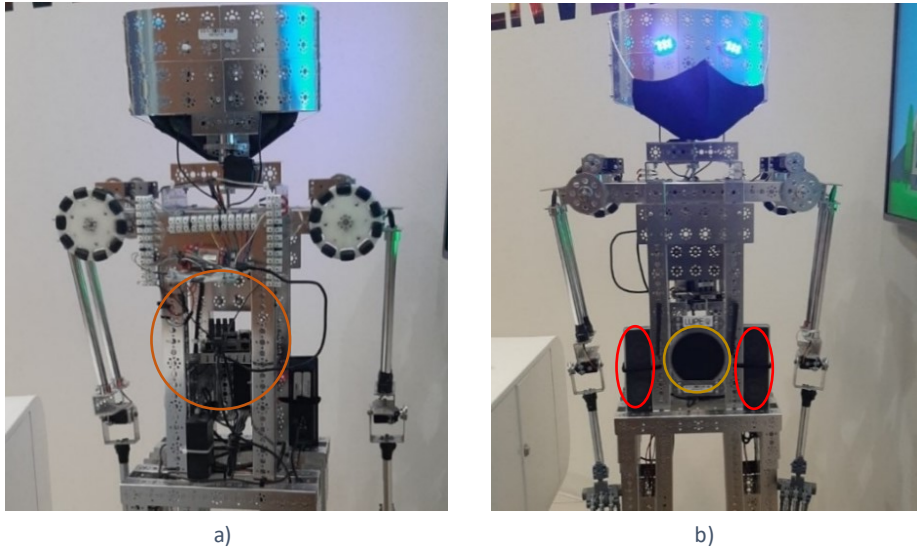


Figura 13: Robot Amikoo con los componentes de hardware.  
a) incluyendo la Upboard Squared, b) incluyendo el arreglo de micrófonos UMA-8

El UMA-8 será colocado fuera del cuerpo metálico y permitiendo tener línea de vista con respecto a los visitantes sentados. Esta línea de vista tiene en cuenta también el mobiliario entre los sillones y el robot.

Colocados los elementos, se realizará la configuración de estos. Si bien, la tarjeta Wifi, el UMA-8 y el sistema de audio USB son detectados por el sistema operativo sin requerir la instalación manual de controladores, se necesita indicar que estos dos últimos componentes serán los predeterminados para grabación y salida de audio respectivamente. Para configurar los dispositivos de entrada y salida como predeterminados se utiliza ALSA, un conjunto de controladores para tarjetas de video y biblioteca para aplicaciones de audio. La manera de configurar y probar la funcionalidad de los dispositivos en ALSA es a través del archivo `.asoundrc`, esto se muestra en la sección 2 de la documentación del UMA-8 (miniDSP, 2020).

Con los componentes de audio configurados, se procede a integrar el componente de software, para realizar la integración final de los elementos de *hardware*.

## 3.6 Integración del software

La integración de los componentes de software es realizada en dos etapas: la relacionada al sistema Amikoo y al sistema Hochob.

### 3.6.1 Integración del software en Amikoo

El software del robot Amikoo estará compuesto por dos contenedores: Rhasspy y HeyAmikoo. El contenedor Rhasspy tiene todos los subsistemas para el procesamiento y generación de voz, es provisto por la documentación y requiere indicar el idioma a utilizar. El contenedor HeyAmikoo será creado a partir del código del sistema Amikoo anteriormente utilizado, con la diferencia que serán removidas funcionalidades no utilizadas.

Para crear la imagen del contenedor HeyAmikoo, es necesario seleccionar una imagen base y generar una lista de todas las dependencias del sistema. La imagen base será Debian Stretch con NodeJS; esto se debe a que la mayor parte del *software* está programado en JavaScript y el sistema operativo utilizado anteriormente era Ubuntu, basado en Debian, esto permite conservar las instrucciones de instalación y el nombre de los paquetes. La lista de dependencias será generada a partir de los 4 *scripts* utilizados para instalar el sistema: instalación del sistema Alexa y Google Cloud, instalación del controlador de la tarjeta de red inalámbrica y las dependencias generales del sistema. De los *scripts* mencionados, serán removidas las dependencias relacionadas a servicios de Nube, ya que estas funcionalidades serán manejadas por Rhasspy. También serán removidas las dependencias relacionadas a la tarjeta de red inalámbrica, debido a que la tarjeta es diferente y no es requerido el controlador.

La lista final de dependencias será utilizada junto con la imagen base para generar el archivo Docker de la imagen HeyAmikoo, como se muestra en la Figura 14.

```

FROM node:14.4.0-stretch-slim

WORKDIR /home/Amikoo

COPY Main .

RUN apt-get update && \
    apt-get install -y --no-install-recommends \
    espeak \
    feh \
    mpv \
    python3 \
    mosquitto-clients \
    build-essential

RUN npm install

CMD node main.js

```

Figura 14: Contenido del archivo Docker de la imagen HeyAmikoo

```

version: '3'
services:
  heyamikoo:
    build:
      context: .
      dockerfile: Dockerfile.heyamikoo
    container_name: heyamikoo
    privileged: true
    network_mode: host
    ports:
      - 1883:1883
    restart: on-failure
    depends_on:
      - rhasspy
  rhasspy:
    image: "rhasspy/rhasspy:2.5.9"
    container_name: rhasspy
    restart: unless-stopped
    volumes:
      - "./rhasspy/profiles:/profiles"
      - "/etc/localtime:/etc/localtime:ro"
    ports:
      - "12101:12101"
    devices:
      - "/dev/snd:/dev/snd"
    command: --user-profiles /profiles --profile es

```

Figura 15: Contenido del archivo Docker Compose del sistema HeyAmikoo

Otro cambio realizado en el sistema es la eliminación de funcionalidades sin utilizar en el código fuente `main.js`, estos elementos están relacionadas con funcionalidades de procesamiento de voz con Alexa y Google Assistant.

Una vez que las imágenes Docker están estructuradas y el código modificado, se genera un archivo Docker Compose para crear los contenedores, como se muestra en la Figura 15. Este archivo contendrá la imagen de Rhasspy obtenida de la documentación y el archivo Docker creado de HeyAmikoo, utilizando el archivo `main.js` de las funcionalidades. Posteriormente, se crean los contenedores y se procede a la configuración de estos.

### 3.6.2 Configuración del contenedor Rhasspy

El primer paso para configurar el contenedor de Rhasspy, es seleccionar las aplicaciones para utilizar en cada servicio. Los servicios y aplicaciones que se utilizaran en este proyecto son los siguientes: grabación de audio, ALSA; detección de frase clave, Raven; voz a texto, Kaldi; reconocimiento de intentos, Fsticuffs; manejador de intentos, Home Assistant; y de texto a voz, espeak; salida de audio, ALSA. Además de estos servicios, es necesario poblar el archivo de entrenamiento con diversas frases, las cuales serán las peticiones que activarán al sistema Amikoo.

La configuración de las aplicaciones seleccionadas anteriormente son las siguientes. En el caso de ALSA para la grabación y salida de audio no es necesario, la configuración se dio en la sección 3.5. Raven necesita grabar 3 veces la frase clave para generar el modelo en el sistema Amikoo, la frase para utilizar en este sistema es “Hey Lupe”; un campo importante en esta aplicación es la sensibilidad, la cual se conservará en el valor predeterminado de 0.5. La aplicación de texto a voz será configurada para utilizar la voz “es-ES” con la finalidad de adecuar la pronunciación de las acciones. Por último, se realiza la configuración del manejador de intentos, sin embargo, esto se realizará más adelante, debido a que es necesario configurar el sistema Hochob.

Adicionalmente, es necesario agregar frases para entrenar el sistema con las peticiones a las que responderá el sistema, estas frases estarán basadas desde la Tabla 6 a la Tabla 9. Las frases de entrenamiento que se utilizaran se muestran en el Apéndice I.

### 3.6.3 Configuración del contenedor Hochob

El repositorio del sistema Hochob presenta diversos errores e incompatibilidades, es necesario repararlo para iniciar la integración de interacciones. Esta reparación se muestra en el Apéndice I.

```
set -x

. /services/main.sh

export CURL_PID=$$
export CURL_BINARY=curl
export AMIKOO_TTS_API_ADDRESS=$AMIKOO_TTS_API_ADDRESS

SPEECH_TEXT="${1}"

if [ $# -eq 1 ]; then

CURL_BINARY -X POST "$AMIKOO_TTS_API_ADDRESS" -H "Content-Type: text/plain" -d "$SPEECH_TEXT"

else

    echo "Invalid number of arguments, see Documentation"
    exit 1

fi
```

Figura 16: Contenido del archivo shell encargado del servicio cURL

Realizadas las reparaciones del repositorio, se agrega un nuevo servicio (cURL) y se actualiza el servicio de voz para que se comuniquen con la interfaz de voz de Rhasspy. El servicio cURL (Figura 16) estará encargado de enviar peticiones HTTP POST a la API de voz de Rhasspy. El servicio requiere el nombre de la aplicación a ejecutar (cURL) y la URL de la API de texto a voz. Esta URL será almacenada en el archivo de variables del sistema Hochob y se ve de la siguiente manera *http://Direccion-IP/api/text-to-speech*. Este servicio cURL será agregado al servicio de voz como una opción cuando el servicio de voz sea invocado con un parámetro *amikoo*.

Después, se agregan las diferentes acciones que tendrá Hochob, de acuerdo con los comandos de voz de Amikoo. Estas acciones se configurarán dentro del contenedor Home Assistant, e internamente son llamadas automatizaciones. Primero, es necesario crear archivos *.yaml* como lo muestra la Figura 17 para crear las automatizaciones. En el campo *alias*, se agregará una descripción de la automatización; en el campo *evento* se agregará *Rhasspy\_NombreDelEventoEnRhasspy*, este nombre es la línea entre corchetes del archivo de entrenamiento, conocido como evento; en el campo *servicio* se indicará que la acción a llevar a cabo es ejecutar un archivo tipo *shell* con diversos comandos. El nombre de este

archivo *shell* requiere ser declarado dentro de la sección de componentes *shell commands* de Home Assistant en formato *.yaml*, de la siguiente manera:

```
pastor: /applications/tacos/pastor.sh.
```

```
alias: "Explicacion Centro de Visitantes"
trigger:
  platform: event
  event_type: rhasspy_CentroDeVisitantes
action:
  service: shell_command.pastor
```

Figura 17: Ejemplo de automatización en Home Assistant

El archivo *shell* donde se realizarán las acciones contiene la frase que será enviada a Amikoo, y los servicios que invocará, teniendo entre las opciones: audio, movimientos de Amikoo, video, voz o comunicarse con otros dispositivos (Juum). En la Figura 18 se muestra un archivo de ejemplo donde se invocan los servicios de video y voz, el video en este caso es presentado en pantalla y la voz es enviada a Amikoo.

Por último, estas automatizaciones deben ser declaradas en la interfaz de usuario. Esto se consigue el archivo *.yaml* dentro de la carpeta *lovelace*, y presenta una estructura como se muestra en la Figura 19. Es necesario indicar la descripción de la automatización y el *script* que será invocado, el cual se declaró anteriormente.

```
#!/bin/sh

. /services/main.sh

export PASTOR_PID=$$

LOCAL_PHRASE="Aquí conocerán el pasado, el presente y el futuro de Intel. \
Al final encontrarán una sección donde podrán jugar y maravillarse \
con la ciencia. Tambien encontrarán un piano cósmico. \
"

/services/video/video.sh "off" "/config/www/media/VisitorsCenter/Video/VisitorsCenter.mp4" &

/services/voice/voice.sh "amikoo" "default" "spanish" "$LOCAL_PHRASE"
```

Figura 18: Archivo shell con las acciones de la automatización

```

- type: entities
  title: Voice Utilities
  show_header_toggle: false
  entities:
    - type: section
      - name: "Centro de Visitantes"
        entity: script.pastor

```

Figura 19: Ejemplo de la declaración de una automatización en la interfaz de Home Assistant

Este proceso de crear automatizaciones debe ser aplicado para cada una de las interacciones desde la Tabla 6 a la Tabla 9. Los contenidos didácticos auxiliares fueron provistos por personal del Centro de Visitantes.

Una vez integradas las diversas interacciones, se procede a realizar la configuración faltante en Rhasspy, la relacionada con el manejador de Intentos.

### 3.6.4 Integración de Rhasspy con Hochob

Para realizar la configuración del manejador de intentos en Rhasspy es necesario agregar la URL donde se ejecuta Home Assistant, el *token* de acceso y el tipo de información que será enviada a Home Assistant, como se muestra en la Figura 20.

Documentation

Hass URL  Address of your Home Assistant server

Access Token  Long-lived access token (automatically filled in Hass.IO)

API Password  Home Assistant password (deprecated)

Send **events** to Home Assistant (/api/events)  
Events will be named `rhasspy_INTENT_NAME`

Send **intents** to Home Assistant (/api/intent/handle)  
Requires the `intent` component and [intent scripts](#) in your `configuration.yaml`

Figura 20: Interfaz del manejador de intentos en Rhasspy

Home Assistant usa el puerto 8123 y la URL que se ingrese debe contenerlo, el *token* de acceso se obtiene en la sección de usuario. Acerca del tipo de información que será enviada a Hochob, serán eventos, estos serán de la forma `Rhasspy_NombreDelEventoEnRhasspy`.

### 3.7 Conclusiones

Como resultado de las acciones realizadas en este capítulo, la interacción de los usuarios del Centro de Visitantes con el sistema Uéej se muestra en la Figura 21. Esta interacción conserva la aplicación móvil, sin embargo en la interfaz de voz puede realizar la invocación de acciones. Como respuesta a las peticiones de los usuarios, se realizan respuestas con voz y en algunos casos con contenido multimedia auxiliar o movimientos. El siguiente paso será realizar diversas pruebas para comprobar que estas interacciones son efectuadas correctamente.

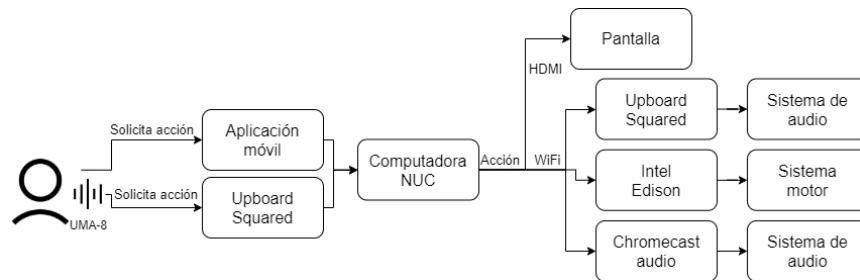


Figura 21: Diagrama de bloques de la interacción del Centro de Visitantes con el proyecto Uéej



## Capítulo 4

### Resultados del desarrollo del sistema Uéej

Se desarrolla la metodología y condiciones para la realización de las pruebas en el sistema Uéej, con la finalidad de comprobar su correcto funcionamiento. Después, serán efectuadas estas pruebas y se valorarán los resultados obtenidos.

## 4.1 Metodología de pruebas

Con la finalidad de comprobar el correcto funcionamiento del sistema Ueej, se realizarán diferentes pruebas en diferentes etapas, abarcando desde nivel componente (hardware y software), hasta sistema completo (integración), considerando el lugar donde será colocado (espaciales). Serán realizadas de acuerdo con la siguiente metodología (mostrada en la Figura 22, cada color representa el sistema del componente).

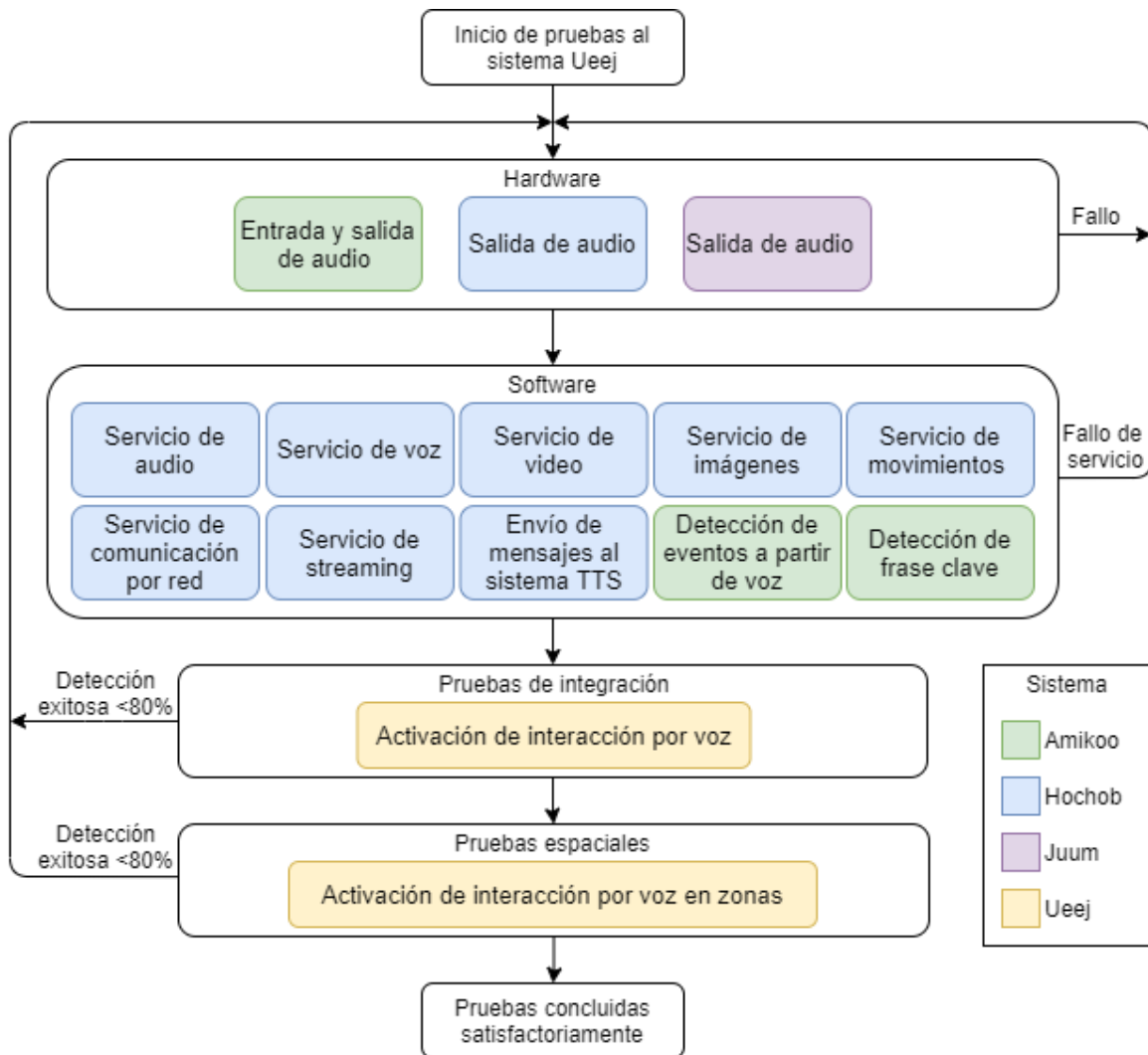


Figura 22: Diagrama metodológico de pruebas

En la primera etapa se realizan pruebas a los componentes de hardware, donde se verificará 10 veces el funcionamiento del sistema de salida y entrada de audio en Amikoo, además de, el sistema de salida de audio en Hochob y el sistema de salida de audio en Juum. En Amikoo y Hochob, se ejecutarán utilidades de ALSA, las cuales permiten hacer funcionar dispositivos de entrada y salida de audio de acuerdo con el archivo de configuración `.asoundrc`. En Juum, al estar compuesto únicamente por un sistema de audio con conector analógico de 3.5 mm,

se conectará un dispositivo compatible para generar audio. Una vez que funcionen correctamente estos elementos, se procederá con la siguiente etapa. En caso contrario se verificará su configuración o se valorará sustituir los componentes.

En la segunda etapa, se realizarán pruebas de los componentes de software, donde se ejecutarán 10 veces los servicios de Hochob: audio, *streaming*, imágenes, comunicación por red, video, voz y movimientos de Amikoo; ejecutando sus respectivos comandos. Posteriormente, serán realizadas pruebas en los componentes encargados de la detección de frase clave y eventos, invocando en 10 ocasiones el sistema con la frase clave “Hey Lupe” y los eventos, cada evento será invocado con una frase del conjunto de entrenamiento, debido a que en el conjunto existen diversas combinaciones de frases y las pruebas podrían incrementar considerablemente en iteraciones posteriores, las frases se obtienen de la Tabla 6 a la Tabla 9. Si el porcentaje de detecciones correctas en los módulos es mayor al 80%, se procederá a realizar la siguiente etapa de pruebas. En caso contrario, serán modificados los archivos de configuración del servicio, el factor de sensibilidad de la frase clave o el conjunto de entrenamiento según sea el caso; si continúa existiendo un porcentaje menor al establecido, se verificarán los elementos de hardware y se repetirán los procedimientos de la segunda etapa.

Como tercera etapa, se harán las pruebas de integración para revisar el comportamiento del sistema en el flujo normal de operación por los guías y visitantes. Este flujo comprende el reconocimiento de voz, la detección de eventos y la realización de acciones. Para comprobar su funcionamiento se invocarán en 10 ocasiones la frase clave y cada evento, los eventos serán invocados con las mismas frases utilizadas en las pruebas de la segunda etapa. Si el porcentaje de detecciones exitosas en ambos casos supera el 80% y se realizan las acciones correspondientes, las pruebas se consideran exitosas y se procede con la siguiente etapa. Si la detección de palabra clave genera un porcentaje menor al porcentaje estipulado, se modificará el valor de sensibilidad y se realizarán las pruebas de las etapas anteriores verificarán, para finalmente repetir las invocaciones de este componente. En caso de presentarse un porcentaje menor en la detección de eventos y realización de acciones, se modificará el conjunto de entrenamiento de la Sección 0, se repetirán los procedimientos de las etapas anteriores y repetirán las invocaciones.

La última etapa de pruebas es verificar el comportamiento del sistema completo en el Centro de Visitantes. Para conocer este comportamiento se invocarán los sistemas de detección de frase clave y de eventos en el espacio destinado para el robot, durante los horarios en que se efectúan las visitas, con la finalidad de conseguir niveles de ruido habituales. Debido a la situación de la pandemia y normativa de Intel, no se realizaron invocaciones con un grupo de personas, sin embargo, se segmentó el vestíbulo para conocer el comportamiento del sistema desde diferentes zonas. En las zonas se invocarán los sistemas en 10 ocasiones. Si el porcentaje de detecciones exitosas de frase clave y eventos supera el 80% y se realizan las acciones, se consideran las pruebas exitosas y el desarrollo

del sistema Uéej por concluido. En caso de presentar un porcentaje de detección menor al 80% en la frase clave, será modificado el valor de sensibilidad y se repetirán los procedimientos de la última etapa. Si el sistema de detección de eventos presenta un porcentaje menor al 80%, será modificado el conjunto de entrenamiento, y se repetirán los procedimientos de la última etapa. En caso de persistir el porcentaje menor al establecido, se efectuarán los procedimientos de las etapas anteriores.

## 4.2 Pruebas del sistema Uéej

A partir de la metodología presentada, se establecerán las acciones que se realizarán en cada etapa de pruebas para evaluar el funcionamiento de los componentes.

### 4.2.1 Primera etapa de pruebas: hardware

En la primera etapa, se realizarán pruebas a los sistemas de audio en 10 ocasiones. Los sistemas de audio están formados por el sistema de entrada y salida de audio en Amikoo, el sistema de salida de audio en Hochob y el sistema de salida de audio en Juum.

Para verificar el funcionamiento de los sistemas de Amikoo, se ejecutarán los comandos `arecord -l` y `alist -l` para mostrar los dispositivos de entrada y salida de audio detectados por el sistema operativo con sus identificadores, en estas listas se verificará que se muestren los dispositivos USB Speaker y UMA-8, correspondientes al sistema de audio USB y el arreglo de micrófonos UMA-8. Después se ejecutará `arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw out.raw` para grabar un archivo de audio con una duración de 5 segundos, posteriormente, este archivo de audio será reproducido con la instrucción `aplay --format=S16_LE --rate=16000 out.raw`. Este procedimiento se realizará en 10 ocasiones. En caso de que no se produzca audio, se ejecutará el comando `speaker-test -c 2` para generar ruido blanco en el dispositivo de salida de audio indicado en `.asoundrc`. Si este comando genera audio, es necesario verificar el identificador del dispositivo de entrada de audio en `.asoundrc` y repetir el proceso; en caso de no existir salida de audio, es necesario modificar el archivo `.asoundrc` de acuerdo con los identificadores de los dispositivos de salida de audio y repetir los comandos `arecord` y `aplay`. Si la ejecución de los comandos resultó exitosa, se procede a verificar el hardware del dispositivo Hochob.

Para comprobar la salida de audio en Hochob, se crea una interfaz de terminal dentro del contenedor de Home Assistant con el comando `docker exec -ti hochob-homeassistant bash`, y se ejecuta el comando `speaker-test -c -2` 10 veces, en caso no escuchar una respuesta audible, es necesario verificar los identificadores de dispositivos de salida en la configuración del archivo `.asoundrc` y repetir el comando `speaker-test -c -2`. Si existió una respuesta audible, se da la prueba como exitosa y se procede a realizar pruebas en Juum.

La salida de audio en Juum se verificará 10 veces reproduciendo audio en un dispositivo que tenga salida de audio con conector audio analógico de 3.5 milímetros y conectarlo al

sistema de Juum, si se reproduce el audio en el sistema, la prueba se da por exitosa; en caso contrario el sistema de audio deberá ser sustituido.

Una vez que los procedimientos anteriores se ejecuten satisfactoriamente, se podrá realizar la siguiente etapa de pruebas.

#### 4.2.2 Segunda etapa de pruebas: software

La segunda etapa de pruebas se realizará en el software de Hochob y Amikoo. En Hochob se revisará el funcionamiento de los servicios de audio, voz, video, imágenes, acciones motrices de Amikoo, comunicación por red, *streaming*; mientras que en Amikoo se verificarán los sistemas de detección de eventos y frase clave.

Para realizar revisar el funcionamiento de los servicios en Hochob, es necesario ingresar al contenedor Home Assistant con el comando `docker exec -ti hochob-homeassistant bash`. Después, se ejecutarán 10 veces las instrucciones de la Tabla 10, donde cada instrucción, refiere a un servicio y se espera como resultado de la ejecución una acción. Los eventos se muestran en la primera columna de la Tabla 10, mientras que las instrucciones y respuestas esperadas están en la segunda y tercera columna respectivamente.

Tabla 10: Pruebas realizadas a los servicios audiovisuales de Hochob y sus respuestas esperadas

Pruebas en los servicios audiovisuales de Hochob		
Servicio	Comandos que se ejecutaran	Respuesta esperada
Audio	<code>/services/audio/audio.sh \</code> <code>'/config/www/media/Amikoo/Sound/TakePictureEllie.mp3'</code>	Sonido reproducido en Hochob
Streaming	<code>/services/streaming/chromecaststream.sh 'Juum' '1.0' \</code> <code>'/config/www/media/Amikoo/Sound/TakePictureEllie.mp3'</code>	Sonido reproducido en Juum
Imágenes	<code>/services/image/feh.sh \</code> <code>'/config/www/media/HeyAmikoo/Image/PianoCosmico'</code>	Imagen desplegada en la pantalla
Protocolos por red (MQTT)	<code>/services/protocol/mosquitto_pub.sh test 'Hola Mundo'</code>	Mensaje Hola Mundo en los clientes suscritos
Protocolos por red (cURL)	<code>/services/protocol/curl.sh 'Hola Mundo'</code>	Voz con el mensaje Hola mundo en Amikoo
Video	<code>/services/video/video.sh 'off' \</code> <code>'/config/www/media/HeyAmikoo/Video/Muones.mp4'</code>	Video reproducido en Hochob
Voz (Juum)	<code>/services/voice/voice.sh 'Juum' 'espeak' 'spanish' 'Hola Mundo'</code>	Voz con el mensaje hola mundo en Juum
Voz (Amikoo)	<code>/services/voice/voice.sh 'amikoo' 'default' 'spanish' 'Hola Mundo'</code>	Voz con el mensaje hola mundo en Hochob

Acciones motrices	/services/amikoo/amikoomovement.sh 'lupe' DESCRIPCION_DEL_MOVIMIENTO	MOVIMIENTO	Movimiento del robot Amikoo
-------------------	--	------------	-----------------------------

Algunos de los servicios, a su vez, realizan invocaciones a otros, por ejemplo el servicio de voz invocando al de *streaming* y audio, o el servicio de movimiento al de protocolos de red. En el caso del servicio de acciones motrices de Amikoo, requiere diversos parámetros en las instrucciones, los cuales se muestran en la Tabla 11 y serán ejecutados en 10 ocasiones. En la primera columna se muestra el movimiento esperado; en la segunda y tercera columna se encuentran los diferentes valores de los campos MOVIMIENTO y DESCRIPCION\_DEL\_MOVIMIENTO.

Tabla 11: Parámetros de las pruebas del servicio de movimiento

Movimiento	Parámetro	
	Movimiento	Descripción
Girar cabeza a la derecha	headright	Lupe Head Right
Girar cabeza a la izquierda	headleft	Lupe Head Left
Alzar brazo derecho	rightup	Lupe Arm Right up
Doblar brazo derecho	rightfold	Lupe Arm Right Fold
Desdoblar brazo derecho	rightunfold	Lupe Arm Right Unfold
Bajar brazo derecho	rightdown	Lupe Arm Right Down
Alzar brazo izquierdo	leftup	Lupe Arm Left up
Doblar brazo izquierdo	leftfold	Lupe Arm Left Fold
Desdoblar brazo izquierdo	leftunfold	Lupe Arm Left Unfold
Bajar brazo izquierdo	leftdown	Lupe Arm Left Down
Desplazar a la izquierda	moveleft	Lupe Legs Move Left
Desplazar a la derecha	moveright	Lupe Legs Move Right
Desplazar hacia adelante	moveforward	Lupe Legs Move Forward
Desplazar hacia atrás	movebackward	Lupe Legs Move Backward

Se probará el funcionamiento de los sistemas de detección de frase clave y eventos invocándolos 10 veces, haciendo uso de la frase “Hey Lupe” y las frases del Apéndice I respectivamente.

Estas pruebas se realizarán en la interfaz web de Rhasspy (mostrada en la Figura 23), a la cual se accede con un navegador web ingresando la dirección IP del sistema Amikoo y el puerto 12101. La interfaz permite realizar la detección de eventos y de frase clave; cuando un evento es detectado se muestra de color rojo, si es detectada la frase clave se genera una respuesta sonora.

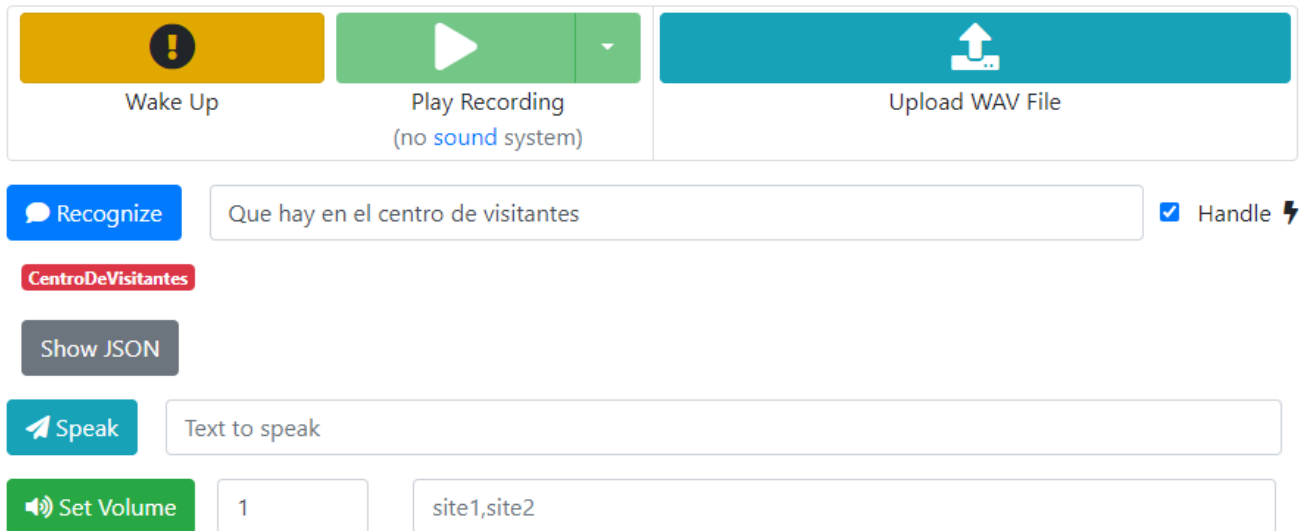


Figura 23: Interfaz de Rhasspy con una petición detectada

Si todos los servicios de Hochob y de detección de voz (frase clave y eventos) se ejecutan correctamente en al menos el 80% de las pruebas, se continuará a la siguiente etapa de pruebas. En caso contrario, serán revisados los archivos de configuración en el caso de Hochob, se actualizará el valor de sensibilidad en detección de frase clave o se modificará el conjunto de entrenamiento en detección de eventos, para posteriormente repetir las pruebas de la etapa actual. En caso de continuar presentándose porcentajes menores al estipulado, se harán los procedimientos de la etapa anterior.

#### 4.2.3 Tercera etapa de pruebas: integración

La tercera etapa de pruebas consiste en comprobar el estado de la integración de los componentes, revisando el flujo completo de las interacciones en Uéej, desde la invocación por voz hasta la ejecución de las respectivas acciones; por este motivo es requerido que las pruebas de hardware y software sean exitosas.

Se realizarán 10 invocaciones de la frase “Hey Lupe” y un subconjunto de las peticiones de el Apéndice II (Tabla 12), esperando como resultado las acciones del evento invocado (mostradas desde la Tabla 6 a la Tabla 9). Si las detecciones exitosas de los dos grupos de invocaciones representan al menos el 80% del total de detecciones y las acciones son llevadas a cabo correctamente, se procederá con la siguiente etapa de pruebas. En caso de presentarse un porcentaje mejor en la detección de frase clave, se modificará el valor de sensibilidad y se repetirá esta etapa de pruebas. Si el porcentaje de la detección de eventos es menor al establecido habrá modificaciones al conjunto de entrenamiento y se repetirá

esta etapa de pruebas. Si después de estas modificaciones sigue presentándose un porcentaje menor al establecido, se realizarán las pruebas de las etapas anteriores.

Tabla 12: Tabla con eventos y ejemplos de peticiones del sistema Uéej

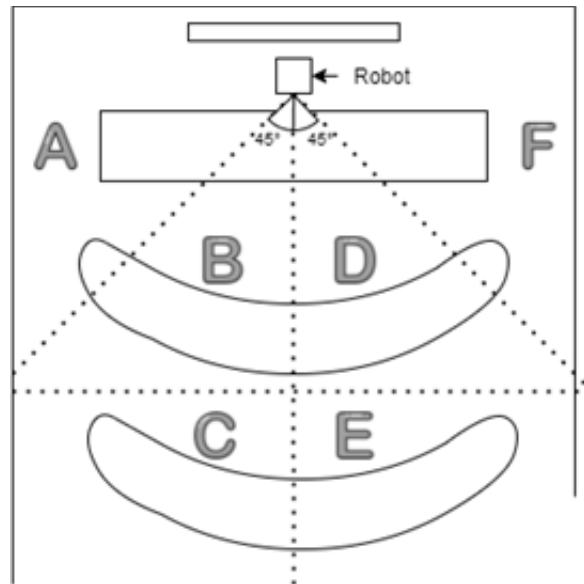
Tabla con eventos y ejemplos de peticiones del sistema Uéej		
ID	Evento	Peticiones
1	Intel	¿A qué se dedica Intel?
2	GDC	¿Qué se hace en GDC?
3	Microprocesador	¿Sabes qué es un microprocesador?
4	CentroDeVisitantes	¿Qué hay en el Centro de Visitantes?
5	PianoCosmico	¿Qué es un Piano cósmico?
6	Muon	¿Qué es un muon?
7	Sustentabilidad	Qué cosas hacemos para cuidar el medio ambiente
8	IA	Explica que es la inteligencia artificial
9	IAUso	Para que se utiliza la inteligencia artificial
10	Saludo	Saluda a los visitantes
11	Presentacion	Preséntate
12	Cafe	Prepara café
13	Creador	Quien es tu creador?
14	TiempoLibre	Que te gusta hacer en tu tiempo libre
15	Chiste	Cuenta un chiste
16	Despedida	Nos tenemos que ir
17	Tour	Inicia el recorrido

En la Tabla 12 se muestra un subconjunto de las peticiones del conjunto de entrenamiento del Apéndice I, donde la primera columna representa el identificador del evento, en la segunda columna el nombre del evento y en la última las peticiones.

#### 4.2.4 Etapa final de pruebas: comportamiento espacial

En la etapa final de pruebas se invocarán los sistemas de voz de Uéej en el Centro de Visitantes, para conocer el comportamiento de la detección de voz en condiciones similares a las encontradas durante las visitas. Para generar estas condiciones las pruebas se realizarán en un horario similar al utilizado en las visitas (de 11 de la mañana a 4 de la tarde), ya que es común la presencia de ruidos externos. Además, durante la planeación de etapa se consideraba tener un grupo de personas para utilizar el sistema, sin embargo, no fue posible, debido a las medidas tomadas por Intel con respecto a la pandemia de SARS-COV2, donde no se permiten aglomeraciones de personas. En sustitución del grupo de personas, se segmentará el área del vestíbulo en 6 zonas. Las diferentes zonas se muestran en la Figura 24, representan las posibles ubicaciones de los visitantes y guías; encontrándose a las

siguientes distancias del arreglo de micrófonos instalado en el robot: A y F a 1.40m, B y D a 2.30m, C y E a 3.20m.



*Figura 24: Diagrama de los puntos seleccionados en el vestíbulo del Centro de Visitantes donde se realizarán las pruebas de la detección de audio*

Las pruebas invocarán en 10 ocasiones los sistemas de detección de frase clave y eventos desde distintas zonas, esperando como resultado las acciones correspondientes.

El primer sistema se invocará con la frase “Hey Lupe” desde las zonas B y D. Estas zonas fueron elegidas debido a que son las más comunes donde el guía del Centro de Visitantes se encuentra en los recorridos. A partir de los resultados obtenidos se calculará el porcentaje de detecciones exitosas y si es igual o mayor al 80% se probará el siguiente sistema. En cambio si el porcentaje es menor, se modificará el valor de sensibilidad y repetirán estas invocaciones. Si el porcentaje continúa presentándose, se harán las verificaciones de las etapas anteriores.

Al segundo sistema se realizarán 10 repeticiones de la invocación de eventos desde todas las zonas y utilizando las frases de la Tabla 12; esperando como resultado las acciones indicadas desde la Tabla 6 a la Tabla 9. A partir de los resultados obtenidos se calculará el porcentaje de detecciones exitosas para cada evento, y si es igual o mayor al 80% en todos los eventos, se darán por concluidas las pruebas de esta etapa. En caso contrario, se modificarán las frases del conjunto de entrenamiento en los eventos afectados y se harán estas repeticiones nuevamente. Si el problema continúa presentándose o es relacionado a las acciones, se realizarán las pruebas de las etapas anteriores.

### 4.3 Resultados de las pruebas realizadas

Los resultados de las diferentes etapas de pruebas se presentan a continuación.

#### 4.3.1 Resultados de las pruebas de hardware

Después de realizar las pruebas de hardware correspondientes a la primera etapa, indicadas en la sección 4.2.1, se obtuvieron los siguientes resultados (mostrados en la Tabla 13). Las instrucciones para comprobar el funcionamiento de los sistemas de entrada y salida de audio en Amikoo fueron ejecutadas exitosamente en la totalidad de los casos, lo que indica su correcta configuración y da certeza en su funcionamiento, reduciendo la posibilidad de errores en los módulos que hagan uso de estos sistemas. Con respecto a la salida de audio en Hochob, existió salida de audio en el contenedor en las 10 ejecuciones, significando que puede ser usado correctamente por módulos del sistema y que la posibilidad de presentar problemas en las etapas de pruebas posteriores se reduce. En el dispositivo de audio de Juum existió audio en los 10 intentos, indicando que el sistema de audio funciona correctamente y la posibilidad de existir errores físicos en Juum es menor.

*Tabla 13: Resultados de las pruebas a nivel hardware*

<b>Resultados de las pruebas de hardware</b>		
<b>Sistema</b>	<b>Prueba</b>	<b>Resultados exitosos</b>
Amikoo	Entrada de audio	10
Amikoo	Salida de audio	10
Hochob	Salida de audio	10
Juum	Salida de audio	10

Al obtener 10 casos exitosos de 10 totales, como lo muestra la última columna de la Tabla 13, se espera un funcionamiento correcto en los módulos y sistemas que hagan uso de los componentes de hardware, reduciendo las probables causas de fallos en futuras etapas de pruebas, por este motivo, puede darse la etapa por concluida y se avanza a la siguiente.

#### 4.3.2 Resultados de las pruebas de software

Los resultados de las pruebas en la etapa de software se presentan en dos secciones, la primera está relacionada a los servicios de audio, streaming, imágenes, comunicación por red, video, voz y acciones motrices en Hochob; mientras que la segunda sección presenta los servicios relacionados a detección de voz.

La primera sección de resultados se obtuvo a partir de la ejecución en 10 ocasiones de los comandos presentados en la Tabla 10 y Tabla 11. Los comandos de la Tabla 10 están relacionados a los servicios de Hochob, con excepción del encargado de acciones motrices, el cual se muestra en la Tabla 11. Los resultados del primer grupo de comandos se muestran en la Tabla 14, donde la primera columna es el nombre de los servicios, la segunda columna la respuesta esperada y la tercera el porcentaje de número de pruebas con resultados exitosos. Los resultados de las pruebas relacionadas con las acciones motrices se muestran en la Tabla 15, donde las columnas nombradas movimiento, representan la acción motriz esperada y las columnas nombradas resultados exitosos, el porcentaje de ejecuciones con resultados exitosos.

*Tabla 14: Resultados de las pruebas a nivel sistema del software: servicios audiovisuales*

<b>Resultados de las pruebas a nivel sistema del software: servicios audiovisuales</b>		
<b>Servicio</b>	<b>Respuesta esperada</b>	<b>Resultados exitosos (%)</b>
Audio	Sonido reproducido en Hochob	100%
Streaming	Sonido reproducido en Juum	100%
Imágenes	Imagen desplegada en la pantalla	100%
Protocolos por red (MQTT)	Mensaje "Hola Mundo" en los clientes suscritos	100%
Protocolos por red (cURL)	Voz con el mensaje Hola mundo en Amikoo	100%
Video	Video reproducido en Hochob	100%
Voz (Juum)	Voz con el mensaje hola mundo en Juum	100%
Voz (Amikoo)	Voz con el mensaje hola mundo en Hochob	100%
Acciones motrices	Movimiento del robot Amikoo	Tabla 15

Los resultados de la Tabla 14 muestran que los comandos relacionados a los servicios de Hochob se ejecutaron correctamente en la totalidad de repeticiones, significando, que los servicios de audio, imágenes, video y voz funcionan correctamente después de las reparaciones efectuadas en la sección 0 y pueden ser utilizados por las aplicaciones del sistema Uéej. Además, los resultados indican que los servicios integrados en el desarrollo (streaming, cURL y voz en Amikoo) funcionan adecuadamente, permitiendo ser utilizados por las aplicaciones ofreciendo una buena experiencia en los visitantes.

Los resultados de la Tabla 15 muestran que las 10 ejecuciones de los comandos relacionados a las acciones motoras no presentaron errores. Significando, que el sistema puede hacer uso de todos los movimientos en las aplicaciones, esperando errores mínimos en su ejecución y por consiguiente, ofrecer una buena experiencia a los visitantes.

Tabla 15: Resultados de las pruebas a nivel sistema del software: servicios motrices

**Resultados de las pruebas a nivel sistema del software: acciones motrices**

Movimiento	Resultados exitosos (%)	Movimiento	Resultados exitosos (%)
Girar cabeza a la derecha	100%	Doblar brazo izquierdo	100%
Girar cabeza a la izquierda	100%	Desdoblar brazo izquierdo	100%
Alzar brazo derecho	100%	Bajar brazo izquierdo	100%
Doblar brazo derecho	100%	Desplazar a la izquierda	100%
Desdoblar brazo derecho	100%	Desplazar a la derecha	100%
Bajar brazo derecho	100%	Desplazar hacia adelante	100%
Alzar brazo izquierdo	100%	Desplazar hacia atrás	100%

La segunda sección de resultados está relacionada con la detección de voz, se obtuvieron a partir de la invocación en 10 ocasiones de la frase clave “Hey Lupe” y de un subconjunto de frases de la sección 6.2 (Tabla 12). Los resultados de las pruebas de detección de frase clave se muestran en la Tabla 16, donde se presenta el porcentaje de detecciones exitosas, el cual será comparado con el establecido en la metodología como referencia de una prueba exitosa. Los resultados de las pruebas de detección de eventos se muestran en la Tabla 17, donde se presentan los porcentajes de detecciones exitosas. Estos porcentajes serán comparados con el establecido en la metodología como referencia de una prueba exitosa.

Tabla 16: Resultados de la detección de frase clave en las pruebas de software

**Resultados de la detección de frase clave en las pruebas de integración**

Invocación	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Hey Lupe	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	90%

De acuerdo con los resultados mostrados en la Tabla 16, el porcentaje de detecciones exitosas de la frase clave supera el 80% establecido en la metodología. A partir de estos resultados se puede decir este subsistema efectúa detecciones correctamente con el arreglo de micrófonos UMA-8. Además, al no presentar errores en las pruebas, se espera que el sistema genere una buena interacción con los usuarios y que la detección de frase clave continúe funcionando adecuadamente en las etapas posteriores de pruebas, aunque en las condiciones acústicas habituales se estima decrecerá, por este motivo es necesario probar el comportamiento del sistema en las condiciones acústicas habituales del Centro de Visitantes.

Tabla 17: Resultados de la detección de eventos en las pruebas de software

<b>Resultados de la detección de eventos en las pruebas de integración</b>			
<b>Evento</b>	<b>Detecciones exitosas (%)</b>	<b>Evento</b>	<b>Detecciones exitosas (%)</b>
<b>Intel</b>	100%	<b>Saludo</b>	100%
<b>GDC</b>	100%	<b>Presentacion</b>	100%
<b>Microprocesador</b>	100%	<b>Cafe</b>	100%
<b>CentroDeVisitantes</b>	100%	<b>Creador</b>	100%
<b>PianoCosmico</b>	100%	<b>TiempoLibre</b>	100%
<b>Muon</b>	100%	<b>Chiste</b>	100%
<b>Sustentabilidad</b>	100%	<b>Despedida</b>	100%
<b>IA</b>	100%	<b>Tour</b>	100%
<b>IAUso</b>	100%		

Como se muestra en la Tabla 17, se obtuvo que la detección de eventos fue correcta en las pruebas realizadas a los eventos. Con esto se puede decir, que el sistema es funcional con el arreglo de micrófonos UMA-8 en condiciones ideales y que el conjunto de entrenamiento es adecuado para la detección. Además, que exista un porcentaje tan alto en la detección de eventos indica que en pruebas posteriores es posible mantener los resultados por arriba del 80%, aunque es probable que los porcentajes decrezcan. Manteniendo las pruebas por encima del 80% da cierta confianza de brindar a los visitantes una buena experiencia de usuario con las interacciones. Sin embargo, es necesario realizar pruebas de las detecciones en conjunto con las acciones y probar las detecciones en condiciones diferentes a las ideales con la finalidad de asegurarse de mantener una buena experiencia de usuario.

En todas las pruebas realizadas en la segunda etapa se presentaron resultados superiores al 80% establecido, por lo que se dan por concluidas de manera exitosa y es posible proseguir con la siguiente etapa de pruebas.

#### 4.3.3 Resultados de las pruebas de integración

Al realizar los procedimientos de la tercera etapa de pruebas con sus respectivas repeticiones, se obtuvieron los resultados mostrados en la Tabla 18 y Los resultados de ambas pruebas porcentajes por encima del 80%, indicando que no se requiere tomar acciones adicionales en esta etapa y se proceden a realizar las pruebas espaciales.

Tabla 19. En la Tabla 18 se presentan el resultado de la invocación de la frase “Hey Lupe” 10 veces, donde las columnas representan cada una de las pruebas, en las detecciones exitosas se coloca el símbolo ✓ y en caso contrario con el símbolo ✗; al terminar los

procedimientos, será obtenido el porcentaje de detecciones exitosas, el cual será comparado con el establecido en la metodología para definir si la prueba es satisfactoria. Los resultados de ambas pruebas porcentajes por encima del 80%, indicando que no se requiere tomar acciones adicionales en esta etapa y se proceden a realizar las pruebas espaciales.

Tabla 19 contiene los resultados de las invocaciones en 10 ocasiones de un subconjunto de frases de la sección 0 (Tabla 12). Los resultados se presentan con el símbolo ✓ si la prueba fue correcta, significando que la detección del evento fue exitosa y que las acciones esperadas se realizaron (de la Tabla 6 a la Tabla 9). En caso de realizarse la detección de otro evento, se representará la prueba con el número identificador del evento (de acuerdo con la Tabla 12). Por último, se colocará el símbolo ✗ cuando el evento no fue reconocido o la acción no fue realizada. De estas pruebas, se calculará el porcentaje de detecciones exitosas para ser comparado con el valor establecido en la metodología. Si ambos porcentajes superan el 80% establecido, se procederá a realizar la siguiente etapa de pruebas.

*Tabla 18: Resultados de la detección de frase clave en las pruebas de integración*

**Resultados de la detección de frase clave en las pruebas de integración**

Invocación	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Hey Lupe	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	80%

A partir de los resultados mostrados en la Tabla 18, se puede observar una disminución en la detección exitosa con respecto a la Tabla 16, a pesar de estar realizados en las mismas condiciones ideales. Al igualar el 80% establecido en la metodología, se puede decir que el sistema de detección de frase clave funciona en condiciones ideales y que es necesario hacer pruebas en las condiciones del Centro de Visitantes para tener una mejor certeza en la correcta interacción de los usuarios con el sistema Uéej. Continuando con la dinámica de resultados, en la siguiente etapa de pruebas es esperado un porcentaje de detecciones exitosas menor la mostrado en la metodología, causando que sean necesarias acciones correctivas para asegurar un buen funcionamiento.

Los resultados de la Los resultados de ambas pruebas porcentajes por encima del 80%, indicando que no se requiere tomar acciones adicionales en esta etapa y se proceden a realizar las pruebas espaciales.

Tabla 19 muestran el comportamiento del sistema de detección de eventos y las acciones realizadas. Estos resultados son similares a los de la Tabla 17, superando el 80% establecido en la metodología en todos los eventos, considerándose un buen resultado, ya que en las mismas condiciones acústicas, se efectuaron las acciones en la totalidad de ocasiones. Además, estos resultados indican, que las acciones (audiovisuales y motrices) están correctamente configuradas, aportando a una buena interacción y experiencia de los visitantes.

Los resultados de ambas pruebas porcentajes por encima del 80%, indicando que no se requiere tomar acciones adicionales en esta etapa y se proceden a realizar las pruebas espaciales.

*Tabla 19: Resultados de la detección de eventos en las pruebas de integración*

<b>Resultados de la detección de eventos en las pruebas de integración</b>											
<b>Evento</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	<b>P6</b>	<b>P7</b>	<b>P8</b>	<b>P9</b>	<b>P10</b>	<b>Detecciones exitosas (%)</b>
Intel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
GDC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Microprocesador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
CentroDeVisitantes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
PianoCosmico	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Muon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Sustentabilidad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IAUso	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Saludo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Presentacion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Cafe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Creador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
TiempoLibre	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Chiste	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Despedida	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Tour	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%

#### 4.3.4 Resultados de las pruebas espaciales

La última etapa de pruebas consiste en realizar repeticiones de invocaciones de voz en las instalaciones del Centro de Visitantes, en condiciones similares a las habituales y desde

distintas zonas. Los resultados de estas pruebas se presentan en dos secciones, la relacionada con el sistema de detección de frase clave y el de eventos.

La primera sección de resultados (Tabla 20) se obtuvo invocando en 10 ocasiones la frase clave desde las zonas B y D. En estas pruebas, se observa que el porcentaje de detección exitosa está por debajo del establecido, por lo que es necesario implementar acciones correctivas. Estas acciones consisten en iterar diversos valores de sensibilidad, para encontrar mejores resultados, por defecto el sistema usa 0.5.

Tabla 20: Resultados de las pruebas de detección de frase clave con sensibilidad 0.5

Resultados de las pruebas de detección de frase clave con 0.5											
Zona	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
B	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	60%
D	✗	✓	✓	✗	✗	✓	✓	✗	✓	✓	60%
<b>PROMEDIO</b>											<b>60%</b>

Durante las iteraciones, se encontró que valores altos en la sensibilidad provocaban un gran número de falsos positivos en la activación de este subsistema. Considerando esto, se encontró que el valor de sensibilidad que iguala el porcentaje establecido es 0.559. Los resultados de las pruebas de detección con este nuevo valor se muestran en la Tabla 21.

Tabla 21: Resultados de las pruebas de detección de frase clave con sensibilidad 0.559

Resultados de las pruebas de detección de frase clave con 0.559											
Zona	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
B	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	80%
D	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	80%
<b>PROMEDIO</b>											<b>80%</b>

Con los resultados de la Tabla 21, se puede decir, que si bien, se cumple el porcentaje mínimo de detecciones exitosas para una buena experiencia de los usuarios, es necesario considerar tomar acciones posteriormente para tener un sistema de detección más robusta.

La segunda sección de resultados se muestra desde la Tabla 22 a la Tabla 27, se obtuvieron invocando los 17 eventos, tomando como base las frases mostradas en la Tabla 12. Las tablas representan las diferentes zonas del Centro de Visitantes (A hasta F). De estas tablas, la primera columna refiere el evento a invocar, las siguientes columnas las pruebas realizadas, y la última columna el porcentaje de detecciones exitosas. Los resultados de estas pruebas se representan con el símbolo ✓ cuando la detección y las acciones son

correctas, en caso de invocar un evento equivocado se coloca su identificador (tomando como referencia la Tabla 12) y si el sistema no detecta el evento será representado con el símbolo ✕.

En la Tabla 22 se muestran los resultados obtenidos en la zona A del Centro de Visitantes. El porcentaje de detecciones exitosas supera el 80% en la mayoría de los eventos, generando un promedio de las detecciones cercano al 95%. Estos valores indican que en la zona A, el sistema realiza una detección adecuada, con una cantidad reducida de errores, lo que favorece a una mejor interacción de los usuarios al utilizar el sistema. Un punto para considerar es la cantidad de falsos positivos de otros eventos, situación que podría suceder en mayor medida en un ambiente con mucho ruido, sin embargo, no se tomarán acciones al respecto.

Tabla 22: Resultados de la detección de eventos en la zona A

Resultados de la detección de eventos en la zona A											
Evento	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Intel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
GDC	✓	✓	✓	✓	✓	4	✓	✓	✓	✓	90%
Microprocesador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
CentroDeVisitantes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
PianoCosmico	12	✓	✓	✓	✓	✓	✓	✓	✓	✓	90%
Muon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Sustentabilidad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IAUso	✓	✓	✓	✓	✓	✓	✓	✓	✓	8	90%
Saludo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Presentacion	✓	✓	✓	✓	✓	✓	✓	6	✓	✓	90%
Cafe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Creador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
TiempoLibre	✓	✓	✓	✓	4	✓	✓	✓	✓	4	80%
Chiste	✓	✓	✓	✓	✓	✓	✓	✓	✓	11	90%
Despedida	✓	✓	✓	✓	✓	✓	✓	✓	✓	11	90%
Tour	✓	✓	✓	✓	✓	✓	4	✓	✓	✓	90%
<b>PROMEDIO</b>											<b>94.71%</b>

En la Tabla 23, se muestran los resultados obtenidos de las invocaciones en la zona B del Centro de Visitantes. El porcentaje de detecciones exitosas supera el 80% en la mayoría de los eventos y con un promedio de detección superior al 97%. Este comportamiento indica

que la zona B es más favorable para la detección que la A, con una cantidad menor de errores y de falsos positivos, convirtiéndola en idea para usarse por los usuarios. Aunque cabe mencionar la existencia de casos falsos negativos, especialmente en el evento GDC. Sin embargo, al existir no existir porcentajes de detección menores al 80%, no se realizarán acciones adicionales.

Tabla 23: Resultados de la detección de eventos en la zona B

Resultados de la detección de eventos en las pruebas de integración											
Evento	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Intel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
GDC	✓	✓	✓	✓	X	✓	✓	✓	X	✓	80%
Microprocesador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
CentroDeVisitantes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
PianoCosmico	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Muon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Sustentabilidad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IAUso	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	90%
Saludo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Presentacion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Cafe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Creador	✓	✓	✓	16	✓	✓	✓	✓	✓	✓	90%
TiempoLibre	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Chiste	✓	✓	✓	✓	✓	✓	✓	✓	✓	11	90%
Despedida	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Tour	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
<b>PROMEDIO</b>											<b>97.06%</b>

En la Tabla 24 se muestran los resultados obtenidos de las invocaciones en la zona C del Centro de Visitantes. El porcentaje de detecciones exitosas en los eventos supera el 80% en la mayoría, sin embargo, en esta zona se presentan varios eventos con porcentajes de 80%, el relacionado al Muon, IAUso y Despedida. Estos valores indican que la detección en esta zona no es tan precisa como en las anteriores, pero al tener un porcentaje promedio de detecciones exitosas de casi 92%, se puede decir que la interacción generada es aceptable. Además, en este conjunto de pruebas, se observa una cantidad similar de falsos positivos y falsos negativos, situación que podría causar problemas en la experiencia de usuario en situaciones de mucho ruido ambiental. Al existir no existir porcentajes de detecciones menores al 80%, no se realizarán acciones adicionales.

Tabla 24: Resultados de la detección de eventos en la zona C

Resultados de la detección de eventos en las pruebas de integración											
Evento	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Intel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
GDC	✓	✓	✓	✓	✓	10	✓	✓	✓	✓	90%
Microprocesador	✓	✓	✓	✓	X	✓	✓	✓	✓	✓	90%
CentroDeVisitantes	✓	✓	14	✓	✓	✓	✓	✓	✓	✓	90%
PianoCosmico	✓	✓	✓	✓	✓	✓	4	✓	✓	✓	90%
Muon	✓	✓	✓	✓	11	11	✓	✓	✓	✓	80%
Sustentabilidad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IA	✓	✓	✓	X	✓	✓	✓	✓	✓	✓	90%
IAUso	✓	✓	X	✓	✓	✓	X	✓	✓	✓	80%
Saludo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Presentacion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Cafe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Creador	✓	✓	✓	✓	✓	✓	✓	16	✓	✓	90%
TiempoLibre	✓	✓	✓	1	✓	✓	✓	✓	✓	✓	90%
Chiste	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Despedida	✓	15	✓	✓	✓	✓	X	✓	✓	✓	80%
Tour	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	90%
<b>PROMEDIO</b>											<b>91.76%</b>

En la Tabla 25 se muestran los resultados obtenidos de las invocaciones en la zona D del Centro de Visitantes. El porcentaje de detecciones exitosas en los eventos supera el 80% en la mayoría, con excepción de dos casos IA y Tour, que igualan este porcentaje. Este comportamiento muestra que en la zona D existe una buena detección, similar a la zona A, cuyo porcentaje promedio es de aproximadamente 94%. Incluso, se repite el comportamiento de mayor cantidad de falsos positivos que falsos negativos. Estos resultados pueden interpretarse como que al tener un porcentaje de detección elevado, los usuarios que utilicen el sistema en la zona D obtendrían una buena comunicación con el sistema Uéej, por este motivo, no es necesario realizar acciones adicionales.

Tabla 25: Resultados de la detección de eventos en la zona D

Resultados de la detección de eventos en las pruebas de integración											
Evento	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Intel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
GDC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Microprocesador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
CentroDeVisitantes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
PianoCosmico	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Muon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Sustentabilidad	✓	✓	✓	1	✓	✓	✓	✓	✓	✓	90%
IA	✓	✓	✓	11	✓	✓	✓	X	✓	✓	80%
IAUso	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Saludo	✓	11	16	✓	✓	✓	✓	✓	✓	✓	80%
Presentacion	✓	✓	✓	✓	10	✓	✓	✓	✓	✓	90%
Cafe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Creador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	90%
TiempoLibre	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Chiste	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Despedida	✓	✓	✓	X	✓	✓	✓	✓	✓	✓	90%
Tour	✓	✓	✓	✓	X	X	✓	✓	✓	✓	80%
<b>PROMEDIO</b>											<b>94.12%</b>

En la Tabla 26 se muestran los resultados obtenidos de las invocaciones en la zona E del Centro de Visitantes. El porcentaje de detecciones exitosas en la mayoría de los eventos supera el 80%, sin embargo, existen diversos casos en los que este porcentaje se equipara al establecido. Este comportamiento puede ocasionar conflictos por falsos positivos de eventos, no obstante, se encuentran en menor medida y la interacción en general en la zona es aceptable, sobre todo considerando que el porcentaje promedio de detección es de 93%, dando certeza de una buena interacción con los usuarios. No serán llevadas a cabo acciones posteriores al no existir valores por debajo del 80% establecido.

Tabla 26: Resultados de la detección de eventos en la zona E

Resultados de la detección de eventos en las pruebas de integración											
Evento	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Intel	✓	✓	11	12	✓	✓	✓	✓	✓	✓	80%
GDC	✓	✓	✓	✓	✓	✓	11	11	✓	✓	80%
Microprocesador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
CentroDeVisitantes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
PianoCosmico	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Muon	✓	✓	✓	2	✓	2	✓	✓	✓	✓	80%
Sustentabilidad	✓	✓	✓	12	✓	✓	✓	12	✓	✓	80%
IA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IAUso	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Saludo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Presentacion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Cafe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Creador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
TiempoLibre	✓	16	✓	✓	✓	✓	✓	✓	✓	✓	90%
Chiste	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Despedida	✓	X	✓	✓	✓	✓	✓	11	✓	✓	80%
Tour	✓	11	✓	✓	✓	✓	✓	✓	✓	✓	90%
<b>PROMEDIO</b>											<b>92.94%</b>

En la Tabla 27 se muestran los resultados obtenidos de las invocaciones en la zona F del Centro de Visitantes. Los porcentajes de detecciones exitosas en la mayoría de los eventos alcanza el 100%, haciendo de esta zona, una de las mejores para la detección de voz. En esta zona, también existe una cantidad reducida de falsos positivos lo que no representa inconvenientes significativos en la interacción esperada para los visitantes. Un punto para considerar es que los eventos TiempoLibre y Chiste tienen los menores porcentajes con 80% y podrían causar conflictos en ambientes con mucho ruido externo. Sin embargo no serán llevadas a cabo acciones posteriores al no existir valores por debajo del 80% establecido.

Tabla 27: Resultados de la detección de eventos en la zona F

Resultados de la detección de eventos en las pruebas de integración											
Evento	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Detecciones exitosas (%)
Intel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
GDC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Microprocesador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
CentroDeVisitantes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
PianoCosmico	✓	✓	✓	✓	✓	✓	1	✓	✓	✓	90%
Muon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Sustentabilidad	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
IAUso	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Saludo	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	90%
Presentacion	✓	✓	✓	✓	✓	16	✓	✓	✓	✓	90%
Cafe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Creador	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
TiempoLibre	✓	✓	1	✓	✓	✓	✓	12	✓	✓	80%
Chiste	✓	✓	✓	✓	✓	✓	✓	12	11	✓	80%
Despedida	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
Tour	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%
<b>PROMEDIO</b>											<b>95.88%</b>

Los resultados obtenidos serán analizados a detalle en la siguiente sección, donde también se realizarán recomendaciones para el mejor funcionamiento del sistema.

#### 4.4 Análisis de los resultados

Las pruebas realizadas a Uéej estuvieron enfocadas a conocer el comportamiento de las integraciones de voz en el flujo normal de operación de los sistemas Hochob, Amikoo y Juum. Estas integraciones son detección de palabra clave y detección de eventos.

El comportamiento de la detección de palabra clave se puede considerar adecuado al cumplir con las condiciones presentadas en la metodología, sin embargo, existe margen de mejora. Como se muestra en la Tabla 28, los resultados que se obtuvieron del primer conjunto de invocaciones con valor de sensibilidad default (0.5), estuvo por debajo del porcentaje mínimo de éxito en las detecciones. Al incrementar este valor, hay una mayor sensibilidad, lo cual provocó diversos falsos positivos de activación. El valor que cumplió con los requerimientos sin causar un gran número de falsos positivos fue 0.559. Este 80%, da la

certeza del buen funcionamiento del sistema y de una interacción adecuada de los usuarios, aunque es posible que en condiciones de ruido excesivo, existan inconvenientes para la operación: falsos negativos y falsos positivos.

Tabla 28: Porcentajes de detección de palabra clave con valores de sensibilidad en las pruebas espaciales

Porcentajes de detección de palabra clave con valores de sensibilidad en las pruebas espaciales		
Invocación	0.5	0.559
Hey Lupe	60%	80%

Después de realizar las pruebas a los sistemas de detección de eventos y realización de acciones, se pueden considerar que sus resultados fueron óptimos. Los porcentajes de detecciones exitosas superaron en su mayoría el 80% establecido en la metodología, así como, los valores promedio de las detecciones es de al menos 90%, estos datos se encuentran en la Tabla 29.

Tabla 29: Resumen de los porcentajes obtenidos en las pruebas espaciales de eventos

Resumen de los porcentajes obtenidos en las pruebas espaciales de eventos							
Evento	A	B	C	D	E	F	Detecciones exitosas (%)
Intel	100%	100%	100%	100%	80%	100%	96.67%
GDC	90%	80%	90%	100%	80%	100%	90.00%
Microprocesador	100%	100%	90%	100%	100%	100%	98.33%
CentroDeVisitantes	100%	100%	90%	100%	100%	100%	98.33%
PianoCosmico	90%	100%	90%	100%	100%	90%	95.00%
Muon	100%	100%	80%	100%	80%	100%	93.33%
Sustentabilidad	100%	100%	100%	90%	80%	100%	95.00%
IA	100%	100%	90%	80%	100%	100%	95.00%
IAUso	90%	90%	80%	100%	100%	100%	93.33%
Saludo	100%	100%	100%	80%	100%	90%	95.00%
Presentacion	90%	100%	100%	90%	100%	90%	95.00%
Cafe	100%	100%	100%	100%	100%	100%	100.00%
Creador	100%	90%	90%	90%	100%	100%	95.00%
TiempoLibre	80%	100%	90%	100%	90%	80%	90.00%
Chiste	90%	90%	100%	100%	100%	80%	93.33%
Despedida	90%	100%	80%	90%	80%	100%	90.00%
Tour	90%	100%	90%	80%	90%	100%	91.67%
<b>PROMEDIO</b>	94.71%	97.06%	91.76%	94.12%	92.94%	95.88%	

Otros comportamientos observados son los valores promedio de detección en las diferentes zonas. Las zonas con mayor porcentaje de detección son A, B, D y F; que de acuerdo con el diagrama mostrado en la Figura 24, son las más cercanas al arreglo de micrófonos. En cambio, Los menores porcentajes se presentaron en C y E, consideradas las más lejanas al sistema de micrófonos. Estos últimos valores, a pesar de ser menores a los de las zonas cercanas, son superiores al 90%, indicando que el sistema puede generar una buena interacción con las personas, incluyendo situaciones con ruido moderado. En condiciones de ruido abundante, es posible que aumenten falsos negativos o falsos positivos. Por estos motivos, se recomienda al personal, invocar el sistema en las zonas más cercanas para un mayor rendimiento e incluso los más lejanos en condiciones de poco ruido.

El comportamiento de los sistemas es adecuado para uso de los visitantes y no se requieren realizar acciones adicionales. Por este motivo, se da por concluido el desarrollo de Uéej y en la siguiente sección se presentan las conclusiones del desarrollo y las actividades que se realizaran en versiones futuras para mejorar el sistema.

## Capítulo 5

### Conclusiones y Trabajo Futuro

Se efectúa la valoración del trabajo realizado en este proyecto. Además, se propondrá el trabajo futuro al desarrollo de Uéej.

## 5.1 Conclusiones

El sistema Uéej fue implementado con el propósito de generar una experiencia interactiva con los usuarios del Centro de Visitantes de Intel Guadalajara con reconocimiento y generación de voz. El espacio recibe alrededor de 4,000 personas cada año, de diversas partes del mundo y con perfiles de estudiantes, empresarios, diplomáticos, por mencionar algunos; dado el número de visitantes, se considera el espacio importante para Intel Guadalajara y requiere un sistema de excelentes características. Para lograr esta meta, fueron requeridas diversas acciones: analizar los sistemas existentes del Centro de Visitantes para comprender el funcionamiento de sus componentes, reparar los componentes defectuosos, integrar un sistema de reconocimiento y generación de voz, diseñar las interacciones y materiales, además de, integrar estas tareas.

El análisis de componentes consistió en revisar el código de los sistemas en los repositorios y en los equipos de cómputo donde eran ejecutados, además de, realizar pruebas de funcionamiento de los sistemas. Esto con la finalidad de conocer el estado actual de los mismos y definir las acciones que se llevarían a cabo para la integración de la nueva interfaz de voz. Los sistemas tenían diversos componentes que no eran funcionales, así como, diversas acciones que no eran realizadas por el sistema, esto motivó a primero reparar los componentes necesarios: remover y actualizar la configuración del contenedor Home Assistant de acuerdo con su nueva especificación, sustituir las aplicaciones que realizaban acciones multimedia por versiones compatibles con la nueva especificación de Home Assistant y restaurar la comunicación entre el sistema Hochob y Juum.

Con los sistemas completamente funcionales, se diseñó la estrategia para integrar las funcionalidades de procesamiento y generación de voz. En primer lugar, fue necesario estudiar diferentes opciones de sistemas de procesamiento y generación de voz, arreglos de micrófonos y computadora de una sola tarjeta. El sistema de procesamiento y generación de voz elegido fue Rhasspy, el cual funciona de manera local favoreciendo la privacidad, presentando alternativas para sus componentes y permitiendo ejecutarse en equipos con pocos recursos, como lo son los ordenadores de una tarjeta; como arreglo de micrófonos, se seleccionó el UMA-8, el cual efectúa un procesamiento a la voz para obtener mejores resultados; finalmente como ordenador, se seleccionó la Upboard Squared, tarjeta con características necesarias para procesamiento de voz y con el potencial de integrar periféricos para aplicaciones de visión por computadora. Posteriormente, estos componentes fueron integrados a los sistemas Amikoo y Hochob. Para su integración, fue necesario diseñar la *Persona*, materiales y las interacciones que los usuarios tendrían con el sistema Uéej (frases, acciones y contenido multimedia); las interacciones están relacionadas con temas como STEM, la empresa Intel y su tecnología. Con las interacciones diseñadas, el sistema Home Assistant fue modificado para incluir las nuevas acciones diseñadas y la comunicación con el sistema de voz.

Integrados los componentes, fueron realizadas múltiples pruebas en diferentes contextos para conocer el comportamiento de este nuevo sistema. Estas pruebas consideraron los componentes individuales y su integración con diferentes contextos ambientales. Los resultados obtenidos en la mayoría de las pruebas fueron excelentes, alrededor del 90% de detecciones correctas, en las pruebas restantes el sistema obtuvo un 80% de detecciones correctas.

A partir de los resultados obtenidos, se puede afirmar que el sistema Uéej, con sus componentes de procesamiento y generación de voz, funcionan correctamente en el Centro de Visitantes de Intel Guadalajara. Este correcto funcionamiento da la certeza de una buena interacción con los futuros visitantes, una vez que las medidas de seguridad de Intel GDC lo permitan.

## 5.2 Trabajo futuro

Durante el desarrollo del sistema Uéej, se visualizaron áreas de mejora en futuras versiones, estas incluyen:

- Incluir visión por computadora en las interacciones del sistema Amikoo
- Añadir temas considerados en las interacciones
- Incrementar el porcentaje de detección en la frase clave y hacerlo robusto ante situaciones de ruido
- Incrementar el porcentaje de detección de eventos y hacerlo robusto ante situaciones de ruido
- Aumentar las interacciones del sistema Uéej
- Incrementar la interacción durante el recorrido del Centro de Visitantes

A futuro, se visualiza coordinar en el sistema Uéej funcionalidades de voz y visión por computadora para hacer las visitas más interactivas y reactivas ante los usuarios.

Este sistema puede aplicarse a diversos espacios como museos o centros educativos, donde se requieren sistemas relacionados con el área de STEM, con la finalidad de inspirar a los niños o jóvenes a estudiar carreras afines al área.



## Apéndice I: Archivo con frases de entrenamiento para Rhasspy

### [Saludo]

Saluda a (la gente | los visitantes | nuestras visitas )

### [Presentacion]

Preséntate

Quién eres

(Háblales | Cuéntales | Platica | Cuéntanos | Háblanos | Platícanos) [(acerca | sobre)] de ti

Que sabes hacer

### [Cafe]

(Prepara | haz | ponte a hacer | pon) café

Puedes (preparar | hacer) café

Qué tal si (haces | preparas | pones) café

### [Chiste]

(Cuenta | Di | Échate | Haz | Cuéntanos) (un chiste | una broma | algo gracioso)

### [Creador]

Quién es tu creador

Quién te (construyó | creó | diseñó | ensablo)

Cómo (naciste | fuiste creada )

### [Microprocesador]

Sabes que es un (microprocesador | procesador)

Que son los (microprocesadores | procesadores)

(Cuenta | Explica) que es un (microprocesador | procesador)

Habla de (microprocesadores | procesadores)

Qué es un (microprocesador | procesador)

### [IA]

(Cuenta | Explica | Platica | Sabes | Conoces ) que es [la] inteligencia artificial

Habla de [la] inteligencia artificial

Qué es [la] inteligencia artificial

[IAUso]

Qué (se | cosas) puede (hacer con | lograr) la inteligencia artificial

Para qué (se (usa | utiliza) | sirve) la inteligencia artificial

[GDC]

Qué se hace [aquí] [en [Intel] (GDC | Guadalajara | México | el Centro de Diseño de Guadalajara)]

Que hacen los ingenieros aquí [en [Intel] ( GDC | Guadalajara | México | el Centro de Diseño de Guadalajara)]

[TiempoLibre]

Qué haces (en tu tiempo libre | cuando no trabajas [aquí] )

Qué [mas] te gusta hacer

[CentroDeVisitantes]

(Que cosas podemos ver | Que hay | De que trata | Que muestran) [en] (el tour | el centro de visitantes | el recorrido | la visita | el museo)

[PianoCosmico]

Qué es (el | un) piano cósmico

De qué trata (el | un) piano cósmico

Cómo es (el | un) piano cósmico

(Dijiste | Mencionaste) piano cósmico

Para que (sirve | funciona | se necesita | hicieron) [un] piano cósmico

[Muon]

Qué (es un muon | son los muones)

Habla (del | de los) (muon | muones)

Que (sabes | conoces | puedes (contarnos | platicarnos | hablarnos)) de los muones

[Intel]

Que [cosas] (hace | fabrica | crea) Intel [a nivel (global | Internacional)]

A que se dedica Intel [a nivel (global | Internacional)]

Que (papel (tiene | cumple) ) Intel [a nivel (global | Internacional)]

[Sustentabilidad]

Que [cosas] hacemos para cuidar el [medio] ambiente [en (Intel Guadalajara | GDC | Guadalajara)]

Como (cuidamos | ayudamos) [en (Intel Guadalajara | GDC | Guadalajara)] (al | el) [medio] ambiente

[Despedida]

Nos (vemos | tenemos que ir) [Lupe]

Hasta (la vista | luego | la próxima) [Lupe]

Te cuidas [Lupe]

[Tour]

(Comienza | empieza) el (recorrido | tour | paseo)

Nos das el (recorrido | tour | paseo ) [por el centro de visitantes]

## Apéndice II: Reparación de los componentes actuales

Al realizar la instalación del sistema Hochob se encontró que el repositorio presentaba diversos errores causados por incompatibilidades, dichos errores pueden mostrarse en Figura 25. Por lo que será necesario repararlos antes de configurar los nuevos componentes.

```
hochob@hochob:~/hochob$ docker-compose build
WARNING: The XAUTH variable is not set. Defaulting to a blank string.
mosquitto uses an image, skipping
glances uses an image, skipping
influxdb uses an image, skipping
grafana uses an image, skipping
portainer uses an image, skipping
nodered uses an image, skipping
plex uses an image, skipping
Building homeassistant
Step 1/7 : FROM homeassistant/home-assistant
--> 04cd34803409
Step 2/7 : MAINTAINER
--> Using cache
--> 7e52d54e6ac0
Step 3/7 : ENV TIMEZONE America/Mexico_City
--> Using cache
--> 4fcc12496631
Step 4/7 : USER root
--> Using cache
--> c1940930b7d7
Step 5/7 : RUN apt-get update && apt-get install -y --no-install-recommends
eog psmisc mosquitto-clients alsa-utils libpulse0 pulsea
udio vim espeak vlc mplayer libgl1-mesa-glx libgl1-mesa-
dri mpg123 && rm -rf /var/lib/apt/lists/*
--> Running in 967e94704482
/bin/ash: apt-get: not found
ERROR: Service 'homeassistant' failed to build : The command '/bin/ash -o pipefa
il -c apt-get update && apt-get install -y --no-install-recommends eog
psmisc mosquitto-clients alsa-utils libpulse0 pulseaudio
vim espeak vlc mplayer libgl1-mesa-glx libgl1-mesa-dri m
pg123 && rm -rf /var/lib/apt/lists/*' returned a non-zero code: 127
```

Figura 25: Error de incompatibilidad al construir los contenedores del sistema Hochob

El error mostrado en la Figura 25, se debe a una incompatibilidad entre el gestor de paquetes que utiliza el contenedor Home Assistant y los comandos ejecutados para la construcción del sistema. Esto es provocado por un cambio en la imagen base del contenedor Home Assistant (Home Assistant, 2017), el cual utiliza el instalador de paquetes *apk*. Para solucionar estos problemas, se requiere cambiar el gestor de paquetes en la instalación de *apt-get install* a *apk-add*, además de, buscar el nombre de los paquetes en el gestor de paquetes *ash*; quedando de la manera mostrada en la Figura 26. Al hacer estos cambios, se generan los contenedores de manera exitosa.

```
RUN apk update && apk --no-cache add \
  alsa-utils alsaconf alsa-utils-doc eog espeak \
  libcanberra-gtk3 mesa-dri-intel mesa-gl mosquitto-clients \
  mplayer openrc psmisc pulseaudio shadow vim vld
```

Figura 26: Fragmento del nuevo conjunto de instrucciones para inicializar Hochob

Después de realizarse la instalación de los contenedores que conforman el sistema Hochob, se presentan nuevos errores, ahora en la interfaz de Home Assistant (Figura 27). El error presentado es la activación del modo seguro de Home Assistant, y es provocado por inconsistencias en su configuración.



Figura 27: Modo seguro de Home Assistant

Estos nuevos errores son causados por cambios en la configuración de Home Assistant en sus constantes actualizaciones. Los componentes que provocan el modo seguro del sistema son los relacionados a HTTP, *weblink* y sensores binarios, los cuales serán modificados.

Los componentes *weblink*, eran componentes visuales utilizados para mostrar otras páginas web. El sistema Hochob los utilizaba para mostrar las interfaces de los contenedores Grafana y Portainer. Sin embargo, los *weblink* fueron depreciados con la versión 0.105 y removidos en la 0.107 de Home Assistant (Nijhof, 0.105: Safe mode, Headers & Footers, New Zones Editor, Garmin, Sighthound, 2020), por lo que, se procede a eliminarlos. Con respecto al error en la configuración del componente HTTP, de acuerdo con la documentación (Home Assistant, s.f.) el campo *api\_password* encontrado en los archivos de configuración del sistema ha sido depreciado y se considera próximo a eliminar, por lo que, también será removido.

Acerca del error de configuración inválida en el grupo sensores binarios, en la actualización 0.107 de Home Assistant, se modificaron algunos campos para los grupos (Nijhof, 0.107: Multiple Lovelace Dashboards, adds helpers, new media player card, 2020). El campo que ocasiona este problema es *view*, y la solución es eliminar esa línea. Este campo se encuentra en los grupos *camera*, *default-view*, *image processing*, *sensors*, *switches*, *weather*, por lo que se realiza el mismo procedimiento.

Al terminar estas configuraciones, el sistema inicia correctamente, sin embargo, como se muestra en la Figura 28, aún existen diversos errores de menor prioridad. Estos errores están relacionados a componentes que han dejado de ser compatibles con Home Assistant, como los son *yweather*, *Darksky*, *Glances*. Estos componentes serán eliminados del sistema.

```
[homeassistant.config] Platform error: weather - Integration 'yweather' not found.  
2) [homeassistant.components.darksky.weather] Unable to connect to Dark Sky. 403 Client Error: Forbidden for url: https://api.darksky.net/forecast/123456  
[homeassistant.config] Platform error: sensor - Integration 'speedtest' not found.  
[homeassistant.components.sensor] The glances platform for the sensor integration does not support platform setup. Please remove it from your config.  
1) [homeassistant.components.darksky.sensor] Unable to connect to Dark Sky: 403 Client Error: Forbidden for url: https://api.darksky.net/forecast/1234567
```

Figura 28: Errores de menor prioridad mostrados al iniciar el sistema

Realizados estos cambios, el sistema no presenta ningún error al iniciarse. No obstante, al intentar ejecutar funcionalidades de audio, video y transmisión a Juum no se obtiene respuesta, por lo que será necesario revisar estos componentes

## Audio y video

Para desplegar las aplicaciones de audio y video, Hochob utiliza la aplicación VLC, sin embargo, como se muestra en la Figura 29, no es posible ejecutarla dentro de un espacio contenerizado con permisos de super usuario. Y dado que este permiso es necesario para la ejecución de Home Assistant, se procede a buscar otra opción de aplicación para desplegar audio y video.

```
bash-5.0# vlc /config/www/media/Amikoo/Video/LupeBrotherToMoon.mp4  
VLC is not supposed to be run as root. Sorry.  
If you need to use real-time priorities and/or privileged TCP ports  
you can use vlc-wrapper (make sure it is Set-UID root and  
cannot be run by non-trusted users first).  
bash-5.0# vlc-wrapper /config/www/media/Amikoo/Video/LupeBrotherToMoon.mp4  
Cannot determine unprivileged user for VLC!  
bash-5.0# vlc /config/www/media/Amikoo/Sound/IAMARobotSoundBibleCom.mp3  
VLC is not supposed to be run as root. Sorry.  
If you need to use real-time priorities and/or privileged TCP ports  
you can use vlc-wrapper (make sure it is Set-UID root and  
cannot be run by non-trusted users first).  
bash-5.0# vlc-wrapper /config/www/media/Amikoo/Sound/IAMARobotSoundBibleCom.mp3  
Cannot determine unprivileged user for VLC!
```

Figura 29: Errores por incompatibilidad de software para desplegar audio y video dentro de Hochob

La alternativa utilizada ha sido MPV, la cual ofrece las mismas capacidades para el despliegue de audio y video, sin conflictos de super usuario, utilizando menos recursos y siendo posible ejecutarla desde línea de comandos. Estos campos se realizan en el archivo Docker de Home Assistant agregando la dependencia MPV y en los servicios audio y video (Figura 30 a y b respectivamente). Para el servicio de audio será agregado el parámetro `--audio-device=alsa` para indicar que el dispositivo de salida de audio es el default de ALSA. En el servicio de video se incluirá el mismo parámetro y se agregaran los siguientes `--fs --keep-open=yes --loop`, estos parámetros indican que se ejecutara en modo pantalla completa, que al terminar el video la ventana del reproductor no se cerrará y que en caso de que el sistema lo requiera, el video se repetirá.

```

#!/bin/sh

set -x

export MPV_PID=$$
export MPV_BINARY=mpv
export MPV_ARGUMENTS="--audio-device=alsa --fs --keep-open=yes "
export MPV_LOOP="--loop"
export KILL_MPV="kill $(ps aux | grep 'mpv')"

LOCAL_LOOP="$1"
LOCAL_FILE="$2"

$KILL_MPV
if [ $# -eq 2 ]
then
    if [ "$LOCAL_LOOP" = "on" ]; then
        $MPV_BINARY $MPV_ARGUMENTS $MPV_LOOP $LOCAL_FILE
    elif [ "$LOCAL_LOOP" = "off" ]; then
        $MPV_BINARY $MPV_ARGUMENTS $LOCAL_FILE
    fi
else
    echo "Invalid number of arguments, see Documentation"
    exit 1
fi

```

a) b)

*Figura 30: Contenido de los archivos de servicios multimedia en Hochob  
a) servicio de audio, b) servicio de video*

## Transmisión de audio a Juum

Para realizar la transmisión de audio a Juum en versiones anteriores del sistema Hochob, se utilizaba la aplicación `stream2chromecast`, la cual está desarrollada en Python y es compatible con archivos de audio y video. Esta aplicación será retomada para reparar este sistema. El código de esta aplicación será colocado dentro del servicio de streaming y posteriormente es creado un archivo *shell* para invocar sus funcionalidades, dicho archivo se muestra en la Figura 31.

```

#!/bin/sh

set -x

. /services/main.sh

export STREAM_PID=$$
export STREAM2CHROMECAST_PYTHON_BINARY=/usr/bin/python2.7
export STREAM2CHROMECAST_BINARY=/services/streaming/stream2chromecast/stream2chromecast.py
export STREAM2CHROMECAST_ARGUMENT_SELECT_DEVICE="-devicename"
export STREAM2CHROMECAST_ARGUMENT_VOLUME="-setvol"
export STREAM2CHROMECAST_ARGUMENT_STOP="-stop"

LOCAL_DEVICE="$1"
LOCAL_VOLUME="$2"
LOCAL_FILE="$3"

if [ $# -eq 3 ]
then

    $STREAM2CHROMECAST_PYTHON_BINARY $STREAM2CHROMECAST_BINARY $STREAM2CHROMECAST_ARGUMENT_SELECT_DEVICE "$LOCAL_DEVICE" \
    $STREAM2CHROMECAST_ARGUMENT_VOLUME $LOCAL_VOLUME

    $STREAM2CHROMECAST_PYTHON_BINARY $STREAM2CHROMECAST_BINARY $STREAM2CHROMECAST_ARGUMENT_SELECT_DEVICE "$LOCAL_DEVICE" \
    $LOCAL_FILE

else
    echo "Invalid number of arguments, see Documentation"
    exit 1
fi

exit 0

```

*Figura 31: Archivo shell para invocar funcionalidades de streaming*

Posteriormente, se añaden referencias a este archivo en las aplicaciones que requieran transmitir video y audio, además, se agrega el servicio de voz como una opción (Figura 32). En el servicio de voz recibirá el nombre del dispositivo Chromecast, el archivo de voz generado y volumen.

```

#Look for chromecast device name
if [ "$LOCAL_ENGINE" = "espeak" ]; then
    espeak.sh $LOCAL_LANGUAGE "${LOCAL_TEXT}"
fi
$CHROMECAST_STREAM "$LOCAL_PLAY" "1.0" "$SPEECH_FILE"
;;

```

*Figura 32: Fragmento del archivo shell del servicio de voz*

Realizado esto las funcionalidades de transmisión a Juum operan con normalidad.

## Bibliografía

- AAEON Europa. (2020). *Especificaciones*. Obtenido de Upboard: <https://upboard.org/up/specifications/>
- AAEON Europa. (2020). *Upboard Squared*. Obtenido de Especificaciones: <https://upboard.org/upsquared/specifications/>
- Alianza MIPI. (Diciembre de 2015). *MIPI Display Serial Interface (MIPI DSI)*. Obtenido de Especificación: <https://www.mipi.org/specifications/dsi>
- Alianza MIPI. (Septiembre de 2019). *MIPI Camera Serial Interface 2 (MIPI CSI-2)*. Obtenido de Especificación: <https://www.mipi.org/specifications/csi-2>
- Alianza para la Promoción de STEM. (2019). *Visión STEM para México*.
- Amazon. (2020). *Alexa, Echo Devices, y su privacidad*. Obtenido de <https://www.amazon.com/gp/help/customer/display.html?nodeId=GVP69FUJ48X9DK8V>
- Amazon Alexa. (2020). *Alexa Voice Service*. Obtenido de <https://developer.amazon.com/en-US/docs/alexa/alexa-voice-service/get-started-with-alexa-voice-service.html>
- Amazon Alexa. (2020). *Development Resources*. Obtenido de <https://developer.amazon.com/en-US/alexa/devices/alexa-built-in/development-resources/international>
- Amazon. (s.f.). *Amazon Developers*. Obtenido de <https://developer.amazon.com/es-ES/alexa/branding/alexa-guidelines/communication-guidelines/brand-voice#:~:text=Alexa%20strives%20to%20be%20clever,at%20the%20expense%20of%20efficiency>.
- Amazon Science. (28 de Octubre de 2020). *How Alexa scientists are advancing speech science*. Obtenido de Amazon Science: <https://www.amazon.science/videos-webinars/how-alexa-scientists-are-advancing-speech-science>
- Benesty, J., Jingdong, C., & Huang, Y. (2008). *Microphone Array Signal Processing*. Springer, Berlin, Heidelberg.
- Bohouta, G., & Zěpuska, V. Z. (2017). Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx). *International Journal of Engineering Research and Applications*, 20-24. doi:10.9790/9622-0703022024
- Chowdhury , G. G. (2003). Natural Language Processing. *Annual Review of Information Science and Technology*, 51-55. Obtenido de <https://strathprints.strath.ac.uk/2611/1/strathprints002611.pdf>
- Chung, H., Iorga, M., Voas, J., & Lee, S. (2017). "Alexa, Can I Trust You?". *Computer*, 50(9), 100-104. doi:10.1109/MC.2017.3571053
- Dey, N., & Ashour, A. S. (2018). *Direction of Arrival Estimation and Localization of Multi-Speech Sources*. Springer, Cham.

Docker. (11 de Febrero de 2020). *Docker*. Obtenido de <https://docs.docker.com/get-started/>

Docker. (2020). *Docker Compose Documentacion*. Obtenido de <https://docs.docker.com/compose/>

Eclipse Foundation. (s.f.). Obtenido de Mosquitto: <https://mosquitto.org/>

Fundacion Raspberry Pi. (2020). *Raspberry Pi 3B*. Obtenido de Raspberry Pi: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Google Assistant. (10 de 11 de 2020). *Introduction to the Google Assistant Service*. Obtenido de Google Assistant SDK: <https://developers.google.com/assistant/sdk/guides/service/python>

Google Assistant. (11 de 02 de 2021). *Languages and locales*. Obtenido de Actions Console: <https://developers.google.com/assistant/console/languages-locales>

Google. (s.f.). *Ayuda*. Obtenido de Chromecast: [https://support.google.com/chromecast/answer/6279372?hl=es-419&ref\\_topic=6279364](https://support.google.com/chromecast/answer/6279372?hl=es-419&ref_topic=6279364)

Google. (s.f.). *Conversation Design*. Obtenido de Actions on Google: <https://designguidelines.withgoogle.com/conversation/conversation-design-process/create-a-persona.html#>

Google. (s.f.). *Website*. Obtenido de Chromecast: <https://store.google.com/mx/product/chromecast>

Grafana Labs. (2020). *Website*. Obtenido de <https://grafana.com/oss/grafana/>

Hennion, N. (s.f.). *Glances Webpage*. Obtenido de Glances: <https://nicolargo.github.io/glances/>

Hirschberg, J. &. (2015). Advances in natural language processing. *Science*, 349(6245), 261-266. Recuperado el 2019, de <https://nlp.stanford.edu/~manning/xyzy/Hirschberg-Manning-Science-2015.pdf>

Home Assistant. (Junio de 2017). *Foros*. Obtenido de Home Assistant: <https://community.home-assistant.io/t/new-base-image-for-hass-io/19926>

Home Assistant. (s.f.). *Authentication Providers*. Obtenido de Documentacion de Home Assistant: <https://www.home-assistant.io/docs/authentication/providers/>

Home Assistant. (s.f.). *Home Assistant*. Obtenido de <https://www.home-assistant.io/>

Honda Robotics. (s.f.). *Asimo*. Obtenido de Honda: <https://www.honda.mx/asimo/>

*Houndify*. (s.f.). Obtenido de <https://www.houndify.com/>

Hoy, M. B. (2018). Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*, 37(1), 81-88. doi:10.1080/02763869.2018.1404391

IBM. (s.f.). *What is edge computing?* Obtenido de Cloud: <https://www.ibm.com/cloud/what-is-edge-computing>

- InfluxData. (2020). *InfluxDB*. Obtenido de <https://www.influxdata.com/products/influxdb-overview/>
- Intel Corporation. (2015). *Intel Edison Documentation*. Obtenido de [https://www.intel.com/content/dam/support/us/en/documents/edison/sb/edison\\_pb\\_331179002.pdf](https://www.intel.com/content/dam/support/us/en/documents/edison/sb/edison_pb_331179002.pdf)
- Intel Corporation. (s.f.). *Website*. Obtenido de NUC: <https://www.intel.la/content/www/xl/es/products/boards-kits/nuc.html>
- Intel. (s.f.). *Intel RealSense*. Obtenido de <https://www.intelrealsense.com/>
- Intel. (s.f.). *Intel RealSense Depth Camera D435i*. Obtenido de Intel RealSense: <https://www.intelrealsense.com/depth-camera-d435i/>
- Kěpuska, V., & Bohouta, G. (2018). Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). En IEEE (Ed.), *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, (págs. 99-103). Las Vegas.
- Lai, C. C., Nordholm, S. E., & Leung, Y. H. (2017). *A Study into the Design of Steerable Microphone Arrays*. Singapur: Springer.
- Lynskey, D. (9 de Octubre de 2019). 'Alexa, are you invading my privacy?' – the dark side of our voice assistants. Obtenido de <https://www.theguardian.com/technology/2019/oct/09/alexa-are-you-invading-my-privacy-the-dark-side-of-our-voice-assistants>
- Makena, K., & Statt, N. (3 de Julio de 2019). *Amazon confirms it holds on to Alexa data even if you delete audio files*. Obtenido de The Verge: <https://www.theverge.com/2019/7/3/20681423/amazon-alexa-echo-chris-coons-data-transcripts-recording-privacy>
- miniDSP. (2020). *UMA-8 USB mic array - V2.0*. Obtenido de MiniDSP: <https://www.minidsp.com/products/usb-audio-interface/uma-8-microphone-array>
- miniDSP. (2020). *Using the miniDSP UMA-8 microphone array with Google Assistant and the Raspberry Pi*. Obtenido de UMA-8 Applications: <https://www.minidsp.com/applications/usb-mic-array/uma-8-google-assistant>
- miniDSP. (2021). *Using the miniDSP UMA-8 microphone array with Google Assistant and the Raspberry Pi*. Obtenido de UMA-8 Applications: <https://www.minidsp.com/applications/usb-mic-array/uma-8-google-assistant>
- Moriuchi, E. (2019). Okay, Google!: An empirical study on voice assistants on consumer engagement and loyalty. *Psychology & Marketing*, 1-13. doi:<https://doi.org/10.1002/mar.21192>
- Movidius e Intel. (2018). *Enhanced Visual Intelligence at the Network Edge*. Obtenido de Intel Movidius Myriad X VPU:

- <https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/myriad-x-product-brief.pdf>
- Moyal, A., Aharonson, V., Tetariy, E., & Gishri, M. (2013). *Phonetic Search Methods for Large Speech Databases*. New York: Springer.
- MQTT. (2020). *MQTT Website*. Obtenido de <https://mqtt.org/>
- Mycroft. (2020). *Documentation*. Obtenido de Mycroft AI: <https://mycroft-ai.gitbook.io/docs/>
- NG, A. (02 de Julio de 2019). Amazon Alexa keeps your data with no expiration date, and shares it too. Obtenido de <https://www.cnet.com/news/amazon-alexa-keeps-your-data-with-no-expiration-date-and-shares-it-too/#:~:text=The%20deadline%20for%20answers%20was,re%20manually%20deleted%20by%20users>.
- Nijhof, F. (2 de Febrero de 2020). *0.105: Safe mode, Headers & Footers, New Zones Editor, Garmin, Sighthound*. Obtenido de Home Assistant Blog: <https://www.home-assistant.io/blog/2020/02/05/release-105/>
- Nijhof, F. (18 de Marzo de 2020). *0.107: Multiple Lovelace Dashboards, adds helpers, new media player card*. Obtenido de Home Assistant Blog: <https://www.home-assistant.io/blog/2020/03/18/release-107/>
- OpenJS Foundation. (s.f.). *Website*. Obtenido de <https://nodered.org/>
- Organizacion Hochob. (s.f.). *Repositorio Hochob*. Obtenido de <https://github.com/hochob/hochob>
- Picovoice. (2021). *Pricing*. Obtenido de Picovoice: <https://picovoice.ai/pricing/>
- Plaza, P., Sancristobal, E., Carro, G., Castro, M., Blázquez, M., Muñoz, J., & Álvarez, M. (2018). Scratch as Educational Tool to Introduce Robotics. En G. D. Auer M. (Ed.), *Teaching and Learning in a Digital World*. 715, págs. 3-14. Springer, Cham.
- Polyakov E.V., M. M. (2018). Investigation and development of the intelligent voice assistant for the Internet of Things using machine learning. *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)* (págs. 1-5). Moscow: IEEE. doi:10.1109/MWENT.2018.8337236
- Raju, A., Filimonov, D., Tiwari, G., Lan, G., & Rastrow, A. (2019). Scalable Multi Corpora Neural Language Models for ASR. *Interspeech*.
- Reyes Montero, A. (s.f.). *Documentacion Amikoo*. Obtenido de Gitbook: <https://alfredo-reyes-montero.gitbook.io/amikoo/>
- Rhasspy. (2021). *Rhasspy Documentation*. Obtenido de Rhasspy Voice Assistant: <https://rhasspy.readthedocs.io/en/latest/>
- Rhasspy. (Enero de 2021). *Speech to Text*. Obtenido de Rhasspy Documentation: <https://rhasspy.readthedocs.io/en/latest/speech-to-text/>

- Rhasspy. (2021). *Wake Word*. Obtenido de Rhasspy Documentation: <https://rhasspy.readthedocs.io/en/latest/wake-word/>
- Rivas-Gomez, S., Pena, A. J., Moloney, D., Laure, E., & Markidis, S. (2018). Exploring the Vision Processing Unit as Co-processor for Inference. *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 589-598. doi:10.1109/IPDPSW.2018.00098
- Seeed Studio. (2020). *ReSpeaker Mic Array v2.0*. Obtenido de Seeed Studio Wiki: [https://wiki.seeedstudio.com/ReSpeaker\\_Mic\\_Array\\_v2.0/#specification](https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/#specification)
- Softbank. (s.f.). *Robot*. Obtenido de Softbank: <https://www.softbank.jp/en/robot/>
- Soumya, S., Anjan, D., & Nilanjan, D. (2019). *Audio Processing and Speech Recognition*. Springer Singapore. doi:10.1007/978-981-13-6098-5
- Sreenivasa Rao, K., & Narendra, N. P. (2019). *Source Modeling Techniques for Quality Enhancement in Statistical Parametric Speech Synthesis*. Springer.
- Terzopoulos, G., & Satratzemi, M. (2020). Voice Assistants and Smart Speakers in Everyday Life and in Education. *Informatics in Education*, 19(3), 473-490. doi:10.15388/infedu.2020.2
- Yu, D., & Deng, L. (2015). *Automatic Speech Recognition*. London: Springer London.