



Universidad Autónoma de Querétaro  
Facultad de Informática  
Maestría en Ciencias de la Computación

Localización del sonido utilizando *beamforming* y su representación en un  
dispositivo de realidad mixta

Tesis

Que como parte de los requisitos para obtener el grado de  
Maestro en Ciencias de la Computación

Presenta:

Ing. Aldo Valencia Palma

Dirigido por:

Dra. Diana Margarita Córdova Esparza

Co-dirigido por:

Dra. Diana Margarita Córdova Esparza  
Presidente

M C Julio Alejandro Romero González  
Secretario

Dra Ana Marcela Herrera Navarro  
Vocal

Dra Ma Teresa García Ramírez  
Suplente

M C Ricardo Chaparro Sánchez  
Suplente

Centro Universitario, Querétaro, Qro.  
Agosto 2019  
México

## RESUMEN

La localización acústica es una técnica que permite medir la intensidad y la dirección de una fuente de sonido. Una técnica popular para su cálculo es el algoritmo de formación de haces "Sum-and-Delay" que genera un ángulo aproximado de su ubicación mientras reduce el ruido en la señal. Combinando esta técnica con avances en entornos de realidad mixta es posible desarrollar aplicaciones esenciales tanto en la industria como más allá. Sin embargo, estos sistemas pueden ser demasiado grandes y costosos. En este trabajo, se diseñó un método para aproximar la dirección sonora de llegada utilizando una técnica de *beamforming* para representarlo en un entorno de realidad mixta utilizando un dispositivo integrado, logrando aproximarlo con un error medio del 2,5% en distancias de hasta 3 metros largo. Entre sus principales ventajas se encuentra que se trata de un sistema portátil y de bajo coste en el que se pueden desarrollar aplicaciones de gran utilidad, como asistentes de conversación de realidad mixta capaces de generar subtítulos en tiempo real para ayudar a las personas con sordera en un aula que no tienen un traductor, o para incorporar un servicio de traducción para entender a los hablantes que hablan otro idioma.

## SUMMARY

Acoustic localization is a technique that allows measuring the intensity and direction of a sound source. A popular technique for its calculation is the beamforming algorithm “Sum-and-Delay” which generates an approximate angle of its location while reducing noise in the signal. Combining this technique with advances in mixed reality environments it is possible to develop essential applications both in the industry and beyond. However, these systems can be too large and expensive. In this work, an algorithm was developed to approximate the sound direction of arrival using a beamforming technique to represent it in mixed reality environment using an embedded device, achieving to approximate it with an average error of 2.5 % in distances up to 3 meters long. Among its main advantages is that it is a portable and low-cost system in which applications of great utility can be developed, such as mixed reality conversation assistants capable of generating subtitles in real-time to assist people with deafness in a classrooms that do not have a translator, or to incorporate a translation service to understand speakers who speak another language.

## AGRADECIMIENTOS

Primeramente agradezco a la Universidad Autónoma de Querétaro por abrirme las puertas y darme la oportunidad de continuar mi desarrollo académico y profesional, así como a los profesores que me brindaron sus conocimientos y apoyo.

Agradezco también el apoyo financiero para la realización de este trabajo al Consejo Nacional de Ciencia y Tecnología (CONACYT) mediante la beca número 494038 y a la Universidad Autónoma de Querétaro (UAQ) a través del proyecto FIF-2018-06.

Sobre todo agradezco a mi asesora de tesis la Dra. Diana Margarita Córdova Esparza, por guiarme y brindarme la oportunidad de recurrir a su capacidad y conocimiento científico durante todo el desarrollo de esta tesis.

Finalmente, también este logro va dedicado a mis padres que siempre me han apoyado en todas mis decisiones.

## TABLA DE CONTENIDOS

<b>1. Introducción</b>	<b>9</b>
1.1. Planteamiento del problema . . . . .	11
1.2. Justificación . . . . .	12
1.3. Objetivo general . . . . .	14
1.4. Objetivos específicos . . . . .	14
1.5. Hipótesis . . . . .	15
1.6. Alcances y limitaciones . . . . .	15
1.7. Estructura de la tesis . . . . .	15
<b>2. Estado del arte</b>	<b>16</b>
<b>3. Marco teórico</b>	<b>18</b>
3.1. Descripción de la técnica <i>beamforming</i> . . . . .	18
3.2. Realidad Mixta . . . . .	20
<b>4. Metodología</b>	<b>21</b>
4.1. Fase de análisis . . . . .	22
4.2. Fase de desarrollo . . . . .	22
4.3. Fase de implementación . . . . .	23
4.3.1. Implementación del algoritmo en el sistema embebido	23

4.3.2. Descripción del hardware . . . . .	23
4.3.3. Funcionamiento del dispositivo . . . . .	24
4.4. Fase de validación . . . . .	26
4.4.1. Cálculo de errores . . . . .	26
<b>5. Resultados</b>	<b>27</b>
<b>6. Conclusiones y trabajo futuro</b>	<b>28</b>
<b>Referencias</b>	<b>30</b>
<b>A. Anexo: Código fuente</b>	<b>33</b>
<b>B. Anexo: Hoja de datos Matrix creator</b>	<b>39</b>
<b>C. Anexo: Experimentos previos</b>	<b>43</b>

## ÍNDICE DE TABLAS

1. Resultados de las pruebas . . . . .	29
--	----

Dirección General de Bibliotecas de la UAQ

## ÍNDICE DE FIGURAS

1.	Técnica de <i>beamforming Sum and Delay</i> . . . . .	19
2.	Metodología basada en el diseño . . . . .	22
3.	Diagrama de conexiones . . . . .	24
4.	Funcionamiento del sistema . . . . .	25
5.	Alineación de centro . . . . .	26
6.	Medición de error absoluto . . . . .	27
7.	Matrix creator . . . . .	40
8.	Diagrama de bloques . . . . .	42
9.	Concepto prototipo . . . . .	43

## 1 Introducción

El proceso de localizar fuentes de sonido mediante el registro de las señales acústicas que se propagan a través de varios sensores y su análisis se le conoce como técnica de localización de la fuente acústica (Kundu, Nakatani, y Takeda, 2012). Los componentes más utilizados que se estiman de una fuente de sonido son dos: el ángulo de llegada y la distancia hasta la fuente. La obtención de estos parámetros puede ser crucial para aplicaciones que requieren tales estimaciones en etapas de procesamiento posteriores, como la separación de la fuente de sonido (Rascon y Meza, 2017), la clasificación del tipo de señal acústica (Dong, Antoni, Pereira, y Kellermann, 2016) y el reconocimiento de voz automático (Acero y Stern, 1990).

En la actualidad, varias aplicaciones comerciales han aprovechado la localización de fuentes de sonido principalmente en sistemas de videoconferencia (Wang y Chu, 1997), control de actuadores en robótica (Zhong, Sun, y Yost, 2016) y sistemas de seguridad (Y. Huang, Benesty, y Elko, 2000). Sin embargo, con los rápidos avances tecnológicos, viene la capacidad de mejorar la forma en que es adquirida y desplegada la información dentro del entorno que nos rodea. Es posible combinar el conocimiento percibido por los sentidos de los seres humanos en dispositivos móviles, y de esta manera crear entornos virtuales (Z. Huang, Hui, Peylo, y Chatzopoulos,

2013). Uno de estos entornos es la realidad mixta, donde el usuario puede ver información útil anclada a puntos reales en el espacio donde coexisten e interactúan en tiempo real con otros objetos, haciendo posible tratarlos como reales (Noor, 2016).

En cuanto a la industria, la representación del sonido puede utilizarse en sistemas de realidad mixta para llevar a cabo diferentes tareas. Por ejemplo, se aplica en cámaras acústicas para el desarrollo de prototipos y la detección y diagnóstico de fallas (Lanslots, Deblauwe, y Janssens, 2010). La desventaja es que este tipo de equipos suele ser especializado, por lo que es de gran tamaño y costoso para los usuarios promedio de electrónica de consumo.

Fuera de las aplicaciones del campo industrial, se han desarrollado prototipos que hacen uso de la localización acústica para ayudar a las personas con sordera total, ya que los implantes cocleares convencionales sólo amplifican la señal, pero no ayudan a localizar la fuente de sonido (Azar, Saleh, y Al-Alaoui, 2007; Jain y cols., 2015).

El trabajo que se presenta en este documento consiste en desarrollar un algoritmo capaz de localizar eventos acústicos en el entorno, visualizar la presencia y dirección del sonido en realidad mixta utilizando un dispositivo embebido. De esta manera, se logró implementar un sistema base para el desarrollo de aplicaciones en dispositivos embebidos que pueda ayudar a los usuarios con problemas auditivos en conversaciones, representando la

información mediante realidad mixta.

## 1.1. Planteamiento del problema

Krishnaveni et al. (Krishnaveni, Kesavamurthy, y Aparna, 2013) describen en su trabajo que la localización de la dirección de llegada de una onda (por sus siglas en inglés, DOA) a un plano es un problema conocido desde principios del siglo XX, siendo las aplicaciones más importantes de la época los radares, sonares y sistemas sísmicos. Sin embargo, en cuanto a señales acústicas no fue hasta el año de 1974 que se utilizaron arreglos de micrófonos para estimar el origen de fuentes de sonido (Michel y cols., 2006). Desde entonces ha habido grandes avances tecnológicos que se manifiestan en nuevas aplicaciones como las videoconferencias (Wang y Chu, 1997) y los sistemas de vigilancia (Y. Huang y cols., 2000).

A pesar de los avances tecnológicos, en la actualidad aún no existe un medio de asistencia para que las personas con discapacidad auditiva que sea capaz de representar en realidad mixta eventos sensoriales importantes para advertir la presencia y dirección de sonidos (Stone y Tata, 2017).

Este tipo de sistemas son necesarios ya que los dispositivos de asistencia colcleares convencionales mejoran el reconocimiento de voz, pero no mejoran la localización del sonido. Las personas sordas o con dificultades auditivas se basan en señales visuales como el lenguaje corporal, las expresiones faciales y el movimiento de los labios para interpretar el ha-

bla. Saber dónde enfocar la atención visual es un requisito para una lectura eficaz del habla (Jain y cols., 2015).

Entre las características deseables para el diseño de un prototipo de asistencia portátil que permita identificar eventos sonoros, se identificaron que los elementos más importantes a representar son la presencia, dirección e intensidad de sonido y también destacó como deseable la clasificación de los sonidos y la conversión de voz a texto (Jain y cols., 2015; Azar y cols., 2007).

Actualmente es posible utilizar las técnicas de localización acústica para generar nuevos medios de asistencia para personas sordas gracias a los avances de las tecnológicos que permiten representar características del sonido en realidad mixta. De esta manera, pueden ayudar al usuario a localizar el origen del sonido con señales visuales a diferencia de los implantes cocleares normales que solo amplifican el sonido (Jain y cols., 2015).

## **1.2. Justificación**

En México, se estima que alrededor de 10 millones de personas tienen algún tipo o grado de problema auditivo y 400,000 presentan sordera total. El término «hipoacusia» o sordera se refiere a los problemas auditivos que pueden abarcar desde la dificultad ligera para comprender lo que se escucha hasta sordera profunda (Carranza Alva, Toral Martiñón, Shkurovich Bialik, y Schabes Rostenberg, 2016).

Sin tecnologías de asistencia no habría posibilidad que las personas con sordera pudieran redireccionar la atención para advertir eventos sensoriales y reasignar recursos perceptivos para lidiar con ellos. Adquirir la capacidad de detectar estas señales es fundamental para interactuar con el mundo (Stone y Tata, 2017).

En el estado de Querétaro existe la asociación civil Comisión de Personas A.C., que tiene por visión eliminar las barreras de comunicación y discriminación impulsando la participación de las personas sordas en la sociedad en general. Teniendo esta información se realizaron encuestas a estudiantes de la comisión para conocer la forma en que tecnológicamente se pudiera brindar una herramienta de apoyo que fuera útil para la comunidad en cuanto a comunicación se refiere. Mediante las encuestas se detectó que el principal interés de los estudiantes es tener un dispositivo de asistencia dentro de las aulas que les permita saber quién es la persona que está hablando y que es lo que está diciendo y de esta manera no depender que haya un traductor para entender al profesor.

De esta necesidad surge el proyecto de investigación planteado en este documento, que tiene por objetivo desarrollar un algoritmo para la localización de un evento sonoro proveniente de un entorno controlado, implementado en un dispositivo de asistencia portátil y fácil de utilizar que identifica al emisor de una fuente de sonido en tiempo real. Además de mostrar el habla a través de un servicio de voz a texto con subtítulos en tiempo real y

un dispositivo de realidad mixta.

De esta manera, el dispositivo desarrollado en este proyecto permite asistir a personas sordas, advertir eventos sonoros en un aula y redireccionar su atención a través de un dispositivo de realidad mixta, y de esta manera identificar la presencia y dirección del emisor de una fuente de sonido en tiempo real.

### **1.3. Objetivo general**

Desarrollar un algoritmo para aproximar y visualizar el origen del sonido en el espacio utilizando la técnica de *beamforming* y realidad mixta que pueda integrarse a un dispositivo de asistencia para personas con discapacidad auditiva.

### **1.4. Objetivos específicos**

- Integrar hardware para un dispositivo embebido capaz de representar el origen del sonido mediante realidad mixta.
- Implementar la aproximación del sonido en el prototipo embebido.
- Realizar pruebas y análisis estadístico de resultados.

## 1.5. Hipótesis

Es posible desarrollar un algoritmo que sea capaz de aproximar y representar el origen del sonido en el espacio, utilizando la técnica de *beam-forming* y realidad mixta. El algoritmo será utilizado en un dispositivo de asistencia que permitirá a las personas con discapacidad auditiva estar conscientes de la presencia y dirección del habla en un entorno controlado.

## 1.6. Alcances y limitaciones

- El prototipo de asistencia es personal, idealmente el alumno usará su monitor dónde se mostrará la presencia y dirección del sonido.
- Puede representar un sonido a la vez.
- Funciona en una distancia de hasta 3 metros.

## 1.7. Estructura de la tesis

El trabajo de tesis se encuentra organizado de la siguiente manera: En el Capítulo 1 se describe la Introducción al proyecto de investigación, en el Capítulo 2 se presenta Estado del arte donde se mencionan una serie de trabajos relacionados al tema, en el Capítulo 3, los fundamentos teóricos del proyecto. En el Capítulo 4, se presenta la Metodología desarrollada para la implementación del proyecto, además se especifica el funcionamiento

del sistema. En el Capítulo 5, se muestran los resultados obtenidos y finalmente, en el Capítulo 6, se describen las conclusiones y las sugerencias para trabajo futuro.

## **2 Estado del arte**

La localización y representación del sonido en un dispositivo embebido, tiene como posible aplicación la asistencia a personas con alguna discapacidad auditiva que les permita advertir eventos sensoriales importantes como la dirección e identificación de sonidos (Stone y Tata, 2017). Sin embargo, los sistemas de asistencia que se han desarrollado están limitados por dificultad de uso, falta de funcionalidad o movilidad del sistema.

(Jain y cols., 2015), tomó en cuenta las consideraciones anteriores durante la encuesta para el diseño de un prototipo de realidad aumentada con el fin de asistir a personas sordas. Los elementos más importantes identificados en la encuesta fueron la presencia, dirección e intensidad de sonido y también destacó como deseable la clasificación de los sonidos y la conversión de voz a texto. El mayor problema con el prototipo fue que dirección del sonido mostrada al usuario correspondía a la posición del sensor el cual estaba fijo sobre la mesa y no al movimiento de la pantalla.

El proceso de localizar fuentes de sonido mediante el registro de las señales acústicas que se propagan a través de varios sensores y su posterior análisis se conoce como la técnica de localización de la fuente acústica

(Kundu y cols., 2012). Los componentes más utilizados que se estiman de una fuente de sonido son dos: el ángulo de llegada y la distancia hasta la fuente. La obtención de estos parámetros puede ser crucial para aplicaciones que requieren tales estimaciones en etapas de procesamiento posteriores, como la separación de la fuente de sonido (Rascon y Meza, 2017), la clasificación del tipo de señal acústica (Dong y cols., 2016) y el reconocimiento de voz automático (Acero y Stern, 1990).

(Krishnaveni y cols., 2013) comentan que la localización de la dirección de llegada de una onda (por sus siglas en inglés, DOA) a un plano es un problema conocido desde principios del siglo XX, siendo las aplicaciones más importantes de la época los radares, sonares y sistemas sísmicos. Sin embargo, (Michel y cols., 2006) describe que en cuanto a señales acústicas no fue hasta el año de 1974 que se utilizaron arreglos de micrófonos para estimar el origen de fuentes de sonido. Desde entonces ha habido grandes avances tecnológicos que se manifiestan en nuevas aplicaciones como las videoconferencias y los sistemas de vigilancia.

En un estudio reciente (Rascon y Meza, 2017), reporta que los componentes de una fuente de sonido que pueden estimarse son dos: la dirección de llegada y la distancia. Obtener estos dos parámetros es crucial para aplicaciones que requieren dichas estimaciones en etapas de procesamiento posteriores, como separación de fuente de sonido, clasificación de fuente de sonido y reconocimiento automático de voz

De igual presenta la clasificación de los diversos algoritmos en los siguientes tipos:

- Estimación unidimensional de dirección de llegada única.
- Estimación bidimensional bidireccional de la dirección de llegada.
- Estimación de dirección de llegada múltiple.
  - Basados en *beamforming*.
  - Métodos de subespacio.
  - Fuente de agrupamiento a través del tiempo.
- Estimación de la distancia.

### **3 Marco teórico**

#### **3.1. Descripción de la técnica *beamforming***

En la ubicación de las fuentes de sonido, la técnica de formación de haces es popular por su versatilidad ya que funciona como un tipo de filtro espacio-temporal. Es decir, manipula las señales en el dominio del tiempo. Esto permite extraer una señal en una dirección específica en un entorno sujeto a interferencias. El algoritmo “Sum-and-Delay” (Pessentheiner, Petrik, y Romsdorfer, 2012) dentro de la categoría *beamforming*, requiere una matriz de sensores para determinar el ángulo de llegada de una señal y

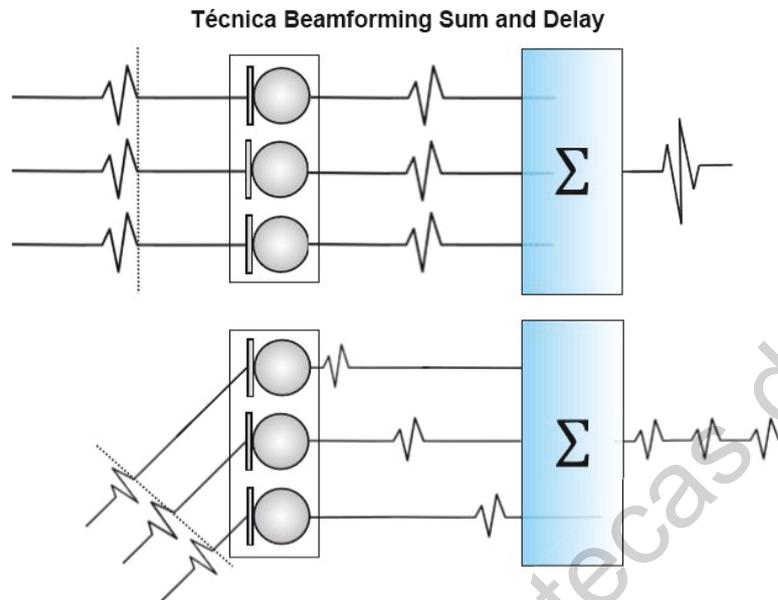


Figura 1: Técnica de *beamforming Sum and Delay*

luego sincronizar todos los canales para mejorar la señal procedente de un cierto ángulo. El resultado es que las señales de una dirección particular experimentan una superposición constructiva, mientras que otras experimentan interferencias destructivas. Por lo general, este algoritmo se utiliza para filtrar señales de ruido estacionarias y no coherentes para que se pueda aplicar en entornos de bajo ruido ambiental (ver Figura 3.1). La señal de interés se amplifica señalando el haz a la señal deseada.

Otra característica poderosa de la técnica de formación de haces es la capacidad de dirigir electrónicamente el haz sin manipular físicamente la matriz de micrófonos. El cálculo se puede realizar dinámicamente cuando detecta nuevas fuentes de sonido y genera continuamente valores de un ángulo con respecto a la fuente de sonido (Greensted, s.f.). El cálculo de

los coeficientes en una matriz uniforme circular en el dominio de frecuencia para un micrófono se expresa como (1):

$$W_n(\omega) = e^{-i\frac{\omega}{c}\frac{d}{2}\cos(\varphi-\phi_n)} \quad (1)$$

y para toda la matriz (2):

$$W(\omega) = \frac{1}{N} \sum_{n=1}^N e^{-i\frac{\omega}{c}\frac{d}{2}\cos(\varphi-\phi_n)} \quad (2)$$

Donde  $\omega$  es la frecuencia angular,  $N$  es el número de micrófonos,  $c$  es la velocidad del sonido  $d$  es el diámetro de la matriz,  $\phi_n$  es el ángulo del índice del micrófono  $n$  y  $\varphi$  es el ángulo en el que debe dirigirse para procesar la señal deseada

### 3.2. Realidad Mixta

La realidad mixta, es la visualización de objetos digitales anclados a puntos reales en el espacio que coexisten e interactúan en tiempo real, haciendo posible que el usuario los trate como reales (Noor, 2016). Aplicado a la asistencia de personas con discapacidad auditiva en realidad mixta puede utilizarse cualquier pantalla como medio de monitoreo, pero requiere que exista una relación conocida entre ésta y la matriz de micrófonos.

Gracias al rápido desarrollo de la tecnología se puede mejorar la manera en que adquirimos, interactuamos y mostramos la información de nuestro

ambiente. Ahora somos capaces de fundir la información de nuestros sentidos con los sensores de nuestros dispositivos, y podemos crear entornos de realidad mixta que antes no eran posibles (Z. Huang y cols., 2013).

## 4 Metodología

En este capítulo se describen cada una de las fases que se llevaron a cabo para implementar un algoritmo que permite aproximar y visualizar el origen del sonido en el espacio utilizando la técnica de *beamforming* y realidad mixta, integrado en un dispositivo de asistencia para personas con discapacidad auditiva.

La metodología se desarrolló de acuerdo al modelo de investigación basado en el diseño (ver Figura 2). En donde la investigación se inicia con el análisis de la situación y la definición del problema. La fase continua es la implementación seguida de la recogida de información, esta última se lleva a cabo a lo largo de todo el proceso. Finalmente, el proceso de investigación se concreta mediante ciclos continuos de diseño, validación, análisis y rediseño hasta el perfeccionamiento de la intervención (de Benito Crosetti y Ibáñez, 2016).

Enseguida se describen cada una de las etapas desarrolladas.

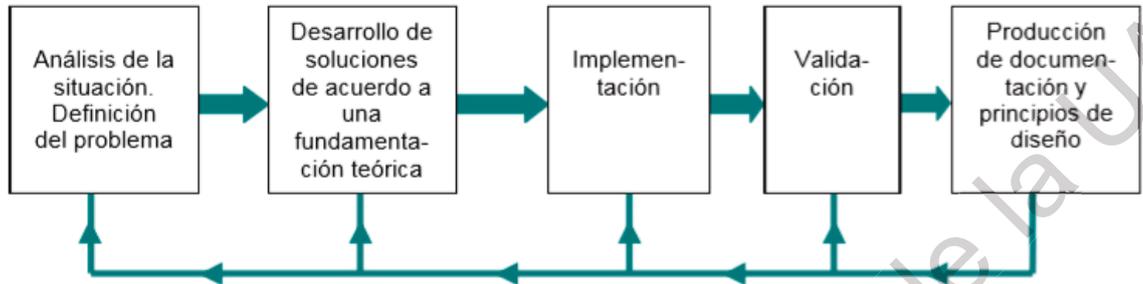


Figura 2: Metodología basada en el diseño

#### 4.1. Fase de análisis

- Búsqueda de artículos en la literatura sobre dispositivos de asistencia a personas con discapacidad auditiva para garantizar la pertinencia de la investigación.
- Investigar diferentes algoritmos basados en la técnica de *beamforming* y arreglos de micrófonos para estimar el origen del sonido.

#### 4.2. Fase de desarrollo

- Análisis de requerimientos del sistema de prueba.
- Desarrollo de un algoritmo para mostrar origen del sonido en realidad mixta.

### **4.3. Fase de implementación**

#### **4.3.1. Implementación del algoritmo en el sistema embebido**

El sistema propuesto para probar el algoritmo debe seguir el siguiente procedimiento, que permitirá representar el origen del sonido en realidad mixta:

- Los micrófonos adquieren una señal de audio.
- Se calcula el ángulo e intensidad de llegada de la señal acústica.
- Se estiman las coordenadas del origen del sonido.
- Se envían coordenadas a algoritmo de representación de sonido en realidad mixta.
- Se toma la imagen de la cámara y se añade animación en coordenadas de origen del sonido.
- Se manda imagen a pantalla.
- Medición de error.

#### **4.3.2. Descripción del hardware**

Se utilizó un dispositivo Raspberry Pi 3 modelo B+ con la placa de desarrollador Matrix Creator que contiene una matriz circular de 8 micrófo-

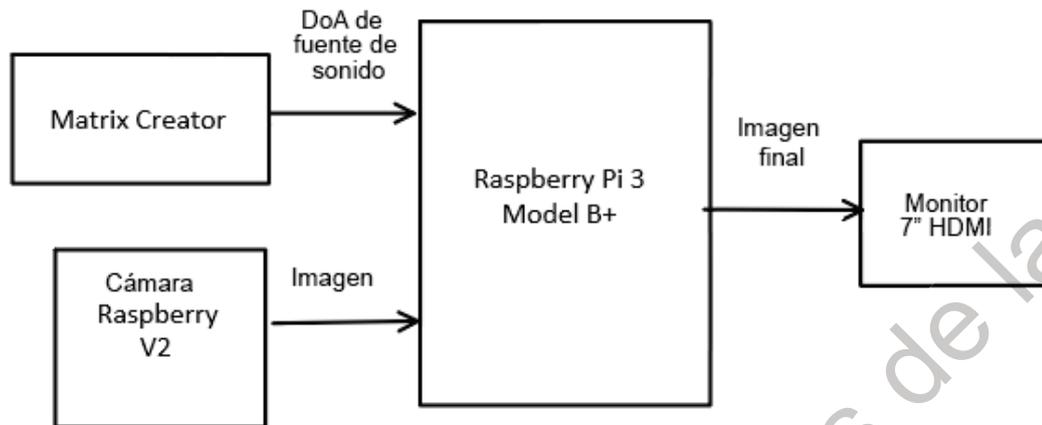


Figura 3: Diagrama de conexiones

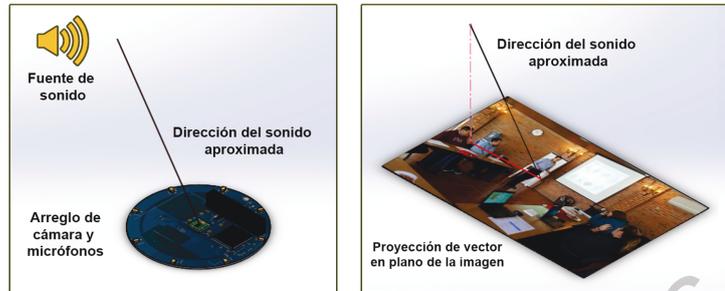
nos y un FPGA para el procesamiento de señales; un módulo de la cámara raspberry pi V2 y un monitor de 7 pulgadas (ver Figura 3).

#### 4.3.3. Funcionamiento del dispositivo

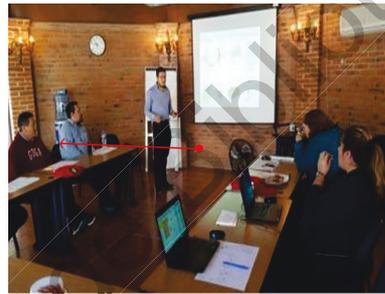
En esta sección, se describe el funcionamiento del sistema (ver Figura 4) que permite la localización acústica y su representación en un entorno de realidad mixta.

En primer lugar, la señal acústica es capturada por los micrófonos a una frecuencia de 16000 Hz, y luego el ángulo de llegada de la onda se calcula utilizando el algoritmo de formación de haces “Sum-and-Delay”, descrito en la Sección 2. A continuación, la dirección del ángulo de llegada se envía a la Raspberry Pi a través del puerto GPIO. El algoritmo propuesto toma la señal de salida de la cámara y crea un plano en el que se proyecta el ángulo de la fuente de sonido. Por último, la imagen resultante del algoritmo se envía

al monitor para mostrar una aproximación de la localización de la fuente de sonido en un entorno de realidad mixta.



(a) Procesamiento de señal y generación de vector (b) Proyección de vector en plano de imagen



(c) Aproximación del ángulo del llegada del sonido

Figura 4: Funcionamiento del sistema

Es importante mencionar que para tener congruencia entre el ángulo del origen del sonido y la imagen mostrada, el centro de la matriz del micrófono, la lente de la cámara y la pantalla del monitor deben ser coincidentes (ver Figura 5). Durante la elaboración del prototipo, aseguramos la congruencia entre los centros de los componentes principales realizando mediciones geométricas antes de ensamblarlos.



Figura 5: Alineación de centro

#### 4.4. Fase de validación

- Análisis de los resultados obtenidos

##### 4.4.1. Cálculo de errores

Antes de que se pueda realizar el cálculo del error es necesario obtener un ángulo aproximado promedio que se utilizará para las mediciones. Este valor se obtiene promediando los valores calculados por el sistema durante tres segundos cuando se está ejecutando. A continuación, este ángulo se proyecta y dibuja en la imagen final desde el centro del plano de imagen. Posteriormente, se dibuja manualmente un segundo ángulo desde el mismo centro hasta la fuente de sonido y se mide la diferencia entre los dos

valores (ver Figura 6). Una vez que se obtiene el error absoluto, se calcula el promedio medio de error porcentual (MAPE por sus siglas en inglés) con la siguiente ecuación (3):

$$MAPE = \sum_{i=1}^n \frac{|Expectedvalue_i - Realvalue_i|}{|Expectedvalue_i|} \quad (3)$$



Figura 6: Medición de error absoluto

## 5 Resultados

Se realizaron mediciones en una sala de 4 x 6 metros a distancias de 3, 2 y 1,5 m de la fuente de sonido con bajo ruido ambiental y con un altavoz como fuente de sonido. El método demostró ser capaz de representar la aproximación de una fuente de audio en un entorno de realidad mixta utilizando un dispositivo integrado y teniendo resultados con un error medio del 2,5% de trabajo mejor para distancias de menos de 3 metros (ver Tabla 5). Una restricción del sistema es que el dispositivo presenta problemas

al localizar fuentes de sonido con distancias superiores a 4 metros porque no puede resolver el ángulo de dirección de llegada y no puede generar el vector. También se observó un alto porcentaje de error en las aproximaciones de los ángulos 0-90 (lado izquierdo) contra las medidas de los ángulos 91-180 (lado derecho), causado por un espacio entre los centros de la matriz del micrófono, lente de la cámara y la pantalla del monitor. Las posibles soluciones para resolver este problema son diseñar un método más robusto o implementar un sistema de calibración. Sin embargo, el sistema también tiene ventajas significativas como estar basado en un dispositivo integrado de bajo costo, es portátil y puede seguir mejorando mediante la implementación de otros tipos de algoritmos que pueden agregar nuevas funcionalidades como el procesamiento de imágenes o voz, servicios de procesamiento, entre otras.

## **6 Conclusiones y trabajo futuro**

El método demostró tener bajas tasas de error y la capacidad de seguir mejorando, aprovechando la versatilidad de los sistemas embebidos.

El trabajo establece las bases para aplicaciones basadas en la localización de fuentes de sonido y la realidad mixta utilizando un dispositivo embebido. Un ejemplo de esto es la creación de un asistente de conversación de realidad mixta donde se identifique a una persona que está hablando y posteriormente utilizar un servicio de conversión de voz a texto, para así

Distancia	Ángulo aproximado	ángulo real	Error absoluto	Error %
3m	139°	136°	3°	2.16 %
	36°	32°	4°	11.11 %
2m	129°	128°	1°	0.78 %
	26°	26°	1°	0.0 %
1.5m	139°	137°	2°	1.44 %
	139°	139.5°	0.5°	0.36 %
	36°	34°	2°	5.56 %
	36°	36°	0°	0.00 %
<b>MAPE</b>				2.51 %

Tabla 1: Resultados de las pruebas

mostrar en realidad mixta de qué se está diciendo utilizando subtítulos en tiempo real. De esta manera, las personas con sordera podrían ser asistidas en aulas que no tienen traductor. De igual forma, se puede incorporar un servicio de traducción para entender a los hablantes de otros idiomas.

## Referencias

- Acero, A., y Stern, R. M. (1990). Environmental robustness in automatic speech recognition. En *International conference on acoustics, speech, and signal processing* (pp. 849–852).
- Azar, J., Saleh, H. A., y Al-Alaoui, M. (2007). Sound visualization for the hearing impaired. *International Journal of Emerging Technologies in Learning (iJET)*, 2(1).
- Carranza Alva, C. A., Toral Martiñón, R., Shkurovich Bialik, P., y Schabes Rostenberg, M. (2016). Resultados del diagnóstico temprano de sordera en recién nacidos sin factores de riesgo. *Anales Médicos de la Asociación Médica del Centro Médico ABC*, 61(2), 93–97.
- de Benito Crosetti, B., y Ibáñez, J. M. S. (2016). La investigación basada en diseño en tecnología educativa. *Revista Interuniversitaria de Investigación en Tecnología Educativa*.
- Dong, B., Antoni, J., Pereira, A., y Kellermann, W. (2016). Blind separation of incoherent and spatially disjoint sound sources. *Journal of Sound and Vibration*, 383, 414–445.
- Greensted, J. (s.f.). *Delay sum beamforming*. Descargado 2019-07-22, de <http://www.labbookpages.co.uk/audio/beamforming/delaySum.html>
- Huang, Y., Benesty, J., y Elko, G. W. (2000). Passive acoustic source locali-

- zation for video camera steering. En *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00ch37100)* (Vol. 2, pp. 11909–11912).
- Huang, Z., Hui, P., Peylo, C., y Chatzopoulos, D. (2013). Mobile augmented reality survey: a bottom-up approach. *arXiv preprint arXiv:1309.4413*.
- Jain, D., Findlater, L., Gilkeson, J., Holland, B., Duraiswami, R., Zotkin, D., ... Froehlich, J. E. (2015). Head-mounted display visualizations to support sound awareness for the deaf and hard of hearing. En *Proceedings of the 33rd annual ACM conference on human factors in computing systems* (pp. 241–250).
- Krishnaveni, V., Kesavamurthy, T., y Aparna, B. (2013). Beamforming for direction-of-arrival (doa) estimation-a survey. *International Journal of Computer Applications*, 61(11).
- Kundu, T., Nakatani, H., y Takeda, N. (2012). Acoustic source localization in anisotropic plates. *Ultrasonics*, 52(6), 740–746.
- Lanslots, J., Deblauwe, F., y Janssens, K. (2010). Selecting sound source localization techniques for industrial applications. *Sound and Vibration*, 44(6), 6.
- Michel, U., y cols. (2006). History of acoustic beamforming. En *Berlin beamforming conference, berlin, germany, nov* (pp. 21–22).
- Noor, A. K. (2016). The hololens revolution. *Mechanical Engineering Magazine Select Articles*, 138(10), 30–35.

- Pessentheiner, H., Petrik, S., y Romsdorfer, H. (2012). Beamforming using uniform circular arrays for distant speech recognition in reverberant environments and double talk scenarios. En *Thirteenth annual conference of the international speech communication association*.
- Rascon, C., y Meza, I. (2017). Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96, 184–210.
- Stone, S. A., y Tata, M. S. (2017). Rendering visual events as sounds: Spatial attention capture by auditory augmented reality. *PloS one*, 12(8), e0182635.
- Wang, H., y Chu, P. (1997). Voice source localization for automatic camera pointing system in videoconferencing. En *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 1, pp. 187–190).
- Zhong, X., Sun, L., y Yost, W. (2016). Active binaural localization of multiple sound sources. *Robotics and Autonomous Systems*, 85, 83–92.

# ANEXOS

## A Anexo: Código fuente

```
#include <json.h>
#include <math.h>
#include <matrix_hal/everloop.h>
#include <matrix_hal/everloop_image.h>
#include <matrix_hal/matrixio_bus.h>
#include <netinet/in.h>
#include <string.h>
#include <sys/socket.h>
#include <array>
#include <iostream>

namespace hal = matrix_hal;

// ENERGY_COUNT : Number of sound energy slots to maintain.
#define ENERGY_COUNT 36

// MAX_VALUE : controls smoothness
#define MAX_VALUE 200

// INCREMENT : controls sensitivity
#define INCREMENT 20

// DECREMENT : controls delay in the dimming
#define DECREMENT 1

// MAX_BRIGHTNESS: Filters out low energy
#define MIN_THRESHOLD 10

// MAX_BRIGHTNESS: 0 – 255
#define MAX_BRIGHTNESS 50

double x, y, z, E;

int energy_array[ENERGY_COUNT];
```

```

const double leds_angle_mcreator[35] = {
    170, 159, 149, 139, 129, 118, 108, 98, 87, 77, 67, 57,
    46, 36, 26, 15, 5, 355, 345, 334, 324, 314, 303, 293,
    283, 273, 262, 252, 242, 231, 221, 211, 201, 190, 180};

const double led_angles_mvoice[18] = {170, 150, 130, 110, 90, 70,
    50, 30, 10, 350, 330, 310,
    290, 270, 250, 230, 210, 190};

void increase_pots() {
    // Convert x,y to angle. TODO: See why x axis from ODAS is inverted
    printf("Valor x = %.10f\n", x);
    printf("Valor y = %.10f\n", y);
    printf("Valor z = %.10f\n", z);
    printf("Valor atan2(y,x) = %.10f\n", atan2(y, x));
    printf("Valor M_PI = %.10f\n", M_PI);
    printf("Valor 180.0 / M_PI = %.10f\n", 180.0 / M_PI);
    printf("Valor (180.0 / M_PI) + 360 = %.10f\n", 180.0 / M_PI);
    printf("Valor atan2 * 180/pi +360 = %.10f\n", atan2(y, x) * (180.0 / M_PI) + 360);
    printf("Valor modulo resultado= %.10f\n", fmodf((atan2(y, x) * (180.0 / M_PI)) + 360, 360));
    printf("Valor energy_count = %.10f\n", ENERGY_COUNT);
    printf("Valor i_angle = %.10f\n", angle_xy / 360 * ENERGY_COUNT);
    printf("Valor E = %.10f\n", E);

    double angle_xy = fmodf((atan2(y, x) * (180.0 / M_PI)) + 360, 360);

    // Convert angle to index
    int i_angle = angle_xy / 360 * ENERGY_COUNT; // convert degrees to index

    // Set energy for this angle
    energy_array[i_angle] += INCREMENT * E;

    // Set limit at MAX.VALUE
    energy_array[i_angle] =

```

```
        energy_array[i-angle] > MAX.VALUE ? MAX.VALUE : energy_array[i-angle];
    }
```

```
void decrease_pots() {
    for (int i = 0; i < ENERGY.COUNT; i++) {
        energy_array[i] -= (energy_array[i] > 0) ? DECREMENT : 0;
    }
}
```

```
void json_parse_array(json_object *jobj, char *key) {
    // Forward Declaration
    void json_parse(json_object * jobj);
    enum json_type type;
    json_object *jarray = jobj;
    if (key) {
        if (json_object_object_get_ex(jobj, key, &jarray) == false) {
            printf("Error parsing json object\n");
            return;
        }
    }
}
```

```
int arraylen = json_object_array_length(jarray);
int i;
json_object *jvalue;
```

```
for (i = 0; i < arraylen; i++) {
    jvalue = json_object_array_get_idx(jarray, i);
    type = json_object_get_type(jvalue);
```

```
    if (type == json_type_array) {
        json_parse_array(jvalue, NULL);
    } else if (type != json_type_object) {
    } else {
```

```

        json_parse(jvalue);
    }
}
}

```

```

void json_parse(json_object *jobj) {
    enum json_type type;
    unsigned int count = 0;
    decrease_pots();
    json_object_object_foreach(jobj, key, val) {
        type = json_object_get_type(val);
        switch (type) {
            case json_type_boolean:
                break;
            case json_type_double:
                if (!strcmp(key, "x")) {
                    x = json_object_get_double(val);
                } else if (!strcmp(key, "y")) {
                    y = json_object_get_double(val);
                } else if (!strcmp(key, "z")) {
                    z = json_object_get_double(val);
                } else if (!strcmp(key, "E")) {
                    E = json_object_get_double(val);
                }
                increase_pots();
                count++;
                break;
            case json_type_int:
                break;
            case json_type_string:
                break;
            case json_type_object:
                if (json_object_object_get_ex(jobj, key, &jobj) == false) {

```

```

        printf("Error parsing json object\n");
        return;
    }
    json_parse(jobj);
    break;
case json_type_array:
    json_parse_array(jobj, key);
    break;
}
}
}

```

```

int main(int argc, char *argv[]) {
    // Everloop Initialization
    hal::MatrixIOBus bus;
    if (!bus.Init()) return false;
    hal::EverloopImage image1d(bus.MatrixLeds());
    hal::Everloop everloop;
    everloop.Setup(&bus);

    // Clear all LEDs
    for (hal::LedValue &led : image1d.leds) {
        led.red = 0;
        led.green = 0;
        led.blue = 0;
        led.white = 0;
    }
    everloop.Write(&image1d);

    char verbose = 0x00;

    int server_id;
    struct sockaddr_in server_address;

```

```

int connection_id;
char *message;
int messageSize;

int c;
unsigned int portNumber = 9001;
const unsigned int nBytes = 10240;

server_id = socket(AF_INET, SOCK_STREAM, 0);

server_address.sin_family = AF_INET;
server_address.sin_addr.s_addr = htonl(INADDR_ANY);
server_address.sin_port = htons(portNumber);

printf("Binding socket..... ");
fflush(stdout);
bind(server_id, (struct sockaddr *)&server_address, sizeof(server_address));
printf("[OK]\n");

printf("Listening socket..... ");
fflush(stdout);
listen(server_id, 1);
printf("[OK]\n");

printf("Waiting for connection in port %d ... ", portNumber);
fflush(stdout);
connection_id = accept(server_id, (struct sockaddr *)NULL, NULL);
printf("[OK]\n");

message = (char *)malloc(sizeof(char) * nBytes);

printf("Receiving data..... \n\n");

```

```

while ((messageSize = recv(connection_id, message, nBytes, 0)) > 0) {
    message[messageSize] = 0x00;

    // printf("message: %s\n\n", message);
    json_object *jobj = json_tokener_parse(message);
    json_parse(jobj);

    for (int i = 0; i < bus.MatrixLeds(); i++) {
        // led index to angle
        int led_angle = bus.MatrixName() == hal::kMatrixCreator
            ? leds_angle_mcreator[i]
            : led_angles_mvoice[i];

        // Convert from angle to pots index
        int index_pots = led_angle * ENERGY_COUNT / 360;
        // Mapping from pots values to color
        int color = energy_array[index_pots] * MAX_BRIGHTNESS / MAX_VALUE;
        // Removing colors below the threshold
        color = (color < MIN_THRESHOLD) ? 0 : color;

        image1d.leds[i].red = color;
        image1d.leds[i].green = 0;
        image1d.leds[i].blue = 0;
        image1d.leds[i].white = 0;
    }
    everloop.Write(&image1d);
}
}
}

```

## B Anexo: Hoja de datos Matrix creator

### Especificaciones de Hardware:

- Microcontroller : Atmel ATSAM3S2C Cortex

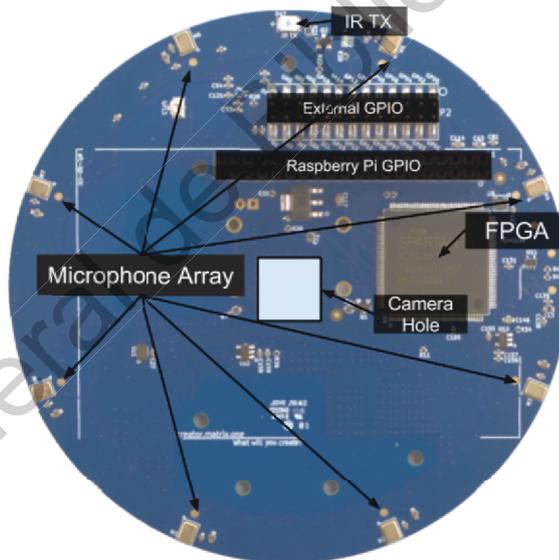
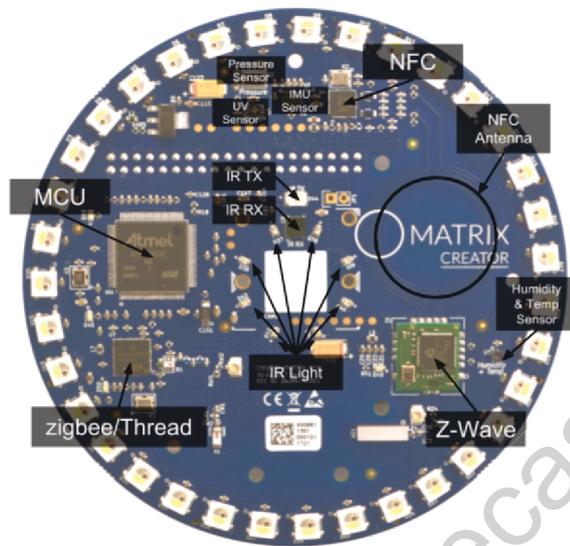


Figura 7: Matrix creator

- FPGA : Xilinx Spartan 6 FPGA XC6SLX4
- Wireless
  - zigbee/Thread : Integrated zigbee/802.15.4

- System-on-Chip EM3588
- Z-Wave : General Purpose Z-Wave® Module ZM5202
- NFC : Full NFC Forum-compliant frontend PN512
- Sensors
  - UVA Light Sensor VEML6070
  - Precision pressure sensor with altimetry
  - MPL3115A2
  - Humidity and temperature sensor HTS221
  - 3D accelerometer, 3D gyroscope, 3D magnetometer LSM9DS1
- Additional features
  - 35 RGBW LEDs
  - IR Receiver Module TSOP57X
  - IR Emitter Diode APT1608SF4C-PRV
  - 8 MEMS audio sensor MP34DB02
- I/O
  - 2 x ADC channels.
  - 17 external GPIO. These are connected to FPGA so they could potentially implement any digital interface e.g PWM, Servo, UART, I2C etc.

- SPI
- I2C

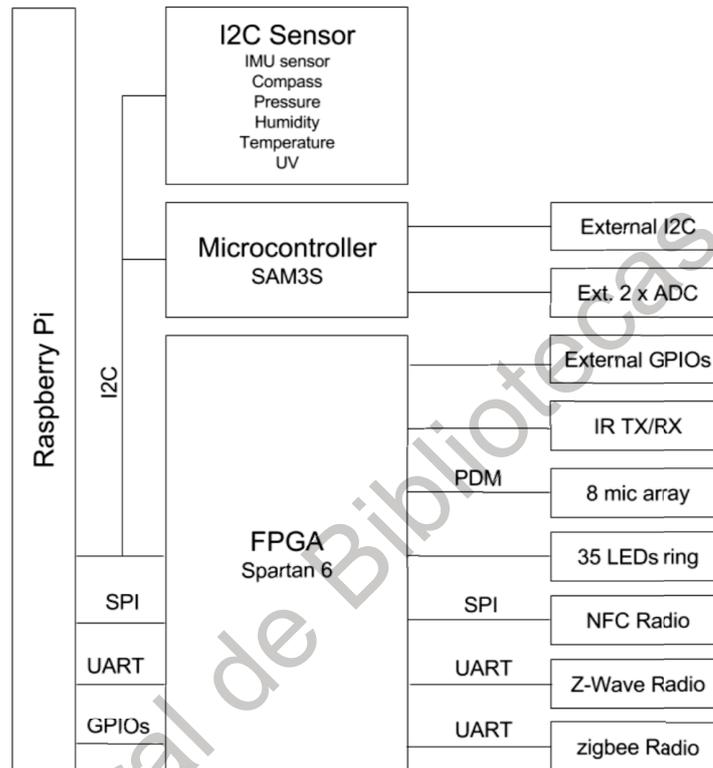


Figura 8: Diagrama de bloques

## C Anexo: Experimentos previos

La propuesta inicial del prototipo estaba basada en un celular android con un casco de realidad virtual y una matriz de micrófonos (ver Figura 9).



Figura 9: Concepto prototipo

Esta propuesta consideraba el uso de un celular como unidad de procesamiento y salida de video al montarse sobre el casco de realidad virtual, la finalidad era mostrar una animación de las ondas de sonido registradas por los micrófonos.

Sin embargo, para realizar el procesamiento de la señal y conexión de los micrófonos externos se requería de hardware externo que ocupaba su propio circuito para limpiar la señal y conexión al celular. Por tal motivo, se optó por cambiar a una matriz de micrófonos comercial.