



UNIVERSIDAD AUTÓNOMA DE YUCATÁN

FACULTAD DE MATEMÁTICAS
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

DETECCIÓN AUTOMÁTICA DE LADRIDOS EN
SISTEMAS EMBEBIDOS PARA LA ASISTENCIA A
PERSONAS CON DISCAPACIDAD AUDITIVA

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

ING. TAKU DANIEL UTSUKI ALEXANDER

DIRECTORES DE TESIS:

DR. JORGE ALBERTO RÍOS MARTÍNEZ Y DR. FRANCISCO ALEJANDRO
MADERA RAMÍREZ

MÉRIDA, YUCATÁN

SEPTIEMBRE, 2019

**Detección automática de ladridos en sistemas
embebidos para la asistencia a personas con
discapacidad auditiva**

por

Ing. Taku Daniel Utsuki Alexander

Tesis presentada para obtener el grado de

Maestro en Ciencias de la Computación

en la

FACULTAD DE MATEMÁTICAS

UNIVERSIDAD AUTÓNOMA DE YUCATÁN

Mérida, Yucatán. septiembre, 2019

Agradecimientos

A mis padres y amigos, que me han acompañado y apoyado en cada decisión y en cada paso que he tomado y que me han permitido alcanzar este logro y muchos más.

Al Dr. Jorge Rios, al Dr. Francisco Madera y al Dr. Humberto Pérez por la guía, el apoyo y los consejos que han permitido la consolidación de este trabajo.

A CONACYT por su apoyo económico para el desarrollo de este trabajo.

A la Asociación Mexicana para la Comunicación y Superación de Personas con Discapacidad Auditiva, por su tiempo y apoyo para el desarrollo del prototipo.

Índice general

1. Introducción	1
1.0.1. Objetivos	4
2. Marco teórico	6
2.1. Personas con discapacidad auditiva	6
2.2. Diseño universal y Usabilidad	8
2.2.1. Similitud entre Diseño Universal y Usabilidad	9
2.3. Ladridos de perros	10
2.3.1. Características de un asistente inteligente.	11
2.3.2. Aprendizaje de máquina y técnica SVM.	12
3. Antecedentes	14
3.1. AAL para personas con discapacidad	14
3.2. Trabajos relacionados	16
3.2.1. Asistencia a PcDA	17
3.2.2. Reconocimiento de ladridos de perros	18
3.2.3. Otros desarrollos: SOPARE	20
3.2.4. Recapitulación de los trabajos encontrados	24
4. Metodología	25
4.1. Introducción	25
4.2. AMCIAAL	25

4.2.1.	Captura y Análisis de requerimientos	26
4.2.2.	Diseño de IUIs (Intelligent User Interfaces)	29
4.2.3.	Implementación	31
4.2.4.	Pruebas	32
4.3.	Obtención de los datos	33
4.4.	Base de datos para la evaluación de las librerías	34
4.5.	Procesos para la implementación final	35
4.6.	Uso de PyAudio.	35
4.7.	Bibliotecas de procesamiento de audio	36
4.7.1.	Notificaciones	38
4.8.	Extracción de características	40
5.	Implementación	45
5.1.	Prototipo	45
5.2.	Diseño del prototipo	46
5.3.	Modelo SVM	48
5.4.	Diagrama para la obtención del modelo SVM	48
5.4.1.	Parámetros C y g	49
5.4.2.	Ajustes en el modelo	49
5.5.	Análisis del aprendizaje de máquina	55
5.5.1.	Curva ROC (Receiver Operating Characteristics)	55
5.5.2.	Matriz de confusión	56
5.6.	Ajustes finales	57
5.7.	Pruebas al prototipo	59
5.7.1.	Resultados	60
6.	Conclusiones y trabajo futuro	64
6.1.	Conclusiones	64
6.2.	Trabajo a futuro	66

Capítulo 1

Introducción

Las personas con discapacidad, independientemente de la que sea, generalmente se enfrentan a problemas de discriminación a través de obstáculos tales como: barreras sociales (lenguaje ofensivo, desconocimiento sobre el tema, sobreprotección, segregación, etc.), políticas (falta de políticas públicas adecuadas y suficientes en las cuales se respeten y defiendan sus derechos como los de cualquier otra persona), religiosas (discriminación, creencias erróneas sobre el origen y la discapacidad), pero sobre todo barreras arquitectónicas (lugares sin las adecuaciones arquitectónicas o tecnológicas mínimas necesarias para la equiparación del libre desplazamiento y desarrollo de las personas). A este respecto los apoyos tecnológicos siempre han sido de gran utilidad para lograr disminuir los diferentes obstáculos a los que todos los días se enfrentan las personas con diferentes discapacidades. Las personas con discapacidad auditiva, según la OMS, a nivel mundial son el 5% ¹ de la población; en México 12.1% son sordos lo que significa que 2 de cada mil nacimientos son sordos según el INEGI ², y en Yucatán tenemos una mayor población de sordos es decir 4 de cada mil niños nace con sordera, inclusive existe un pueblo llamado Chicán que pertenece a la comisaría de Tixméhuac, en donde hay muchas familias completas de personas sordas. A pesar de ello la discapacidad auditiva también

¹www.who.int

²www.inegi.org.mx

se conoce como discapacidad invisible, ya que si vemos a una persona con su bastón o con un perro guía, sabemos que es una persona ciega o débil visual, si vemos a una persona en silla de ruedas, con bastones o muletas sabemos que tiene una discapacidad motriz, las gesticulaciones o características del rostro en ocasiones identifican a una persona con discapacidad intelectual, pero a simple vista no podemos saber que la persona con discapacidad auditiva no escucha aunque tenga auxiliares auditivos, es probable que tenga una gorra o que el pelo tape sus orejas y no podremos ver los auxiliares, por eso es que al no poder identificar de manera visual o inmediata la discapacidad es que también se llama discapacidad invisible; de igual forma es conveniente saber que la población sorda no siempre es alfabeta ya que, según la encuesta Nacional de la Dinámica Demográfica³ en 2014, solo el 64 % de la población sorda menor de 14 años ha acudido a un centro educativo.

Los auxiliares auditivos son en la mayoría de los casos un apoyo indispensable para poder desarrollarse, estos dispositivos han evolucionado tecnológicamente desde la trompetilla o en 1898 el Akouphone que fue el primer auxiliar auditivo eléctrico; posteriormente los digitales como el BTE, (Behind The Ear, que en español significa, detrás de la oreja) o también llamado retroauricular, evolucionando al intracanal, intraauricular y en la actualidad el intra CIC (Completely In the Canal), que son auxiliares diminutos que se colocan en el conducto auditivo o en algunos casos el implante coclear. De igual forma existen otro tipo de apoyos tecnológicos como el sistema FM (Frecuencia Modulada) el cual emite señales que van de un transmisor de la persona hablante a un receptor en la persona sorda, sistemas de vibración, sistemas para la traducción de Lenguaje de Señas Mexicanas (LSM) a un lenguaje audible y viceversa. Lastimosamente los apoyos tecnológicos casi siempre están lejos del alcance de la mayoría de las personas, ya que su costo es muy elevado. En la actualidad con el desarrollo de la comunicación móvil podemos ver que tanto para la población con discapacidad o sin ella los móviles inteligentes son un gran apoyo con sus diferentes aplicaciones que permiten poner mensajes y grabar

³www.inegi.org.mx

imágenes para mandarlas en tiempo real, no obstante las personas sordas casi siempre dependen de otras para algunas actividades cotidianas, que a las personas no sordas les pasan desapercibidas, tales como saber si alguien toca la puerta estando en casa, escuchar si algo se rompió como podría ser un vidrio, si silba la tetera, si alguien grita o si se dejó abierto el grifo del agua por error. El no escuchar algunos de estos sonidos podría significar un riesgo; por eso la mayoría de las personas sordas no viven solas, o temen estar solas inclusive en su propia casa. Además, las personas con discapacidad auditiva por lo general no cuentan con un empleo que les permita contar con ingresos suficientes para adquirir tecnología que le ayude a independizarse como los asistentes inteligentes. En este trabajo nos enfocamos a la parte de la población con discapacidad auditiva que le gusta tener como mascota a un perro, el cual tiene una gran capacidad auditiva y una gran empatía con el ser humano; estas mascotas sin necesidad de tener un entrenamiento específico previo, siempre tratan de avisarnos de algún ruido fuera de lo común o de algún peligro a través de sus ladridos, Faragó *et al.* [2010] y Pongrácz *et al.* [2010].

Las tecnologías avanzan cada día para proporcionar a las personas con discapacidad mayor comodidad, seguridad e independencia en sus actividades cotidianas. La vida asistida por el entorno (AAL por sus siglas en inglés) (Monekosso *et al.* [2015]) aplica muchas de esas tecnologías con el objetivo de identificar actividades y reconocer comportamientos de los humanos que están siendo asistidos en un determinado espacio; este concepto está relacionado con el desarrollo de ambientes inteligentes: “es la visión de un futuro en la que el ambiente asiste a las personas que lo habitan. Este entorno previsto es discreto, interconectado, adaptable, dinámico, embebido e inteligente. En esta visión, las entradas y salidas tradicionales de un computador desaparecerán. En su lugar procesadores y sensores estarán integrados en objetos cotidianos” (Sadri [2011]). Un componente importante de un ambiente inteligente (AmI) para tareas de AAL es su capacidad para reconocer el contexto a partir de los sensores instalados en él, así el sistema podrá asociar a dicho contexto la actividad detectada en el usuario y ejecutar una acción de asistencia predeter-

minada, por ejemplo, enviar una señal de alerta. La mayoría de los sistemas desarrollados para dar asistencia tienen dificultades para asistir a personas con discapacidad auditiva (PcDA) puesto que se diseñaron principalmente para dar asistencia a personas con discapacidad motriz o visual (Utsuki [2016]), o bien se debe a que la discapacidad auditiva es considerada como discapacidad invisible porque no es perceptible por los demás.

En este trabajo se desarrolló un prototipo de dispositivo con capacidades para ser integrado en un AmI para AAL que atienda a PcDA en el caso de la detección de ladridos de perro. Para este trabajo previamente se realizó una encuesta a un grupo de 32 personas con la discapacidad mencionada para saber qué tipo de situaciones les sucedían frecuentemente al no poder percibir los sonidos de manera adecuada y que podrían representar algún riesgo, mismo que se explicará a detalle en el Capítulo 4. En tal encuesta se obtuvo que la situación más frecuente y que los participantes mencionaron como riesgo fue la de no escuchar los ladridos de sus perros, esta tesis tuvo como caso particular de asistencia detectar ladridos de perros y notificarlos a PcDA por medio de un sistema inteligente que simule un AAL.

Para el diseño se consideraron los puntos siguientes, la necesidad de saber cuándo hay algún sonido de alerta en casa, costo final del dispositivo, falta de capacidad de lectura en las PcDA, compañía de mascotas, capacidad auditiva de los perros y su facilidad de adaptarse al ser humano, familiaridad con el uso de móviles inteligentes y sus aplicaciones, de tal forma que el binomio de un hardware especializado y un teléfono inteligente, hará conjunto con un perro (no necesariamente estar entrenados ni de alguna raza en particular) y esto dotará de manera económica a la PcDA de la capacidad de poder detectar cuando se produce un ladrido de alerta.

1.0.1. Objetivos

Objetivo general.

Desarrollar un dispositivo que permita reconocer ladridos de un perro de manera automática y enviar notificación por medio de un dispositivo móvil, para la asistencia a

personas con discapacidad auditiva.

Objetivos específicos.

- Realizar la revisión de algoritmos de aprendizaje de máquina para la detección automática de ladridos de perros.
- Entrenar un modelo para la detección de ladridos de perro usando la técnica SVM.
- Implementación de un sistema para la obtención de audios por medio del micrófono ReSpeaker, la predicción en una tarjeta Raspberry Pi 3 y notificación de la detección del ladrido a un dispositivo móvil.
- Mejorar la precisión de un algoritmo seleccionado de aprendizaje de máquina.
- Evaluar la aceptación del dispositivo por parte de usuarios con discapacidad auditiva.

Capítulo 2

Marco teórico

En este capítulo se describirán de manera general los principales conceptos utilizados en este trabajo.

2.1. Personas con discapacidad auditiva

El Glosario de Términos sobre discapacidad de la comisión de política gubernamental en materia de derechos humanos dice que la discapacidad auditiva es la restricción en la función de la percepción de los sonidos externos; cuando es de superficial a moderada, se necesita el uso de auxiliares auditivos, pero pueden adquirir la lengua oral a través de la retroalimentación de información que reciben por la vía auditiva. Cuando la pérdida auditiva no es funcional para la vida diaria, la adquisición de la lengua oral no se da de manera natural, es por ello que utilizan la visión como principal vía de entrada de la información para aprender y para comunicarse, por lo que en la mayoría de las veces la lengua natural de las personas con esta condición es la Lengua de Señas Mexicana. Cabe agregar que no todas las sorderas son iguales y que se clasifican por varios factores según el INCH (Instituto Nacional de Comunicación Humana).

- El momento de la aparición de la lesión: Sordera prelocutiva, Sordera perilocutiva y Sordera postlocutiva.

- El grado de pérdida: Audición normal (umbral de 0 a 20 dB), Pérdida leve o ligera (umbral de 20 a 40 dB), Pérdida media o moderada (umbral de 40 a 70 dB), Pérdida auditiva severa (umbral de 70 a 90 dB), Pérdida profunda (umbral mayor a los 90 dB) y Cofosis (pérdida total).
- La localización de la lesión: Sorderas de transmisión o conductivas, Sorderas neurossensoriales o de percepciónn y Sorderas mixtas.

Ahora para la medición de una pérdida auditiva se utilizan los dos parámetros con los que se mide el sonido: frecuencia e intensidad. La frecuencia indica los tonos (de graves o bajas, en torno a 250 Hz, a agudos o altos, hasta 8,000Hz). El oído humano se mueve de 0 dB hasta 120 dB (sonido muy alto). En la Figura 2-1, se detalla el nivel de pérdida auditiva.

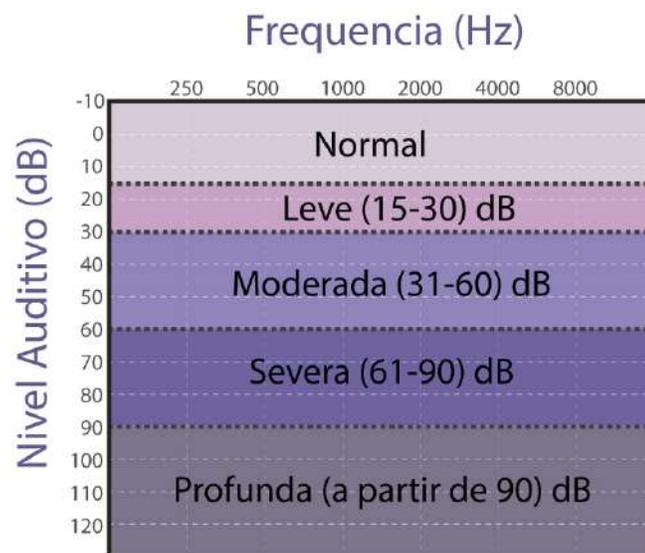


Figura 2-1: Audiometría que muestra el nivel de pérdida auditiva.

Diseño Universal	Usabilidad
<ul style="list-style-type: none"> • Igualdad de uso <ul style="list-style-type: none"> • Flexibilidad • Simple e intuitivo • Información comprensible <ul style="list-style-type: none"> • Tolerante a errores • Escaso esfuerzo físico • Dimensiones apropiadas 	<ul style="list-style-type: none"> • Visibilidad del estado del sistema • Adecuación entre el sistema y el mundo real <ul style="list-style-type: none"> • Control y libertad del usuario • Consistencia y estándares <ul style="list-style-type: none"> • Prevención de errores • Reconocer mejor que recordar • Flexibilidad y eficiencia de uso • Estética y diseño minimalista • Ayudar a los usuarios a reconocer, diagnosticar y solucionar errores <ul style="list-style-type: none"> • Ayuda y documentación

Tabla 2-1: Comparativa entre Diseño Universal y Usabilidad.

2.2. Diseño universal y Usabilidad

El concepto de diseño universal “busca estimular el desarrollo de productos atractivos y comerciales que sean utilizables por cualquier tipo de persona. Está orientado al diseño de soluciones ligadas a la construcción y al de objetos que responden a las necesidades de una amplia gama de usuarios.” (Wiehr *et al.* [2016]), es decir todos los objetos, sistemas, etc, deben poder ser utilizadas por cualquier persona sin importar su condición física, lo que convierte en un reto en desarrollar un hogar inteligente más eficiente y accesible (Utsuki [2016]).

El diseño universal comparte los principios de la usabilidad; en la Tabla 2-1, se muestra una comparativa de los principios entre el diseño universal propuesto por Ronald Mace¹ y la usabilidad propuesto por Jakob Nielsen (Nielsen [1994]).

Este concepto se relaciona con el prototipo, ya que se pretendió diseñarlo de acuerdo a las necesidades y comodidades de los usuarios.

¹https://projects.ncsu.edu/ncsu/design/cud/about_ud/udprinciples.htm

2.2.1. Similitud entre Diseño Universal y Usabilidad

A simple vista se percibe diferencia entre los principios de estos dos conceptos, sin embargo, si se analiza y se entiende el significado de cada uno de esos principios se puede encontrar una gran similitud, o dicho de otra forma, la usabilidad es el diseño universal de las tecnologías.

En la Tabla 2-2 y en la Tabla 2-3 se detalla el significado de cada principio tanto de los de Diseño Universal como de los de Usabilidad.

Igualdad de uso: Como se habla de la facilidad de manipular el producto y cuidar que puede ser utilizado por cualquier clase de usuario, en ese principio puede entrar el principio “Flexibilidad y eficiencia de uso” de Usabilidad.

Flexibilidad: Adaptarse a diversas situaciones para su uso, el principio “Flexibilidad y eficiencia de uso” es similar.

Simple e intuitivo: La facilidad de entender y suprimir informaciones innecesarias, entonces tiene relación con los principios “Control y libertad del usuario”, “Reconocer mejor que recordar”, “Estética y diseño minimalista” y “Ayudar a los usuarios a reconocer, diagnosticar y solucionar errores”.

Información comprensible: Tener una buena comunicación para que el usuario puede comprender de las acciones que realiza en el sistema, “Visibilidad del estado del sistema”, “Adecuación entre el sistema y el mundo real”, “Ayudar a los usuarios a reconocer, diagnosticar y solucionar errores” y “Ayuda y documentación” son los principios que se asemejan.

Tolerante a errores: Previene errores o el mismo sistema sea capaz de solucionar errores, un principio que comparte la misma idea con “Prevención de errores”.

Escaso esfuerzo físico: Un buen sistema hará que el usuario no tenga que realizar algún esfuerzo físico por lo que tiene similitud con “Ayudar a los usuarios a reconocer, diagnosticar y solucionar errores” ya que en el principio de usabilidad ofrece ayudar al usuario en todo lo posible para facilitar el manejo del sistema y no depender del mismo usuario.

Dimensiones apropiadas: Aunque no hay un principio de usabilidad similar, se puede usar este principio para el desarrollo del prototipo, proponiendo medidas adecuadas para su uso.

Diseño universal	
Principio	Definición breve
Igualdad de uso	Debe ser fácil de usar y adecuado para todas las personas sin importar sus capacidades y habilidades.
Flexibilidad	Tiene y se adapta a diversas alternativas de uso.
Simple e intuitivo	Debe ser fácil de entender, elimina la complejidad innecesaria.
Información comprensible	Capaz de intercambiar información con el usuario utilizando diversas formas de información (gráfica, verbal, etc.).
Tolerante a errores	Se reduce en lo posible todo aquello que pueda producir un error o un riesgo.
Escaso esfuerzo físico	Se evita el esfuerzo físico y las acciones repetitivas, además de poder ser usado eficazmente.
Dimensiones apropiadas	Tamaño y espacios apropiados para su uso.

Tabla 2-2: Principios de Diseño Universal

2.3. Ladridos de perros

La Real Academia Española, define al ladrido como, “una voz que emite con fuerza el perro, más o menos parecida a la onomatopeya guau”².

De acuerdo al estudio de Sakamoto *et al.* [2013], consiste en un audio que tiene una

²<https://dle.rae.es/?id=MmIcQI>

Usabilidad	
Principio	Definición breve
Visibilidad del estado del sistema	Siempre informar al usuario.
Adecuación entre el sistema y el mundo real	Debe poder comunicarse con el usuario por cualquier tipo de mensaje.
Control y libertad del usuario	Ofrecer al usuario una solución en determinadas situaciones de forma sencilla.
Consistencia y estándares	Tener claro la información a mostrar, sin malentendidos.
Prevención de errores	Evitar errores que tener varios mensajes de error.
Reconocer mejor que recordar	Ofrecer facilidad en el uso del sistema, sin que el usuario se esfuerce.
Flexibilidad y eficiencia de uso	Debe poder ser usado por cualquier persona.
Estética y diseño minimalista	Nunca mostrar información irrelevante.
Ayudar a los usuarios a reconocer, diagnosticar y solucionar errores	Los avisos y errores deben ser sencillos para el usuario para poder corregirlas.
Ayuda y documentación	Siempre tener a la mano una documentación de uso.

Tabla 2-3: Principios de Usabilidad.

alta frecuencia de entre 1000 - 2000 Hz y un nivel de sonido de aproximadamente 80 - 90 dB medidos a una distancia de 5 metros desde el perro. De acuerdo a Pérez-Espinosa *et al.* [2015], un ladrido se compone de un segmento de audio de entre 0.1 - 0.8 segundos, en la Figura 2-2 se presenta un gráfico en la que se puede observar la forma que tiene el sonido de un ladrido.

2.3.1. Características de un asistente inteligente.

Un asistente inteligente o bien un asistente virtual, es un agente de software que permite realizar tareas o servicios para asistir a un usuario, cuidando que sea lo menos

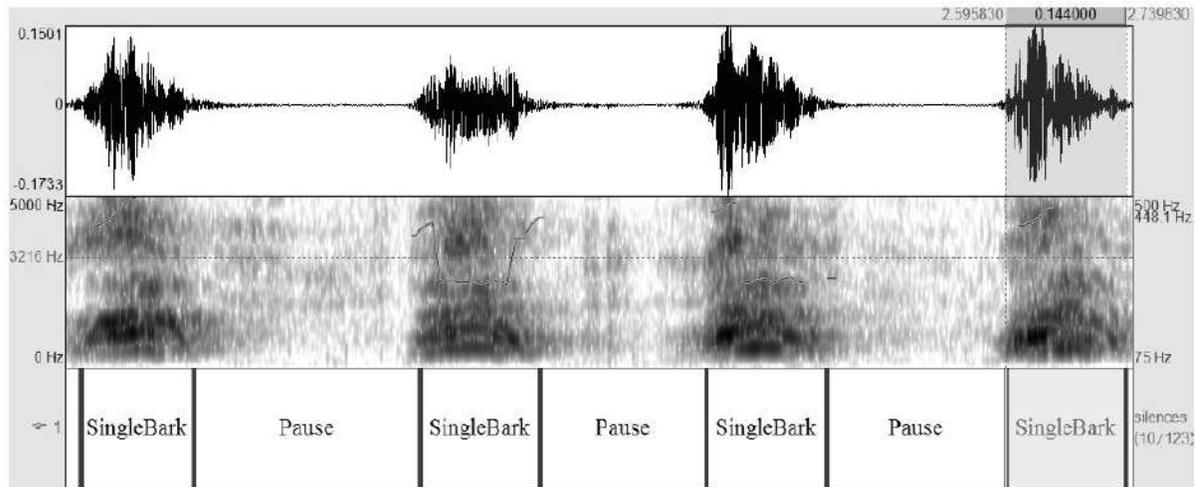


Figura 2-2: Ladridos en una grabación. Pérez-Espinosa *et al.* [2015]

intrusivo posible. La forma de pedir asistencia a esos dispositivos pueden ser por vía mensaje o por voz; actualmente Alexa de Amazon, Siri de Apple y Google Assistant son asistentes inteligentes conocidos mundialmente.

El prototipo de este trabajo escuchará el ruido del ambiente, en este caso ladridos, para poder asistir a la PcDA por medio de luces o por mensajes.

2.3.2. Aprendizaje de máquina y técnica SVM.

Un algoritmo de aprendizaje de máquina consiste en que las mismas computadoras puedan aprender a realizar algunas tareas por sí mismas; existen algunos tipos de aprendizaje, en este trabajo se trabajó con un algoritmo de aprendizaje supervisado, que consiste en entrenar al algoritmo con datos etiquetados de manera manual, mostrándole a la máquina cuál es la clase o categoría de cada muestra.

Uno de los algoritmos supervisados que se utilizó fue el SVM (Support Vector Machine), que consiste en una serie de algoritmos supervisados que permite clasificar dos o varias clases separados por uno o varios hiperplanos, de tal forma que esas separaciones tengan el mínimo error posible; además permite trabajar en problemas que pueden considerarse no-lineales, es decir, cuando la solución al problema no se puede resolver

con ecuaciones lineales (polinomial, radial y sigmoide), en este caso, existen se pueden utilizar las funciones Kernel que permite transformarlo en un problema lineal.

Otras características del algoritmo SVM, es que para separar los datos en sus respectivas clases hace uso de los parámetros de costo (C), que define que tan estricto se clasificarán los datos y de gamma (g), que determina la distancia entre puntos, o bien ajusta la curvatura de la línea de decisión.

Capítulo 3

Antecedentes

En este capítulo se presentan trabajos relacionados al tema de AAL o sistemas inteligentes que dan soporte a las PcDA. También se incluyen aquellos trabajos que se enfocaron en detectar ladridos de perros; en cada trabajo se revisaron a detalle las técnicas de aprendizaje de máquina propuestos para la ejecución de esas investigaciones.

3.1. AAL para personas con discapacidad

Se revisó en Google Scholar todos los artículos, proyectos y otros, donde el objetivo es la de dar asistencia a cualquier tipo de discapacidad, en este caso, se revisaron para personas con discapacidad auditiva, visual, intelectual y motora; partiendo desde el año 2012 hasta 2019. Con el fin de determinar la demanda en investigación que se tiene en cada tipo de discapacidad.

En la Tabla 3-1 se muestra las combinaciones de palabras clave escritas en la barra de búsqueda.

En cada búsqueda se revisó cada liga, ya sea revisando el resumen, el título y/o la introducción; posteriormente se eliminan aquellos que no dan asistencia en específico a un tipo de discapacidad o libros y/o artículos muy generales. En este caso, como esta tesis se enfoca en PcDA, se revisó a detalle cada liga para encontrar aquellos que sí dan

Clave principal	Clave objetivo	Se busca resultados donde den asistencia a Personas con discapacidad:
'Ambient assisted living'	'Visual impairment'	Visual
	'Blindness'	
	'Vision loss'	
	'Deaf'	Auditiva
	'Hearing loss'	
	'Mobility problems'	Motora
	'Mobility limitations'	
	'Physical disability'	
'Intellectual disability'	Intelectual	
'Mental illness'		

Tabla 3-1: Búsquedas realizadas.

asistencia a ese tipo de discapacidad.

En la Tabla 3-2, se muestran los resultados de las búsquedas en el período 2012 – 2018.

En cada columna debajo de los años representan el total de proyectos realizados para asistir al tipo de discapacidad correspondiente; la columna “Total” representa el total de cada fila, es decir de cada palabra clave y la última columna representa la suma de los números de la columna “Total” para cada tipo de discapacidad. Como se podrá notar se utilizaron palabras claves en inglés, esto es debido, a que se quiere conocer la cantidad de investigaciones sobre esas discapacidad que existen en el mundo y no sólo en países de habla hispana; además en varias universidades latinas se realizan publicaciones en inglés.

De acuerdo al resultado de la tabla anterior, se puede decir que hay mucho interés en la discapacidad visual, probablemente es debido al gran crecimiento en las investigaciones en visión computacional, y menos interés en la discapacidad intelectual.

Las discapacidades visual y motora no son de gran interés desde el punto de vista del AAL, pero tampoco se descarta que no exista investigaciones que apoyen a esos tipos de discapacidad, quizá en un futuro, el AAL hará que existan más proyectos o investigaciones.

	2012	2013	2014	2015	2016	2017	2018	Total	Suma de los totales
Visual impairment	6	3	6	9	1	4	4	33	60
Blindness	2	1	4	9	1	2	1	20	
Vision loss	1	1	0	1	0	2	2	7	
Deaf	7	1	2	5	7	0	2	24	46
Hearing loss	2	2	6	5	2	3	2	22	
Mobility problems	0	0	2	4	3	2	2	13	33
Mobility limitations	1	1	1	0	2	5	0	10	
Physical disability	0	0	2	2	3	2	1	10	
Intellectual disability	3	0	0	3	5	2	0	13	19
Mental illness	2	0	0	0	0	1	3	6	

Tabla 3-2: Resultados de la búsqueda.

Mientras tanto la discapacidad auditiva existe un interés regular, probablemente a que apenas se están desarrollando tecnologías para dar asistencia a ese grupo de personas, aunque no tenga mayor impulso como con la discapacidad visual.

Cabe destacar que en los resultados de la búsqueda existen proyectos y patentes que den asistencia en general, es decir proponen un sistema AAL para dar asistencia a personas sin importar la diversidad social, tales como un cuarto de un hospital, aplicaciones, etc.

Finalmente, en cada artículo revisado que en relación a la búsqueda sobre PcDA, se revisó la bibliografía para encontrar otras investigaciones que no aparecieron en la búsqueda.

3.2. Trabajos relacionados

De acuerdo a los resultados obtenidos en la búsqueda, se consideraron sólo aquellos artículos que tienen una mayor relación con la propuesta de la presente tesis y se des-

criben a continuación separados en dos áreas principales: asistencia a PcDA y detección y clasificación de ladridos de perros. No se encontró algún trabajo que de asistencia a PcDA y a la vez la detección de ladridos de perros.

3.2.1. Asistencia a PcDA

Tessendorf *et al.* [2012], crearon un sistema para mejorar los auxiliares auditivos integrando un sensor de movimiento, además de detectar sonidos simples que ocurren en una oficina y mandar una notificación a un dispositivo móvil. Además realizaron en diferentes pruebas, como son evaluación de satisfacción, reconocimiento de audios, etc., la mayoría fueron probadas con personas sin discapacidad auditiva; para probar el sensor de movimiento utilizó a 12 personas sordas, sin embargo, para la prueba de audio no reclutó a personas sordas, en su lugar fueron personas sin sordera. El tamaño de la base de datos para crear el modelo de aprendizaje de máquina, fue de aproximadamente 1000 audios en general.

Nanayakkara *et al.* [2012], construyeron una silla especializada para PcDA que consiste en generar vibraciones de acuerdo al ritmo de una música o canción, a diferencia de los demás trabajos no usaron un método de ML, en su lugar usaron la librería MIDI para determinar las notas musicales y posteriormente mostrarlos en una pantalla que se encuentra instalada en la silla. Se realizaron varias pruebas cambiando el número de PcDA, 43 para la experiencia musical, 36 para evaluar la propuesta de visualización, 12 para la prueba de gestos humanos que ocurren visualmente en alguna canción y 6 para la prueba de satisfacción.

Mielke y Brück [2015], diseñaron e implementaron una automatización del hogar con un entorno específico para asistir a las PcDA; su sistema se basa en una red Bluetooth Low Energy, además de detectar sonidos en general, los avisos se enviarán a un smartphone o a un smartwatch, este último es la parte donde los autores se enfocan, ya sea mandando un tipo de aviso visual o por medio de vibraciones, sin embargo, en la Tabla 3-3 se presentan los datos de uno de sus artículos que está enfocada en el smartphone.

Yaganoglu y Köse [2017], desarrollaron un dispositivo para detectar sonidos débiles y fuertes, con un filtro de ruido implementado, usando 4 micrófonos y con base a ella notificar al usuario por medio de vibración colocados en un dedo en cada mano con el fin de conocer el punto de origen del sonido, probaron dos métodos de clasificación, K Nearest Neighborhood (KNN) y SVM; concluyendo según los autores, el SVM resultó más efectiva.

Los mismos autores escribieron otro artículo (Yaganoglu y Köse [2018]), en la que desarrollan el mismo dispositivo, pero usando una tarjeta Raspberry Pi; además de detectar sonidos como, teléfono, alarma, timbre, claxon, ladridos y ruidos por humanos, en la que al detectar uno de los sonidos mencionados tendrá una vibración distinta, en este caso usaron KNN para la clasificación.

Fanzeres *et al.* [2018], crearon una aplicación móvil para detectar sonidos en general, es decir, todos los procesos de extracción de características hasta el reconocimiento de sonidos se realiza en un teléfono inteligente, además de mostrar gráficamente las ondas de los audios. La razón por la que escogieron por medio de aplicación móvil, es debido a una encuesta realizada por los autores, donde el resultado de la encuesta fue que las PcDA prefirieron un sistema inteligente que se pudiera utilizar en la casa, en el trabajo y en la calle, por lo que idearon una solución de utilizar un teléfono inteligente de tal manera que las PcDA pudieran mostrar mayor interés en su trabajo; incluso en el reconocimiento del audio son hechos por los mismos usuarios, es decir, grabando audios y por medio de ajustes se le puede asignar una etiqueta para el algoritmo ML previamente entrenado por los mismos autores.

3.2.2. Reconocimiento de ladridos de perros

En la revisión de artículos relacionados en el reconocimiento de ladridos de perros se tomaron todos los artículos recientes, es decir, desde el año 2012 hasta la actualidad, sin embargo para fines de este trabajo se tomó en cuenta un artículo publicado en el año 2008 por Molnár *et al.* [2008], ya que sirvió como base para varios artículos sobre

detección de ladridos de perros.

En el trabajo de Molnár *et al.* [2008], utilizó 6000 ladridos para el entrenamiento del algoritmo de ML, en su caso utilizaron el Naive Bayes, con el fin de reconocer ladridos en determinadas situaciones que ellos mismos decidieron analizar: “Pelear”, “Caminar”, “Solo”, “Pelota” y “Jugar”; sin embargo de ese trabajo se enfocará con más detalle en la metodología usada, que se explicará en el Capítulo 4.

Barris [2015], sugirió un clasificador para reconocer ladridos de perros, es decir entre Correlación cruzada, Cuantificación vectorial y Red neuronal artificial, con los resultados obtenidos los ingresa a un clasificador de decisión llamado Jury (Combinación de los 3 clasificadores que utilizó), usando como características varios filtros de MFCC. De acuerdo al autor, la red neuronal artificial fue el más consistente y robusto; sin embargo, menciona que Jury tuvo un resultado bastante óptimo y que se podría mejorar.

Perez-Espinosa *et al.* [2018], realizaron un trabajo similar al de Barris, pero la diferencia está en que determinaron cuáles eran las mejores características y método de clasificación para detectar ladridos de perros; entre los métodos de clasificación que usaron fueron: SVM, Naive Bayes, Random Forest, Bagging, Red neuronal convolucional y árbol de decisión; de entre ellas el que tuvo un mejor resultado fue SVM, así como la peor que es el árbol de decisión. Y como mejor característica determinaron que es una combinación de varios conjuntos de características tomados del INTERSPEECH 2009, 2010 y 2011. En la sección 2.3, se mencionó que la segmentación realizada por los autores fue manual, en este caso, la segmentación fue realizada de manera automática, que sirvió para eliminar partes sin sonido o con ruido, esos audios segmentados varían entre 0.3 y 5 segundos; y la de poder clasificar la raza del perro, en su caso clasificaron chihuahua, french poodle, schnauzer y otros.

El trabajo de Hantke *et al.* [2018] consistió en detectar ladridos en diferentes situaciones (Pelota, Pelar, Comida, Jugar, Desconocido, Caminar), con el fin de estudiar y conocer las emociones o reacciones de los perros en esos eventos como son: Agresión, Desesperación, Temor, Divertido y Feliz.

En la Tabla 3-3 y en la Tabla 3-4 se muestran una comparativa entre los trabajos mencionados.

3.2.3. Otros desarrollos: SOPARE

Existe un proyecto de reconocimiento de sonido, aunque no fue específicamente desarrollado para asistir a PcDA o reconocer ladridos de perros; fue realizado en el lenguaje Python que pueda ser compatible con la Raspberry Pi, se le conoce como SOPARE (Sound Pattern Recognition), Martin Kauss¹, el desarrollador, presume los siguientes puntos:

- Procesamiento de audio en tiempo real.
- Funciona sin fallos en computadoras “pequeñas” como Raspberry Pi, Banana Pi, etc.
- No hay necesidad de tener internet o dependencias en la nube para su funcionamiento.
- Permite ingresar código para el control de dispositivos y a su vez crear un sistema inteligente.

Sin embargo, el mismo desarrollador menciona que el proyecto aún no está terminado del todo, puesto que aún no funciona correctamente en Python 3 y además falla en oraciones (palabras separadas).

Se revisaron los archivos que son utilizados por SOPARE, y se encontró que las características que se extraen son: Transformada Rápida de Fourier (FFT), el valor máximo del FFT y la frecuencia fundamental.

Aunque hay escaso detalle sobre el algoritmo de ML usado, con la revisión, se pudo ver que se realiza comparaciones entre el dato a entrenar y los datos entrenados, con la finalidad de encontrar una similitud entre ellas, una vez que se realiza el entrenamiento,

¹www.bishop.org

Artículos sobre asistencia a PcDA						
Artículo	No. de PcDA	Método ML con mayor precisión	Características extraídas	Sonidos	No. de datos	¿Raspberry Pi?
Tessendorf, et al., 2012	12	SVM	Promedio, Varianza, Magnitud, Promedio cruzado, ZCR, RMS	Múltiples sonidos de la calle, del hogar y del ambiente	1000	No
Nanayakkara, et al., 2013	49	-	MIDI	Música	-	No
Mielke y Brück, 2015	9	Red neuronal	MFCC, ZCR	Gaído, cláxon, calle, saludo	311	No
Yaganoglu y Köse, 2017	4	SVM	RMS, ZCR, Varianza	Múltiples sonidos de la calle, del hogar y del ambiente	1000	No
Yaganoglu y Köse, 2018	5	K Nearest Neighbors	ZCR, MFCC, SSC, LSF, LPC, LAR	Teléfono, alarma, timbre, cláxon, ladridos y ruidos por humanos	10000	Sí
Fanzeres, et al., 2018	-	Nearest Neighbors	LPC, MFCC, Desviación estándar del flujo espectral, Variabilidad espectral, Flujo espectral, Punto de reducción espectral, Compacidad	Múltiples sonidos de la calle, del hogar y del ambiente	300	No

Tabla 3-3: Comparativa sobre asistencia a PcDA.

Artículos sobre ladridos de perros				
Artículo	No. de ladridos	Método ML con mayor precisión	Características extraídas	¿Raspberry Pi?
Barris, 2015	537	Red neuronal artificial	MFCC	Sí
Pérez, et al., 2015	6614	SVM	MFCC, Variedades de espectros, Frecuencia fundamental, ZCR	No
Pérez, et al., 2018	6103	SVM	Combinación de conjuntos de características IS-09, IS-10 y IS-11	No
Hantke, et al., 2018	227	SVM	Geneva Minimalistic Acoustic Parameter, ComParE	No

Tabla 3-4: Comparativa sobre detección de ladridos.

se crea un diccionario donde contiene todas las características y valores para ser usados en el reconocimiento de un sonido, en particular en tiempo real.

En la Figura 3-1 se aprecia la arquitectura de SOPARE.

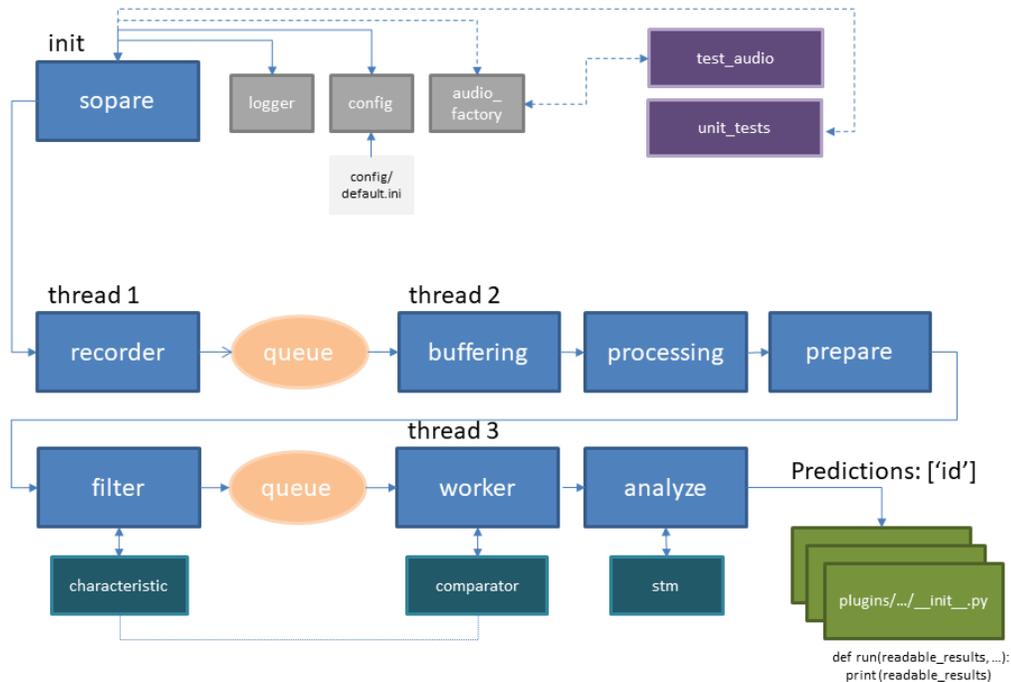


Figura 3-1: Arquitectura de SOPARE (Sound Pattern Recognition).²

De acuerdo al desarrollador del proyecto, su funcionamiento se basa en tres procesos que se ejecutan al mismo tiempo en distintos núcleos de la CPU, después de iniciar la fase `init`, que consiste en verificar las configuraciones y los audios almacenados antes de comenzar con el reconocimiento o la obtención de datos.

Proceso 1: Ese proceso únicamente se encarga de grabar audios de manera continua para pasarlos a la siguiente tarea.

Proceso 2: Con los audios capturados se procede a transformar en datos para el proceso de análisis (Tarea 3), además se crean las características y algunos filtros para

²www.bishoph.org/sopare-architecture-and-plugins/

tener datos adecuados para su análisis.

Proceso 3: El conjunto de características son comparadas con el resultado del entrenamiento, es decir con el modelo, una vez que se comparan se crea un nuevo modelo; además se ejecutan plugins que son archivos modificados por algún desarrollador ya sea para automatizar algún dispositivo o crear un sistema inteligente como lo es el AAL.

3.2.4. Recapitulación de los trabajos encontrados

Se puede concluir que en los trabajos revisados el mejor clasificador para ladridos de perros es el SVM, por lo tanto ese se usó para la realización del prototipo en cuanto al reconocimiento automático de ladridos de perros; además se realizó la segmentación de audio de manera manual para tener datos lo más homogéneo posible con el fin de obtener mejores resultados.

Fueron 32 PcDA con las que se trabajó para realizar las encuestas y 5 personas para las pruebas del prototipo, además se determinó la satisfacción de usuario, en el Capítulo 4 se detalla sobre la encuesta aplicada y en el Capítulo 5 se explica el procedimiento para evaluar el prototipo. En los trabajos realizados sobre asistencia a PcDA, los autores utilizaron audios genéricos, en este trabajo sólo se enfocará en los ladridos de perros.

De la misma manera que el proyecto SOPARE, en este trabajo se hizo el reconocimiento de ladridos sin utilizar internet, o bien, en la nube y el desarrollo del prototipo en la Raspberry Pi.

Capítulo 4

Metodología

4.1. Introducción

Para la realización de este trabajo se siguió principalmente la “Metodología para la creación de interfaces inteligentes aplicadas a AAL” (Sánchez [2018]), la cual se explicará con más detalle en este capítulo; además, se explicarán las técnicas o algoritmos usados para este trabajo.

4.2. AMCIAAL

Con los resultados obtenidos en la subsección 4.2.1, se procedió a seguir un proceso llamado “Metodología para la creación de interfaces inteligentes aplicados a AAL” (AMCIAAL por sus siglas en inglés) que consiste en cuatro fases: 1) Captura y Análisis de requerimientos, 2) Diseño de IUIs (Intelligent User Interfaces), 3) Implementación y 4) Pruebas. En la Figura 4-1 se ilustra en forma de diagrama el flujo en el método mencionado. Este modelo fue de utilidad para la realización de un sistema AAL.

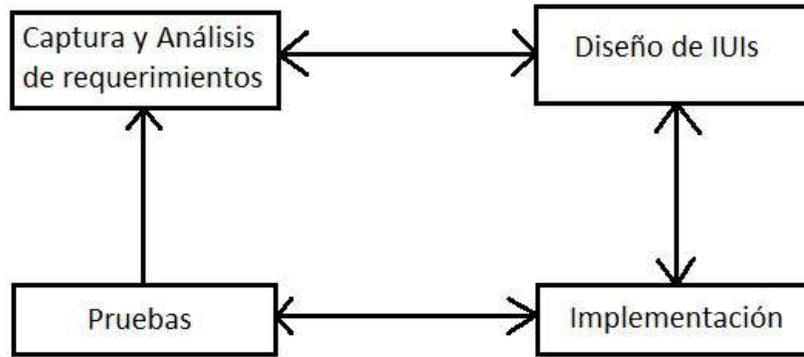


Figura 4-1: Diagrama AMCIAAL (A Methodology for Creation of Intelligent Interfaces applied to AAL) Sánchez [2018].

4.2.1. Captura y Análisis de requerimientos

Se define el problema y los objetivos para poder obtener información que será de utilidad para desarrollar el software. En este caso, el problema consistía en la necesidad de una PcDA de tener independencia en su hogar, tal como se explicó en el Capítulo 1, por esa razón se realizó una encuesta para determinar cuáles eran las situaciones de mayor frecuencia y mayor riesgo del usuario objetivo con respecto a su discapacidad.

Encuesta

Se realizó una encuesta a un grupo de 32 PcDA, todos pertenecientes a la Asociación Mexicana para la Comunicación y Superación de Personas con Discapacidad Auditiva; previamente se les preguntó a los usuarios de manera libre cuáles son los sonidos que necesitaban escucharlos, posteriormente se seleccionó los 5 sonidos con mayor popularidad para elaborar la encuesta utilizando la metodología Likert. A partir de los resultados de la encuesta se decidió trabajar con el reconocimiento automático de ladridos de perro a partir de los audios colectados en el hogar. En la Figura 4-2 y en la Figura 4-3 se presentan las preguntas de la encuesta aplicada para su posterior análisis.

Al evaluar a las PcDA, se realizó el conteo y análisis de los resultados obtenidos, nos sirvió para decidir el audio a detectar por el prototipo; se puede observar en la Figura

Encuesta de diagnóstico.

Esta encuesta tiene fines académicos con el propósito de desarrollar un prototipo que permita ayudar a personas con discapacidad auditiva. Toda información en esta encuesta es completamente anónima.

- 1- Tiene sordera de un oído ____ o de ambos ____
- 2- Cuál es tu nivel de pérdida auditiva:



- 3- De las siguientes situaciones señala con una "X" con qué frecuencia ocurre en tu hogar:

	Frecuencia				
	Siempre	Casi siempre	A veces	Casi nunca	Nunca
No puedo saber cuándo un aparato electrodoméstico termina su ciclo (lavadora, microondas, etc.).					
No puedo percibir los ladridos de un perro. (En caso de no tener perros, no contestar esta pregunta).					
Cuando una puerta está cerrada no puedo saber lo que sucede del otro lado. (Baño, Cuarto, Puerta principal, etc.)					
No puedo saber cuándo una llave de agua se quedó abierta por accidente.					
No puedo darme cuenta cuando alguien toca a la puerta principal de mi casa o el timbre.					

Figura 4-2: Preguntas sobre la frecuencia que ocurre un sonido en particular.

4-4 y en la Figura 4-5, que el sonido con la mayor puntuación fue el correspondiente a los ladridos de perros, siendo esa la razón por la cual el prototipo está enfocado en detectar

- 4- De las siguientes situaciones señala con una "X" la importancia que tiene para usted los siguientes eventos que ocurren en tu hogar:

	Importancia				
	Vital	Muy importante	Importante	Poco importante	Sin ninguna importancia
No puedo saber cuándo un aparato electrodoméstico termina su ciclo (lavadora, microondas, etc.).					
No puedo percibir los ladridos de un perro. (En caso de no tener perros, no contestar esta pregunta).					
Cuando una puerta está cerrada no puedo saber lo que sucede del otro lado. (Baño, Cuarto, Puerta principal, etc.)					
No puedo saber cuándo una llave de agua está abierta por accidente.					
No puedo darme cuenta cuando alguien toca a la puerta de mi casa o el timbre.					

¡Gracias por apoyarnos!

Figura 4-3: Preguntas sobre el nivel de riesgo que puede presentarse si ocurre un determinado sonido.

ese sonido.

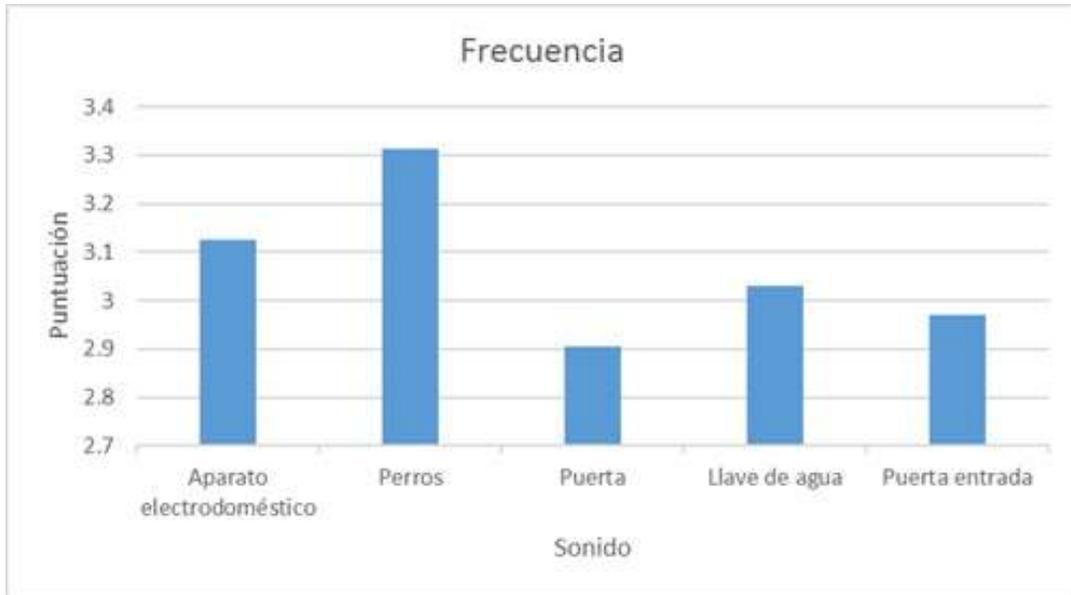


Figura 4-4: Resultado sobre la frecuencia que ocurre un sonido en particular.

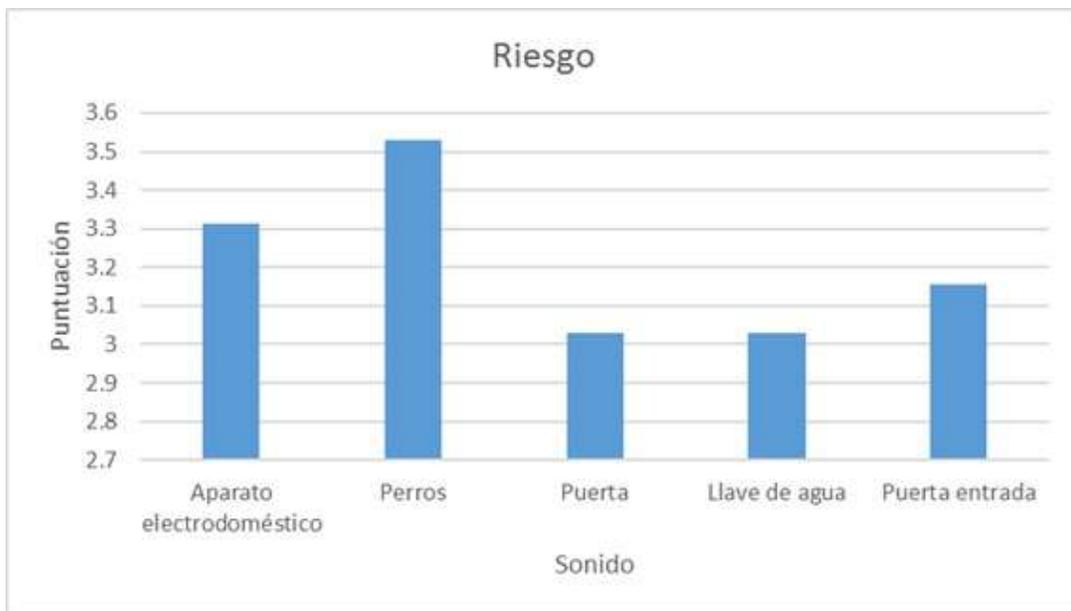


Figura 4-5: Resultado sobre el nivel de importancia que puede presentarse si ocurre un determinado sonido.

4.2.2. Diseño de IUIs (Intelligent User Interfaces)

En este trabajo se crea una interfaz de comunicación entre el usuario y el dispositivo, además de crear un diseño en la que se pueda manipular de manera eficaz siguiendo los

requisitos obtenidos en la fase anterior, el de “Captura y Análisis de requerimientos”. El diseño del prototipo de este trabajo consistirá en una conexión entre la Raspberry Pi 3 y un arreglo de micrófono ReSpeaker Mic Array v2.0; además para cumplir con los requisitos del usuario de hacerle saber la situación (ladridos de perros) se usó una comunicación con un teléfono inteligente el cual puede mostrar mensajes escritos o vibraciones cuando recibe notificaciones.

Raspberry Pi 3 Modelo B

Es un ordenador de placa reducida (SBC)¹ que sirve principalmente para desarrollar prototipos embebidos. La Raspberry Pi es muy usada en el ámbito académico para crear pruebas de concepto tanto para ambientes inteligentes como para internet de las cosas, gran parte de su éxito radica en la gran disponibilidad de información, tutoriales y código fuente ofrecidos por la comunidad. Tal tarjeta cuenta con las siguientes características:

- Memoria RAM de 1 GB.
- Procesador Quad-Core, que corre a 1.20 GHz.
- Sistema Operativo Raspbian.
- Cuatro puertos USB.
- Salida de audio y vídeo.
- Conexión Ethernet, WiFi y Bluetooth.

En este trabajo se pudo constatar la manera en la cual se puede usar la Rpi3 para llevar a cabo experimentos de aprendizaje de máquina.

¹www.raspberrypi.org/

²www.raspberrypi.org/



Figura 4-6: Raspberry Pi 3 Modelo B²

ReSpeaker Mic Array v2.0

Una placa que contiene un arreglo de 4 micrófonos³ que permite mejorar la calidad del reconocimiento de audio y poder realizar en conjunto con otras tareas en tiempo real; también incluye un algoritmo que permite la localización del sonido origen. Se puede conectar por medio de USB para el control del dispositivo y suplir energía para su funcionamiento; y por medio de un salida Jack de 3.5 mm para las bocinas o audífonos. En la Figura 4-7 se muestra el dispositivo. Es posible usar un micrófono más pequeño para realizar el prototipo sin embargo esta placa tiene un arreglo de micrófonos que lo hace más sensible y también tiene una arreglo de leds que pueden programarse de acuerdo a diferentes eventos por lo cual resultó una buena elección para trabajar.

4.2.3. Implementación

Se realizó la integración física y lógica del dispositivo y con la ayuda del usuario final se pudo verificar que se cumple con los requisitos previamente planteados; se realizó la

³wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/

4.3. Obtención de los datos

Para obtener datos suficientes para el algoritmo de aprendizaje automático, se grabaron ladridos de 3 perros de raza mixta utilizando ReSpeaker, mencionado anteriormente, fueron 6 grabaciones con duración entre un minuto a cinco minutos. En cada audio se realizó la segmentación de ladridos de forma manual, en la Figura 4-8 y 4-9, se muestran un ejemplo de uno de los audios y el ladrido obtenido por el micrófono, respectivamente.

Además de esos ladridos, también se utilizó una base de datos proporcionada por Perez-Espinosa *et al.* [2018], la “Mescalina Bark ID Database” propiedad de Efecto Mescalina SAPI de CV, una base de datos de más de 100 perros, eso con el fin de obtener un mejor resultado.

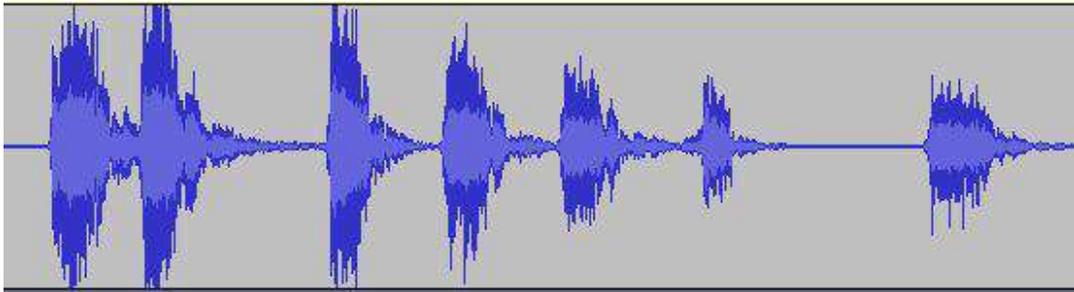


Figura 4-8: Serie de ladridos en un fragmento de audio.

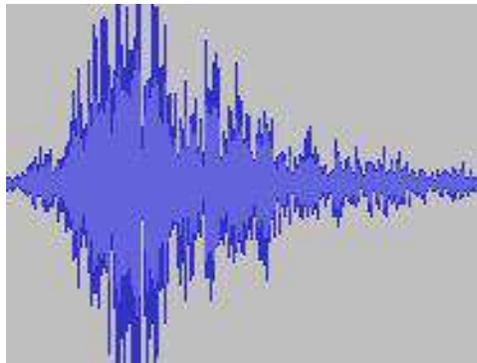


Figura 4-9: Un ladrido individual.

De esta forma se obtuvieron 1480 ladridos y 1500 no-ladridos, estos últimos se obtuvieron de grabar sonidos de golpear la madera, ladridos hechos por humanos, decir la

	Audios para entrenamiento	Audios para evaluación	Total
Ladridos	1033	447	1480
No ladridos	1053	447	1500
		Total	2980

Tabla 4-1: Número de audios utilizados en este trabajo.

palabra “hola”, sonido de llaves, ruidos de ventilador, segmentos de canciones, golpear libros y otras voces humanas; en el caso de la palabra “hola” se pronunció la palabra de una manera grave o aguda de tal forma que podría asemejar a un ladrido o bien que podría confundirse con un ladrido; en la Tabla 4-1, se presenta la cantidad de audios utilizados para este trabajo. Esos audios se separaron en 70% para el entrenamiento y en 30% para las pruebas por parte del algoritmo SVM, además esos datos, es decir el 100%, se usaron para encontrar los mejores parámetros del SVM, costo (C) y gamma (g).

Después de obtener esos datos, se procedió a verificar la precisión del modelo utilizando unos 50 audios aparte de las mencionadas anteriormente.

4.4. Base de datos para la evaluación de las librerías

Como se mencionó en la sección anterior, se usó una buena cantidad de audios para un buen reconocimiento, eso para evitar falsos positivos durante las pruebas en SVM.

De un total de 2980 audios, fueron separados en 1480 ladridos (positivos) y 1500 no-ladridos (negativos); se verificó que esos datos fueran homogéneos, es decir que cada audio positivo sea un solo ladrido y no una secuencia de los mismos ladridos, de una manera similar se procedió con los negativos, por ejemplo, cada audio no contiene una secuencia de golpes de puerta, sino sólo una; en el caso de canciones, se separaron por tonos.

La totalidad de audios fueron almacenados en una base de datos no estructurada, utilizando MongoDB, para el fácil manejo de los datos almacenados; a cada audio de los ladridos se le asignó una etiqueta “+1” y para los audios de no-ladridos un “-1”;

lo anterior se guardó en una base de datos llamada “DBark”, además se creó otra base de datos, “DBTest” para los audios de prueba que sirvió para evaluar la eficiencia del entrenamiento en cada librería usada en este trabajo, misma que se conocerá a detalle en la sección 4.7.

La base de datos fue únicamente para evaluar las librerías, ya que al encontrar la adecuada, se utilizaron los audios de manera directa y no desde una base de datos, en este caso, MongoDB, una base de datos no estructurada, que funciona muy bien para trabajar con una base de datos grande.

4.5. Procesos para la implementación final

Utilizando los mismos datos de la sección anterior, se procedió a utilizar las herramientas proporcionadas por Chang y Lin [2011], libsvm, para encontrar los mejores valores de costo (C) y gamma (g) que son parámetros esenciales en el SVM. Con los parámetros obtenidos se procedió a verificar el mejor modelo para el reconocimiento de ladridos, es decir, entre las funciones kernel lineal, polinomial, radial y sigmoide, se determinó el adecuado para el prototipo.

4.6. Uso de PyAudio.

El reto de este trabajo fue encontrar una mejor manera para procesar sonidos en tiempo real, ya que el objetivo es proporcionar un sistema inteligente que sea capaz de reaccionar a los eventos inmediatos y en respuesta a eso, notificar a los usuarios.

Se usó la librería PyAudio⁵ para poder procesar el sonido en segmentos de un tiempo asignado, en este caso de 1 segundo, con ese segmento realizar un análisis de aprendizaje de máquina, es decir, extraer las características y comparar con el modelo obtenido del entrenamiento del SVM.

⁵<http://people.csail.mit.edu/hubert/pyaudio/>

Características	
Procesador	Intel Core i3-5005U 2GHz
RAM	4 GB
SO	Windows 10 Home
HDD	448 GB

Tabla 4-2: Características de la computadora utilizada para implementar el entrenamiento del modelo SVM.

En la Tabla 4-2, se presenta las características de la computadora donde se realizaron la evaluación de las librerías de extracción de características, de entrenamiento y de notificaciones, antes de implementarse en la Raspberry Pi.

4.7. Bibliotecas de procesamiento de audio

Se evaluaron distintas librerías para determinar la calidad y el tiempo de procesamiento de los audios, con el fin de seleccionar las que resulten adecuadas para el desarrollo del prototipo.

En el análisis, se verificó la documentación, el tiempo de procesamiento, entre otras; en la Tabla 4-3 se resume la evaluación realizada, a continuación se describe cada una de esas librerías:

OpenSmile (Eyben *et al.* [2013]): Una librería escrita en C++ que puede ser usada en diversos sistemas operativos como son Linux, MacOS y Windows, utilizan PortAudio para el procesamiento de audios, permite leer y escribir en formato WAV. Proporciona una buena documentación, en el uso de cada uno de sus comandos, sin embargo al no contar con una API para Python se recurrió a utilizar la librería “subprocess” para su uso.

Para su uso en streaming, se emplea un comando particular que consta en grabar, almacenar y procesar esos audios; sin embargo uno de los obstáculos que se presentaba en su uso es almacenar los audios y posteriormente eliminarlos, puesto que uno de los objetivos en el desarrollo del prototipo es la detección de ladridos de perros en tiempo

real. Tiene una variedad de opciones de características, sin embargo, durante su revisión hubo dificultades para seleccionar las características definidas en este trabajo.

Parselmouth (Jadoul *et al.* [2018]): Fue desarrollada en Python para tener acceso a las funcionalidades principales de PRAAT, Boersma y Weenink [2018], como son el análisis de sonidos, ya sea la intensidad, entonación, y otros más; su documentación esta bien detallado, además de mostrar ejemplos de uso.

Similar a OpenSmile en cuanto se refiere a streaming, sin embargo en la selección de características es más sencilla.

Librosa (McFee *et al.* [2019]): Una API de Python para el manejo de audios y extracción de características, similar a Parselmouth, su documentación detallada y con ejemplos sencillos de uso.

Permite trabajar archivos de audio o en streaming sin problema, además de su manejo flexible de las características usadas para el procesamiento de audio. Con la desventaja que en la instalación en la Raspberry Pi, se vuelve tediosa por requerir diversos componentes para su correcto funcionamiento.

PyAudioAnalysis (Giannakopoulos [2015]): Otra librería que permite extraer características, ofrece herramientas para facilitar el proceso de aprendizaje de máquina como puede ser la separación de conjunto de pruebas y de entrenamiento, determinar la probabilidad de predicción de cada clase; al igual que Librosa, permite trabajar con archivos de audio y en streaming; desafortunadamente no permite seleccionar características libremente, pero son suficientes para un correcto proceso de reconocimiento de ladridos.

Para evaluar el tiempo de procesamiento de las distintas librerías presentadas anteriormente se usó la librería `time` de Python, se evaluó desde que se ejecuta el comando hasta la finalización del proceso para la extracción de características.

Una vez terminadas las evaluaciones realizadas en una computadora de escritorio se procedió a ejecutarlo en la Raspberry Pi. Como resultado de la evaluación se decidió utilizar PyAudioAnalysis, ya que cumplía con la mayoría de los requisitos del prototipo desarrollado, además de la fácil instalación en la Raspberry Pi.

	OpenSmile	Parselmouth	Librosa	PyAudioAnalysis
Streaming	No	No	Si	Si
Archivo de audio	Si	Si	Si	Si
Gratuito	Si	Si	Si	Si
SO	Windows, Linux, Mac	Windows, Linux, Mac	Windows, Linux, Mac	Windows, Linux, Mac
API Python	No	Si	Si	Si
Tiempo de procesamiento	41.3 ms	0.9 ms	13.5 ms	26.4 ms
Selección de características	No	Si	Si	No

Tabla 4-3: Librerías para la extracción de características.

4.7.1. Notificaciones

Otra parte muy importante del sistema propuesto en esta tesis es la entrega de notificaciones al usuario, por tanto se procedió a seleccionar una biblioteca de software que permita enviar mensajes a un dispositivo móvil y que se pueda ejecutar desde la Raspberry Pi. Entre las opciones elegidas para evaluación están: Telegram⁶, PushOver⁷ y OneSignal⁸; en la Tabla 4-4, se muestra una comparativa entre las librerías mencionadas.

PushOver: Es una plataforma para recibir y enviar notificaciones, que ofrece una aplicación para dispositivos móviles y computadoras de escritorio, totalmente simple y fácil de usar; desgraciadamente no tiene una documentación detallada para los programadores como tal, pero sí ofrece ejemplos de su uso en distintos lenguajes de programación. Otra desventaja es que sólo se puede usar por 7 días de manera gratuita.

OneSignal: Es un servicio de notificaciones para páginas web y dispositivos móviles, que además permite la descarga de un paquete de archivos para compilarlo en alguna IDE, se usó Android Studio para compilar la aplicación, dentro de sus características,

⁶<https://telegram.org/>

⁷<https://pushover.net/>

⁸<https://onesignal.com/>

Librería	Gratuito	SO	API python	Tiempo de procesamiento
PushOver	Por 7 días	Win, Linux, Mac	Si	0.4 s
OneSignal	Si	Win, Linux, Mac	Si	0.45 s
Telegram	Si	Win, Linux, Mac	Si	0.91 s

Tabla 4-4: Librerías para realizar notificaciones via móvil.

con las configuraciones realizadas para su conexión, se puede enviar notificaciones desde su plataforma; su documentación esta bien detallada, sin embargo, solo ofrece soporte para navegadores web y dispositivos móviles como son Android, iOS, etc. Para su uso en Python se utilizó la librería “onesignal-sdk” que permite utilizar las herramientas que ofrece en OneSignal, misma que tiene una documentación totalmente simple y detallada.

Telegram: Es un servicio móvil basado en la nube muy popular, enfocado en la rapidez y seguridad para los usuarios, ofrece una API para poder desarrollar aplicaciones que usen Telegram por medio de la Bot API; existen varias librerías para su uso en Python, de entre ellas se usó Telebot, su documentación contiene buenos ejemplos de uso con una explicación detallada.

En la evaluación de esas librerías se tomó el tiempo desde que finaliza el procesamiento del audio hasta el envío de la notificación, aclarando que en este procedimiento no se consideró el reconocimiento de ladridos de perros; de acuerdo a la Tabla 4-4, PushOver y OneSignal fueron las más rápidas, sin embargo por la documentación y el tiempo de uso se prefirió usar OneSignal.

Además de realizar las notificaciones vía móvil, también se aprovechó el control de luces que ofrece el ReSpeaker; que sirvió para avisar al usuario que el mismo prototipo está enviando notificaciones en caso que no tenga el dispositivo móvil a su alcance. Las luces se configuraron con el máximo brillo, además de códigos de colores que se explica a continuación. En las Figuras 4-10, 4-11 y 4-12, se pueden observar visualmente las luces que emite el prototipo.

Azul parpadeante: Se está inicializando el código de reconocimiento de ladridos

Amarillo: El prototipo está listo para reconocer ladridos.

Verde: Se envía una notificación al detectar un ladrido.



Figura 4-10: La luz azul, se refiere a que se está inicializando, antes de ponerse en funcionamiento.



Figura 4-11: En amarillo, se encuentra preparado para reconocer ladridos.



Figura 4-12: En verde, se envió una notificación al usuario.

4.8. Extracción de características

En PyAudioAnalysis, además de ofrecer herramientas para el aprendizaje de máquina, se usaron 34 características definidas por la librería⁹. En la Figura 4-13, se muestran a manera de ejemplo, un ladrido y un no-ladrido, en este caso el sonido de una llave, con el

⁹<https://github.com/tyiannak/pyAudioAnalysis/wiki/3.-Feature-Extraction>

propósito de observar las diferencias visuales entre esos sonidos; a continuación se listan las características que proporciona la biblioteca mencionada, además de una imagen de ejemplo:

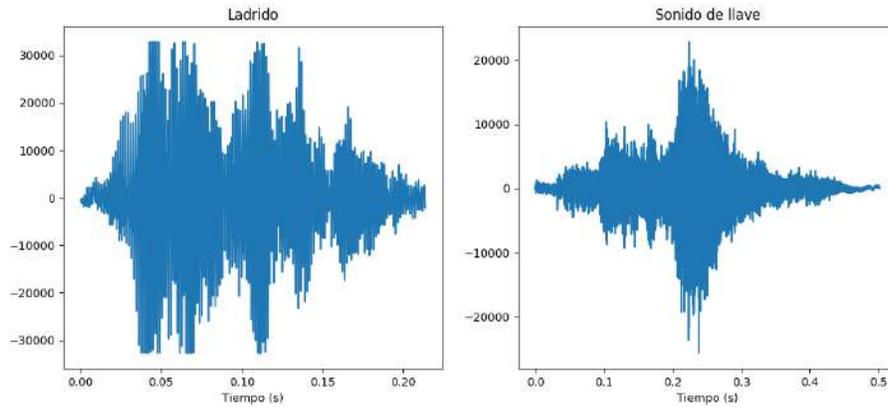


Figura 4-13: Audios de ejemplo.

1. Tasa de cruce por cero (ZCR): Calcula las veces que cambia de signo en un audio en particular. (Figura 4-14).

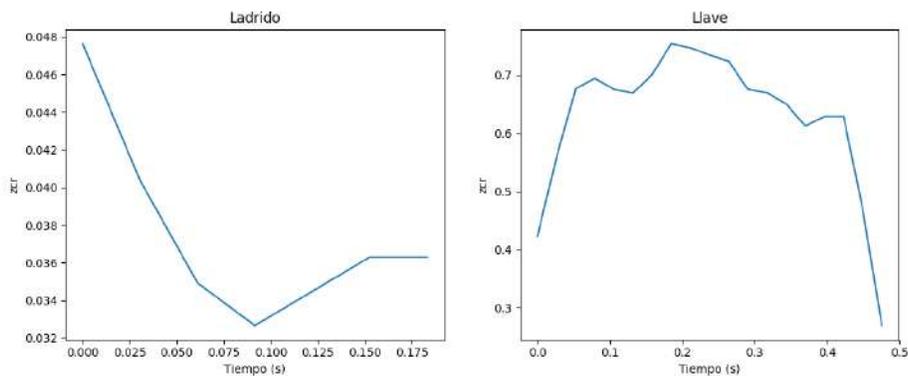


Figura 4-14: Gráfica de ZCR.

2. Energía: La suma de los cuadrados de los valores de la señal. (Figura 4-15).

3. Energía de entropía: Mide los cambios abruptos en la señal. (Figura 4-16).

4-8. Espectros: Mide algunos valores en el espectro del audio como son, centro de gravedad, momento central, frecuencia, energía y el cambio entre dos cuadros (frames,

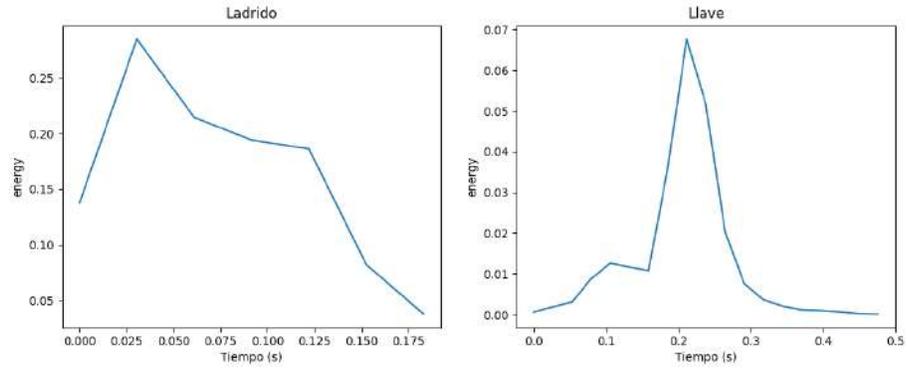


Figura 4-15: Gráfica de energía.

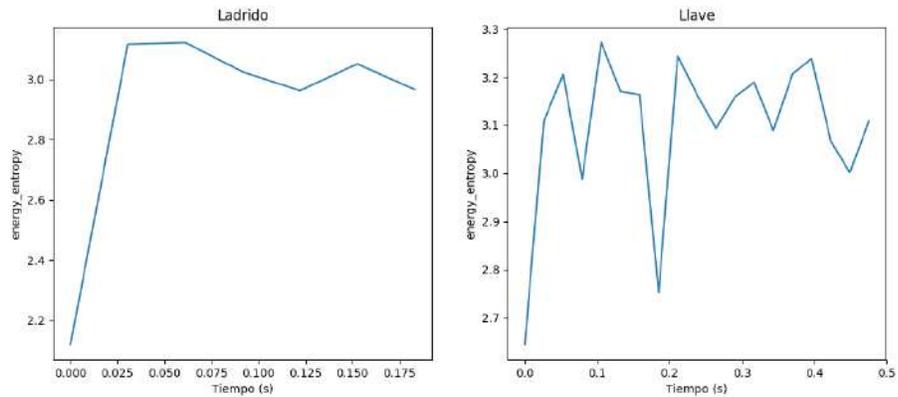


Figura 4-16: Gráfica de la energía de entropía.

en un audio en general, por segundo contiene 44100 frames) sucesivos. (Figura 4-17).

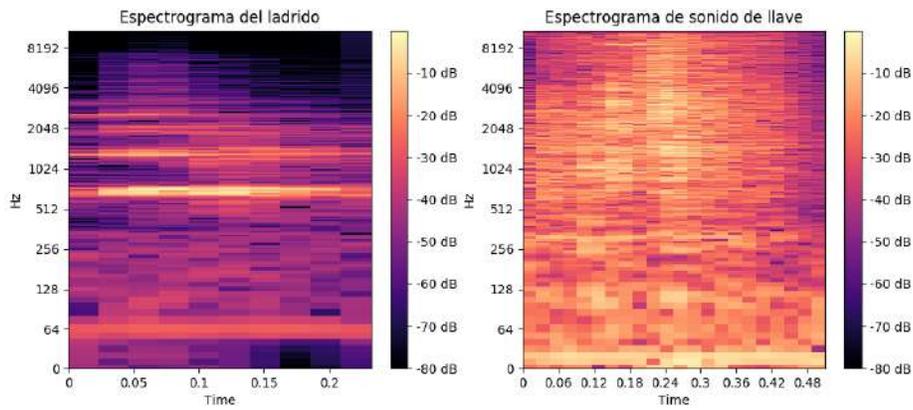


Figura 4-17: Gráfica de espectrograma.

9-21. MFCC: Se obtiene distintos coeficientes cepstrales de la escalada de mel, esos valores sirven para representar el habla desde la perspectiva del oído humano, sin embargo, es muy utilizado para el reconocimiento de un audio en general. (Figura 4-18).

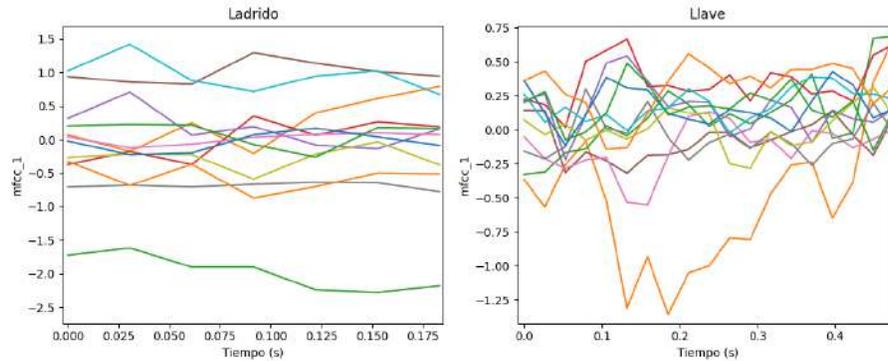


Figura 4-18: Gráfica de MFCC.

22-33. Cromagramas: Cada cromagrama representa a un grupo de nota musical particular que puede encontrarse en un audio. (Figura 4-19).

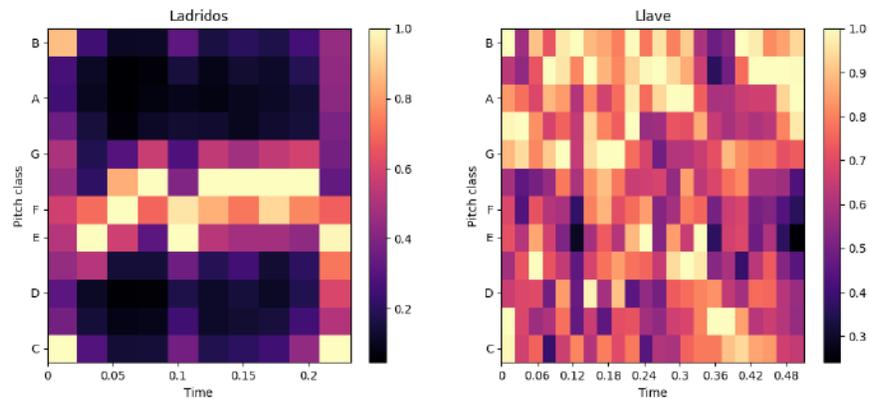


Figura 4-19: Gráfica de cromagrama.

34. Desviación del cromagrama: Calcula la desviación estándar de los cromagramas mencionados anteriormente. (Figura 4-20).

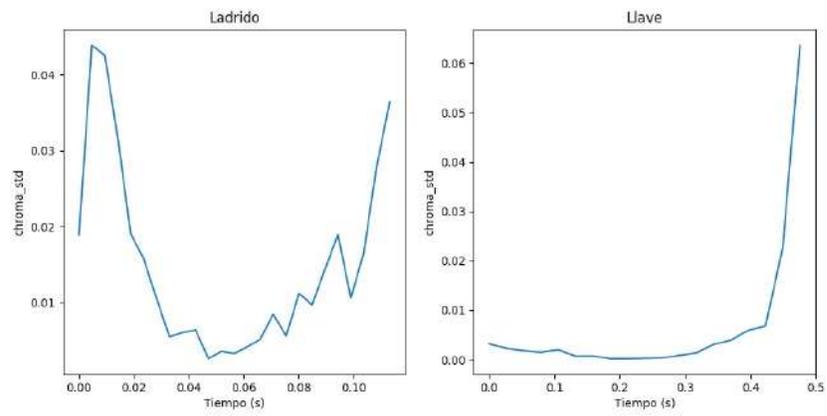


Figura 4-20: Gráfica de las desviaciones estándar de los cromagramas.

Capítulo 5

Implementación

5.1. Prototipo

Como se mencionó en el Capítulo 4, se usó Raspberry Pi como el núcleo del procesamiento de audio, la predicción usando SVM y de las notificaciones al usuario, además de usar ReSpeaker para capturar los ladridos y determinar por medio de ML si es necesario notificar al usuario. El diagrama del prototipo se visualiza en la Figura 5-1 y en la Figura 5-2 se puede apreciar su conexión en la realidad.

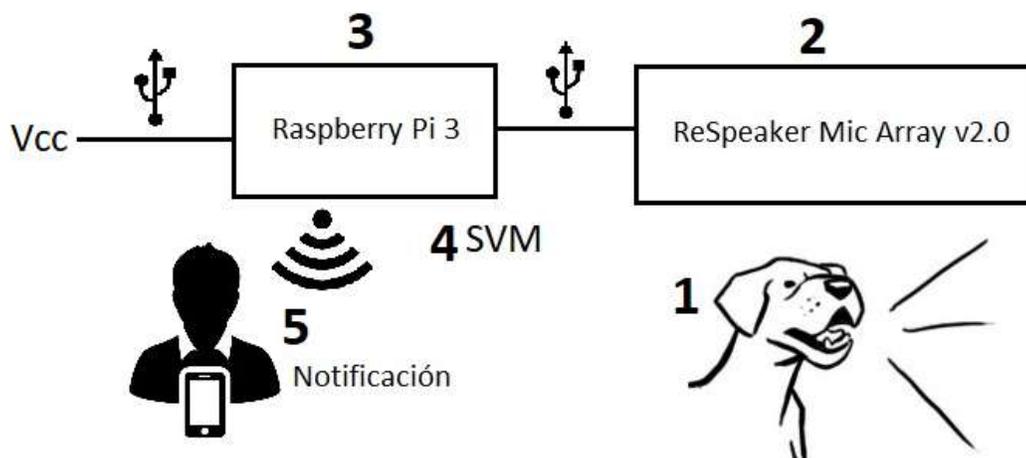


Figura 5-1: Diagrama del prototipo.



Figura 5-2: Representación real del diagrama.

De acuerdo al diagrama del prototipo la secuencia de números indica lo siguiente:

1. El perro ladra debido a un suceso ocurrido a su alrededor.
2. El micrófono ReSpeaker captará el sonido del ambiente, mismo que los transferirá a la Raspberry Pi.
3. La Raspberry Pi realizará el procedimiento de la extracción de características.
4. Estando en la misma placa, hay un algoritmo SVM que se encargará de verificar si el audio recibido contiene un ladrido o no.
5. En caso de tener un resultado positivo y tras validar que se deba informar al usuario, se enviará una notificación a un dispositivo móvil de una PcDA.

5.2. Diseño del prototipo

Al tratarse de un prototipo dedicado para PcDA, se pensó en un diseño que fuera fácil, se pensó en un objeto de forma cilíndrica, donde el micrófono estará colocado en

la parte superior conectado a la Raspberry Pi que será colocada en el fondo del cilindro; el mismo objeto tendrá un orificio para la dar corriente al prototipo, en la Figura 5-3, se puede ver el boceto.

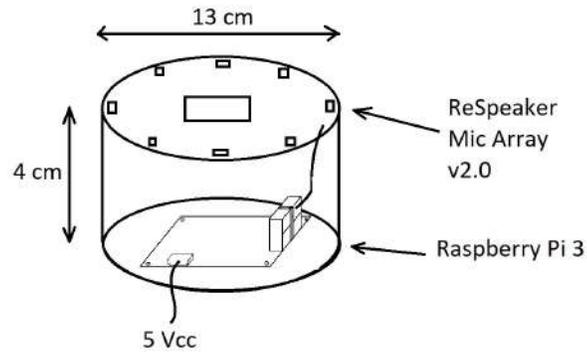


Figura 5-3: Diseño de prototipo.

En las Figuras 5-4 y 5-5, se pueden observar las representaciones reales del prototipo; se decidió colocar el ReSpeaker afuera de la caja, para evitar alguna obstrucción al tratar de detectar los ladridos de perros. El cilindro se imprimió en una impresora 3D con las medidas especificadas en el diseño junto con otros detalles como son los orificios que permite fijar las placas y permitir el paso de cables.



Figura 5-4: Representación real del prototipo (base).



Figura 5-5: Representación real del prototipo (tapa).

5.3. Modelo SVM

En el Capítulo 4, se mencionó que se escogieron las librerías OneSignal y PyAudioAnalysis, para notificación y procesamiento de audio, respectivamente, en esta sección se detallarán los pasos realizados para la obtención de un modelo SVM, misma que servirá para realizar el reconocimiento de ladridos en tiempo real.

5.4. Diagrama para la obtención del modelo SVM

Para obtener un modelo para ejecutarlo en la Raspberry Pi, como se mencionó anteriormente, se usó una base de datos de ladridos de las cuales por medio la librería PyAudioAnalysis se separaron en conjuntos de entrenamiento (70 % del total) y de evaluación (30 % del total), además de obtener los parámetros necesarios para un correcto proceso. Una vez entrenados se procede a crear un archivo “pkl”, una extensión de archivo creado por el módulo Pickle de Python que permite serializar una estructura de objetos convirtiendo en una cadena de bytes, donde está almacenado el modelo final que sirvió para realizar pruebas en tiempo real; además de una base de datos externa, obtenidas a partir de grabaciones con el ReSpeaker y debidamente segmentados, que sirvió para verificar la eficiencia del modelo, en la Figura 5-6 se detalla el proceso mencionado anteriormente; cabe recalcar que todo el proceso fue realizado en una computadora portátil

a excepción de los procesos realizados en la Raspberry Pi; eso se debe a que la placa no tiene la capacidad suficiente para procesar, o bien, de entrenar una mayor cantidad de datos, en su lugar sólo ejecutará el modelo final para su reconocimiento.

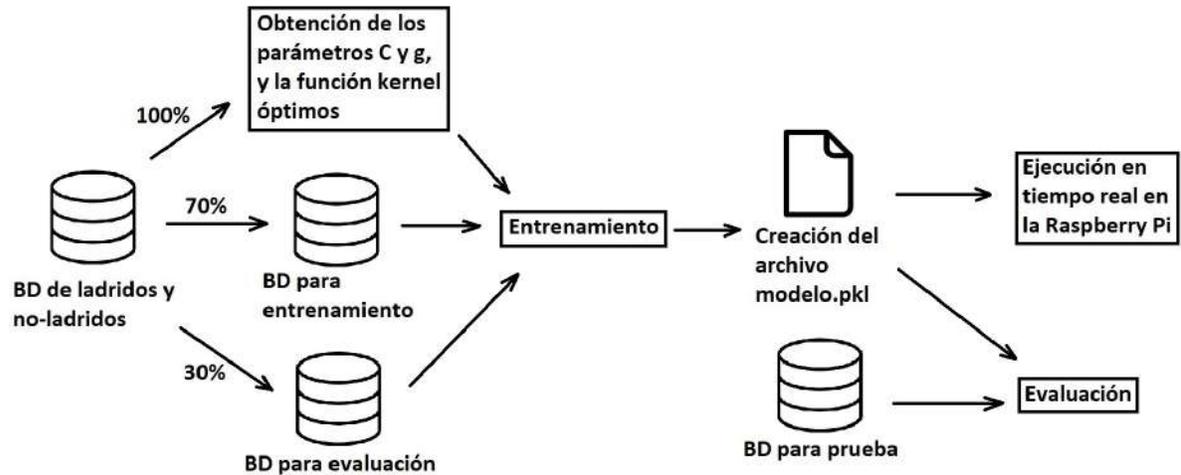


Figura 5-6: Diagrama para la implementación del SVM.

5.4.1. Parámetros C y g

Con ayuda de la base de datos y tras varios procesos se determinaron los mejores parámetros (C y g) y la mejor función kernel (lineal, polinomial, radial y sigmoide); con el propósito de tener un mejor entrenamiento y por ende un mejor modelo para reconocimiento; esto usando el 100 % de la base de datos.

Una vez que se obtuvo los valores deseados, se procedió al entrenamiento. En la Tabla 5-1, se puede apreciar los resultados de los dos mejores porcentajes para cada función kernel.

5.4.2. Ajustes en el modelo

Después del entrenamiento se procedió a crear un archivo, “modelo.pkl”, y a la vez se obtuvo su porcentaje de precisión, posteriormente se verificó la eficiencia con la base de datos externa, se comprobó que pudiera reconocer los audios de manera correcta;

Función Kernel	Precisión	Costo (C)	Gamma (g)	Grado
Lineal	94.5 %	50	-	-
	94.6 %	100	-	-
Polinomial	98.6 %	5	0.01	3
	98.7 %	100	0.01	2
Radial	98.9 %	50	0.01	-
	99.0 %	1000	0.01	-
Sigmoide	89.9 %	1000	0.00001	-
	91.8 %	1000	0.0001	-

Tabla 5-1: Precisión de cada kernel.

durante este proceso, se encontró con un obstáculo, todos los audios que se obtuvo de la segmentación manual para el entrenamiento tienen una duración variada entre 0.1 a 1 segundo y además, aunque algunos audios tienen la misma duración, las dimensiones de sus características son variadas; es decir, si dos audios que dura 1 segundo, el tamaño del vector característica de cada audio serán diferentes. Es por eso que se realizó la obtención de valores estadísticos para cada vector de característica de un audio en particular, con el fin de crear datos homogéneos y no variadas, en ese proceso se repitió las veces necesarias obteniendo la media, mediana y/o desviación estándar de cada vector de característica que se obtuvo en la extracción de un audio en particular, y finalmente se obtiene la precisión con la ayuda de la herramienta de libsvm, en total fueron 7 porcentajes que se compararon para usar la combinación de valores estadísticos con la mejor precisión.

En las Figuras 5-7 y 5-8 se puede ver el proceso mencionado anteriormente, en las Tablas 5-2, 5-3, 5-4, 5-5, 5-6, 5-7, 5-8, se pueden comprobar las diferentes precisiones para cada combinación, en cada tabla se muestran las probabilidades de predicción de las clases para 10 audios de tantas que fueron probadas.

Después de observar las precisiones arrojadas de diferentes combinaciones de características de vector, se seleccionó la combinación de media con desviación estándar, por tener la mejor precisión y mejores resultados en las probabilidades de predicción.

Otro punto a destacar, en algunas probabilidades hubo un falso positivo aún cuando el porcentaje sea muy bajo, por lo que decidió para evitar futuros falsos positivos, con la

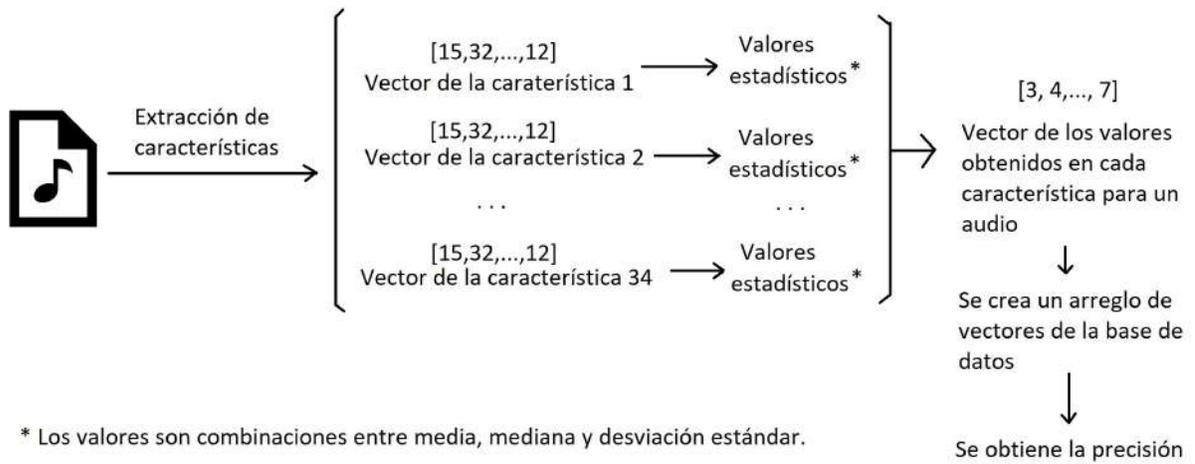


Figura 5-7: Proceso para buscar la precisión de una combinación de valores estadísticos.

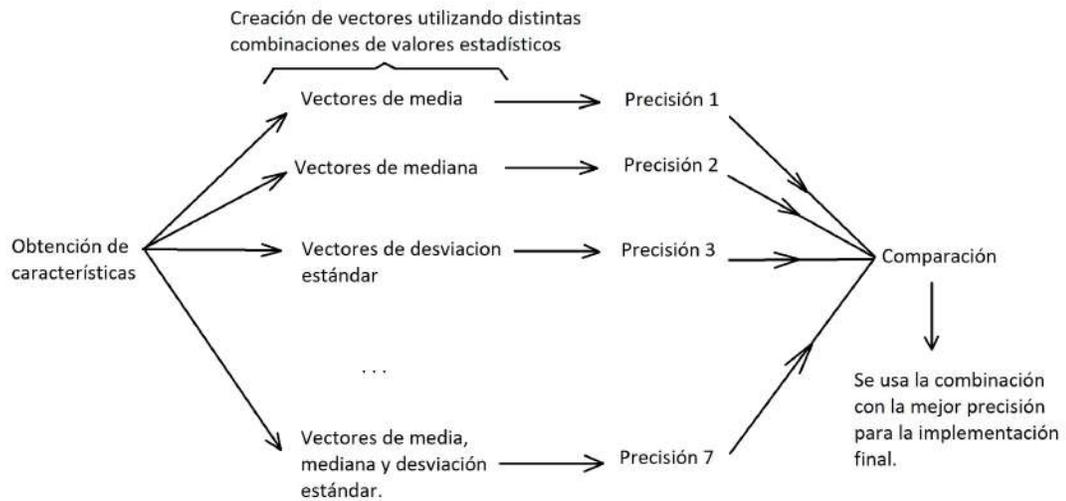


Figura 5-8: Proceso para encontrar la combinación de valores estadísticos con mejor precisión.

Precisión: 97.98 %	Media (m)
Audio	Probabilidad de que sea un ladrido
No-ladrado 1 (palabra 'hola')	42.45 %
No-ladrado 2 (golpear libro)	62.96 %
No-ladrado 3 (palabra 'hola')	63.49 %
No-ladrado 4 (segmento de canción)	98.31 %
No-ladrado 5 (ruido de ventilador)	34.57 %
No-ladrado 6 (ruido de ventilador)	0.45 %
Ladrado 1	85.16 %
Ladrado 2	99.99 %
Ladrado 3	97.76 %
Ladrado 4	97.99 %

Tabla 5-2: Precisión, utilizando sólo la media.

Precisión: 96.53 %	Mediana (M)
Audio	Probabilidad de que sea un ladrido
No-ladrado 1	29.16 %
No-ladrado 2	35.46 %
No-ladrado 3	20.75 %
No-ladrado 4	76.72 %
No-ladrado 5	34.06 %
No-ladrado 6	0.30 %
Ladrado 1	93.50 %
Ladrado 2	99.99 %
Ladrado 3	91.47 %
Ladrado 4	96.55 %

Tabla 5-3: Precisión utilizando sólo la mediana.

Precisión: 89.93 %	Desviación Estándar (DE)
Audio	Probabilidad de que sea un ladrido
No-ladrido 1	48.18 %
No-ladrido 2	80.85 %
No-ladrido 3	20.62 %
No-ladrido 4	0.21 %
No-ladrido 5	83.98 %
No-ladrido 6	0.42 %
Ladrido 1	64.66 %
Ladrido 2	59.89 %
Ladrido 3	99.20 %
Ladrido 4	99.99 %

Tabla 5-4: Precisión utilizando sólo la desviación estándar.

Precisión: 97.98 %	m + M
Audio	Probabilidad de que sea un ladrido
No-ladrido 1	20.32 %
No-ladrido 2	80.63 %
No-ladrido 3	38.39 %
No-ladrido 4	91.93 %
No-ladrido 5	13.53 %
No-ladrido 6	2.31 %
Ladrido 1	96.92 %
Ladrido 2	99.99 %
Ladrido 3	98.62 %
Ladrido 4	99.99 %

Tabla 5-5: Precisión utilizando media y mediana.

Precisión: 99.10 %	
m + DE	
Audio	Probabilidad de que sea un ladrido
No-ladrido 1	0.17 %
No-ladrido 2	19.14 %
No-ladrido 3	5.42 %
No-ladrido 4	10.77 %
No-ladrido 5	6.10 %
No-ladrido 6	0.48 %
Ladrido 1	99.99 %
Ladrido 2	99.99 %
Ladrido 3	99.99 %
Ladrido 4	99.99 %

Tabla 5-6: Precisión utilizando media y desviación estándar

Precisión: 98.32 %	
M + DE	
Audio	Probabilidad de que sea un ladrido
No-ladrido 1	0.83 %
No-ladrido 2	37.26 %
No-ladrido 3	6.43 %
No-ladrido 4	1.87 %
No-ladrido 5	7.31 %
No-ladrido 6	0.11 %
Ladrido 1	99.99 %
Ladrido 2	99.42 %
Ladrido 3	99.99 %
Ladrido 4	99.99 %

Tabla 5-7: Precisión utilizando mediana y desviación estándar.

Precisión: 98.99 %	m + M + DE
Audio	Probabilidad de que sea un ladrido
No-ladrado 1	0.72 %
No-ladrado 2	53.17 %
No-ladrado 3	11.41 %
No-ladrado 4	25.20 %
No-ladrado 5	6.04 %
No-ladrado 6	0.64 %
Ladrado 1	99.99 %
Ladrado 2	99.73 %
Ladrado 3	99.36 %
Ladrado 4	99.99 %

Tabla 5-8: Precisión utilizando media, mediana y desviación estándar.

combinación de valores estadísticos establecidos, poner un límite de 80 %, es decir, todas las probabilidades con un resultado menor a ese valor, se considerará parte de la clase de no-ladridos.

Por último, se creó un archivo “pkl” donde estará toda la información del modelo, misma que podrá ser ejecutada en la Raspberry Pi, sin ningún inconveniente y poder enviar notificaciones por medio de OneSignal.

5.5. Análisis del aprendizaje de máquina

5.5.1. Curva ROC (Receiver Operating Characteristics)

Con las mejores configuraciones obtenidas, es decir, la mejor combinación de valores estadísticos y los parámetros del SVM, sirvió para tener un mejor modelo de aprendizaje de máquina, con los 2980 audios, se separó en 70 % para re-entrenar el modelo y el 30 % para realizar los análisis con el modelo re-entrenado, y verificar su comportamiento contra los falsos positivos; se graficó la curva ROC, que permite visualizar qué tanto puede fallar el modelo. En la Figura 5-9, se puede observar la curva, en esa misma figura, todas las curvas que se grafican por encima de la línea punteada se consideran como una mejor

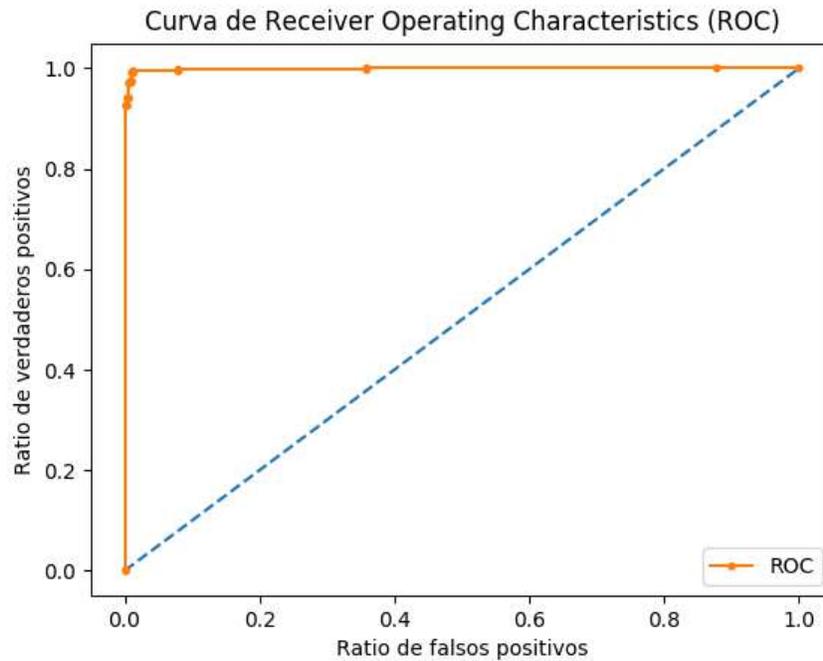


Figura 5-9: Curva ROC (Receiver Operating Characteristics).

clasificación, en caso contrario, si se encontrara alguna gráfica por debajo de la línea, existe una mala clasificación, más propensa a falsos positivos.

5.5.2. Matriz de confusión

Se obtuvo la matriz de confusión para determinar si el modelo puede confundir los datos. En la Tabla 5-9, se observa la matriz; los valores del lado que dice “Actual”, se refiere a los datos reales y en la “Predicción” es el resultado que arroja el algoritmo usando del modelo SVM; un buen resultado en la matriz de confusión, es tener un valor grande en las celdas donde coincidan los datos de “Actual” con los de “Predicción”, en caso contrario significa que el modelo SVM no es el adecuado.

Como se puede observar en los análisis, tiene un buen comportamiento para evitar múltiples falsos positivos, no quiere decir que sea perfecto, ya que en una baja probabilidad puede fallar.

		Actual	
		0	1
Predicción	0	414	6
	1	2	472

Tabla 5-9: Matriz de confusión.

5.6. Ajustes finales

Se realizaron algunos ajustes finales para dar soporte al modelo de predicción final obtenido del algoritmo de aprendizaje de máquina. Dado que los ladridos pueden variar entre una fracción de segundo hasta alrededor de un segundo se buscó una forma de procesar de manera simultánea los audios que captura el ReSpeaker para evitar ladridos incompletos; se ejecutaron 4 rutinas casi idénticas en la forma de escuchar sonidos, la diferencia está en el tiempo de escucha por cada determinado tiempo; el proceso a manera de ejemplo es como sigue; dentro de 1.5 segundos, la rutina 1 procesará cada 0.2 segundos; la rutina 2 lo hará cada 0.5 segundos, la rutina 3 lo hará 0.8 segundos y la rutina 4 cada 1 segundo; en cada rutina se creará un arreglo de resultados de probabilidades de predicción de la clase ladrido del modelo. En la Figura 5-10, se puede apreciar el proceso mencionado. Con esos arreglos se obtendrán un promedio de probabilidades que se compararán entre esas cuatro rutinas y se decidirá si dentro de ese periodo hubo un ladrido, además de verificar si fueron ladridos consecutivos para mandar notificaciones de manera adecuada. En la Figura 5-11, se observa gráficamente el proceso explicado anteriormente.

Otra característica que se programó es la auto-ejecución del sistema, añadiendo unas líneas de código en el archivo “rc.local”, que se ejecuta al iniciar el sistema operativo, esto con el fin de proporcionar al usuario la facilidad de ejecutar el prototipo con solo conectar la RPI a la corriente.

Gracias al código abierto del ReSpeaker, se programó una secuencia de luces que permite indicar el encendido del producto y el inicio de la ejecución del algoritmo principal.

Finalmente la estructura de los archivos usados para este trabajo se puede apreciar en la Fig. 5-12; donde “rc.local” es un archivo que se ejecuta al iniciar el SO de Raspberry Pi,

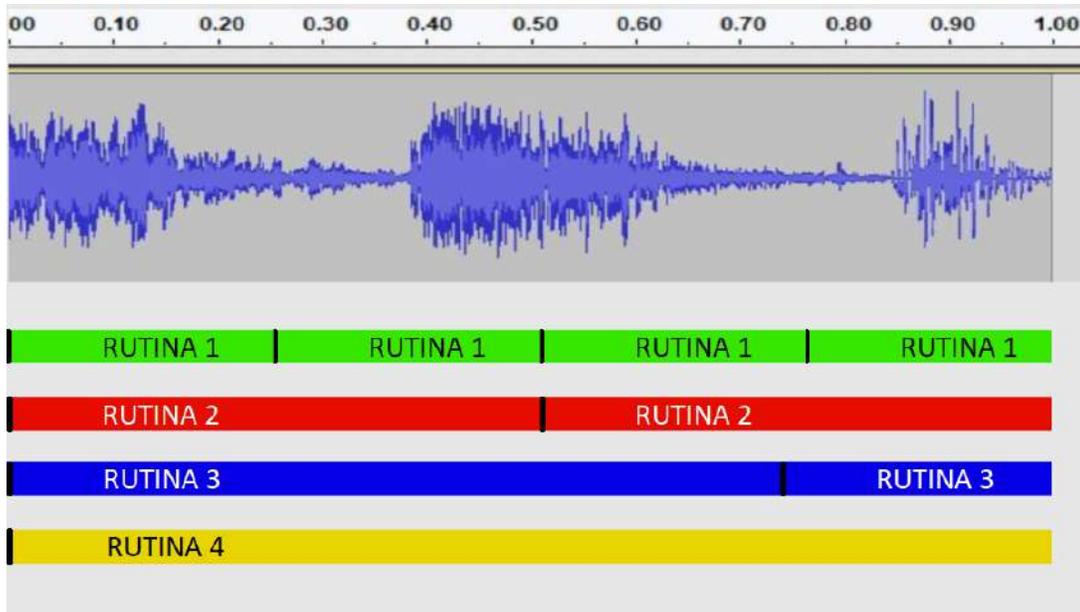


Figura 5-10: Procesamiento de audio en tiempo real.

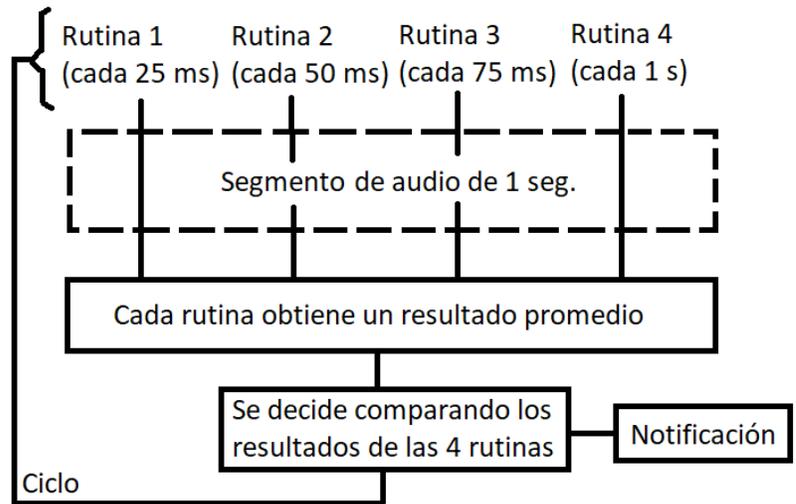


Figura 5-11: Diagrama de notificación que depende de una decisión.

“Código.py” es el algoritmo que ejecuta el SVM e inicializa el ReSpeaker y “modelo.pkl”, el modelo SVM de los ladridos de perros.



Figura 5-12: Localización de los archivos.

5.7. Pruebas al prototipo

Se realizó una encuesta para evaluar la usabilidad del prototipo y obtener opiniones de las PcDA para poder mejorar a futuro el dispositivo, previamente se les explicó el código de colores de las luces, esas preguntas fueron las siguientes:

- ¿La intensidad y colores de las luces le parecen óptimos?
- ¿El diseño del prototipo le pareció adecuado?
- ¿Las notificaciones fueron claras?
- Agregue algún comentario u opinión acerca del prototipo que pudiera mejorar para su uso por personas con discapacidad auditiva.

Esa encuesta fueron contestadas por 5 PcDA (3 hombres y 2 mujeres), de diferente género, edad y pérdida auditiva, para tener una conclusión más general.

Para la realización de las pruebas, se les pidió a los participantes estar en 4 posiciones diferentes:

- Sentado al frente del prototipo, con el celular en mano. Esto es para determinar si ambos tipos de notificaciones están sincronizados y verificar si la intensidad de la luz es adecuada. Ver en la Figura 5-13.



Figura 5-13: Posición 1.

- Sentado de lado al prototipo, con el celular en mano. Para verificar si puede captar la luz del prototipo, mientras se envía la notificación al dispositivo móvil. Ver en la Figura 5-14.
- Sentado a espaldas del prototipo, con el celular en mano. Para determinar si la notificación por texto es adecuado, independientemente de la luz. Ver en la Figura 5-15.
- Sentado de lado, sin el celular. Para verificar si la intensidad de luz y el color son adecuados. Ver en la Figura 5-16.

5.7.1. Resultados

Los usuarios se mostraron satisfechos con el prototipo, en su mayoría le pareció adecuado las notificaciones, en la Figura 5-17, se puede observar que no hubo una opinión negativo con el objeto desarrollado.



Figura 5-14: Posición 2.



Figura 5-15: Posición 3.

Entre las opiniones de los usuarios se encontró que:

- Más luces alrededor del prototipo.
- Aunque si enviaba notificaciones a tiempo, prefirieron que sea más rápida.



Figura 5-16: Posición 4.



Figura 5-17: Resultado de satisfacción sobre el prototipo.

Y en general, les pareció adecuado los colores de las luces, el tamaño y las notificaciones.

En las Figuras 5-18 y 5-19, se pueden observar la capacidad de reconocimiento de ladridos y de los no-ladridos; además en la “Rutina 1” del audio de no-ladrado se obtuvo algunos falsos positivos, sin embargo, con la ayuda de las demás rutinas, se puede eliminar

ese falso positivo para asegurar un buen reconocimiento de ladridos de perros.

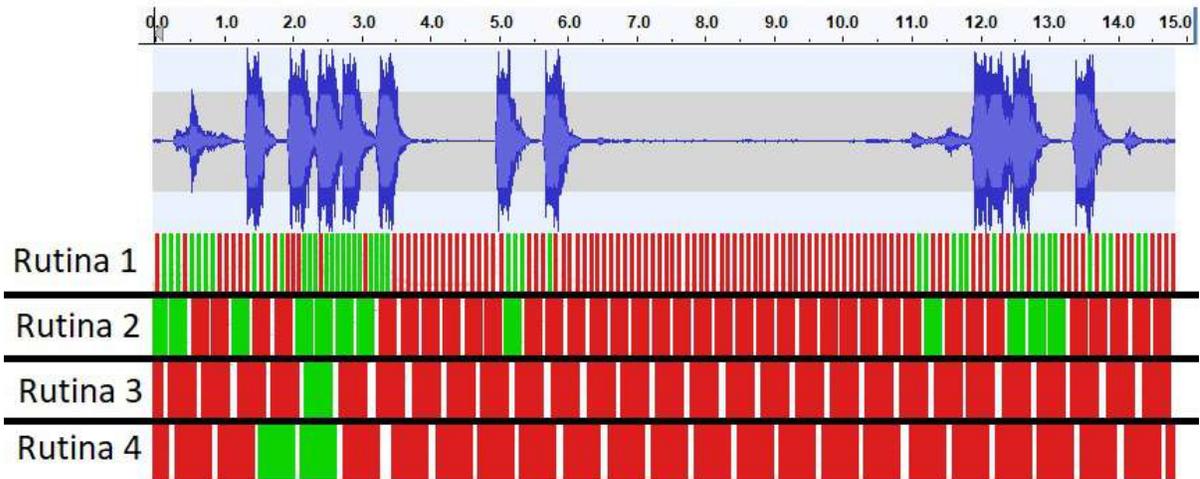


Figura 5-18: Capacidad de reconocimiento en un audio de ladridos.

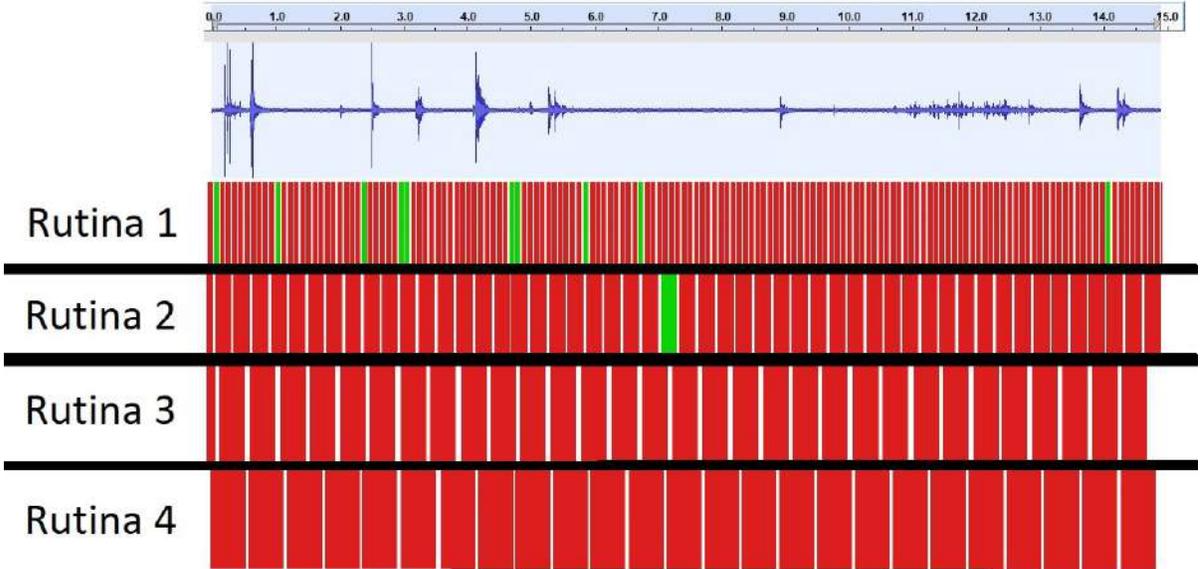


Figura 5-19: Capacidad de reconocimiento en un audio de no-ladridos.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo de tesis se diseñó e implementó un dispositivo capaz de reconocer de manera automática un ladrido de perro como una herramienta de asistencia a personas con discapacidad auditiva. Se revisaron trabajos similares y se determinó el uso del algoritmo de aprendizaje automático conocido como SVM para las tareas de clasificación de audios.

Previo a la implementación del algoritmo de aprendizaje de máquina en una tarjeta Raspberry Pi 3, se realizó una encuesta que permita conocer las necesidades del usuario e investigar los requerimientos del prototipo para ofrecer una mejor experiencia de uso y confiabilidad.

Se investigaron las librerías de software candidatas para el procesamiento de audio y se determinó aquella que cumplía con el objetivo de este trabajo, sobre todo que pudiera procesar en tiempo real, en este caso fueron Librosa y PyAudioAnalysis, luego se comparó la velocidad en su ejecución; en el caso de Librosa tuvo mejor tiempo de procesamiento, sin embargo, su instalación no fue sencilla en la Raspberry Pi, por lo que se optó por PyAudioAnalysis. El mismo proceso se realizó para las librerías de notificaciones, en la que se enfatizó la disponibilidad y su facilidad de implementación, es por eso que se

seleccionó OneSignal.

Como se mencionó en el capítulo 3, el SVM es el más usado para el reconocimiento de ladridos y por medio la librería libsvm permitió obtener los resultados para verificar la eficiencia del modelo final, misma que se crea un archivo listo para ser usado para reconocimiento; recordando que el proceso para la creación del archivo “pkl” fueron realizados en una computadora portátil, ya que por la limitada capacidad de la Raspberry Pi no fue posible entrenar una base de datos grande, y se puede confirmar por las características de la placa y de la computadora vistas en el capítulo 4.

Con respecto a los trabajos realizados por otros autores, en la parte de asistencia a PcDA, se puede asimilar con el trabajo de Yaganoglu y Köse [2018], usando la Raspberry Pi para notificar al usuario, con la diferencia que en este trabajo se enfocó en enviar un aviso a un dispositivo móvil, en lugar de una vibración.

En el caso de reconocimiento de ladridos, a diferencia de los trabajos presentados en el Capítulo 3, en este trabajo se enfocó en detectar ladridos en general; se usó el modelo final en una Raspberry Pi, como lo hizo Barris [2015], aunque no con el mismo objetivo; se usó SVM como la mayoría de los trabajos relacionados estudiados, se usaron las características usadas por la mayoría de los autores (MFCC, ZCR, cromagramas y espectrogramas), sin embargo, se usaron características aparte como se vió en el capítulo 5.

Se requirió más tiempo, para seleccionar las mejores librerías, y verificar su complejidad en su uso e instalación en la Raspberry Pi, y las mejores características, para lograr una buena integración de tecnologías para su uso en el contexto de AAL; además se resolvió los problemas que se presentaron durante su desarrollo como son las rutinas para evitar captar audios incompletos en tiempo real, y se mejoró la extracción de características utilizando la mediana y la desviación estándar, para asegurar un buen resultado.

6.2. Trabajo a futuro

Para seguir mejorando el prototipo, se propone coleccionar más tipos de audio como son, sonido de claxon, golpe de puerta, pitidos de electrodomésticos, tono del teléfono, etc., y para entrenar el modelo, realizar los mismos procesos para obtener la precisión, es decir, encontrar el mejor kernel y posteriormente las mejores características de un vector. Por cuestiones de tiempo no se realizó una comparación con otro método de aprendizaje como las Redes Neuronales, sin embargo se esperaba que las redes neuronales recurrentes tengan un buen desempeño tal como lo han hecho con el reconocimiento de voz humana.

Se propone integrar como característica adicional la de determinar el origen del sonido y la distancia, así ofrecer al usuario, es decir a la PcDA, la capacidad de conocer con más detalle sobre lo que ocurre a su alrededor y probarlo con más PcDA para un mejor resultado de satisfacción y necesidades.

Una posibilidad más ambiciosa de trabajo futuro, es integrar esas tecnologías en un ambiente biológico, para ofrecer prototipos amigables al medio ambiente, ya que en su mayoría estará construida a base de bioquímicos, un ejemplo de eso es la que presenta Kassal *et al.* [2018], sobre lograr una comunicación wireless con biosensores; pero a la vez ofrecer sensores que tengan las mismas o mejores funciones que las convencionales. Para este trabajo, el biosensor puede mejorar la calidad de los audios, para un mejor reconocimiento de sonidos.

Bibliografía

- BARRIS, T. *Automatic Dog Barking Recognition*. Thesis, University of Cape Town (2015)
- BOERSMA, P. Y WEENINK, D. Praat: doing phonetics by computer [Computer program]. Version 6.0.43, retrieved 8 September 2018 <http://www.praat.org/> (2018)
- CHANG, C.C. Y LIN, C.J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2**:27:1–27:27 (2011). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- EYBEN, F., WENINGER, F., GROSS, F., Y SCHULLER, B. Recent Developments in openSMILE, the Munich Open-source Multimedia Feature Extractor. En *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, págs. 835–838. ACM, New York, NY, USA (2013)
- FANZERES, L., VIVACQUA, A., Y BISCAINHO, L. Mobile Sound Recognition for the Deaf and Hard of Hearing (2018)
- FARAGÓ, T., PONGRÁCZ, P., RANGE, F., VIRÁNYI, Z., Y ÁDÁM MIKLÓSI. ‘The bone is mine’: affective and referential aspects of dog growls. *Animal Behaviour* **79**(4):917 – 925 (2010)
- GIANNAKOPOULOS, T. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. *PloS one* **10**(12) (2015)

- HANTKE, S., CUMMINS, N., Y SCHULLER, B. What is my Dog Trying to Tell Me? the Automatic Recognition of the Context and Perceived Emotion of Dog Barks. págs. 5134–5138 (2018)
- JADOUL, Y., THOMPSON, B., Y DE BOER, B. Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics* **71**:1 – 15 (2018)
- KASSAL, P., STEINBERG, M.D., Y STEINBERG, I.M. Wireless chemical sensors and biosensors: A review. *Sensors and Actuators B: Chemical* **266**:228 – 245 (2018)
- McFEE, B., McVICAR, M., BALKE, S., LOSTANLEN, V., THOMÉ, C., RAFFEL, C., LEE, D., LEE, K., NIETO, O., ZALKOW, F., ELLIS, D., BATTENBERG, E., YAMAMOTO, R., MOORE, J., WEI, Z., BITTNER, R., CHOI, K., NULLMIGHTYBOFO, FRIESCH, P., STÖTER, F.R., THASSILO, VOLLRATH, M., GOLU, S.K., NEHZ, WALOSCHEK, S., SETH, NAKTINIS, R., REPETTO, D., HAWTHORNE, C.F., Y CARR, C. librosa/librosa: 0.6.3 (2019)
- MIELKE, M. Y BRÜCK, R. Design and evaluation of a smartphone application for non-speech sound awareness for people with hearing loss. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* págs. 5008–5011 (2015)
- MOLNÁR, C., KAPLAN, F., ROY, P., PACHET, F., PONGRÁCZ, P., DÓKA, A., Y MIKLÓSI, A. Classification of dog barks: A machine learning approach. *Animal Cognition* **11**(3):389–400 (2008)
- MONEKOSSO, D., FLÓREZ-REVUELTA, F., Y REMAGNINO, P. Ambient Assisted Living [Guest editors' introduction]. *IEEE Intelligent Systems* **30**:2–6 (2015)
- NANAYAKKARA, S., WYSE, L., ONG, S., Y A. TAYLOR, E. Enhancing Musical Experience for the Hearing-Impaired Using Visual and Haptic Displays. *Human-computer Interaction* **28** (2012)

- NIELSEN, J. Usability Inspection Methods. En *Conference Companion on Human Factors in Computing Systems*, CHI '94, págs. 413–414. ACM, New York, NY, USA (1994)
- PEREZ-ESPINOSA, H., REYES-MEZA, V., AGUILAR-BENITEZ, E., Y M. SANZÓN-ROSAS, Y. Automatic individual dog recognition based on the acoustic properties of its barks. *Journal of Intelligent Fuzzy Systems* **34**:3273–3280 (2018)
- PONGRÁCZ, P., MOLNÁR, C., Y ÁDÁM MIKLÓSI. Barking in family dogs: An ethological approach. *The Veterinary Journal* **183**(2):141 – 147 (2010)
- PÉREZ-ESPINOSA, H., MARTÍN PÉREZ-MARTÍNEZ, J., JOSÉANGEL DURÁN-REYNOSO, J., Y REYES-MEZA, V. Automatic Classification of Context in Induced Barking. *Research in Computing Science* **100**:63–74 (2015)
- SADRI, F. Ambient Intelligence: A Survey. *ACM Comput. Surv.* **43**(4):36:1–36:66 (2011)
- SAKAMOTO, S., HOSHINO, A., SUTOU, K., Y SATO, T. Estimating Sound-Absorption Coefficient and Transmission Loss by the Dimensions of Bundle of Narrow Holes (Comparison between Theoretical Analysis and Experiments). *TRANSACTIONS OF THE JAPAN SOCIETY OF MECHANICAL ENGINEERS Series C* **79**:4164–4176 (2013)
- SÁNCHEZ, B. *METODOLOGÍA PARA LA CREACIÓN DE INTERFACES INTELIGENTES APLICADAS A AMBIENT ASSISTED LIVING (AAL)*. Thesis, Universidad de Cuenca (2018)
- TESSENDORF, B., TRÖSTER, G., DEBEVC, M., Y HOHMANN, V. *Multimodal Sensor and Actuator System for Hearing Instruments: A Dissertation Submitted to ETH Zurich for Degree of Doctor Od Sciences*. B. Tessedorf (2012)
- UTSUKI, T. *Accesibilidad en las viviendas de interés social del programa 'Casa justa' en Ciudad Cuncel, Mérida*. Thesis, Universidad Autónoma de Yucatan (2016)
- WIEHR, F., VOIT, A., WEBER, D., GEHRING, S., WITTE, C., KÄRCHER, D., HENZE, N., Y KRÜGER, A. Challenges in Designing and Implementing Adaptive Ambient

Notification Environments. En *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, págs. 1578–1583. ACM, New York, NY, USA (2016)

YAGANOGLU, M. Y KÖSE, C. Real-Time Detection of Important Sounds with a Wearable Vibration Based Device for Hearing-Impaired People (2018)

YAGANOGLU, M. Y KÖSE, C. Wearable Vibration Based Computer Interaction and Communication System for Deaf. *Applied Sciences* **7**:1296 (2017)