|  |  |
| --- | --- |
|  | **Universidad Católica Andrés Bello**  **Facultad de Ingeniería**  **Escuela de Ingeniería Informática** |

# Sistema de Monitoreo Acústico, para Identificar Sonidos y Generar Alertas de Emergencia

**Trabajo de Grado**

Presentado ante la

**UNIVERSIDAD CATÓLICA ANDRÉS BELLO**

Como parte de los requisitos para optar al título de

**INGENIERO EN INFORMÁTICA**

|  |  |
| --- | --- |
| Realizado por | Naim Arcoiza, Carmelo Jesús  Sotillo Vallejo, César Enrique |
| Tutor | Larez Mata, Jesús José |
| Fecha | Marzo, 2025 |

**Dedicatoria**

Bla bla bla

**Agradecimiento**

Bla bla bla

**Índice de contenidos**

[**Resumen 7**](#_Toc184848720)

[**Introducción 8**](#_Toc184848721)

[**Capitulo I. El Problema 9**](#_Toc184848722)

[Planteamiento del Problema 9](#_Toc184848723)

[Objetivo General 9](#_Toc184848724)

[Objetivos Específicos 9](#_Toc184848725)

[Alcance 9](#_Toc184848726)

[Limitaciones 10](#_Toc184848727)

[Justificación 11](#_Toc184848728)

[**Capitulo II. Marco Metodológico 12**](#_Toc184848729)

[Tipo de Investigación 12](#_Toc184848730)

[Según Diseño de la Investigación 12](#_Toc184848731)

[Según el Nivel de Investigación 12](#_Toc184848732)

[Población (opcionales) 12](#_Toc184848733)

[Muestra (opcionales) 12](#_Toc184848734)

[Sistema de Variables (opcionales) 12](#_Toc184848735)

[Tecnologías e Instrumentos de Recolección de Datos 12](#_Toc184848736)

[Metodología de Desarrollo Utilizada 12](#_Toc184848737)

**Índice de Tablas**

bla bla bla

**Índice de imágenes**

Bla bla bla

|  |  |
| --- | --- |
|  | **Universidad Católica Andrés Bello**  **Facultad de Ingeniería**  **Escuela de Ingeniería Informática** |

# Sistema de Monitoreo Acústico, para Identificar Sonidos y Generar Alertas de Emergencia

|  |  |
| --- | --- |
| Autor: | Naim Arcoiza, Carmelo Jesus  Sotillo Vallejo, César Enrique |

|  |  |
| --- | --- |
| Tutor Académico: | Bello Castillo, Franklin Bismar |
| Tutor Empresarial: | Medina Cuida, Luz Esperanza |
| Fecha: | Marzo, 2024 |

# 

# Resumen

Bla bla bla bla bla y bla

*Palabras clave:* bla, bla, bla. y bla,

**NOTA**

Exposición del problema, objetivo, enfoque teórico, aspectos metodológicos, resultados y conclusiones.

Debe ser: ***preciso***: reflejar el propósito y contenido del informe, sin incluir información que no aparezca en el informe; ***coherente, legible y conciso***

En un solo párrafo, se utiliza interlineado sencillo, SIN sangría. Extensión máxima de **250** palabras

# Introducción

Este trabajo de investigación propone un **sistema distribuido** para la **detección temprana** de emergencias mediante el uso de **inteligencia artificial**, enfocada en proteger no solo a **personas mayores o con discapacidades subyacentes, sino también, a personas que a raíz de cualquier situación, quedaron en una situación de vulnerabilidad**. El sistema utiliza dispositivos que capturan **sonidos**, clasificándolos y detectando palabras clave relevantes.

Esta información se envía en tiempo real a un servidor central donde se analizan patrones y anomalías con técnicas predictivas mediante **Prophet**. La **importancia** de este proyecto radica en su capacidad de prevenir que se agraven las consecuencias ante un evento que comprometa la salud de un individuo si no se detectan a tiempo, ofreciendo un **monitoreo no invasivo** que respeta la privacidad. Además, beneficia a **familias, cuidadores, instituciones públicas y privadas** y facilita una intervención más rápida de los **servicios de emergencia**, incluso en casos como **violencia doméstica o el monitoreo de personas con depresión**. El **propósito** es crear un entorno más seguro mediante una identificación de situaciones peligrosas y alertar a quienes puedan ofrecer ayuda, contribuyendo a una mejor calidad de vida y una respuesta efectiva ante emergencias.

Para el desarrollo del sistema, se empleó la metodología espiral, que permite un enfoque iterativo y flexible. Cada ciclo del proceso abarca fases de planificación, análisis de riesgos, desarrollo y evaluación, permitiendo ajustes continuos en el diseño y la implementación del sistema. Esta metodología es ideal para proyectos con incertidumbre, ya que facilita la evolución del sistema según la retroalimentación obtenida, asegurando su mejora progresiva a lo largo del desarrollo.

El presente trabajo está estructurado en cinco capítulos, En el Capítulo I, referido al Planteamiento del Problema, se describe la problemática, se establecen los objetivos, justificación, alcance y limitaciones del proyecto. En el Capítulo II, se expone el Marco Teórico donde se recopilan los antecedentes y las bases teóricas que sustentan el trabajo. En el Capítulo III, se presenta el Marco Metodológico, que describe el tipo de investigación, técnicas e instrumentos de recolección de datos, metodología de desarrollo y el procedimiento metodológico. En el Capítulo IV, se expone el Desarrollo y Resultados, donde se describe como el procedimiento metodológico dio respuesta a cada uno de los objetivos planteados, y en el Capítulo V, se presentan las conclusiones y recomendaciones sobre el trabajo realizado. Finalmente se listan las referencias bibliográficas utilizadas, seguidas de los anexos y apéndices.

**NOTA**

* ~~Proporciona una perspectiva del trabajo de investigación.~~
* ~~importancia de la investigación,~~
* ~~propósito,~~
* enfoque teórico,
* ~~metodología utilizada.~~
* ~~Finaliza con la organización del trabajo, escrito en un solo párrafo.~~

# Capitulo I. El Problema

## Planteamiento del Problema

### Objetivo General

Desarrollar un sistema para generar alertas en casos de emergencia basado en el monitoreo acústico.

### Objetivos Específicos

1. Analizar los conceptos asociados a la analítica de sonidos ambientales identificando los conceptos necesarios para diseñar el sistema.
2. Diseñar un sistema para generar alertas en casos de emergencia basado en el monitoreo acústico en función del análisis realizado.
3. Implementar el sistema para generar alertas en casos de emergencia basado en el monitoreo acústico según el diseño realizado.
4. Validar el sistema para generar alertas en casos de emergencia basado en el monitoreo acústico con respecto al análisis realizado.
5. Elaborar la documentación técnica y guías del usuario y sistema.

## Alcance

El proyecto tiene como objetivo desarrollar un sistema de monitoreo acústico que perfile el comportamiento de los habitantes de una vivienda, a través del reconocimiento de los sonidos ambientales en las diferentes estancias de la misma. El sistema generará alarmas en tiempo real que alertarán a contactos de emergencia predefinidos, estas notificaciones podrán tener múltiples niveles dependiendo del grado de incertidumbre de la situación. Para ello se llevará a cabo un análisis de los conceptos asociados a la analítica de sonidos donde se identificaran los conceptos necesarios para diseñar el sistema.

Con base en los resultados del análisis, se procederá a diseñar el sistema de monitoreo acústico capaz de reconocer sonidos. El diseño comprende la definición de la arquitectura del sistema, contemplando los componentes esenciales, como la red de micrófonos para la captura de sonidos y los mecanismos de alerta, detección de anomalías y de reconocimiento acústico. El diseño considerará la importancia de la privacidad de los usuarios. Según los resultados del análisis realizado para el desarrollo del sistema de monitoreo acústico, se evaluará si es necesario crear un dataset desde cero, obtener uno de internet y modificarlo según sea necesario, o bien emplear un modelo ya preentrenado.

Una vez completado el diseño, se procederá a la implementación del sistema de monitoreo acústico. Esta fase incluirá la configuración e instalación de los componentes físicos, como los micrófonos distribuidos en las estancias de la vivienda, asegurando una cobertura adecuada para la captura de sonidos relevantes. Los algoritmos de procesamiento de señales y los modelos de inteligencia artificial, definidos en la fase de diseño, serán desarrollados y adaptados para realizar el reconocimiento y clasificación de los sonidos.

Se desarrollarán los mecanismos de alerta y notificación, que emitirán avisos a los contactos de emergencia predefinidos, permitiendo la personalización de los niveles de alarma en función de la detección de anomalías o eventos específicos. Durante esta fase, se asegurará medidas de privacidad para proteger la información sensible capturada por los micrófonos.

Luego se realizará la validación del sistema, se llevarán a cabo pruebas funcionales en ambientes controlados y no controlados para comparar los resultados obtenidos con los objetivos planteados y realizar ajustes en caso de ser necesario.

Finalmente se elaborará la documentación del sistema, manuales de usuario, manuales de sistema, descripción de los componentes, diagramas y cualquier otro documento necesario.

## Limitaciones

En la implementación del sistema, se prevén ciertas limitaciones que podrían influir en el desempeño de su desarrollo. A continuación, se describirán algunas de las restricciones que se podrían presentar:

Limitaciones tecnológicas: La precisión del sistema puede verse afectada por la calidad de los sensores acústicos y la capacidad de procesamiento de los microcontroladores empleados.

Limitaciones de datos: La recolección de suficientes muestras de sonidos para entrenar el modelo puede ser limitada, lo que puede comprometer la eficacia del sistema en escenarios no previstos.

## Justificación

Bla bla bla

NOTA

Razones por las cuales se formula la investigación, debe reflejar los aportes más relevantes teóricos y/o metodológicos referidos al objeto de conocimiento, y dar cuenta de la contribución de la investigación al conjunto de conocimientos de un área y/o reflejar su utilidad, aportes o implicaciones prácticas, de valor teórico, relevancia social, alcance, entre otros *(****el porqué y el para qué del estudi*o**) e indicar quiénes se benefician con los resultados de la investigación. Puede incluir los aportes tecnológicos y funcionales de la misma.

# Capitulo II. Marco Metodológico

## Este capítulo presenta las bases teóricas y las investigaciones previas que sustentan el desarrollo del Sistema de Monitoreo Acústico para Identificar Sonidos y Generar Alertas de Emergencia. A través de la revisión de fuentes documentales, se exponen los conceptos y enfoques principales relacionados con el procesamiento de señales acústicas, la clasificación de sonidos mediante técnicas de inteligencia artificial y los sistemas de alertas automatizadas.

## El marco teórico no solo proporciona un contexto académico para el proyecto, sino que también justifica las decisiones técnicas y metodológicas adoptadas en el diseño del sistema. En las siguientes secciones se detallan los antecedentes investigativos, los modelos de IA aplicados a la clasificación de sonidos y los fundamentos teóricos que permiten la generación de alertas en situaciones de emergencia.

## En las siguientes secciones se detallan los antecedentes y los conceptos clave que sustentan este trabajo.

## Antecedentes de investigación

**Real-time Audio Classification on an Edge Device**

Christoffer y David (2021) Abordan la implementación de modelos de aprendizaje automatico en dispositivos edge para la clasificación de audio en tiempo real, Se centra en el uso del modelo YAMNet, que fue reentrenado para detectar eventos acústicos como disparos, rotura de cristales, animales y habla humana. Este modelo puede desplegarse en dispositivos edge en su versión completa con TensorFlow como en versiones optimizadas con TensorFlow Lite , con el objetivo de comparar la precisión, el tiempo de inferencia y el uso de memoria de cada variante​.

Con el propósito de evaluar el desempeño, trabajaron en una serie de experimentos en los que compararon la precisión del modelo en ambas versiones de TensorFlow. Se encontraron con que aunque existía una pérdida de precisión en la versión lite los resultados eran comparables a los de la versión completa. Esto implica que TensorFlow Lite es una opción viable y crea la posibilidad de trabajar con dispositivos de bajo consumo y recursos limitados como ESP32

Este enfoque no solo demuestra la factibilidad de implementar modelos de clasificación de audio en tiempo real en entornos con recursos restringidos, sino que también abre la puerta a futuras investigaciones que busquen optimizar aún más estos modelos. La capacidad de desplegar soluciones de inteligencia artificial en dispositivos edge amplía las posibilidades de aplicaciones en áreas como la seguridad, la vigilancia y el monitoreo ambiental, permitiendo respuestas más rápidas y reduciendo la dependencia de infraestructuras centralizadas.

## Bases Teóricas

### Dispositivos Edge:

El incremento en la demanda de los servicios y aplicaciones en las últimas décadas del uso de la Internet,ha contribuido a un fuerte aumento de los requisitos de procesamiento y almacenamiento de datos. Son diversos,en términos de los recursos que requieren las diferentes aplicaciones y, por lo tanto, a menudo invocan soluciones a medida (Doluí y Kanti Datta, 2017).

Según Medina (2019) Edge Device, en este contexto, se refiere a elementos con capacidades limitadas que tiene su propio conjunto de recursos: CPU, memoria, almacenamiento, y red. Pueden ser Smartphone, Smartglasses, smartwatches, tablets, routers, vehículos autónomos, o cualquier dispositivo de IoT con capacidad de proceso. En este escenario, Edge Computing, representa una solución para enfrentar y aliviar la carga de procesamiento y almacenamiento en la nube, en aplicaciones y tecnologías que requieren de un ancho de banda exponencial y una baja o nula latencia.

Shi, Cao, Zhang, Li y Xu (2016) definen Edge Computing como un paradigma de computación distribuida que acerca el procesamiento y almacenamiento de datos a la fuente de generación, es decir, al "borde" de la red, en lugar de enviar toda la información a la nube centralizada. Este enfoque permite reducir la latencia, optimizar el uso del ancho de banda y mejorar la eficiencia en aplicaciones que requieren respuestas en tiempo real, como el Internet de las Cosas (IoT), la realidad aumentada, los vehículos autónomos y las ciudades inteligentes. Edge Computing se basa en dispositivos periféricos (edge devices) y nodos locales que realizan tareas de procesamiento y almacenamiento, lo que reduce la dependencia de la infraestructura centralizada de la nube.

### Raspberry pi

Según Richardson y Wallace (2016), la Raspberry Pi es un dispositivo de computación de bajo costo y alto rendimiento que ha ganado popularidad en diversos campos debido a su versatilidad y facilidad de uso. Este dispositivo, del tamaño de una tarjeta de crédito, está equipado con un procesador ARM, memoria RAM, puertos de entrada/salida y conectividad de red, lo que lo hace ideal para proyectos educativos, de automatización y desarrollo de prototipos. La Raspberry Pi es capaz de ejecutar sistemas operativos basados en Linux, lo que permite a los usuarios programar y personalizar sus aplicaciones según sus necesidades (Richardson y Wallace, 2016).

Además, Upton y Halfacree (2020) destacan que la Raspberry Pi ha evolucionado significativamente desde su lanzamiento, con modelos más potentes como la Raspberry Pi 4, que ofrece mayores capacidades de procesamiento, almacenamiento y conectividad. Este modelo incluye soporte para redes Gigabit Ethernet, puertos USB 3.0 y salidas de video de alta definición, lo que lo convierte en una herramienta poderosa para aplicaciones más exigentes, como servidores domésticos, centros multimedia y sistemas embebidos avanzados (Upton y Halfacree, 2020).

### Python

Python es un lenguaje de programación poderoso, elegante y fácil de leer, diseñado para simplificar la creación de programas mediante una sintaxis clara y estructurada. Según el documento, Python destaca por su versatilidad en aplicaciones del mundo real, su enfoque en la legibilidad del código y su capacidad para integrar paradigmas como la programación orientada a objetos y funcional. Además, es software libre con una comunidad activa y una implementación estándar consolidada (Yuill & Halpin, 2006).

### Tensorflow

Según Goldsborought (2016) TensorFlow es una biblioteca de software de deep learning de código abierto desarrollada por Google que permite definir, entrenar y desplegar modelos de machine learning mediante la representación de algoritmos como grafos computacionales.

La librería TensorFlow opera construyendo un grafo computacional en el que cada nodo representa una operación (por ejemplo, una función matemática, una transformación o una capa de una red neuronal) y cada arista transporta un tensor, es decir, un arreglo multidimensional de datos. Esta arquitectura facilita diversas optimizaciones, como la eliminación de subgrafos redundantes, y permite distribuir la ejecución de la computación a lo largo de múltiples dispositivos (CPUs, GPUs, TPUs) e incluso en entornos distribuidos. De esta manera, se optimiza tanto el uso de memoria como el rendimiento, haciendo viable el entrenamiento y despliegue de modelos complejos a gran escala. (Goldsborought, 2016).

### NestJs

Según Sabo (2020) NestJS es un framework para el desarrollo de aplicaciones del lado del servidor basado en Node.js, que se escribe en TypeScript. Proporciona una estructura modular y escalable mediante el uso de patrones modernos como la inyección de dependencias, controladores y módulos, facilitando la creación de aplicaciones backend mantenibles y robustas.

NestJS aprovecha la solidez de Node.js y Express.js, pero se diferencia al introducir un enfoque inspirado en Angular para la organización de la aplicación. Gracias a su arquitectura basada en módulos, cada parte de la aplicación se encapsula en unidades independientes que facilitan la reutilización y la escalabilidad. Además, el framework implementa un avanzado sistema de inyección de dependencias, lo que permite gestionar y suministrar las instancias de servicios de manera automática, reduciendo el acoplamiento entre componentes y promoviendo un diseño orientado a pruebas. Otro pilar fundamental de NestJS es su fuerte integración con TypeScript, lo cual aporta tipificación estática y facilita la detección temprana de errores durante el desarrollo. La utilización de decoradores en NestJS permite agregar metadatos a clases, métodos y propiedades, lo que habilita la implementación de características como interceptores, pipes y controladores para la validación y transformación de datos, así como para el manejo de rutas HTTP. Además, la herramienta Nest CLI agiliza la generación de nuevos proyectos y componentes, garantizando que se siga una estructura coherente en toda la aplicación, lo que resulta especialmente útil en proyectos complejos y colaborativos (Sabo, 2020).

### Inteligencia artificial

La Inteligencia Artificial (IA) se define como el estudio de agentes que perciben su entorno a través de sensores y actúan sobre él mediante actuadores, con el objetivo de maximizar su utilidad esperada. Según Russell y Norvig (2022), " La IA es el estudio de agentes que reciben percepciones del entorno y realizan acciones. Cada uno de estos agentes implementa una función que asigna secuencias de percepción a acciones, y cubrimos diferentes formas de representar estas funciones para lograr el mejor resultado esperado." (p. 19).

Los fundamentos de la IA se entrelazan con múltiples disciplinas. La filosofía aporta marcos éticos y lógicos, como señalan los autores: "El filósofo griego Aristóteles fue uno de los primeros en intentar codificar el "pensamiento correcto" sus silogismos proporcionaron patrones para estructuras argumentales que siempre produjeron conclusiones correctas." (p. 21). Las matemáticas y la estadística proporcionan herramientas para el razonamiento probabilístico: "La probabilidad rápidamente se convirtió en una parte invaluable de las ciencias cuantitativas, ayudando a lidiar con mediciones inciertas y teorías incompletas." (p. 26). La economía contribuye con teorías de decisión y utilidad: "La teoría de la decisión, que combina la teoría de la probabilidad con la teoría de la utilidad, proporciona un marco formal y completo para las decisiones individuales tomadas en condiciones de incertidumbre." (p. 28). La neurociencia y la psicología inspiran modelos cognitivos: "El campo interdisciplinario de la ciencia cognitiva reúne modelos informáticos de la IA y técnicas experimentales de la psicología para construir teorías precisas y comprobables de la mente humana." (p. 21). Por último, la ingeniería y la computación permiten implementar sistemas eficientes: "La historia de la IA es también la historia del diseño de arquitecturas cada vez más sofisticadas para programas de agentes." (p. 65).

La IA actual ha alcanzado hitos significativos en diversos dominios. De acuerdo a los expuesto por Russell & Norvig (2022): “Los sistemas que usan IA han alcanzado o superado el rendimiento humano en ajedrez, Go, póquer, Pac-Man, Jeopardy, detección de objetos ImageNet, reconocimiento de voz y diagnóstico de retinopatía diabética." (p. 46). En aplicaciones prácticas, destacan los vehículos autónomos: "Los vehículos de prueba de Waymo superaron la marca de 10 millones de millas recorridas en vías públicas sin sufrir accidentes graves." (p. 47), y sistemas de diagnóstico médico: "Los algoritmos de IA ahora igualan o superan a los médicos expertos en el diagnóstico de muchas afecciones, como el cáncer metastásico y las enfermedades oftálmicas." (p. 48). Además, herramientas como "Los sistemas de traducción automática ahora permiten la lectura de documentos en más de 100 idiomas, lo que genera cientos de miles de millones de palabras por día." (p. 47) evidencian su impacto global. No obstante, persisten retos éticos y técnicos, como la alineación de valores humanos y la escalabilidad en entornos complejos, que definen la frontera actual de investigación. (corregir)

### Rms

### Cadenas de Markov

### Eventos discretos

### Agente

## Bases Legales (si aplica).

## Terminología Básica (si aplica)

# Referencias Bibliográficas

Malmberg, C., Radszuweit, D. (2021). *Real-time Audio Classification on an Edge Device - Using YAMNet and TensorFlow Lite.*

Doluí, K., & Kanti Datta, S. (2017). *Comparison of Edge Computing implementations: Fog computing, cloudlet and mobile Edge Computing.* Global Internet of Things Summit (GloTS). Recuperado de: <https://doi.org/10.1109/giots.2017.8016213>

Medina, M. (2019). Edge Computing para IoT. Recuperado de: <https://openaccess.uoc.edu/bitstream/10609/91207/7/mmedinabarTFM0119memoria.pdf>

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. IEEE Internet of Things Journal, 3(5), 637-646. Recuperado de: <https://doi.org/10.1109/JIOT.2016.2579198>

Richardson, M., & Wallace, S. (2016). *Getting Started with Raspberry Pi: An Introduction to the Fastest-Selling Computer in the World (4th ed.).* Maker Media, Inc. Recuperado de: <https://books.google.co.ve/books/about/Getting_Started_With_Raspberry_Pi.html?id=w4CkDAAAQBAJ&redir_esc=y>

Upton, E., & Halfacree, G. (2020). *Raspberry Pi User Guide (4th ed.)*. John Wiley & Sons. Recuperado de: <https://www.perlego.com/book/997788/raspberry-pi-user-guide-pdf>

Yuill, S., Halpin, H. (2006). *Python.* Recuperado de: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97bfafd514ca2abb7e2c5c86bb>

Goldsborough, P. *A Tour of TensorFlow, Proseminar Data Mining*. Technische Universität München. Recuperado de: <https://arxiv.org/pdf/1610.01178>

Sabo, M. (2020). NestJS. Završni rad. Josip Juraj Strossmayer University of Osijek, Departamento de Matemáticas. Recuperado de: <https://zir.nsk.hr/islandora/object/mathos:441>

Russell, S., & Norvig, P. (2022). *Artificial Intelligence: A Modern Approach (4th ed.)*. Pearson. Recuperado de: <http://lib.ysu.am/disciplines_bk/efdd4d1d4c2087fe1cbe03d9ced67f34.pdf>