**Informe Final**

***“Reconocimiento de Problemas Automotrices con ayuda de Redes Neuronales”***

Bonomi Ariel Matías

Universidad del Salvador

Ingeniería Informática

Proyecto final de Ingeniería Informática

# Índice

[Índice](#_heading=h.44sinio) **2**

[**Abstract**](#_heading=h.4xct5jrqsevl) **4**

[**Capítulo 1. Introducción**](#_heading=h.z337ya) **5**

[**Capítulo 2. Metodología y procedimientos**](#_heading=h.3j2qqm3) **9**

[**Capítulo 3. Síntesis de literatura consultada**](#_heading=h.1y810tw) **16**

[Problemas en Automóviles](#_heading=h.4i7ojhp) **30**

[Comportamiento del sonido y espectrogramas](#_heading=h.2xcytpi) **32**

[Detección de sonidos](#_heading=h.1ci93xb) **35**

[Almacenamiento de Sonidos (AudioSet)](#_heading=h.3whwml4) **39**

[Minería de datos](#_heading=h.2bn6wsx) **40**

[Redes Neuronales Artificiales](#_heading=h.qsh70q) **43**

[Clasificación de Imágenes](#_heading=h.3as4poj) **46**

[Clasificación supervisada](#_heading=h.1pxezwc) 46

[Clasificación no supervisada](#_heading=h.49x2ik5) 46

[**Capítulo 4. Justificación técnico-económica**](#_heading=h.2p2csry) **47**

[Justificación técnica](#_heading=h.147n2zr) **47**

[Justificación económica](#_heading=h.3o7alnk) **50**

[Supuestos y restricciones](#_heading=h.23ckvvd) **52**

[Riesgos y contingencias](#_heading=h.ihv636) **52**

[**Capítulo 5. Presentación de Resultados**](#_heading=h.32hioqz) **54**

[Requerimientos Funcionales](#_heading=h.1hmsyys) 55

[Requerimientos No Funcionales](#_heading=h.41mghml) 56

[Casos de Uso](#_heading=h.2grqrue) 58

[Diagrama de Actividad](#_heading=h.vx1227) 63

[Pantallas](#_heading=h.3fwokq0) 65

[Proceso de desarrollo:](#_heading=h.1v1yuxt) 68

[**Capítulo 6. Implicancias, conclusiones y recomendaciones**](#_heading=h.4f1mdlm) **82**

[Conclusiones](#_heading=h.2u6wntf) 82

[Futuras líneas de investigación](#_heading=h.19c6y18) 83

[BIBLIOGRAFÍA](#_heading=h.3tbugp1) **84**

Índice de Imágenes

Figura 1. Espectrograma

Figura 2. Pasos de Metodología

Figura 3. Metodología Planteada

Figura 4.1. Tabla de Efectividad

Figura 5.1. Motores en estado ralenti

Figura 5.2. Motores en aceleración

Figura 5.3. Comparación

Figura 6.1. Conversión de una señal continua a discreta

Figura 6.2. Diagrama de dispersión

Figura 7. Ejemplo de Espectrograma

Figura 8. Estructura de micrófono analógico

Figura 9. Estructura de micrófono digital

Figura 10. Red Neuronal Artificial

Figura 11. Neuronal Artificial

Figura 12. Tabla de Componentes Arduino

Figura 12.1. Tabla de Riesgos

Figura 13. Cómo utilizar Teachable Machine

Figura 13.1. Cargar Imágenes

Figura 13.2. Exportar Modelo

# Abstract

Hoy en día el 79% de transacciones realizadas en Argentina son de autos usados, autos que rondan los 100.000 - 200.000 km.

Cuanto los autos llegan a estos kilometrajes y más siendo autos de gama media/baja como en este caso, estos suelen empezar a presentar distintos problemas.

Debido al muy bajo conocimiento de la población Argentina acerca de Mecánica (ya sea de la más simple), estos problemas se vuelven un dolor de cabeza debido a que se desconoce la causa, la importancia y más aún como solucionarlo.

Este proyecto busca resolver dicho problema, brindando toda esta información de manera sumamente sencilla, a través del reconocimiento de los ruidos que estos problemas producen e informando con interpretación imágenes toda la información que da el tablero del auto, esta información poco conocida se puede tener a la mano en unos pocos segundos.

Todo esto a través de una aplicación mobile, inteligencia artificial y diferentes lenguajes de programación que permitirán crear los modelos los cuales van a predecir todos los sonidos e imágenes que le sean ingresados para así obtener respuesta a los problemas.

La aplicación esta creada con los lenguajes Java para el Backend y Kotlin para el Frontend, lo que es la aplicación como tal, realizada en Android Studio.

# Capítulo 1. Introducción

Uno de los problemas más habituales para las personas que conducimos, es la detección

de los problemas que nuestros autos suelen tener a lo largo de su vida útil.

Todos conocemos ese típico “Ruido” que empiezan a aparecer con el uso cotidiano, los

cuales pueden indicar desde cosas leves, a otras realmente urgentes que

deberían ser atendidos de inmediato.

Al igual que aquellas luces que aparecen en nuestros tableros los cuales no necesariamente conocemos cada significado de estas, lo que intentan avisar y la manera de solucionarlo.

Mi objetivo es poder llevarle a las personas que, como la gran mayoría no tienen un

entendimiento profundo de los automóviles, una forma fácil de saber que le está pasando

su auto, porque existe este ruido, esa luz del tablero, el nivel de emergencia y sus posibles soluciones.

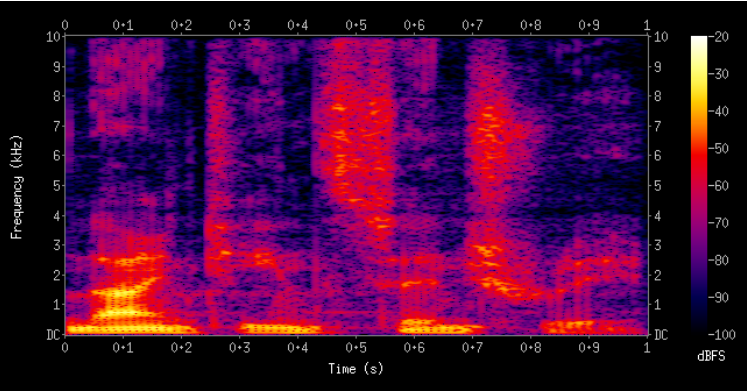
Todo esto a través del micrófono para la detección de sonidos y con una simple imagen sacada desde el teléfono que, ayudado con la tecnología de redes neuronales artificiales dará respuesta a todas estas cosas y así conseguir el objetivo planteado en el proyecto.

Se realizó modelos en base a los sonidos e imágenes para que nuestra aplicación móvil pueda detectar y clasificar cada una de estas y así darnos una respuesta acertada sobre los problemas que nuestro auto posee.

Para la detección de sonidos se utiliza principio por el cual trabaja la aplicación “Shazam” para detectar canciones con solo oír un fragmento de estas, de esta misma aplicación recolectar toda la información necesaria para el reconocimiento tan preciso de diferentes tipos de sonidos.

Realizando Espectrogramas y clasificando cada uno dependiendo de las frecuencias del sonido.

Figura 1. Espectrograma



Principalmente el detector de sonidos comienza reconociendo aquellos sonidos que produzcan un sonido particular, entre los cuales son:

- Fuga en el caño de escape

- Problemas con la batería

- Problema con la correa

Esto debido a que algunos sonidos producidos pueden ser muy similares y representar muchos problemas diferentes, o bien un único problema que produzca una inmensa variedad de sonidos diferentes entre ellos.

En cuanto a las luces del tablero, se sabe que en el manual se brinda esta información, pero en esta era de digitalización las personas buscan obtener respuesta lo más rápido posible, y el uso del manual en jóvenes es cada vez menor.

Algunas de las barreras que existen para esta aplicación:

**Barrera Cultural**: Se desconoce si las empresas automovilísticas quieren adaptar un software ajeno que apunte específicamente a los problemas del automotor, debido a que muchas agencias poseen personal especializado y herramientas especializadas para detectar estos problemas.

Al existir la posibilidad de que las personas conozcan lo que le está sucediendo a su automóvil sin llevarlo a los lugares recomendados por el manual lo más probable es que estos intentarán solucionarlo por sus propios medios (Ya sea por comodidad o por temas económicos), perdiendo así garantía o la seguridad de arreglar los autos en los talleres oficiales.

**Barrera de Espacio**: Véase como espacio al almacenamiento necesario para todos los sonidos que tienen los automóviles, sin contar que cada modelo o marca tiene su propio sonido en cuanto a problemas se refiere.

**Barrera de Tiempo**: Esta barrera va de la mano con la anterior, la inmensidad de autos que existen hoy en día da como resultado un tiempo en la carga, de todos los sonidos que estos pueden hacer, muy grande, además del espacio que las bases de datos requieren.

**Barrera Técnica**: La tecnología utilizada hoy en día para reconocer sonidos es escasa y más en el área del automovilismo donde algunos sonidos de un mismo problema pueden escucharse de maneras muy distintas, por lo cual una barrera importante es la del reconocimiento como tal de ciertos sonidos los cuales no son demasiados característicos, como un sonido metálico o de chapa.

Además, muchos problemas que poseen los autos no tienen ningún ruido hasta que el auto se descompone por completo, como podría ser la pérdida de agua refrigerante y la subida de temperatura del automóvil. Estos problemas pasan desapercibidos por la aplicación debido a que no producen ruidos significativos, hasta que el auto se rompe.

El proyecto busca solucionar el problema más grande de las personas con los automóviles, el que es el entendimiento mecánico del auto y más específicamente el entendimiento de problemas de estos. En una era de digitalización la adaptación del Software va a ser inmediata y evitará miles de complicaciones a largo plazo.

El estudio de los sonidos y la recolección de datos se hace de una manera única, debido a la complejidad de estos sonidos, lo cual agrega una relevancia importante al proyecto. En el enfoque vemos que el sistema cuenta con dos fases importantes y la conexión entre estas.

Si bien se utiliza como guía algunos softwares existentes, lo que se plantea en el proyecto lleva toda esta tecnología un paso más adelante, no solo comparando un único sonido, sino detectando miles de variantes de este mismo, ya que un mismo problema puede tener muchas variaciones a la hora de ser detectados.

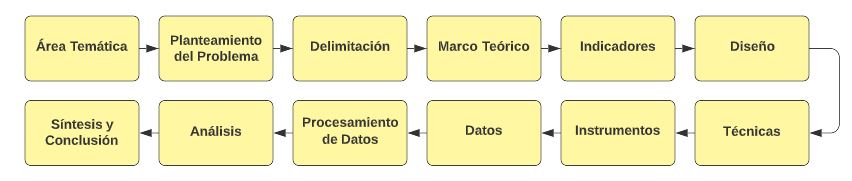
Hoy en día los automóviles de alta gama vienen con un sistema incorporado de detección de problemas, pero en Argentina, aproximadamente el 22% de la población puede acceder a estos vehículos. El otro 78% tiene autos de gama media o baja, por lo cual el proyecto adquiere una relevancia enorme para todo este sector de la sociedad.

# Capítulo 2. Metodología y procedimientos

Durante el proyecto se adopta una metodología de investigación creada por Carlos A. Sabino con pasos a seguir para una correcta documentación de la misma, la cual facilita el entendimiento del problema y ayuda a tener una conclusión certera de todo lo planteado a continuación.

La metodología cuenta con doce pasos los cuales están representados en la figura 2, y además se acompaña con un diagrama de Gantt con el objetivo de marcar tiempos a cumplir y llevar un detalle con cada uno de los pasos de la metodología.

Figura 2 Pasos de Metodología



Área temática: El área temática son los problemas de los vehículos los cuales producen un sonido particular, esto acompañado con la tecnología de detección de imágenes y audio para poder dar una explicación a estos ruidos que producen los problemas de los autos.

Planteamiento del problema: El problema a tratar está enfocado en aquellas personas, más bien jóvenes, que poseen autos usados o antiguos los cuales luego de un tiempo empiezan a tener fallas notorias, fallas las cuales producen un sonido particular.

Delimitación de la investigación: El alcance del proyecto se limita en una primera instancia a Argentina, debido a que los estudios realizados son sobre los vehículos en este país. Además, gran parte de la población argentina posee autos de gama media o baja, donde suceden estos sonidos con mucha más frecuencia.

Con la escalabilidad del proyecto en un futuro se podría pensar en expandir hacia Latinoamérica.

Marco teórico: Desarrollo de la información que respalda el proyecto tomando la bibliografía referenciada. Investigando se encuentran muchos estudios al respecto, no estrictamente en el área del automovilismo, o no al menos de la forma planteada en el proyecto, pero sí con fines industriales y/o mecánicos.

Algunos documentos importantes encontrados podrían ser “Sistema inteligente para el tratamiento de ruidos”, investigación creada para la Universidad de Palermo o también “Diseño de sistemas de control pasivo de ruido en recintos industriales” de la asociación Argentina de Mecánica Computacional, entre muchos otros documentos.

Diseño concreto: El diseño concreto es llevar a cabo un dispositivo que registre sonidos e imágenes para dar respuesta a problemas provenientes de vehículos. Llevando una nueva tecnología con base en la detección de audios e imágenes.

Instrumentos: Para medir el nivel de éxito de la solución se hicieron exhaustivas pruebas hasta que el sistema dio resultados exitosos, luego se compartió con diferentes usuarios para que dieran su perspectiva sobre la solución y midan el nivel de éxito. Siendo exitoso cuando los problemas detectados sean efectivamente los correctos.

Datos: Se realiza el preprocesamiento de datos siguiendo los pasos siguientes pasos, seleccionados del Knowledge Discovery in DataBases (K.D.D), ya que este concepto comprende la limpieza, interpretación y descubrimiento de patrones en la información.   
- *Limpieza*: Consiste en remover toda la información no representativa del dataset seleccionado.

- *Selección*: Consiste en seleccionar las columnas del dataset que se apliquen al problema que se está tratando.

- *Transformación*: Consiste en realizar la transformación de los datos de modo tal, que puedan ser utilizados por los algoritmos y también para sacar mejor provecho de los modelos.

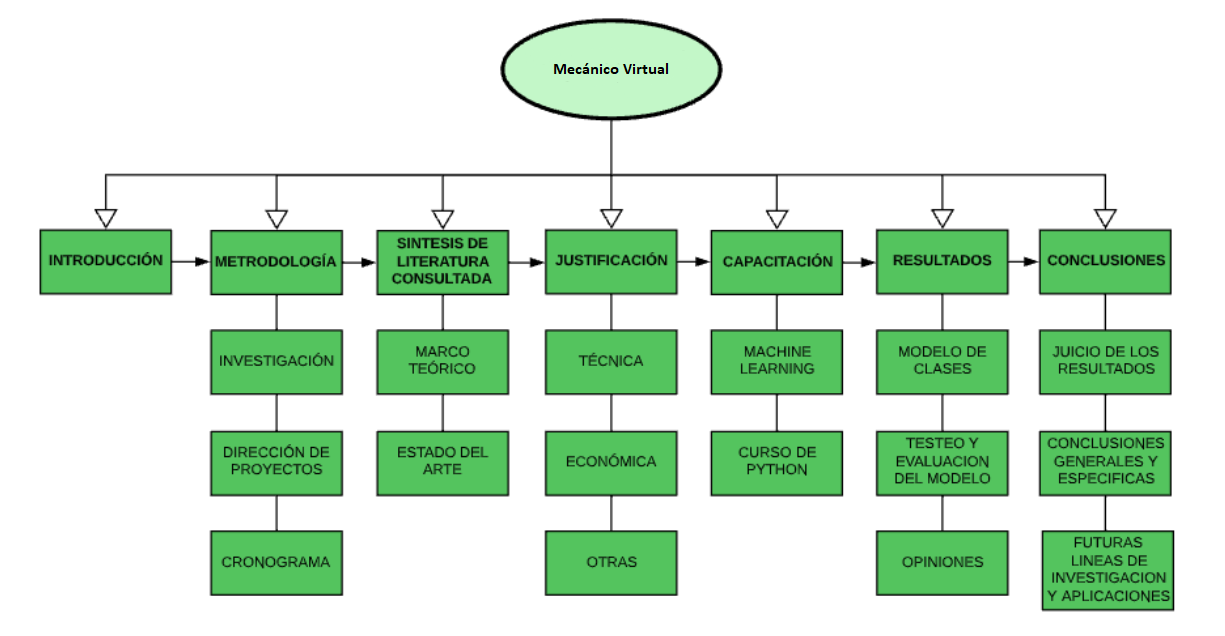
- *Minería de datos*: Consiste en seleccionar el algoritmo que mejor se adapte a nuestro problema y ejecutarlo para ver los resultados.

Análisis de datos obtenidos: Para el análisis se utilizó el paso de “Evaluación” del K.D.D que consiste en evaluar si la información y eficiencia obtenidas resultan relevantes para el problema planteado.

Conclusiones: Luego del análisis, se especifica los resultados obtenidos y se realizó una conclusión global en base a dichos resultados.

**La estructura de la metodología de dirección de proyectos planteada es la siguiente:**

Figura 3. Metodología Planteada

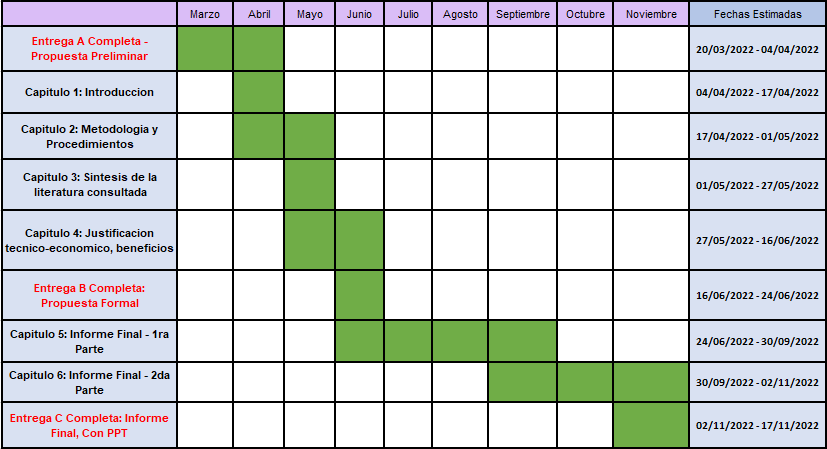


El gráfico anterior debe ser leído de izquierda a derecha y de arriba abajo. Cabe remarcar, que el proyecto es desarrollado siguiendo la presente estructura en forma iterativa e incremental, sin necesidad de cumplir con un orden específico. El orden de los elementos del gráfico no debe ser tomado en cuenta para su entendimiento, finalizando el proyecto cuando todas sus etapas estén completas.

**Cronograma**

El cronograma se presenta en forma de cuadro, explicando el paso a paso de la investigación que se irá realizando, acompañado con diagramas de Gantt para un entendimiento más gráfico del tiempo y el proceso realizado.

Comenzando la primera semana de abril de 2022 y finalizando a mediados de noviembre de 2022, se cuenta aproximadamente con 8 meses para la finalización del proyecto.

**Cronograma general**:

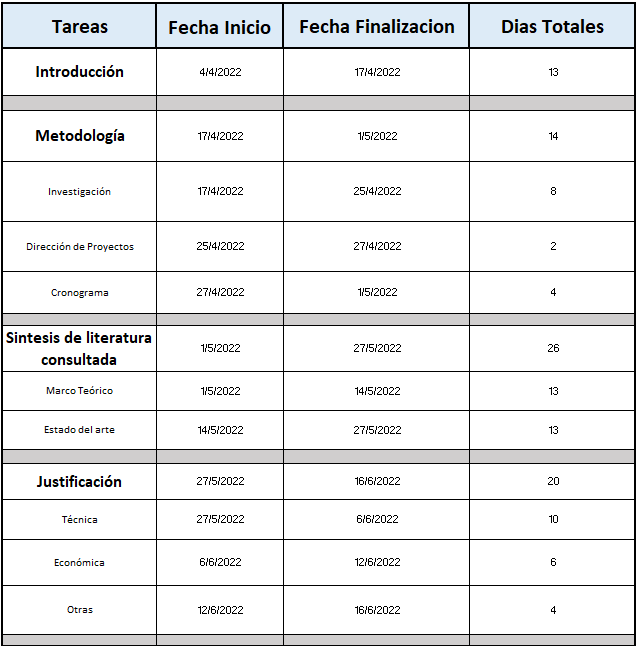
**Cronograma específico:**

Dentro del cronograma específico se realizó otro cronograma donde se explica cada una de las tareas mostradas en la estructura de la metodología de dirección del proyecto.

Comenzando con la introducción y finalizando con las conclusiones del mismo.

Está dividido en dos partes, la primera haciendo hincapié en entregar una propuesta formal del proyecto, y la segunda, en terminar con todas las tareas del informe final del mismo, concluyendo en la presentación final.

Primera parte:



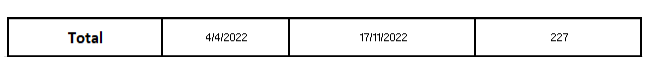
En esta primera sección del proyecto se realiza un enfoque de entrega completa del informe formal en la fecha 16 de junio, con un total de 60 días pasados desde el inicio de la propuesta preliminar.

Segunda parte:



En la segunda sección del proyecto tal como se especificó anteriormente, comienza con 4 meses de capacitación para llevar a cabo el prototipo del proyecto, y luego pasa por los resultados del mismo y así obtener de esta manera conclusiones sobre el proyecto.

Una vez finalizado se hace una presentación final del proyecto.

Total:

# Capítulo 3. Síntesis de literatura consultada

**Estado del Arte**

En esta sección del capítulo se hace un resumen de algunas investigaciones encontradas acerca de los temas a tratar en el proyecto.

Revisar todo el conjunto de saberes o el desarrollo que se ha conseguido en el área en la cual se investiga es clave para lograr el objetivo propuesto para toda esta investigación.

El estado del arte se divide en dos importantes pilares, la detección de audio y una aplicación la cual aplica esta detección de audio.

**Clasificación y Detección de Sonidos**:

Primer Investigación:

* Título: “*Detección de voz y música en un corpus a gran escala de eventos de audio*”.
* Objetivos:   
    
  1. Estudio, comprensión y descripción de la base de datos AudioSet de Google.   
    
  2. Consulta del estado del arte del área, experimentos relacionados y resultados previos.   
    
  3. Diseño de modelos de clasificación basados en diferentes arquitecturas de redes neuronales.   
    
  4. Diseño y ejecución de experimentos para la evaluación del rendimiento de los modelos de clasificación propuestos.

5. Interpretación y discusión de los resultados obtenidos.

6. Generación de código orientado al manejo de la base de datos AudioSet y su uso en futuras investigaciones.

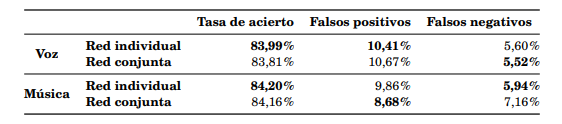
* Muestra: 77.396 segmentos de audio de 10 segundos (216 horas) pertenecientes a vídeos de YouTube, extraídos de la reciente base de datos AudioSet de Google.

Evaluados con arquitecturas de redes neuronales, basadas en DNN, CNN y LSTM.

* Palabras Claves: Aprendizaje automático, deep learning, redes neuronales, redes convolucionales, LSTM, audio, voz, música, reconocimiento de eventos de audio.
* Resultados:

Las redes entrenadas para las tareas de detección de voz, detección de música y detección conjunta de voz y de música alcanzan resultados muy similares en cuanto a tasas de acierto, que se recogen en la tabla 4.1. Las tasas de falsos positivos y falsos negativos se expresan como porcentaje sobre el total de segmentos clasificados. En ambos casos (música y voz), la precisión es levemente superior con las redes individuales, pero no lo suficiente como para considerarlo una diferencia significativa.

Figura 4.1 Tabla de Efectividad



* Conclusión

Han sido evaluadas diferentes arquitecturas de redes neuronales, basadas en DNN, CNN y LSTM. Los resultados indican un rendimiento muy destacable de las arquitecturas basadas en redes convolucionales, que logran rendimientos muy similares en voz y en música, con tasas de acierto en torno al 84%. También se ha destacado el rendimiento de la red conjunta, que alcanza resultados semejantes a los de las dos redes individuales combinadas, pero contando aproximadamente el mismo número de parámetros entrañables que sólo una de ellas.

Segunda Investigación:

* Título: “*Análisis y Clasificación de Sonidos de Motores de Carros, para Diagnosticar Fallas, con Ayuda de Redes Neuronales*”.
* Objetivo:   
    
  El objetivo de la investigación es reconocer problemas en el motor a través de su sonido, utilizando las vibraciones que este realiza cuando existe algún tipo de falla. Con ayuda también de redes neuronales.

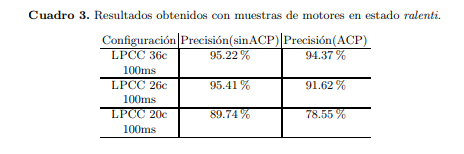
* Muestra:

Las muestras de los sonidos fueron sacadas de tres talleres se especializan en motores de autos. Cada muestra grabada fue obtenida con ayuda de una grabadora digital ZicPlay y los archivos de salida fueron guardados en formato WAV, con las siguientes características: 8 khz, 16 bits y canal monoaural. Cada uno de estos archivos tiene un periodo de tiempo de 3 a 5 segundos, y fueron grabados en total 26 diferentes máquinas. Estas muestras fueron clasificadas en 4 categorías: motores en buenas condiciones (clase 1), problemas relacionados con el cigüeñal (clase 2), con pistones (clase 3) y con problemas en válvulas (clase 4)

* Resultados:

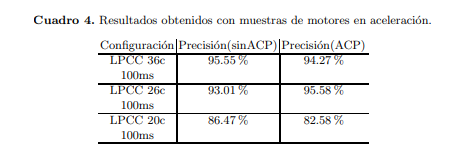
En el Cuadro 3, se pueden ver los resultados obtenidos usando muestras de máquinas en estado ralentí, el Cuadro 4, muestra los resultados obtenidos de muestras grabadas de máquinas en aceleración. Cada resultado, es el obtenido de la media aritmética de 10 experimentos previos cada uno. En ambos cuadros, la primera columna muestra el número de características extraídas por cada segmento de tiempo (ms).

Figura 5.1 Motores en estado ralentí



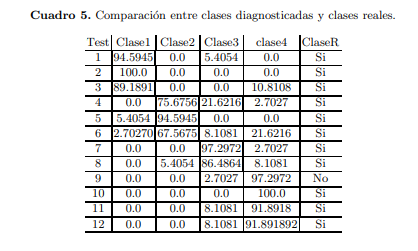
Los resultados en ambas tablas muestran que la mejor precisión fue dada cuando usamos 36 y 26 LPCC por cada 100 ms, cuando usamos muestras de motores con aceleración; podemos decir que con 36 coeficientes por segmento de tiempo, el lapso de entrenamiento fue mayor y no había mucha diferencia en los resultados de reconocimiento, mostrando con esto, que 26 LPCC dan mejores resultados globales.

Figura 5.2 Motores en aceleración



Tomando esto en cuenta, se seleccionaron al azar doce muestras de sonido en formato WAV, provenientes de ruido no usado en la etapa de entrenamiento, y separado para este fin; tomando una RNA previamente entrenada, se alimentaron estas muestras para demostrar la precisión que logra alcanzar dicha red. Los resultados obtenidos se muestran en el Cuadro 5.

Figura 5.3 Comparación



* Conclusiones

Los resultados muestran que en un motor en malas condiciones existen diferencias acústicas, comparándolo con un motor afinado o en buenas condiciones. Al menos estas diferencias entre sonidos pueden ser reconocidas por una Red Neuronal con Retraso en el Tiempo, claro está, dependiendo del tipo de falla que esté afectando la eficiencia del motor. Esta información puede ayudar al especialista del motor a dar diagnósticos rápidos, precios estimados por la reparación y permitirle trabajar directamente con el problema del motor. También podemos concluir que es viable trabajar en este problema e implementar un sistema barato que pueda ser usado por fabricantes de motores o dueños de talleres.

**Aplicaciones Similares**: ***Shazam***

Título: “*Shazam: Cómo funciona el algoritmo de reconocimiento de canciones de la popular aplicación”*

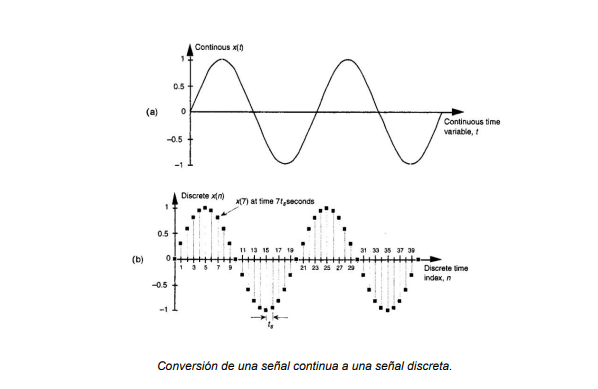
Introducción: Shazam es una aplicación móvil que permite la identificación de canciones. La aplicación hace uso del micrófono que llevan incorporados la mayoría de smartphones para poder grabar una muestra de música de 10 segundos que se esté reproduciendo en el ambiente. A partir de esta breve muestra de audio, la aplicación es capaz de reconocer exactamente qué canción está sonando, brindándole al usuario información útil y precisa.

Funcionamiento: La aplicación se compone de un amplio catálogo de “huellas digitales” de audio que son almacenadas y catalogadas en una base de datos. Cuando un usuario graba con su teléfono una canción durante algunos segundos, esa grabación es procesada inmediatamente por la aplicación, creando una huella acústica (más adelante profundizaremos sobre este concepto) y enviándola a los servidores de Shazam a través de Internet. Una vez allí, se ejecuta una búsqueda de coincidencias en una base de datos que contiene millones de canciones (entre 8 y 11 millones, según algunas fuentes). Cuando existe una coincidencia, el usuario recibe en pocos segundos una respuesta con información de la pista original como el título de la canción, artista, álbum, e información adicional.

¿Cómo se logra este funcionamiento?

El micrófono del teléfono convierte la señal de presión de aire continúa producida por la misma canción, en una señal de voltaje analógica. De aquí se toma un muestreo que luego será analizado.

Figura 6.1 Conversión de una señal continua a discreta



Si utilizamos esa frecuencia de muestreo, podremos reproducir la señal posteriormente a partir de las muestras tomadas.

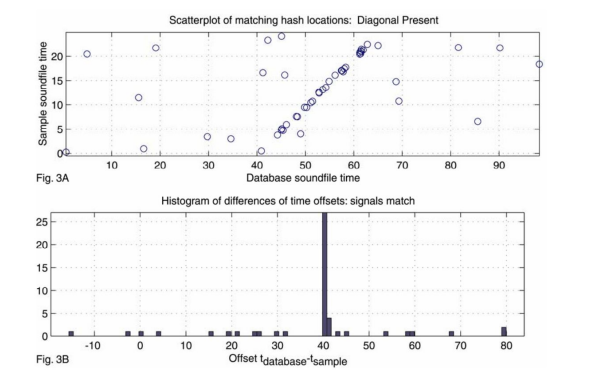
Gracias a estas muestras se consigue a través de un largo procedimiento las denominadas “**Huellas Acústicas**” las cuales son un “resumen” digital del archivo, generado a partir de la señal de audio original. Gracias a este procedimiento, es posible identificar o localizar unívocamente un archivo de audio en una base de datos, sin necesidad de contar con información adicional.

Luego de analizar estas huellas acústicas, se pasa a la **detección de coincidencias**:

Ahora que hemos creado la huella acústica de nuestro archivo de audio, la información codificada en un conjunto de hashes con cada uno de los pares de puntos de anclaje de la grabación del usuario se envía a la base de datos de Shazam. Esta búsqueda devolverá las huellas acústicas de todas las canciones que contengan coincidencias de hash, es decir, que tengan los mismos pares de puntos de anclaje.

Si visualizamos este proceso en un diagrama de dispersión donde el eje y representa el momento en que se produce el hash en la grabación y el eje x representa el tiempo en que se produce el hash en el archivo de audio de la base de datos, podemos ver cómo se formarán los puntos coincidentes en la línea diagonal:

Figura 6.2 Diagrama de dispersión



El desplazamiento en el tiempo se calcula restando el tiempo de ocurrencia del par de puntos de anclaje en la señal de entrada del tiempo de hash correspondiente en el archivo de audio de la base de datos. Si una cantidad significativa de hashes coincidentes tienen el mismo coeficiente de desplazamiento, se determina que esa canción es una coincidencia.

Este método de búsqueda de audio es lo suficientemente preciso para encontrar coincidencias incluso cuando la señal de entrada contiene ruido, como personas hablando, ruido de calle o incluso otras canciones sonando en simultáneo. Debido a que el número de hashes de puntos de anclaje creados por una huella acústica es mucho mayor que la cantidad de coincidencias de puntos de anclaje requeridas para devolver un resultado de búsqueda positivo, los hashes de puntos de anclaje que están enmascarados por el ruido externo no son suficientes para evitar que Shazam siga buscando coincidencias recursivamente en otros fragmentos de la señal.

**Clasificación de Imágenes**:

* Título: “*EJECUCIÓN DE REDES NEURONALES EN MÓVILES ANDROID CON ACELERACIÓN HARDWARE MEDIANTE KERAS Y TENSORFLOW LITE*”.
* Objetivos: El objetivo de este TFG consistirá principalmente en el estudio de distintas redes neuronales ya entrenadas aplicadas al problema de reconocimiento o clasificación de imágenes. Las redes neuronales empleadas en el estudio serán ejecutadas en dispositivos móviles, en concreto dispositivos Android, y se evaluará la influencia de distintas herramientas de aceleración hardware y software para intentar optimizar las redes ya existentes. Entre las redes neuronales a evaluar se encuentran algunas especialmente diseñadas para dispositivos móviles como MobileNet o EfficientNet y otros modelos consolidados como es el caso de VGG16 y ResNet50. El desarrollo del trabajo puede ser dividido en dos secciones bien diferenciadas. La primera consistirá en la puesta en marcha del entorno de desarrollo, desde la configuración de una aplicación que permite identificar los objetos captados por la cámara del dispositivo móvil en tiempo real aplicando una red neuronal a elección, junto con una serie de opciones de aceleración, hasta la puesta a punto de ciertas herramientas de medición para lograr una cuantificación de los tiempos de ejecución y de la precisión de clasificación de un mismo conjunto de imágenes usando distintas redes neuronales y configuraciones para su posterior comparación.
* Resultados: Haciendo un primer análisis rápido, se puede observar como el modelo de EfficientNet tiene el mejor Top-1 y Top-5 accuracy de todos los modelos estudiados antes de transformarlos a TensorFlow Lite. Pero realmente no existe una diferencia excesivamente notable entre unos modelos y otros, ya que las tasas de acierto varían como máximo entre unos modelos y otros no más de un 5%.

Tras realizar la conversión de los modelos a TensorFlow Lite, se puede medir su accuracy gracias al script del Anexo D Medición de accuracy de un modelo TensorFlow Lite y al contenido explicado en el punto 3.3 de Medición de Accuracy.

En el caso de este proyecto, por temas de recursos y tiempo el accuracy de los modelos en TensorFlow Lite se ha hecho únicamente sobre las 5000 primeras imágenes de ILSVRC, en vez de las 50000 imágenes de validación con las que cuentan, por lo que la tasa obtenida puede que no sea perfecta, pero permitirá tener una aproximación bastante acertada.

Una vez se tienen los valores de accuracy para los modelos de TensorFlow Lite, tal y como se observa en la Figura 4-3 en todas las redes neuronales a excepción de MobileNet desciende su Top-1 entre un 5% y un 6%, mientras que el Top-5 desciende algo menos entorno al 3% o 4%. El caso de MobileNet es muy curioso, ya que el Top-1 en vez de disminuir aumenta mínimamente, lo cual puede haber sucedido perfectamente por lo explicado previamente acerca de que las mediciones de las tasas de acierto no se han hecho sobre el conjunto total de validación, sino sobre el primer 10% del mismo. El Top-5 de MobileNet también decrece muy poco, de hecho, es la tasa que menos se reduce entre todos los modelos, la cual ronda en torno a un 0,5%

* Conclusión: Este trabajo partió del objetivo inicial de la puesta en marcha y estudio de redes neuronales en dispositivos móviles gracias a las herramientas de TensorFlow Lite. Para la puesta en marcha se ha necesitado entender el funcionamiento del conversor de TensorFlow Lite y sus diferentes optimizadores. Para el estudio de las redes, se han adaptado una serie de scripts en C++ y Python que han permitido monitorizar el impacto del uso de diferentes configuraciones hardware y software a través de los delegados establecidos en TensorFlow Lite, entre los que se han podido probar se encuentran GPU y NNAPI, además del uso de múltiples hilos. Por último, para demostrar la verdadera potencia de este tipo de redes, se han incorporado a una aplicación de clasificación de objetos en tiempo real a partir del video captado por la cámara del teléfono. Para conseguir esto, se han necesitado introducir los pre procesamientos adecuados para cada una de las redes, incluyendo la generación del código necesario para poder transformar las imágenes de RGB a BGR, junto con un proceso de zero-centered que consiste en centrar los valores de todos los píxeles respecto al eje de coordenadas. En cuanto a los resultados obtenidos se podría considerar que cualquier modelo capaz de conseguir un tiempo de inferencia inferior a los 40 ms se le reconoce de tiempo real, puesto que con esos tiempos una cámara podría captar y analizar un mínimo de 25 imágenes por segundo, imágenes suficientes para hacer una evaluación del video captado por la cámara a tiempo real.

**Marco Teórico**

En el Marco Teórico se realiza una recopilación de antecedentes y consideraciones teóricas en las que se sustenta este proyecto de investigación.

A continuación, se definen algunos conceptos importantes para el entendimiento del mismo.

# Problemas en Automóviles

Si bien saber mucho sobre automóviles o mecánica no es primordial para los fines del proyecto, es importante hacer una introducción sobre estos, principalmente sobre las problemáticas que suelen sucederles.

Tener al menos un leve conocimiento sobre el tema es importante para saber en qué se enfoca el estudio realizado.

Un automóvil también denominado coche, carro, auto, entre otros, es un vehículo motorizado con ruedas utilizado para el transporte. La mayoría de las definiciones de automóvil, hacen referencia a los vehículos que circulan principalmente por las carreteras, con capacidad de uno a ocho personas, cuentan con cuatro neumáticos y transportan principalmente a personas, antes que mercancías.

¿Por qué un automóvil tiene fallas?

Curiosamente, el uso frecuente de un automóvil no es la principal causa de los daños. Por el contrario, en la mayoría de los casos, las fallas mecánicas en autos ocurren por falta de mantenimiento o por ignorar las señales que indican problemas. Conocer la mecánica del automóvil es la mejor forma para detectar las posibles alarmas y no descuidar los aspectos cruciales del mantenimiento.

Las malas prácticas por parte del conductor son otro motivo de fallas, por ejemplo, no controlar periódicamente la presión de los neumáticos genera desgastes irregulares y reventones. Abusar de los frenos en bajadas prolongadas tiene como consecuencia un desgaste mayor de los discos, pastillas y deteriora el líquido de frenos.

Tener el auto parado por mucho tiempo tampoco es recomendable porque causa que los neumáticos se deformen, los frenos queden inmovilizados debido al óxido, o se produzcan fugas de aceite tanto en el motor como en la caja de cambios.

Algunos de los problemas más frecuentes son:

* Batería
* Bujías
* Frenos
* Filtraciones
* Neumáticos

Y esto solo es el comienzo de una larga lista…

Como se mencionó en el capítulo 1 de la tesis, durante la introducción, todos estos problemas suelen producir cambios en el automóvil, muchos de ellos suelen producir sonidos particulares y la detección sencilla de estos problemas va a producir a futuro un ahorro grande de dinero, tiempo y problemas mayores.

# Comportamiento del sonido y espectrogramas

Denominamos sonido a cualquier fenómeno que involucre la propagación de ondas mecánicas a través de un medio que esté generando el movimiento vibratorio de un cuerpo.

La física del sonido es estudiada por la acústica, que trata tanto de la propagación de las ondas sonoras en los diferentes tipos de medios continuos como la interacción de estas ondas sonoras con los cuerpos físicos.

Estos a su vez son ondas muy complicadas por tener muchos componentes de diferente índole (armónicos, parciales y ruidos) que evolucionan muy rápidamente en el tiempo por lo que su análisis y estudio resulta bastante moroso.

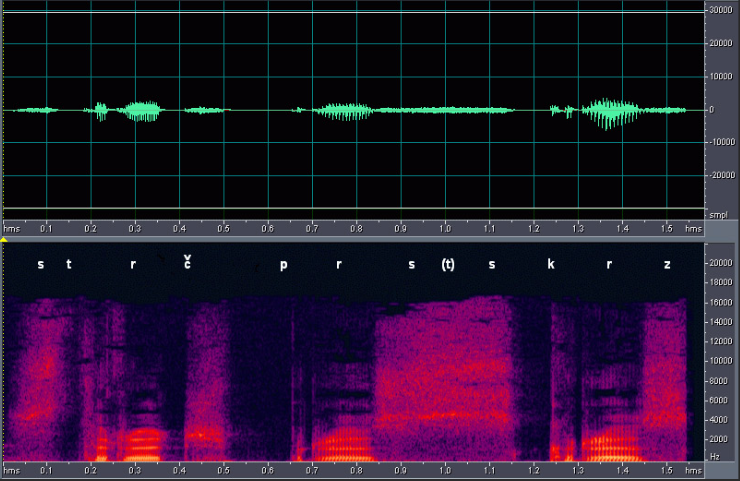
Cada sonido posee una propagación, resonancia, magnitud y velocidad única las cuales unidas generan un espectro de frecuencia, completamente diferentes uno al otro, y en consecuencia de esto nos resulta sencillo diferenciar un sonido de otro, o reconocer de dónde provienen estos.

¿Pero cómo puede un sistema reconocer la diferencia entre un sonido y otro?

Y aquí es donde entran los **Espectrogramas**.

Un Espectrograma es resultado de calcular el espectro de una señal por ventanas de tiempo de la misma. Resulta una gráfica tridimensional que representa la energía del contenido frecuencial de la señal según va variando ésta a lo largo del tiempo.

Figura 7. Ejemplo de Espectrograma



Estos se crean a partir del análisis por tramas, que es el primer paso a seguir, debido a la naturaleza variante de una señal de audio, se aplica una ventana para escoger un número limitado de muestras a procesar.

La elección de la ventana es un compromiso entre el tamaño de la trama, el procesado y el análisis de la señal.

Existen muchos tipos de ventanas, con más o menos resolución y esta elección va a depender del uso que se le quiera dar al espectrograma.

Algunas aplicaciones de estos espectrogramas son:

* Los espectrogramas son útiles para ayudar en la superación de defectos del habla y en la formación del habla para la parte de la población que es profundamente sorda.
* Los estudios de la fonética y síntesis de voz se han facilitado a través del uso de espectrograma.
* Al invertir el proceso de elaboración de un espectrograma, es posible crear una señal donde el espectrograma es una imagen arbitraria. Esta técnica se puede utilizar para ocultar una imagen en un pedazo de audio y ha sido empleado por varios artistas de la música electrónica.
* Algunos tipos de música moderna se crean utilizando un espectrograma como medio intermedio, para cambiar la intensidad de diferentes frecuencias a través del tiempo, o incluso para crear otros nuevos, mediante la elaboración y después la transformación inversa.
* El Espectrograma se puede utilizar para analizar los resultados al pasar una señal de prueba a través de un procesador de señales, como un filtro para comprobar su rendimiento.
* Análisis frecuencial de cualquier señal audiovisual
* Visualización de señales eléctricas y de comunicaciones.
* Análisis musical.
* Aplicaciones acústicas.
* Aplicaciones sismológicas.
* Análisis de señales biomédicas.
* Extracción de características.

# Detección de sonidos

La herramienta principal que se utiliza para la detección de sonidos son los micrófonos, este es un elemento que transforma una señal acústica de un audio en una señal eléctrica.

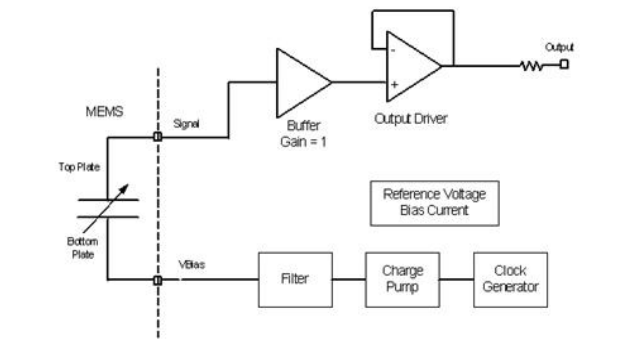
Existen muchos tipos diferentes de micrófonos que utilizan diferentes técnicas para realizar la transformación de la señal acústica en la señal eléctrica.

Todos los micrófonos se basan en la misma propiedad de transducir una variación de presión a una variación de tensión y varían en el concepto físico del procedimiento.

Los micrófonos se dividen en una gran clasificación, pudiendo ser dinámicos, de carbono, piezoeléctrico, entre muchos otros, pero todos ellos a su vez se agrupan en dos tipos principales, los **analógicos** y los **digitales**.

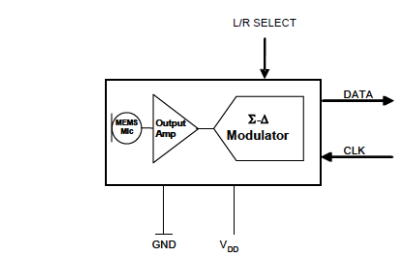
**Los micrófonos analógicos** son del tipo silicio, también llamados “MEMS”. Suelen estar construidos en encapsulados para montaje superficial para mayor comodidad a la hora de integrarlos en un circuito impreso para alguna aplicación. Su reducido tamaño permite poder colocarlos en el mejor lugar acústico para la aplicación. Las dimensiones mínimas rondan los 3mm de largo, 2 mm de ancho y 0.88 mm de altura.

Figura 8. Estructura de micrófono analógico



Y luego están los **micrófonos digitales**, que, a diferencia de los micrófonos analógicos, tienen la característica de llevar un conversor analógico-digital en el propio dispositivo de transducción para tener en la salida una señal digitalizada y en formato PWM. Esta mejora de prestación en la estructura de los micrófonos de silicio permite reducir el ruido del sistema considerablemente ya que no hay ni cables ni conexiones entre el codificador y el micrófono y permite eliminar directamente dicho componente lo cual nos deja la oportunidad de crear dispositivos más pequeños y con menos consumo.

Figura 9. Estructura de micrófono digital



Por último, es necesario nombrar a los **Códecs Audio** que es una abreviatura de codificador-decodificador.

El códec es un dispositivo electrónico que une las funciones de codificador y decodificador. Las funciones que realizan son muy simples. Convierten una señal analógica de audio o vídeo en una señal de audio o vídeo digital. Transforman la señal en primer lugar y luego la envían a través de un canal de comunicaciones y en un formato específico para que sea procesado en un dispositivo electrónico digital como por ejemplo un DSP.

Lo que importa de estos “Códecs” son sus aplicaciones, donde las más comunes suelen ser en cámaras de video, cámaras fotográficas, tabletas digitales, teléfonos, ordenadores, y todos aquellos dispositivos que requieran un detector de sonido. En todos estos sistemas existe algún micrófono o sensor de fuente analógica que ha de ser transformado para su almacenamiento, tratamiento o transmisión.

Un teléfono por ejemplo recoge una señal de voz hablada, que ha de ser transformada a una señal digital para poder ser procesada y transmitida correctamente.

# Almacenamiento de Sonidos (AudioSet)

Uno de los problemas más grandes a la hora de trabajar con audio/sonido siempre fue la inmensa cantidad de espacio que se requiere para almacenar cada uno de estos dentro de una base de datos. Sabiendo que un audio de 60 segundos pesa alrededor de 138 KB (0,138MB) el solo hecho de pensar en crear aplicaciones donde se tenga que mantener un registro de estos hacía muy complicado llevar a cabo cualquier proyecto relacionado. Es por esto mismo que Google se vio impulsado a publicar lo que denominaron como “AudioSet”, basado en los conocidos Dataset.

AudioSet es una base de datos publicada por Google en 2017 para impulsar el problema de reconocimiento de eventos acústicos. Además de tener una gran variedad de sonidos para reconocer, con más de 500 clases, el número de ejemplos es mucho mayor que en los otros casos, con más de dos millones de ejemplos.

Esta gran cantidad de ejemplos ha sido recopilada a partir de videos de YouTube, de los cuales se han extraído 10 segundos de audio, lo que hace que el total AudioSet cuente con más de 5000 horas de audio. Al provenir de videos que no han sido grabados para este propósito, los sonidos no están aislados y pueden solaparse. Las 527 clases están organizadas bajo una estructura jerárquica con profundidad máxima 6, siendo los nodos más altos categorías de sonidos y los más bajos sonidos específicos. Cada ejemplo puede estar etiquetado como una o más clases. Los datos están disponibles en dos formatos. El primero contiene los enlaces a los videos de forma que se puede acceder directamente al audio para tratarlo con libertad, mientras que el segundo contiene las salidas de una red de embedding con la que Google ha tratado previamente el audio, de modo que se puedan entrenar modelos sin necesidad de procesar señales de audio

# Minería de datos

La minería de datos es el proceso de analizar los datos desde diferentes perspectivas y resumiéndolos en información útil.

Este término está tomando cada vez más relevancia en la actualidad, y más entre las empresas, porque se sabe que el valor de los datos cada vez es más grande. Empresas tan grandes como Facebook o otras redes sociales, poseen un grado de relevancia mayor a cualquier otra empresa debido al valor de los datos de cada persona que estas poseen.

Sin embargo, raramente se puede sacar valor de los datos en bruto, es por eso que nace esta técnica que intenta sacar el mayor provecho de estos datos.

El verdadero valor, o los objetivos que busca la minería de datos son:

* Tener la habilidad para extraer información útil de la toma de decisiones o la exploración
* La comprensión del fenómeno gobernante en la fuente de datos.

En muchos dominios, el análisis de datos fue tradicionalmente un proceso manual. Uno o más analistas familiarizados con los datos, con la ayuda de técnicas estadísticas, proporcionaban resúmenes y generaban informes. En efecto, el analista hacía de procesador de preguntas sofisticado. Sin embargo, tal enfoque cambió como consecuencia del crecimiento del volumen de datos. Cada vez es más común encontrarse con bases de datos con un número de ejemplos del orden de 109 o superior y 103 dimensiones. Cuando la escala de manipulación de datos, exploración e inferencia va más allá de la capacidad humana, se necesita la ayuda de las tecnologías informáticas para automatizar el proceso.

Todo apunta a la necesidad de metodologías de análisis inteligente de datos, las cuales puedan descubrir conocimiento útil de los datos. El término KDD (Knowledge Discovery in Databases), acuñado en 1989 se refiere a todo el proceso de extracción de conocimiento a partir de una base de datos y marca un cambio de paradigma en el que lo importante es el conocimiento útil que seamos capaces de descubrir a partir de los datos.

El concepto de KDD se ha desarrollado, y continúa desarrollándose, desde la intersección de la investigación de áreas tales como bases de datos, aprendizaje automático, reconocimiento de patrones, estadística, teoría de la información, inteligencia artificial, razonamiento con incertidumbre, visualización de datos y computación de altas prestaciones.

El proceso global consiste en transformar información de bajo nivel en conocimiento de alto nivel. El proceso KDD es interactivo e iterativo conteniendo los siguientes pasos:

1. Comprender el dominio de aplicación: este paso incluye el conocimiento relevante previo y las metas de la aplicación.
2. Extraer la base de datos objetivo: recogida de los datos, evaluar la calidad de los datos y utilizar análisis exploratorio de los datos para familiarizarse con ellos.
3. Preparar los datos: incluye limpieza, transformación, integración y reducción de datos. Se intenta mejorar la calidad de los datos a la vez que disminuir el tiempo requerido por el algoritmo de aprendizaje aplicado posteriormente.
4. Minería de datos: como se ha señalado anteriormente, esta es la fase fundamental del proceso. Está constituido por una o más de las siguientes funciones, clasificación, regresión, clustering, resumen, recuperación de imágenes, extracción de reglas, etc.
5. Interpretación: explicar los patrones descubiertos, así como la posibilidad de visualizarlos.
6. Utilizar el conocimiento descubierto: hacer uso del modelo creado.

Algunas de las tareas importantes de la minería de datos incluyen la identificación de aplicaciones para las técnicas existentes, y desarrollar nuevas técnicas para dominios tradicionales o de nueva aplicación, como el comercio electrónico y la bioinformática. Existen numerosas áreas donde la minería de datos se puede aplicar, prácticamente en todas las actividades humanas que generen datos:

* Comercio y banca: segmentación de clientes, previsión de ventas, análisis de riesgo.
* Medicina y Farmacia: diagnóstico de enfermedades y la efectividad de los tratamientos.
* Seguridad y detección de fraude: reconocimiento facial, identificaciones biométricas, accesos a redes no permitidos, etc.

Entre muchas otras aplicaciones, pero la que nos centraremos en este proyecto es en la identificación y reconocimiento de algunos problemas puntuales en automóviles.

Que nuestro modelo sea capaz de identificar cuando un problema pertenece a una clase determinada, e intentar que el número de aciertos sea el mayor posible.

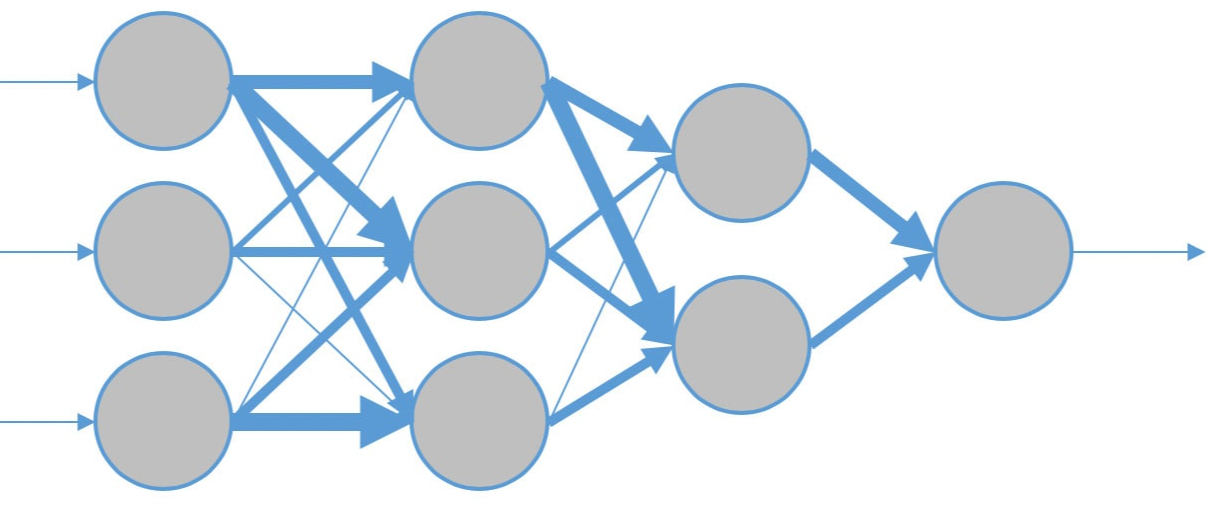
# Redes Neuronales Artificiales

Las redes neuronales artificiales se tratan de un modelo computacional evolucionado a partir de diversas aportaciones científicas que están registradas en la historia.

Consiste en un conjunto de unidades llamadas neuronas, conectadas entre sí para transmitir señales. La información de entrada atraviesa la red neuronal, donde se somete a diversas operaciones, y produce unos valores de salida.

Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una función limitadora o umbral, que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona. Esta función se conoce como función de activación.

Figura 10. Red Neuronal Artificial

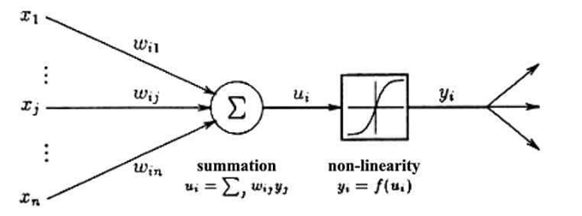


Cada uno de estos círculos representa una neurona artificial, la cual dentro realiza numerosas células para tomar una decisión y continuar con la siguiente neurona que esta misma asignó.

Las flechas representan la dirección, y el grosor de la misma su peso dentro de las decisiones.

Una neurona artificial se ilustra de la siguiente manera:

Figura 11. Neurona Artificial



Lo asombroso acá es que al conseguir que algoritmos, como las redes neuronales, puedan simular el comportamiento de inteligencia humana a cierto nivel, y sin importar los datos con los que se le entrene, trae una gran cantidad de beneficios y no solo económicos.

Las redes neuronales artificiales están presentes en nuestro día a día, y más en las redes sociales donde se utilizan algoritmos basados en los gustos de cada persona, y a través de estas, dichas redes te muestran publicaciones que tengan similitudes con los gustos de cada usuario.

Un ejemplo podría ser Spotify y YouTube, donde si bien son dos plataformas con fines distintos y funcionamiento distinto, tienen algo en común, los algoritmos que utilizan para recomendar y clasificar su contenido.

Con esto logran destacar el contenido que tenga una mayor probabilidad de aceptación para el usuario, lo cual seguramente terminará en un like, una suscripción al canal, agregar a una lista de reproducción o simplemente disfrutar del contenido recomendado.

Esta herramienta es de gran utilidad acompañada con la minería de datos realizada anteriormente, dando como resultado de la unión, un modelo predictivo que clasifica cada uno de los sonidos producido por los automóviles, esto sumado a la interacción de los usuarios que retroalimentan al modelo, el sistema inteligente irá aumentando cada vez más su capacidad de acierto.

# Clasificación de Imágenes

La clasificación de imagen hace referencia a la tarea de extraer clases de información de una imagen ráster multibanda. El ráster resultante de la clasificación de imagen se puede utilizar para crear mapas temáticos. Dependiendo de la interacción entre el analista y el equipo durante la clasificación, existen dos tipos de clasificación: supervisada y no supervisada.

## Clasificación supervisada

La clasificación supervisada utiliza firmas espectrales obtenidas de las muestras de capacitación para clasificar una imagen. Con la ayuda de la barra de herramientas Clasificación de imagen, puede crear fácilmente muestras de capacitación para representar las clases que desea extraer.

## Clasificación no supervisada

La clasificación sin supervisión busca clases espectrales (o clústeres) en una imagen multibanda sin la intervención del analista.

# Capítulo 4. Justificación técnico-económica

# Justificación técnica

Los recursos utilizados en el proyecto provienen de código abierto, debido a que ya existe una gran cantidad de ejemplos que realizan tareas similares para poder tener al menos una base de lo que se logra en esta aplicación. Más aún viendo que el proyecto se desglosa en, lectores de sonidos, trabajos con AudioSet, diseño de pantallas, entre otros ejemplos.

Existe una gran cantidad de proyectos junto a una explicación de cómo crear, diseñar e implementar estos proyectos completamente gratis en internet.

Muchos usuarios realizaron proyectos donde el fin era detectar y categorizar sonidos o imágenes, dentro de la lectura consultada se puede encontrar información sobre esto, y estos mismos usuarios dejaron toda una investigación acompañada del código fuente para poder replicar sus aplicaciones y entender cómo funcionan.

Los Dataset son una parte fundamental del proyecto, como se explicó anteriormente de una manera mucho más detallada, estos ayudarán a crear y entrenar un sistema inteligente capaz de reconocer ciertos sonidos y categorizarlos.

Existen muchos dataset de pago, pero en este caso la empresa Google lanzó recientemente y de manera gratuita un “AudioSet”, partiendo de la base de los dataset Google facilitó a los usuarios una colección de sonidos del día a día para que sean utilizados como fuente de entrenamiento para sistemas inteligentes.

Para beneficio del proyecto, existe un apartado con sonidos de automóviles, y a su vez estos están categorizados en sonidos de arranque, sonidos al andar y otro ejemplo que es el que resulta beneficioso para el proyecto, sonidos de problemas en automóviles. Por lo cual el uso de Dataset, o Audioset en este caso no supone un problema a la hora de utilizarse.

En cuanto a Imágenes, no se encontró un dataset específicamente de luces del tablero pero la creación de un dataset es realmente sencilla, basta con seleccionar nuestras propias imágenes y con esta crear el modelo de aprendizaje, además la aplicación se retroalimenta con las imágenes que realicen los usuarios y el dataset se irá aumentando y mejorando poco a poco.

En cuanto a Hardware, lo único que será necesario por parte de los usuarios de la aplicación es poseer un teléfono con micrófono y cámara, lo cual es algo sumamente común hoy en día. Todavía no existe un requisito mínimo de que teléfono o micrófono se debe tener, pero con el prototipo se podrán realizar comparaciones entre cada uno.

***Análisis de alternativa técnica:***

Se analiza como parte de esta investigación la posibilidad de detectar los sonidos a través de la tecnología Arduino y que éste envíe lo procesado al sistema en caso de que los micrófonos de los teléfonos no sean capaces de reconocer, o no tengan la suficiente nitidez en el sonido.

A continuación se detallan los gastos y la posibilidad de obtención de todos los componentes necesarios para poder llevar a cabo esta alternativa junto al proyecto:

Figura 12. Tabla de Componentes Arduino



Como se puede ver todos los componentes son de fácil obtención y el precio no supone una barrera significativa.

Por todas estas razones se concluye que el proyecto es factible técnicamente, ya que todas las tecnologías utilizadas son de manera gratuita o de obtención sumamente sencilla.

# Justificación económica

Durante la Justificación Técnica ya se especificó algunos precios y de cuanto supone de forma económica llevar a cabo el proyecto. Pero en esta sección se ahonda en temas de almacenamiento de datos ya que este puede ser uno de los gastos más grandes del proyecto y además, el servicio de hosting de la aplicación lo cual mantendrá el servicio funcionando.

Existen dos formas de realizar el almacenamiento de datos, uno es creando una base de datos enorme, el cual guarde mucho de los sonidos para que la aplicación compare y pueda así categorizar cada una de las fallas al realizar un “Match” con alguno de los sonidos dentro de la base de datos.

Esto implica una enorme cantidad de espacio por lo cual se necesitará un servicio de hosting muy grande, y mientras más grande sea el espacio en la nube, más costoso será de mantener.

Actualmente se desconoce la cantidad exacta de espacio requerida, pero lo que se conoce es el precio que tiene mantener un servidor con un espacio de 1 TB en la nube, y esto cuesta alrededor de 170$ pesos argentinos al mes. Lo cual se traduce en el día de la fecha (20/06/2020) a menos de 1 dólar. Si bien la suma no es significativa, 1TB puede no ser un espacio significativo para la enorme cantidad de información que se requerirá.

***Análisis de alternativa económica:***

Una alternativa es utilizar como almacenamiento de datos un Dataset. Ya que este se utilizara para entrenar el sistema inteligente que detecta de donde provienen las fallas, también se analiza la opción de guardar todos estos datos dentro de un nuevo dataset, el cual supone un tamaño mucho menor de almacenamiento ya que este no tiene una estructura como tal, la información puede ser almacenada hasta dentro de un archivo CSV en forma de texto plano y así el espacio requerido para guardar este tipo de archivo se volverá significativamente menor que toda una base de datos relacional.

Otro punto a tener en cuenta es la forma en la que este proyecto pueda generar ganancias, si bien no es el fin del mismo existen algunas alternativas como subscripciones de pago o la integración de anuncios para amortiguar muchos de los gastos del mismo.

# Supuestos y restricciones

Supuestos:

* El sistema podrá tener la capacidad de aislar el sonido del problema con el sonido ambiental. (Hay un riesgo si no se cumple)
* La calidad del sonido va a ser suficiente para que la falla pueda categorizarse.

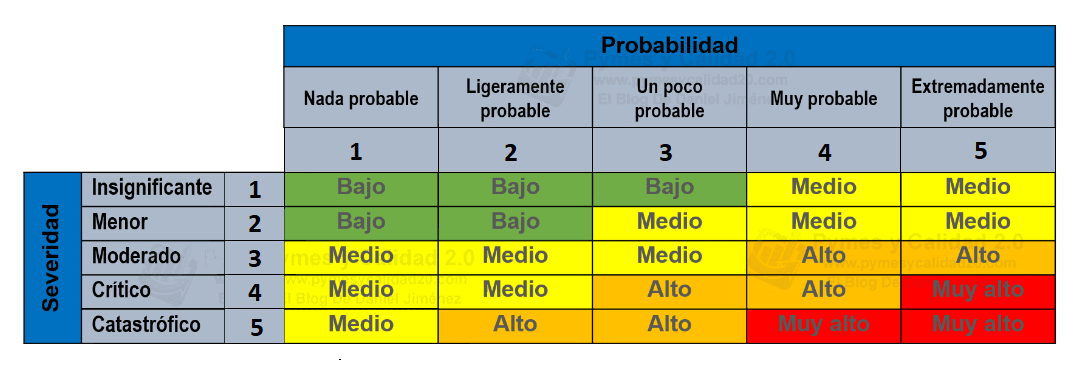
Restricciones:

* Las fechas de entrega de las etapas podrían afectar la correcta continuidad del proyecto

# Riesgos y contingencias

Para calcular la exposición de cada riesgo se utilizará una matriz de riesgo, en el cual podremos ver de una forma numérica y cuantificada que valor tiene cada riesgo. En el cuadro se ve que los riesgos de 25 son los más severos, y los de 1 los menos severos.

Figura 12. Tabla de Riesgos



**Riesgo 1:**

|  |  |
| --- | --- |
| **Riesgo** | Tiempo |
| **Probabilidad** | Bajo |
| **Causa** | Cronograma de Entregas |
| **Efecto** | Atraso del proyecto e incumplimiento de entregas |
| **Contingencia** | Ajustar próximas entregas a fines de respetar el cronograma |
| **Exposición** | Severidad 2 x Probabilidad 2 = 4 |

**Riesgo 2:**

|  |  |
| --- | --- |
| **Riesgo** | Supuesto 1: Aislamiento de Sonidos |
| **Probabilidad** | Alto |
| **Causa** | El sistema no es capaz de aislar el sonido de la falla del resto de sonidos |
| **Efecto** | Gran posibilidad de falla a la hora de categorizar el problema |
| **Contingencia** | Lograr una mejor calidad en la separación de estos problemas mediante alternativas de captación de audio |
| **Exposición** | Severidad 5 x Probabilidad 3 = 15 |

**Riesgo 3:**

|  |  |
| --- | --- |
| **Riesgo** | Supuesto 2: Calidad del Sonido |
| **Probabilidad** | Media |
| **Causa** | La calidad del sonido no es la suficiente para categorizarse |
| **Efecto** | Gran posibilidad de falla a la hora de categorizar el problema |
| **Contingencia** | Buscar alternativas que mejoren la calidad del sonido lo captado |
| **Exposición** | Severidad 4 x Probabilidad 2 = 12 |

# Capítulo 5. Presentación de Resultados

Antes de comenzar el capítulo, es necesario recordar el objetivo del proyecto para así entender mejor la presentación de todos los resultados dados en este capítulo.

El objetivo del proyecto es reconocer el problema de un automóvil a través del sonido que produce el mismo y de esta forma no solo conocer su procedencia, sino la importancia y algunas posibles soluciones.

A continuación, se detallarán los requerimientos del sistema, tanto funcionales como no funcionales, el diseño, partes esenciales del código, algunas pruebas del sistema en funcionamiento, encuestas y entrevistas a expertos mecánicos para conocer su perspectiva sobre la aplicación, y por último los resultados obtenidos.

## Requerimientos Funcionales

|  |  |
| --- | --- |
| **Requerimiento** | **Descripción del Requerimiento** |
| RF1 | El sistema debe permitir a un usuario registrarse, los campos serán: Nombre, Apellido, Contraseña, Teléfono, Mail. |
| RF2 | El sistema debe ser capaz de retroalimentar el dataset con la información que va adquiriendo de los usuarios y los problemas que estos van detectando. |
| RF3 | El sistema debe permitir al usuario loggearse con su cuenta y mostrar la información de su cuenta. Además de este poder eliminar o modificar dicha cuenta. |
| RF4 | El sistema debe ser capaz de interpretar y procesar un audio grabado en el momento. |
| RF5 | El sistema debe ser capaz de aislar el sonido el cual se desea procesar al sonido ambiente en el cual es grabado. |
| RF6 | El sistema debe ser capaz de reconocer la procedencia del problema automotor, y de esta forma categorizarlo. |
| RF7 | El sistema debe ser capaz de aislar el sonido el cual se desea procesar al sonido ambiente en el cual es grabado. |
| RF8 | El sistema al hallar un problema debe dar la respuesta al usuario con la siguiente información: Origen del problema, Importancia del problema, Posibles soluciones. |
| RF9 | El sistema debe permitir al usuario realizar una fotografía, para así utilizar la función sobre detección de problemas en el tablero. |
| RF10 | El sistema debe tener una sección para que el usuario envíe opiniones o mensajes a los administradores de la app. |

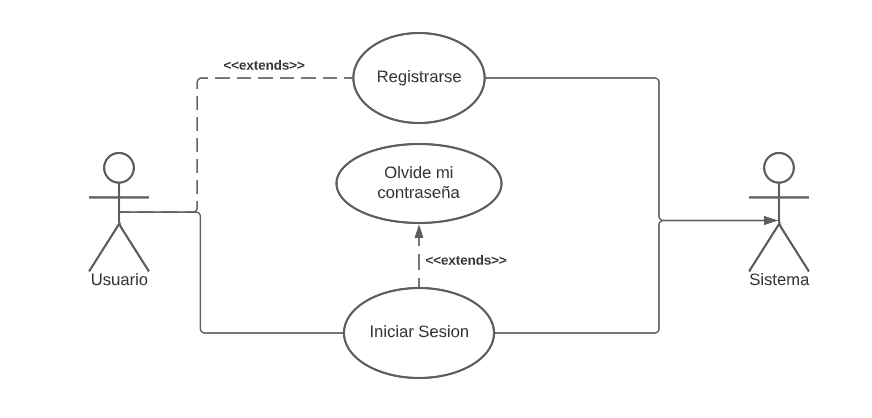
## Requerimientos No Funcionales

|  |  |
| --- | --- |
| **Requerimiento** | **Descripción del Requerimiento** |
| Requerimientos Orientados al Usuario | |
| RNF1 | El sistema podrá ser descargado en cualquier smartphone, sin importar el modelo, marca o sistema operativo. |
| RNF2 | El sistema poseerá una recuperación frente a los fallos del sistema. La información enviada por los usuarios no se perderá en caso de que suceda un error. |
| Requerimientos de Seguridad | |
| RNF3 | El sistema contará con un sistema de loggeo para que solo usuarios registrados puedan utilizar la aplicación. |
| RNF4 | El sistema contará con la confirmación vía mail para los usuarios. |
| RNF5 | El sistema contará con confidencialidad, la información manejada será protegida de acceso no autorizado y divulgación. |
| RNF6 | El sistema contará con integridad, ya que la información manejada por éste será objeto de cuidadosa protección contra la corrupción y estados inconsistentes. |
| Requerimientos de Rendimiento y Escalabilidad | |
| RNF7 | Se prevé que el sistema tenga tiempos de respuestas menores a los 10seg. |
| RNF8 | El sistema contará con actualizaciones de mejoras de rendimiento y en sus funciones. |

|  |  |
| --- | --- |
| Requerimientos de Disponibilidad | |
| RNF10 | El sistema está pensado para tener un esquema de disponibilidad, una vez que se pase de Testing a Producción, tendrá una semana de disponibilidad para personas preseleccionadas. Luego de esto pasará una última semana fuera de servicio para últimos retoques. Para finalmente lanzar la aplicación al público, en caso de tener que bajar el servicio los usuarios serán notificados previamente. |
| RNF11 | El sistema debe estar conectado a una red para funcionar, ya sea wifi o utilizando los datos del teléfono. |
| Requerimientos de Continuidad | |
| RNF12 | El sistema contará con backups cada dos semanas. |
| Requisito de Soporte | |
| RNF13 | El sistema será fácil de instalar, fácil de mantener y contará con un código y diseño documentado. |
| Requisitos de Hardware | |
| RNF14 | El sistema funcionará en smartphones, y estos deben tener la capacidad de grabar correctamente los sonidos. |

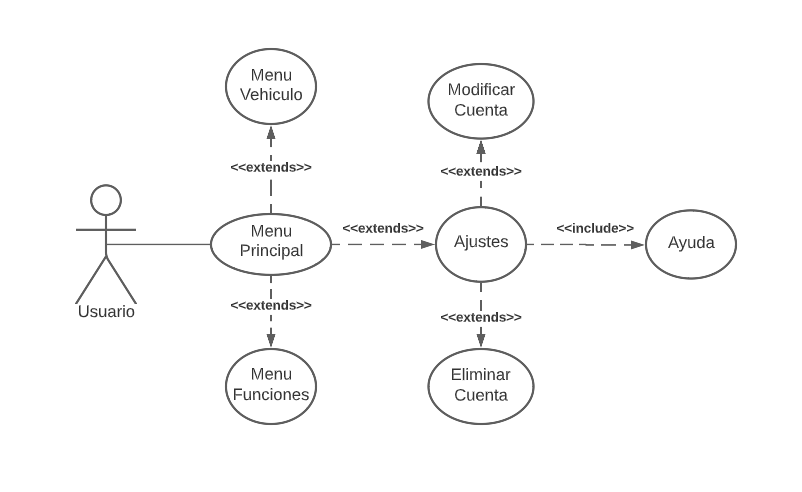
## Casos de Uso

*Caso de uso 1: Iniciar Sesión*



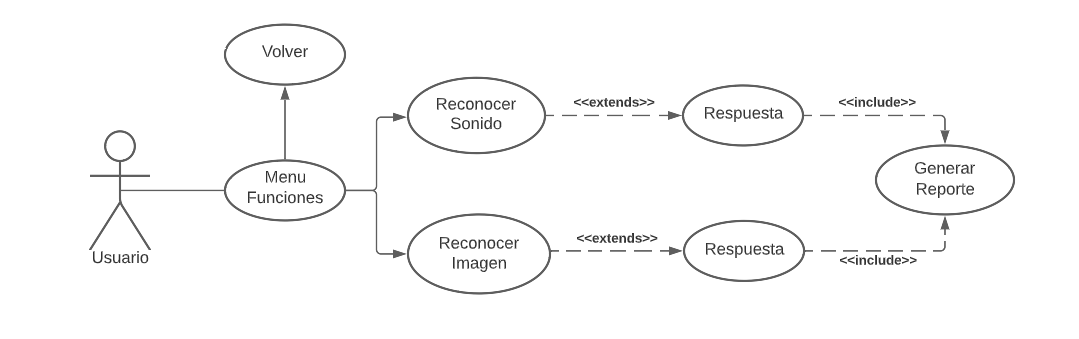
|  |  |  |
| --- | --- | --- |
| Título: Iniciar Sesión | | |
| Identificador | CU1 | |
| Nombre | Iniciar Sesión | |
| Versión | 1.0.0 | |
| Autores | Sistema | |
| Fuentes | Usuario | |
| Descripción | El Usuario inicia sesión en el sistema | |
| Alcance | Usuarios | |
| Nivel | Tarea Primaria | |
| Actor Principal | Usuario | |
| Actor Secundario | Sistema | |
| Pre-Condición | Abrir la aplicación | |
| Post-Condición | Sesión Iniciada | |
| Condición Final con Éxito | Ingreso en el sistema | |
| Condición Final con fracaso | *Error: Esa cuenta no existe, intente nuevamente.*  *Error: La contraseña es incorrecta.* | |
| Trigger | Login | |
| Secuencia Normal | 1 | El Usuario desea iniciar sesión. |
| 2 | El Sistema le pide identificación. |
| 3 | El Usuario coloca su email y contraseña correctamente. Si los datos no son correctos se genera la excepción 2-1. |
|  | 4 | El Sistema permite el acceso y muestra el menú principal. |
| Excepción  2 -1 | 1. El Usuario ingresa datos incorrectos. 2. El Sistema no permite el acceso y envía un mensaje al usuario con el problema. | |
| Frecuencia | Diaria, muy frecuente | |
| Importancia | Alta | |

*Caso de uso 2: Menú Principal*



|  |  |  |
| --- | --- | --- |
| Título: Menú Principal | | |
| Identificador | CU2 | |
| Nombre | Menú Principal | |
| Versión | 1.0.0 | |
| Autores | Usuario | |
| Fuentes | Usuario | |
| Descripción | El usuario navega por el menú principal del sistema | |
| Alcance | Usuarios | |
| Nivel | Tarea Primaria | |
| Actor Principal | Usuario | |
| Actor Secundario | Sistema | |
| Pre-Condición | Loggearse | |
| Post-Condición | Llegar a la opción deseada | |
| Condición Final con Éxito | Bienvenido a “Opción Deseada” | |
| Condición Final con fracaso | *Error: A ocurrido un error al intentar seleccionar la opción* | |
| Trigger | Menú Principal | |
| Secuencia Normal | 1 | El usuario ingresa al sistema y ve el menú principal |
| 2 | El usuario ingresa una opción:   * En caso de querer ingresar al “Menú Vehículo” seguir la extensión 2-1. * En caso de querer ingresar a “Ajustes” seguir la extensión 3-1. * En caso de querer ingresar al “Menú Funciones” seguir la extensión 4-1.   El usuario para salir del sistema selecciona la opción “Salir”. |
| 3 | El sistema muestra una confirmación mediante un cartel que se dispara. |
| 4 | El usuario confirma que quiere salir del sistema. |
| 5 | La aplicación se cierra. |
| Extensión 2-1 | 1. El Usuario ingresa a Menú Vehículo. 2. El Sistema muestra todas las opciones que puede realizar el usuario dentro de este menú. 3. El Usuario realiza las operaciones que desea y regresa al menú principal. | |
| Extensión 3-1 | 1. El Usuario ingresa a Ajustes. 2. El Sistema muestra todas las opciones que puede realizar el usuario dentro de los ajustes. 3. El Usuario realiza las operaciones que desea y regresa al menú principal. En caso de eliminar la cuenta el Usuario volverá al CU1. Iniciar Sesión. | |
| Extensión 4-1 | 1. El Usuario ingresa a Menú Funciones. 2. El Sistema muestra todas las opciones que puede realizar el usuario dentro de este menú (todas estas opciones se especifican en el CU3. Menú Funciones). 3. El Usuario realiza las operaciones que desea y regresa al menú principal. | |
| Frecuencia | Diaria, muy frecuente | |
| Importancia | Alta | |

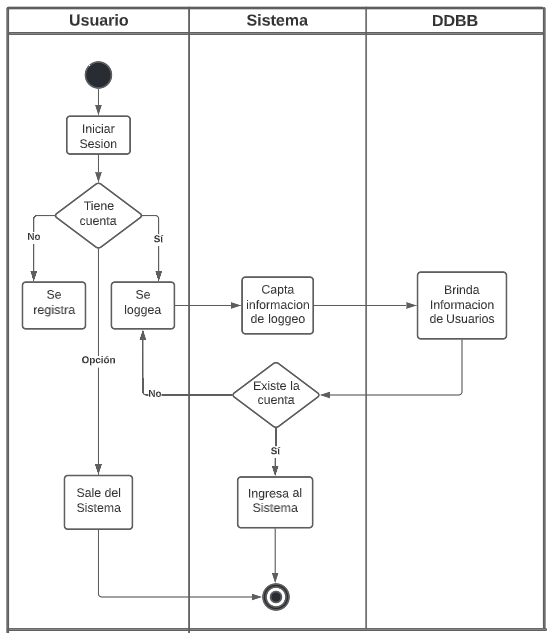
*Caso de uso 3: Menú Funciones*



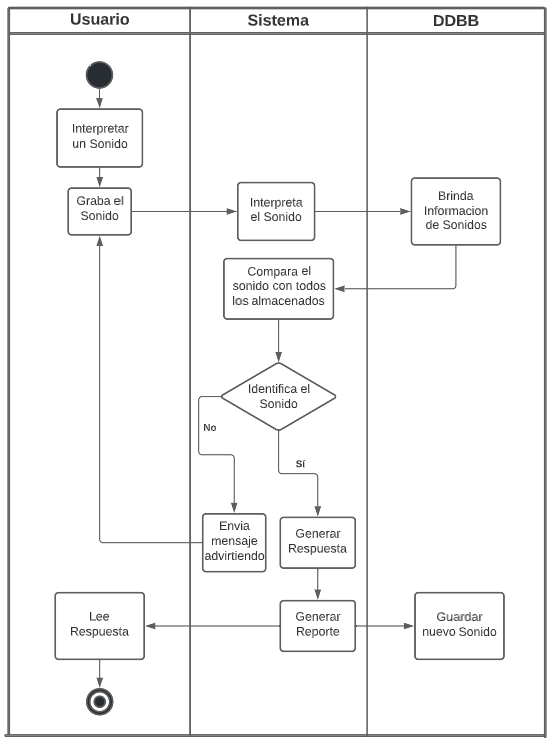
|  |  |  |
| --- | --- | --- |
| Título: Menú Funciones | | |
| Identificador | CU3 | |
| Nombre | Menú Funciones | |
| Versión | 1.0.0 | |
| Autores | Usuario | |
| Fuentes | Sistema | |
| Descripción | El usuario desea utilizar las funciones del sistema, para esto ingresa al menú de funciones y coloca una de las dos opciones:   * Reconocer Sonido * Reconocer Imagen | |
| Alcance | Usuarios | |
| Nivel | Tarea Primaria | |
| Actor Principal | Usuario | |
| Actor Secundario | Sistema | |
| Pre-Condición | Estar loggeado con la cuenta correspondiente | |
| Post-Condición | Respuesta: xxxxxxx | |
| Condición Final con Éxito | Se ha encontrado el problema del vehículo | |
| Condición Final con fracaso | *Error: No se ha podido reconocer, intente nuevamente* | |
| Trigger | Menú Funciones | |
| Secuencia Normal | 1 | El usuario ingresa al menú funciones |
| 2 | El usuario ingresa una opción:   * En caso de querer reconocer un sonido selecciona “Reconocer sonido” (extensión 2-1). * En caso de querer reconocer una imagen selecciona “Reconocer imagen” (extensión 3-1).   El usuario para volver al menú principal selecciona la opción “Volver”. |
| 3 | El sistema envía al usuario al menú principal. |
| Extensión 2-1 | 1. El Usuario selecciona Reconocer Sonido. 2. El Sistema muestra un botón para empezar a reconocer el sonido que el usuario grabe 3. El sistema reconoce el sonido y envía una respuesta. En caso de no reconocer el sonido se dispara excepción 1-1. 4. El sistema luego de generar la respuesta crea un reporte el cual es guardado dentro de las bases de datos. 5. El usuario ve la respuesta y vuelve al menú de funciones. | |
| Extensión 3-1 | 1. El Usuario selecciona Reconocer Imagen. 2. El Sistema abre la cámara para que el usuario saque una fotografía. 3. El sistema reconoce la fotografía y envía una respuesta. En caso de no reconocer la fotografía se dispara la excepción 1-1. 4. El sistema luego de generar la respuesta crea un reporte el cual es guardado dentro de las bases de datos. 5. El usuario ve la respuesta y vuelve al menú de funciones. | |
| Excepción 1-1 | 1. El sistema no reconoce lo enviado por el usuario 2. El sistema dispara un cartel advirtiendo que *“No se ha podido reconocer, intente nuevamente”* 3. El sistema envía al usuario de nuevo a la opción que éste seleccionó para que vuelva a intentar un reconocimiento. | |
| Frecuencia | Diaria, muy frecuente | |
| Importancia | Alta | |

## Diagrama de Actividad

*Loggeo de Usuario*



*Captar Sonido*

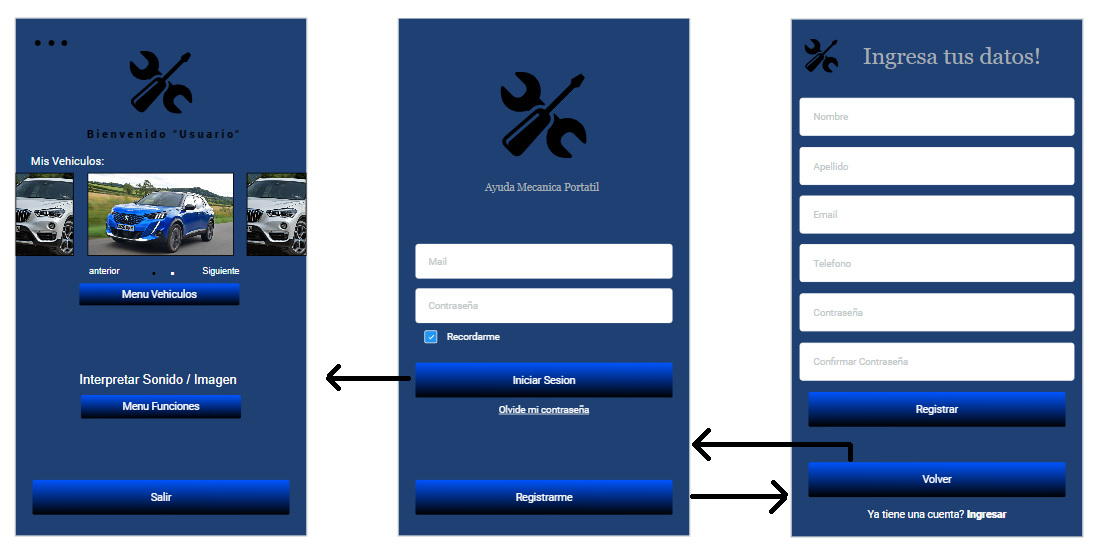


## Pantallas

*Inicio de Sesión / Registrarse / Menú principal:*

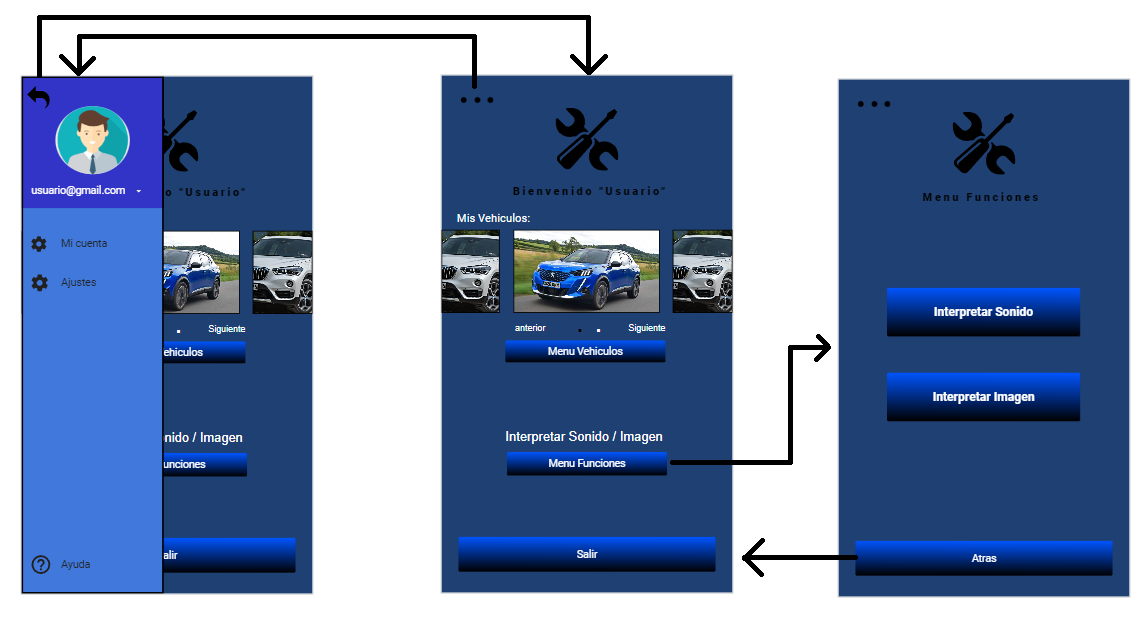
En estas pantallas se observa lo primero que ve el usuario al ingresar a la aplicación, el cual es el inicio de sesión. Si el usuario no posee una cuenta debería registrarse en el sistema, lo cual lo enviaría a la pantalla de la derecha.

En caso de estar registrado, el usuario colocando su mail y contraseña correctamente debe ser enviado al menú principal, el cual es la pantalla de la izquierda.



*NavBar / Menú Principal:*

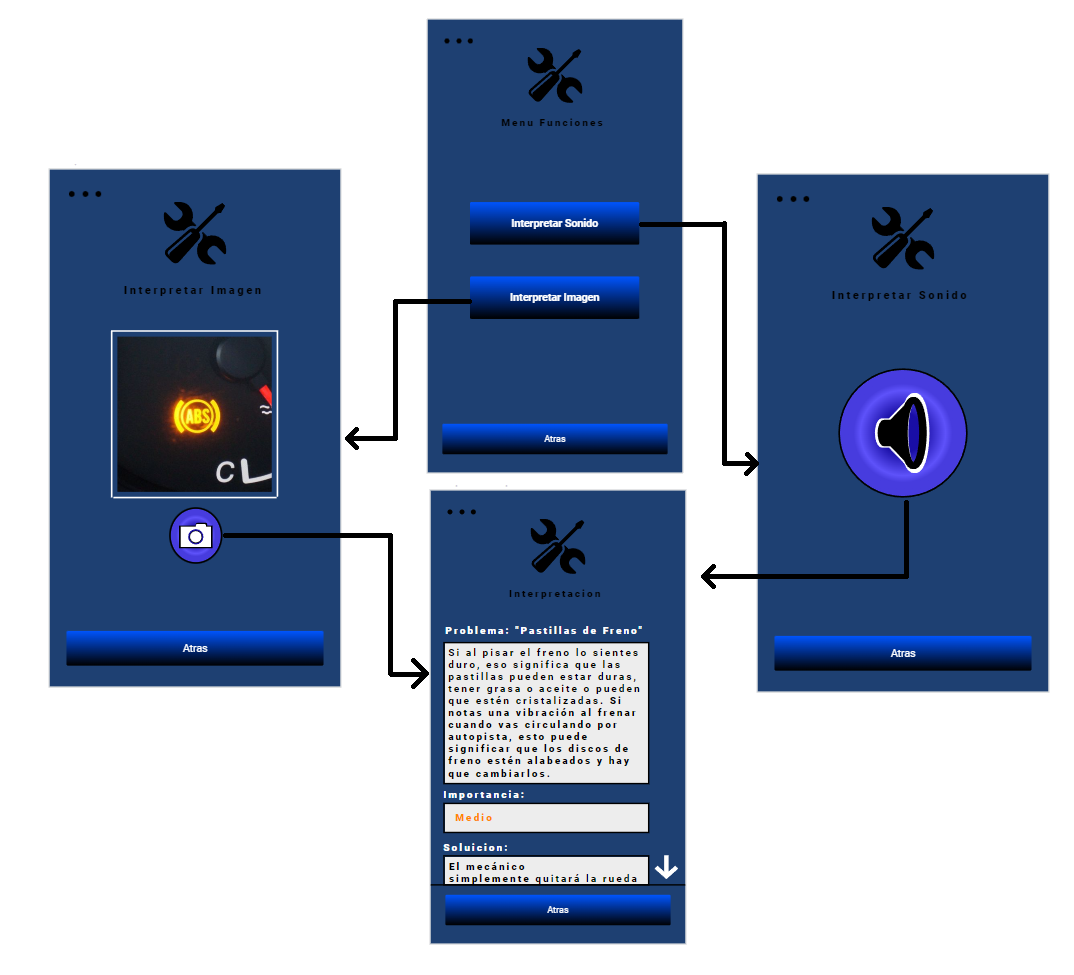
En este caso se enseña las funciones que trae el menú principal, algunas de estas son el menú funciones, el cual tiene todas las funciones de reconocimiento del sistema, como se ve en la pantalla de la derecha y por otro lado se observa el NavBar de lado izquierdo.



*Menú Funciones:*

En este menú funciones se observa cómo se puede interactuar con los botones de Interpretar Sonido e Imagen.

El primero envía al usuario a una pantalla en la cual este puede grabar el sonido en cuestión y el sistema reconoce dicho sonido y da una posible respuesta.

Por otro lado, está el botón de interpretar imagen que al igual que el anterior, envía al usuario a una pantalla en la cual este puede sacar una fotografía y el sistema le va a brindar toda la información acerca del problema que está teniendo.

### Proceso de desarrollo:

Para realizar el desarrollo del sistema fueron necesarios dos componentes principales, el primero fue la forma de realizar la aplicación Mobile y el segundo, el desarrollo de los modelos de aprendizaje los cuales detectan y clasifican las imágenes y sonidos de los problemas automotores.

Del lado de desarrollo Mobile hubo dos opciones principales, la primera es un Framework creado por Google llamado “Flutter” el cual permite realizar proyectos tanto en IOS (Apple) como en Android, y la segunda opción fue Android Studio, que fue finalmente la opción elegida, debido al conocimiento previo que tenía sobre este IDE y su metodología de trabajo.

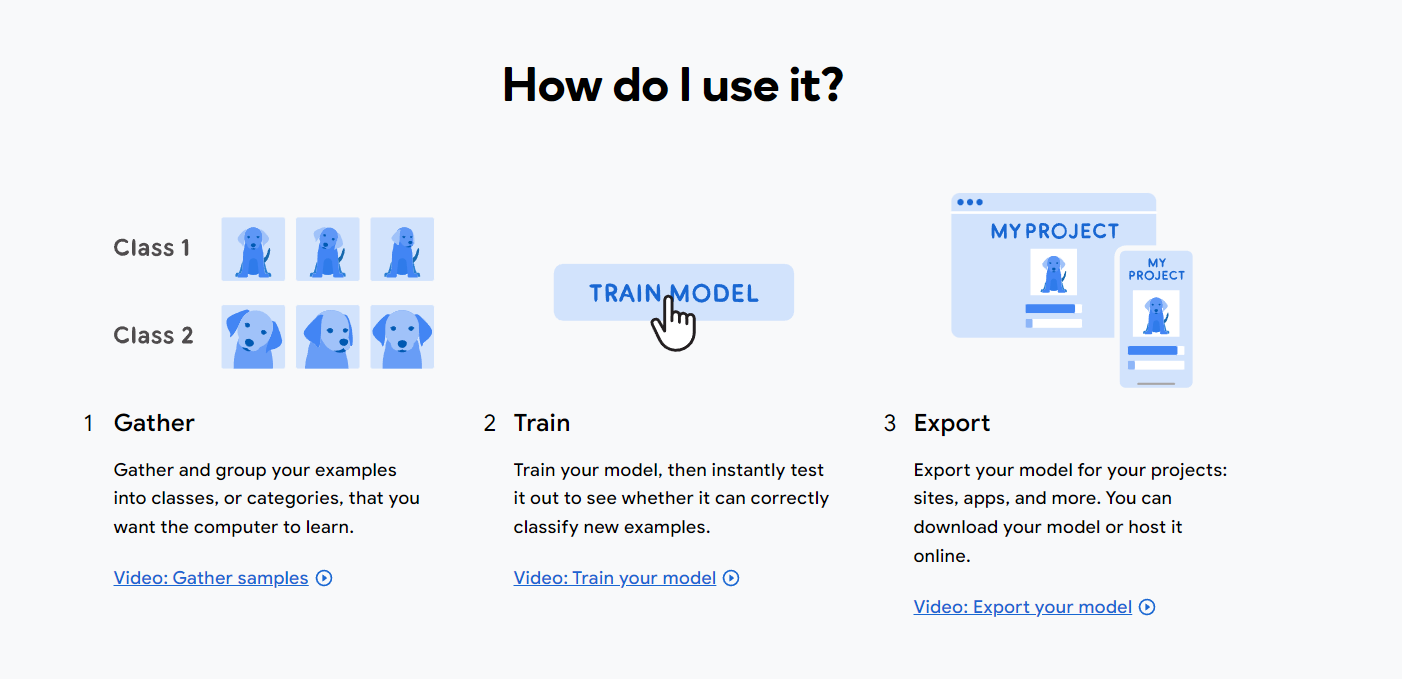
Android Studio usa como lenguajes de desarrollo Java, Kotlin (Que es un lenguaje que parte de Java pero con una sintaxis mucho más práctica) y XML.

Estos lenguajes fueron principalmente los que nos acompañaron desde el primer momento de la carrera y fue esta la razón por la cual preferí quedarme con Android Studio sobre Flutter (Framework que iba a tener que aprender completamente de cero e integrarlo en un sistema sumamente complejo como lo es este).

Luego el otro gran componente del sistema es toda la parte de IA, la cual comprende y clasifica los apartados de “Problemas Automotores”. Si bien el primer enfoque para trabajar con IA, entrenamiento de modelos y puesta a prueba del mismo es Python, la realidad es que encontré una alternativa más sencilla online la cual crea, entrena y provee un modelo el cual es capaz de realizar acciones como las que estaba precisando para el proyecto, la cual es “Teachable Machine”.

Teachable Machine es una página web creada por Google, la cual nos permite crear redes neuronales y exportarlas de manera completamente gratuita.

Figura 13. Cómo utilizar Teachable Machine



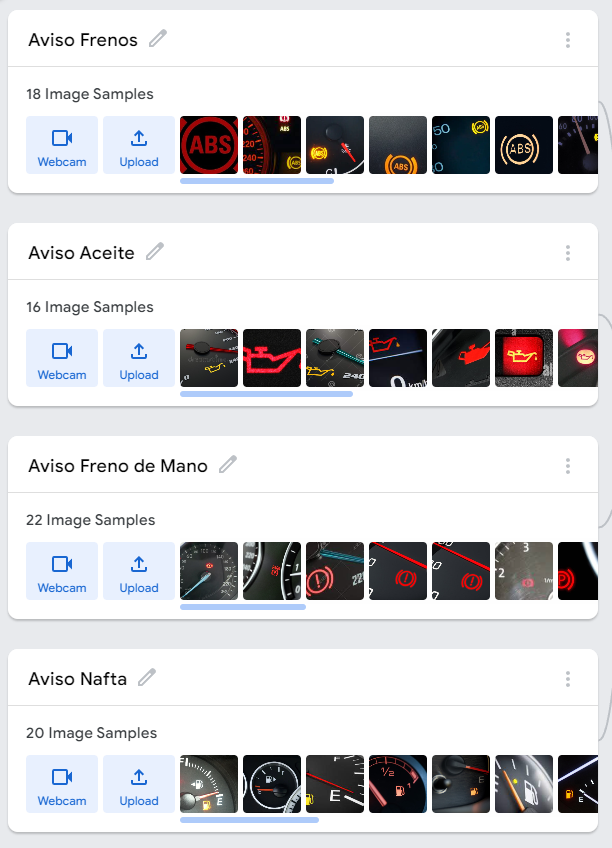
Como vemos en la imagen, la forma de utilización es tan sencilla como, captar todos los ejemplos que tenemos para entrenar la red, dividir estos ejemplos por categoría e indicarle a qué categoría pertenece cada grupo de ejemplos.

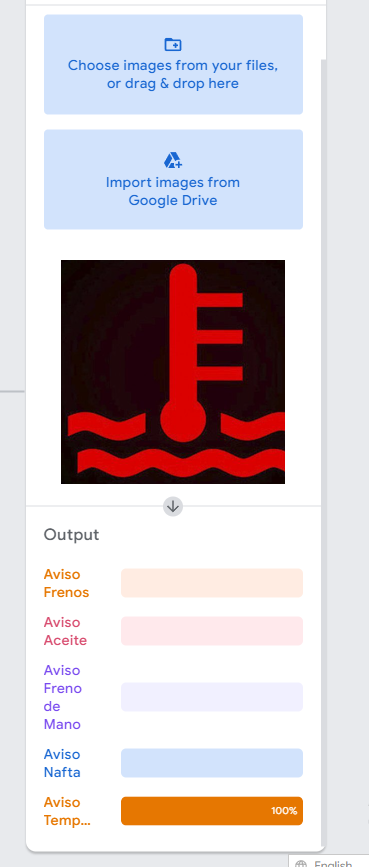
Una vez realizado esto, seleccionamos el botón de “Entrenar” y luego exportamos el modelo entrenado de la forma que más nos convenga.

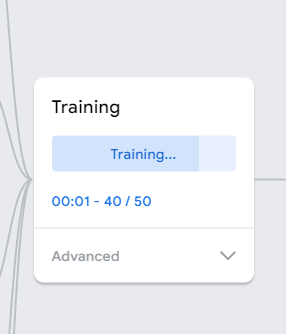
Ejemplo de clasificación de Imágenes en este caso:

Primer Paso:

Figura 13.1 Carga de imágenes



Segundo Paso:



🡪

Tercer paso:

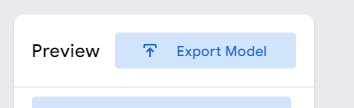
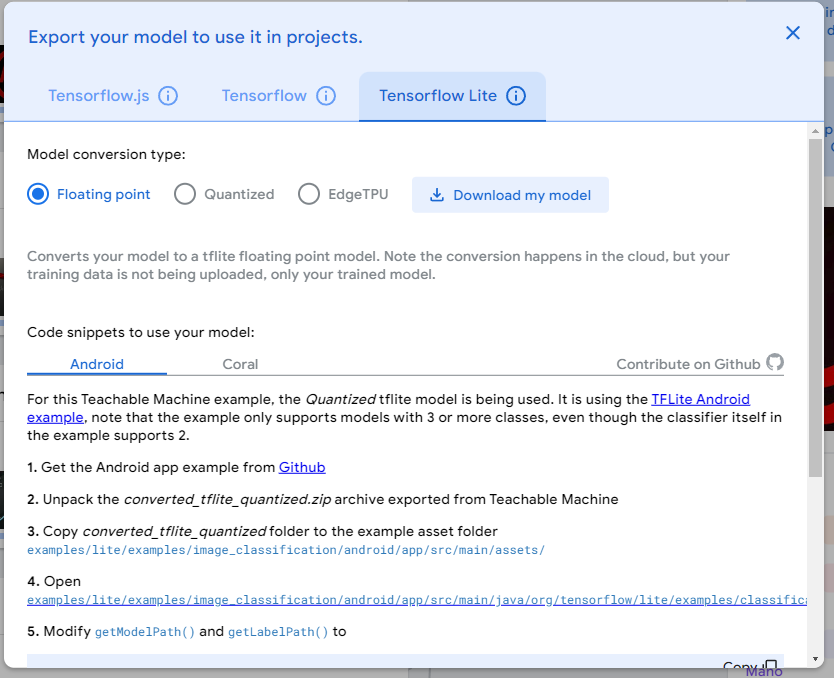


Figura 13.2: Exportar Modelo

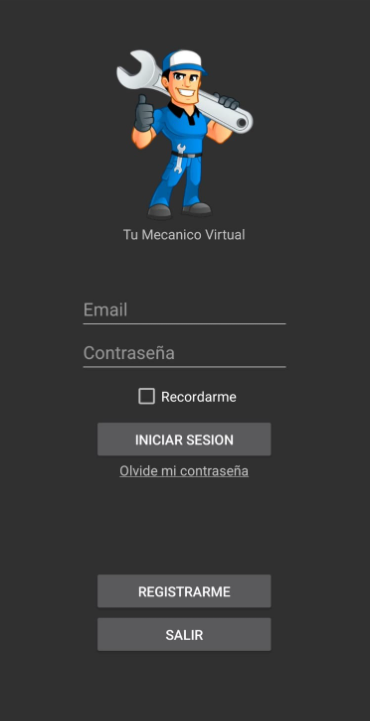


Con la opción “Tensorflow Lite” indicamos que va a ser un modelo para Mobile y con Floating point vamos a poder acceder a este modelo desde el código.

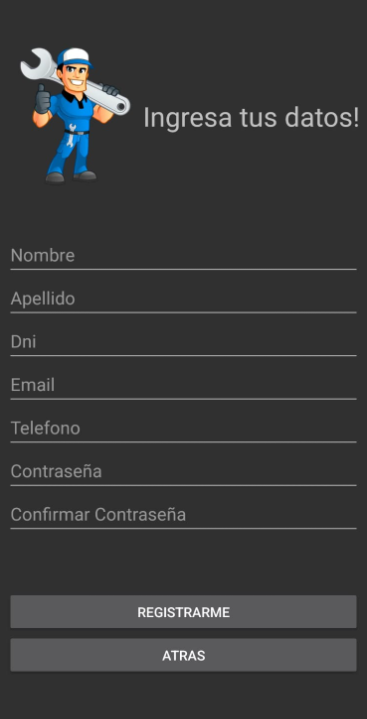
Y de esta forma ya tendríamos el modelo, más adelante explico cómo lo incluí dentro del proyecto para que las fotos puedan ser clasificadas.

En cuanto a Android Studio realice un curso online para poder crear la aplicación, la cual cuenta actualmente con 7 pantallas.

Los resultados son los siguientes:

*Pantalla Login*:

*Pantalla Registro*:



*Menú Principal*:

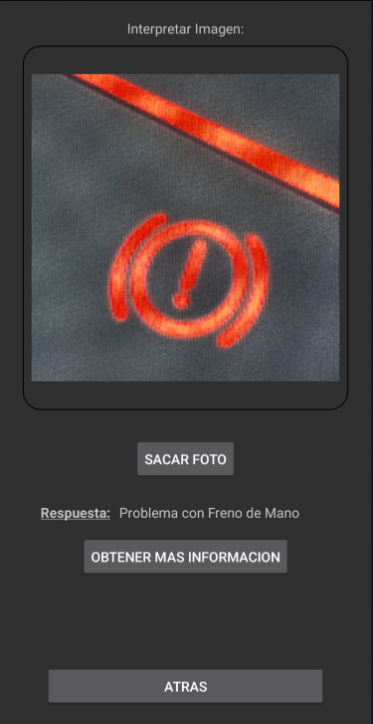


*Menú Funciones*:



*Interpretar Imagen*:

Como observamos en estas imágenes, el usuario debe seleccionar el botón “Sacar Foto” para así la cámara del teléfono se active y pueda tomar dicha fotografía, la cual luego es clasificada. Una vez seleccionada la opción obtener más información, el usuario será enviado a la sección explicada en el siguiente apartado.



🡪

*Interpretar Sonido*:

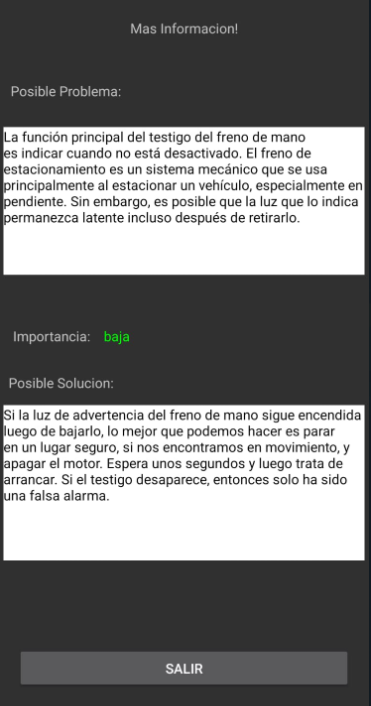
En esta pantalla, el usuario debe seleccionar el botón de grabar y colocar el teléfono cerca del sonido que genera el auto, luego de unos segundos selecciona “Frenar” y automáticamente el modelo comenzará a analizar el sonido, y así dar una respuesta. En caso de querer escuchar el sonido basta con seleccionar “Escuchar” y si se desea más detalle con el botón “Obtener más información” lo llevará a una pantalla detallada más adelante.



*Información del Problema*:

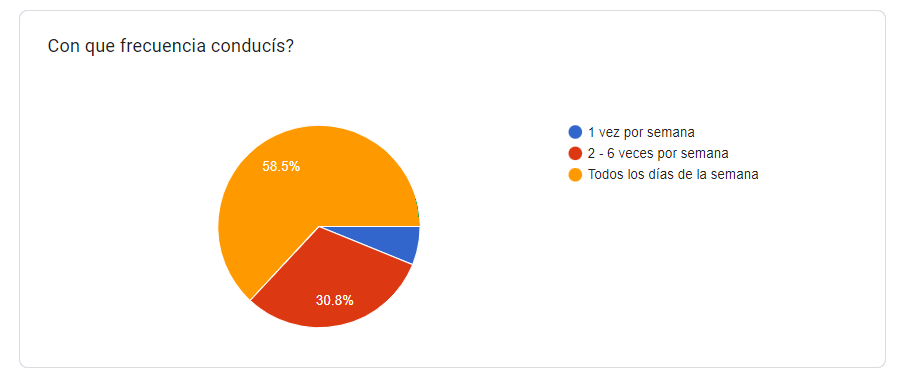
En esta se ve información más detallada de lo captado por la función, ya sea interpretar sonido o imagen.

Se puede observar posibles causas del problema, la importancia que este problema puede tener y una serie de posibles soluciones.

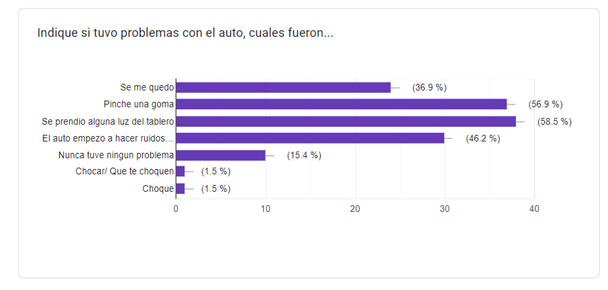


Además, para adquirir información importante sobre los usuarios a los cuales va dirigido el sistema se realizó una encuesta sobre 100 personas de entre 18 a 30 años, esto para tener un parámetro sobre la accesibilidad, potencial y aceptación que tendrá la aplicación en el momento de ser lanzada.

Se registró la siguiente información:

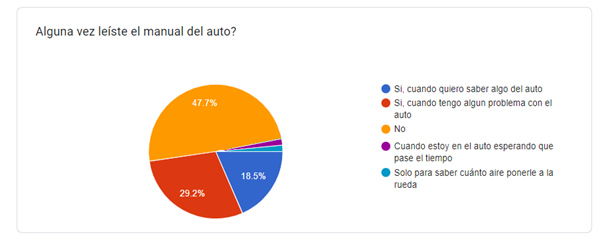


De todos los usuarios, la mayoría utiliza el auto todos los días de la semana, por lo cual se concluye que la aplicación tendrá un uso diario de usuarios, además más adelante veremos que en una de las preguntas a la gran mayoría le sucedió un problema con el auto el cual se podría haber consultado con la aplicación.



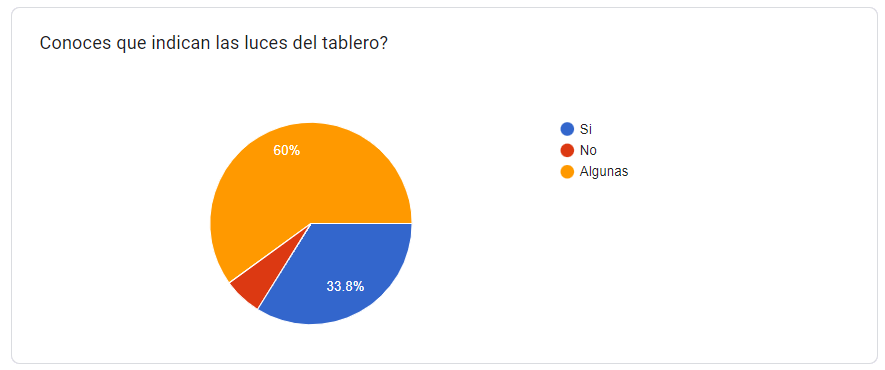
Aquí podemos ver que, si bien uno de los problemas más grandes de los usuarios es que se le pinchó una goma, es el único problema donde la aplicación no tiene un objetivo a resolver…

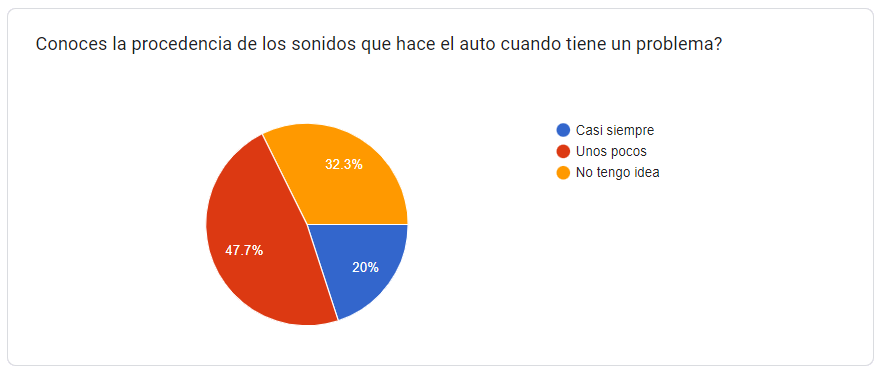
El resto de problemas tienen que ver sobre luces del tablero o ruidos extraños, que además se puede deducir que estos fueron alerta previa a que el auto se quede (no arranque) como sucede la mayoría de las veces.



Esta es una de las preguntas que dispararon mi idea sobre este proyecto, hoy en día la gente más joven, con un promedio de 25 años, raramente eligen solucionar estos problemas visto anteriormente con el manual de usuario respectivo de cada vehículo. Únicamente un cuarto de las personas encuestadas veía el manual en los momentos que lo necesitaban…

Más adelante veremos que el 100% de los encuestados posee un teléfono celular a la mano para consultar cualquiera de estas cosas, de manera accesible y sencilla, como es la costumbre en estos tiempos donde estamos conectados a la tecnología permanentemente.

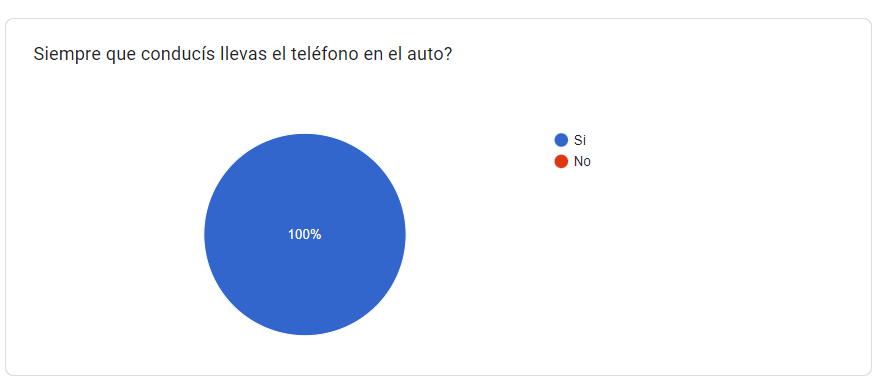




La mayoría de usuarios conocen muy poco sobre todas las señalizaciones que un auto posee antes de que surja un problema más grave, por eso la importancia de llevar a todas las personas que manejan una forma sencilla de chequear estos problemas.

**Encuesta de aceptación:**

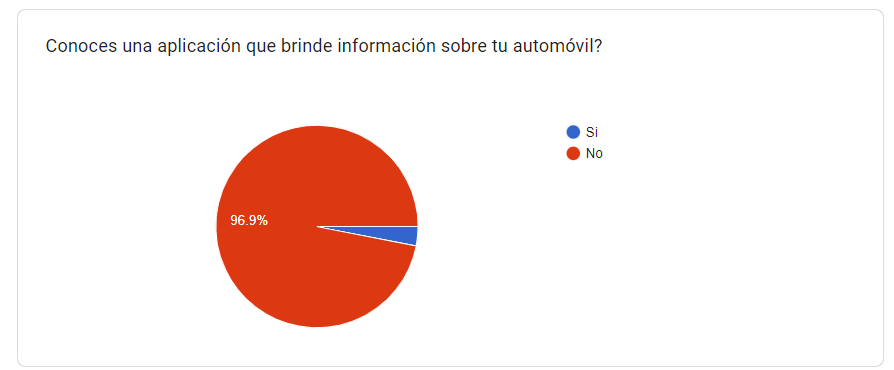
Para comenzar, el 100% de usuarios llevan el teléfono en el vehículo.



Esto significa que la accesibilidad de los usuarios para tener la aplicación en caso de ser necesario usarla no va a presentar ninguna complicación.



Además, la gran mayoría de encuestados accedieron a utilizar una aplicación con las características mencionadas durante todo el proyecto, esto da un aspecto sumamente positivo a la idea de la aplicación y su alcance.



Y como último dato de interés, casi ningún encuestado conoce una aplicación con características similares a las expuestas anteriormente, y los que pusieron que conocían, hablaron de aplicaciones de empresas como Volkswagen que posee una app pero no posee el mismo objetivo de esta aplicación.

Con toda esta información recolectada se concluye que la aplicación va a ser más que bien recibida y la aceptación es enorme.

# Capítulo 6. Implicancias, conclusiones y recomendaciones

## Conclusiones

Retomando el objetivo principal del proyecto que se fue desarrollando recordamos que se buscaba llevarle a las personas que, como la gran mayoría no tienen un

entendimiento profundo de los automóviles, una forma fácil de saber que le está pasando

su auto, porque existe este ruido, esa luz del tablero, el nivel de emergencia y sus posibles soluciones.

Todo esto a través del micrófono para la detección de sonidos y con una cámara para las imágenes, que ayudado con la tecnología de redes neuronales artificiales dará respuesta a todos estos problemas…

La conclusión es satisfactoria.

Dando una mirada hacia el prototipo se ve que se logró todo lo propuesto, que, si bien el prototipo no da los resultados de manera 100% correctas, se ve que logra su cometido.

Para lograr esa efectividad que se busca y se habló en el proyecto, el tamaño del dataset y el entrenamiento del modelo debe ser mucho más exponencial, el cual lleva meses de preparación y no se hubiese podido realizar las pruebas donde se muestra que efectivamente el proyecto se logra.

En cuanto a las tecnologías utilizadas, se descartó el uso de Arduino debido a que se ve que los resultados hechos con los teléfonos celulares son satisfactorios.

El uso de TensorFlowLite para todo el reconocimiento y categorización resultó indispensable, sin esta librería hecha para Mobile no se hubiese podido llevar a cabo el proyecto, el procesamiento que realiza sobre la información utilizando modelos creados ya se en Python o en mi caso con Teachable Machine.

Y por último el IDE de Android Studio facilitó la creación del prototipo, si bien la aplicación será lanzada únicamente para teléfonos Android se provee en un futuro realizar la aplicación en tecnologías que soporten tanto Android como IOS.

## Futuras líneas de investigación

El punto más fuerte de la investigación es la escalabilidad que esta posee, la aplicación puede expandirse en todos los puntos y mejorar radicalmente.

Si bien las funcionalidades de detectar y clasificar tanto imágenes como sonidos son un punto crucial de la aplicación, se puede crear todo un sistema alrededor de estos.

Donde estas funcionalidades quedan como un apartado de la aplicación y además se pueda realizar muchas otras acciones.

Para empezar, se tuvo que dejar afuera la funcionalidad de que cada usuario pueda dar de alta su vehículo, esto debido a la cantidad inmensa de vehículos y validaciones que se debe hacer para que aparezcan todas las marcas y modelos. Además, una vez que se haya un alta de vehículo el sistema podría mostrar especificaciones técnicas del mismo, el manual digitalizado, entre muchas otras opciones…

Luego expandir el proyecto a distintos puntos, distintos países para así tener un alcance mucho mayor.

# BIBLIOGRAFÍA

“Shazam: Cómo funciona el algoritmo de reconocimiento de canciones de la popular aplicación” – Guido Sirna – ACADEMIA.

PFC\_AdquisicionAudioCodecsDSP\_MGH\_2011(uam.es) - UNIVERSIDAD AUTÓNOMA DE MADRID ESCUELA POLITÉCNICA SUPERIOR.

“Clasificación de imágenes”:

https://repositorio.uam.es/bitstream/handle/10486/698211/fragua\_baeza\_angel\_tfg.pdf?sequence=1

Minería de Datos: Conceptos y Tendencias - José C. Riquelme, Roberto Ruiz, Karina Gilbert

https://developer.android.com/studio

https://developer.android.com/training/basics/firstapp?hl=es-419

https://formacion.intef.es/catalogo/mod/book/view.php?id=69&chapterid=338

https://empresas.blogthinkbig.com/creando-y-desplegando-modelos-de-machine-learning-con-teachable-machine/

Requisitos no funcionales – EcuRed

https://teachablemachine.withgoogle.com/

**Libros Consultados:**

UML Práctico: Aprende UML paso a paso - Javier Martin Juan y Lorena López Resusta

Redes Neuronales Guía Sencilla De Redes Neuronales - Rudolph Russell

Una aproximación práctica a las redes neuronales artificiales - Eduardo Francisco Caicedo y Jesús Alfonso López

El proceso de investigación - Carlos Sabino