

# Andrea Asistente Personal

Andres Sebastian Salazar Alturo, Brahian Steven Espinosa Ruiz

Departamento Tolima

Universidad de Ibagué

2420151021@estudiantesunibague.edu.co, 2420151009@estudiantesunibague.edu.co

**Resumen-** Este documento presenta el desarrollo del proyecto de Andrea Asistente Personal para la clase de Inteligencia Artificial, en este informe se presentan los aspectos importantes, librerías implementadas, algoritmo de aprendizaje automático y lógica de programación.

**Palabras Clave-** Asistente personal, librerías, python3, aprendizaje automático.

## I. OBJETIVOS

- Objetivo general: Desarrollar un asistente personal capaz de realizar diversas tareas aplicados en diferentes campos aplicando asistencia artificial.
- Objetivos específicos: Implementar reconocimiento de voz para acceder a diferentes clases o tareas.
- Utilizando un algoritmo de aprendizaje automático, entrenar y realizar las tareas eficientemente.

## II. INTRODUCCIÓN

La idea de desarrollar un asistente personal surgió de la necesidad de incorporar la tecnología a las tareas más sencillas [1] del día a día para una persona que no sufre de ninguna discapacidad como las que tienen alguna y algunas tareas pueden ser más complicadas, por ello se desarrolló el algoritmo que puede ser implementado en domótica [2], asistente personal en celulares o automóviles e incluso la industria.

Para el desarrollo de este proyecto se realizó en Linux más específicamente en Ubuntu 16.04, programado en python3 y utilizando dos librerías fundamentales para su funcionamiento las cuales son: PyAudio y SpeechRecognition; En este caso el asistente funciona con clasificación supervisada, debido a que se generó una base de datos; para el entrenamiento se utilizó el RandomForest o árbol de decisiones y así lograr clasificar la entrada de por voz al sistema. Como hardware para este caso de desarrollo se utilizó el asistente en un computador portátil y como entrada se implementó un micrófono externo.

## III. METODOLOGÍA

Para desarrollar el asistente se manejara machine learning [3] [4] específicamente un clasificador supervisado en el cual se divide en diferentes fases: El tipo de información a suministrar, los descriptores o información determinante que usará el sistema para clasificar, la acciones que se ejecutaran mediante la información de entrada que serían dentro del machine learning las clases, la realización de una base de

datos en la cual tendrá en cuenta todo lo anteriormente dicho, el entrenamiento el cual fue realizado por Random Forest.

Primero tomando en cuenta que la información la cual se le va a entrenar al clasificador va a ser las cadenas de palabras entregadas por la librería Speech Recognition la cual mediante un API de google cloud y otra librería la cual toma y guarda la información de un micrófono llamada Pyaudio convertía directamente un audio a textos.

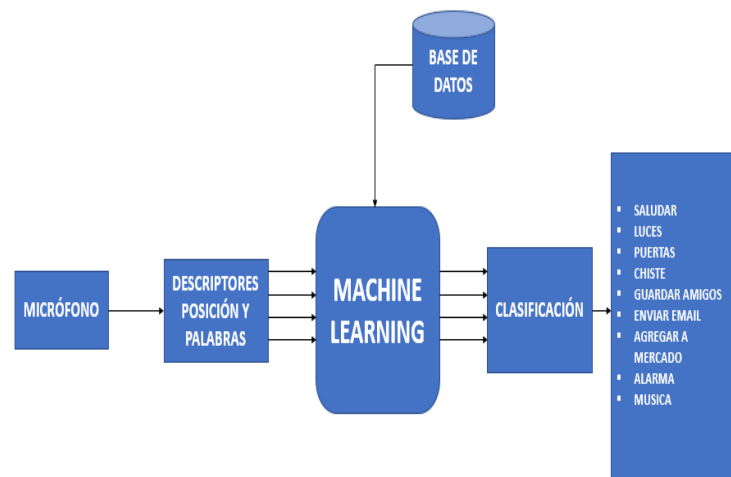


Imagen1. Diagrama de bloques del sistema.

Se plantea un total de 11 acciones que bien pueden variar tanto el tipo de acción como la cantidad de funciones, pero en este caso se plantearon las siguientes funciones:

- Saludo
- Encender Luz
- Apagar Luz
- Mercado
- Alarma
- Chiste
- Contactos
- Abrir puerta
- Cerrar puerta
- Reproducir música
- Enviar correo

Durante el desarrollo del asistente se planteó primero la base de datos que correspondería principalmente a las palabras claves en las cual va a interpretar, y de ahí se toma básicamente un máximo de 6 palabras por cadena de texto para luego transformar esas palabras en un correspondiente en ASCII de toda la palabra pasando letra por letra a este código. Después de codificar la información para que pueda ser clasificada se implementa random forest para su clasificación.

Siendo random forest un modelo de clasificación que comparten similitudes con el árbol de decisiones, pero llegando a mejorar su predicción reduciendo el error y volviéndolo más robusto usando un proceso de empaquetamiento llamado bagging. En python este clasificador puede ser usado mediante la librería del mismo nombre que reúne toda la información de entrenamiento y entrega un modelo dentro de una caja negra que es guardada en una variable para su uso en la correspondiente clasificación. Para una mayor precisión se modifica la característica max\_depth que corresponde a la cantidad de nodos que posee el modelo.

#### IV. RESULTADOS

Tomando una base de datos de un total de 1500 datos en los que se encuentran las diferentes posiciones en las cuales se pueden acomodar las frases ya anteriormente seleccionadas y con su respectiva clase, esta información se codifica y luego de tener su equivalente en ASCII se adquiere el modelo proporcionado por Random Forest con un max\_depth de 20 para asegurar una precisión casi del 100% planteada directamente a la base de datos.

Ya durante las pruebas del sistema se eliminaron los conectores tales como el, los, la, un entre otros filtrar potenciales errores dentro de la prueba. Igualmente se manejó una diadema con micrófono para asegurar una correcta asimilación de la voz y así facilitar la interpretación, así como el filtro de ruido mismo que posee la librería de Speech Recognition con el cual se guarda una información en forma de audio previo para cuando se ejecute el reconocimiento tener establecido un posible ruido de fondo.

En la mayoría de los casos de prueba se logró un correcto funcionamiento llegando a ser la frase adquirida correspondiente a la clase deseada y así ejecutar la función esperada ignorando cualquier problema que llegó ser la conexión a internet.

Para una mejor interacción entre el usuario y el asistente se le otorgó una voz con la cual se simulaba las reacciones que podría llevar a cabo mediante las funciones determinadas.



*Imagen2. Montaje final.*

#### V.CONCLUSIONES

- En un entorno en el cual se presente alto ruido, se debe aplicar un filtro que permita identificar mejor las palabras, de no ser así el sistema no reconocerá correctamente las palabras.
- Para la identificación de la palabra se realiza la conversión de cada palabra a su correspondiente en ASCII, debido a que la información que entrega el SpeechRecognition es una cadena de datos que se debe separar para poder clasificarla con el Random forest.
- Al eliminar los conectores de las cadenas de texto que entrega SpeechRecognition incrementa la probabilidad de acierto a las clases.
- Debido a que la API que utiliza el SpeechRecognition requiere conexión a internet, es necesario asegurar una buena conexión de red.
- El sistema tiene la posibilidad de migrar a diferentes áreas, creando nuevas clases que representan funciones.

#### REFERENCIAS

- [1] L. Llaurado, «Blog Shneider Electronic,» Shneider Electronic, 24 01 2018. [En línea]. Available: <https://blogespanol.se.com/gestion-de-procesos-y-maquinas/2018/01/24/la-inteligencia-artificial-llega-las-fabricas/>. [Último acceso: 19 05 13].
- [2] S. Raschka, «Python Machine Learning,» de Python Machine Learning, Birmingham, Packt Publishing Ltd, 2015, pp. 17-42.
- [4] G. B. & V. Kepuska, «ResearchGate,» 12 05 2018. [En línea]. Available: [https://www.researchgate.net/publication/325256842\\_Next-Generation\\_of\\_Virtual\\_Personal\\_Assistants\\_by\\_using\\_Python\\_Node\\_Red\\_and\\_IBM\\_Watson](https://www.researchgate.net/publication/325256842_Next-Generation_of_Virtual_Personal_Assistants_by_using_Python_Node_Red_and_IBM_Watson). [Último acceso: 19 05 12].
- [5] A. C. & S. Guido, «Introduction to Machine Learning with Python,» de Introduction to Machine Learning with Python, Sebastopol, O'Reilly Media, 2016, pp. 27-126.