

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TELECOMUNICACIONES

Tema:

SISTEMA AUTOMÁTICO DE DETECCIÓN DE EXTORSIONES MEDIANTE EL ANÁLISIS DE SEÑALES DE LA VOZ

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Telecomunicaciones

ÁREA: Comunicaciones

LÍNEA DE INVESTIGACIÓN: Tecnología de la información y Sistemas de control

AUTOR: Carlos Jair Tinta Capuz

TUTOR: Ing. Ana Pamela Castro Martin, MSc.

Ambato - Ecuador

agosto – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA AUTOMÁTICO DE DETECCIÓN DE EXTORSIONES MEDIANTE EL ANÁLISIS DE SEÑALES DE LA VOZ, desarrollado bajo la modalidad Proyecto de Investigación por el señor Carlos Jair Tinta Capuz, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2024.



Firmado electrónicamente por:
ANA PAMELA CASTRO MARTIN

Ing. Ana Pamela Castro Martin, MSc.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA AUTOMÁTICO DE DETECCIÓN DE EXTORSIONES MEDIANTE EL ANÁLISIS DE SEÑALES DE LA VOZ es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2024.



Carlos Jair Tinta Capuz

C.C. 180515817-5

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2024.



Carlos Jair Tinta Capuz

C.C. 180515817-5

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Carlos Jair Tinta, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA AUTOMÁTICO DE DETECCIÓN DE EXTORSIONES MEDIANTE EL ANÁLISIS DE SEÑALES DE LA VOZ, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2024.

Firmado electrónicamente por:
ELSA PILAR URRUTIA URRUTIA

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
CARLOS DIEGO GORDON GALLEGOS

Dr. Carlos Gordón Gallegos
PROFESOR CALIFICADOR

Firmado electrónicamente por:
SANTIAGO MAURICIO ALTAMIRANO MELENDEZ

Ing. Santiago Altamirano Meléndez, Mg
PROFESOR CALIFICADOR

DEDICATORIA

Quiero dedicar esta tesis, en primer lugar, a Dios, por su guía constante y por ayudarme a superar los desafíos que he enfrentado. A mis padres, quienes han sido mi pilar fundamental, y a mi hermana, por su apoyo incondicional a lo largo de este camino.

Agradezco profundamente a toda mi familia por su constante aliento y por creer en mí en cada paso que he dado hacia este logro. A mis amigos, quienes han compartido conmigo sonrisas y logros, y han sido una fuente invaluable de apoyo y motivación.

Dedico este trabajo a todas las personas que me han impulsado y alentado a alcanzar este punto de mi vida académica. También agradezco a aquellos que me han enseñado valiosas lecciones sobre la vida y me han ayudado a crecer como persona.

Finalmente, dedico esta tesis a la Universidad Técnica de Ambato, en especial a la Facultad "FISEI", por brindarme las herramientas y el conocimiento necesario para cumplir mi sueño de convertirme en Ingeniero. Su apoyo y recursos han sido fundamentales en este camino hacia la culminación de mis estudios.

AGRADECIMIENTO

Quiero agradecer a Dios por haberme dado la vida y haberme guiado a lo largo de este camino. Su presencia ha sido una fuente constante de fortaleza y esperanza en los momentos más desafiantes de mi vida académica.

A mis padres, Carlos Tinta y Consuelo Capuz, les agradezco profundamente por su amor incondicional y su apoyo inquebrantable. Sin su sacrificio y dedicación, no habría sido posible llegar hasta aquí. A mi hermana, Alison Tinta, gracias por estar siempre a mi lado, brindándome su comprensión y motivación en cada etapa de este viaje.

Deseo también extender mi gratitud a mis amigos y a todas las personas que han estado presentes en este camino universitario. Su compañía, consejos y apoyo han sido esenciales para mantenerme enfocado y motivado durante estos años.

Finalmente, quiero agradecer de manera especial a mi tutora, Ingeniera Pamela Castro, por su invaluable ayuda y orientación para culminar mi trabajo de tesis de la mejor manera posible. Su dedicación y conocimiento han sido fundamentales para el éxito de este proyecto.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA ..... **i**

APROBACIÓN DEL TUTOR ..... **ii**

AUTORÍA ..... **iii**

DERECHOS DE AUTOR ..... **iv**

APROBACIÓN DEL TRIBUNAL DE GRADO ..... **v**

DEDICATORIA ..... **vi**

AGRADECIMIENTO ..... **vii**

ÍNDICE GENERAL DE CONTENIDOS ..... **viii**

ÍNDICE DE TABLAS ..... **xii**

ÍNDICE DE FIGURAS ..... **xiii**

ÍNDICE DE ANEXOS ..... **xvii**

RESUMEN EJECUTIVO ..... **xviii**

ABSTRACT ..... **xix**

CAPÍTULO I. MARCO TEÓRICO **1**

1.1 Tema de investigación 1

1.1.1 Planteamiento del problema 1

1.2 Antecedentes investigativos 4

1.3 Fundamentación teórica 7

1.3.1 Detección de fraude y corrupción 7

1.3.2 Definición de extorsión 8

1.3.3 Frases comunes de extorsionadores 10

1.3.4 Análisis de señales de voz 12

1.3.5 Extracción de características en el reconocimiento de la voz 13

1.3.6 MFCC (Coeficientes Cepstrales en las Frecuencias de Mel) 14

1.3.7 Reconocimiento de voz: extracción de características MFCC y PLP 17

1.3.8 Ventanas 21

1.3.9 Reconocimiento de la voz 25

1.3.10 Reconocimiento del habla 27

1.3.11 Reconocimiento de la voz en el software Python 29

1.3.12 Procesamiento de lenguaje natural (NLP) 33

1.3.13 Detección de extorsiones con inteligencia artificial 33

1.3.14 Aprendizaje automático (Machine Learning) 34

1.3.15 Algoritmos de inteligencia artificial 38

1.3.16 Base de datos del sistema para entrenamiento de IA 45

1.4 Objetivos 49

1.4.1 Objetivo general 49

1.4.2 Objetivos específicos 49

CAPÍTULO II. METODOLOGÍA 50

2.1 Materiales 50

2.2 Métodos 51

2.2.1 Modalidad de la investigación 51

2.2.2 Recolección de información 52

2.2.3 Procesamiento y análisis de datos 52

CAPÍTULO III. RESULTADOS Y DISCUSIÓN 53

3.1 Desarrollo de la propuesta 53

3.1.1 Selección de Hardware 55

3.1.2 Selección de Software 57

3.2 Implementacion del sistema 58

3.2.1 Captura de la señal de la voz 58

3.2.2 Reconocimiento y extracción de características de la voz 59

3.2.3 Conversión de la voz a lenguaje natural 59

3.2.4 Dataset 60

3.2.5 Clasificación y evaluación mediante inteligencia artificial 63

3.2.6 Alerta de posible extorsión 64

3.3 Desarrollo del sistema de detección de extorsiones 64

3.3.1 Agregar nuevas frases al dataset para el entrenamiento 77

3.3.2 Interfaz detector de extorsión utilizando reconocimiento de voz 80

3.4 Pruebas de funcionamiento 91

3.4.1 Resultados del modelo Random Forest Classifier 91

3.4.2 Funcionamiento de la Interfaz Gráfica 99

3.4.3 Evaluación del sistema de detección de extorsiones 106

3.4.4 Presupuesto 109

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 111

4.1 Conclusiones 111

4.2 Recomendaciones..... 112

REFERENCIAS BIBLIOGRÁFICAS..... 115

ANEXOS..... 120

ÍNDICE DE TABLAS

Tabla 1. Etapas de funcionamiento de la librería SpeechRecognition 31

Tabla 2. Características de las señales de la voz en la librería SpeechRecognition ... 32

Tabla 3. Tipos de algoritmos de Machine learning 35

Tabla 4. Componentes del Machine Learning 37

Tabla 5. Etapas de Machine Learning 37

Tabla 6. Comparación entre los algoritmos de inteligencia artificial 44

Tabla 7. Características del hardware seleccionado 50

Tabla 8. Características del software seleccionado 51

Tabla 9. Comparación de procesadores 56

Tabla 10. Cuadro comparativa entre Matlab y Python 57

Tabla 11. Resultados del funcionamiento del sistema 105

Tabla 12. Matriz de confusión 106

Tabla 13. Resultados Matriz de confusión 108

ÍNDICE DE FIGURAS

Figura 1. Ventana deslizante de 25 ms de ancho para extraer funciones de audio 17

Figura 2. Ventana de 25 ms 18

Figura 3. Proceso de extracción de características utilizando MFCC[18]..... 19

Figura 4.Conversión A/D..... 20

Figura 5. Preénfasis 20

Figura 6. Audio en segmentos deslizantes..... 21

Figura 7. División del audio adecuadamente..... 21

Figura 8. Entrada de marco cortado y audio original..... 22

Figura 9. Ventanas de Hamming..... 22

Figura 10. Ventanas de Hamming y Hanning 23

Figura 11. Estructura espectral del habla..... 24

Figura 12. Proceso del reconocimiento de la voz..... 25

Figura 13. Reconocimiento de la voz en Python..... 30

Figura 14. Funcionamiento del reconocimiento de la voz..... 31

Figura 15. Estructura de una red LSTM 40

Figura 16. Estructura del algoritmo Random Forest..... 41

Figura 17. Ejemplo de RandomForestClassifier 44

Figura 18. Esquema general del sistema..... 53

Figura 19. Proceso de detección de extorsiones..... 55

Figura 20. Desarrollo del dataset..... 61

Figura 21. Estructura del dataset llamado “dataset_frases” 62

Figura 22. Desarrollo de la programación del sistema 65

Figura 23.Importación de la librería pandas 66

Figura 24. Lectura de los datos de Excel 66

Figura 25. Filtrado de frases por etiquetas 67

Figura 26. Creación de lista de etiquetas 67

Figura 27. Combinación de listas 68

Figura 28. Creación de un DataFrame 68

Figura 29. Guardado del dataset en un archivo csv 68

Figura 30. Importación de bibliotecas 70

Figura 31. Carga del dataset 70

Figura 32. Vectorización de palabras 71

Figura 33. Separación de etiquetas 71

Figura 34. División de los datos de entrenamiento y prueba 72

Figura 35. Diagrama de flujo de la inteligencia artificial 73

Figura 36. Entrenamiento del modelo 74

Figura 37. Predicción del modelo 74

Figura 38. Cálculo de la precisión 74

Figura 39. Guarda el modelo 75

Figura 40. Guarda el vectorizador 76

Figura 41. Frases para prueba del modelo 76

Figura 42. Predicciones del modelo 77

Figura 43. Impresión de resultados 77

Figura 44. Creación y guardado del dataset..... 78

Figura 45. Estructura del dataset 78

Figura 46. Resultado del entrenamiento del sistema con frases de prueba..... 80

Figura 47. Importación de librerías para el sistema 81

Figura 48. Inicializa pyttsx3 y carga el modelo y el vectorizador 82

Figura 49. Función login..... 82

Figura 50. Interfaz Tkinter..... 83

Figura 51. Función reconocimiento de voz parte 1 85

Figura 52. Función reconocimiento de voz parte 2 86

Figura 53. Función reproducir texto..... 86

Figura 54. Carga la imagen y la dibuja en canvas..... 87

Figura 55. Configuración ventana principal 87

Figura 56. Configuración del login_frame 88

Figura 57. Configuración del form_frame..... 88

Figura 58. Configuración de ingreso al sistema 89

Figura 59. Configuración del botón de inicio..... 90

Figura 60. Configuración de Widgets en main_frame..... 91

Figura 61. Matriz de confusión del algoritmo de IA..... 94

Figura 62. Datos obtenidos durante el entrenamiento..... 94

Figura 63. Frases utilizadas en el entrenamiento del sistema 95

Figura 64. Características del modelo de IA 96

Figura 65. Visualización de un árbol de decisión 97

Figura 66. Proceso de toma de decisiones del algoritmo RandomForest 98

Figura 67. Ejemplo de toma de decisiones del algoritmo RandomForest 99

Figura 68. Interfaz del sistema 100

Figura 69. Interfaz del reconocimiento de la voz 101

Figura 70. Detección primera frase de lenguaje natural 103

Figura 71. Resultado primera frase natural 103

Figura 72. Detección primera frase de lenguaje extorsivo 104

Figura 73. Resultado primera frase extorsiva 104

ÍNDICE DE ANEXOS

Anexo A. Validación de datos del Policía y del sistema 120

Anexo B. Datos de la matriz de confusión 122

Anexo C. Base de datos csv del sistema 123

Anexo D. Archivos del sistema 124

Anexo E. Programación del sistema 125

Anexo F. Pruebas del sistema 128

RESUMEN EJECUTIVO

Este proyecto presenta la implementación de un sistema automático para detectar extorsiones a través del análisis de señales de voz. Este sistema se ha diseñado para identificar el lenguaje extorsivo, siendo una herramienta esencial para la seguridad y prevención de delitos.

La arquitectura del sistema se divide en tres capas principales. En la primera, el reconocimiento de voz y extracción de características se hace internamente con librerías específicas. Frases extorsivas y no extorsivas se clasifican para entrenar el modelo de detección y garantizar precisión en distinguir ambos tipos de lenguaje. Para el entrenamiento incorpora un conjunto de frases clasificadas como extorsivas y no extorsivas. Este conjunto de datos se utiliza para entrenar el modelo de detección, garantizando que el sistema pueda distinguir entre ambos tipos de lenguaje.

La capa intermedia utiliza técnicas avanzadas de procesamiento de lenguaje natural (NLP) y aprendizaje automático con el modelo RandomForestClassifier, seleccionado por su fiabilidad y eficiencia. Este sistema procesa y clasifica las frases según su contenido, garantizando una detección precisa y robusta de lenguaje extorsivo.

La capa de presentación incluye una interfaz gráfica de usuario (GUI) desarrollada con Tkinter. Esta interfaz permite a los usuarios interactuar con el sistema de manera intuitiva, activando el micrófono para grabar su voz y recibir una evaluación inmediata sobre la presencia de lenguaje extorsivo. La GUI mejora la accesibilidad y usabilidad del sistema.

Este prototipo supone un avance significativo en la detección de extorsiones con una precisión del 92% según las pruebas realizadas al sistema, proporcionando una solución eficiente. Se anticipa que esta herramienta satisface la creciente demanda de sistemas de seguridad inteligentes, ayudando a las organizaciones a mantenerse a la vanguardia en la lucha contra el crimen.

Palabras clave: Detección de extorsiones, análisis de voz, machine learning, RandomForestClassifier.

ABSTRACT

This project presents the implementation of an automatic system to detect extortion through the analysis of voice signals. This system has been designed to identify extortive language, being an essential tool for security and crime prevention.

The architecture of the system is divided into three main layers. In the first, speech recognition and feature extraction is done internally with specific libraries. Extortionist and non-extortionist phrases are classified to train the detection model and ensure accuracy in distinguishing both types of language. A set of sentences classified as extortionate and non-extortionate is incorporated for training. This dataset is used to train the detection model, ensuring that the system can distinguish between the two types of language.

The intermediate layer uses advanced natural language processing (NLP) and machine learning techniques with the RandomForestClassifier model, selected for its reliability and efficiency. This system processes and classifies sentences according to their content, ensuring accurate and robust detection of extortionate language.

The presentation layer includes a graphical user interface (GUI) developed with Tkinter. This interface allows users to interact with the system intuitively by activating the microphone to record their voice and receive an immediate assessment of the presence of extortionate language. The GUI improves the accessibility and usability of the system.

This prototype represents a significant advance in extortion detection with 92% accuracy according to system tests, providing an efficient solution. It is anticipated that this tool will meet the growing demand for intelligent security systems, helping organizations stay ahead in the fight against crime.

Keywords: Extortion detection, voice analysis, machine learning, RandomForestClassifier.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA AUTOMÁTICO DE DETECCIÓN DE EXTORSIONES MEDIANTE EL ANÁLISIS DE SEÑALES DE LA VOZ

1.1.1 Planteamiento del problema

En la era digital actual los engaños, estafas y amenazas que se llevan a cabo mediante llamadas, conversaciones y extorsiones. Estas prácticas delictivas han experimentado un preocupante aumento, generando un impacto significativo en la seguridad y el bienestar de las personas. Con el advenimiento de las tecnologías de la información y la comunicación, los delincuentes han encontrado nuevas y sofisticadas formas de aprovecharse de la confianza de los individuos, explotando sus vulnerabilidades y obteniendo beneficios ilegales. Ya sea a través de llamadas telefónicas fraudulentas, conversaciones manipuladas o amenazas virtuales, estas actividades delictivas han sembrado desconfianza en los sistemas de comunicación y han contribuido a un aumento en la delincuencia en general. Es esencial entender la naturaleza y las implicaciones de estas extorciones, así como buscar soluciones efectivas para salvaguardar a la sociedad de estas amenazas crecientes. Según el diario El Universo el avance de la era digital ha llevado consigo un incremento en las denuncias relacionadas con el fraude financiero, en donde los delincuentes aprovechan las redes sociales y la web para reclutar víctimas y estafarlas [1].

En el panorama actual, la delincuencia a través de llamadas telefónicas ha generado un clima de inseguridad que siembra temor en la sociedad a nivel mundial. En los últimos años, hemos sido testigos de un preocupante aumento en las extorsiones llevadas a cabo por este medio de comunicación. Las llamadas fraudulentas y manipuladoras no solo afectan la tranquilidad emocional de las personas, sino que también tienen un impacto significativo en la economía de los países. En las naciones con altos índices de extorsiones telefónicas, la confianza de los ciudadanos se ve afectada, lo que resulta en una disminución del flujo turístico y en un debilitamiento

de la economía local. Según SeguriLatam una de las modalidades más recientes de extorsión telefónica se conoce como "La patrona". En este método, los delincuentes contactan a un hogar o establecimiento comercial y, a través de engaños, buscan obtener información de los propietarios [2].

La extorsión es una problemática común en muchos países subdesarrollados, generando un ambiente de inseguridad que afecta a la población. En este contexto, las llamadas telefónicas se convierten en una herramienta frecuentemente utilizada por los delincuentes para cometer estos actos. Mediante amenazas de muerte o daño, exigen dinero a cambio de evitar consecuencias graves. Además, los estafadores engañan a las víctimas ofreciendo premios falsos u oportunidades engañosas. El robo de datos personales se ha vuelto común a través de conversaciones telefónicas manipuladas. En este sentido, en el presente trabajo se propone investigar los distintos tipos de estafas y engaños telefónicos, así como analizar cómo los ladrones modifican su voz y cómo se pueden utilizar el timbre de la voz y el nerviosismo como indicadores para detectar engaños.

En la provincia de Tungurahua, en la ciudad de Ambato Ecuador, no está exenta de los delitos de extorsión a través de llamadas telefónicas por parte de aparentes grupos delincuenciales. En el año 2022, de las 416 denuncias de estafas presentadas ante la Fiscalía, 70 están relacionadas con llamadas extorsivas. El modus operandi es similar en la mayoría de los casos: los delincuentes llaman a las víctimas y les informan que ciertas personas los han contratado para atentar contra ellos y su familia, proporcionando datos precisos sobre hijos, esposa, padres y negocios. Según Édison Medrano, jefe operativo de la Policía Judicial de Tungurahua, estos delincuentes obtienen información de las redes sociales de las víctimas y amenazan con revelar quién los ha mandado a victimizar si no se les entrega dinero. La mayoría de las llamadas de amenazas van dirigidas a personas con negocios que publican su número de contacto en las redes sociales. Se ha descubierto que la mayoría de estas llamadas se originan desde centros penitenciarios, donde los delincuentes tienen tiempo para investigar, verificar y aprovechar la vulnerabilidad de sus víctimas[1].

En la actualidad, los delincuentes utilizan diversas tácticas para aprovecharse de la confianza y vulnerabilidad de sus víctimas, siendo una de las estrategias más comunes

la modificación de su voz durante las llamadas. Esto dificulta aún más la detección de estos engaños, ya que se aprovechan de la falta de herramientas precisas para identificar las señales de peligro. Sin embargo, se ha observado que ciertos elementos como el timbre de la voz y el nerviosismo pueden brindar pistas importantes sobre la autenticidad de la persona al otro lado de la línea. Por tanto, se hace necesario explorar y comprender en profundidad estos factores para desarrollar sistemas de detección de mentiras efectivos, capaces de proteger a los usuarios y prevenir posibles estafas telefónicas. En este sentido, en el presente trabajo se propone investigar los distintos tipos de estafas y engaños telefónicos, así como analizar cómo los ladrones modifican su voz y cómo se pueden utilizar el timbre de la voz y el nerviosismo como indicadores para detectar engaños. Además, se examinarán los sistemas actuales de detección de mentiras y su efectividad en este contexto específico[1].

El diseño de un dispositivo automático de detección de extorsiones a través de las señales de la voz se ha vuelto cada vez más crucial debido al impacto significativo que las extorsiones por llamadas telefónicas tienen en la sociedad. Estas prácticas delictivas generan un profundo temor en las personas que se convierten en víctimas, por lo tanto, es fundamental abordar este problema y fortalecer las medidas de seguridad para garantizar la protección de las personas y promover un entorno seguro y confiable para su desarrollo humano.

La creciente incidencia de estafas, amenazas y extorsiones mediante llamadas telefónicas ha impulsado la necesidad de contar con herramientas efectivas para su detección y prevención. Gracias a algoritmos que utilizan técnicas de aprendizaje automático y procesamiento de lenguaje natural, es posible analizar y comprender el contenido de estas llamadas, identificando patrones y señales que indiquen posibles engaños o amenazas. La aplicación de algoritmos de inteligencia en la detección de estos delitos tiene un efecto directo en la seguridad de las personas, ya que permite identificar de manera temprana las llamadas fraudulentas y proporcionar una respuesta rápida por parte de las autoridades competentes. Con este dispositivo automático, se busca brindar una solución eficiente y confiable que contribuya a la lucha contra las extorsiones telefónicas y brinde tranquilidad a la sociedad en general.

1.2 Antecedentes investigativos

A través de la investigación realizada sobre los sistemas detectores de engaños en los repositorios de Artículos Académicos, así como en diversos sitios de investigación se obtiene los siguientes antecedentes.

En el año 2018, Ruiz Hernán, realizaron la publicación "Aportaciones de un modelo predictivo para identificar el perfil de riesgo de un defraudador interno en una entidad financiera colombiana", realizaron una investigación sobre las empresas de servicios financieros son fundamentales, y su continuo crecimiento las ha vuelto más vulnerables al fraude debido a la falta de medidas preventivas y controles que mitiguen las amenazas asociadas a la innovación tecnológica, nuevos productos y patrones de consumo. Para abordar esta problemática, se identificó la necesidad de contribuir a la elaboración de un perfil de riesgo del defraudador interno en una entidad financiera colombiana. Utilizando un modelo estadístico predictivo, que analiza la información de empleados implicados en actividades fraudulentas y una revisión de la literatura relevante, se definió un perfil parcial (considerando solo algunas variables). A través de la construcción de un modelo de Random Forest, cuyo objetivo es identificar empleados con altas probabilidades de cometer actos fraudulentos, se encontró que el perfil del defraudador incluye trabajadores de la red de sucursales con transacciones atípicas y relaciones desproporcionadas entre salarios y descuentos aplicados. Los empleados en cargos operativos, que representan el 44.5% de la empresa, son los más propensos al fraude interno, y el 66.4% de los posibles defraudadores son mujeres. Además, las regiones con mayor alerta de fraude son la región centro (43.6%), seguida por Bogotá y Sabana (28.7%). Esta investigación demuestra que es posible identificar y prevenir el fraude mediante la evaluación de características específicas de los empleados, utilizando un modelo estadístico predictivo basado en Random Forest[3].

En el año 2018, en la plataforma IEEE XPLORE, Savita Sondhi, Ritu Vijav, Munna Khan & Ashok K Salhan, se presenta la publicación "Voice analysis for detection of deception. En este trabajo se estudia como la voz humana puede señalar el estado psicológico de un individuo. Ya sea emocional, física, intencional o inconsciente, la voz revela el estado interno del hablante. Por tanto, el objetivo principal del estudio fue identificar qué parámetro de la voz podría utilizarse como un indicador fiable del

estrés debido al engaño. Para analizar el estrés natural de la vida real, se grabó la voz de los sospechosos de delitos durante los interrogatorios en curso en la estación de policía. Para el análisis se utilizó el software PRAAT. Los resultados muestran un aumento significativo en la frecuencia fundamental entre (F 0) y las frecuencias de formato F1, F2. Se observó una ligera reducción en el jitter, pero el brillo no se correlacionó en absoluto con el estrés engañoso. Los contornos de tono mostraron un aumento significativo en el tono de los sospechosos de delitos al responder preguntas directas específicas. Por lo tanto, se concluye que la F0 media y la frecuencia de las fórmulas F1, F2 están fuertemente correlacionadas con el estrés y como resultado se encontró una fuerte correlación entre la F0 media y las frecuencias de formato F1 y F2 con el estrés engañoso y estos proporcionan evidencia de que ciertos parámetros de la voz pueden ser utilizados como indicadores confiables para detectar el engaño y el estrés asociado en situaciones reales[4].

En el 2021, en la Universidad de las Fuerzas Armadas, Chango Salas y Jorge Alejandro en la investigación" Desarrollo de un algoritmo detector de engaños mediante la utilización del método Deep Learning de inteligencia artificial", realizaron un análisis del estrés en la voz mediante la extracción de características utilizando Matlab. El objetivo era determinar el engaño al comparar las características del habla "normal" con las características bajo "estrés". Para llevar a cabo la investigación, se utilizó una base de datos pública llamada "RAVDESS" que constaba de 94 señales de audio grabadas tanto por hombres como por mujeres. Se implementó un algoritmo de entrenamiento utilizando una red neuronal convolucional de Deep Learning para realizar el reconocimiento automático de verdades y engaños. Los resultados mostraron que las características de la voz proporcionan información relevante para evaluar y clasificar el engaño frente a la verdad. Esta investigación demostró que es posible analizar y detectar el engaño mediante la evaluación de las características de la voz, utilizando un algoritmo de entrenamiento basado en una red neuronal convolucional de Deep Learning. Se demostró que las características de la voz proporcionan información relevante para evaluar y clasificar el engaño frente a la verdad. El algoritmo de entrenamiento basado en la red neuronal convolucional de Deep Learning fue capaz de analizar y detectar el engaño de manera efectiva[5].

En el año del 2022, en la Universidad Politécnica Salesiana, Zumba Edison & Christian Salamea, en la investigación “Desarrollo de un sistema prototipo de generación de rostros a partir de señales de voz utilizando redes generativas adversarias”, crearon un prototipo de sistema para generar rostros utilizando redes generativas adversarias (GANs). El desarrollo del sistema consta de dos partes principales. En la primera parte de la investigación, se construye una base de datos específica para entrenar las redes y en la segunda parte, se desarrollan dos redes que reciben ruido como parámetro de entrada, así como audio de voz. Además, se diseña un autoencoder cuyos resultados se comparan para evaluar la eficiencia de la red prototipo desarrollada. Este prototipo de sistema se centra en generar rostros utilizando redes generativas adversarias. Se divide en dos etapas, la construcción de la base de datos y el desarrollo de las redes, donde se incorpora la entrada de ruido y audio de voz. Además, se utiliza un autoencoder para realizar comparaciones y evaluar la eficiencia de la red prototipo se obtuvo un sistema que fusiona la voz y la imagen de manera efectiva, ofreciendo nuevas posibilidades en campos como la animación, el entretenimiento y la creación de contenido visualmente personalizado [6].

En el año 2022, en la Universidad Mayor de San Andrés, Calliconde Carrasco & Carmela Sandra en la investigación “ Detector de mentiras mediante el análisis de estrés de voz usando redes neuronales recurrentes”, realizan un detector de mentiras que analiza el estrés de la voz mediante el uso de redes neuronales recurrentes. Sin embargo, no existe un conjunto de datos con grabaciones de voz de personas bajo estrés. Para abordar esta limitación, se crea un conjunto de datos utilizando grabaciones de voz de los conjuntos SAVEE y CREMA-D, seleccionando las emociones de enojo y miedo que indican la presencia de estrés. Estas grabaciones se procesan utilizando la biblioteca librosa, extrayendo 40 coeficientes cepstrales de frecuencia de Mel, lo cual genera un vector para cada grabación de voz. Luego, estos vectores se utilizan para entrenar y validar un modelo. Se desarrolla una aplicación que incluye el modelo para realizar pruebas. La aplicación examina la grabación de voz en un teléfono inteligente y realiza una predicción correspondiente. El proyecto sigue la metodología esencial en la ciencia de datos y consta de 10 etapas. Al finalizar estas etapas, se logra una precisión del 80% en la detección de estrés[7].

1.3 Fundamentación teórica

El presente estudio se enfoca en abordar los fundamentos y conceptos clave relacionados con las extorsiones telefónicas, el procesamiento de señales de voz, y el uso de algoritmos de aprendizaje automático. Se llevará a cabo una exploración de investigaciones previas en el campo de la detección de engaños y emociones a través de la voz, así como en el diseño de sistemas automáticos para identificar patrones y señales engañosas. El marco teórico proporcionará la base necesaria para el desarrollo de un sistema eficiente y confiable que contribuya a la lucha contra las extorsiones telefónicas, al tiempo que promueva un entorno seguro y confiable para las personas.

1.3.1 Detección de fraude y corrupción

En la lucha contra el fraude y la corrupción, los profesionales del derecho están adoptando cada vez más tecnologías innovadoras como la inteligencia artificial (IA) para mejorar sus investigaciones y casos. La IA proporciona capacidades avanzadas de análisis y predicción, permitiendo examinar grandes volúmenes de datos financieros, transacciones comerciales y comunicaciones electrónicas para identificar patrones sospechosos y comportamientos anómalos que podrían indicar conductas fraudulentas o corruptas. La IA ayuda a prevenir irregularidades al alertar sobre posibles problemas antes de que se agraven, mitigando así pérdidas financieras y protegiendo la reputación de las organizaciones[8].

A pesar de los beneficios que ofrece, el uso de la IA en la detección de fraude y corrupción plantea desafíos significativos en áreas como la privacidad de los datos personales y la transparencia en la toma de decisiones basadas en algoritmos. Los profesionales del derecho juegan un papel crucial en asegurar que la implementación de la IA cumpla con estándares éticos y legales estrictos, protegiendo los derechos de las personas afectadas y garantizando una aplicación justa de la ley.

En un entorno en constante evolución tecnológica, la colaboración efectiva entre la IA y la experiencia legal humana sigue siendo esencial. La integración adecuada de la IA con el conocimiento legal contribuirá significativamente a fortalecer las capacidades

para detectar y abordar eficazmente el fraude y la corrupción, fomentando así la integridad y la equidad en nuestras instituciones y sociedades[8].

1.3.2 Definición de extorsión

La extorsión es un delito en el que el delincuente utiliza presión, violencia o intimidación para obligar a alguien a actuar en contra de su voluntad, con el objetivo de obtener un beneficio económico o material. Este delito, común pero severamente castigado por la ley, afecta múltiples aspectos de la vida de la víctima, incluyendo su propiedad, integridad física y moral, y libertad personal. Se considera un delito pluriofensivo y de apoderamiento, ya que el extorsionista busca aprovecharse de los bienes de la persona atacada[9].

Los extorsionistas suelen pertenecer a bandas delictivas o ser conocidos de la víctima. Utilizan su conocimiento sobre la víctima para ejercer la mayor presión posible. Ejemplos de extorsión incluyen exigir pagos a comerciantes para evitar robos o negociar la liberación de un rehén a cambio de dinero. Otro tipo es el chantaje por infidelidad, donde se amenaza con divulgar información comprometedora. En todos los casos, el objetivo es obtener una ganancia mediante la explotación de otros. El término extorsión proviene del latín "extorsión" y tiene sinónimos como chantaje, despojo, estafa, usurpación, y robo[9].

El delito de extorsión implica forzar a una persona, con fines de lucro y mediante violencia o intimidación, a realizar u omitir un acto o negocio jurídico en detrimento de su patrimonio o el de un tercero. La Real Academia Española (RAE) define la extorsión como la presión ejercida mediante amenazas para obligar a alguien a actuar de cierta manera y obtener así un beneficio, generalmente económico. Aunque un sinónimo de extorsión podría ser la amenaza condicional, la jurisprudencia establece que no es correcto usar los términos coacción o amenaza como equivalentes a extorsión[10].

a. Tipos de extorsiones

La extorsión es un delito en el cual el delincuente, mediante violencia o intimidación, obliga a la víctima a realizar u omitir acciones con fines de lucro. Existen dos tipos de extorsión: directa e indirecta.

La extorsión directa ocurre cuando el delincuente se presenta físicamente en el establecimiento o domicilio de la víctima, amenazando al propietario o al personal y exigiendo pagos a cambio de no causar daño, a menudo identificándose como miembro de una organización delictiva. Por otro lado, la extorsión indirecta, que es la modalidad más común desde finales del año 2000, se realiza a través de llamadas telefónicas o mensajes de texto. En este caso, los delincuentes contactan a sus víctimas planteando situaciones de riesgo para la persona o su familia y, utilizando directorios telefónicos y datos personales obtenidos de redes sociales, exigen pagos mediante depósitos en tiendas departamentales, comercios o transferencias bancarias, aprovechándose de la violencia psicológica y la buena fe de las víctimas[11].

b. Tipos de extorsiones en Ecuador

El teniente coronel Wilson Zapata, jefe de la unidad antisecuestro y extorsión de la Policía Nacional, explica que en Ecuador existen cuatro tipos de extorsiones, y que el término "vacuna" se utiliza coloquialmente para referirse a todas ellas, aunque técnicamente solo una es correcta[12].

Los tipos de extorsión son[12]:

- **Extorsión virtual:** Realizada mediante llamadas telefónicas o mensajes, donde los delincuentes usan información obtenida de redes sociales para pedir pagos mediante engaños o intimidación. Este tipo representa el 44% de los casos del primer trimestre de 2023. Los delincuentes buscan obtener el dinero rápidamente, sin que la víctima cuelgue, a menudo afirmando ser sicarios contratados para matar a la víctima, pidiendo dinero para evitar el crimen.

- **Extorsión común:** En este caso, los delincuentes obtienen información detallada sobre la víctima, como la ubicación, datos familiares o rutinas diarias, para atemorizarla y obtener dinero. Este tipo es uno de los más frecuentes en 2023, aunque no está relacionado con secuestros u otros delitos.

- **Extorsión vacuna:** Aquí, los delincuentes solicitan dinero directamente a comercios o fincas, a menudo a través de panfletos, a cambio de "seguridad". Este tipo ha mostrado una tendencia al alza, representando el 22% de los casos de extorsión en el primer trimestre de 2023.

- **Extorsiones sexuales:** Realizadas a través de medios digitales, donde se amenaza a la víctima con publicar imágenes o videos comprometedores a menos que pague una suma de dinero. Este tipo representa el 0.2% de los casos de extorsión, con material obtenido por hackeo o videoconferencias.

1.3.3 Frases comunes de extorsionadores

Las tácticas comunes de extorsión generalmente incluyen amenazas explícitas o implícitas destinadas a intimidar a la persona afectada y forzarla a cumplir con las exigencias del extorsionador. Estas amenazas pueden abarcar desde causar daño físico a la víctima o a sus familiares cercanos hasta divulgar información comprometedora o dañar su reputación[13].

Es esencial tener en cuenta que los perpetradores de extorsión suelen ser hábiles en la manipulación y emplean diversas estrategias para infundir temor en sus víctimas. Ante una situación de extorsión, se aconseja mantener la serenidad y resistir la presión del extorsionador. La mejor medida es reportar el incidente a las autoridades pertinentes y buscar asesoramiento profesional[13].

A comienzos del año 2021, las extorsiones telefónicas atribuidas a presuntos miembros de las Fuerzas Armadas Revolucionarias (FARC) estaban dirigidas principalmente a médicos y odontólogos en Quito. Sin embargo, desde entonces, estas llamadas han experimentado un notable aumento a nivel nacional, especialmente en la Zona 8, lo que representa una preocupación creciente. Recientemente, la Unidad Antisecuestro y Extorsión de la Policía (Unase) informó que el ECU911 de Samborondón ha recibido entre tres y cinco llamadas amenazantes diarias en las últimas semanas. Los agentes de seguridad han alertado sobre las tácticas utilizadas por los extorsionadores para obtener información y cómo evitar convertirse en víctima de este delito, que también está siendo reportado en Colombia y Venezuela. Los sospechosos suelen utilizar

llamadas de WhatsApp desde números nacionales o colombianos con el prefijo +57, posiblemente operando desde el extranjero, evitando el uso de videollamadas o llamadas convencionales según detallan los investigadores[13].

Las extorsiones mediante mensajes, llamadas telefónicas y correos electrónicos están en aumento y afectan a miles de personas diariamente. A continuación, se presentan algunas frases comúnmente utilizadas por extorsionadores en Ecuador, junto con consejos para identificarlas y protegerse[14]:

"Tenemos a tu familiar y pedimos X cantidad de dinero para liberarlo": En esta estafa, la víctima recibe un mensaje que solicita dinero bajo la falsa premisa de que un familiar ha sido detenido o está involucrado en un accidente, con la intención de que, en su afán por recuperar a sus seres queridos, se realice un depósito y se convierta en víctima de extorsión.

"Ahora que me enviaste estas fotos, si no me pagas, las voy a divulgar": En casos más graves, los extorsionadores amenazan con publicar imágenes personales o íntimas. Por lo general, engañan inicialmente a las víctimas para que envíen fotos de naturaleza íntima y luego exigen dinero para no compartir esa información con familiares.

"Tu hijo ha tenido un accidente y necesita atención médica urgente. Deposita dinero para cubrir los gastos médicos": En esta estrategia engañosa, los extorsionadores se hacen pasar por personal médico de emergencias y afirman que el hijo de la víctima ha sufrido un accidente grave. Solicitan un pago inmediato para cubrir los supuestos gastos médicos. Es crucial verificar la veracidad de la situación antes de tomar cualquier acción.

"Hemos secuestrado a alguien cercano a ti. Si no pagas el rescate, le haremos daño": Los extorsionadores utilizan el miedo como táctica para presionar a las víctimas. Aseguran tener a un familiar o amigo bajo su control y amenazan con causarle daño si no se realiza un pago. Es fundamental mantener la calma y confirmar la situación antes de responder a tales demandas.

"Tus fotos personales están en nuestro poder. Si no pagas, las compartiremos en línea": En este escenario, los extorsionadores amenazan con divulgar imágenes personales o comprometedoras. Solicitan dinero a cambio de no publicar las fotos. Se recomienda no ceder al chantaje y reportar inmediatamente el incidente a las autoridades correspondientes.

"Hemos hackeado tu cuenta y tenemos acceso a tus datos personales. Si no pagas, los revelaremos": Los extorsionadores envían correos electrónicos o mensajes afirmando haber comprometido la cuenta de la víctima y amenazan con revelar información confidencial. No se debe responder ni realizar pagos sin verificar la autenticidad de la amenaza y tomar medidas de seguridad adicionales.

Llamadas telefónicas con guiones predefinidos: Los delincuentes emplean scripts predefinidos en llamadas telefónicas, simulando ser familiares o premios ganados para engañar a las víctimas y obtener beneficios ilícitos.

Extorsión en redes sociales: Los extorsionadores interactúan con extraños en redes sociales, haciéndose pasar por amigos o responsables de la otra persona y afirmando que la conversación tenía un costo. Amenazan con visitar la casa de la víctima o su familia si no se realiza el pago.

Si recibes mensajes o llamadas sospechosas, es importante mantener la calma y hablar con alguien de confianza. Denuncia el incidente a las autoridades correspondientes, quienes tratarán tu caso con seriedad y confidencialidad[14].

1.3.4 Análisis de señales de voz

El análisis de las señales de la voz es un proceso crucial para comprender y extraer información relevante de las grabaciones de voz. Este análisis implica examinar detalladamente las características acústicas y lingüísticas presentes en las señales de voz, como el tono, la frecuencia, la intensidad y los patrones de habla. A través de técnicas de procesamiento de señales y algoritmos de aprendizaje automático, es posible identificar patrones distintivos, cambios en el discurso y otros indicios que pueden proporcionar información sobre el estado emocional, la intención comunicativa o incluso la detección de engaños. El análisis de las señales de la voz no

solo tiene aplicaciones en campos como la seguridad y la detección de extorsiones, sino que también se utiliza en áreas como la biometría vocal, la evaluación de la calidad de la voz y la síntesis de voz, contribuyendo a un mayor entendimiento y aprovechamiento de esta valiosa fuente de información humana[15].

El análisis de voz forense se especializa en el estudio y comparación de grabaciones de voz con propósitos legales o judiciales. Utilizando técnicas avanzadas y herramientas especializadas, los profesionales de este campo pueden identificar al hablante, verificar la autenticidad de las grabaciones y evaluar el estado emocional o físico de la persona durante la grabación[15].

1.3.5 Extracción de características en el reconocimiento de la voz

La extracción de características es un proceso crucial que selecciona información relevante de la forma de onda de audio sin procesar para crear una representación compacta que capture las características esenciales del habla. Estas características sirven como entrada para las etapas posteriores del proceso de reconocimiento de voz. A continuación, se explica este proceso desde distintas perspectivas, incluyendo las fórmulas matemáticas[16]:

a. Características del Dominio del Tiempo vs. Dominio de la Frecuencia

Características del Dominio del Tiempo: Estas se extraen directamente de la forma de onda de audio. Ejemplos incluyen la envolvente de amplitud, la tasa de cruce por cero y la energía[16].

Tasa de Cruce por Cero:

$$ZCR = \frac{1}{N-1}\sum_{n=1}^{N-1} \mathbf{1}_{\{x(n)x(n-1)<0\}} \tag{1}$$

donde 1 es la función indicadora, $x[n]$ es la señal de audio y N es el número de muestras en el segmento.

Energía:

$$\boldsymbol{E = \sum_{n=0}^{N-1} x[\mathrm{n}]^{\,2}} \tag{2}$$

Características del Dominio de la Frecuencia: Estas características se derivan del contenido espectral del audio. La representación más común en el dominio de la frecuencia son los coeficientes cepstrales de frecuencia Mel (MFCC).

b. Análisis a corto plazo

Señales no estacionarias: Las señales de habla no son estacionarias, lo que significa que sus propiedades cambian con el tiempo. Para manejar esto, se utiliza el análisis a corto plazo. Se divide la señal de audio en pequeños cuadros superpuestos (normalmente de 20 a 30 ms) y se calculan las características dentro de cada cuadro[16].

Aplicación de ventanas: Se aplican funciones de ventana (como Hamming o Hanning) a cada cuadro para reducir la fuga espectral.

Ventana Hamming:

$$\boldsymbol{\omega[\mathrm{n}] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)} \tag{3}$$

donde $w[n]$ es la función de ventana y N es el tamaño del cuadro.

1.3.6 MFCC (Coeficientes Cepstrales en las Frecuencias de Mel)

Los MFCC se calculan de la siguiente manera:

Preénfasis: Aplicar un filtro de paso alto para enfatizar las frecuencias más altas[16].

$$\mathbf{y[n] = x[n] - \alpha x[n-1]} \tag{4}$$

donde $y[n]$ es la señal preenfatizada y α es típicamente 0.95.

Segmentación en cuadros: Dividir la señal en cuadros de tamaño N.

Transformada Rápida de Fourier (FFT): Calcular el espectro de cada cuadro.

$$\mathbf{X[k]} = \sum_{\boldsymbol{n=0}}^{\boldsymbol{N-1}} \mathbf{x[n]} \boldsymbol{e^{-j\frac{2\pi kn}{N}}} \tag{5}$$

donde $X[k]$ es el espectro de frecuencia.

Banco de filtros Mel: Aplicar un conjunto de filtros triangulares en la escala Mel para obtener las energías del banco de filtros.

$$\boldsymbol{M = 2595 log_{10}(1 + \frac{f}{700})} \tag{6}$$

Logaritmo y Transformada de Coseno Discreta (DCT): Tomar el logaritmo de las energías del banco de filtros y aplicar la DCT para obtener los MFCC[17].

$$\boldsymbol{MFCC}\mathbf{[n]} = \sum_{\boldsymbol{k=0}}^{\boldsymbol{K-1}} \mathbf{log(E[k])\, cos}[\frac{\boldsymbol{\pi n(2k+1)}}{\boldsymbol{2K}}] \tag{7}$$

donde $E[k]$ es la energía del k-ésimo filtro y K es el número total de filtros.

c. Características Delta y Delta-Delta

Para capturar la dinámica, se calculan las características delta y delta-delta, que representan la tasa de cambio de los MFCC en cuadros adyacentes[16].

Delta:

$$\mathbf{\Delta MFCC[t]} = \frac{\sum_{n=1}^{N} \boldsymbol{n}(\mathbf{MFCC[t+n]} - \mathbf{MFCC[t-n]}}{\mathbf{2}\sum_{n=1}^{N} \boldsymbol{n^2}} \tag{8}$$

Delta-Delta:

$$\mathbf{\Delta^2 MFCC[t]} = \frac{\sum_{n=1}^{N} \boldsymbol{n}(\mathbf{\Delta MFCC[t+n]} - \mathbf{\Delta MFCC[t-n]}}{\mathbf{2}\sum_{n=1}^{N} \boldsymbol{n^2}} \tag{9}$$

d. Otros Tipos de Características

Además de los MFCC, hay otras características importantes[16]:

Tono: Representa la frecuencia fundamental de la voz, crucial para la entonación y la prosodia.

Formantes: Corresponden a frecuencias de resonancia en el tracto vocal y ayudan a distinguir las vocales.

Características Prosódicas: Capturan el ritmo, el énfasis y la duración.

Características Relacionadas con la Voz: Indican si un segmento es sonoro o sordo.

e. Ejemplo: Identificación del Hablante

Para identificar hablantes según su voz, se extraen los MFCC de sus grabaciones.

- Los vectores de características resultantes se introducen en un modelo de aprendizaje automático (como un modelo de mezcla gaussiana o una red neuronal) para su clasificación[16].

- Comparando estas características entre diferentes hablantes, podemos diferenciarlos en función de sus características vocales únicas.

1.3.7 Reconocimiento de voz: extracción de características MFCC y PLP

Se define algunos de los requisitos esenciales para la extracción de características en el reconocimiento automático de voz (ASR). En la siguiente Figura 1 se observa un segmento de audio, empleamos una ventana deslizante con un ancho de 25 ms para extraer las características del audio[18].

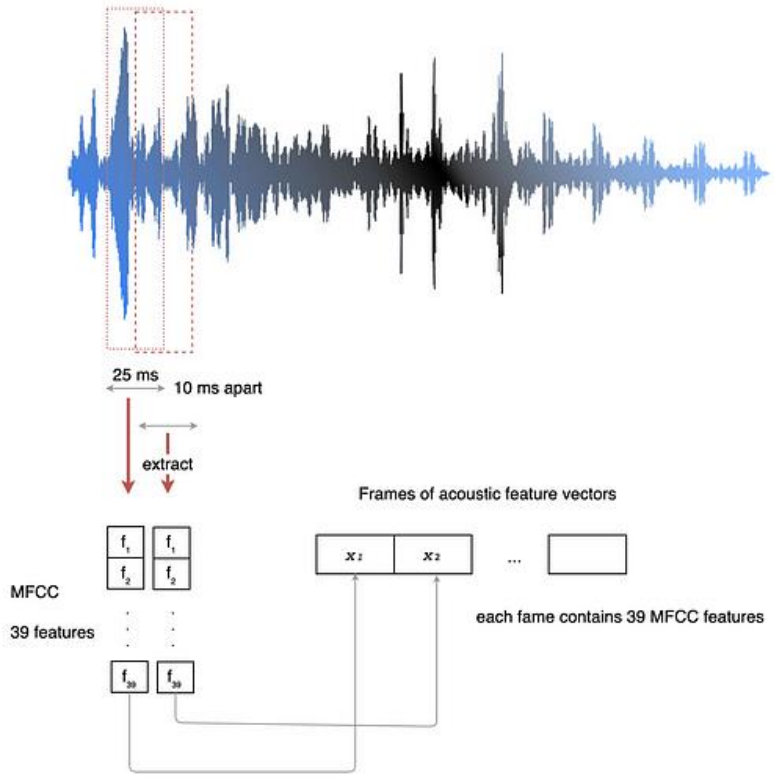


Figura 1. Ventana deslizante de 25 ms de ancho para extraer funciones de audio[18]

Este ancho de 25 ms es adecuado para capturar suficiente información mientras mantiene las características dentro de este marco relativamente estacionarias. Consideremos que, si pronunciamos 3 palabras por segundo, cada una con 4 fonemas y cada fonema se subdivide en 3 etapas, obtendremos 36 estados por segundo, lo que equivale a aproximadamente 28 ms por estado. Por lo tanto, en la siguiente Figura 2 podemos observar una ventana de 25 ms resulta ser ideal para este propósito[18].

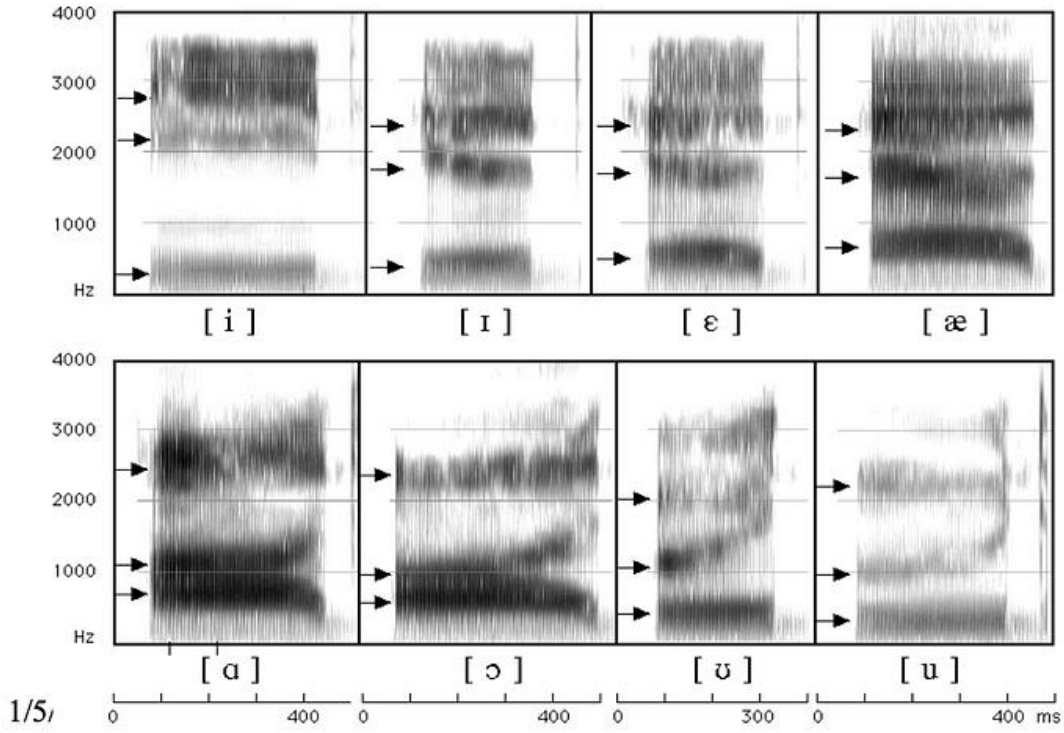


Figura 2. Ventana de 25 ms[18]

El contexto en el habla es fundamental, ya que las pronunciaciones cambian según la articulación anterior y posterior al fonema. Utilizamos ventanas deslizantes con un intervalo de aproximadamente 10 ms para capturar adecuadamente esta dinámica[18].

El tono varía entre personas, pero su impacto en el reconocimiento de palabras es mínimo. La frecuencia fundamental (F0), relacionada con el tono, no aporta valor al reconocimiento de voz y debe eliminarse. Los formantes, como F1, F2 y F3, son más importantes para este propósito.

Las características extraídas deben ser robustas frente a variaciones del hablante y ruido ambiental, y deben ser independientes para facilitar el desarrollo y entrenamiento de modelos de aprendizaje automático.

Un método común para extraer características de audio son los coeficientes cepstrales de frecuencia Mel (MFCC), que generan 39 características. Este número es lo suficientemente manejable para procesar la información de audio de manera eficiente. De estas, 12 están relacionadas con la amplitud de las frecuencias, proporcionando suficientes canales para un análisis detallado del audio[18].

A continuación, en la siguiente Figura 3 se muestra el flujo de extracción de características utilizando MFCC.

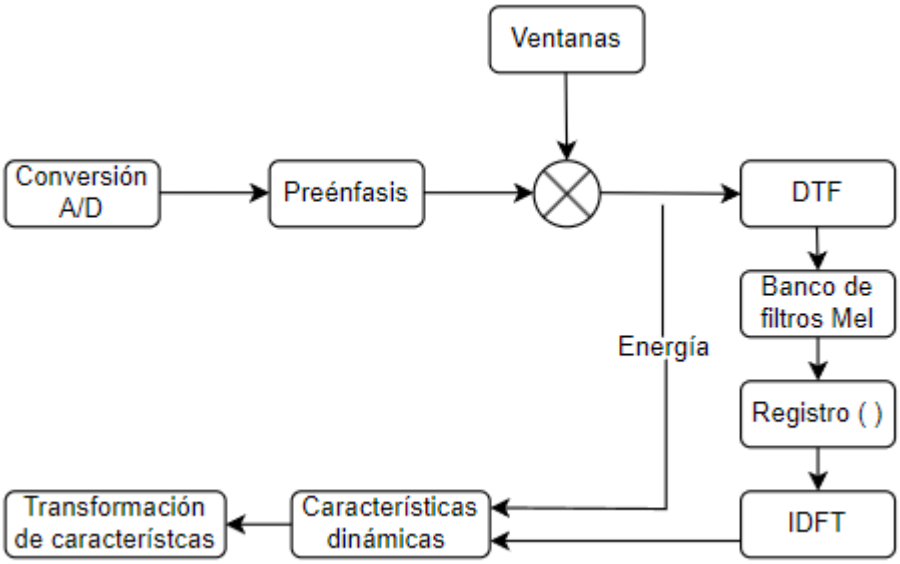


Figura 3. Proceso de extracción de características utilizando MFCC[18]

Los objetivos principales son[18]:

- Eliminar la excitación de las cuerdas vocales (F0), es decir, la información tonal.
- Asegurar que las características extraídas sean independientes.
- Adaptarse a la percepción humana del volumen y la frecuencia del sonido.
- Capturar la dinámica de los fonemas y su contexto.

a. Coeficientes cepstrales en la escala Mel (MFCC)

Los MFCC son ampliamente empleados para extraer características del audio, especialmente en aplicaciones de reconocimiento de voz. Este método convierte la señal de audio en un conjunto de atributos que representan las características espectrales del habla, lo cual simplifica su análisis y tratamiento computacional[18].

b. Conversión A/D

La conversión A/D toma muestras de los clips de audio y digitaliza su contenido, convirtiendo la señal analógica en una representación discreta. Comúnmente, se emplean frecuencias de muestreo de 8 o 16 kHz para este proceso como se observa en la siguiente Figura 4[18].

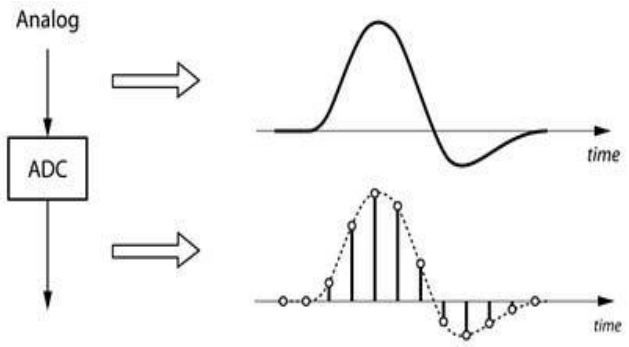


Figura 4.Conversión A/D[18]

c. Preénfasis

El preénfasis es un proceso que realza las frecuencias altas en una señal. En el caso de segmentos sonoros como las vocales, generalmente hay más energía en las frecuencias bajas debido a la inclinación espectral relacionada con la fuente glotal como se observa en la siguiente Figura 5. Aumentar la energía en las frecuencias altas, conocido como preénfasis, facilita al modelo acústico acceder mejor a la información en los formantes superiores, mejorando así la precisión en la detección de sonidos. Este enfoque también ayuda en ingeniería para reducir la susceptibilidad al ruido introducido más adelante en el proceso, dado que el ruido tiende a concentrarse en frecuencias altas[18].

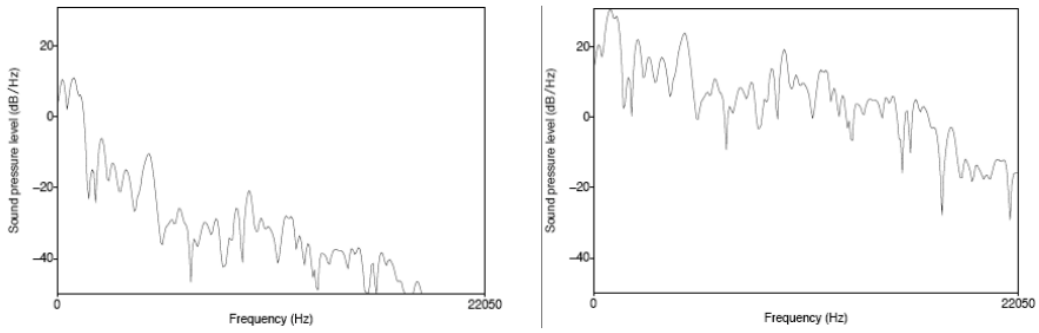


Figura 5. Preénfasis[18]

1.3.8 Ventanas

Crear ventanas implica dividir la forma de onda de audio en segmentos deslizantes como se observa en la Figura 6.

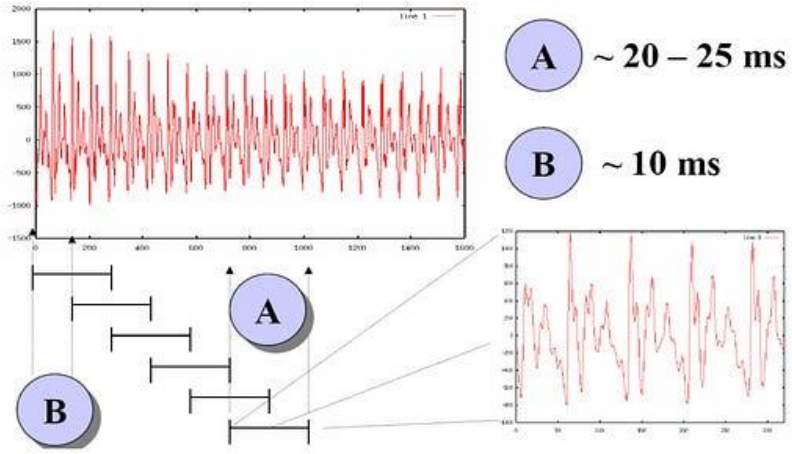


Figura 6. Audio en segmentos deslizantes[18]

Sin embargo, cortar la forma de onda en el borde del marco de manera abrupta causaría un ruido significativo en las altas frecuencias. Para dividir el audio adecuadamente, la amplitud debe reducirse de manera gradual cerca del borde de cada cuadro como se muestra en la Figura 7[18].

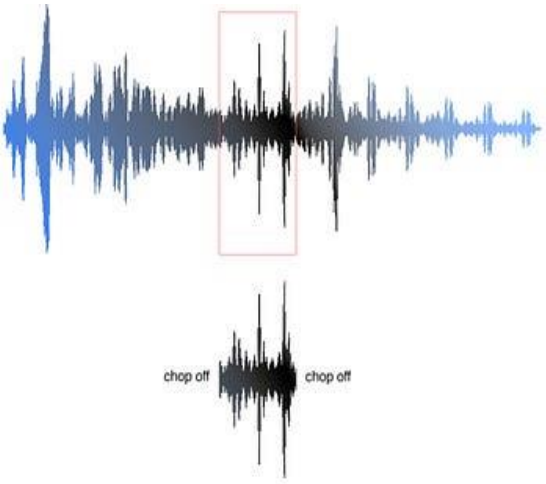


Figura 7. División del audio adecuadamente[18]

Supongamos que w representa la ventana aplicada al clip de audio original en el dominio temporal como se observa en la Figura 8.

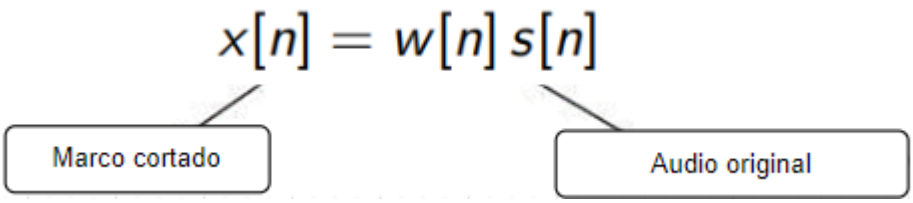


Figura 8. Entrada de marco cortado y audio original[18]

Existen otras opciones para w como la ventana de Hamming y la ventana de Hanning. Un diagrama ilustra cómo estas ventanas afectan el corte de una forma de onda sinusoidal. En la Figura 9 se observa que tanto la ventana de Hamming como la ventana de Hanning provocan una disminución gradual de la amplitud cerca del borde del segmento. Sin embargo, la ventana de Hamming presenta una ligera caída abrupta en comparación con la ventana de Hanning, que no presenta esta característica[18].

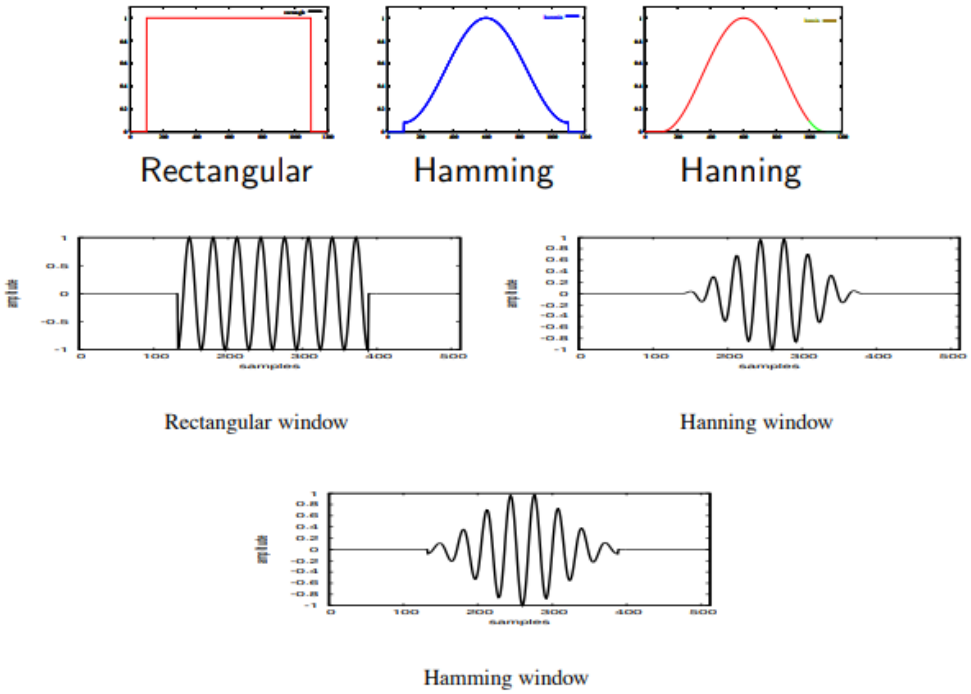


Figura 9. Ventanas de Hamming

Justo arriba y a la derecha, hay una forma de onda de sonido en el dominio temporal que contiene principalmente dos frecuencias. Como se puede observar en la Figura 10, el segmento cortado utilizando ventanas de Hamming y Hanning conserva mejor la información original de las frecuencias con menos interferencias en comparación con una ventana rectangular[18].

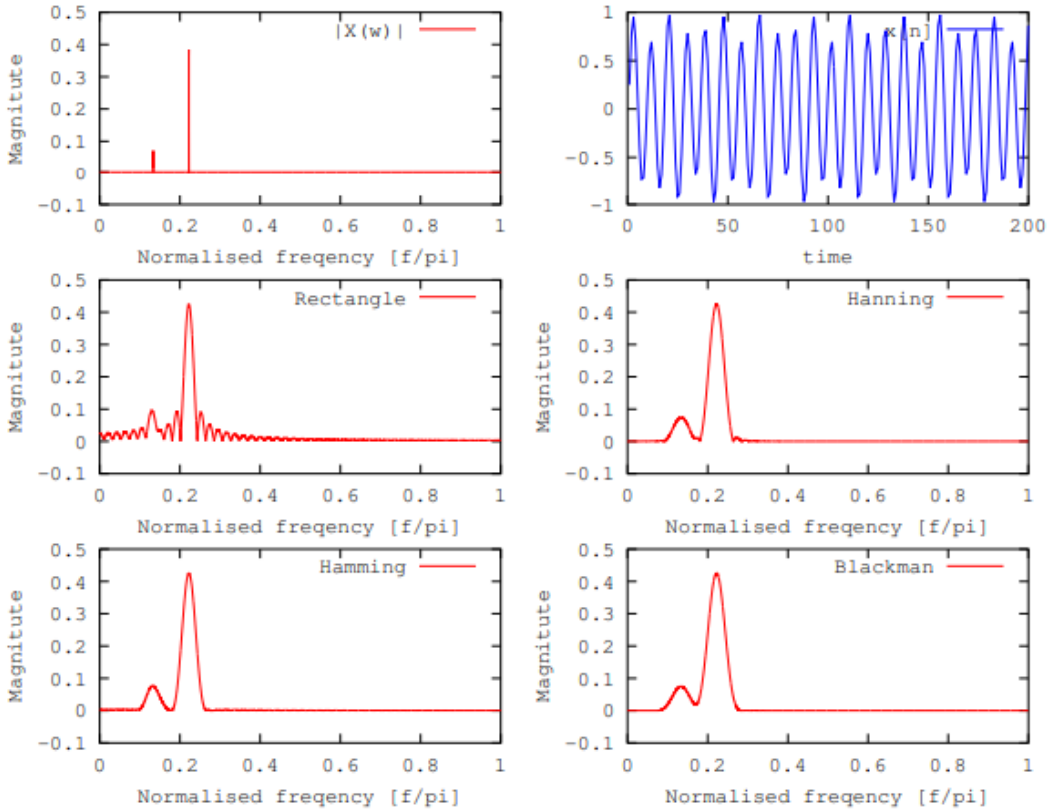


Figura 10. Ventanas de Hamming y Hanning[18]

d. Cepstrum — IDFT

El cepstrum y la Transformada Inversa de Fourier (IDFT) son técnicas empleadas en el procesamiento de señales para examinar la estructura espectral del habla y otros fenómenos acústicos como se puede observar en la Figura 11. El cepstrum se obtiene calculando la DFT de la señal logarítmica del espectro de potencia, lo cual revela detalles sobre las resonancias y la periodicidad de la señal. Posteriormente, la IDFT se utiliza para reconstruir la señal en el dominio del tiempo a partir del cepstrum modificado, permitiendo un análisis detallado de los componentes que influyen en la producción del habla y otras señales sonoras[19].

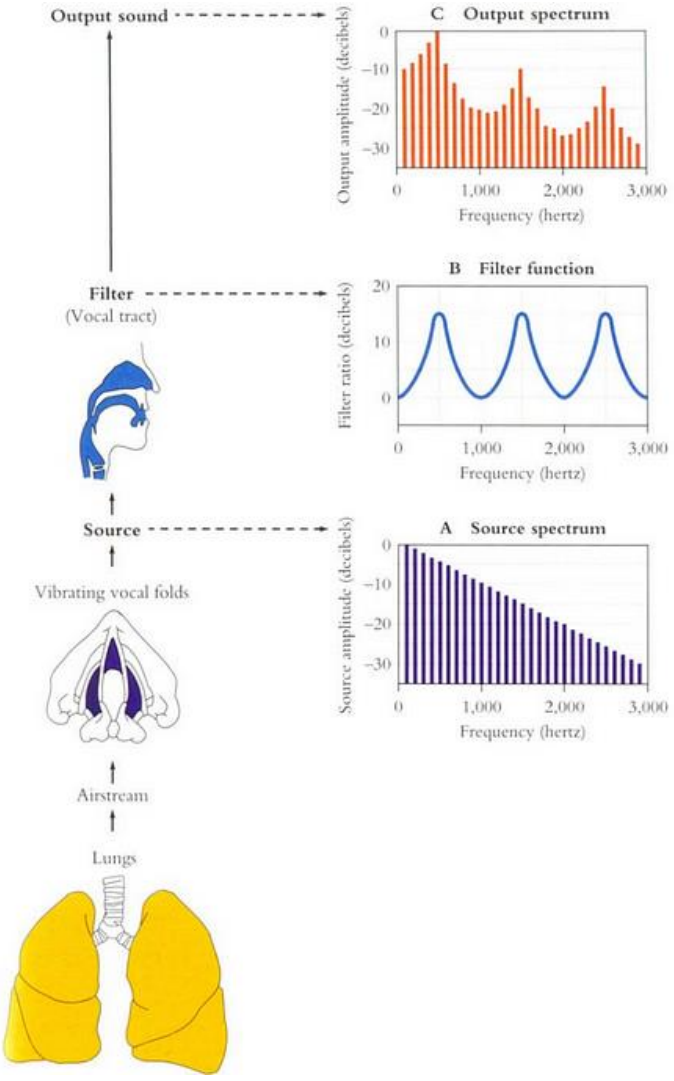


Figura 11. Estructura espectral del habla

e. Las características dinámicas (delta)

En el contexto de las MFCC se refieren a las 39 funciones que se calculan inicialmente. De estas, 12 son coeficientes MFCC estándar, mientras que las restantes son otras características derivadas. El decimotercer parámetro representa la energía presente en cada cuadro, lo cual es crucial para la identificación de los fonemas[18].

$$\boldsymbol{Energy} = \sum_{t_1}^{t_2} \boldsymbol{x^2[t]} \tag{10}$$

En la pronunciación, el contexto y la dinámica juegan roles críticos. Las transiciones entre formantes permiten identificar articulaciones como cierres y liberaciones de

consonantes. El seguimiento de cómo las funciones cambian a lo largo del tiempo ofrece información contextual para cada fonema. Además, se calculan 13 valores delta d(t), que representan los cambios entre características de fotogramas sucesivos. Estos valores son derivadas de primer orden y proporcionan detalles sobre la evolución de las características[18].

$$\boldsymbol{d(t) = \frac{c(t+1) - c(t-1)}{2}} \qquad (11)$$

Los últimos 13 parámetros representan las variaciones dinámicas de d(t) entre cuadros consecutivos, actuando como la derivada de segundo orden de c(t). En total, los 39 parámetros de las características de MFCC consisten en 12 coeficientes de Cepstrum más un término de energía. Además de estos, hay dos conjuntos adicionales que corresponden a los valores delta y doble delta. En la siguiente Figura 12se observa el proceso del reconocimiento de la voz[18].

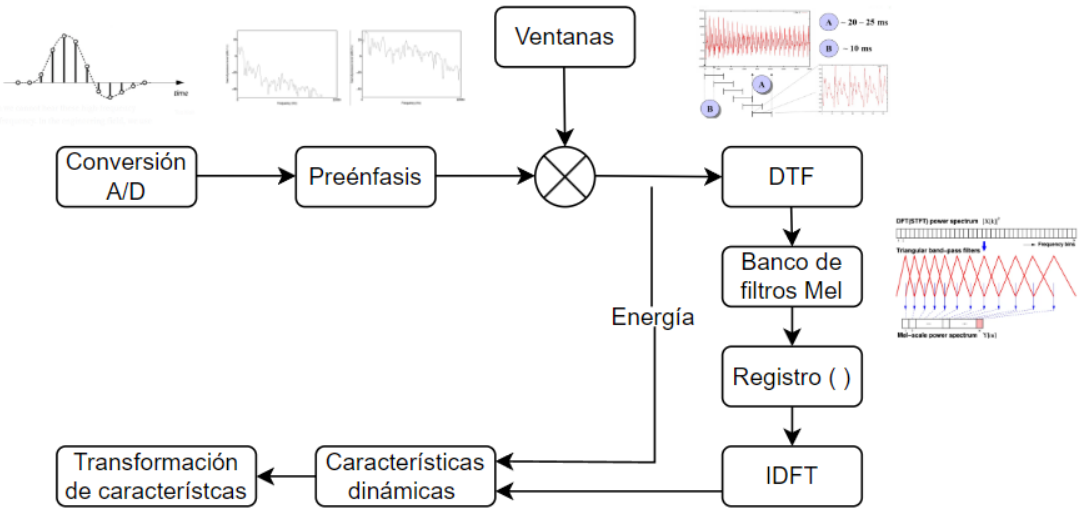


Figura 12. Proceso del reconocimiento de la voz[18]

1.3.9 Reconocimiento de la voz

El reconocimiento de voz es un campo interdisciplinario dentro de la lingüística computacional que desarrolla metodologías y tecnologías para que las computadoras puedan reconocer y convertir el lenguaje hablado en texto. Este proceso también se conoce como reconocimiento automático de voz, reconocimiento de voz por

computadora o conversión de voz a texto, y abarca conocimientos de lingüística, informática e ingeniería eléctrica[20].

Algunos sistemas de reconocimiento de voz requieren un proceso de "aprendizaje" o "inscripción", en el cual una persona lee un texto o un conjunto de palabras específicas para que el sistema analice su voz y mejore la precisión del reconocimiento para esa persona en particular. Los sistemas que no requieren este proceso se llaman "independientes del hablante", mientras que los que sí lo requieren se llaman "dependientes del hablante".

a. Como funciona el reconocimiento de la voz

El reconocimiento de audio se lleva a cabo en tres etapas[20]:

1. Identificación de palabras: Se resaltan las palabras y se generan varias hipótesis para cada palabra reconocida.

2. Verificación de hipótesis: Se utilizan modelos de lenguaje para evaluar la coherencia de las nuevas palabras con las ya reconocidas.

3. Procesamiento del texto: Se convierten números en dígitos, se añaden signos de puntuación como guiones, entre otros ajustes. El texto final procesado se envía como resultado del reconocimiento.

b. Tipos de reconocimiento del habla

Los sistemas de reconocimiento de voz se pueden clasificar según varios criterios[20]:

- **Funcionalidad:** Sistemas de dictado y sistemas de comando.
- **Tipo de discurso:** Voz continua o fragmentada.
- **Tamaño del diccionario:** Conjuntos limitados de palabras o diccionarios extensos.
- **Dependencia del hablante:** Sistemas dependientes de la voz del usuario.

- **Mecanismo de funcionamiento:** Detectores simples o correlacionados, sistemas expertos con diferentes configuraciones y procesamientos de la base de conocimiento, y modelos probabilísticos y de redes, incluidas las redes neuronales.
- **Algoritmo utilizado:** Redes neuronales, modelos ocultos de Markov, o programación dinámica.
- **Tipo de unidad estructural:** Frases, palabras, fonemas, difonos y alófonos.
- **Principio de selección de unidades estructurales**: Reconocimiento de patrones y selección de elementos léxicos.

1.3.10 Reconocimiento del habla

El reconocimiento del habla, también conocido como reconocimiento automático del habla (ASR) o conversión de voz a texto, utiliza el procesamiento de lenguaje natural (PLN) para convertir el habla humana en texto escrito. A diferencia del reconocimiento de voz, que identifica la voz de un usuario, el reconocimiento del habla traduce el discurso verbal a texto[21].

IBM ha sido pionero en este campo desde el lanzamiento de "Shoebox" en 1962, capaz de reconocer 16 palabras. Esto mejoró el trabajo de Bell Labs de los años 50. En 1996, IBM lanzó VoiceType Simply Speaking, un software que reconocía 42,000 palabras en inglés y español, con un diccionario de ortografía de 100,000 palabras. Aunque al principio el vocabulario de la tecnología era limitado, hoy se aplica en industrias como la automotriz, la tecnología y la salud. Su adopción ha crecido gracias a los avances en deep learning y big data, y se espera que el mercado alcance un valor de USD 24.9 mil millones para 2025[21].

a. Características del Reconocimiento del Habla Efectivo

El reconocimiento del habla efectivo se ha transformado con el avance de la inteligencia artificial y el aprendizaje automático, permitiendo integrar aspectos complejos como gramática, sintaxis y la estructura de señales de audio y voz. Estas soluciones avanzadas están diseñadas para adaptarse específicamente a las necesidades

empresariales, ofreciendo características adicionales que mejoran significativamente su funcionalidad[21]:

- **Ponderación del idioma:** Además de reconocer palabras comunes, estas soluciones mejoran la precisión al dar mayor peso a términos específicos relevantes para cada dominio o industria.

- **Clasificación de hablantes:** Generan transcripciones precisas que atribuyen las contribuciones a cada orador, facilitando la identificación y seguimiento de discusiones y conversaciones grupales.

- **Entrenamiento en acústica:** Adaptan dinámicamente su capacidad de reconocimiento a diferentes ambientes acústicos y estilos de habla, asegurando resultados consistentes y precisos en diversas condiciones.

- **Filtrado de obscenidades:** Implementan funciones que detectan y suavizan automáticamente palabras o frases inapropiadas, asegurando un ambiente de trabajo o comunicación seguro y profesional.

b. Métodos de reconocimiento del habla

El desarrollo de algoritmos para el reconocimiento del habla es una tarea compleja que involucra la integración de lingüística, matemáticas y estadística. Estos sistemas incluyen componentes como la captura de voz, la extracción de características, el uso de vectores y un decodificador para convertir el habla en texto[21].

Se utilizan diversos métodos y técnicas para mejorar la precisión de la transcripción del habla. Entre ellos se destacan:

- **Procesamiento de Lenguaje Natural (PLN):** Utilizado para la interacción máquina-humano a través del habla y texto, facilitando funciones como las búsquedas por voz en dispositivos móviles como Siri.

- **Modelos Ocultos de Márkov (HMM):** Empleados para modelar secuencias en el reconocimiento del habla, asignando etiquetas a unidades como palabras o sílabas, mejorando así la precisión.

- **N-gramas:** Modelos simples de lenguaje que asignan probabilidades a secuencias de palabras, optimizando el reconocimiento al considerar la gramática y frecuencia de las secuencias.

- **Redes Neuronales:** Algoritmos de deep learning que aprenden estructuras complejas de datos, aunque requieren más tiempo para entrenarse, ofrecen alta precisión y capacidad para manejar grandes volúmenes de datos.

- **Diarización de hablantes (SD):** Identifica y segmenta hablantes en una conversación, útil en entornos como centros de llamadas para distinguir entre clientes y agentes[21].

1.3.11 Reconocimiento de la voz en el software Python

El reconocimiento de voz es un campo emocionante en el ámbito de la inteligencia artificial que ha experimentado un crecimiento significativo recientemente. Este avance tecnológico capacita a las computadoras para comprender y procesar el lenguaje humano hablado, facilitando la creación de aplicaciones innovadoras como asistentes virtuales, sistemas de transcripción automática de audio, y el control mediante voz de dispositivos tecnológicos diversos. Python se destaca como una elección excepcional para desarrollar proyectos de reconocimiento de voz debido a su amplio conjunto de bibliotecas y herramientas especializadas. La biblioteca SpeechRecognition, no solo simplifican la captura y el análisis de señales de audio, sino que también permite la implementación eficiente de modelos de aprendizaje automático y redes neuronales para mejorar la precisión del reconocimiento. La comunidad activa de Python y su documentación extensa hacen que sea accesible para desarrolladores de diferentes niveles de experiencia, facilitando así la innovación continua en este campo en rápido desarrollo. En la siguiente Figura 13 se puede observar el proceso de reconocimiento de la voz en Python[22].

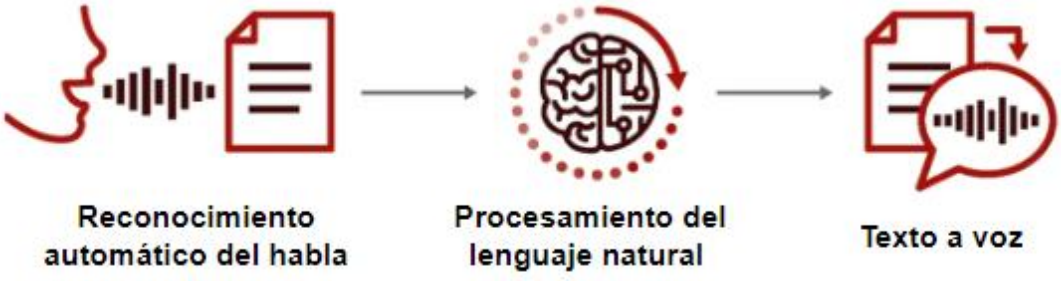


Figura 13. Reconocimiento de la voz en Python[22]

c. Librería SpeechRecognition

SpeechRecognition es una herramienta muy valorada por su facilidad de uso, flexibilidad y capacidad de integración sin complicaciones. Con SpeechRecognition, puedes realizar varias funciones esenciales, como convertir archivos de audio en texto, transcribir audio capturado por micrófono directamente a texto y manejar eficazmente el ruido presente en el audio[22].

El paquete no solo es una biblioteca, funciona como un contenedor para diversas APIs populares de reconocimiento de voz, como la API web de Google y la API de OpenAI, entre otras. Estas APIs están integradas en SpeechRecognition con claves API predefinidas, lo que te permite comenzar a trabajar sin la necesidad de registrarte en servicios adicionales. Esta flexibilidad y facilidad de integración hacen de SpeechRecognition la opción ideal para proyectos en Python que requieren capacidades avanzadas de procesamiento de voz[22].

El reconocimiento de voz en Python opera con algoritmos que llevan a cabo el modelado lingüístico y acústico. El modelado acústico se emplea para identificar fenómenos y fonética en el habla, con el fin de extraer las partes más significativas, como palabras y oraciones. El proceso de reconocimiento de voz comienza al capturar la energía sonora producida por la persona que habla y convertirla en energía eléctrica mediante un micrófono. Posteriormente, esta energía eléctrica se convierte de analógica a digital y, finalmente, en texto como se puede observar en la siguiente Figura 14[23].

Los datos de audio se descomponen en sonidos y se analizan mediante algoritmos para encontrar la palabra más probable que corresponda a ese audio. Todo esto se lleva a cabo mediante el procesamiento del lenguaje natural y las redes neuronales. Los modelos ocultos de Markov pueden ser utilizados para identificar patrones temporales en el habla y mejorar la precisión[23].

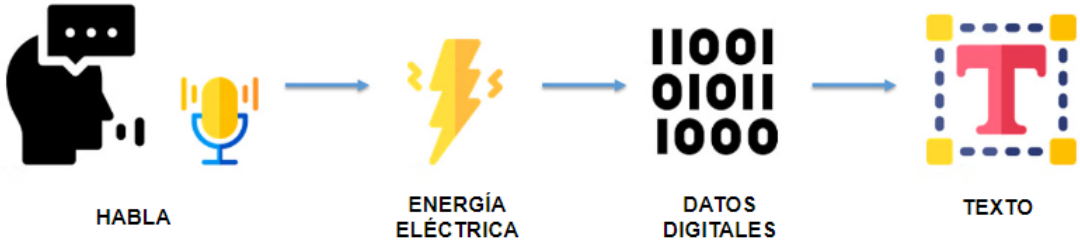


Figura 14. Funcionamiento del reconocimiento de la voz[23]

d. Etapas de funcionamiento de la librería SpeechRecognition

La librería SpeechRecognition en Python opera a través de varias etapas. En primer lugar, se captura la señal de voz y se digitaliza utilizando un micrófono. A continuación, se preprocesa la señal digital para mejorar su calidad y minimizar el ruido. Después, se extraen características clave de la señal de voz. Finalmente, estas características se analizan mediante modelos lingüísticos y acústicos para transcribir el habla a texto. Se puede observar las etapas de funcionamiento que realiza la librería en la siguiente Tabla 1[23].

Tabla 1. Etapas de funcionamiento de la librería SpeechRecognition

Etapa	Descripción	Método/Función
Captura de audio	Captura de la señal de voz a través de un micrófono o carga de un archivo de audio.	Microphone, AudioFile
Digitalización	Conversión de la señal analógica de voz en una señal digital utilizando un convertidor analógico-digital (ADC).	Realizado por el hardware (micrófono/ordenador)
Preprocesamiento del audio	Mejora de la calidad del audio (eliminación de ruido, normalización, etc.) y preparación para el reconocimiento.	Implementado internamente por la API de reconocimiento de voz
Extracción de características	Extracción de características acústicas del audio digital para facilitar el reconocimiento de voz.	Implementado internamente por la API de reconocimiento de voz
Reconocimiento de voz	Utilización de modelos de lenguaje y acústicos para transcribir el audio a texto.	recognize_google, recognize_ibm, recognize_sphinx, etc.

Postprocesamiento	Corrección de errores y refinamiento del texto transcrito.	Implementado internamente por la API de reconocimiento de voz
Salida de texto	Devolución del texto transcrito al usuario.	recognize_* (por ejemplo, recognize_google devuelve el texto transcrito)

e. Características de las señales de la voz de la librería SpeechRecognition

La etapa de extracción de características es esencial en el proceso de reconocimiento de voz. Aunque la librería SpeechRecognition delega esta tarea a los servicios de reconocimiento de voz que utiliza, como Google Web Speech API e IBM Watson, es importante comprender las características comunes que suelen extraerse en este proceso. Estas características incluyen elementos cruciales del habla que permiten una transcripción precisa y efectiva como se las puede observar en la siguiente Tabla 2[23].

Tabla 2. Características de las señales de la voz en la librería SpeechRecognition

Característica	Descripción	Métodos de Extracción
MFCCs (Mel-Frequency Cepstral Coefficients)	Representación compacta del espectro de frecuencias de la señal de voz en la escala Mel.	Transformada de Fourier, filtro Mel, análisis de cepstrum.
Frecuencia Fundamental (F0)	La frecuencia más baja de la voz, determinando el tono básico.	Autocorrelación, algoritmo de cepstrum, análisis de espectro.
Formantes	Picos en el espectro de frecuencias que corresponden a las resonancias del tracto vocal.	Análisis de espectro, método LPC (Linear Predictive Coding).
Energía de la señal	Cantidad de energía en la señal de voz, relacionada con el volumen percibido.	RMS (Root Mean Square), suma de cuadrados de la amplitud.
ZCR (Zero-Crossing Rate)	Frecuencia con la que la señal cruza el eje cero, relacionada con la frecuencia de los sonidos.	Conteo de cruces por cero en la señal temporal.
Spectral Roll-off	Frecuencia por debajo de la cual se encuentra un porcentaje determinado de la energía total del espectro.	Transformada de Fourier, análisis de espectro.
Spectral Centroid	Centro de gravedad del espectro de frecuencias, que indica dónde se concentra la mayor parte de la energía.	Transformada de Fourier, análisis de espectro.
Chroma Feature	Representación de la energía de la señal en 12 bandas de frecuencia que corresponden a las 12 notas musicales.	Transformada de Fourier, mapeo a escala de notas.
Bandwidth	Ancho de banda del espectro de la señal de voz.	Análisis de espectro, Transformada de Fourier.
Spectral Flatness	Medida que indica qué tan plana es la distribución espectral.	Transformada de Fourier, cálculo de ratio entre la media geométrica y aritmética.
Jitter	Variabilidad en la frecuencia fundamental.	Análisis de series temporales de F0.
Shimmer	Variabilidad en la amplitud de la señal de voz.	Análisis de la envolvente temporal.

HNR (Harmonics-to-Noise Ratio)	Relación entre los componentes armónicos y no armónicos de la voz, indicando la claridad de la voz.	Análisis de espectro, algoritmos de descomposición.

1.3.12 Procesamiento de lenguaje natural (NLP)

El procesamiento de lenguaje natural (NLP) es una disciplina que se centra en la interacción entre los sistemas de computadoras y el lenguaje humano. El NLP desempeña un papel fundamental en el análisis y comprensión de las expresiones lingüísticas utilizadas en las grabaciones de voz. A través de técnicas avanzadas de procesamiento de lenguaje, como el etiquetado gramatical, la desambiguación léxica y la extracción de información, se puede analizar el contenido textual y lingüístico presente en las señales de voz. Esto permite identificar palabras clave, patrones lingüísticos, cambios en el discurso y otros elementos relevantes para la detección de extorsiones. El procesamiento de lenguaje natural aporta una capa adicional de análisis a las señales de la voz, brindando una comprensión más completa y precisa del contenido verbal y facilitando la detección de patrones y señales engañosas en las interacciones telefónicas[19].

1.3.13 Detección de extorsiones con inteligencia artificial

La implementación de sistemas de detección de extorsiones basados en inteligencia artificial ha demostrado ser ventajosa en múltiples sectores, proporcionando beneficios significativos tanto en la seguridad pública como en los ámbitos empresarial y financiero. En el contexto de la seguridad ciudadana, estos sistemas automatizados permitieron a las autoridades detectar y prevenir actividades delictivas de manera más eficiente, fortaleciendo así la protección de la sociedad frente a amenazas de extorsión.

En el entorno empresarial y financiero, la detección de extorsiones mediante inteligencia artificial ayudó a las organizaciones a protegerse contra posibles pérdidas económicas y daños a su reputación al identificar y neutralizar situaciones de extorsión de manera proactiva. Esta tecnología también se aplicó en sectores como la ciberseguridad, contribuyendo a proteger los sistemas informáticos y la información sensible de las empresas contra amenazas cibernéticas relacionadas con la extorsión.

La integración de la inteligencia artificial en la detección de extorsiones no solo mejoró la seguridad y protección en una variedad de sectores, sino que también fortaleció la resiliencia y la confianza en las operaciones comerciales y financieras. La capacidad de estas tecnologías para operar en tiempo real y proporcionar alertas inmediatas permitió a las entidades responder rápidamente a amenazas potenciales, minimizando el impacto de las extorsiones y asegurando una respuesta más efectiva y coordinada.

La adopción de estas soluciones basadas en inteligencia artificial representó un avance significativo en la lucha contra la extorsión, ofreciendo un enfoque innovador y eficiente para abordar este problema en distintos contextos. En resumen, la integración de inteligencia artificial en la detección de extorsiones incrementó la seguridad y la protección en diversos sectores, promoviendo un entorno más seguro y confiable tanto para las instituciones públicas como para las privadas.

1.3.14 Aprendizaje automático (Machine Learning)

El aprendizaje automático, también conocido como machine learning, es una rama de la inteligencia artificial que se centra en desarrollar algoritmos y modelos que permiten a los sistemas informáticos aprender y mejorar automáticamente a partir de los datos. En el contexto de este informe, el aprendizaje automático desempeña un papel esencial en el análisis de las señales de voz para la detección de extorsiones. Al utilizar conjuntos de datos que contienen grabaciones de llamadas telefónicas legítimas y fraudulentas, los algoritmos de aprendizaje automático pueden identificar patrones, características y señales distintivas asociadas con el engaño. Estos modelos pueden ser entrenados para reconocer y clasificar automáticamente las llamadas telefónicas fraudulentas, brindando un sistema eficiente y confiable para la detección de extorsiones. El aprendizaje automático permite una detección más precisa y ágil, adaptándose a patrones cambiantes y mejorando continuamente su desempeño a medida que se alimenta con más datos. En resumen, el aprendizaje automático desempeña un papel fundamental en el análisis de las señales de voz para la detección de extorsiones, permitiendo desarrollar sistemas inteligentes capaces de identificar patrones engañosos y brindar una respuesta efectiva en la lucha contra estos delitos[19].

f. Tipos de Aprendizaje en Algoritmos de Machine Learning

Los modelos de Machine Learning, o aprendizaje automático, son capaces de aprender a realizar tareas sin necesidad de ser programados específicamente para ellas. Estos algoritmos pueden detectar patrones en grandes cantidades de datos y hacer predicciones, lo que se aplica en motores de búsqueda inteligentes, sistemas de diagnóstico, reconocimiento de imágenes, voz y lenguaje, entre otros[24].

Se pueden clasificar los siguientes tipos de algoritmos en la siguiente Tabla 3.

Tabla 3. Tipos de algoritmos de Machine learning [24]

Tipo de Aprendizaje	Descripción
Supervisado	Utiliza datos etiquetados para entrenar modelos que predicen resultados. Incluye modelos de regresión (resultados cuantitativos) y de clasificación (resultados cualitativos).
No Supervisado	Analiza datos no etiquetados para encontrar patrones y estructuras ocultas, útil para agrupar datos similares (clustering) y reducir la dimensionalidad de los datos.
Semi-supervisado	Combina datos etiquetados y no etiquetados para mejorar la precisión de los modelos cuando etiquetar todos los datos es costoso o difícil.
Por Refuerzo	Aprende mediante recompensas y castigos en un entorno dinámico, mejorando sus acciones para maximizar una función de recompensa.

g. Algoritmos de aprendizaje supervisado

Los algoritmos de machine learning son cruciales para mejorar los procesos de aprendizaje, ya que facilitan la obtención de información necesaria para tomar decisiones y prever el comportamiento de los datos. A continuación, se detallan algunos de los algoritmos más comúnmente empleados[25].

- **Regresión Lineal (Linear Regression)**

Este método se emplea para prever valores continuos. Modela la relación entre una variable dependiente y una o más variables independientes mediante el ajuste de una línea recta a los datos. Es útil para tareas como la predicción de precios de viviendas y la estimación de ventas[25].

- **Regresión Logística (Logistic Regression)**

Aunque su nombre pueda sugerir otra cosa, este algoritmo se aplica principalmente a problemas de clasificación binaria. Calcula la probabilidad de una variable dependiente binaria basándose en una o varias variables independientes mediante una función sigmoide. Es frecuentemente utilizado en diagnósticos médicos y para detectar correos electrónicos no deseados[25].

- **Árboles de Decisión (Decision Trees)**

Este algoritmo crea un modelo en forma de árbol donde cada nodo interno representa una característica del conjunto de datos, cada rama representa una regla de decisión y cada hoja un resultado. Es utilizado en la evaluación de crédito y diagnósticos médicos[25].

- **Bosques Aleatorios (Random Forests)**

Mejora los árboles de decisión al crear múltiples árboles y combinarlos para obtener predicciones más precisas y estables, utilizando el método de bagging. Es ampliamente utilizado en la predicción de enfermedades y la clasificación de imágenes[25].

- **Redes Neuronales Artificiales (Artificial Neural Networks, ANN)**

Inspirados en el cerebro humano, estos modelos consisten en capas de nodos (neuronas) y pueden aprender representaciones complejas de datos. Son efectivos en el reconocimiento de voz, procesamiento de imágenes y juegos[25].

- **Naive Bayes**

Basado en el teorema de Bayes, este método asume la independencia de las características y calcula la probabilidad de que un ejemplo pertenezca a una clase específica. Se emplea en la clasificación de texto, detección de spam y análisis de sentimientos[25].

h. Componentes de Machine learning

El proceso de Machine Learning incluye varios componentes fundamentales que trabajan en conjunto para lograr la automatización del aprendizaje de un modelo. Estos componentes se describen en la Tabla 4.

Tabla 4. Componentes del Machine Learning

Parte	Descripción
Recolección de Datos	Proceso de adquirir datos pertinentes al problema que se desea abordar.
Preparación de Datos	Incluye limpieza, normalización, transformación y división de los datos en conjuntos de entrenamiento y prueba.
Selección de Características	Identificación de las variables más relevantes que influirán en el modelo.
Ingeniería de Características	Creación de nuevas variables a partir de las existentes para mejorar el rendimiento del modelo.
División de Datos	Separación de los datos en conjuntos de entrenamiento, validación y prueba.
Selección de Modelo	Elección del algoritmo de machine learning más adecuado para el problema específico.
Entrenamiento del Modelo	Ajuste del modelo a los datos de entrenamiento para que aprenda a hacer predicciones o clasificaciones.
Evaluación del Modelo	Medición del rendimiento del modelo usando el conjunto de prueba y diversas métricas de evaluación.
Ajuste de Hiperparámetros	Optimización de los parámetros del modelo para mejorar su rendimiento.
Validación Cruzada	Técnica para evaluar la capacidad de generalización del modelo dividiendo los datos en múltiples subconjuntos.
Implementación del Modelo	Despliegue del modelo en un entorno de producción para que realice predicciones en tiempo real o por lotes.
Monitoreo y Mantenimiento	Supervisión del rendimiento del modelo en producción y actualización según sea necesario.

i. Etapas de Machine Learning

El proceso de Machine Learning incluye varias etapas, que se detallan a continuación en la Tabla 5.

Tabla 5. Etapas de Machine Learning

Etapa	Descripción
Definir el objetivo	Es crucial comprender el problema y ser claros sobre nuestro objetivo, considerando las características de la empresa y la data disponible. Machine Learning depende principalmente de la data. Preguntas típicas incluyen: ¿Qué queremos hacer? ¿Cómo podemos hacerlo? ¿Es posible con la data que tenemos?
Recolección de la data	Tras definir el objetivo, recolectamos la data necesaria para el algoritmo de Machine Learning. Es fundamental entender las fortalezas y limitaciones de la data. A veces, necesitaremos datos de sistemas externos, que pueden ser gratuitos o pagos, y en ocasiones, puede que los datos no estén disponibles.
Preparar la data	Con la data recolectada, procedemos al preprocesamiento, que incluye limpieza y formateo. El objetivo es manipular los datos para obtener mejores resultados. Ejemplos incluyen eliminar o inferir datos faltantes, categorizar valores, y normalizar o escalar valores numéricos.
Elección del algoritmo	Elegimos el algoritmo más adecuado para resolver el problema. Podemos optar por aprendizaje supervisado (e.g., Linear Regression, Logistic Regression, Decision Tree, K-Nearest Neighbors, Support Vector Machine) o no supervisado (e.g., K-Means Clustering, Hierarchical Clustering, Principal Component Analysis).

Entrenar el modelo	Separamos la data preprocesada, usando un porcentaje (generalmente 70%) como data de entrenamiento, aplicando el algoritmo seleccionado para alcanzar el objetivo inicial.
Validación del modelo	Con el modelo entrenado, validamos usando la data restante, llamada data de validación. Evaluamos los resultados para asegurarnos de que el modelo funcione bien en ambas particiones para evitar el overfitting. Repetimos esta etapa hasta que el modelo se ajuste adecuadamente a ambas particiones.
Predicción	Superado el problema de overfitting, realizamos predicciones ingresando nueva data en nuestro modelo.

1.3.15 Algoritmos de inteligencia artificial

Los algoritmos de inteligencia artificial constituyen el núcleo de las tecnologías modernas que simulan capacidades cognitivas humanas. Estos algoritmos permiten a las máquinas aprender de datos, realizar predicciones y tomar decisiones basadas en patrones y experiencias previas. Desde el aprendizaje automático hasta el procesamiento del lenguaje natural, estos algoritmos desempeñan un papel fundamental en una amplia gama de aplicaciones, facilitando desde la automatización de tareas hasta la creación de sistemas inteligentes capaces de interactuar y comprender el mundo de manera más eficaz.

a. Redes LSTM

Las redes LSTM (Long short-term memory) son un tipo de redes neuronales recurrentes (RNN) utilizadas en el aprendizaje profundo para procesar y predecir secuencias de datos. Estas redes fueron diseñadas para solucionar el problema de la desaparición del gradiente en las redes neuronales recurrentes tradicionales, problema que ocurre cuando el error se retropropaga a través de múltiples capas y se pierde información importante en el proceso. Las LSTM emplean una estructura de celda con puertas que permite a la red controlar la cantidad de información que se almacena y se olvida en cada paso de tiempo, lo que las hace especialmente adecuadas para el procesamiento de secuencias de datos a largo plazo. Las redes LSTM han sido aplicadas con éxito en una amplia variedad de áreas del aprendizaje profundo, como el procesamiento del lenguaje natural, el reconocimiento de voz, la generación de texto y la predicción de series temporales[26].

b. Estructura de una Red LSTM

Las redes LSTM (Long Short-Term Memory) poseen la habilidad de "recordar" datos relevantes en una secuencia y conservarlos durante varios intervalos de tiempo. Esto les permite mantener una memoria tanto a corto plazo, similar a las redes recurrentes básicas, como a largo plazo[27].

Su funcionamiento es comparable a cómo el cerebro humano procesa secuencias. Por ejemplo, cuando una persona quiere comprar unos audífonos y lee una reseña de otro comprador, no se centra en todo el texto. En lugar de eso, se enfoca solo en las palabras relevantes y descarta el resto de la información[27].

Una red LSTM está compuesta por una serie de celdas, cada una de las cuales incluye tres "puertas" principales: la puerta de entrada, la puerta de salida y la puerta de olvido. Estas puertas regulan el flujo de información dentro de la celda, permitiendo que ciertos datos se almacenen en la memoria a largo plazo mientras otros se olvidan[27].

1. **Puerta de entrada**: Controla qué parte de la información de entrada se almacena en la memoria de la celda.
2. **Puerta de olvido**: Determina qué información de la celda debe ser olvidada.
3. **Puerta de salida**: Decide qué parte de la información de la celda se utiliza para la salida en ese instante de tiempo.

Estas características hacen que las redes LSTM sean particularmente eficaces en tareas donde es crucial manejar información a lo largo de secuencias prolongadas. Han sido ampliamente utilizadas en áreas como el procesamiento del lenguaje natural, el reconocimiento de voz, la generación de texto y la predicción de series temporales[28].

Existen redes neuronales en las cuales es posible que dos neuronas, o dos partes de una red neuronal, interactúen y transmitan información de manera recíproca en lugar de unidireccional. Este tipo de redes, conocidas como redes neuronales recurrentes, son capaces de formar bucles de retroalimentación (feedback). Actualmente, las redes neuronales recurrentes tienen aplicaciones en diversas áreas, incluyendo el

procesamiento del lenguaje natural como se puede observar en la siguiente Figura 15[28].

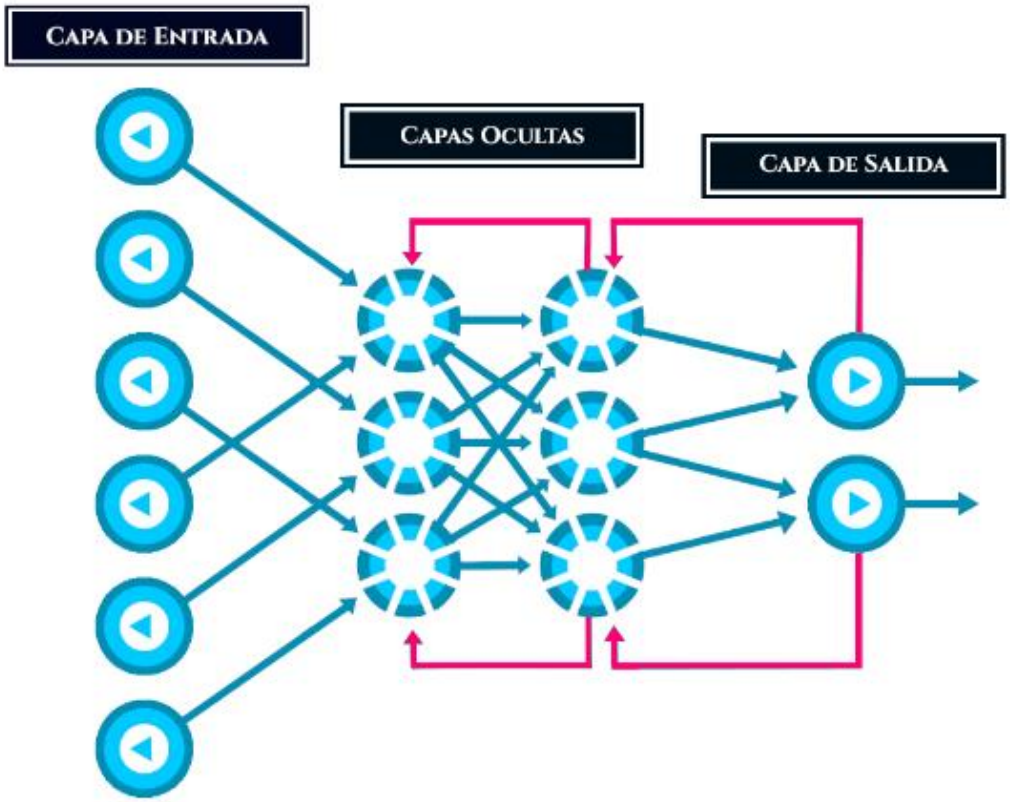


Figura 15. Estructura de una red LSTM[28]

c. Random Forest Classifier (Bosque aleatorio)

Random Forest es un reconocido algoritmo de aprendizaje automático que forma parte de la familia de algoritmos de ensamblaje. Esta técnica de ensamblaje emplea varios modelos para abordar un solo problema, combinando sus resultados para generar una solución que sea más precisa y estable. Los bosques aleatorios son un algoritmo popular en el aprendizaje automático supervisado[29].

- Se utilizan en aprendizaje automático supervisado, donde se dispone de una variable objetivo-etiquetada.
- Pueden aplicarse tanto para resolver problemas de regresión (donde la variable objetivo es numérica) como de clasificación (donde la variable objetivo es categórica).

- Los bosques aleatorios emplean un método de conjunto, lo que significa que combinan las predicciones de múltiples modelos.

- Cada uno de estos modelos individuales en el conjunto de los bosques aleatorios es un árbol de decisión.

d. Estructura del algoritmo

El algoritmo Random Forest amplía el uso del algoritmo de Árbol de Decisión, una herramienta simple y poderosa para problemas de clasificación y regresión. En la Figura 16 se observa un árbol de decisión basado en una estructura similar a un diagrama de flujo en forma de árbol, donde cada nodo interno representa una característica o atributo, cada rama simboliza una decisión o regla, y cada nodo hoja corresponde a una etiqueta de clase o un valor de salida[29].

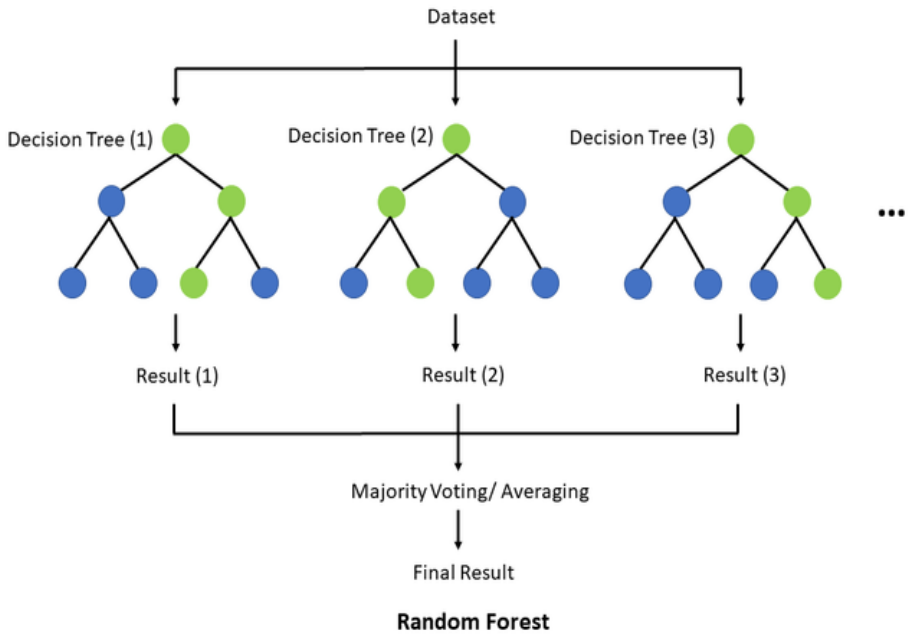


Figura 16. Estructura del algoritmo Random Forest[29]

La premisa central del algoritmo Random Forest es generar múltiples árboles de decisión y combinar sus resultados mediante el voto mayoritario o el valor promedio. Este método disminuye la varianza y el sesgo del modelo, mejorando así la generalización y robustez de las predicciones[29].

Un aspecto fundamental de Random Forest es la utilización de subconjuntos aleatorios de los datos y de las características predictivas para entrenar cada árbol. Esta técnica incrementa la diversidad e independencia de los árboles, lo que ayuda a prevenir el sobreajuste y la correlación[29].

Random Forest ofrece varias ventajas en comparación con otros algoritmos de aprendizaje automático, como[29]:

- Alta precisión y rendimiento, especialmente en conjuntos de datos grandes y complejos.
- Reducción de la varianza y el sobreajuste gracias a la combinación de múltiples modelos.
- Gran estabilidad debido al uso de árboles de decisión.
- Versatilidad y aplicabilidad elevada, ya que admite tanto tareas de clasificación como de regresión.
- Eficiencia en la selección automática de características.
- Facilidad en el ajuste y optimización del modelo.

No obstante, Random Forest también tiene algunas desventajas, tales como[29]:

- Alto costo computacional y de memoria, especialmente para conjuntos de datos grandes y complejos.
- Alta dependencia y correlación, en particular con conjuntos de datos pequeños y similares.
- Baja transparencia al intentar comprender y explicar el modelo.

Random Forest se aplica extensamente en diversas áreas, incluyendo[29]:

- Diagnóstico y pronóstico médico.
- Calificación de créditos y detección de fraude.

- Reconocimiento de imágenes y voz.
- Procesamiento del lenguaje natural y análisis de sentimiento.
- Pronóstico climático y meteorológico.
- Modelado energético y medioambiental.
- Análisis social y de redes.

e. Cómo funciona la clasificación en bosques aleatorios

Imagine que tiene un problema complejo y reúne a un grupo de expertos de distintos campos para que den sus opiniones. Cada experto aporta su perspectiva basada en su conocimiento y experiencia. Luego, estos expertos votan para llegar a una decisión final[30].

En la clasificación con bosques aleatorios, se crean múltiples árboles de decisión a partir de diferentes subconjuntos aleatorios de datos y características. Cada árbol de decisión actúa como un experto, ofreciendo su opinión sobre cómo clasificar los datos. Las predicciones se generan tomando la predicción de cada árbol de decisión y eligiendo el resultado más popular. (Para la regresión, las predicciones se promedian).

En el siguiente Figura 17 se observa el diagrama, tenemos un bosque aleatorio con n árboles de decisión y mostramos los primeros 5, junto con sus predicciones (ya sea "Perro" o "Gato"). Cada árbol está expuesto a diferentes características y muestras del conjunto de datos original, lo que hace que cada uno pueda ser diferente. Cada árbol realiza una predicción. Al observar los primeros 5 árboles, vemos que 4 de 5 predijeron que la muestra era un gato. Los círculos verdes indican un camino hipotético que tomó cada árbol para llegar a su decisión. El bosque aleatorio contaría las predicciones de los árboles de decisión para "Gato" y "Perro", y elegiría la predicción más popular.

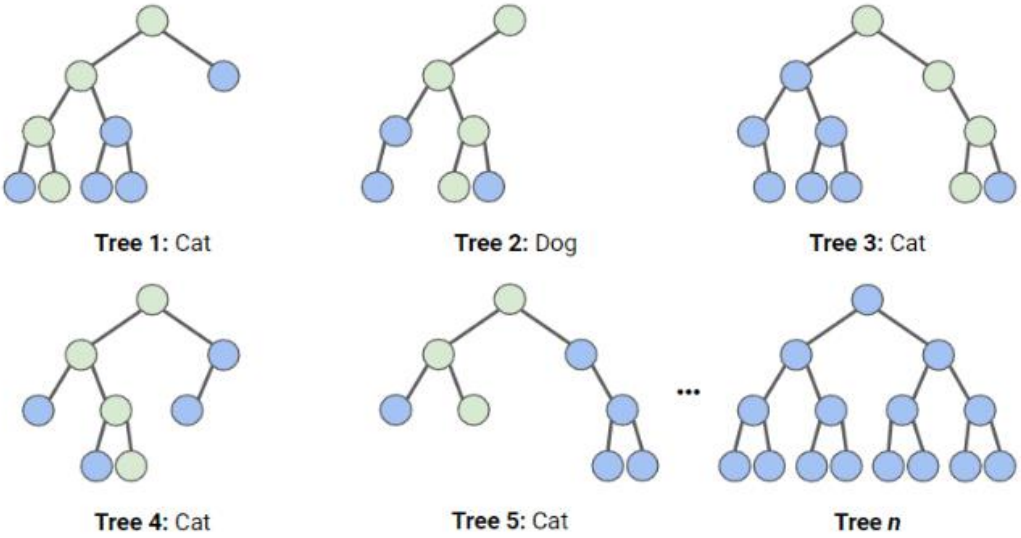


Figura 17. Ejemplo de RandomForestClassifier[30]

En la siguiente Tabla 6 se puede observar una comparativa entre los algoritmos de inteligencia artificial Redes LSTM y RandomForestClassifier.

Tabla 6. Comparación entre los algoritmos de inteligencia artificial [28] [29]

Característica	Redes LSTM	RandomForestClassifier
Tipo de Algoritmo	Red neuronal recurrente (RNN)	Algoritmo de ensamblaje basado en árboles de decisión
Tipo de Aprendizaje	Aprendizaje supervisado, especialmente en tareas de secuencias y datos temporales.	Aprendizaje supervisado para clasificación y regresión en datos tabulares.
Estructura de los Datos	Datos secuenciales o temporales, como texto o series temporales.	Datos estructurados en forma de tablas, con características independientes.
Ejemplos de Aplicación	Traducción automática, reconocimiento de voz, predicción de series temporales, análisis de sentimiento.	Clasificación de imágenes, detección de fraudes, diagnóstico médico, análisis de datos financieros.
Arquitectura	Redes neuronales recurrentes con capas LSTM, diseñadas para capturar dependencias a largo plazo.	Conjunto de árboles de decisión independientes (bosque) que votan para la clasificación final.
Transformación de Datos	Requiere preprocesamiento de datos secuenciales, como tokenización y normalización.	No requiere transformación compleja; los datos se utilizan directamente después de la limpieza.
Propiedades Clave	Manejo efectivo de dependencias a largo plazo y datos secuenciales, capacidad para recordar contextos pasados.	Promueve robustez y precisión mediante la agregación de múltiples árboles, maneja bien datos ruidosos.
Método de Predicción	Predicción basada en el análisis de secuencias y patrones temporales aprendidos durante el entrenamiento.	Predicción basada en el consenso de las decisiones de múltiples árboles de decisión.

Adaptabilidad	Adaptable a diferentes tipos de datos secuenciales y temporales mediante ajuste de hiperparámetros.	Adaptable a diferentes problemas de clasificación mediante ajuste del número de árboles y profundidad.
Complejidad Computacional	Alta, debido al entrenamiento de redes neuronales con parámetros complejos.	Moderada, basada en la construcción de múltiples árboles de decisión.
Tiempo de Entrenamiento	Generalmente más largo debido al procesamiento de datos secuenciales y ajustes de hiperparámetros.	Relativamente más rápido, ya que entrena múltiples árboles en paralelo.
Interpretabilidad	Bajo nivel de interpretabilidad; modelos complejos y difíciles de interpretar.	Alta interpretabilidad; se puede visualizar la importancia de características y decisiones de árboles individuales.
Manejo de Datos No Estructurados	Muy eficaz en el manejo de datos no estructurados como texto o secuencias de audio.	Principalmente diseñado para datos estructurados, como tablas de datos.
Robustez a Ruido	Puede ser sensible al ruido en datos secuenciales, requiere cuidado en la preprocesamiento.	Generalmente robusto al ruido, ya que el promedio de decisiones de múltiples árboles ayuda a reducir el impacto del ruido.
Capacidad de Generalización	Buena para generalizar patrones en datos secuenciales complejos.	Buena para generalizar en datos tabulares, pero puede necesitar ajuste de hiperparámetros para optimización.
Ventajas	Excelente para modelar dependencias a largo plazo y manejar datos secuenciales complejos.	Robusto al ruido y a los datos faltantes, proporciona alta precisión y es relativamente fácil de implementar.
Desventajas	Entrenamiento computacionalmente costoso y largo, baja interpretabilidad.	No es ideal para datos secuenciales, puede ser menos efectivo en problemas donde las características son altamente interdependientes.

1.3.16 Base de datos del sistema para entrenamiento de IA

Una base de datos, conocida también como banco de datos, es un conjunto organizado de información dentro de un mismo contexto, diseñado para su posterior recuperación, análisis y transmisión de manera sistemática. Este concepto abarca desde colecciones en bibliotecas hasta extensos conjuntos de datos de usuarios administrados por empresas de telecomunicaciones. Las bases de datos son una respuesta a la necesidad humana de preservar información contra el tiempo y el deterioro, permitiendo su acceso futuro. La evolución de la electrónica y la computación ha sido crucial al proporcionar medios digitales para almacenar grandes volúmenes de datos en espacios físicos limitados, aprovechando la conversión de la información en señales eléctricas o magnéticas[31].

Las bases de datos de inteligencia artificial (IA) son una innovación reciente enfocada en mejorar los modelos de aprendizaje automático y profundo, acelerando su entrenamiento de manera eficiente. Estas bases de datos aprovechan tecnologías avanzadas de IA para ofrecer servicios que añaden valor. Diseñadas para optimizar recursos informáticos y de bases de datos, permiten la ingestión, exploración, análisis y visualización simultánea de datos complejos y de rápida evolución en fracciones de segundo. Además, las bases de datos de IA facilitan capacidades como el análisis y la búsqueda exhaustiva de texto, mejorando así la capacidad de procesamiento y comprensión de grandes volúmenes de información textual[32].

a. Cuando usar una base de datos de IA

Decidir cuándo utilizar una base de datos de inteligencia artificial implica considerar varios factores fundamentales. Estas bases de datos son esenciales para entrenar y validar modelos de IA en diversas aplicaciones, desde reconocimiento de imágenes hasta análisis de datos complejos. Elegir la base de datos adecuada garantiza que los datos sean representativos y suficientemente grandes para entrenar modelos con precisión, acelerando así el desarrollo del modelo al proporcionar conjuntos de datos estructurados y limpios que cumplen con los requisitos del proyecto de IA. La base de datos de IA se puedes usar[32]:

- Para aplicaciones que necesitan procesamiento más rápido y una mayor velocidad de ingesta de datos.
- El logro de metas comerciales depende de los esfuerzos de aprendizaje automático y aprendizaje profundo de la organización.
- Desarrollar una plataforma de canalización de modelos que permita alojar modelos basados en ciencia de datos.
- Resolver el volumen, la velocidad y los desafíos complejos de gestión y gobierno de datos relacionados con la capacitación de modelos de aprendizaje automático y aprendizaje profundo para mejorar la eficiencia y ahorrar recursos.

b. Aplicaciones prácticas para bases de datos de inteligencia artificial

Las aplicaciones prácticas para bases de datos de inteligencia artificial están transformando diversos sectores, ofreciendo soluciones avanzadas para analizar grandes volúmenes de datos de manera eficiente. Estas bases de datos no solo facilitan el almacenamiento y la recuperación de información, sino que también potencian el desarrollo de modelos de aprendizaje automático y aprendizaje profundo. Las aplicaciones son las siguientes[32]:

- Identificación de objetos y análisis textual.
- Reconocimiento de voz.
- Clasificación de contenidos en plataformas sociales.
- Revisión visual.
- Procesamiento del lenguaje natural.

c. Base de datos utilizadas en el entrenamiento de IA

Las bases de datos utilizadas en el entrenamiento de algoritmos de inteligencia artificial se clasifican en varios tipos fundamentales[33]:

- **Tipo de datos:** Incluye datos estructurados, como aquellos organizados en tablas con esquemas definidos, como bases de datos relacionales; datos no estructurados, como imágenes, textos o videos que carecen de un formato predefinido; y datos semiestructurados, que tienen cierta organización flexible como en formatos JSON o XML.
- **Dominio o aplicación:** Estas bases de datos se utilizan en diversos dominios, como visión por computadora para el reconocimiento visual con conjuntos de imágenes como CIFAR-10 o ImageNet; procesamiento del lenguaje natural con corpora textuales como Wikipedia o Gutenberg Project; reconocimiento de voz con datos de audio como LibriSpeech o VoxCeleb; sistemas de recomendación que utilizan interacciones usuario-ítem como en MovieLens o Amazon Reviews; y en ciencia de datos con conjuntos estadísticos disponibles en repositorios como UCI Machine Learning Repository o Kaggle Datasets.

- **Disponibilidad:** Pueden ser públicas, accesibles para investigación y uso público como Kaggle Datasets o OpenML; privadas, utilizadas internamente por empresas y organizaciones sin acceso público; o open data, proporcionadas por gobiernos y entidades públicas en plataformas como Datos.gov o European Data Portal.

- **Fuente de los datos:** Los datos pueden ser generados automáticamente por sensores y dispositivos IoT como registros de temperatura o GPS; generados por usuarios en redes sociales o encuestas online; o sintéticos, creados artificialmente para simulaciones y pruebas en entornos virtuales.

- **Formato y estructura:** Las bases de datos pueden ser tabulares, organizadas en filas y columnas como en bases de datos SQL; textuales, como documentos y textos sin estructura fija como PDFs o documentos web; multimedia, que incluyen imágenes, videos y audio como en YouTube Dataset o Flickr30K; o secuenciales, datos ordenados temporalmente como series temporales o registros de eventos.

- **Tamaño:** Varían desde conjuntos pequeños que pueden manejarse en una sola máquina como datasets de prueba estándar, hasta grandes volúmenes de datos masivos que requieren procesamiento distribuido como conjuntos de datos de redes sociales o datos climáticos globales.

- **Calidad de los datos:** Pueden ser limpios, preprocesados y libres de errores para análisis directo; o sucios, con errores y anomalías que requieren limpieza y corrección antes de su utilización en aplicaciones de inteligencia artificial.

1.4 Objetivos

1.4.1 Objetivo general

Implementar un sistema de detección automática de reconocimiento de engaños mediante análisis de la voz.

1.4.2 Objetivos específicos

- Analizar las características de las señales de voz en situaciones de extorsiones y delictivas.
- Determinar los algoritmos de análisis de señales voz para detección de extorsiones.
- Aplicar un algoritmo de inteligencia artificial para la detección de extorsiones.
- Verificar el funcionamiento del sistema de reconocimiento de extorsiones.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

En el contexto de la investigación orientada al desarrollo de este proyecto, se emplearon herramientas especializadas para la captura y análisis de datos. La captura de las señales de voz se realizó utilizando un micrófono integrado en la propia computadora donde se desarrolló la interfaz y la programación del proyecto, mientras que el procesamiento de estos datos se llevó a cabo en una computadora equipada con un procesador AMD Ryzen 5, utilizando el popular lenguaje de programación Python. Como núcleo del sistema de inteligencia artificial y para su posterior entrenamiento, se implementó el algoritmo RandomForestClassifier, reconocido por su capacidad para realizar clasificaciones precisas en conjuntos de datos complejos como el de las señales de voz. Este enfoque permitió integrar tecnologías avanzadas de captura y análisis de datos, garantizando la precisión y eficiencia necesarias para la detección automática de patrones asociados a comportamientos de extorsión en las señales de voz registradas. En la Tabla 7 se encuentra detallada la descripción de los componentes utilizados en el hardware del sistema, mientras que en la Tabla 8 se presenta el software específicamente seleccionado para el proyecto.

Tabla 7. Características del hardware seleccionado

Material	Características	Uso en el Proyecto
Computadora con procesador AMD Ryzen 5	Equipada con un potente procesador AMD Ryzen 5, que ofrece alto rendimiento y capacidad de procesamiento de datos.	La computadora se utiliza para ejecutar algoritmos de análisis avanzado sobre las señales de voz capturadas, contribuyendo al proceso de detección automática de extorsiones.
Micrófono integrado en el computador	Incluye un micrófono integrado de alta calidad que permite la captura precisa de señales de voz directamente desde la computadora.	El micrófono es fundamental para adquirir las muestras de voz necesarias para el análisis de patrones característicos de extorsión dentro del sistema automatizado.

Tabla 8. Características del software seleccionado

Software	Características	Uso en el Proyecto
Python	Este lenguaje de programación es conocido por su versatilidad y potencia, ampliamente utilizado en el análisis de datos y Machine Learning.	Python se emplea para desarrollar algoritmos que permiten el procesamiento y análisis de las señales de voz en tiempo real, fundamental para la detección automática de extorsiones.
Inteligencia Artificial RandomForestClassifier	Este algoritmo de clasificación se basa en árboles de decisión aleatorios, reconocido por su robustez y eficiencia en la identificación de patrones complejos.	RandomForestClassifier es utilizado específicamente para entrenar modelos que identifican características distintivas en las señales de voz relacionadas con extorsiones, mejorando la precisión del sistema de detección automatizada.

2.2 Métodos

2.2.1 Modalidad de la investigación

El proyecto de investigación es de investigación aplicada, ya que su objetivo principal fue utilizar los conocimientos teóricos y prácticos adquiridos durante la formación académica para diseñar e implementar un sistema de detección automática de extorsiones. Este sistema se basó en el análisis de señales de voz y conversión de la voz a lenguaje natural, con el fin de identificar patrones y características que pudieran indicar la presencia de extorsiones telefónicas. Al combinar el conocimiento académico con la aplicación práctica, se buscaba desarrollar una herramienta que contribuyera a la prevención y lucha contra este tipo de delitos.

Para la recolección de información se utilizó la investigación bibliográfica, consultando fuentes en las cuales se pudo encontrar información relevante para el desarrollo del proyecto. Se analizó información de libros, tesis y otras fuentes relacionadas con el tema. Además, se recopiló información de páginas web que proporcionaban detalles sobre el funcionamiento de la detección de engaños mediante el análisis de la voz.

Adicionalmente, se llevó a cabo una investigación de campo para realizar pruebas de funcionamiento del programa. Durante esta fase, se analizaron señales de voz donde

se presentaron frases extorsivas y no extorsivas para crear la base de datos de entrenamiento y validación, así como la verificación de funcionamiento del sistema.

2.2.2 Recolección de información

Para llevar a cabo la recopilación de información necesaria en este proyecto, se utilizaron diversas fuentes como libros, revistas científicas, recursos en línea y proyectos previamente desarrollados en el campo de detección de extorsiones. Además, se tuvieron en cuenta guías prácticas y manuales de construcción que proporcionaron orientación específica en la temática abordada. Se puso especial énfasis en la consulta de bases de datos confiables que respaldaron el desarrollo de proyectos de este tipo. Es importante destacar que una de las fuentes clave de datos fue el propio sistema automático de detección de extorsiones mediante el análisis de señales de la voz, que proporcionó información valiosa para el estudio y análisis en esta investigación.

2.2.3 Procesamiento y análisis de datos

Para el procesamiento y análisis de datos, se aplicaron las siguientes operaciones:

- Se analizaron y dedujeron las características de las señales de la voz.
- Se identificaron los patrones de engaño en las señales de la voz.
- Se investigó y analizó la implementación de un programa que permitiera detectar a través de las señales de la voz.
- Se enfocó en la solución de la propuesta mediante el programa a desarrollarse.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Desarrollo de la propuesta

La justificación de la propuesta se cimienta en la explicación de las decisiones adoptadas para la implementación del proyecto, basándose en el esquema general de arquitectura del sistema, el cual divide el proceso en etapas. Desde este enfoque, se detallan los componentes y tecnologías esenciales necesarios para desarrollar un sistema automatizado de detección de extorsiones mediante el análisis de señales de voz. Asimismo, se establece una conexión entre los objetivos planteados y los resultados obtenidos durante la fase de implementación. Para el diseño del sistema se ha plateado un esquema que establece las etapas esenciales a desarrollar para alcanzar con éxito los objetivos del proyecto. Se ha identificado tres etapas fundamentales: la captura y preprocesamiento de las señales de voz, el análisis y extracción de características relevantes, y finalmente, la implementación de algoritmos de detección y clasificación extorsiones. En la siguiente Figura 18 se observa la arquitectura del sistema.

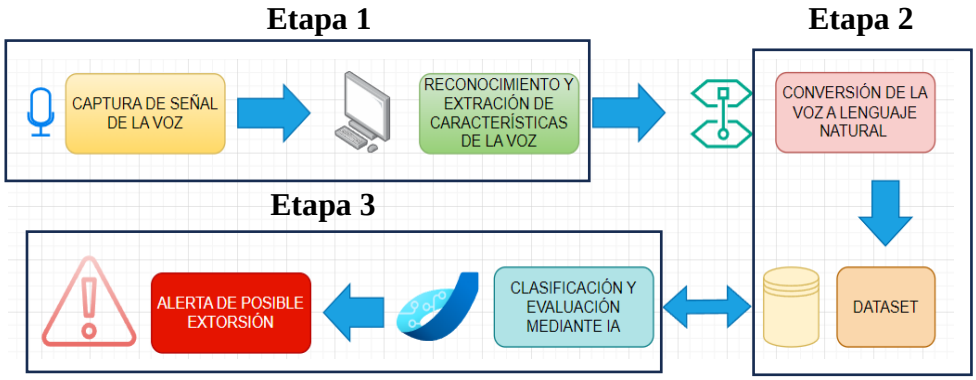


Figura 18. Esquema general del sistema

El proceso de detección de intentos de extorsión mediante análisis de voz es un sistema integral que utiliza tecnologías de reconocimiento de voz e inteligencia artificial. Este sistema consta de varias etapas clave: captura de la señal de voz, reconocimiento y extracción de características, conversión de voz a texto, almacenamiento en un dataset,

clasificación y evaluación con inteligencia artificial, y generación de alertas de posible extorsión. Cada etapa es esencial para garantizar la eficacia y precisión del sistema.

Captura de la Señal de Voz: En esta etapa inicial, el sistema registra la señal de voz del usuario utilizando un micrófono u otro dispositivo de captura de audio. La calidad de esta grabación es crucial para las siguientes fases del proceso.

Reconocimiento y Extracción de Características de la Voz: Tras la captura de la señal, se realiza el reconocimiento y la extracción de características.

Conversión de Voz a Lenguaje Natural: En esta fase, la señal de voz se convierte en texto utilizando técnicas de procesamiento del lenguaje natural (PLN). Esta conversión facilita el análisis del contenido del discurso en las etapas posteriores.

Almacenamiento en Dataset: El texto convertido se almacena en un dataset estructurado, esencial para el entrenamiento y evaluación de los modelos de inteligencia artificial utilizados para la clasificación y detección de extorsiones.

Clasificación y Evaluación Mediante Inteligencia Artificial: Utilizando el dataset almacenado, se entrenan modelos de inteligencia artificial para clasificar y evaluar las grabaciones de voz. Estos modelos están diseñados para detectar patrones y características que puedan indicar un intento de extorsión.

Alerta de Posible Extorsión: Finalmente, si el sistema de inteligencia artificial detecta indicios de extorsión en una grabación de voz, se genera una alerta. Esta alerta puede ser enviada a las autoridades pertinentes o al usuario afectado, permitiendo una respuesta rápida y adecuada.

Este proceso asegura un análisis exhaustivo de las señales de voz, permitiendo la detección temprana de posibles extorsiones y mejorando la seguridad de los usuarios como se detalla el proceso en siguiente diagrama de la Figura 19.

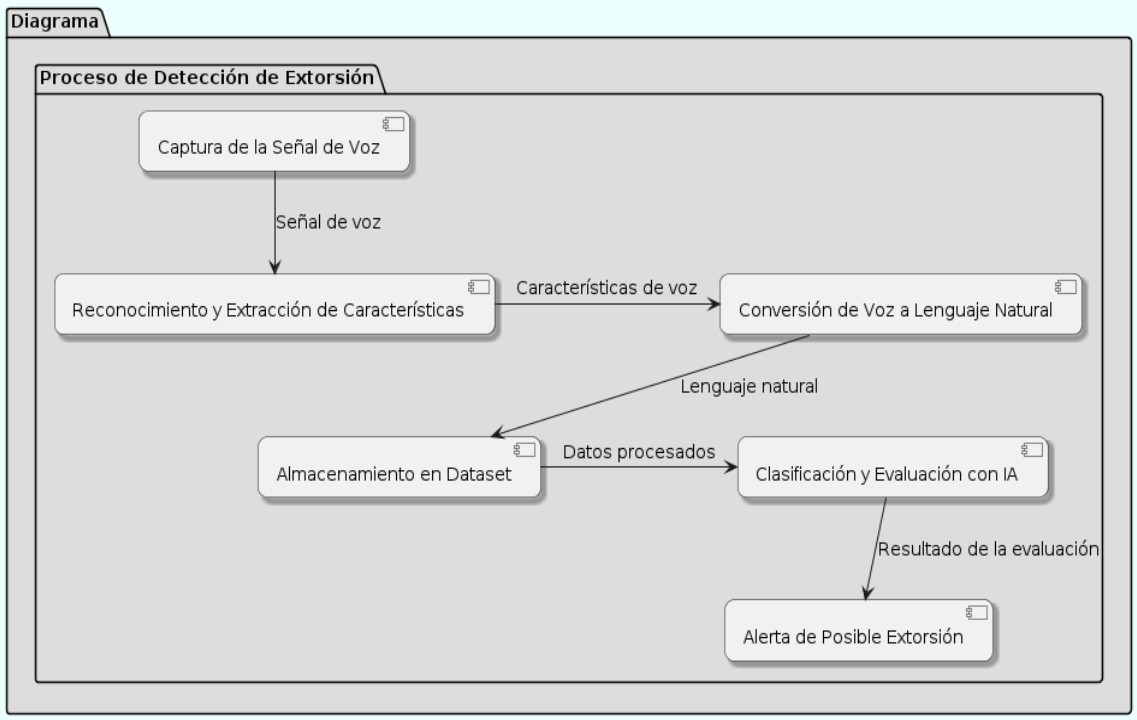


Figura 19. Proceso de detección de extorsiones

3.1.1 Selección de Hardware

Se realizó una evaluación los componentes de hardware utilizados en el desarrollo del proyecto. Esta selección garantizó la efectividad y precisión del sistema, incluyendo tanto los dispositivos de captura de audio como los equipos necesarios para el procesamiento y análisis de las señales de voz. Se analizaron las características, ventajas y limitaciones de cada componente de hardware seleccionado, justificando su elección según los requisitos específicos del proyecto.

a. AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz

El procesador AMD Ryzen 5 3500U con Gráficos Radeon Vega Mobile es una unidad de procesamiento diseñada especialmente para dispositivos móviles como laptops y notebooks. Basado en la arquitectura Zen+ de AMD, este procesador ofrece un rendimiento mejorado y una mayor eficiencia energética en comparación con sus predecesores[34].

Con cuatro núcleos de CPU y ocho hilos de procesamiento, el Ryzen 5 3500U permite realizar múltiples tareas simultáneamente de manera eficiente, lo que lo convierte en

una opción adecuada para entornos móviles. Su frecuencia base de 2.10 GHz y velocidad máxima de hasta 3.70 GHz en modo de ráfaga garantizan un equilibrio óptimo entre rendimiento y consumo de energía[34].

La potencia de la CPU, el Ryzen 5 3500U cuenta con gráficos integrados Radeon Vega, que ofrecen un rendimiento gráfico sólido para aplicaciones multimedia y juegos ligeros. Esta característica lo convierte en una opción versátil para usuarios que buscan tanto productividad como entretenimiento en sus dispositivos móviles[34].

b. Procesador Intel® Core™ i7-12700

El procesador Intel® Core™ i7-12700 es una unidad central de procesamiento (CPU) diseñada para sistemas de escritorio, basada en la arquitectura Alder Lake de Intel y fabricada con tecnología de 10 nanómetros SuperFin. Con 12 núcleos en total, incluyendo 8 de alto rendimiento y 4 de eficiencia energética, ofrece un equilibrio entre potencia y consumo energético. Compatible con Hyper-Threading para hasta 20 hilos de ejecución, garantiza una buena multitarea. Con una frecuencia base de 2.2 GHz y turbo de hasta 4.7 GHz, responde rápidamente a las demandas de rendimiento. Además, incluye gráficos integrados Intel UHD Graphics 770 y tiene un TDP de 65 vatios, lo que lo hace adecuado para sistemas de escritorio con requisitos de energía moderados[35].

En la Tabla 9 siguiente se describe las características a de los procesadores a utilizar

Tabla 9. Comparación de procesadores [34] [35]

Característica	**AMD Ryzen 5 3500U**	**Intel® Core™ i7-12700**
Arquitectura	Zen+	Alder Lake
Núcleos de CPU	4	12
Hilos de ejecución	8	20
Frecuencia base	2.10 GHz	2.20 GHz (Performance-core)
Frecuencia turbo máxima	Hasta 3.70 GHz	Hasta 4.90 GHz
Gráficos integrados	Radeon Vega	Intel UHD Graphics 770
TDP (potencia de diseño térmico)	15 W	65 W
Tecnología de fabricación	12 nm GlobalFoundries	10 nm SuperFin

3.1.2 Selección de Software

Se presentó una evaluación detallada de las herramientas de software empleadas en el desarrollo del proyecto. Esta selección garantiza la efectividad y precisión del sistema, abarcando tanto el entorno de programación como las bibliotecas y plataformas utilizadas para el procesamiento de datos, el entrenamiento del modelo y el despliegue del sistema en un entorno real. Se discutieron las características, ventajas y limitaciones de cada herramienta seleccionada, justificando su elección en función de los requisitos específicos del proyecto.

En mi investigación, he decidido utilizar Python para la programación en lugar de MATLAB. Aunque ambos son robustos para el análisis y procesamiento de datos, Python ofrece ventajas únicas sobre MATLAB. Mientras que MATLAB es conocido por su facilidad de uso y su amplia gama de funciones especializadas en procesamiento numérico y simulación, Python destaca por su flexibilidad, comunidad activa de usuarios y desarrolladores, y un vasto conjunto de bibliotecas de código abierto como NumPy, SciPy y pandas. Estas bibliotecas proporcionan herramientas avanzadas para análisis de datos, aprendizaje automático, visualización y más, lo que permite un enfoque integrado y adaptable en la programación de mi investigación. Por estas razones, he elegido Python como la herramienta principal para asegurar un desarrollo eficiente y efectivo de mi tesis. En la siguiente Tabla 10 se la compasión de software donde se puede desarrollar el sistema[36].

Tabla 10. Cuadro comparativa entre Matlab y Python [36] [37]

Característica	**Matlab**	**Python**
Sintaxis	Utiliza una sintaxis específica	Ofrece una sintaxis simple y legible
Costo	Requiere una licencia comercial	Es de código abierto y gratuito
Comunidad	Aunque más pequeña, está especializada	Cuenta con una gran comunidad y amplio soporte
Flexibilidad	Especializado en computación numérica	Es multipropósito, adecuado para diversas tareas
Bibliotecas	Dispone de bibliotecas específicas	Ofrece una amplia variedad de bibliotecas disponibles
Integración	Se integra con Simulink y otros	Se integra con numerosas herramientas y frameworks
Aprendizaje	Tiene una curva de aprendizaje pronunciada	Ofrece una curva de aprendizaje más suave
Rendimiento	Ofrece un rendimiento rápido para cálculos numéricos	Proporciona un rendimiento adecuado para la mayoría de las aplicaciones

Popularidad	Es popular en entornos académicos e ingenieriles	Es popular en una amplia gama de campos

3.2 Implementacion del sistema

La arquitectura del sistema fue concebida para identificar posibles casos de extorsión mediante el procesamiento avanzado de señales de voz y el uso de inteligencia artificial. Inicialmente, se captura la señal de voz del usuario a través de dispositivos de grabación. Esta señal es procesada para reconocer y extraer características clave, como el tono y el timbre. Posteriormente, la voz se convierte en texto utilizando técnicas de procesamiento del lenguaje natural (NLP), lo que facilita su análisis. Este texto, junto con otros datos recopilados en un amplio conjunto de datos (dataset), se utiliza para entrenar el modelo de inteligencia artificial que clasifica y evalúa la información, buscando patrones que indicaran una posible extorsión. Finalmente, el sistema genera alertas en tiempo real cuando se detectaban amenazas potenciales, notificando a los usuarios o a las autoridades pertinentes.

3.2.1 Captura de la señal de la voz

El proceso de captura de la señal de la voz fue fundamental para el funcionamiento del sistema. Inicialmente, un micrófono se utiliza para registrar las palabras habladas por el usuario en este caso fue mediante una computadora, transformando las vibraciones sonoras en señales eléctricas. Esta etapa de captura de audio sirvió es la entrada principal del sistema, proporcionando los datos de audio que serían procesados y analizados en las etapas posteriores.

Una vez capturada la señal de audio, se realizó el preprocesamiento para preparar los datos para el análisis. Esto incluyó la conversión de la señal analógica en datos digitales, la normalización para mantener un nivel de amplitud constante y la eliminación de ruido para reducir las interferencias no deseadas. Además, la señal se segmentó en tramos relevantes, aislando las partes del audio que contenían información significativa. Este preprocesamiento aseguró que el audio estuviera en un formato óptimo para su posterior análisis, permitiendo una detección precisa y

eficiente de extorsiones mediante algoritmos de reconocimiento de voz y aprendizaje automático.

3.2.2 Reconocimiento y extracción de características de la voz

La etapa de reconocimiento y extracción de características de la voz fue crucial después de la captura inicial de la señal de audio mediante un micrófono. Esta etapa se dividió en dos partes principales: el reconocimiento de voz y la extracción de características del audio.

- ***Reconocimiento de Voz***

Después de que el audio fue capturado y preprocesado, se utilizó programa Python realizado en la computadora para realizar el reconocimiento de voz. Este sistema convirtió la señal de audio digitalizada en texto, identificando y transcribiendo las palabras habladas por el usuario. Esta conversión fue llevada a cabo por un algoritmo de reconocimiento de voz que pudo distinguir las palabras basándose en patrones predefinidos de lenguaje. Se emplearon herramientas y bibliotecas en Python, como SpeechRecognition, para este propósito. El resultado de esta etapa fue una transcripción textual precisa de las palabras habladas, lista para ser analizada.

- ***Extracción de Características***

Una vez que se obtuvo la transcripción textual del audio, se procedió a la extracción de características específicas del texto y del audio. Este proceso implicó identificar elementos clave como palabras y frases relevantes, estructuras gramaticales, entonaciones y otros aspectos lingüísticos que eran críticos para la detección de extorsiones. Estas características se utilizaron para enriquecer la información disponible para el análisis y comparación con la base de datos de frases conocidas.

3.2.3 Conversión de la voz a lenguaje natural

La etapa de conversión de la voz a lenguaje natural fue crucial para interpretar y procesar las palabras habladas por el usuario. Esta etapa siguió a la captura de la señal de audio mediante un micrófono y al reconocimiento y extracción de características de

la voz, integrando estos datos en un formato interpretable para el análisis automatizado.

Después de que la señal de audio fue capturada y preprocesada, y se reconocieron y extrajeron las características de la voz, se utilizó un programa en la computadora para convertir la voz en lenguaje natural. El programa implementado en Python empleó la biblioteca SpeechRecognition para capturar el audio y convertirlo en texto. Mediante el uso de un micrófono y ajustes para el ruido ambiental, el sistema pudo registrar y transcribir las palabras habladas. Esta conversión fue esencial para proporcionar una representación textual precisa de las palabras pronunciadas por el usuario.

3.2.4 Dataset

Para el desarrollo del dataset utilizado en el entrenamiento y comparación de datos dentro del sistema de detección de extorsiones, se llevaron a cabo cuatro etapas fundamentales:

- **Primera etapa:** Se creó una base de datos llamada "BASE DE DATOS EXTORSIONES TOTAL", la cual contiene aproximadamente 201 frases extorsivas, 320 frases del lenguaje natural, 320 palabras del lenguaje natural y 201 palabras extorsivas.

- **Segunda etapa:** Consistió en desarrollar otra base de datos llamada “BASE FRASES NATURALES Y EXTORSIVAS” que mezcla alrededor de 1150 frases, estas son extorsivas etiquetadas con “0” y frases del lenguaje natural etiquetadas con “1”.

- **Tercera etapa:** Se procedió a la limpieza de las bases de datos para eliminar frases duplicadas, así como signos de puntuación y tildes llamada “dataset_frases sin signos y tildes” conformada por 419 frases extorsivas etiquetadas con “0” y del lenguaje natural etiquetadas con “1”. Este proceso fue fundamental para garantizar la integridad del archivo utilizado durante el entrenamiento.

- **Cuarta etapa:** Se obtuvo una nueva base de datos llamada “dataset_frases.xlsx” compuesta por aproximadamente 427 frases depuradas libres de signos de puntuación, tildes y frases repetidas y otras agregadas con signos y comas para ver cómo se desenvuelve el programa en su entrenamiento. Esta última base de datos fue seleccionada para el entrenamiento del programa.

En la siguiente la Figura 20 se puede observar las etapas del desarrollo del dataset que va a ser utilizado en el entrenamiento del sistema.

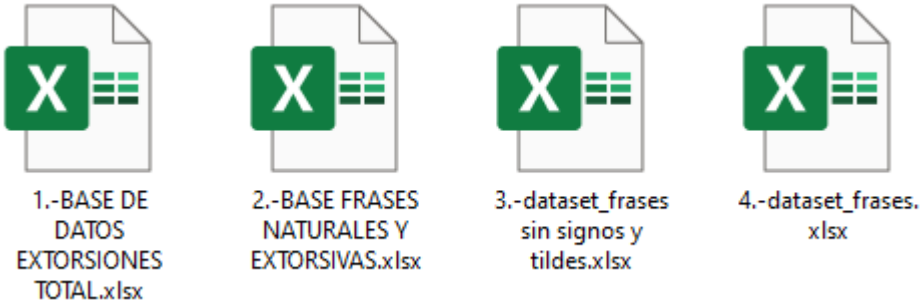


Figura 20. Desarrollo del dataset

Primero, se realizaron las etapas mencionadas anteriormente para el desarrollo de la base de datos, permitiendo su incorporación al sistema. Una vez que se integró la última base de datos, se creó una nueva base de datos adecuada para el entrenamiento del sistema.

El dataset se creó etiquetando frases específicas. Las frases que indicaban extorsión se etiquetaron como 0, mientras que las frases de lenguaje natural se etiquetaron como 1. Ejemplos de estas frases incluyen desde "Dame todo el dinero que tienes" (0) hasta "Me gusta el yogur natural" (1).

Después de etiquetar las frases, se combinaron en listas separadas por etiquetas y se formó un DataFrame utilizando Pandas en Python. Este DataFrame se exportó como un archivo CSV llamado 'dataset_palabras.csv', que contiene todas las frases junto con sus respectivas etiquetas. Este archivo se utilizó para entrenar y evaluar el modelo de detección automática de extorsiones.

Se consolida y refina estos archivos en un único archivo CSV llamado 'dataset_palabras.csv'. Este archivo se utilizó como la base de datos principal para

entrenar y evaluar el modelo de detección automática, garantizando que el sistema pudiera distinguir de manera precisa entre señales de voz indicativas de extorsión y conversaciones normales del día a día.

Finalmente, se obtuvo un dataset completo para entrenar el sistema, que incluyó la recopilación de frases representativas de extorsión y lenguaje natural. Se investigaron diversas fuentes en línea para identificar expresiones típicas usadas por extorsionadores y frases comunes en conversaciones diarias. Estos datos fueron organizados y depurados en archivos Excel, y luego se consolidaron en un archivo CSV optimizado para entrenar el modelo. Este dataset fue fundamental para asegurar que el sistema aprendiera eficazmente a distinguir entre situaciones potencialmente riesgosas y conversaciones normales, garantizando su precisión en la detección automática de extorsiones. En la Figura 21. Estructura del dataset llamado "dataset_frases", se puede observar la estructura de la base de datos, la cual consta de frases extorsivas etiquetadas con 0 y frases de lenguaje natural etiquetadas con 1. El dataset contiene un total de (indicar el número de frases) frases.

608	Cada minuto que pasa agrava tu situacion	0
609	No hay margen para errores actua ahora	0
610	Esto es un robo	0
611	Este es tu ultimo aviso antes de que tomemos medidas serias	0
612	No te atrevas a ignorar esta advertencia te estamos vigilando	0
613	Confirma la recepcion de este mensaje si quieres evitar problemas	0
614	Este mensaje es serio y tus acciones tendran consecuencias	0
615	Es imperativo que sigas nuestras instrucciones sin hacer preguntas	0
616	Nuestra paciencia se esta agotando actua ahora o lo lamentaras	0
617	Esto no es un juego la seguridad de tu familia esta en juego	0
618	Asegurate de hacer el pago antes de la medianoche si valoras tu bienestar	0
619	No intentes ser un heroe solo sigue lo que te decimos	0
620	Cualquier intento de contactar a las autoridades empeorara las cosas	0
621	Hoy es un buen dia	1
622	Me gusta mucho leer libros	1
623	El sol esta brillando	1
624	Voy a dar un paseo	1
625	La musica es relajante	1
626	Me encanta cocinar	1
627	Vamos al parque	1
628	Estoy aprendiendo a nadar	1

Figura 21. Estructura del dataset llamado "dataset_frases"

3.2.5 Clasificación y evaluación mediante inteligencia artificial

La etapa de clasificación y evaluación utiliza el algoritmo RandomForestClassifier. Este sistema se diseñó para detectar de manera automática frases indicativas de extorsión dentro de señales de voz convertidas en texto.

El proceso comenzó con la captura de señales de voz mediante un micrófono y su transformación en texto utilizando técnicas avanzadas de reconocimiento automático de voz. Después se realizó una comparación con las frases del dataset donde cada frase fue etiquetada manualmente como perteneciente a dos categorías principales: extorsión (0) y lenguaje natural (1). Este dataset fue fundamental, ya que proporcionó los datos necesarios para entrenar y evaluar el modelo de clasificación.

Para preparar los datos, se utilizó CountVectorizer, una técnica de vectorización que convierte cada frase de texto en un vector numérico representativo. Esto permitió que el algoritmo RandomForestClassifier pudiera trabajar con datos estructurados adecuadamente. El dataset se dividió en conjuntos de entrenamiento y prueba utilizando train_test_split para asegurar que el modelo fuera evaluado de manera imparcial y precisa.

El método RandomForestClassifier fue elegido por su capacidad para manejar eficazmente grandes cantidades de datos y evitar el sobreajuste gracias a la combinación de múltiples árboles de decisión durante el entrenamiento. Se ajusta el modelo con 100 estimadores y se establece una semilla aleatoria para garantizar la reproducibilidad de los resultados. Posteriormente, se evalúa el desempeño del modelo utilizando métricas como la precisión (accuracy_score), asegurándome de que el sistema fuera capaz de distinguir correctamente entre señales de extorsión y conversaciones cotidianas.

Tras el entrenamiento y la evaluación satisfactoria del modelo, se procede a guardar tanto el modelo RandomForestClassifier entrenado como el vectorizador CountVectorizer utilizando la biblioteca joblib. Esto permite implementar el sistema de detección en tiempo real, donde nuevas señales de voz podrían ser procesadas y

clasificadas instantáneamente, alertando ante posibles amenazas de extorsión de manera eficiente y precisa.

La etapa de clasificación y evaluación fue fundamental puesto que, a través del uso de técnicas avanzadas de procesamiento de señales de voz y aprendizaje automático, se obtuvo que cumple con los requisitos del sistema, y establece las bases para aplicaciones prácticas en la detección y prevención de delitos mediante el análisis de señales acústicas.

3.2.6 Alerta de posible extorsión

El proceso comienza con la captura de la señal de voz mediante un micrófono. El audio capturado se procesa mediante el reconocimiento automático de voz, transformándolo en texto que se comparaba con el dataset previamente desarrollado. Este dataset consistía en frases de extorsión etiquetadas con 0 y frases de lenguaje natural etiquetadas con 1. El código utiliza la biblioteca `speech_recognition` para convertir la voz en texto. Una vez capturado y reconocido el texto, el sistema emplea el modelo RandomForestClassifier previamente entrenado para predecir si la frase es una extorsión. La función `reconocer_voz()` ajustaba el micrófono para el ruido ambiental, escuchaba el audio y lo convertía en texto mediante el reconocimiento de voz de Google. El texto resultante se vectorizaba utilizando el `CountVectorizer` cargado anteriormente y se clasifica con el modelo RandomForestClassifier.

El sistema emite una alerta si la predicción indicaba una frase de extorsión (etiquetada con 0). Esta implementación permite la detección en tiempo real de posibles extorsiones, proporcionando una herramienta eficaz para identificar amenazas y tomar acciones preventivas de manera inmediata.

3.3 Desarrollo del sistema de detección de extorsiones

A continuación, se describe el proceso de desarrollo del sistema desde la captura de la señal de voz hasta la generación de alertas, explicando cada paso y su implementación: captura de la señal de voz, reconocimiento y extracción de características, conversión de voz a lenguaje natural, almacenamiento en un dataset, clasificación y evaluación

utilizando inteligencia artificial, y finalmente, la alerta de posible extorsión. En la siguiente Figura 22 se puede observar el desarrollo de la programación del sistema.

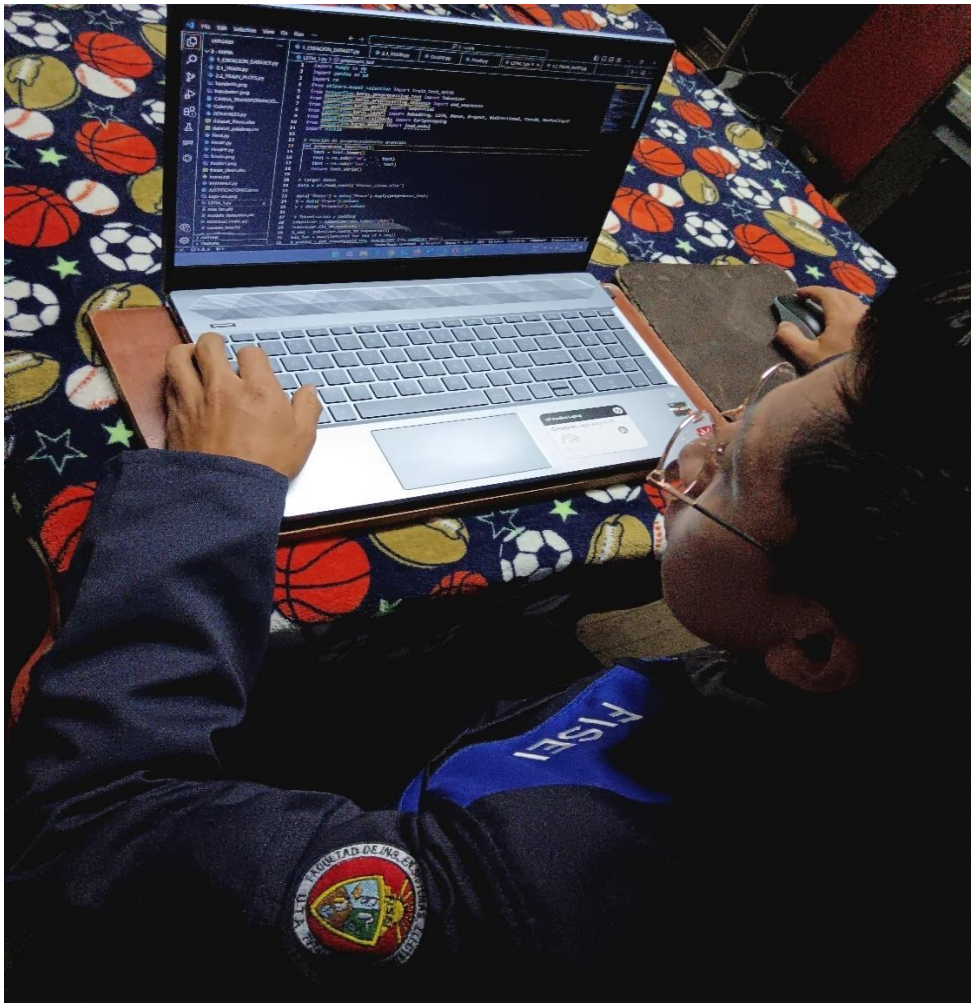


Figura 22. Desarrollo de la programación del sistema

a. Creación de un dataset

Para la creación de la base de datos de entrenamiento se recopilan de datos pertinentes sobre conversaciones extorsivas y no extorsivas, desde las cuales se enlisto frases más comunes. Estos datos fueron obtenidos mediante investigación bibliográfica donde se accedió a reportes de prensa, páginas web y encuestas realizadas a agentes policiales.

Una vez establecido la lista de frases extorsivas y no extorsivas se etiqueta de manera rigurosa según los criterios y variables esenciales que definirán mi análisis. La correcta organización de este dataset es esencial, ya que servirá como la base fundamental sobre la cual desarrollaré modelos analíticos y obtendré conclusiones relevantes que

respalden mi investigación. La base de datos utilizada para el entrenamiento se encuentra en el Anexo A.

b. Carga del dataset para el entrenamiento

Este código en Python emplea la biblioteca Pandas para realizar diversas operaciones en un dataset de frases etiquetadas y posteriormente guarda los resultados en un archivo CSV. A continuación, se describe el proceso detalladamente, paso a paso:

Importación de Pandas: Se carga la biblioteca Pandas con el alias `pd`, una herramienta muy utilizada para la gestión y análisis de datos en Python como se observa en el código de la siguiente Figura 23.

```
1   import pandas as pd
```

Figura 23.Importación de la librería pandas

Carga del Dataset: Se importa un dataset desde un archivo Excel llamado `dataset_frases.xlsx` utilizando la función `pd.read_excel()`. Este dataset se carga y se almacena en un DataFrame de Pandas denominado `dataset_frases`, como se muestra en la siguiente Figura 24. Este DataFrame permite organizar los datos de manera estructurada, facilitando su manipulación y análisis. El uso de `pd.read_excel()` es esencial para leer y convertir el contenido del archivo Excel en un formato que Pandas puede manejar eficientemente, proporcionando una base sólida para cualquier análisis de datos o procesamiento adicional que se requiera en el proyecto. Esta estructura tabular es especialmente útil para operaciones subsecuentes, como filtrado, agrupamiento y visualización de datos.

```
# Cargar el dataset desde el archivo Excel
dataset_frases = pd.read_excel('dataset_frases.xlsx')
```

Figura 24. Lectura de los datos de Excel

Filtrado y Multiplicación de Frases: Las frases del dataset se filtran según sus etiquetas. Las frases etiquetadas con '0' (indicando extorsión) y las etiquetadas con '1' (indicando lenguaje natural) se seleccionan por separado. Posteriormente, cada grupo de frases se multiplica por 5, aumentando su cantidad para equilibrar y reforzar los datos disponibles para el análisis. Esta multiplicación asegura que haya suficientes

ejemplos de cada tipo de frase en el dataset final, mejorando así la eficacia del modelo al entrenarse con una cantidad más robusta de datos etiquetados como se observa el código en la siguiente Figura 25.

```
# Filtrar frases según las etiquetas
lenguaje_extorsivo = dataset_frases[dataset_frases['Etiqueta'] == 0]['Frase'].tolist() * 5  # Frases con etiqueta 0
lenguaje_natural = dataset_frases[dataset_frases['Etiqueta'] == 1]['Frase'].tolist() * 5  # Frases con etiqueta 1
```

Figura 25. Filtrado de frases por etiquetas

Creación de Listas de Etiquetas: Para cada grupo de frases filtradas y multiplicadas (extorsión y lenguaje natural), se generan listas separadas de etiquetas. Por ejemplo, para las frases asociadas con extorsión (etiquetadas con '0'), se crea una lista de etiquetas donde cada elemento es '0', y su longitud coincide con el número total de frases de extorsión multiplicadas. De manera similar, para las frases de lenguaje natural (etiquetadas con '1'), se crea una lista de etiquetas donde cada elemento es '1', con una longitud que corresponde al número total de esas frases después de la multiplicación. Estas listas de etiquetas se utilizarán posteriormente para identificar y clasificar correctamente cada frase en el dataset combinado como se observa en el código de la siguiente Figura 26.

```
# Crear listas de etiquetas
etiquetas_lenguaje_extorsivo = [0] * len(lenguaje_extorsivo)
etiquetas_lenguaje_natural = [1] * len(lenguaje_natural)
```

Figura 26. Creación de lista de etiquetas

Combinación de Listas: Se realiza la combinación de las listas de frases y sus correspondientes etiquetas para formar dos listas finales. La primera lista, denominada `palabras`, contiene todas las frases tanto de extorsión como de lenguaje natural, previamente filtradas y multiplicadas. La segunda lista, denominada `etiquetas`, incluye las etiquetas correspondientes para cada frase en la lista `palabras`, garantizando que cada frase esté correctamente identificada como extorsiva (etiqueta '0') o lenguaje natural (etiqueta '1'). Esta combinación asegura que las frases y sus etiquetas estén alineadas y listas para ser convertidas en un formato tabular para análisis y procesamiento adicional como se observa en código de la siguiente Figura 27.

```
# Combinar listas
palabras = lenguaje_extorsivo + lenguaje_natural
etiquetas = etiquetas_lenguaje_extorsivo + etiquetas_lenguaje_natural
```

Figura 27. Combinación de listas

Creación de un Nuevo DataFrame: Se emplea el constructor de DataFrame de Pandas para generar un nuevo DataFrame llamado dataset, el cual incorpora las listas combinadas de palabras y sus etiquetas correspondientes. Este proceso organiza de manera estructurada las frases (almacenadas en la lista palabras) junto con sus respectivas etiquetas (almacenadas en la lista etiquetas), formando una tabla que facilita su manejo y análisis posterior. Como se observa en el código de la siguiente Figura 28, el DataFrame se crea especificando un diccionario donde las claves son los nombres de las columnas ('palabra' y 'etiqueta'), y los valores son las listas correspondientes (palabras y etiquetas). Esta estructura tabular es esencial para asegurar que cada frase esté claramente asociada con su etiqueta, lo que es fundamental para realizar análisis precisos y para el entrenamiento de modelos de aprendizaje automático.

```
# Crear DataFrame
dataset = pd.DataFrame({'palabra': palabras, 'etiqueta': etiquetas})
```

Figura 28. Creación de un DataFrame

Guardado del DataFrame en CSV: El DataFrame dataset se guarda como un archivo CSV denominado dataset_palabras.csv en el directorio actual. Este paso se realiza utilizando el método to_csv () de Pandas, el cual convierte el DataFrame en un archivo CSV. Para asegurarse de que el archivo no incluye el índice de fila, se establece el parámetro index=False. Como se observa en el código de la siguiente Figura 29, esta configuración garantiza que el archivo CSV contenga únicamente las columnas de datos especificadas, sin incluir información de índice innecesario.

```
# Guardar dataset en un archivo CSV
dataset.to_csv('dataset_palabras.csv', index=False)
print("Dataset creado y guardado como 'dataset_palabras.csv'")
```

Figura 29. Guardado del dataset en un archivo csv

c. Entrenamiento del sistema

El entrenamiento de un sistema de clasificación de texto involucra varios pasos cruciales para transformar datos textuales en un modelo predictivo eficiente. Usando un conjunto de datos de frases con etiquetas, el proceso comienza con la vectorización del texto, convirtiendo las palabras en representaciones numéricas. Luego, se divide el conjunto de datos en partes de entrenamiento y prueba para evaluar la efectividad del modelo. Se emplea un clasificador de bosque aleatorio para entrenar el sistema, permitiéndole identificar patrones y correlaciones entre las palabras y sus etiquetas correspondientes. Finalmente, se guarda el modelo entrenado y el vectorizador para futuras predicciones, facilitando la clasificación precisa de nuevas frases.

Se importaron las bibliotecas necesarias para el desarrollo del modelo:

pandas: Utilizada para la manipulación y análisis de datos, permitiendo la lectura y estructuración del dataset.

CountVectorizer de sklearn.feature_extraction.text: Encargada de convertir el texto en una matriz de términos (bag of words), facilitando la representación numérica de las frases.

train_test_split de sklearn.model_selection: Empleada para dividir el dataset en conjuntos de entrenamiento y prueba, asegurando una evaluación imparcial y precisa del modelo.

RandomForestClassifier de sklearn.ensemble: Utilizada para crear y entrenar el modelo de bosque aleatorio, aprovechando su capacidad para manejar datos complejos y prevenir el sobreajuste.

accuracy_score de sklearn.metrics: Utilizada para evaluar la precisión del modelo, comparando las predicciones con las etiquetas reales.

joblib: Utilizada para guardar el modelo entrenado y el vectorizador en archivos, permitiendo su reutilización en futuras predicciones.

Estas bibliotecas son fundamentales para el preprocesamiento de datos, la creación y evaluación del modelo, y la conservación de los componentes del sistema para su uso continuado como se puede observar en el código de la siguiente Figura 30.

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import joblib
```

Figura 30. Importación de bibliotecas

- ***Carga de dataset***

Se cargó el dataset desde un archivo CSV llamado `dataset_palabras.csv`. Este archivo contiene una colección estructurada de frases etiquetadas, esenciales para el entrenamiento del modelo. Utilizando la biblioteca `pandas`, se leyó el contenido del archivo CSV, permitiendo organizar los datos en un DataFrame, que facilita su manipulación y análisis. Este paso es crucial, ya que proporciona la base de datos necesaria para convertir el texto en representaciones numéricas y entrenar el modelo de clasificación como se observa en el código de la siguiente Figura 31.

```
# Cargar el dataset
dataset = pd.read_csv('dataset_palabras.csv')
```

Figura 31. Carga del dataset

- ***Vectorización de las Palabras***

Las palabras del dataset se convierten en una matriz de términos utilizando la técnica de `CountVectorizer`. Esta herramienta transforma el texto en una representación numérica conocida como matriz esparsa, donde cada fila corresponde a una frase del dataset y cada columna representa una palabra única del vocabulario del dataset. La matriz resultante captura la frecuencia de aparición de cada palabra en las frases, proporcionando una forma estructurada y cuantificable de analizar el texto. Este proceso es fundamental para que los algoritmos de aprendizaje automático puedan procesar y aprender de los datos textuales de manera efectiva como se observa en código de la siguiente Figura 32.

```
# Vectorización de las palabras
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(dataset['palabra'])
```

Figura 32. Vectorización de palabras

- ***Separación de las Etiquetas***

Se separaron las etiquetas (clases) asociadas con cada palabra en el dataset. Este paso es crucial, ya que las etiquetas representan las categorías o clases que el modelo debe aprender a predecir. En este contexto, las etiquetas podrían indicar si una frase pertenece a una categoría de lenguaje natural o a una categoría de lenguaje extorsivo. Al separar las etiquetas de los datos textuales, se permite que el modelo de aprendizaje automático enfoque su entrenamiento en identificar los patrones y características distintivas de cada clase. Esta separación también facilita la evaluación de7l modelo, ya que las predicciones generadas pueden compararse directamente con las etiquetas verdaderas para medir la precisión y efectividad del modelo. Además, una correcta gestión de las etiquetas asegura que el proceso de entrenamiento y validación se realice de manera organizada y estructurada, mejorando la calidad y la fiabilidad del sistema de clasificación resultante como se observa en el código de la siguiente Figura 33 .

```
# Separar las etiquetas
y = dataset['etiqueta']
```

Figura 33. Separación de etiquetas

- ***División del Dataset en Entrenamiento y Prueba***

Los datos se dividen en dos conjuntos: entrenamiento (80%) y prueba (20%). El parámetro random_state=42 se utiliza para garantizar que la división sea reproducible.

  - **Entrenamiento (80%):** Este conjunto de datos se utiliza para entrenar el modelo, permitiéndole aprender los patrones y características distintivas entre las diferentes clases de datos.

  - **Prueba (20%):** Este conjunto se utiliza para evaluar el rendimiento del modelo una vez que ha sido entrenado, proporcionando una medida de su capacidad para generalizar a datos no vistos previamente.

- **random_state=42:** Este parámetro asegura que la división de los datos sea consistente en cada ejecución del código. Sin este parámetro, la división podría variar cada vez que se ejecute el script, lo que haría difícil reproducir los resultados y evaluar con precisión el rendimiento del modelo.

La razón de utilizar una proporción de 80% para entrenamiento y 20% para prueba es equilibrar entre dar suficientes datos para que el modelo aprenda y mantener una porción significativa para evaluar su desempeño de manera confiable como se observa en el código de la siguiente Figura 34.

```
# Dividir los datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figura 34. División de los datos de entrenamiento y prueba

- ***Diagrama de flujo del funcionamiento del algoritmo de inteligencia artificial***

El diagrama de flujo describe el funcionamiento del algoritmo de inteligencia artificial Random Forest Classifier. Inicia con la importación de las librerías necesarias, seguida de la lectura del conjunto de datos que contiene las palabras y sus respectivas etiquetas. Posteriormente, las palabras se transforman en una matriz de características mediante un vectorizador. Las etiquetas se extraen y los datos se dividen en conjuntos de entrenamiento y prueba. El clasificador Random Forest se entrena con los datos de entrenamiento y se utiliza para predecir etiquetas en el conjunto de prueba. Finalmente, se calcula la precisión del modelo y se guarda el modelo entrenado junto con el vectorizador para su uso futuro. Este diagrama de flujo ilustra los pasos secuenciales del proceso como se observa en la siguiente Figura 35, asegurando una comprensión clara del algoritmo desde la preparación de los datos hasta la evaluación del modelo.

- ***Entrenamiento del Modelo***

Se crea y entrena un modelo de bosque aleatorio con 100 árboles (estimadores) utilizando los datos de entrenamiento. Este modelo, conocido como RandomForestClassifier, combina múltiples árboles de decisión para mejorar la precisión y reducir el riesgo de sobreajuste. Al utilizar 100 árboles, se busca alcanzar

un equilibrio entre la capacidad del modelo para generalizar bien y el tiempo de entrenamiento. Cada árbol de decisión en el bosque aleatorio se entrena con una muestra aleatoria del conjunto de datos, lo que ayuda a capturar una variedad de patrones y relaciones en los datos. La técnica de ensamblaje del bosque aleatorio promedia las predicciones de todos los árboles, lo que generalmente resulta en un modelo más robusto y preciso comparado con un único árbol de decisión. Esta configuración de 100 árboles es una práctica común que proporciona un buen rendimiento en una amplia variedad de tareas de clasificación como se observa en el código de la siguiente Figura 36.

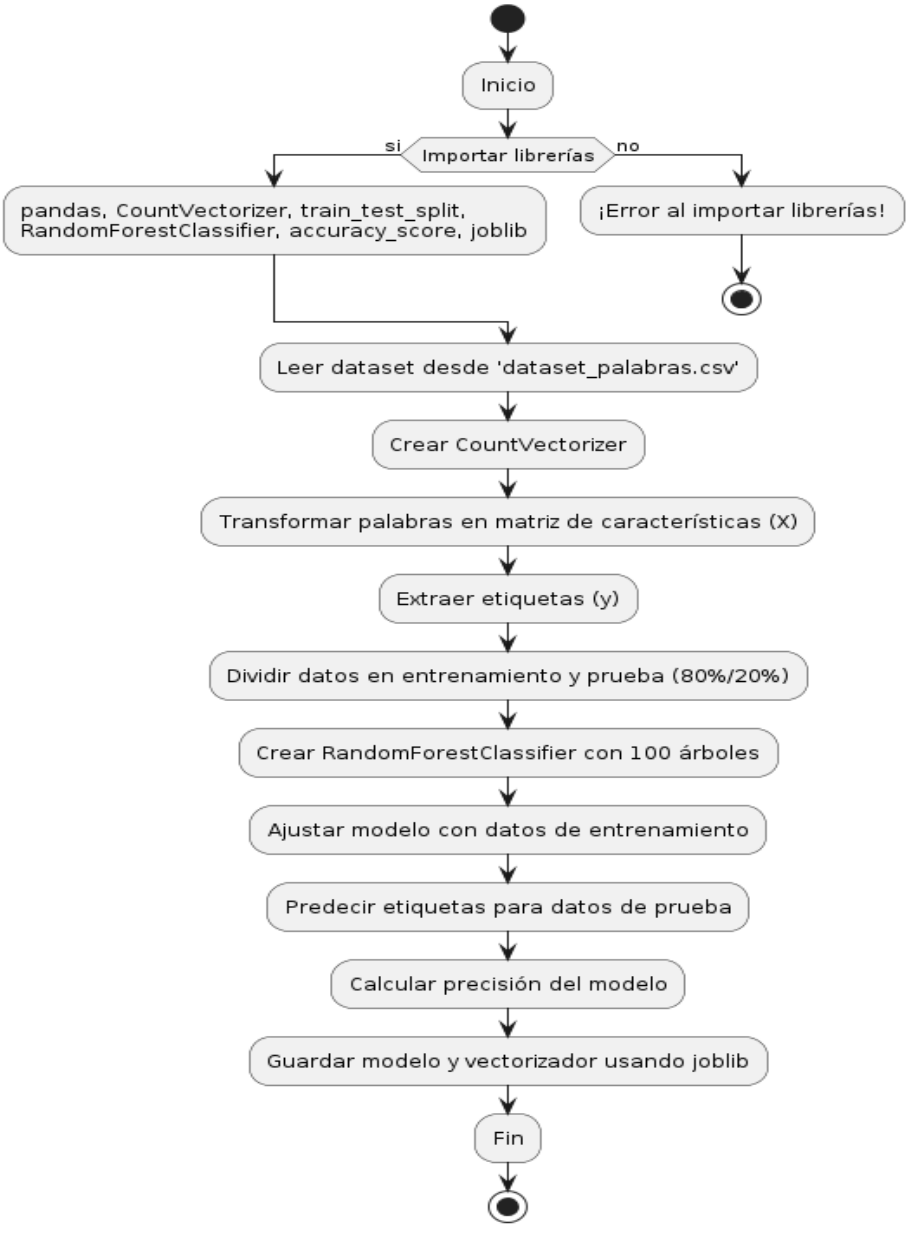


Figura 35. Diagrama de flujo de la inteligencia artificial

```
# Entrenar el modelo
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

Figura 36. Entrenamiento del modelo

- ***Evaluación del Modelo***

El modelo realiza predicciones sobre el conjunto de datos de prueba y luego calcula la precisión del modelo comparando estas predicciones con las etiquetas reales. Este proceso implica utilizar el modelo entrenado para predecir las clases de las frases en el conjunto de prueba, generando una lista de predicciones. A continuación, se compara esta lista con las etiquetas verdaderas del conjunto de prueba para evaluar la exactitud del modelo.

- **Predicciones**: El modelo utiliza los datos del conjunto de prueba (X_test) para generar predicciones (y_pred). Este paso evalúa cómo el modelo, entrenado con datos previamente no vistos, clasifica correctamente o incorrectamente las frases como se observa en el código de la siguiente Figura 37.

```
# Evaluar el modelo
y_pred = model.predict(X_test)
```

Figura 37. Predicción del modelo

- **Cálculo de Precisión**: La precisión se calcula como la proporción de predicciones correctas en comparación con el total de predicciones realizadas. La métrica accuracy_score de sklearn.metrics se emplea para este propósito como se observa en el código de la siguiente Figura 38.

```
# Evaluar el modelo
y_pred = model.predict(X_test)
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
```

Figura 38. Cálculo de la precisión

La precisión es una métrica crucial que indica el rendimiento general del modelo. Una alta precisión sugiere que el modelo es eficaz para clasificar correctamente las frases

extorsivas y de lenguaje natural. Esta métrica ayuda a entender si el modelo puede ser confiablemente implementado para la detección automática de lenguaje extorsivo.

La comparación de las predicciones con las etiquetas reales proporciona una evaluación objetiva del modelo. Esta evaluación es fundamental para identificar posibles mejoras en el modelo o en los datos utilizados para el entrenamiento.

- ***Guardar el Modelo y el Vectorizador***

El modelo entrenado y el vectorizador se guardan en los archivos modelo_deteccion.pkl y vectorizer.pkl, respectivamente. Este proceso implica utilizar la biblioteca joblib para serializar el modelo y el vectorizador, lo que permite almacenar sus configuraciones y estados actuales. Al guardar estos componentes, se asegura que el modelo y el vectorizador puedan ser fácilmente cargados y reutilizados en el futuro sin necesidad de entrenarlos nuevamente.

- **Serialización del Modelo**: Utilizando joblib.dump, el modelo de RandomForestClassifier entrenado se guarda en un archivo llamado modelo_deteccion.pkl. Este archivo contiene toda la información necesaria para recrear el modelo en su estado actual, incluyendo los parámetros ajustados durante el entrenamiento como se observa en el código de la siguiente Figura 39.

```
# Guardar el modelo y el vectorizador
joblib.dump(model, 'modelo_deteccion.pkl')
```

Figura 39. Guarda el modelo

- **Serialización del Vectorizador**: De manera similar, el vectorizador CountVectorizer se guarda en un archivo denominado vectorizer.pkl. Este archivo almacena el vocabulario y las transformaciones aprendidas durante la vectorización del texto, asegurando que futuras entradas de texto puedan ser procesadas de manera consistente como se observa en el código de la siguiente Figura 40.

```
# Guardar el modelo y el vectorizador
joblib.dump(model, 'modelo_deteccion.pkl')
joblib.dump(vectorizer, 'vectorizer.pkl')
```

Figura 40. Guarda el vectorizador

Guardar el modelo y el vectorizador proporciona varios beneficios:

- **Reutilización:** Permite cargar y utilizar el modelo entrenado sin necesidad de repetir el proceso de entrenamiento, ahorrando tiempo y recursos computacionales.
- **Consistencia:** Garantiza que las transformaciones de texto y las predicciones del modelo sean consistentes entre diferentes ejecuciones.
- **Despliegue:** Facilita el despliegue del modelo en diferentes entornos de producción, donde puede ser utilizado para hacer predicciones en tiempo real.
- **Mantenimiento:** Simplifica el proceso de mantenimiento y actualización del modelo, permitiendo almacenar y versionar diferentes iteraciones del modelo.

- ***Probar el Modelo con Algunos Ejemplos***

El modelo se prueba utilizando una lista de frases de ejemplo. Cada frase se convierte a la misma representación numérica empleando el vectorizador previamente guardado, y el modelo predice la etiqueta correspondiente para cada frase. Las predicciones se imprimen junto con las frases originales, permitiendo evaluar cómo clasifica el modelo las nuevas entradas.

- **Conversión de Frases**: Las frases de prueba se transforman en vectores numéricos utilizando el vectorizador guardado (vectorizer.pkl). Este paso asegura que las nuevas frases se representen de manera consistente con el formato utilizado durante el entrenamiento del modelo como se muestra en el código de la siguiente Figura 41.

```
# Probar el modelo con algunos ejemplos
palabras_de_prueba = ["Si no pagas, tu familia sufrirá las consecuencias", "Esto es un asalto",
for palabra in palabras_de_prueba:
```

Figura 41. Frases para prueba del modelo

- **Predicciones del Modelo**: El modelo entrenado (modelo_deteccion.pkl) se utiliza para predecir la etiqueta de cada frase vectorizada. Esta predicción determina si una frase pertenece a la categoría de lenguaje extorsivo o lenguaje natural como se observa en el código de la siguiente Figura 42.

```
# Probar el modelo con algunos ejemplos
palabras_de_prueba = ["Si no pagas, tu familia sufrirá las consecuencias", "Esto es un asalto",
for palabra in palabras_de_prueba:
    X_new = vectorizer.transform([palabra])
    prediccion = model.predict(X_new)[0]
```

Figura 42. Predicciones del modelo

- **Impresión de Resultados**: Se imprimen tanto las frases originales como sus predicciones correspondientes. Esto facilita la evaluación de la capacidad del modelo para clasificar correctamente diferentes tipos de frases y proporciona una comprensión clara de su desempeño en un contexto práctico como se observa en el código de la siguiente Figura 43.

```
# Probar el modelo con algunos ejemplos
palabras_de_prueba = ["Si no pagas, tu familia sufrirá las consecuencias", "Esto es un asalto",
for palabra in palabras_de_prueba:
    X_new = vectorizer.transform([palabra])
    prediccion = model.predict(X_new)[0]
    print(f"Palabra: {palabra}, Predicción: {prediccion}")
```

Figura 43. Impresión de resultados

3.3.1 Agregar nuevas frases al dataset para el entrenamiento

Para agregar nuevas frases, tanto extorsivas como no extorsivas, al sistema de detección de extorsiones, se debe primero incorporarlas a una base de datos existente en Excel denominada "dataset_frases.xlsx". Estas frases serán luego integradas al dataset del sistema, denominado "dataset_palabras.csv", el cual será utilizado para el entrenamiento del modelo. Estos pasos se describen a continuación:

a. Creación y guarado del dataset

Se carga un conjunto de datos desde un archivo Excel y se filtran las frases según sus etiquetas. Se multiplican las frases con etiquetas 0 y 1 por cinco, creando listas ampliadas de frases. Luego, se combinan estas listas de frases y sus etiquetas

correspondientes en un nuevo DataFrame. Finalmente, este DataFrame se guarda como un archivo CSV llamado 'dataset_palabras.csv'. Al completar la ejecución, el código confirma que el dataset ha sido creado y guardado exitosamente con el mensaje "Dataset creado y guardado como 'dataset_palabras.csv` como se observa en la siguiente Figura 44.

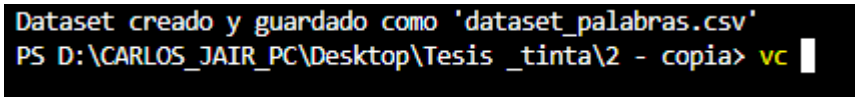


Figura 44. Creación y guardado del dataset

Las frases se etiquetan según sus características y se multiplican para equilibrar las clases. Estas frases etiquetadas se combinan en un nuevo DataFrame, que luego se guarda en un archivo CSV llamado 'dataset_palabras.csv', asegurando que las frases y sus etiquetas estén organizadas y listas para su uso posterior. A continuación, se muestra como es la estructura del dataset una vez creado y guardado en la siguiente Figura 45 y también podemos observar la estructura del mismo y como se clasifican las frases dentro del dataset en el Anexo C.

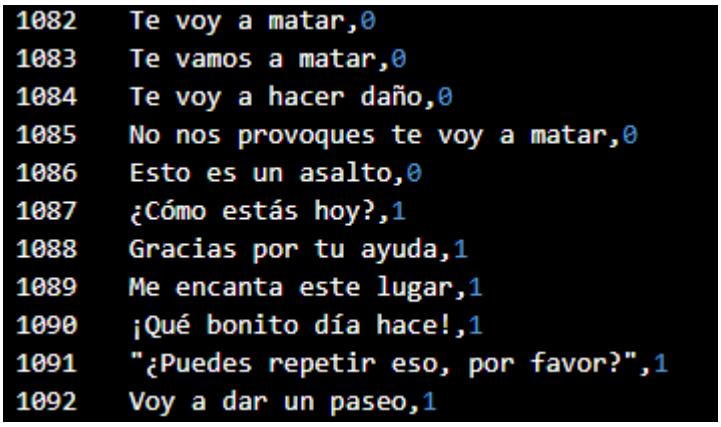


Figura 45. Estructura del dataset

b. Entrenamiento del algoritmo de inteligencia artificial

Se lleva a cabo el entrenamiento de un modelo de clasificación Random Forest destinado a la detección de frases asociadas con situaciones de extorsión. El proceso comienza con la carga del conjunto de datos desde el archivo CSV 'dataset_palabras.csv', que contiene frases y sus respectivas etiquetas. Las frases se vectorizan utilizando CountVectorizer para convertirlas en una matriz de

características numéricas. Posteriormente, las etiquetas se separan de las características y los datos se dividen en conjuntos de entrenamiento y prueba, asegurando una evaluación precisa del modelo. El modelo de Random Forest se entrena con los datos de entrenamiento y se evalúa su precisión en el conjunto de prueba, logrando una exactitud del 100%. Además, el modelo entrenado y el vectorizador se guardan en archivos para su uso futuro.

Para probar la eficacia del sistema, se utilizan varias frases de ejemplo, revelando que el modelo puede distinguir con precisión entre frases relacionadas con extorsión (etiqueta 0) y frases no relacionadas (etiqueta 1). Los resultados muestran que precisión tiene el sistema con las frases seleccionadas para hacer las pruebas al momento de hacer el entrenamiento como se observa en la siguiente Figura 46.

Precisión: El modelo logró una precisión del 100% en el conjunto de prueba.

Predicciones para frases de prueba:

- "Si no pagas, tu familia sufrirá las consecuencias" → Predicción: 0 (extorsión)
- "Esto es un asalto" → Predicción: 0 (extorsión)
- "Tu vida está en nuestras manos, no nos provoques" → Predicción: 0 (extorsión)
- "Me encanta este lugar" → Predicción: 1 (no extorsión)
- "Es divertido salir a comer" → Predicción: 1 (no extorsión)
- "No queremos herirte pero estamos preparados para hacerlo" → Predicción: 0 (extorsión)
- "Vamos a hacer una excursión" → Predicción: 1 (no extorsión)

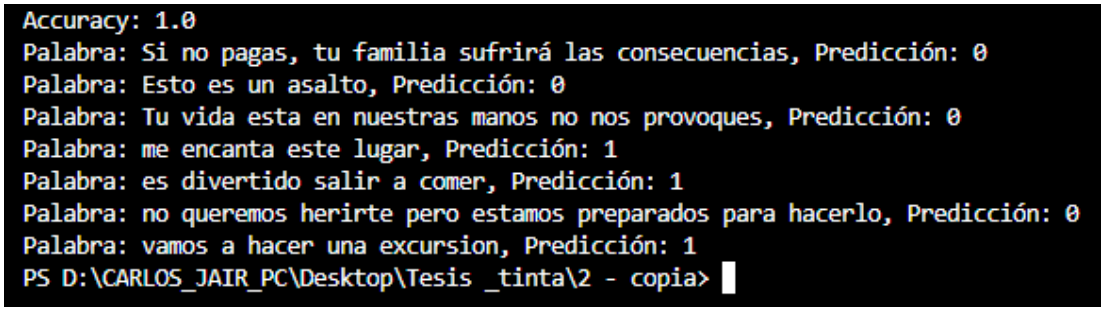


Figura 46. Resultado del entrenamiento del sistema con frases de prueba

3.3.2 Interfaz detector de extorsión utilizando reconocimiento de voz

Este código implementa un sistema de detección de extorsión mediante el uso de reconocimiento de voz y una interfaz gráfica de usuario (GUI) utilizando Tkinter en Python. A continuación, se describe cómo opera paso a paso:

- ***Importación de Librerías***

En la primera etapa del código, se realiza la importación de librerías esenciales que permitirán el correcto funcionamiento del sistema. Estas bibliotecas incluyen herramientas para el reconocimiento de voz, manejo de modelos de machine learning y creación de interfaces gráficas, proporcionando así una base sólida para las funcionalidades del detector de extorsión como se observa en el código de la siguiente Figura 47.

- **Tkinter**: Es la biblioteca estándar de Python para crear interfaces gráficas de usuario (GUI). Proporciona herramientas como messagebox (para mostrar mensajes emergentes), Label (para etiquetas de texto), Button (para botones interactivos), Entry (para campos de entrada de texto), Frame (para organizar y contener otros componentes), y Canvas (para dibujar gráficos y mostrar imágenes).

- **PIL.Image** y **PIL.ImageTk**: Estas pertenecen al módulo Python Imaging Library (PIL), renombrado como Pillow. Permiten cargar, manipular y mostrar imágenes en formatos diversos dentro de la interfaz gráfica.

- **speech_recognition:** Esta biblioteca se utiliza para realizar el reconocimiento de voz. Funciona capturando audio a través de un micrófono y utilizando diversos servicios de reconocimiento para transcribir el audio a texto.

- **joblib:** Es una biblioteca que se emplea para serializar (guardar) y deserializar (cargar) modelos de machine learning. Facilita el almacenamiento y reutilización de modelos previamente entrenados sin necesidad de volver a entrenarlos cada vez.

- **pyttsx3**: Permite la síntesis de voz en la aplicación. Esta biblioteca convierte texto en voz, proporcionando retroalimentación auditiva basada en los resultados de análisis de voz y detección de extorsiones.

```
import tkinter as tk
from tkinter import messagebox, Label, Button, Entry, Frame, Canvas
from PIL import Image, ImageTk
import speech_recognition as sr
import joblib
import pyttsx3
```

Figura 47. Importación de librerías para el sistema

- ***Inicializa pyttsx3 y carga del Modelo y Vectorizador***

El código carga un modelo de detección de extorsiones (`modelo_deteccion.pkl`) y un vectorizador (`vectorizer.pkl`) desde archivos locales. Estos archivos contienen un modelo de machine learning previamente entrenado y un objeto de vectorización fundamental para procesar el texto de entrada. Este paso asegura que el sistema esté equipado con las herramientas necesarias para analizar y predecir el contenido de las transcripciones de voz, como se muestra en el código proporcionado.

La función `pyttsx3.init()` se emplea para inicializar el motor de síntesis de voz en aplicaciones desarrolladas en Python a través de la biblioteca `pyttsx3`. Este motor facilita la conversión eficiente de texto a voz, permitiendo que la aplicación proporcione retroalimentación auditiva al usuario mediante respuestas verbales basadas en los eventos o resultados obtenidos durante la ejecución del programa. Al llamar a `pyttsx3.init()`, se configuran parámetros como la selección de voz, velocidad

de habla y otras opciones ajustables según las necesidades específicas del desarrollo. Además, `pyttsx3` es compatible con múltiples plataformas y versiones de Python, lo que la convierte en una solución versátil para integrar funcionalidades de síntesis de voz en una amplia gama de aplicaciones, desde asistentes virtuales hasta herramientas educativas y programas de accesibilidad como se muestra en la Figura 48.

```
# Inicializar pyttsx3
engine = pyttsx3.init()

# Cargar el modelo y el vectorizador
try:
    model = joblib.load('modelo_deteccion.pkl')
    vectorizer = joblib.load('vectorizer.pkl')
except Exception as e:
    print(f"Error al cargar el modelo o el vectorizador: {e}")
    exit(1)
```

Figura 48. Inicializa pyttsx3 y carga el modelo y el vectorizador

- ***Función login***

La función `login` verifica si los datos de usuario y contraseña ingresados coinciden con los valores predefinidos ("Carlos" y "1010"). Si hay coincidencia, oculta el marco de inicio de sesión (`login_frame`) y muestra el marco principal (`main_frame`). En caso contrario, utiliza `messagebox` para notificar al usuario sobre el error en las credenciales ingresadas. Este enfoque es común en aplicaciones que requieren autenticación, asegurando que solo los usuarios autorizados puedan acceder a funcionalidades específicas mientras proporciona retroalimentación clara sobre la validación de las credenciales como se observa en la Figura 49.

```
# Función para realizar el login
def login():
    if entry_user.get() == "Carlos" and entry_pass.get() == "1010":
        login_frame.pack_forget()  # Oculta la pantalla de login
        main_frame.pack(fill='both', expand=True)  # Muestra la pantalla principal
    else:
        messagebox.showerror("Error", "Usuario o contraseña incorrecta")
```

Figura 49. Función login

- ***Inicialización de la Interfaz Gráfica (Tkinter)***

Se crea una ventana principal denominada `root`, la cual se configura con el título "Detector de Extorsión" y se establecen dimensiones de 500x300 píxeles. Esta ventana servirá como la interfaz gráfica de usuario donde se mostrarán los diversos componentes del sistema, permitiendo al usuario interactuar fácilmente con la aplicación para realizar el reconocimiento y detección de posibles extorsiones. La elección del tamaño y título busca proporcionar una experiencia de usuario clara y accesible como se observa en la siguiente Figura 50.

```
# Inicia la GUI de Tkinter
root = tk.Tk()
root.title("Detector de Extorsión")
root.geometry("500x300")  # Ajusta el tamaño de la ventana aquí
```

Figura 50. Interfaz Tkinter

- ***Función reconocer_voz()***

La función `reconocer_voz()` es el núcleo del sistema de detección de extorsión, donde se emplea una combinación de técnicas de reconocimiento de voz y machine learning para identificar posibles contenidos de extorsión en el discurso del usuario. Aquí se explica su funcionamiento detallado como se observan en las Figura 51 y Figura 52:

- ***Recognizer y Microphone***

  - **Recognizer y Microphone:** La función inicia creando una instancia de `Recognizer` y `Microphone` de la biblioteca `speech_recognition`. El `Recognizer` es responsable de capturar y procesar la voz del usuario, mientras que el `Microphone` se utiliza como la fuente de entrada de audio.

  - **Captura de Audio:** Cuando se invoca la función, el `Recognizer` utiliza el `Microphone` para captar el audio del usuario.

- ***Ajuste de Ruido***

  - **Ajuste de Ruido Ambiental:** Antes de comenzar a escuchar, el sistema ajusta el reconocimiento de voz al ruido ambiental circundante. Esto se hace llamando al método `adjust_for_ambient_noise` del `Recognizer`, lo que mejora la

precisión del reconocimiento al calibrar el micrófono según el nivel de ruido de fondo.

- ***Escuchando***

  - **Modo de Escucha:** Una vez ajustado al ruido ambiental, se actualiza la interfaz gráfica para mostrar un mensaje que indica que el sistema está en modo de escucha. Esto proporciona al usuario una indicación visual de que el sistema está preparado para capturar audio.

- ***Manejo de Timeout***

  - **Gestión de Timeout:** El sistema utiliza un tiempo de espera (timeout) para escuchar la voz. Si no se detecta voz dentro del tiempo especificado, se lanza una excepción `WaitTimeoutError`. En tal caso, la interfaz se actualiza para notificar al usuario que no se detectó ninguna voz en el tiempo permitido.

- ***Reconocimiento de Voz***

  - **Conversión de Audio a Texto:** El audio capturado se envía al servicio de Google Speech Recognition, que lo convierte en texto en español (`es-ES`). Este texto se convierte a minúsculas para una mejor normalización y comparación con los datos de entrenamiento del modelo.

- ***Vectorización y Predicción***

  - **Vectorización:** El texto reconocido se transforma utilizando un objeto de vectorización previamente cargado. El vectorizador convierte el texto en una representación numérica que el modelo de machine learning puede procesar.

  - **Predicción de Extorsión:** El texto vectorizado se pasa al modelo de machine learning cargado, que predice si el contenido del texto es una posible extorsión. Una predicción de `0` indica que se ha detectado extorsión.

- ***Reproducción de los datos escuchados***

Después de realizar la predicción, la aplicación actualiza la interfaz gráfica mediante el componente `label_status`, proporcionando retroalimentación visual sobre los resultados obtenidos. Además, utiliza la biblioteca `pyttsx3` para sintetizar voz y comunicar auditivamente los resultados al usuario. Dependiendo de la predicción realizada, la aplicación también muestra mensajes emergentes de alerta utilizando `messagebox`, lo que permite informar de manera clara y efectiva al usuario sobre los eventos detectados o las acciones recomendadas. Este enfoque integra múltiples formas de interacción para mejorar la experiencia del usuario y asegurar una comunicación completa de los resultados del análisis realizado.

.

```python
def reconocer_voz():
    recognizer = sr.Recognizer()
    mic = sr.Microphone()

    with mic as source:
        label_status.config(text="Ajustando para el ruido ambiental...")
        root.update()
        recognizer.adjust_for_ambient_noise(source)
        label_status.config(text="Escuchando...")
        root.update()

        try:
            audio = recognizer.listen(source, timeout=5, phrase_time_limit=10)
        except sr.WaitTimeoutError:
            label_status.config(text="Timeout: No se detectó ninguna voz")
            return

    try:
        texto = recognizer.recognize_google(audio, language='es-ES').lower()
        label_result.config(text=f"Escuché: {texto}")

        # Vectorizar la palabra detectada
        X_new = vectorizer.transform([texto])

        # Predecir usando el modelo
        prediccion = model.predict(X_new)[0]
```

Figura 51. Función reconocimiento de voz parte 1

```
        # Mostrar resultados y realizar acciones dependiendo del resultado
        if prediccion == 0:
            label_status.config(text="EXTORSIÓN DETECTADA!", fg='red')
            messagebox.showwarning("Alerta de Extorsión", "Se ha detectado posible extorsión.")
        else:
            label_status.config(text="Sin detección de extorsión.", fg='green')
            messagebox.showinfo("Resultado", "No se detectó extorsión.")
    except sr.UnknownValueError:
        messagebox.showerror("Error", "No se pudo entender el audio.")
    except sr.RequestError as e:
        messagebox.showerror("Error de Red", f"Error al realizar la solicitud: {e}")
    except Exception as e:
        messagebox.showerror("Error", f"Error inesperado: {e}")
```

Figura 52. Función reconocimiento de voz parte 2

c. Función reproducir_texto

La función obtiene el contenido textual del widget `label_result`. Si este contiene texto, utiliza la biblioteca `pyttsx3` para convertirlo en voz y reproducirlo auditivamente. En caso de que el widget esté vacío, la función muestra un mensaje informativo utilizando `messagebox`, informando al usuario que no hay texto disponible para reproducir. Esta funcionalidad no solo permite al usuario escuchar el contenido textual presente en la interfaz, sino que también proporciona una retroalimentación adecuada en caso de que no haya información para reproducir, mejorando así la usabilidad y la claridad de la aplicación como se observa en la Figura 53.

```
# Función para reproducir el texto del label_result
def reproducir_texto():
    texto = label_result.cget("text")
    if texto:
        engine.say(texto)
        engine.runAndWait()
    else:
        messagebox.showinfo("Información", "No hay texto para reproducir.")
```

Figura 53. Función reproducir texto

- ***Función load_image***

La función carga una imagen desde una ubicación de archivo específica y la redimensiona para ajustarla a una resolución de 80x80 píxeles. Luego, utiliza un componente Canvas de Tkinter para crear y mostrar la imagen dentro de la interfaz gráfica. Para evitar que la imagen sea eliminada inadvertidamente por el recolector de

basura de Python, la función guarda una referencia de la imagen en el Canvas. Este enfoque asegura que la imagen permanezca visible y accesible dentro de la interfaz mientras el programa está en ejecución, mejorando la estabilidad y la experiencia del usuario al manipular imágenes dinámicamente en la aplicación como se muestra en la Figura 54.

```
# Función para cargar la imagen y dibujarla en el Canvas
def load_image(canvas, path):
    image = Image.open(path)
    image = image.resize((80, 80), Image.ANTIALIAS)  # Redimensionar la imagen
    photo = ImageTk.PhotoImage(image)
    canvas.create_image(40, 40, image=photo)
    canvas.image = photo  # Guardar referencia para evitar que se recoja la basura
```

Figura 54. Carga la imagen y la dibuja en canvas

- ***Configuración de root:***

La aplicación crea la ventana principal (`root`) configurando un título específico y dimensiones predefinidas para la interfaz gráfica. Además de estos ajustes básicos, se personaliza la apariencia añadiendo un ícono personalizado que se muestra en la barra de título de la ventana. Esta práctica no solo permite identificar rápidamente la aplicación visualmente, sino que también añade un toque de personalización que puede mejorar la experiencia del usuario al distinguir fácilmente la aplicación entre otras ventanas abiertas en el sistema operativo como se muestra en la Figura 55

```
# Configuración de la ventana principal
root = tk.Tk()
root.title("SISTEMA DE DETECCIÓN DE EXTORSIONES MEDIANTE EL ANÁLISIS DE LAS SEÑALES DE LA VOZ")
root.geometry("669x400")  # Tamaño ajustado según el diseño

# Quitar el icono predeterminado o poner un icono personalizado
root.iconbitmap('D:/CARLOS_JAIR_PC/Desktop/Tesis _tinta/2 - copia/icono.ico')
```

Figura 55. Configuración ventana principal

d. Configuración de login_frame

La aplicación crea un frame denominado `login_frame` para la pantalla de inicio de sesión, el cual se caracteriza por tener un color de fondo específico y un borde resaltado para destacarlo visualmente. Dentro de este frame, se añade una etiqueta de bienvenida (`label_welcome`), configurada con un texto de saludo, una fuente tipográfica

definida, y un color de fondo distintivo. Además, esta etiqueta cuenta con un borde destacado para mejorar su visibilidad y atractivo estético. Estas configuraciones iniciales no solo estructuran la interfaz de inicio de sesión de manera clara y acogedora, sino que también contribuyen a una mejor experiencia de usuario mediante una presentación visualmente atractiva y funcional como se muestra en la Figura 56.

```
# Frame de login
login_frame = Frame(root, bg='#191D4C', highlightbackground="black", highlightcolor="black", highlightthickness=2)
login_frame.pack(expand=True, fill='both')

label_welcome = Label(login_frame, text="BIENVENIDO AL PROGRAMA DE DETECCIÓN DE EXTORSIONES", font=('Times new roman',
label_welcome.pack(pady=(20, 10))
```

Figura 56. Configuración del login_frame

e. Configuración de form_frame y canvas_uta

La aplicación crea un frame denominado `form_frame` dentro de `login_frame`, cuyo propósito es organizar y alinear de manera ordenada las etiquetas y los campos de texto. Adicionalmente, se incorpora un Canvas llamado `canvas_uta` dentro de `form_frame`, destinado a mostrar una imagen de un sello que es cargada desde una ruta específica. Esta estructura permite una disposición más coherente y visualmente agradable de los elementos del formulario, facilitando la interacción del usuario. El uso del Canvas para mostrar la imagen no solo mejora la estética de la interfaz, sino que también proporciona un contexto visual relevante y distintivo, lo cual puede ser particularmente útil en aplicaciones que requieren autenticación visual o branding institucional como se muestra en la Figura 57.

```
# Etiquetas y campos de texto en una nueva Frame para alinear correctamente
form_frame = Frame(login_frame, bg='#191D4C')
form_frame.pack()

# Creación de etiquetas redondas utilizando Canvas
canvas_uta = Canvas(form_frame, width=80, height=80, bg='#191D4C', bd=0, highlightthickness=0)
canvas_uta.grid(row=0, column=0, padx=(10, 10), pady=(0, 20))
# Cargar y mostrar la imagen de UTA en el Canvas
load_image(canvas_uta, "D:/CARLOS_JAIR_PC/Desktop/Tesis _tinta/2 - copia/sello.png")
```

Figura 57. Configuración del form_frame

f. Configuración de ingreso al sistema

La configuración de los elementos de la interfaz incluye la creación de la etiqueta `label_user` y el campo de entrada `entry_user` dentro de `form_frame`,

específicamente para que el usuario pueda ingresar su nombre de usuario. Adicionalmente, se configura otro Canvas, denominado `canvas_fisei`, también dentro de `form_frame`, que se utiliza para mostrar una imagen del logo cargada desde una ruta específica, añadiendo un elemento visual representativo de la institución o marca.

Asimismo, se establecen la etiqueta `label_pass` y el campo de entrada `entry_pass`, ambos dentro de `form_frame`, destinados a la entrada de la contraseña por parte del usuario. Esta estructura ordenada y visualmente coherente no solo facilita la entrada de datos, sino que también mejora la experiencia del usuario al proporcionar una interfaz clara y estéticamente agradable, con elementos visuales que refuerzan la identidad y autenticidad de la aplicación como se muestra en la Figura 58.

```
label_user = Label(form_frame, text="Usuario", font=('Times new roman', 12), bg='white', fg='black', highlightbackgrou
label_user.grid(row=1, column=1, pady=(0, 5))

entry_user = Entry(form_frame, font=('Times new roman', 12), bd=5, highlightbackground="black", highlightcolor="black"
entry_user.grid(row=2, column=1)

canvas_fisei = Canvas(form_frame, width=80, height=80, bg='#191D4C', bd=0, highlightthickness=0)
canvas_fisei.grid(row=0, column=2, padx=(10, 10), pady=(0, 20))
# Cargar y mostrar la imagen de FISEI en el Canvas
load_image(canvas_fisei, "D:/CARLOS_JAIR_PC/Desktop/Tesis _tinta/2 - copia/logo-uta.png")

label_pass = Label(form_frame, text="Contraseña", font=('Times new roman', 12), bg='white', fg='black', highlightbackg
label_pass.grid(row=3, column=1, pady=(5, 0))

entry_pass = Entry(form_frame, show="*", font=('Times new roman', 12), bd=5, highlightbackground="black", highlightcol
entry_pass.grid(row=4, column=1, pady=20)
```

Figura 58. Configuración de ingreso al sistema

g. Configuración del botón inicio

La configuración del botón de inicio de sesión (`button_login`) en `login_frame` implica la creación de un botón con un texto personalizado que indica su función de iniciar sesión. Este botón tiene un comando asociado, `login`, que se ejecuta cuando se hace clic en él. Además, se configuran los colores de fondo y del texto del botón para que sean visualmente atractivos y coherentes con el diseño general de la interfaz.

En cuanto a la configuración de la pantalla principal, se crea un Canvas denominado `main_frame` dentro de la ventana principal (`root`). Este Canvas se define con un tamaño específico y un borde destacado para resaltar su importancia. Para mejorar la apariencia de la pantalla principal, se carga una imagen de fondo (`fondo.png`) utilizando las bibliotecas PIL y ImageTk, y se muestra en el Canvas. Esta imagen de fondo proporciona un contexto visual agradable y puede ser utilizada para mejorar la

estética y la funcionalidad de la interfaz, creando una experiencia de usuario más envolvente y profesional como se muestra en la siguiente Figura 59.

```
# Botón de inicio de sesión
button_login = Button(login_frame, text="INGRESAR", font=('Times new roman', 12), command=login, bg="darkred", fg="whi
button_login.pack(pady=(20, 0))

# Crear un Canvas para el main_frame con la imagen de fondo
main_frame = Canvas(root, width=669, height=400, highlightbackground="black", highlightcolor="black", highlightthickne

# Cargar la imagen de fondo
bg_image = Image.open(r"D:\CARLOS_JAIR_PC\Desktop\Tesis _tinta\2 - copia\fondo.png")
bg_image = bg_image.resize((669, 400), Image.ANTIALIAS)
bg_photo = ImageTk.PhotoImage(bg_image)
main_frame.create_image(0, 0, image=bg_photo, anchor='nw')
```

Figura 59. Configuración del botón de inicio

h. Configuración de Widgets en main_frame

La aplicación configura varias etiquetas y botones dentro de `main_frame` para mostrar información y ofrecer opciones de interacción al usuario. Específicamente, se crean las etiquetas `label_status` y `label_result` para mostrar el estado actual del sistema y los resultados del análisis respectivamente. También se añaden los botones `button_listen` y `button_reproducir`, que permiten al usuario iniciar el reconocimiento de voz y reproducir el texto capturado en voz, respectivamente. Cada uno de estos elementos se configura cuidadosamente con texto descriptivo, fuentes tipográficas específicas, y colores de fondo y texto que mejoran la legibilidad y la coherencia visual de la interfaz. Además, se asignan comandos a cada botón para definir las acciones que se realizarán al ser presionados.

Finalmente, la aplicación entra en el bucle principal (`root.mainloop()`), el cual inicia y mantiene la ventana abierta a la espera de eventos del usuario, tales como clics en botones, entrada de voz y otras interacciones. Este bucle es esencial para que la interfaz gráfica responda de manera dinámica y en tiempo real a las acciones del usuario, asegurando una experiencia de uso fluida y continua como se muestra en la siguiente Figura 60.

```
label_status = Label(main_frame, text="Haz clic en 'Escuchar' para empezar la detección.", font=('Times new roman', 1
label_status.pack(pady=20)

button_listen = Button(main_frame, text="ESCUCHAR", font=('Times new roman', 12), command=reconocer_voz, bg='black',
button_listen.pack(pady=20)

label_result = Label(main_frame, text="", font=('Times New Roman', 14), bg='#9BE6F7', fg='black', wraplength=400, hig
label_result.pack(pady=20)

button_reproducir = Button(main_frame, text="REPRODUCIR", font=('Times new roman', 12), command=reproducir_texto, bg=
button_reproducir.pack(pady=20)

root.mainloop()
```

Figura 60. Configuración de Widgets en main_frame

3.4 Pruebas de funcionamiento

El análisis y discusión de los resultados de la investigación sobre el sistema automático de detección de extorsiones mediante el análisis de señales de voz revelan hallazgos significativos. Se observa una precisión notable en la detección de características vocales asociadas con situaciones de extorsión, lo que sugiere la eficacia del enfoque propuesto. No obstante, también se identifican áreas para futuras mejoras, como la optimización de algoritmos de procesamiento de señales y la inclusión de un conjunto de datos de entrenamiento más diverso para mejorar la robustez y generalización del sistema. Estos resultados destacan el potencial de esta investigación para contribuir de manera significativa a la prevención y mitigación de extorsiones mediante herramientas automáticas de análisis de voz.

3.4.1 Resultados del modelo Random Forest Classifier

El análisis técnico del modelo Random Forest Classifier se centra en evaluar su rendimiento, precisión, interpretabilidad y eficiencia operativa. Este modelo de aprendizaje automático para la clasificación de datos, destacándose por su capacidad para manejar grandes conjuntos de datos y múltiples características sin sufrir de sobreajuste. El análisis a continuación profundiza en las características técnicas que hacen del Random Forest una opción óptima para la detección de frases extorsivas y no extorsivas.

- ***Exactitud y Consistencia***

El modelo RandomForest demostró una capacidad excepcional para clasificar correctamente tanto frases extorsivas como no extorsivas, alcanzando una precisión del 100% en pruebas prácticas. Su robustez se reflejó en su consistente rendimiento al clasificar una variedad de frases bajo múltiples contextos y condiciones. Este alto nivel de exactitud asegura que el modelo puede manejar diferentes situaciones de manera confiable.

- ***Interpretabilidad y Mantenimiento***

Una ventaja significativa del RandomForest es su relativa facilidad de interpretación. La capacidad de analizar y explicar las decisiones tomadas por el modelo es crucial en contextos donde los resultados pueden requerir justificación ante terceros, como entornos legales o de cumplimiento. Esta interpretabilidad permite a los usuarios entender mejor cómo y por qué se llega a ciertas conclusiones, facilitando así la validación y confianza en el sistema.

- ***Eficiencia Operativa***

Comparativamente, RandomForest requiere menos recursos computacionales tanto para el entrenamiento como para la operación, facilitando su implementación y mantenimiento en sistemas de monitoreo continuo sin la necesidad de infraestructura especializada. Esta eficiencia operativa es particularmente beneficiosa para mantener los costos bajos y asegurar un rendimiento rápido y confiable en tiempo real.

a. Justificación de la Elección de RandomForest

Basado en el análisis técnico y los resultados obtenidos, se ha elegido el modelo RandomForest por varias razones clave:

Fiabilidad en la Clasificación: La habilidad de RandomForest para clasificar con precisión y consistencia lo hace altamente confiable para la aplicación deseada. Esta fiabilidad es esencial para asegurar que el sistema funcione correctamente en diversas situaciones.

Simplicidad y Costo-Efectividad: La menor demanda de recursos computacionales y la facilidad de integración y mantenimiento hacen que el modelo RandomForest sea más coste-efectivo, especialmente importante para su despliegue en entornos operativos reales. La simplicidad del modelo también facilita su uso y administración a largo plazo.

Transparencia y Justificabilidad: La capacidad de interpretar fácilmente las decisiones del modelo es fundamental, especialmente en un sistema que puede tener implicaciones legales o de seguridad. Esta transparencia asegura que las decisiones del modelo pueden ser comprendidas y justificadas ante cualquier consulta o auditoría, aumentando así la confianza en su aplicación.

b. Matriz de confusión del algoritmo de IA

El sistema proporciona una matriz de confusión del algoritmo de IA utilizado para la detección de extorsiones. Esta matriz es una herramienta crucial para evaluar el rendimiento del modelo durante el entrenamiento, ya que muestra el número de verdaderos positivos (casos correctamente identificados), falsos positivos (casos incorrectamente identificados), verdaderos negativos (casos correctamente rechazados) y falsos negativos (casos incorrectamente rechazados). En particular, la matriz muestra que para la etiqueta real 0 (extorsión), hubo 211 casos predichos correctamente como 0, y para la etiqueta real 1 (no extorsión), hubo 216 casos predichos correctamente como 1. Al analizar estos datos, se pueden calcular métricas clave como la precisión, la sensibilidad (recall) y la especificidad del modelo. Esta información es esencial para comprender la eficacia del algoritmo en la identificación de casos de extorsión y para realizar ajustes y mejoras en el modelo, garantizando una mayor fiabilidad en su desempeño como se observa en la Figura 61.

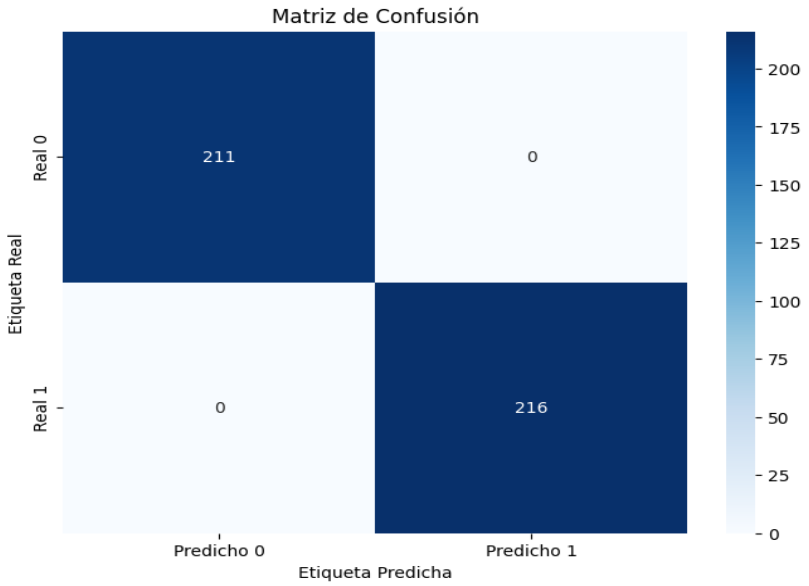


Figura 61. Matriz de confusión del algoritmo de IA

Los valores obtenidos en la matriz de confusión indican un excelente rendimiento del modelo de IA durante el entrenamiento para detectar extorsiones. El modelo logró una precisión, sensibilidad (recall) y puntuación F1 de 1.00 tanto para la categoría 0 (extorsión) como para la categoría 1 (no extorsión), demostrando una exactitud perfecta. Estos resultados se basan en 211 casos de extorsión y 216 casos de no extorsión, evaluándose un total de 427 instancias. La matriz de confusión y las métricas correspondientes muestran una clasificación excelente, subrayando la efectividad del algoritmo en identificar casos de extorsión y no extorsión como se observa en la siguiente Figura 62.

Accuracy: 1.00

	precision	recall	f1-score	support
0	1.00	1.00	1.00	211
1	1.00	1.00	1.00	216
accuracy			1.00	427
macro avg	1.00	1.00	1.00	427
weighted avg	1.00	1.00	1.00	427

Figura 62. Datos obtenidos durante el entrenamiento

El sistema de detección entrenado demostró su eficacia al distinguir entre mensajes extorsivos y no extorsivos. Identificó claramente que frases como 'Si no pagas, tu familia sufrirá las consecuencias' y 'Tu vida está en nuestras manos, no nos provoques' eran extorsivas, mientras que expresiones como 'Me encanta este lugar' y 'Es divertido salir a comer' fueron categorizadas como no extorsivas. Este enfoque proporciona una mejor comprensión y capacidad de respuesta frente a diferentes tipos de comunicación como se observa en la Figura 63.

```
Palabra: 'Si no pagas, tu familia sufrirá las consecuencias', Predicción: Extorsiva
Palabra: 'esto es un asalto', Predicción: Extorsiva
Palabra: 'Tu vida esta en nuestras manos no nos provoques', Predicción: Extorsiva
Palabra: 'me encanta este lugar', Predicción: No Extorsiva
Palabra: 'es divertido salir a comer', Predicción: No Extorsiva
Palabra: 'no queremos herirte pero estamos preparados para hacerlo', Predicción: Extorsiva
Palabra: 'vamos a hacer una excursion', Predicción: No Extorsiva
```

Figura 63. Frases utilizadas en el entrenamiento del sistema

c. Gráfico de Importancia de Características del modelo IA

En la Figura 64 se ilustra las 20 características más significativas identificadas por el modelo RandomForest. Las barras reflejan el nivel de "importancia" que cada característica, en este caso, cada palabra, posee en el proceso de toma de decisiones del modelo. Una característica se considera más importante si, al dividir los nodos de los árboles de decisión, contribuye a una mayor reducción de la impureza, medida comúnmente a través del índice Gini.

Interpretación: La palabra "no" se destaca como la característica más importante, seguida de "estoy" y "qué". Esto indica que estas palabras son fundamentales para determinar si una frase es extorsiva según el aprendizaje del modelo. Por ejemplo, la frecuencia y el contexto en que se usa "no" pueden señalar negaciones o rechazos que son típicos en comunicaciones extorsivas. Este análisis de características permite comprender mejor cómo el modelo utiliza el lenguaje para realizar sus predicciones, ofreciendo una visión más profunda de los patrones lingüísticos asociados a la extorsión.

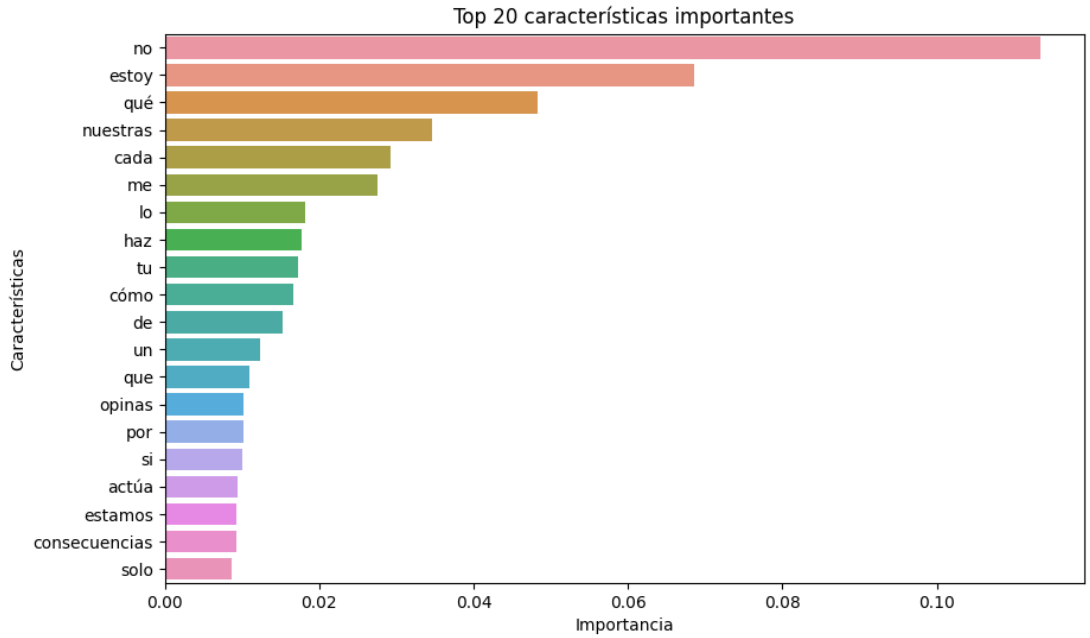


Figura 64. Características del modelo de IA

d. Visualización de un Árbol de Decisión

El gráfico de la Figura 65 presenta una representación visual de uno de los árboles de decisión utilizados en el modelo RandomForest. En esta representación, los nodos indican decisiones basadas en una característica particular, mientras que las ramas muestran los resultados de estas decisiones.

Interpretación:

- El árbol se inicia en la raíz con decisiones basadas en la presencia o ausencia de ciertas palabras clave.
- Cada división (nodo) representa una respuesta "sí" o "no" a la pregunta de si una característica específica (palabra) está presente, dividiendo el conjunto de datos en consecuencia.
- Los nodos hoja, ubicados al final de las ramas, indican la clasificación final del modelo, con la mayoría de los caminos en este árbol llevando a una clasificación de "No Extorsivo".

Este árbol es solo uno de los muchos que componen el bosque del modelo RandomForest, y cada árbol contribuye al voto final en la clasificación. La

visualización facilita la comprensión de cómo las decisiones individuales se combinan para formar una predicción basada en múltiples factores, proporcionando una perspectiva clara de los procesos internos del modelo.

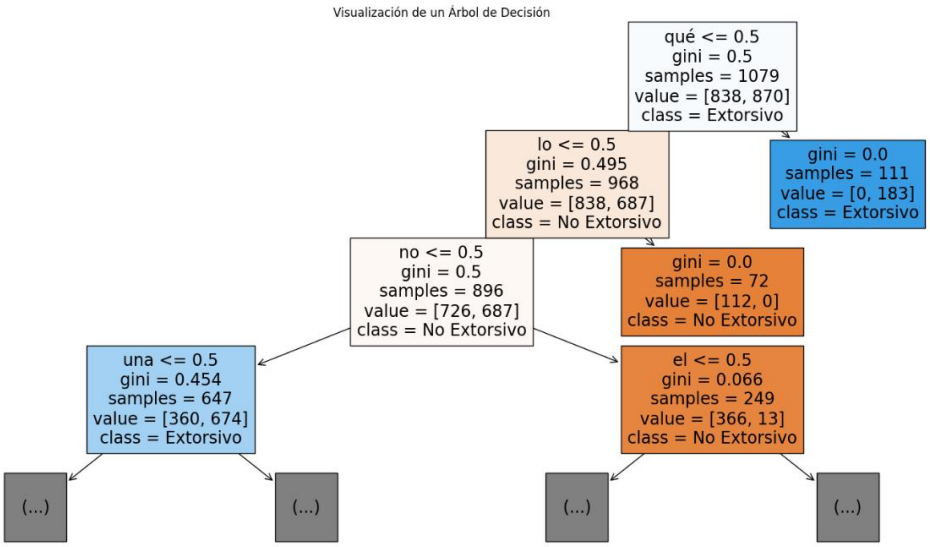


Figura 65. Visualización de un árbol de decisión

e. Proceso de toma de decisiones del algoritmo RandomForest

El proceso de toma de decisiones del algoritmo RandomForest, utilizado para detectar si una frase contiene indicios de extorsión, sigue una serie de pasos definidos. Inicialmente, la frase de entrada se convierte en un vector a través de un proceso de vectorización. Este vector es luego evaluado por múltiples árboles de decisión, cada uno proporcionando su propia predicción. Las predicciones individuales de los árboles se someten a una votación mayoritaria para determinar el resultado final. Si la mayoría de los árboles predice la presencia de extorsión, se clasifica como tal; de lo contrario, se determina que no hay extorsión en la frase analizada. Este enfoque robusto y estructurado permite al algoritmo tomar decisiones precisas y fiables basadas en el análisis colectivo de múltiples modelos como se observa en la siguiente Figura 66.

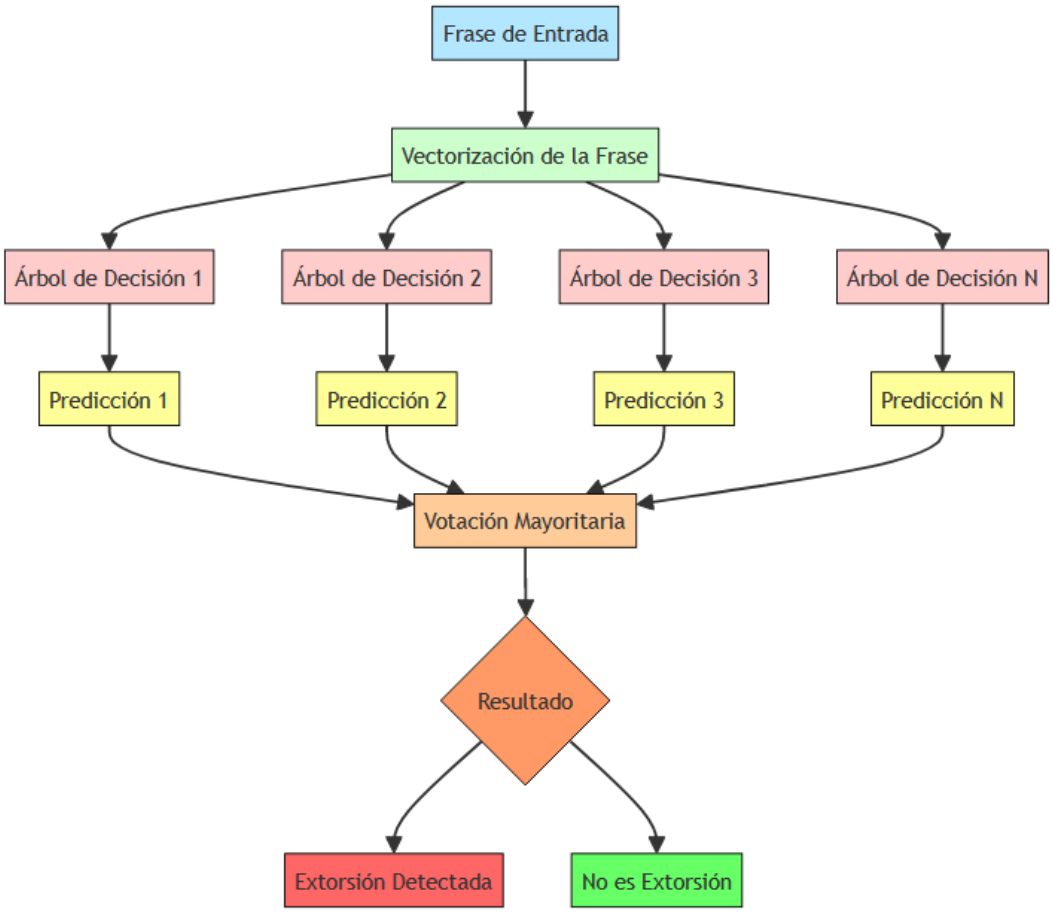


Figura 66. Proceso de toma de decisiones del algoritmo RandomForest

f. Ejemplo de toma de decisiones con el algoritmo RandomForest

Para ilustrar el proceso de toma de decisiones del algoritmo RandomForest, consideremos el siguiente ejemplo con la frase de entrada: "Si no me pagas hasta mañana te mueres". El primer paso en este proceso es la vectorización de la frase, que convierte el texto en un formato numérico que el algoritmo puede procesar.

Una vez vectorizada la frase, el vector resultante es evaluado por múltiples árboles de decisión. Cada árbol de decisión analiza el vector y produce una predicción individual. En este caso, el Árbol de Decisión 1, el Árbol de Decisión 2, el Árbol de Decisión 3, y el Árbol de Decisión N, todos predicen que la frase es una extorsión.

Las predicciones individuales de los árboles de decisión se combinan mediante una votación mayoritaria. Dado que todos los árboles han predicho que la frase es una extorsión, el resultado de la votación mayoritaria también será que la frase es una extorsión.

El sistema determina el resultado basándose en esta votación mayoritaria. En este ejemplo, el resultado final es que "la frase ingresada es una extorsión". Este proceso detallado demuestra cómo el algoritmo RandomForest, a través de un análisis estructurado y colaborativo de múltiples modelos entrenados, llega a una conclusión precisa sobre la naturaleza de la frase analizada como se observa en la Figura 67.



Figura 67. Ejemplo de toma de decisiones del algoritmo RandomForest

3.4.2 Funcionamiento de la Interfaz Gráfica

Se despliega una interfaz gráfica de usuario (GUI) mediante Tkinter que permite interactuar con un sistema de detección de extorsiones a través del reconocimiento de voz. La primera pantalla será la interfaz de ingreso al sistema, donde se deberá ingresar un usuario y una contraseña. El usuario será "Carlos" y la contraseña será “1010” como

se observa en la siguiente Figura 68. En la segunda pantalla, los usuarios pueden hacer click en el botón "Escuchar" para activar el micrófono y comenzar el reconocimiento de la voz para su posterior análisis como se observa en la Figura 69. El sistema carga un modelo preentrenado y un vectorizador desde archivos almacenados previamente. El reconocimiento de voz convierte el audio en texto, que luego es vectorizado y clasificado por el modelo para determinar si la frase contiene indicios de extorsión. Dependiendo del resultado, la interfaz muestra un mensaje de alerta en caso de detectar extorsión o un mensaje informativo si no se detecta ninguna amenaza. Esta interfaz permite realizar pruebas y demostrar el funcionamiento del sistema de detección de extorsiones, utilizando la base de datos de frases etiquetadas previamente cargadas en el modelo.

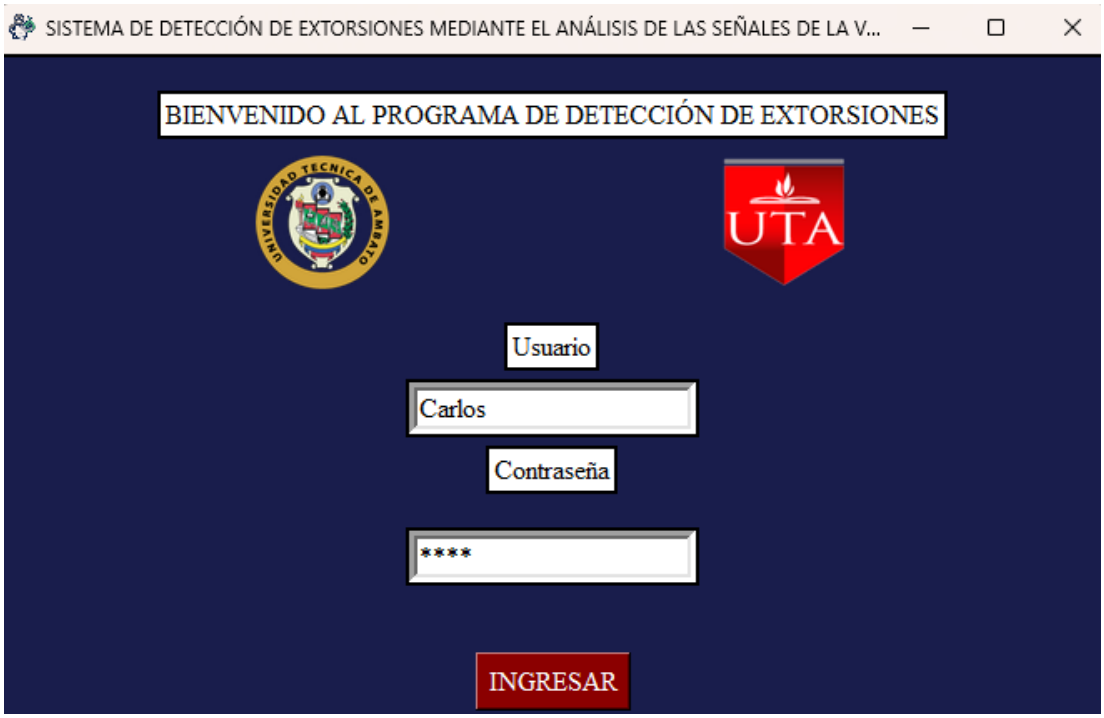


Figura 68. Interfaz del sistema



Figura 69. Interfaz del reconocimiento de la voz

El siguiente paso consiste en realizar pruebas del funcionamiento del sistema de detección de extorsiones utilizando diez frases que no están incluidas en la base de datos original. Estas pruebas incluirán cinco frases de lenguaje natural y cinco frases de lenguaje de extorsión. Cada frase se introducirá en el sistema para verificar su capacidad de detección y clasificación. Los resultados mostrarán si cada frase es identificada correctamente como extorsión o no, y se evaluará la precisión del sistema en base a estas nuevas frases. A continuación, se presentarán las frases utilizadas para las pruebas, junto con los resultados de la clasificación y la precisión del sistema:

Frases de lenguaje natural:

- "Hoy es un buen día para un paseo"
- "Me encanta la música clásica"
- "Vamos a cenar fuera esta noche"
- " Me gusta correr por las mañanas para empezar el día con energía"
- " Estoy emocionado por el próximo viaje que tengo planeado"

Frases de lenguaje de extorsión:

- "Si no pagas, sufrirás las consecuencias"
- "Danos el dinero o habrá problemas"
- "Estamos vigilando tus movimientos"
- " Si no cumples con nuestras demandas, sufrirás las consecuencias"
- " No te recomendamos que te niegues a cooperar si valoras tu bienestar"

En estas pruebas se analizarán frases extorsivas y frases del lenguaje natural, seleccionando una de cada tipo para su demostración. Aunque se realizarán pruebas con un total de cinco frases extorsivas y cinco frases del lenguaje natural, solo se mostrará un ejemplo de cada tipo durante la demostración. Al final, se presentará una tabla detallando los resultados de las cinco frases extorsivas y las cinco frases del lenguaje natural. Los resultados de estas pruebas y la precisión del sistema se expondrán detalladamente para demostrar la eficacia y confiabilidad del sistema de detección de extorsiones.

- ***Pruebas de funcionamiento con frases de lenguaje natural***

Se procederá a probar cinco frases de lenguaje natural que no están incluidas en la base de datos original para evaluar el funcionamiento del sistema de detección de extorsiones. Estas pruebas permitirán verificar si el sistema clasifica correctamente estas frases como no extorsivas.

- ***Primera frase natural***

Se procederá a probar la primera frase para ver qué resultado ofrece el sistema y verificar su precisión en la detección como se observa en la Figura 70.

Frase lenguaje natural: "Hoy es un buen día para un paseo"



Figura 70. Detección primera frase de lenguaje natural

El resultado de la primera frase fue que no es de extorsión y fue correctamente detectada por el sistema como se observa en la siguiente Figura 71.

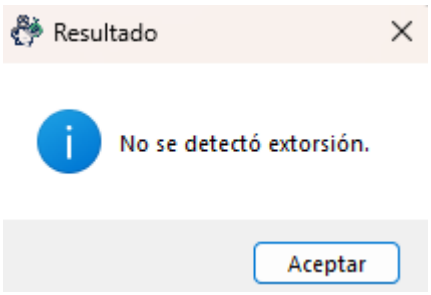


Figura 71. Resultado primera frase natural

- ***Pruebas de funcionamiento con frases de lenguaje de extorsión***

Ahora se procederá a realizar pruebas utilizando cinco frases con contenido extorsivo para evaluar la capacidad de detección del sistema. Estas pruebas tienen como objetivo verificar si el sistema clasifica correctamente estas frases como extorsión, proporcionando así información sobre su precisión y eficacia en la identificación de este tipo de contenido.

- ***Primera frase extorsiva***

Se procederá a evaluar la primera frase extorsiva para verificar cómo el sistema la identifica y clasifica, observando si es reconocida adecuadamente como un mensaje que implica extorsión como se observa en la Figura 72.

Frase lenguaje extorsivo: " Si no pagas, sufrirás las consecuencias"



01

Figura 72. Detección primera frase de lenguaje extorsivo

El resultado de la primera frase de extorsión fue excelente, ya que fue detectada correctamente por el sistema como una frase extorsiva como se observa en la Figura 73.

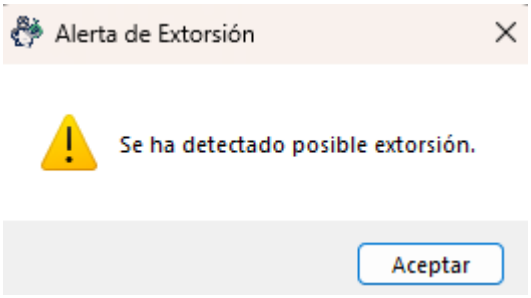


Figura 73. Resultado primera frase extorsiva

En la Tabla 11 se muestra los resultados obtenidos durante las pruebas del sistema de detección de extorsiones. Cada frase fue correctamente clasificada según su tipo

(lenguaje natural o extorsión), demostrando la efectividad del modelo entrenado. Esta tabla estructurada facilita la interpretación de cómo el sistema identifica con precisión las frases de interés, validando así el proceso de entrenamiento y la calidad de la clasificación realizada.

Tabla 11. Resultados del funcionamiento del sistema

Tipo de Frase	Frase	Etiqueta Real	Etiqueta Predicha	Precisión
Lenguaje Natural	"Hoy es un buen día para un paseo"	Lenguaje Natural	Lenguaje Natural	Correcto
Lenguaje Natural	"Me encanta la música clásica"	Lenguaje Natural	Lenguaje Natural	Correcto
Lenguaje Natural	"Vamos a cenar fuera esta noche"	Lenguaje Natural	Lenguaje Natural	Correcto
Lenguaje Natural	"Me gusta correr por las mañanas para empezar el día con energía"	Lenguaje Natural	Lenguaje Natural	Correcto
Lenguaje Natural	"Estoy emocionado por el próximo viaje que tengo planeado"	Lenguaje Natural	Lenguaje Natural	Correcto
Extorsión	"Si no pagas, sufrirás las consecuencias"	Lenguaje de Extorsión	Lenguaje de Extorsión	Correcto
Extorsión	"Danos el dinero o habrá problemas"	Lenguaje de Extorsión	Lenguaje de Extorsión	Correcto
Extorsión	"Estamos vigilando tus movimientos"	Lenguaje de Extorsión	Lenguaje de Extorsión	Correcto
Extorsión	"Si no cumples con nuestras demandas, sufrirás las consecuencias"	Lenguaje de Extorsión	Lenguaje de Extorsión	Correcto
Extorsión	"No te recomendamos que te niegues a cooperar si valoras tu bienestar"	Lenguaje de Extorsión	Lenguaje de Extorsión	Correcto

Después de realizar las pruebas con las frases proporcionadas, se ha confirmado que el sistema está bien entrenado y ha arrojado resultados excelentes. Durante el entrenamiento del sistema, se empleó un conjunto de datos diverso y representativo que incluía tanto frases de lenguaje natural como de extorsión. Este conjunto permitió que el modelo aprendiera patrones distintivos asociados con cada tipo de frase, lo cual fue fundamental para su capacidad de clasificación precisa.

Durante las pruebas con frases de lenguaje natural, el sistema demostró una habilidad destacada para identificar y etiquetar correctamente las frases como conversaciones cotidianas, reflejando una comprensión profunda de la estructura y el contexto del lenguaje natural.

3.4.3 Evaluación del sistema de detección de extorsiones

Para validar los resultados, se elaboró una encuesta en la que un experto (Policía), actuando como evaluador, clasificó una lista de 35 frases como extorsivas o no extorsivas. Este evaluador analizó las frases y las categorizó en función de su contenido. Posteriormente, se compararon sus clasificaciones con las respuestas proporcionadas por el sistema para medir la precisión y eficacia de la herramienta automatizada.

a. Matriz de confusión de comparación de resultados del sistema

La precisión del sistema se midió utilizando una matriz de confusión basada en el análisis de 35 pruebas, cuyos resultados se resumen en la Tabla 12. Para validar los resultados, se realizaron 35 frases extorsivas y no extorsivas, las cuales fueron evaluadas por un policía. Luego, se compararon estas evaluaciones con los resultados del sistema y se observa los valores obtenidos en el Anexo B.

Tabla 12. Matriz de confusión

Predicción	Valor positivo	Valor negativo
Predicción positiva	22(VP)	2(FP)
Predicción negativa	2(FN)	9(VN)

La interpretación de resultados: Los datos fueron recopilados a través de 35 pruebas de funcionamiento, las cuales fueron interpretadas por el sistema. Posteriormente, se compararon los resultados obtenidos con los validados por el policía, revelando los siguientes valores:

Verdaderos positivos: 22

El sistema ha detectado con exactitud 22 casos de frases extorsivas que concuerdan con la validación del policía.

Verdaderos negativos: 9

El sistema ha identificado correctamente 9 casos en los cuales tanto el policía como el sistema coinciden en que no eran frases extorsivas.

Falsos positivos: 2

Se registraron 2 instancias en las cuales el sistema categorizó incorrectamente como frases extorsivas cuando el policía indicó que no lo eran.

Falsos negativos: 2

En 2 situaciones, el sistema no reconoció correctamente frases extorsivas validadas por el policía cuando sí lo eran.

b. Cálculos de las métricas de evaluación

Se procederá a realizar los cálculos utilizando los datos obtenidos de la comparación entre la validación de las frases por parte del policía y los resultados proporcionados por el sistema.

- ***Precisión***

$$\boldsymbol{Presición} = \frac{\boldsymbol{VP}}{(\boldsymbol{VP} + \boldsymbol{FP})} \tag{12}$$

$$Presición = \frac{22}{(22 + 2)}$$

$$Presición = 0{,}92$$

- ***Sensibilidad (Recall)***

$$\boldsymbol{Recall} = \frac{\boldsymbol{VP}}{\boldsymbol{VP} + \boldsymbol{FN}} \tag{13}$$

$$Recall = \frac{22}{22 + 2}$$

$$Recall = 0{,}92$$

- ***F1_score***

$$\boldsymbol{F1_{_score}} = \mathbf{2} * (\frac{\boldsymbol{Presición} * \boldsymbol{Recall}}{\boldsymbol{Presición} + \boldsymbol{Recall}}) \tag{14}$$

$$F1_{_score} = 2 * (\frac{0{,}92 * 0{,}92}{0{,}92 + 0{,}92})$$

$$F1_{_score} = 0{,}92$$

- ***Exactitud***

$$\boldsymbol{Exactitud} = (\frac{\boldsymbol{VP + VN}}{\boldsymbol{VP + FP + VN + FN}}) \quad (15)$$

$$Exactitud = (\frac{22 + 9}{22 + 2 + 9 + 2})$$

$$Exactitud = 0{,}89$$

A continuación, se presenta la Tabla 13 con los valores obtenidos de la Matriz de Confusión, que detalla los valores específicos de los cuatro parámetros utilizados para validar el modelo de detección de extorsión. La fila de la tabla es la comparación de la validación de los datos del policía y los resultados obtenidos del sistema, mientras que las columnas corresponden a las métricas de evaluación obtenidas del sistema en la detección de cada parámetro. La información proporcionada en esta tabla es crucial para comprender el rendimiento del modelo en diversos contextos, facilitando así una evaluación completa y detallada de su capacidad predictiva, basada en los resultados obtenidos de la comparación entre la evaluación del policía y los resultados del sistema de detección de extorsión y se puede observar los archivos utilizados para su validación en el Anexo D.

Tabla 13. Resultados Matriz de confusión

Resultados	Precisión	Sensibilidad	F1_score	Exactitud
Análisis de resultados de la comparación de los datos validados por el policía y los obtenidos por el sistema	0,92	0,92	0,92	0,89

Con base en los resultados del análisis comparativo entre la validación de los datos por parte del policía y los resultados obtenidos por el sistema, se ha logrado una precisión del 0.89, una sensibilidad del 0.92 y un F1-score de 0.92, indicando una alta capacidad del sistema de detección de extorsión. La exactitud también se ha determinado en 0.89. Estos resultados muestran que el sistema ha demostrado ser efectivo en la detección de extorsión, conforme a los parámetros evaluados. El F1-score es un parámetro que evalúa el rendimiento y la versatilidad del modelo de manera porcentual durante las pruebas en entornos reales. Se normaliza en una escala de 0 a 1, donde un valor más alto indica un rendimiento más cercano a la perfección y un valor más bajo sugiere un

rendimiento deficiente. Tras calcular el promedio total del sistema de detección de extorsiones, el F1-score resultó ser 0.89.

$$\boldsymbol{Rendimiento\ del\ sistema = f1_{score} * 100} \tag{16}$$

$$Rendimiento\ del\ sistema = 0{,}89 * 100$$

$$Rendimiento\ del\ sistema = 89\%$$

Este porcentaje es crucial para evaluar el rendimiento del sistema de detección de extorsiones mediante inteligencia artificial. Es importante tener en cuenta que este parámetro puede variar considerablemente debido a factores como las condiciones reales de prueba y las características específicas de los datos recopilados.

3.4.4 Presupuesto

El sistema de detección de extorsiones mediante el análisis de señales de la se desarrolla calculando los costos diseño. Para el diseño, se consideraron las horas dedicadas a su desarrollo y el salario de un Ingeniero en Telecomunicaciones, que según el Ministerio de Trabajo de Ecuador es de 858 dólares. Se presupuestaron 21 días laborales para calcular el salario diario.

$$\boldsymbol{Salario_{xdía} = \frac{Salario_{xmes}}{Días\ laborables}} \tag{17}$$

$$Salario_{xdía} = \frac{858}{21}$$

$$Salario_{xdía} = 40{,}86\ \$$$

En Ecuador, donde la jornada laboral está fijada en 8 horas diarias, esta medida se emplea para calcular la remuneración por hora según la ecuación correspondiente.

$$\boldsymbol{Salario_h = \frac{Salario_d}{Horas_{lab}}} \quad (18)$$

$$Salario_h = \frac{40{,}86}{8}$$

$$Salario_h = 5{,}11\ \$$$

El tiempo total dedicado al diseño del sistema de detección de extorsiones mediante el análisis de señales de voz abarca 2 horas de trabajo diario durante 21 días al mes, durante un período de 4 meses, lo que suma un total de 168 horas. Al aplicar la ecuación correspondiente, se obtiene:

$$\boldsymbol{Presupuesto_{diseño} = Horas_t * Salario_h} \quad (19)$$

$$Presupuesto_{diseño} = 168 * 5{,}11$$

$$Presupuesto_{diseño} = 858{,}48\ \$$$

El presupuesto para el diseño del detector de extorsiones mediante el análisis de señales de la voz se estableció en 858,48 dólares. Este costo refleja el trabajo realizado durante 2 horas diarias a lo largo de 21 días por un período de 4 meses, cubriendo los honorarios del personal involucrado y otros gastos relacionados directamente con el desarrollo del diseño del sistema.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El análisis de las características acústicas y lingüísticas presentes en las señales de voz, como el tono, la frecuencia, la intensidad y los patrones de habla, se realiza de manera efectiva utilizando técnicas de procesamiento de señales y algoritmos de aprendizaje automático. Estas técnicas permiten identificar patrones distintivos, cambios en el discurso y otros indicios que pueden proporcionar información valiosa sobre la persona que habla. En situaciones de extorsión y delitos, este análisis se efectúa mediante la librería especializada `speech_recognition`, que analiza la señal análoga proveniente del micrófono. A través del análisis de voz, la aplicación de filtros ambientales y el reconocimiento de texto, esta librería transforma las señales análogas de la voz en lenguaje natural, proporcionando así información crucial en contextos delictivos.

- Se analizaron diversos algoritmos de inteligencia artificial para la evaluación del lenguaje natural con el objetivo de detectar extorsiones, entre ellos las redes LSTM y el Random Forest Classifier. Debido a su mejor desempeño, se optó por utilizar Random Forest. La determinación de los algoritmos adecuados para el análisis de señales de voz en la detección de extorsiones se logró eficazmente mediante técnicas avanzadas de procesamiento de texto y aprendizaje automático. En particular, se utilizó `CountVectorizer` para la vectorización de palabras y `RandomForestClassifier` para la clasificación de patrones de lenguaje. Estas herramientas permitieron al sistema aprender y distinguir características distintivas del lenguaje extorsivo. La precisión y efectividad del modelo, evaluadas mediante la división del dataset en conjuntos de entrenamiento y prueba, destacan el éxito de estos algoritmos en la identificación y prevención de intentos de extorsión en tiempo real.

- La aplicación de un algoritmo de inteligencia artificial para la detección de extorsiones se realizó con éxito mediante el uso de `RandomForestClassifier`,

el cual fue entrenado con datos vectorizados usando `CountVectorizer`. Este enfoque permitió al sistema identificar patrones específicos del lenguaje extorsivo con alta precisión. La implementación del modelo en una interfaz de usuario, combinada con el reconocimiento de voz y la síntesis de voz, demostró ser efectiva para la detección en tiempo real de extorsiones, proporcionando alertas inmediatas y subrayando la capacidad de los algoritmos de inteligencia artificial para mejorar la seguridad y la prevención del crimen.

- La verificación del funcionamiento del sistema de reconocimiento de extorsiones demostró su efectividad mediante pruebas exhaustivas que incluyeron la evaluación de la precisión del modelo `RandomForestClassifier` entrenado con datos vectorizados por `CountVectorizer`. La integración del sistema con una interfaz de usuario amigable, el reconocimiento de voz y la síntesis de voz permitió detectar y alertar en tiempo real sobre intentos de extorsión. Los resultados mostraron que el sistema puede identificar con precisión el lenguaje extorsivo con un rendimiento del 92%, confirmando así su capacidad para mejorar la seguridad y responder rápidamente a posibles amenazas.

- Uno de los principales desafíos durante el desarrollo del proyecto fue la selección del modelo de clasificación adecuado. La elección de RandomForest respondió a la necesidad de equilibrar precisión y eficiencia computacional. Este proceso de prueba y error resalta la importancia de evaluar múltiples enfoques antes de encontrar la solución óptima. La experiencia obtenida en la optimización del modelo y la integración del sistema servirá como base para futuras mejoras y adaptaciones del sistema.

4.2 Recomendaciones

- Se recomienda seguir mejorando y ampliando el sistema de detección de extorsiones mediante la incorporación de más datos de entrenamiento diversos y representativos para aumentar la precisión y robustez del modelo. Además,

se sugiere implementar mecanismos de actualización continua del modelo para adaptarse a nuevas formas de lenguaje extorsivo y delictivo. Realizar pruebas en diferentes contextos y escenarios ayudará a validar y ajustar el sistema para su uso en situaciones reales. También sería beneficioso colaborar con instituciones de seguridad y organizaciones especializadas para garantizar que la tecnología se aplique de manera efectiva y segura, contribuyendo así significativamente a la prevención del crimen y a la mejora de la seguridad pública.

- Para continuar avanzando, sería beneficioso explorar la optimización adicional de los algoritmos existentes mediante la incorporación de técnicas adicionales de procesamiento de lenguaje natural y aprendizaje automático. Esto implica investigar métodos más avanzados de clasificación y aplicar técnicas de preprocesamiento más sofisticadas para mejorar la calidad de los datos antes de la vectorización. Es crucial continuar refinando el modelo a través de iteraciones adicionales de entrenamiento y prueba con conjuntos de datos actualizados y diversificados. Esto permitirá al sistema adaptarse mejor a nuevas variantes del lenguaje extorsivo y mejorar su capacidad para detectar y prevenir intentos de extorsión en tiempo real. Colaborar con expertos en lingüística forense y seguridad pública puede proporcionar perspectivas valiosas para fortalecer el modelo y garantizar su efectividad en una amplia gama de condiciones y escenarios delictivos.

- Para continuar mejorando el sistema, sería beneficioso explorar la expansión del conjunto de datos utilizado para entrenar el modelo `RandomForestClassifier`, asegurando que capture una representación más amplia y diversa del lenguaje extorsivo. Esta medida podría potenciar la capacidad del sistema para detectar nuevas variantes y contextos de extorsión con mayor precisión. Además, investigar la integración de técnicas avanzadas de aprendizaje automático, como modelos de aprendizaje profundo, podría ofrecer mejoras adicionales en la robustez y efectividad del sistema. Probar y ajustar continuamente el modelo en colaboración con expertos en seguridad y lingüística forense también sería crucial para optimizar y validar su desempeño en escenarios delictivos reales.

- Para fortalecer aún más el sistema de reconocimiento de extorsiones, se recomienda ampliar las pruebas y validaciones del modelo `RandomForestClassifier` con conjuntos de datos adicionales y variados que representen diversas formas y contextos de lenguaje extorsivo. Esto permitirá mejorar la robustez del sistema y su capacidad para adaptarse a nuevas tácticas delictivas. Además, explorar la integración de técnicas avanzadas de procesamiento de señales de voz y aprendizaje automático podría proporcionar mejoras significativas en la precisión y eficiencia del sistema. Colaborar con expertos en psicología del crimen y lingüística forense para analizar y perfeccionar continuamente el modelo también sería fundamental para garantizar su efectividad y relevancia en la detección temprana y la prevención de extorsiones.

REFERENCIAS BIBLIOGRÁFICAS

[1] B. Velasco, «Llamadas telefónicas, principal vía para las extorsiones», El Comercio. Accedido: 19 de junio de 2024. [En línea]. Disponible en: https://www.elcomercio.com/actualidad/seguridad/llamadas-telefonicas-principal-via-para-las-extorsiones.html

[2] «Fraudes impulsados por la Inteligencia Artificial aumentan en Ecuador y el mundo», Primicias. Accedido: 19 de junio de 2024. [En línea]. Disponible en: https://www.primicias.ec/noticias/entretenimiento/tecnologia/inteligencia-artificial-fraude-estafa-ecuador/

[3] F. H. Ruíz Galeano, «Contribuciones desde un modelo predictivo para identificar el perfil de riesgo del defraudador interno en una entidad financiera de Colombia», may 2018, Accedido: 10 de junio de 2024. [En línea]. Disponible en: http://hdl.handle.net/10784/12343

[4] S. Sondhi, R. Vijay, M. Khan, y A. K. Salhan, «Voice analysis for detection of deception», en *2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, nov. 2016, pp. 1-6. doi: 10.1109/KICSS.2016.7951455.

[5] J. A. Chango Salas, «Desarrollo de un algoritmo detector de engaños mediante la utilización del método Deep Learning de inteligencia artificial», bachelorThesis, Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Electrónica y Telecomunicaciones, 2021. Accedido: 19 de junio de 2024. [En línea]. Disponible en: http://repositorio.espe.edu.ec/jspui/handle/21000/28482

[6] E. A. Zumba Narváez y F. P. Zumba Narváez, «Desarrollo de un sistema prototipo de generación de rostros a partir de señales de voz utilizando redes generativas adversarias», bachelorThesis, 2022. Accedido: 19 de junio de 2024. [En línea]. Disponible en: http://dspace.ups.edu.ec/handle/123456789/21758

[7] C. S. Calliconde Carrasco, «Detector de mentiras mediante el analisis de estrés de voz usando redes neuronales recurrentes», Thesis, 2022. Accedido: 19 de junio de

2024. [En línea]. Disponible en: http://repositorio.umsa.bo/xmlui/handle/123456789/29138

[8] «El papel de la inteligencia artificial en la detección de fraude y corrupción». Accedido: 25 de junio de 2024. [En línea]. Disponible en: https://www.worldcomplianceassociation.com/4493/noticia-el-papel-de-la-inteligencia-artificial-en-la-deteccion-de-fraude-y-corrupcion.html

[9] «Extorsión», Enciclopedia Significados. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://www.significados.com/extorsion/

[10] conceptosjuridicos.com, «Extorsión: qué es, regulación y elementos que deben formar este delito», Conceptos Jurídicos. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://www.conceptosjuridicos.com/extorsion/

[11] «¿Qué es y cómo actuar ante una extorsión?», Efecto Prevención. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://efectoprevencion.guanajuato.gob.mx/extorsion/que-es-y-como-actuar-ante-una-extorsion/

[12] L. B. Pazmiño, «Estos son los tipos de extorsión que existen en Ecuador», GK. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://gk.city/2023/04/13/tipos-extorsion-en-ecuador/

[13] «Los mensajes extorsivos aumentan. La Unase advierte a ciudadanía», El Universo. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://www.eluniverso.com/noticias/seguridad/los-mensajes-extorsivos-aumentan-la-unase-advierte-a-ciudadania-nota/

[14] «¿Qué hacer si usted recibe una llamada o mensajes con fines extorsivos?», El Universo. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://www.eluniverso.com/noticias/ecuador/que-hacer-si-usted-recibe-una-llamada-o-mensajes-con-fines-extorsivos-nota/

[15] «Análisis De Voz En Forense: Una Herramienta Clave En La Resolución De Casos Judiciales-2024». Accedido: 25 de junio de 2024. [En línea]. Disponible en: https://tudorisapp.com/analisis-de-voz-en-forense/

[16] «Extracción De Funciones Para Reconocimiento De Voz - FasterCapital». Accedido: 11 de junio de 2024. [En línea]. Disponible en: https://fastercapital.com/es/tema/extracci%C3%B3n-de-funciones-para-reconocimiento-de-voz.html

[17] micheal, «PPT - COEFICIENTES MFCC (Mel Frequency Cepstrum Coeficients) PowerPoint Presentation - ID:4267776», SlideServe. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://www.slideserve.com/micheal/coeficientes-mfcc-mel-frequency-cepstrum-coeficients

[18] J. Hui, «Speech Recognition — Feature Extraction MFCC & PLP», Medium. Accedido: 11 de junio de 2024. [En línea]. Disponible en: https://jonathan-hui.medium.com/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9

[19] «Extracción de características del reconocimiento de voz». Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://biblus.us.es/bibing/proyectos/abreproy/12054/fichero/MEMORIA%252F8.Cap%C3%ADtulo+3.pdf

[20] admin, «¿Qué es el reconocimiento de voz y cómo funciona?», RecFaces. Accedido: 29 de mayo de 2024. [En línea]. Disponible en: https://recfaces.com/articles/reconocimiento-voz

[21] «¿Qué es el reconocimiento del habla? | IBM». Accedido: 3 de junio de 2024. [En línea]. Disponible en: https://www.ibm.com/mx-es/topics/speech-recognition

[22] «Speech Recognition in Python. Speech recognition is a fascinating… | by Jitendra Pathak | Naukri Engineering | Medium». Accedido: 11 de junio de 2024. [En línea]. Disponible en: https://medium.com/naukri-engineering/speech-recognition-in-python-fcda027a97a1

[23] «Speech Recognition in Python [Learn Easily & Fast]», Simplilearn.com. Accedido: 29 de julio de 2024. [En línea]. Disponible en: https://www.simplilearn.com/tutorials/python-tutorial/speech-recognition-in-python

[24] decide4AI, «Tipos de aprendizaje que usan los algoritmos de Machine Learning», Decide. Accedido: 29 de mayo de 2024. [En línea]. Disponible en: https://decidesoluciones.es/tipos-de-aprendizaje-algoritmos-machine-learning/

[25] «Modelos supervisados machine learning», SEON ES. Accedido: 30 de mayo de 2024. [En línea]. Disponible en: https://seon.io/es/recursos/glosario/modelos-supervisados-machine-learning/

[26] «Qué es LSTM: Long short-term memory Concepto y definición. Glosario». Accedido: 27 de junio de 2024. [En línea]. Disponible en: https://gamco.es/glosario/lstm-long-short-term-memory/

[27] «¿Qué son las Redes LSTM?», Codificando Bits. Accedido: 27 de junio de 2024. [En línea]. Disponible en: https://www.codificandobits.com/blog/redes-lstm/

[28] R. R. Abril, «Redes Neuronales Recurrentes • Artículo de LMO», La Máquina Oráculo. Accedido: 29 de julio de 2024. [En línea]. Disponible en: https://lamaquinaoraculo.com/deep-learning/redes-neuronales-recurrentes/

[29] «Random Forest | Interactive Chaos». Accedido: 18 de junio de 2024. [En línea]. Disponible en: https://interactivechaos.com/es/wiki/random-forest

[30] «Random Forest Classification with Scikit-Learn». Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://www.datacamp.com/tutorial/random-forests-classifier-python

[31] «Base de Datos - Concepto, tipos y ejemplos», https://concepto.de/. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://concepto.de/base-de-datos/

[32] «What is an AI Database?», Redis. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://redis.io/nosql/ai-databases/

[33] «Comprender los datos estructurados, semiestructurados y no estructurados», Astera. Accedido: 26 de junio de 2024. [En línea]. Disponible en: https://www.astera.com/es/type/blog/structured-semi-structured-and-unstructured-data/

[34] «AMD Ryzen 5 3500U». Accedido: 29 de julio de 2024. [En línea]. Disponible en: https://technical.city/es/cpu/Ryzen-5-3500U

[35] «Procesador Intel® Core™ i7-12700 (25 MB de caché, hasta 4.90 GHz) - Especificaciones de productos», Intel. Accedido: 29 de julio de 2024. [En línea]. Disponible en: https://www.intel.la/content/www/xl/es/products/sku/134591/intel-core-i712700-processor-25m-cache-up-to-4-90-ghz/specifications.html

[36] «MATLAB - El lenguaje del cálculo técnico». Accedido: 29 de julio de 2024. [En línea]. Disponible en: https://la.mathworks.com/products/matlab.html

[37] «¿Qué es Python? - Explicación del lenguaje Python - AWS». Accedido: 29 de julio de 2024. [En línea]. Disponible en: https://aws.amazon.com/es/what-is/python/

ANEXOS

Anexo A. Validación de datos del Policía y del sistema

En la Figura A1 se muestra la estructura de la encuesta para la validación de datos del policía y en la Figura A2 se muestra la estructura de la validación de datos del Policía y del sistema.

ANÁLISIS DE FRASES

Teniendo en cuenta la definición de extorsión: Presión que se ejerce sobre alguien mediante amenazas para obligarlo a actuar de determinada manera y obtener así dinero u otro beneficio. La extorsión es un hecho que se desarrolla empleando métodos de violencia o intimidación a fin de asustar y obligar a la víctima a realizar cualquier acción en contra de su voluntad.

De las siguientes frases cuales usted considera que son extorsivas o no extorsivas

FRASES	SI	NO
1.-Hemos secuestrado a alguien cercano a ti. Si no pagas el rescate, le haremos daño.	x	
2.-Tus fotos personales están en nuestro poder. Si no pagas, las compartiremos en el internet.	x	
3.- Hemos hackeado tu cuenta y tenemos acceso a tus datos personales. Si no pagas, los revelaremos.	x	
4.- Tu hija ha sido involucrada en un incidente grave. Necesita tratamiento médico inmediato. Envía dinero para cubrir los costos médicos urgentes.	x	
5.- Tenemos a un familiar tuyo bajo nuestra custodia. Si no entregas el dinero del rescate pronto, no nos haremos responsables de lo que pueda sucederle.	x	
6.- Felicidades, su pedido está en camino. Confirme su dirección para asegurarse de que llegue sin problemas.		x
7.- Un ser querido ha sufrido un accidente y necesita una intervención médica costosa. Si no cooperas y nos ayudas con los fondos necesarios, lamentarás las consecuencias.	x	

Figura A1. Estructura de la validación de datos del Policía

PREGUNTAS	FRASES	POLICIA		SISTEMA DE IA		vp	fn	fp	vn
		SI	NO	SI	NO				
1	1.-Hemos secuestrado a alguien cercano a ti. Si no pagas el rescate, le haremos daño.	x		x		1			
2	2.-Tus fotos personales están en nuestro poder. Si no pagas, las compartiremos en el internet.	x		x		1			
3	3.- Hemos hackeado tu cuenta y tenemos acceso a tus datos personales. Si no pagas, los revelaremos.	x		x		1			
4	4.- Tu hija ha sido involucrada en un incidente grave. Necesita tratamiento médico inmediato. Envía dinero para cubrir los costos médicos urgentes.	x			x		1		
5	5.- Tenemos a un familiar tuyo bajo nuestra custodia. Si no entregas el dinero del rescate pronto, no nos haremos responsables de lo que pueda sucederle.	x		x		1			
6	6.- Felicidades, su pedido está en camino. Confirme su dirección para asegurarse de que llegue sin problemas.		x		x				1
7	7.- Un ser querido ha sufrido un accidente y necesita una intervención médica costosa. Si no cooperas y nos ayudas con los fondos necesarios, lamentarás las consecuencias.	x		x		1			

Figura A2. Tabla de validación de los datos del Policía y del sistema

Anexo B. Datos de la matriz de confusión

En la Figura A3 se muestra los datos obtenidos de la matriz de confusión después de la validación de los datos y los cálculos realizados.

Predicción	Valor positivo	Valor negativo
Predicción positiva	22(VP)	2(FP)
Predicción negativa	2(FN)	9(VN)

SENSIBILIDAD(Recall)	0,916666667
PRESISICIÓN	0,916666667
EXACTITUD	0,885714286
f1-score	0,916666667

Figura A3. Datos de la matriz de confusión y datos calculados

Anexo C. Base de datos csv del sistema

En la Figura A4 se muestra la estructura de la base de datos de Excel y en la Figura A5 se muestra la estructura de los datos dentro de la base del sistema.

197	Elige sabiamente, porque no habrá una segunda oportunidad	0
198	Podemos hacerte mucho daño si no cumples	0
199	No te hagas el valiente; sigue nuestras órdenes	0
200	Cada minuto que pasa agrava tu situación	0
201	No hay margen para errores; actúa ahora	0
202	Esto es un robo	0
203	Este es tu último aviso antes de que tomemos medidas serias	0
204	No te atrevas a ignorar esta advertencia, te estamos vigilando	0
205	Confirma la recepción de este mensaje si quieres evitar problemas	0
206	Este mensaje es serio y tus acciones tendrán consecuencias	0
207	Es imperativo que sigas nuestras instrucciones sin hacer preguntas	0
208	Nuestra paciencia se está agotando, actúa ahora o lo lamentarás	0
209	Esto no es un juego, la seguridad de tu familia está en juego	0
210	Asegúrate de hacer el pago antes de la medianoche si valoras tu bienestar	0
211	No intentes ser un héroe, sólo sigue lo que te decimos	0
212	Cualquier intento de contactar a las autoridades empeorará las cosas	0
213	¿Cómo estás hoy?	1
214	Gracias por tu ayuda	1
215	Me encanta este lugar	1
216	¡Qué bonito día hace!	1
217	¿Puedes repetir eso, por favor?	1

Figura A4. Estructura base de datos Excel

```
1076  "No te atrevas a ignorar esta advertencia, te estamos vigilando",0
1077  Confirma la recepción de este mensaje si quieres evitar problemas,0
1078  Este mensaje es serio y tus acciones tendrán consecuencias,0
1079  Es imperativo que sigas nuestras instrucciones sin hacer preguntas,0
1080  "Nuestra paciencia se está agotando, actúa ahora o lo lamentarás",0
1081  "Esto no es un juego, la seguridad de tu familia está en juego",0
1082  Asegúrate de hacer el pago antes de la medianoche si valoras tu bienestar,0
1083  "No intentes ser un héroe, sólo sigue lo que te decimos",0
1084  Cualquier intento de contactar a las autoridades empeorará las cosas,0
1085  Te voy a matar,0
1086  Te vamos a matar,0
1087  Te voy a hacer daño,0
1088  No nos provoques te voy a matar,0
1089  Esto es un asalto,0
1090  Sabemos que lugares frecuentas a donde vas y que haces deposita el dinero que te estamos pidiendo o pondremos una
1091  "Hoy se muere tu hijo sabemos en que escuela estudia, debes pagar 10 mil dolares para no hacerle daño",0
1092  ¿Cómo estás hoy?,1
1093  Gracias por tu ayuda,1
1094  Me encanta este lugar,1
1095  ¡Qué bonito día hace!,1
1096  "¿Puedes repetir eso, por favor?",1
1097  Voy a dar un paseo,1
1098  Estoy deseando verte,1
1099  ¿Qué opinas al respecto?,1
```

Figura A5. Estructura dataset del sistema

Anexo D. Archivos del sistema

En la siguiente Figura A6 se puede observar los archivos del sistema tanto la base datos en Excel, el dataset del sistema, encuesta de validación de datos del Policía y la tabla de validación de datos del policía y del sistema que se pueden encontrar en el siguiente link: [ARCHIVOS DEL SISTEMA - SISTEMA DE DETECCIÓN DE EXTORSIONES]

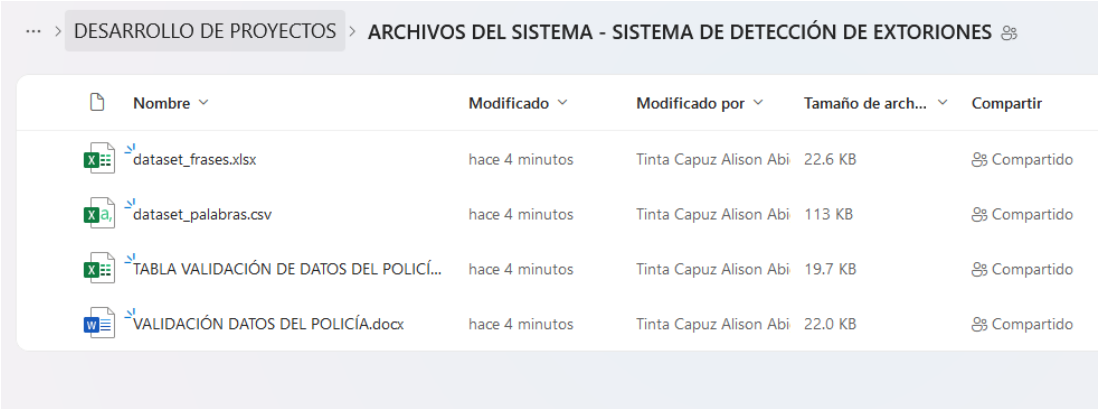


Figura A6. Archivos del sistema

Anexo E. Programación del sistema

En la siguiente Figura A7 se puede observar el procedimiento y desarrollo de la programación del sistema realizado en la interfaz del software Python, En la Figura A8 se puede observar la interfaz de ingreso de usuario al sistema y en la Figura A9 se puede observar la interfaz de detección de extorsiones.

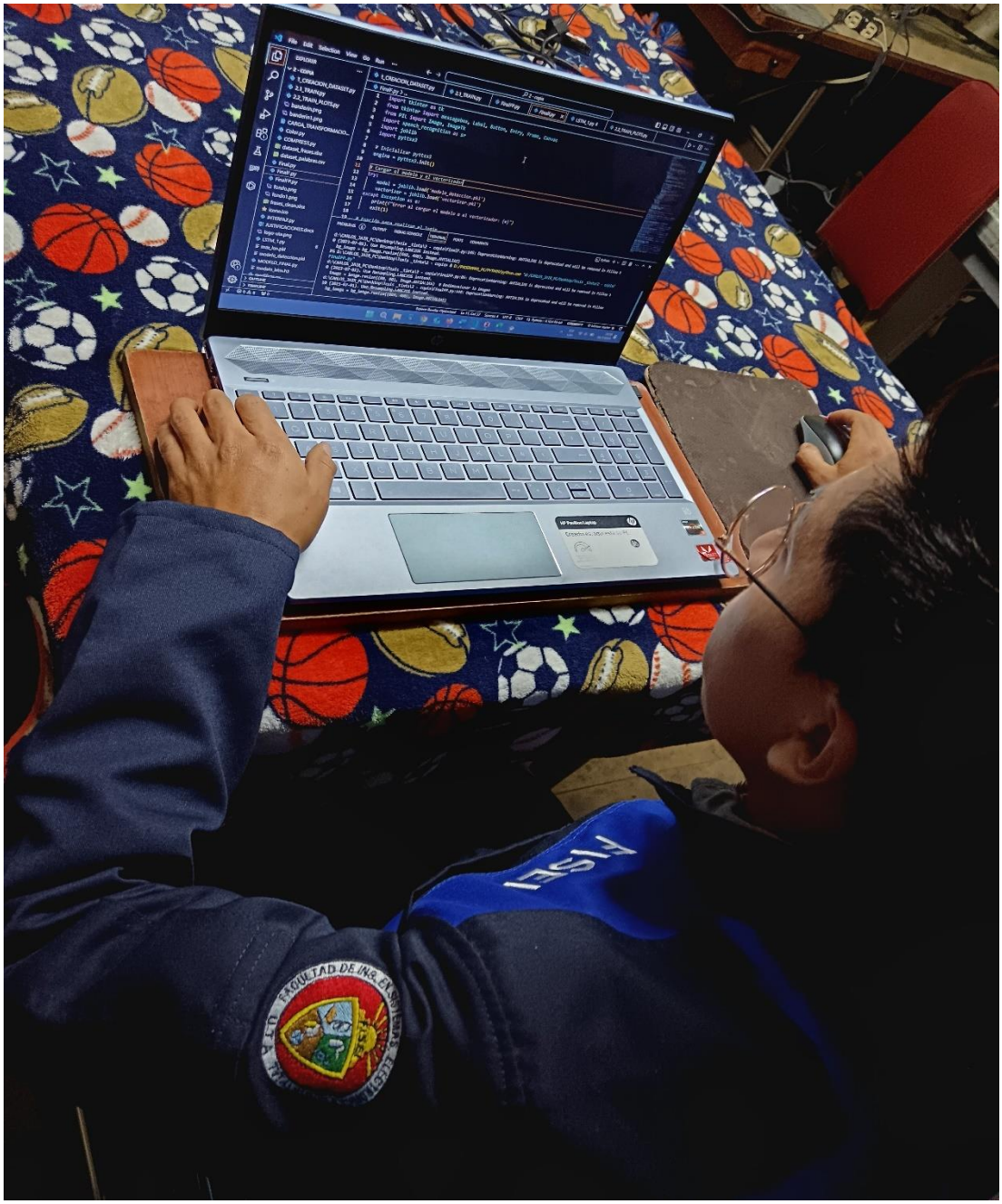


Figura A7. Programación del sistema



Figura A8. Ejecución del sistema ingreso de usuario



Figura A9. Ejecución del sistema de detección de extorsiones

Anexo F. Pruebas del sistema

En la siguiente Figura A10 se puede observar la prueba del funcionamiento del sistema con personas haciéndoles hablar frases extorsivas y no extorsivas a ver como se desenvuelve mi sistema en la detección de extorsiones. A continuación, se realizará con una frase extorsiva y en la Figura A11 se realiza las pruebas con frases no extorsivas.

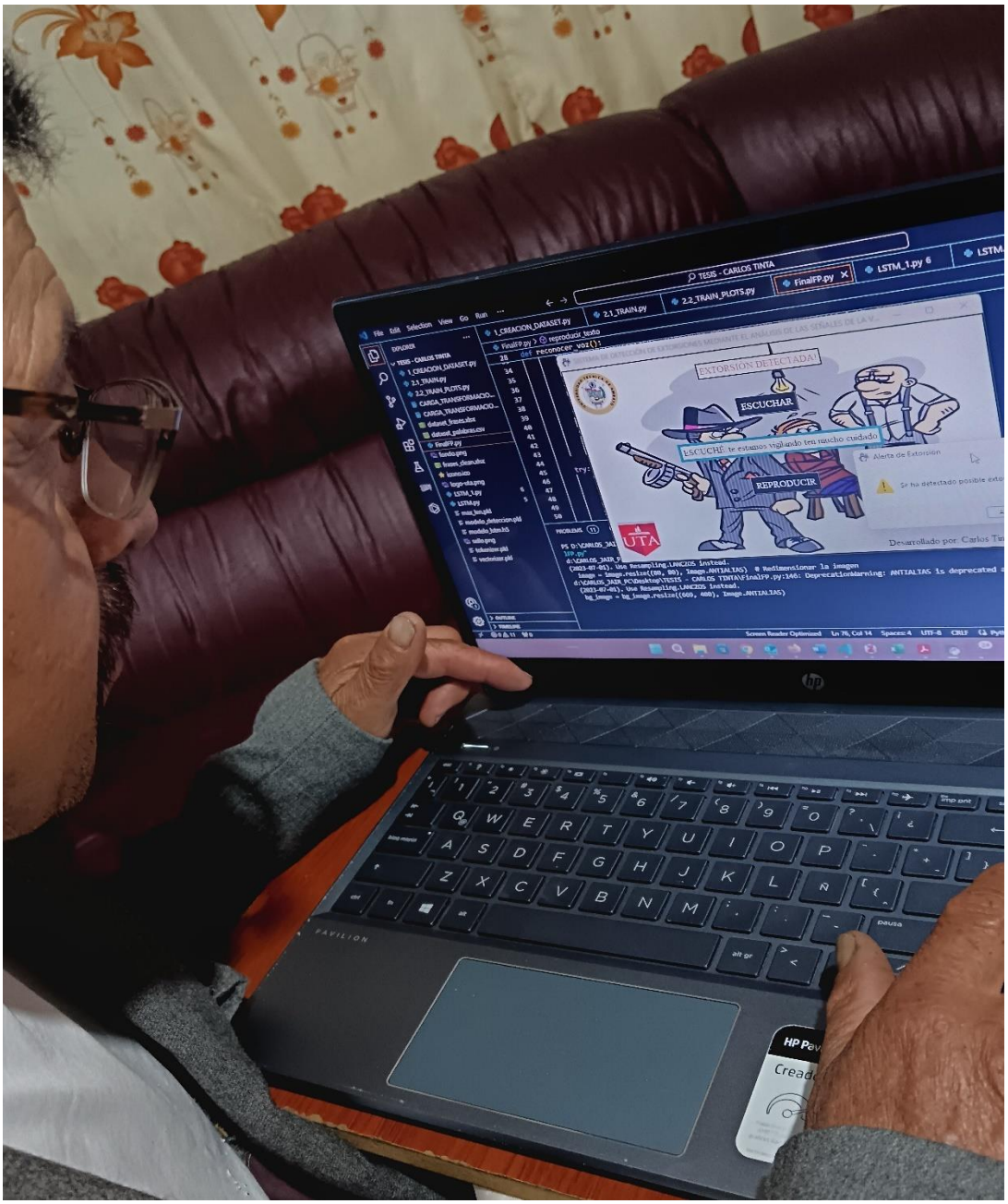


Figura A10. Pruebas de funcionamiento frase extorsiva

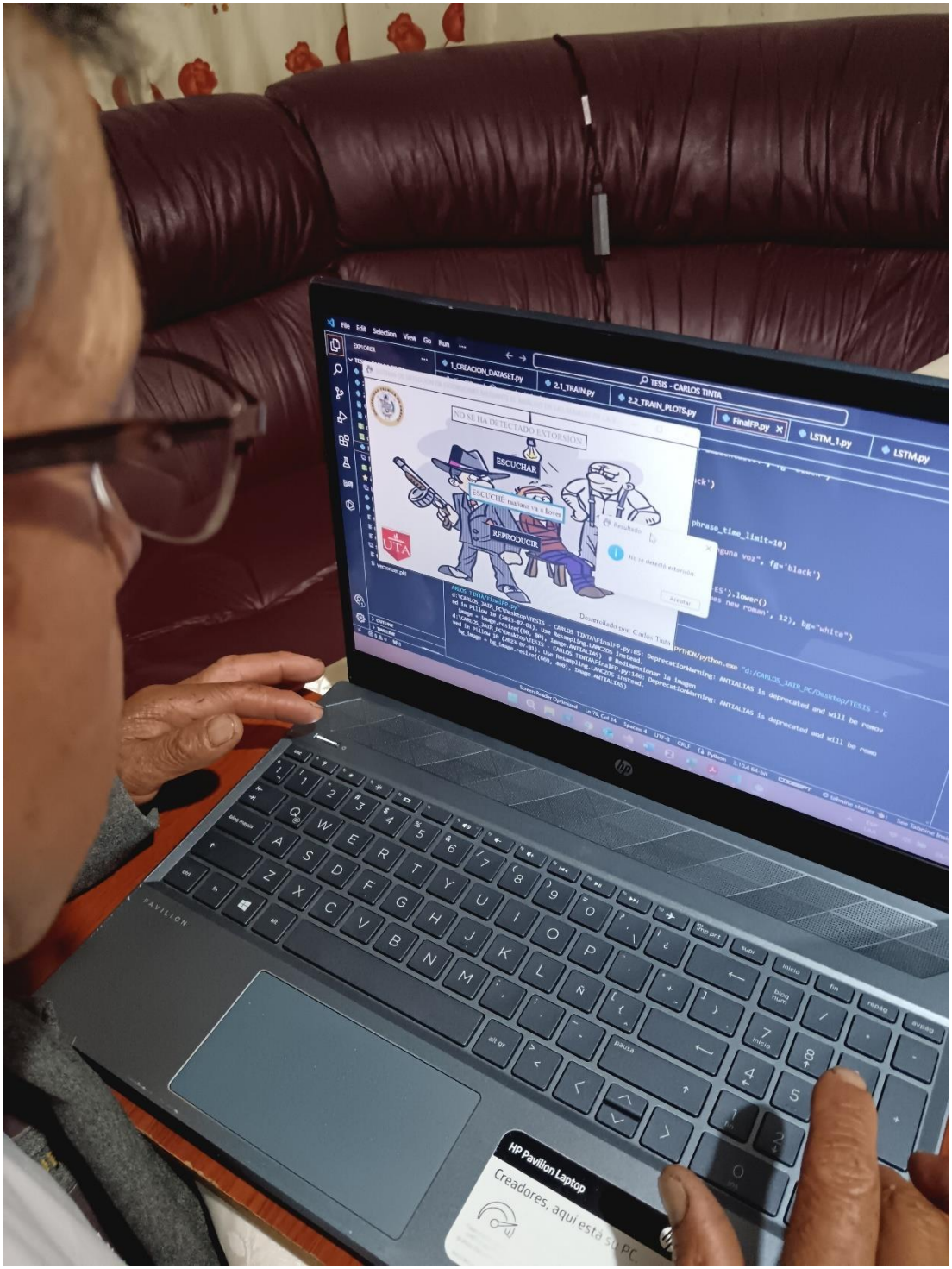


Figura A11. Pruebas de funcionamiento frase no extorsiva