Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Carrera de Computación

Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en español.

Trabajo de Integración Curricular, previa a la obtención del título de Ingeniero en Ciencias de la Computación

AUTOR:

Josue Alejandro Sauca Pucha

DIRECTOR:

Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc.

CODIRECTOR:

Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.

Loja – Ecuador

2025

Certificación de directores

Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.
CODIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

C E R T I F I C O:

Que hemos revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominado: **Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en español**, previo a la obtención del título de Ingeniera en Ciencias de la Computación, de la autoría del estudiante **Josue Alejandro Sauca Pucha**, con cédula de identidad **1105580581**, una vez que se determina que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.
CODIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo **Josue Alejandro Sauca Pucha**, declaro ser autor del presente Trabajo de Integración Curricular eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular o de Titulación, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma: 

Cédula de identidad: 1105580581

Fecha: 12/03/2025

Correo electrónico: josue.sauca@unl.edu.ec

Teléfono: 0939253857

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular

Yo, **Josue Alejandro Sauca Pucha**, declaró ser el autor del Trabajo de Integración Curricular denominado: **Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en español**, como requisito para optar por el título de **Ingeniero en Ciencias de la Computación**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 12 días del mes de marzo de dos mil veinticinco.

Firma: 

Autor: Josue Alejandro Sauca Pucha
Cédula de identidad: 110550581
Dirección: Las peñas la inmaculada 2
Correo electrónico: josue.sauca@unl.edu.ec
Teléfono: 07272442

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc.

Codirector del Trabajo de Integración Curricular: Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.

Dedicatoria

Dedico este trabajo a Dios, por ser mi guía y fortaleza en cada paso de este camino. A mis padres, por su amor incondicional, su apoyo constante y por enseñarme que con esfuerzo todo es posible, mi hermana, por ser una gran compañía, a mis amigos y a todas las personas que me brindaron su ayuda y conocimiento, haciendo que este proceso fuera más llevadero, y con especial cariño, a quienes conocí en el camino y, aunque ya no estén, dejaron una huella muy bonita en mi vida, los llevó en mi corazón.

Josue Alejandro Sauca Pucha

Agradecimientos

Quiero expresar mi más profundo agradecimiento, en primer lugar, a Dios, por darme la fuerza, la sabiduría y las oportunidades necesarias para completar este proyecto. Su guía ha sido fundamental en cada paso de mi vida.

Agradezco de todo corazón a mis padres Nancy Pucha y Fredi Sauca, quienes han sido mi mayor inspiración, brindándome amor incondicional y apoyo constante, y a mi hermana Claudia, que ha estado siempre a mi lado, ofreciéndome ánimo y compañía.

Extiendo mi más sincero agradecimiento al Ing. Oscar Cumbicus, por haberme proporcionado la idea inicial para este proyecto y por su orientación en las etapas tempranas, y al Ing. Ing. Roberth Figueroa, por su valiosa supervisión, guía y conocimiento durante el desarrollo del trabajo.

Mi reconocimiento también va a las personas que contribuyeron directamente a la recolección de datos, dedicando su tiempo y esfuerzo a las grabaciones: Danny Martínez, Juan Castillo, Juan Carreño, Luis Delgado, Paula Vaca, Paulina Chalco, Jimmy Cajamarca, Melissa Tuza, Edy Jiménez, Jennifer Quizhpe, Antony Luzuriaga, Gerardo Quizhpe, Vanessa Loja, David Rodríguez y Claudia Sauca, su colaboración fue importante para hacer realidad este proyecto.

Agradezco profundamente a quienes me brindaron su apoyo logístico, ofreciéndome un hogar y posada cuando las condiciones se tornaron difíciles, especialmente durante los momentos en los que no contaba con luz ni recursos adecuados para avanzar en mi trabajo, su generosidad y solidaridad jamás serán olvidadas.

Finalmente, quiero dedicar un espacio a las personas que conocí a lo largo de este camino, quienes, de una u otra manera, dejaron una huella en mi vida, aunque ya no estén presentes, su importancia sigue viva en mis recuerdos. Más que un simple agradecimiento, este proyecto también es un tributo a las memorias compartidas, al aprendizaje que cada encuentro dejó en mí. En cada conversación, en cada instante de confianza y complicidad, quedó grabada una parte invaluable de esta etapa. Las despedidas a veces llegan sin avisar, pero no por ello desvanecen lo que una vez fue genuino. Independientemente de cómo termino, en mi corazón siempre habrá un lugar para aquellos que fueron parte de este viaje.

Josue Alejandro Sauca Pucha

Índice de Contenidos

Índice de Tablas

Índice de Figuras

Índice de Anexos

1. Título

Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en español.

Developing a dataset of voice spoofing with a sample of 15 people from the Computer Science Career at the National University of Loja by using the StarGAN network and its evaluation with a model LFCC-LCNN pre-trained in Spanish.

2. Resumen

El presente trabajo abordó la la carencia de datasets representativos para la detección de voice spoofing, un desafío creciente en el campo de la inteligencia artificial, centrado en el análisis del modelo LFCC-LCNN utilizando un dataset generado mediante la red StarGAN, basado en muestras de 15 personas del cantón Loja. El objetivo principal fue evaluar la Tasa de Error de Igualación (EER) del modelo con un enfoque específico en las características fonéticas del español ecuatoriano, por lo que se lo desarrolló bajo la metodología CRISP-ML(Q), que estructuró el proceso en fases, desde la recolección y generación del dataset, pasando por su preprocesamiento y transformación, hasta la evaluación del modelo. Las grabaciones se realizaron en entornos no controlados, utilizando equipos básicos, lo que generó un dataset no tan robusto. Los resultados mostraron que la Tasa de Error de Igualación varió significativamente dependiendo de la composición del dataset, en el peor escenario, con el dataset generado se obtuvo un EER del 63.85%, mientras que en el mejor caso, al combinar datos generados con muestras originales, se alcanzó un EER del 26.71%, lo que indicar que el desempeño del modelo no se vio limitado por su arquitectura, sino por la discrepancia entre los acentos presentes en el dataset de entrenamiento, que incluía datos venezolanos, argentinos y chilenos, y el dataset evaluado con acento lojano, además la calidad del dataset y el desbalance en la muestra también influyeron en los resultados. El estudio evidenció la importancia de desarrollar datasets representativos de la población objetivo para cubrir la carencia de datasets en este ambito, por lo que se recomienda continuar esta línea de investigación ampliando el dataset y explorando arquitecturas más robustas que integren técnicas de aprendizaje transferido.

Palabras Claves*: Voice spoofing, Redes generativas, Acentos ecuatorianos, LFCC-LCNN, Tasa de Error de Igualación, CRISP-ML(Q)*

Abstract

This study addressed the lack of representative datasets for voice spoofing detection, a growing challenge in the field of artificial intelligence. It focused on analyzing the LFCC-LCNN model using a dataset generated through the StarGAN network based on samples from 15 individuals from the Loja canton. The main objective was to evaluate the Equal Error Rate (EER) of the model with a specific focus on the phonetic characteristics of Ecuadorian Spanish. The study was developed using the CRISP-ML(Q) methodology, structuring the process into phases—from dataset collection and generation to preprocessing, transformation, and model evaluation. The recordings were conducted in uncontrolled environments using basic equipment, resulting in a dataset that was not highly robust. The results showed that the Equal Error Rate varied significantly depending on the dataset composition. In the worst-case scenario, the generated dataset yielded an EER of 63.85%, whereas in the best case, by combining generated data with original samples, the EER improved to 26.71%. This indicates that the model's performance was not limited by its architecture but rather by discrepancies in accents between the training dataset—which included Venezuelan, Argentine, and Chilean data—and the evaluated dataset, which featured the Loja accent. Additionally, dataset quality and sample imbalance influenced the results. The study highlighted the importance of developing representative datasets for the target population to address the shortage of suitable datasets in this field. Therefore, it is recommended to continue this line of research by expanding the dataset and exploring more robust architectures that integrate transfer learning techniques.

Keywords: *Voice spoofing, Generative networks, Ecuadorian accents, LFCC-LCNN, Equal Error Rate, CRISP-ML(Q)*

3. Introducción

La detección de voice spoofing, o suplantación de voz, ha emergido como un área crítica dentro del campo de la inteligencia artificial (IA) y el aprendizaje profundo, dada su relevancia en sistemas de autenticación por voz y seguridad biométrica, este fenómeno consiste en el uso de tecnologías avanzadas para imitar o generar voces humanas con el propósito de engañar sistemas de reconocimiento de voz. Según estudios recientes, los avances en redes neuronales generativas, como las redes StarGAN, han incrementado significativamente la calidad y realismo de estas suplantaciones [1]. Sin embargo, la capacidad de los modelos actuales para detectar suplantaciones se ve limitada cuando se enfrentan a variaciones regionales del habla y contextos específicos no representados en los datasets de entrenamiento.

En este contexto, el presente trabajo se centra en explorar la efectividad del modelo LFCC-LCNN en la detección de voice spoofing utilizando un dataset generado mediante StarGAN con acento lojano, una variante del español ecuatoriano, la relevancia de este estudio radica en la falta de datasets representativos que incluyan variaciones fonéticas y características acústicas propias del habla ecuatoriana, las cuales podrían influir en el desempeño de los modelos existentes, este vacío plantea la necesidad de investigar cómo los modelos actuales pueden ser optimizados o adaptados a contextos locales específicos, contribuyendo al desarrollo de soluciones más robustas y precisas en la detección de suplantación de voz.

El desarrollo de este trabajo tiene múltiples beneficios, particularmente para la región de Loja y, en general, para los sistemas de seguridad basados en voz en español, ya que al generar un dataset representativo y evaluar su impacto en la precisión de los modelos, se busca no solo mejorar el desempeño de los sistemas actuales, sino también aportar un recurso valioso para futuras investigaciones en el ámbito local y regional. Estudios previos han demostrado que los modelos de detección de voice spoofing presentan dificultades al enfrentarse a variaciones en los datos de entrenamiento y prueba, particularmente cuando los datos de evaluación difieren significativamente en términos de acento o calidad de grabación [2].

Inspirados por estas limitaciones, el presente trabajo adopta la metodología CRISP-ML(Q), que permite estructurar de manera eficiente el proceso de desarrollo, la metodología seleccionada se hizo en base que tiene un enfoque iterativo que permite abordar las fases del proyecto que van desde la recolección de datos hasta la evaluación del modelo, y como se las fases tienen flexibilidad para adaptarse a los ajustes necesarios a lo largo del proceso.

El principal objetivo de este estudio fue evaluar la Tasa de Error de Igualación (EER) del modelo LFCC-LCNN utilizando un dataset generado con StarGAN basado en grabaciones de 15 personas de la región de Loja, este objetivo se complementó con la generación y preprocesamiento de un dataset local que capturara las características distintivas del habla lojano, destacando la importancia de representar adecuadamente los acentos y las condiciones acústicas en los sistemas de seguridad biométrica.

El alcance de este trabajo se limita al análisis de un modelo y una arquitectura específicos (LFCC-LCNN), así como a un dataset de tamaño reducido debido a restricciones en recursos computacionales y logísticos, sin embargo, estos límites abren la posibilidad de futuras investigaciones orientadas a ampliar el dataset, explorar otras arquitecturas y evaluar la generalización del modelo en diferentes contextos de habla.

Finalmente, este proyecto busca no solo contribuir al entendimiento de las limitaciones actuales en la detección de voice spoofing, sino también sentar las bases para el desarrollo de sistemas más inclusivos y efectivos que respondan a las necesidades específicas de las comunidades hispanohablantes.

4. Marco teórico

4.1. Antecedentes

La suplantación de identidad basada en la voz, conocida como “voice spoofing”, consiste en imitar o clonar la voz de una persona con el propósito de engañar a sistemas de verificación por voz [3]. Aunque la verificación de identidad mediante la voz se ha convertido en una herramienta valiosa para la seguridad, su efectividad depende en gran medida de disponer de conjuntos de datasets de voz [4]. En el contexto ecuatoriano, la falta de datos con acentos locales limita la adaptación de los modelos a las variaciones lingüísticas del país, debido a la falta de datos. Esta carencia adquiere relevancia en base a datos de la **Fiscalía General del Estado Ecuatoriano** presentados en el **Anexo 3. Datos delitos cibernéticos consulta a la fiscalía**, los delitos informáticos han mostrado un incremento significativo, reportándose aproximadamente 17 480,00 denuncias en 2017 y aumentando a 37 614 en 2023, según un reporte oficial obtenido mediante solicitud directa a la Fiscalía [5], [6]

La creciente amenaza de la suplantación de identidad por voz plantea riesgos significativos para la seguridad, para enfrentar esta problemática, es importante contar con conjuntos de datos especializados. Mediante una revisión literaria, se pudo evidenciar una amplia cantidad de datasets de spoofing en inglés [5] [6], japonés [7] e indio[7], [8], [9], [10]. En Latinoamérica existe un dataset que incluye acentos de países como Colombia, Chile, Perú, Venezuela y Argentina [11]. Sin embargo, no se han encontrado estudios que aborden datasets con acentos ecuatorianos y menos en la provincia de Loja.

En base a una entrevista con el Ing. Oscar Cumbicus, docente de la Carrera de Computación en la Universidad Nacional de Loja **Anexo 1. Entrevista realizada al experto en inteligencia artificial**, existen numerosas herramientas de Text-to-Speech (TTS) que permiten clonar voces y generar audios a partir de texto. En la presente entrevista se se habló sobre la red StarGAN para la conversión de voz [1], ya que esta técnica que facilita la creación de un dataset variado y realista con un menor número de ejemplos de entrenamiento.

Luego de una revisión de trabajos relacionados respecto al presente tema, se pudo evidencia que el modelo LFCC-LCNN, tiene una capacidad para extraer características acústicas relevantes y su rendimiento en tareas de detección de spoofing. Este modelo se usará para la evaluación del conjunto de datos de generado, se espera que la comparación de los resultados del modelo LFCC-LCNN con las métricas de Tasa de Error de Igualación (EER) que de información importante sobre el dataset desarrollado.

4.2. Fundamentación Teórica

4.2.1. Inteligencia artificial

La Inteligencia Artificial (IA) es un campo de la informática que desarrolla sistemas que pueden emular la inteligencia de las personas en muchas tareas, tales como el reconocimiento de voz hasta la resolución de problemas complejos.[12] Su capacidad distintiva de aprendizaje autónomo, principalmente a través del machine learning, permite que estos sistemas mejoren continuamente con la exposición a más datos, aumentando su eficiencia y precisión con el tiempo.[13]

La aplicación de la IA abarca múltiples sectores como en el ámbito legal, facilita la navegación y análisis de documentos complejos interrelacionados en educación, transforma el aprendizaje y la resolución creativa de problemas; en salud, optimiza diagnósticos y tratamientos; y en la industria creativa y el transporte, revoluciona procesos mediante la automatización inteligente[14]. Esta versatilidad, combinada con su capacidad de procesar grandes volúmenes de datos y adaptarse a nuevas situaciones, hace de la IA una herramienta transformadora que está redefiniendo cómo se abordó en prácticamente todos los campos profesionales.

4.2.1.1. Inteligencia artificial generativa

La Inteligencia Artificial Generativa (IA Generativa) generativa es un tipo de IA que puede crear contenido nuevo y original a partir de datos existentes[15] Estos sistemas pueden generar una variedad de resultados, como texto, imágenes o audio, basados en indicaciones o entradas específicas proporcionadas por los usuarios, por ejemplo, modelos de lenguaje como ChatGPT, un caso popular de IA Generativa, utilizan grandes cantidades de datos extraídos de la web para generar respuestas similares a las humanas mediante el procesamiento de entradas a través de algoritmos avanzados, como los transformadores, que determinan la relevancia de los datos en relación con una consulta.

La IA Generativa permite automatizar tareas complejas, como la codificación de datos en contextos educativos, donde tradicionalmente se requerían técnicas manuales que eran costosas y propensas a errores. Los modelos como GPT han simplificado estos flujos de trabajo mediante el uso de chatbots basados en modelos de lenguaje, lo que proporciona una forma accesible de aplicar tecnologías de vanguardia [16][17]. Aunque se han explorado las capacidades de la IA Generativa en diversas tareas lingüísticas, su eficacia en la codificación automática de discursos en contextos educativos aún se está investigando, lo que abre oportunidades para mejorar la eficiencia y precisión en el análisis de datos educativos.

4.2.2. Suplantación de Identidad por Voz

La suplantación de voz, conocida como ataques de falsificación de identidad vocal, se refiere al uso indebido de técnicas avanzadas para replicar o manipular la voz de un individuo con el propósito de engañar sistemas de verificación automática del hablante, estos sistemas, comúnmente son utilizados en dispositivos como asistentes virtuales, aplicaciones bancarias y controles de acceso a hogares inteligentes, confían en la singularidad de las características vocales de los usuarios para autenticar su identidad, sin embargo se puede vulnerarlos mediante el uso de métodos de falsificación que consiste en reproducir grabaciones de la voz del usuario para engañar al sistema de autenticación, también los ataques de repetición donde se utiliza inteligencia artificial para generar imitaciones precisas del hablante objetivo, y la conversión de voz donde se usa una voz existente para cambiarle características para que coincida con la voz objetivo [18], [19].

Este tipo de ataques va más allá de los riesgos tecnológicos, ya que pueden facilitar actividades de índole fraudulentas, una de ellas es el acceso no autorizado a cuentas bancarias o a realizar la manipulación de dispositivos domóticos inteligentes [20]. Las contramedidas contra la suplantación de voz se encuentran en evolución ya que mediante el uso de métodos como modelos de mezcla gaussiana, el uso de redes neuronales convolucionales y redes neuronales recurrentes (RNN), se buscan la identificación de patrones en las señales de audio que diferencian las voces genuinas de las falsificadas, como los artefactos de procesamiento en las voces sintéticas o las distorsiones acústicas en los ataques de repetición.

4.2.3. Dataset

Un dataset, es conocido como un conjunto de datos, es una colección organizada de información estructurada o no estructurada que se utiliza para entrenar, evaluar y validar modelos en diversas áreas, como aprendizaje automático, visión por computadora, biometría, procesamiento de lenguaje natural, ciberseguridad y otros campos científicos y tecnológicos, estos datos pueden representarse en diferentes formatos, como texto, imágenes, audio, video o datos tabulares, y su calidad, composición y adecuación determinan en gran medida el rendimiento de los sistemas y modelos que los utilizan [21], [22], [23]. Para que un modelo aprenda y generalice de manera efectiva es importante que el dataset sea representativo, equilibrado y libre de sesgos, los datasets suelen dividirse en subconjuntos de entrenamiento, validación y prueba para que el modelo se entrene de manera adecuada.

Existen diferentes tipos de datasets que se adaptan a diversas disciplinas, por ejemplo existen datasets que pueden incluir imágenes del iris o de rostros capturadas bajo condiciones controladas o en entornos dinámicos, dependiendo del objetivo de investigación, también en

el área de ciberseguridad, los datasets recopilan información de redes, patrones de tráfico o registros de intrusión para entrenar sistemas de detección de anomalías o ataques, también en el área de visión por computadora, los datasets pueden incluir imágenes etiquetadas para tareas como reconocimiento de objetos o segmentación semántica [24], [25].

4.2.4. Redes LCNN

Las redes Light Convolutional Neural Networks (LCNN) son un tipo de red neuronal profunda diseñada para tareas como clasificación o detección datos complejos, como audio, imágenes, video, estas redes utilizan filtros especializados llamados filtros locales que analizan características específicas de los datos tales como patrones para identificar el tipo de dato en lugar de aplicar los mismos filtros en todas las regiones, como lo hacen las CNN estándar, lo que permite que las LCNN manejen datos que contienen múltiples características como colores, formas, movimientos que suelen presentar grandes diferencias entre sí [26].

Un aspecto de las LCNN es el uso de Local Convolutional Layers (LCL), las cuales implementan una estrategia distinta a las redes convolucionales estándar (CNN) al aplicar filtros específicos para cada región de los datos, mientras que en las CNN estándar se utilizan filtros uniformes, que analizan todas las partes de los datos de la misma manera, las LCL emplean filtros adaptados a las características locales de cada área, lo que significa que diferentes regiones de los datos pueden ser analizadas con filtros especializados, optimizados para detectar patrones únicos en dichas áreas[20], [27].

4.2.5. Redes Generativa StarGAN

Las redes StarGAN son una de las variaciones de las redes generativas adversariales (GAN) las mismas que pueden realizar conversiones de muchos a muchos en dominios de atributos no paralelos, lo que significa que no es necesario contar con grabaciones exactas o emparejadas del mismo contenido en diferentes voces o estilos. Las redes StarGAN son herramienta eficiente para abordar el aprendizaje no supervisado en escenarios complejos y con datos diversos [1], [28].

En la **Figura 1** se visualiza la arquitectura general de StarGAN-VC que tiene el generador (G) y el discriminador (D), los cuales se detallan:

- **Generador(G):** El generador mapea características acústicas de un dominio a otro, utilizando una entrada compuesta por las características de origen y una etiqueta auxiliar que indica el dominio de destino, el objetivo es producir muestras que parezcan auténticas y pertenezcan al dominio de destino especificado. La fórmula del generador $\hat{y} = G(x, c)$ donde la variable x es la entrada acústica y la variable c es el atributo de destino.

- **Discriminador(D):** El discriminador se encarga de clasificar las muestras generadas como reales o falsas y predice los atributos a los que pertenecen, el objetivo de la misma es minimizar la pérdida adversaria definida como:

$$L_D(D) = -E_{y\sim p(y)}[\log D(y)] - E_{x\sim p(x)}\left[\log\left(1 - D\left(G(x,c)\right)\right)\right], \text{ donde:}$$

  - $-E_{y\sim p(y)}[\log D(y)]$ mide la capacidad del discriminador para clasificar correctamente la muestra real de la variable y.
  - $-E_{x\sim p(x)}\left[\log\left(1 - D\left(G(x,c)\right)\right)\right]$ mide la capacidad del discriminador para clasificar correctamente la muestra generada por $G(x,c)$ como falsa, este valor representa la valida del generador que intenta engañar al discriminador [29].

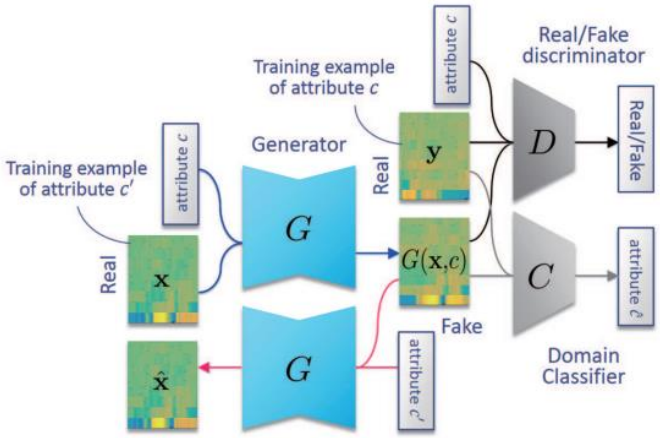


Figura 1. Funcionamiento red StarGAN.

4.2.6. CRISP-ML

Los avances en el área de machine learning abarcan diversas industrias en constante evolución, como el cuidado de la salud, el transporte, la manufactura, entre otras. Estas áreas buscan aprovechar la tecnología para transformar, renovar y optimizar procesos a través del análisis de datos y la automatización de tareas. Sin embargo, muchos proyectos de machine learning no logran cumplir completamente con las expectativas debido a la falta de estandarización, lo que impacta negativamente en la calidad y en los resultados esperados [30].

En vista a la problemática respecto a la calidad de los proyectos de machine learning, la comunidad ha tratado de establecer un marco referencial como estándar para los proyectos que abordan temas de machine learning debido a la falta de organización y la dificultad de ser reproducibles [30][31]. Por ende, se ha establecido una guía para los proyectos de ML llamada

CRISP-ML(Q), que tiene como objetivo principal garantizar la calidad y reproducibilidad de los modelos a lo largo de su ciclo de vida.

En la **Figura 2** se ven las etapas de CRISP-ML(Q) que cubre metodologías específicas para asegurar la calidad desde la fase de recolección de datos hasta el despliegue del modelo, la metodología consta de 6 pasos, los mismos que componen el ciclo de vida en CRISP-ML(Q) son [32]:

- **Comprensión del negocio y de los datos:** La comprensión del negocio y los datos, donde se definen los objetivos comerciales y se traducen en metas específicas para machine learning. Esta fase es la encargada de evaluar la calidad y disponibilidad de los datos, y determina la viabilidad del proyecto considerando los recursos disponibles.
- **Ingeniería de datos:** Aquí se prepararán los datos para el modelado, se incluye la selección de características relevantes, la limpieza de datos para eliminar errores y duplicados, y la ingeniería de características, donde se pueden crear nuevas variables a partir de las existentes
- **Ingeniería de modelos de aprendizaje automático:** Aquí se realiza el entrenamiento del modelo de machine learning, basado en los datos preparados. Se seleccionan algoritmos adecuados para el problema y se ajustan hiperparámetros para mejorar el rendimiento. El objetivo es crear un modelo que generalice bien y sea capaz de hacer predicciones precisas en datos nuevos.
- **Control de calidad para aplicaciones de aprendizaje automático:** Durante esta etapa se verifica que el modelo cumple con los requisitos establecidos, como el rendimiento en diferentes conjuntos de datos. Se evalúa la reproducibilidad del modelo, de manera que los resultados obtenidos puedan replicarse con los mismos datos y configuraciones.
- **Despliegue:** El despliegue del modelo permite integrar el modelo con aplicaciones empresariales o de usuario final una vez que ha superado las fases de prueba y de validación. Es importante asegurar que el modelo puede escalar y manejar diferentes cargas de trabajo.
- **Monitoreo y Mantenimiento:** Uno de los mayores riesgos que puede ocurrir es la obsolescencia del modelo, ya que su rendimiento disminuye al enfrentarse nuevos datos. Para evitar este tipo de errores es necesario realizar un monitoreo sobre el rendimiento del modelo y evaluar si requiere reentrenamiento, esta práctica es conocida como la evaluación continua del modelo.

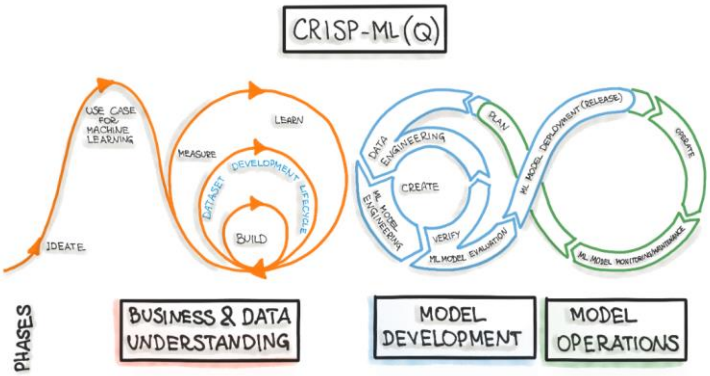


Figura 2. Metodología CRIPS-ML(Q).

4.2.7. Tasa de Error de Igualación

La Tasa de Error de Igualación (EER) es una métrica utilizada para evaluar el desempeño de sistemas de clasificación binaria, particularmente en el contexto de verificación biométrica y detección de suplantación de voz, lo que representa el punto en el que las tasas de falsos rechazos (False Rejection Rate, FRR) y falsas aceptaciones (False Acceptance Rate, FAR) son iguales o lo más cercanas posibles, este valor es importante ya que proporciona una medida unificada del equilibrio entre los errores de ambos tipos, eliminando la necesidad de priorizar uno sobre el otro [27], [11].

Para realizar el cálculo del EER se debe ajustar el umbral de decisión del sistema hasta encontrar el valor de t', donde se minimiza la diferencia entre FRR y FAR, lo que la fórmula matemática $EER = \frac{FRR(t')+FAR(t')}{2}$, se define como:

- **FRR (False Rejection Rate):** Mide la proporción de instancias genuinas que un sistema rechaza incorrectamente como si fueran fraudulentas, como en un sistema biométrico como de reconocimiento de voz, ocurre un rechazo falso cuando un usuario legítimo intenta autenticarse y el sistema no lo reconoce debido a variaciones en su voz, ruido de fondo o limitaciones en el modelo, valores altos en FRR alto puede generar frustración en los usuarios genuinos, ya que el sistema les niega acceso injustamente, afectando la experiencia de uso y, en algunos casos, la eficiencia operativa del sistema.
- **FAR (False Acceptance Rate):** Mide la proporción de instancias fraudulentas que un sistema acepta incorrectamente como si fueran legítimas, dentro de un contexto de seguridad biométrica, como la detección de suplantación de voz, ocurre una aceptación falsa cuando el sistema no detecta una voz manipulada o falsificada y la auténtica como si fuera del usuario real, valores altos en FAR representan un riesgo significativo para la seguridad, ya que permite accesos no autorizados, lo que puede comprometer datos sensibles o sistemas protegidos.

El umbral t' tiene su fórmula $t' = \arg\min_t |FRR(t) - FAR(t)|$, que se lo define como:

- Este umbral representa el punto de equilibrio entre las dos tasas de error como **FRR** y **FAR,** dentro de un sistema de verificación o detección, es importante ya que permite evaluar el rendimiento del sistema en condiciones balanceadas, donde los errores al rechazar instancias genuinas y al aceptar instancias fraudulentas son lo más similares posibles.

4.3. Tecnologías y Herramientas

4.3.1. Google Drive

Google Drive es una plataforma de almacenamiento en la nube que ofrece 15 GB de espacio gratuito, brinda almacenamiento, creación y modificación de documentos en tiempo real de manera colaborativa. Facilita el trabajo sincrónico y asincrónico, es accesible desde múltiples dispositivos como computadoras, celulares entre otros. Además, incluye diversas herramientas de ofimática como procesador de texto, hojas de cálculo y presentaciones, con la capacidad de sincronización automática y soporte para múltiples formatos, como PDF, lo que lo convierte en una herramienta versátil para entornos educativos y profesionales [33], [34], [35].

4.3.2. Python

Python es un lenguaje de programación multiparadigma de alto nivel, creado por Guido van Rossum a principios de la década de 1990. Es un lenguaje de programación interpretado, multiplataforma y conocido por tener una escritura sencilla, además tiene usos en áreas como desarrollo web, ciencia de datos, inteligencia artificial y automatización [36], [37]. Python admite módulos y paquetes, lo que permite modularidad del programa y la reutilización del código.

Python es muy utilizado en el campo de la inteligencia artificial debido a su variedad de bibliotecas disponibles, las aplicaciones que tiene van desde la creación de modelos avanzados de machine learning, mediante el uso de librerías como TensorFlow, PyTorch, Keras, Numpy, Seaborn, entre otras, hasta tareas como la clasificación de imágenes y procesamiento de lenguaje natural mediante uso de modelos como Yolo, chatgpt, etc [38].

4.3.3. Google Colab

Google Colab es un sitio web gratuito que sirve como entorno de desarrollo para lenguajes de programación como Python y R. Al estar alojada en la nube, permite ejecutar código en notebooks de Python desde cualquier dispositivo con acceso a Internet, ofreciendo

además recursos potentes como GPUs y TPUs para proyectos que requieren alta capacidad de procesamiento [39].

Colab se ejecuta en un backend gestionado por Google, lo que significa que los usuarios no tienen control directo sobre el hardware, pero pueden aprovechar configuraciones preestablecidas para la optimización de recursos. Colab ofrece sesiones gratuitas de a 12 horas sin usar GPU [40], [41].

4.3.4. Visual Studio Code

Visual Studio Code (VS Code) es un entorno de desarrollo integrado (IDE) multiplataforma, desarrollado por la empresa Microsoft, iene una interfaz personalizable y extensible, lo que permite a los desarrolladores adaptar el entorno en base a las necesidades que tiene. VS Code tiene un amplio soporte de lenguajes de programación, tales como Python, Java, Jascript, Go, C++, entre otros. Su interfaz intuitiva y funciones avanzadas, como el autocompletado de código, depuración en tiempo real y terminal integrado, lo convierten en gran opción tanto para principiantes como para desarrolladores experimentados [42], [43], [44].

4.3.5. Pytorch

PyTorch es una biblioteca de aprendizaje profundo de código abierto desarrollada por Meta AI la misma la misma que es una de las herramientas más importantes para la construcción de redes neuronales profundas, así como la implementación de modelos de aprendizaje automático y aprendizaje profundo, una de las principales características de PyTorch es su dinamismo en los gráficos computacionales, lo que significa que los gráficos se pueden construir y modificar de forma dinámica mientras se ejecuta el código, esta característica permite una mayor flexibilidad durante la experimentación y la depuración, ya que se pueden realizar cambios sobre la marcha [45].

PyTorch tiene una documentación y soporte para una gran variedad de operaciones en tensores, así como acceso a funciones avanzadas como autograd y módulos de optimización para la actualización de pesos, lo que facilita la implementación de modelos personalizados, la librería PyTorch tiene compatibilidad con GPUs lo que le permite acelerar el entrenamiento de modelos con el hardware disponible, es una característica permite a los desarrolladores utilizar GPUs para acelerar los tiempos de entrenamiento y optimización [46]. PyTorch es utilizado en diversas áreas como visión por computadora, procesamiento de lenguaje natural, uno de los casos más comunes es en el uso de crear y entrenar la red ResNet que detecta si una persona usa o no una mascarilla con un porcentaje de precisión del 97% en la clasificación[47], también PyTorch ha permitido la implementación de técnicas avanzadas de optimización, como los optimizadores de gradiente fraccional [48].

4.3.6. Speech Recognition

El reconocimiento de voz Automatic Speech Recognition (ASR), es una tecnología diseñada para transformar señales de voz en texto comprensible por máquinas. Su propósito es facilitar interacciones naturales y fluidas, donde las personas puedan comunicarse con sistemas mediante comandos o preguntas habladas, promoviendo una experiencia de usuario más accesible y eficiente. La arquitectura típica de un sistema ASR se compone de tres elementos clave: el modelo acústico que se encarga de traducir las características del audio en unidades fonéticas, también el modelo de lenguaje el cual utiliza estadísticas sobre ocurrencias de palabras para mejorar la precisión del texto generado y finalmente el diccionario de pronunciación, que relaciona las palabras con sus representaciones fonéticas [49]. En aplicaciones modernas estos componentes se ven potenciados por técnicas de redes neuronales profundas, como las arquitecturas Transformer, que mejoran significativamente la precisión y flexibilidad del reconocimiento [50].

El ASR tiene aplicaciones en sectores como salud, banca, telecomunicaciones, educación, seguridad, entre otras, donde se utiliza para autenticación biométrica mediante huellas de voz o como interfaz manos libres en dispositivos inteligentes [39].

4.3.7. Audacity

Es un software libre de edición y grabación de audio utilizado en campos educativos, científicos y creativos debido a su versatilidad y facilidad de uso, el programa permite la manipulación de señales de audio tanto internas como externas al computador, mediante herramientas para analizar, editar y procesar sonidos, dentro de sus capacidades más importantes está la grabación de señales de audio mediante micrófonos o dispositivos externos [44].

En el ámbito educativa la herramienta Audacity ha sido utilizada como una herramienta didáctica para enseñar conceptos de ondas sonoras, tales como frecuencia, amplitud y periodo. De igual manera en estudios científicos, el software es de gran ayuda para grabar y analizar espectros de sonidos como el canto de seres humanos, animales por su precisión para identificar frecuencias dominantes y variaciones temporales en las señales de audio [51], [52].

Además, el Software Audacity ofrece una experiencia accesible para usuarios no técnicos, ya que su tiene una interfaz intuitiva lo que permite la aplicación de filtros, normalización de volúmenes y segmentación de audio. Esto lo hace una herramienta practica y fácil de usar al momento de grabar y editar audios [53].

4.3.8. Trabajos Relacionados

En la **Tabla 1** se presentan investigaciones enfocadas en datasets y evaluaciones de sistemas de detección de voz sintética y spoofing, destacando aquellos que utilizan técnicas de aprendizaje profundo y procesamiento de señales de audio multilingüe.

Tabla 1. Trabajos relacionados.

Código	Título	Resumen	Fuente
TR01	Voice anti-spoofing data-set built from Latin American Spanish accents implementing voice conversion and text-to-speech technique	El trabajo de Pablo Andrés Tamayo Flórez aborda la creación de un conjunto de datos anti-spoofing de voz con acentos del español latinoamericano, generando muestras falsas con acentos de Colombia, Chile, Perú, Venezuela y Argentina mediante técnicas de conversión de voz y texto a voz, compara el rendimiento de varios modelos de conversión de voz, como StarGAN, CycleGAN y modelos basados en difusión, evaluando su efectividad con la métrica de tasa de error igual (EER). El objetivo es mejorar los sistemas anti-spoofing al incorporar una variedad de acentos y proporcionar una metodología integral para la construcción y evaluación de conjuntos de datos	[11]
TR02	A Crowdsourced Multidialectal Corpus of Latin American Spanish	El presente trabajo es un corpus multidialectal del español latinoamericano, recolectado de hablantes nativos de dialectos como el argentino, chileno, colombiano, peruano, puertorriqueño y venezolano, el corpus facilita el desarrollo de tecnologías de habla, como text-to-speech (TTS) y reconocimiento automático del habla (ASR).	[2]
TR03	STARGAN-VC: NON-PARALLEL MANY-TO-MANY VOICE CONVERSION WITH STAR GENERATIVE ADVERSARIAL NETWORKS	El presente trabajo es sobre la red StarGAN-VC, la cuál es un método de conversión de voz que utiliza redes generativas adversariales (GAN) sin requerir datos paralelos, transcripciones ni alineación temporal, esta red aprende múltiples mapeos con una única red generadora lo que permite la generación de voz y necesita solo unos minutos de ejemplos de entrenamiento. Las evaluaciones subjetivas indican que StarGAN-VC ofrece una calidad de sonido y similitud de hablante superiores a otros métodos avanzados basados en GAN de codificación variacional.	[1]
TR04	Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges	La presente investigación examina los métodos de identificación de hablantes, analizando desde la captura de datos vocales hasta la evaluación de modelos avanzados, por lo que en el estudio se ve la importancia de las características acústicas y biométricas de la voz para aplicaciones en seguridad, verificación forense y control de acceso, donde se destacan dos enfoques principales para la extracción de características: el manual tradicional y el automatizado mediante aprendizaje profundo, resaltando además la relevancia del preprocesamiento de señales para optimizar la precisión en la identificación.	[54]
TR05	Open-Source High Quality Speech Datasets for Basque, Catalan and Galician	El presente trabajo tiene como objetivo principal el desarrollo de recursos lingüísticos de alta calidad y de acceso abierto para el desarrollo de tecnologías del habla en tres lenguas minoritarias	[55]

	de España: el vasco, el catalán y el gallego. Para ello, se han creado tres corpus de habla multivocal con más de 33 horas de grabaciones provenientes de 132 hablantes nativos. Estas grabaciones incluyen transcripciones detalladas y se han diseñado para aplicaciones como la síntesis de texto a voz (TTS), el reconocimiento automático del habla (ASR) y el análisis fonético y fonológico.	

5. Metodología

El presente trabajo de integración curricular (TIC) de nombre “Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en español”, estuvo centrado en la creación de un dataset de audios con su respectiva limpieza y etiquetado para ser utilizados en una Red Stargan para crear audios de suplantación de identidad con su respectiva evaluación en un modelo LFCC-LCNN preentrenado en español.

En el punto **Sección 5.1** se detalla la información respecto al área de estudio, en la **Sección 5.2** se detallaron las tareas que fueron parte de la experimentación y finalmente en la **Sección 5.3** se detallaron los recursos utilizados.

5.1. Área de estudio

La recolección de las muestras de estudio se las obtuvo en la Universidad Nacional de Loja, específicamente en la Facultad de Energía de los Recursos no Renovables en la Carrera de Computación, la cual está ubicada en las calles Av. Pio Jaramillo y Eduardo Kigman, en la Figura 3**.** se muestra la ubicación aproximada en las coordenadas -4.030370 de latitud y -79.199524 de longitud.

Figura 3. Universidad Nacional de Loja, Carrera de Computación**.**

5.2. Procedimiento

Para llevar a cabo el objetivo general del TIC, se basó en la metodología CRISP ML(Q) donde se tomó en los pasos y procedimientos adaptándolos a los objetivos del presente trabajo, para lo cual se desglosaron las actividades de la siguiente manera:

5.2.1. Objetivo 1: Crear un dataset de audios con acento ecuatoriano de una muestra de 15 personas de la carrera de Computación de la Universidad Nacional de Loja para la creación de un dataset de voice spoofing, utilizando la red StarGAN para su procesamiento.

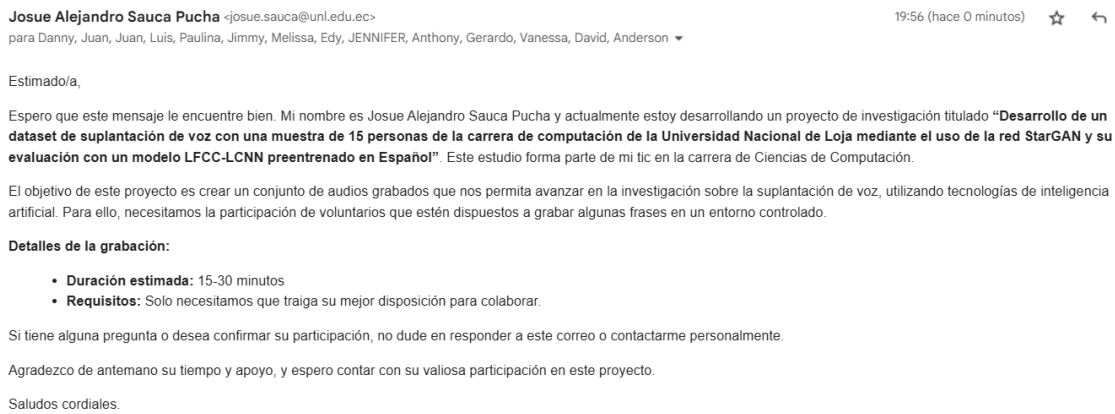
5.2.1.1. Recolección de datos

Para la recolección de datos se inició con la primera fase de la metodología CRISPML(Q), la cual consta de la comprensión del negocio y de los datos, una vez identificado el paso se procedió a crear un dataset desde cero. Para dar inicio con el dataset se utilizó la técnica no probabilística del muestreo por conveniencia según [56], [57], la cual habla acerca de la facilidad de obtener información en base a una muestra accesible y disponible para el investigador sin la necesidad de establecer criterios estrictos de selección.

Selección de la muestra de estudio

En base a estudios previos [2], [55], los cuales muestran diversos dialectos de español latinoamericano, el dataset en cuestión consta de un conjunto de audios tanto del género masculino como del género femenino, pero se optó por trabajar con una muestra reducida con un total de 15 participantes de la carrera de Computación de la Universidad Nacional de Loja.

En la Figura 4 se muestra realizó una invitación mediante el uso del correo electrónico, para aplicar la técnica del muestreo por conveniencia, en este caso se vio la disponibilidad horaria de cada participante como el principal criterio.



Josue Alejandro Sauca Pucha <josue.sauca@unl.edu.ec> 19:56 (hace 0 minutos)

para Danny, Juan, Juan, Luis, Paulina, Jimmy, Melissa, Edy, JENNIFER, Anthony, Gerardo, Vanessa, David, Anderson

Estimado/a,

Espero que este mensaje le encuentre bien. Mi nombre es Josue Alejandro Sauca Pucha y actualmente estoy desarrollando un proyecto de investigación titulado **"Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en Español"**. Este estudio forma parte de mi tic en la carrera de Ciencias de Computación.

El objetivo de este proyecto es crear un conjunto de audios grabados que nos permita avanzar en la investigación sobre la suplantación de voz, utilizando tecnologías de inteligencia artificial. Para ello, necesitamos la participación de voluntarios que estén dispuestos a grabar algunas frases en un entorno controlado.

Detalles de la grabación:

- **Duración estimada:** 15-30 minutos
- **Requisitos:** Solo necesitamos que traiga su mejor disposición para colaborar.

Si tiene alguna pregunta o desea confirmar su participación, no dude en responder a este correo o contactarme personalmente.

Agradezco de antemano su tiempo y apoyo, y espero contar con su valiosa participación en este proyecto.

Saludos cordiales.

Figura 4. Invitación cordial al presente proyecto.

Criterios para la generación de las frases

Una vez identificado la muestra se procedió a la generación de las frases que cada participante grabó, se tomaron en cuenta trabajos relacionados [55][2], donde con la ayuda del Modelo Extenso de Lenguaje Chatgpt[1], se pudo identificar [10], [11]:

Contextos cotidianos

Las frases de los trabajos relacionados pertenecen a contextos y situaciones cotidianas que reflejan el habla de los distintos países donde se abordó el estudio.

- Los corazones de pollo son una delicia
- Es un plato muy nutritivo
- Los sandwiches están fríos
- He estado pensando que necesito hacer ejercicio

Expresiones coloquiales

Las frases abordan variaciones en el uso del lenguaje, incluyendo expresiones coloquiales y modismos de cada región.

- ¡Qué bajón, che!
- ¡Cachai o no cachai!
- Estoy muy emocionado porque está Madona aquí en Panamá.

Longitud de las frases

Las frases tienen una longitud adecuada, evitando que sean demasiado largas o excesivamente cortas, con algunas excepciones.

- La oscuridad del pozo era obscena por naturaleza.
- En este momento estoy enviando a sus mails unos links de unas meditaciones en You Tube.
- Tienes que entrar al metro por la línea nueve, según Google esta línea tiene tres estaciones de transbordo.
- Tiene el tiempo contado.

Análisis de parámetros técnicos para la generación del dataset de audios

Para identificar los detalles técnicos de los audios, se usó un recurso en línea Maztr Audio File Analyzer[2] que permite evaluar las características técnicas de los audios del dataset original. En este paso, se seleccionaron de manera aleatoria dos archivos de los trabajos relacionados [2], [55]para su análisis, se va a tener en cuenta los siguientes parámetros a continuación:

[1] Modelo lllm https://chatgpt.com/
[2] MAZTR: Free Online Audio File Analyzer

- Formato de archivo
- Frecuencia de muestreo
- Tipo de canal
- Tamaño del archivo
- Duración
- Codificación del archivo

Proceso para grabar los audios

En la Figura 5 se muestra el equipo armado que se utilizó para la grabación de los audios, el mismo que tiene un micrófono detallado en la Tabla 2 y Tabla 3 un filtro de sonido.

Tabla 2. Característica del micrófono utilizado.[3]

Característica	Descripción
Marca	Logitech for Creators
Modelo	Blue Snowball
Tipo de Micrófono	Condensador cardioide
Conectividad	USB
Color	Blanco
Frecuencia de Muestreo	44.1kHz
Frecuencia de Bist	16 bist
Respuesta de Frecuencia	40 Hz – 18 kHz
Peso	460 g
Compatibilidad	Windows® (10 o posterior), Mac® (10.14 o posterior)

Tabla 3. Característica del filtro de sonido.[4]

Característica	Descripción
Marca	Apogee
Modelo	PS01
Color	Negro
Material	Nailon
Largo	35 cm
Número de Capas de Filtro	2
Brazo Articulado	Si

[3] Venta del micrófono Snowball iCE - Micrófono USB Plug and Play | Logitech G
[4] Venta del filtro Pantalla Filtro Antipop Para Micrófono - Ps01 | MercadoLibre

Figura 5. Micrófono Blue Snowball con el Filtro de Ruido.

5.2.1.2. Preprocesamiento de datos

Una vez tomado en cuenta los condiciones para la generación de las frases y las condiciones para graba, se procedió al preprocesamiento de los audios dentro del marco metodológico CRISP-ML(Q) el cual fue la limpieza de los datos, para ello se tuvo en cuenta los siguientes puntos:

- **Eliminar ruidos a medida de lo posible en los audios:** Se eliminaron los sonidos que podían presentar todo tipo de ruido para los audios grabados.
- **Normalización de audios:** Se ajustó el volumen de todos los audios para que tenga un nivel uniforme y que ciertas partes no tengan mayor intensidad que otras.
- **Comprimir los audios:** Aquí se disminuyó la diferencia entre las partes más altas y más bajas del sonido para que el volumen sea más parejo.
- **Ecualizador de la curva de filtros en graves y agudos:** Aquí se realzó las frecuencias graves y agudas de los audios para darles mayor profundidad en los bajos y más claridad en los agudos.
- **Normalización de audios:** Luego de haber realizado los pasos anteriores nuevamente se realiza una segunda normalización para asegurar que el volumen general del audio esté balanceado tras los cambios realizados.
- **Eliminar el ruido de los audios:** Se identificó el perfil de ruido a partir de toda la grabación y luego se aplicó un filtro en la pista de audio para reducir al máximo posible esos ruidos no deseados.
- **División y exportación de los audios:** Los audios se dividieron en los segmentos necesarios y se exportan en el formato que se indicó en las **Tabla 5** y **Tabla 6.**

5.2.1.3. Transformación de datos mediante StarGAN

Una vez generado los audios en el punto en cuenta los condiciones para la generación de las frases y las condiciones para grabar, viene una parte bastante importante dentro del

proyecto el cual es la generación de audios mediante el uso de la Red Stargan[5] para lo se llevaron a cabo las siguientes etapas:

- **Preparación de los datos:** En este apartado los datos se deben organizar de manera adecuada para que no exista dificultad al momento de usarlos en el código.
- **Preprocesamiento de los Audios:** En este apartado se envía el dataset generado en la fase de Preparación de los datos, donde va a generar unos archivos de formato **npy** los cuales contienen características acústicas extraídas de los audios que se van a ocupar en la fase de entrenamiento.
- **Entrenamiento del Modelo StarGAN:** En este apartado se utiliza una red StarGAN que tiene unos valores por defecto la cual se debe entrenar y va a generar un modelo que será capaz de convertir audios verdaderos en audios generados por Inteligencia Artificial Generativa.
- **Conversión de Voz entre Hablantes:** Finalmente en este apartado se utiliza el modelo generado en la fase del entrenamiento donde se le indica a la voz original que se quiere convertir con la voz objetivo que se la quiere mezclar.

5.2.1.4. División de los Datos

Como parte de la tarea final del tratamiento del dataset, se considera la división de datos, por lo que se debe hacer una separación para datos de entrenamiento y pruebas, por lo que se tienen en cuenta tareas como:

- **Preparación de los Datos:** En este apartado se va a trabajar con el dataset original que no tiene ningún tipo de orden.
- **Aleatorización de los Datos:** En este apartado para evitar sesgos en la asignación de los archivos a los conjuntos de entrenamiento y prueba, se va a utilizar un generador de números aleatorios con una semilla fija, para que los resultados no cambien y sean reproducibles.
- **Separación de Datos:** La división de los datos se realizó utilizando la proporción **80-20** (80% de los datos para entrenamiento y 20% para prueba).
- **Organización en Carpetas:** Finalmente en este apartado los datos se copiaron y organizaron en dos carpetas principales: la carpeta train contiene los datos asignados al conjunto de entrenamiento, mientras que la carpeta test contiene los datos reservados para la evaluación del modelo.

[5] Codigo base stargan GitHub - liusongxiang/StarGAN-Voice-Conversion

5.2.2. Objetivo 2: Utilizar el modelo LFCC-LCNN con el dataset generado para medir la métrica Tasa de Error de Igualación en la muestra de 15 personas de la Carrera de Computación de la Universidad Nacional de Loja.

5.2.2.1. Evaluación del Modelo

Como parte de la evaluación del modelo se tomaron en cuenta una serie de pasos como:

- **Preparación y ajuste del dataset:** Se realizó ajustes al dataset generado a las carpetas de evaluación y test, ya que no seguían el formato adecuado para usarlas con el código y modelo que se utilizó.
- **Carga del modelo y configuración del código:** Se realizó la carga del código que se va a utilizar, así como el modelo con el que se evaluó el dataset adaptado, se tuvo que eliminar archivos que no eran relevantes para el experimento y ajustar rutas de los scripts.
- **Generación de los logs de la evaluación:** Se ejecutó el código encargado de evaluar el desempeño del dataset creado con el modelo utilizado.
- **Cálculo de la métrica:** Se realizó la evaluación de la métrica en base a los logs generados donde se obtuvo el valor de la Tasa de Error de Igualación, además se realizaron experimentos con el dataset generado para evaluar el rendimiento del modelo.
- **Documentación de la evaluación:** Se documentó los resultados obtenidos en la fase de evaluación con la finalidad de cumplir con el objetivo 2.

5.3. Recursos

5.3.1. Recursos Científicos

- **StarGAN:** Un modelo de red generativa que permite generar audios sintéticos a partir de una muestra recolectada de dataset, que sirvió para completar gran parte de la investigación.
- **Muestreo por conveniencia:** Esta fue la técnica utilizada al momento de la selección de personas que van a ayudar con el proyecto ya que no se tuvo en cuenta un muestreo estadístico, sino fue en base a las necesidades del investigador.
- **Metodologia CRISP-ML(Q):** Metodologia utilizada para llevar a cabo el proyecto y cubrir en gran parte las fases de hacer proyectos relacionados al área de inteligencia artificial.

5.3.2. Recursos Técnicos

- **Equipo de sonido para la grabación:** Como se indicó la sección **Proceso para grabar los audios** en la Tabla 2 y Tabla 3, los equipos de sonido usados fueron un micrófono y un filtro de ruido para tener un mejor resultado a la hora de grabar las muestras de audios.
- **Google Colab:** En este proyecto se utilizaron las GPUs de Google Colab debido a la falta de recursos computacionales locales adecuados para entrenar modelos de inteligencia artificial, por lo que se contrató un plan básico del precio de $10 dólares.
- **Software de grabación y edición de audio**: Se utilizó el programa **Audacity** para la grabación, limpieza y segmentación de los audios.
- **Python**: Lenguaje de programación utilizado en la ejecución del proyecto.

5.3.3. Participantes

Los participantes del presente proyecto que ayudaron al desarrollo de del mismo fueron:

- Josue Alejandro Sauca Pucha, estudiante encargado del desarrollo del Trabajo de Integración Curricular TIC y del cumplimiento de los objetivos planteados.
- Ing. Roberth Gustavo Figueroa Díaz, encargado de la supervisión a lo largo del desarrollo del proyecto y correcciones del mismo.
- Los 15 estudiantes de la carrera de computación que se utilizó para el muestreo por conveniencia como se observa en el **Anexo 2. Invitación y respuesta del muestreo por conveniencia.**

6. Resultados

6.1. Objetivo 1: Crear un dataset de audios con acento ecuatoriano de una muestra de 15 personas de la carrera de Computación de la Universidad Nacional de Loja para la creación de un dataset de voice spoofing, utilizando la red StarGAN para su procesamiento.

Para cumplir con el primer objetivo planteado se adaptó mediante la adaptación de la primera fase de la metodología CRISP-ML(Q) donde se utilizó las fases Comprensión del negocio y de los datos, en este apartado se detallaron las tareas realizadas en cada fase.

6.1.1. Recolección de datos

Selección de participantes

La muestra seleccionada fueron 15 participantes utilizando la técnica de muestreo por conveniencia, uno de los criterios más importantes para la selección fue la disponibilidad horaria, los detalles de los participantes seleccionados, así como sus nombres, se presentaron en el **Anexo 2. Invitación y respuesta del muestreo por conveniencia.** Las muestras recolectadas fueron 15 con una duración aproximada de 5 a 8 minutos cada una.

Criterios para la generación de las frases

Como se mencionó en el punto Criterios para la generación de las frases, se realizó el proceso de generación de audios siguiendo los pasos descritos a continuación:

- **Elección de la muestra a comparar:** En la Figura 6 se muestra varios guiones al azar. Como primer paso se realizó la elección de las frases ya creadas del conjunto de los trabajos relacionados [2], [11], [55], luego se escogió de manera aleatorio alrededor de 100 frases por cada dataset no se utilizó algún criterio en específico simplemente se eligió de manera manual ya sea desde el inicio, la mitad o desde el final.

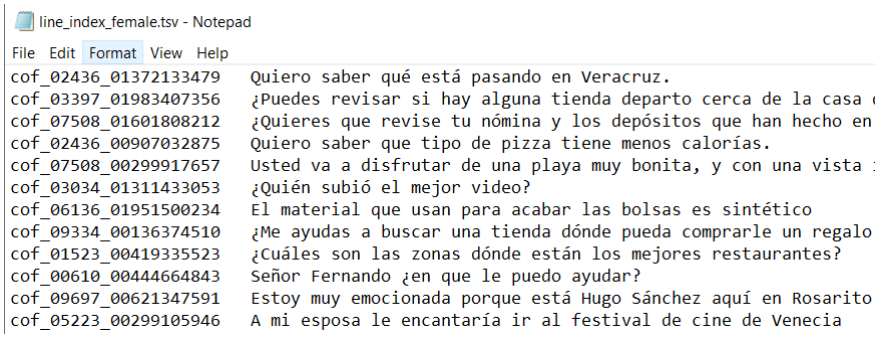


Figura 6. Frases de guiones de los trabajos relacionados.

- **Uso del LLM Gpt para la generación de nuevas frases:** En la Figura 7 se muestra el proceso posterior a la selección de las frases de los distintos archivos, una vez

seleccionado las frases de los distintos archivos, se procedió a usar la Gpt, por lo que se ingresaron inicialmente las frases seleccionadas.

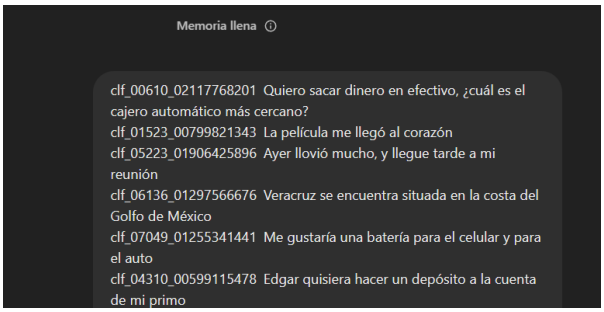


Figura 7. Ingreso de frases a gpt.

En la Figura 8 se mostró el proceso posterior a la inserción de las frases en el modelo GPT, luego de este paso el modelo generó una serie de opciones sobre las posibles acciones que se podían realizar con las frases ingresadas.

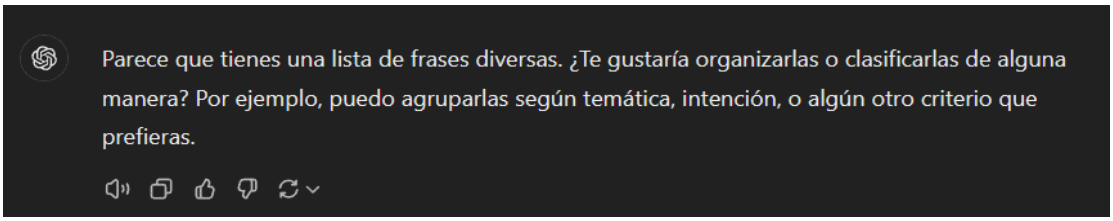


Figura 8. Opciones que brinda GPT

A continuación, se le indico las instrucciones que se quería lograr con dichas frases por lo que se ingresó una lista de inputs al modelo LLM Gpt para generar frases acordes a los Criterios para la generación de las frases.

Tabla 4. Ejemplos de inputs para la generación de frases mediante el uso de GPT.

Descripción del Prompt	Texto del Prompt
Generación de nuevas frases con acento ecuatoriano (Loja)	"En base a las frases dadas, generar 70 frases nuevas con el mismo formato, pero con acento ecuatoriano y fonética ecuatoriana enfocada en la provincia de Loja, sin mencionar tanto las palabras 'Loja' y 'Ecuador'."
Generación de frases tropicalizadas	"En base a las siguientes frases dadas, quiero que me generes 70 frases más con palabras ecuatorianas, tropicalizando las mismas."
Generación de frases adicionales adaptadas a Loja	"En base a las siguientes frases dadas, quiero que me generes 70 frases más con palabras ecuatorianas, tropicalizando las mismas, enfocadas en Loja."
Frases con enfoque en Loja y sierra ecuatoriana	"En base a las frases dadas, generar 70 frases nuevas que incluyan referencias a la región andina de Ecuador, especialmente Loja y sus alrededores, sin mencionar directamente el nombre de la provincia."

Finalmente, en la Tabla 4, se obtuvo una seria de frases acorde a los criterios mencionados, y el resultado de los inputs fueron algunas frases como:

- Anda, dile a tu ñaño que venga a comer.
- Aquí el cuy es lo mejor para un almuerzo dominguero.
- Ya pues, no te hagas el malito, ponte las pilas.
- ¿Te acuerdas de la última vez que fuimos a bailar en La Tebaida?
- Aquí la tradición manda, hay que hacer el repe para la cena.
- Ese man es más lojeño que el hornado del mercado.
- No te olvides de traer poncho, acá el clima de la sierra es traicionero.
- Ñaño, hagamos una colada morada bien lojana este feriado.

Luego de que se obtuvieran las frases acordes a los criterios generados se almaceno 15 guiones para cada participante, cada uno con un total de 70 frases, dando como resultado las 1050 frases necesarias para el proyecto.

Análisis de parámetros técnicos para la generación del dataset de audios

A partir del análisis realizado, se obtuvieron los siguientes resultados, que muestran las características técnicas de los audios seleccionados:

Tabla 5. Característica del audio al azar*.*

Parámetro	Valor
Nombre del archivo	arf_00295_00000740990.wav
Canales	1 (mono)
Frecuencia del muestreo	16,000 Hz
Precisión	16-bit
Duración	00:00:04.52
Tamaño del archivo	145 k
Codificación de la Muestra	16-bit Signed Integer PCM

Tabla 6*.* Característica del audio al azar*.*

Parámetro	Valor
Nombre del archivo	arm_06136_00077283089.wav
Canales	1 (mono)
Frecuencia del muestreo	16,000 Hz
Precisión	16-bit
Duración	00:00:06.74
Tamaño del archivo	216 k
Codificación de la Muestra	16-bit Signed Integer PCM

Con base en la **Tabla 5** y **Tabla 6** , se pudo evidenciar que los audios cumplen con las mismas características técnicas como, la frecuencia de muestre, canal mono y la codificación de 16-bit Signed Integer PCM. Las características presentadas sirvieron de referencia para

asegurar la coherencia en la generación de los audios posteriores, manteniendo la misma calidad técnica y facilitando un procesamiento adecuado.

Proceso para grabar los audios

Para cumplir con ese punto importante para crear el dataset se tuvo en cuenta los siguientes pasos:

- **Condiciones para grabar los audios**: Al momento de realizar las grabaciones de los audios se tuvo en cuenta que no se contaba con un estudio profesional de grabación para hacerlo, por lo que se tuvieron en cuenta las siguientes consideraciones:
    - **Ambientes seleccionados:** Se vio lugares que tengan poco eco, por lo que se evitó espacios grandes para no afectar del todo la calidad del sonido. Las grabaciones fueron en aulas pequeñas de la Universidad Nacional de Loja en horarios poco concurridos para que no exista mucho ruido, también se utilizó casas al ser lugares tranquilos y no tienen mucho ruido exterior.
    - **Configuración de los equipos:** En la Figura 9 con ayuda del Software Audacity[6], se pudo configurar el micrófono para que sea detectado en el ordenador donde se lo va ocupar.

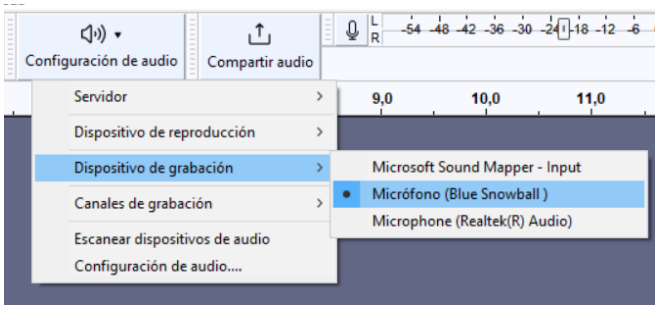


Figura 9. Configuración del micrófono en el software Audacity.

- **Ajuste del micrófono para la grabación*:*** En la Figura 10 se indica la colocación del micrófono a una distancia prudente de la persona que se va a grabar la voz, adicionalmente se realizaron pruebas de sonido para verificar que funcione bien.

Figura 10. Preparación del micrófono con las personas.

[6] Software Audacity Audacity ® | Downloads

- **Instrucciones a los participantes**: Al momento de realizar las grabaciones de los audios, los participantes recibieron unas pequeñas instrucciones al momento de la grabación las cuales fueron:
  - Mantener un ritmo normal para los audios, y en caso de que se equivoquen no importa paren y nuevamente retomen el ritmo.
  - También que tuvieran un volumen de voz moderado ni tan alto ni tan grabe.
  - Pueden repetir las frases algunas veces en caso de equivocarse.
  - Se recomendó que mantuvieran el acento que se busca en la muestra.
  - Si escuchan ruido de fondo parar y esperar que pase el ruido.
- **Grabación de los audios**: Una vez entendido lo pasos anteriores se procedió a grabar los audios el cual es el enfoque principal, para ello en base a las frases obtenidas en el punto Criterios para la generación de las frases, se otorgó un guion a cada participante para que lo lea durante la grabación.
  En la Figura 11 cada participante leyó una por una hasta completar las 70 frases, la lectura resultó ser algo cansada, ya que el número considerable de frases requería tiempo, lo que ocasionaba fatiga en los participantes. Sin embargo, se logró cumplir con el cometido.

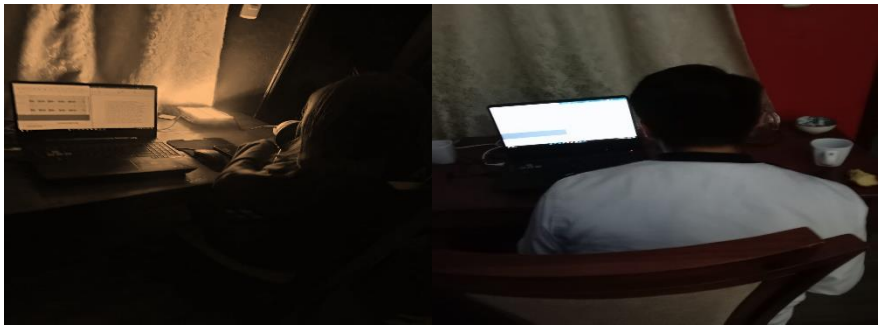


Figura 11. Grabación de los audios con las personas.

- **Guardar las grabaciones:** El punto final a ser considerado fue el guardar las grabaciones de manera local para poder continuar con el punto Preprocesamiento de datos, también se tuvo a consideración respaldar los archivos de los audios para mayor seguridad.

6.1.2. Preprocesamiento de datos

Ruido en los audios

Uno de los problemas más que se encontraron al momento de hacer la limpieza de los audios fue el ruido, el mismo se originaba ya sea de manera exterior, así como al momento de hacer la grabación, los más comunes que se encontraron fueron:

- **Ruido ambiental:** sonidos de fondo como viento, movimiento de carros, sonido del claxon de carros, conversaciones cercanas que afectaban la calidad de los audios.
- **Ruido de manejo del equipo:** sonidos generados por el movimiento o contacto con el micrófono, como golpes o roces.
- **Ecos:** causados por la falta de aislamiento acústico en el lugar de grabación o en algunos casos porque eran aulas e igual tenían eco.

Ruidos al decir los audios

Durante la grabación de los audios se produjeron ruidos involuntarios al momento de hablar, los cuales también afectan directamente la calidad de la grabación, los más comunes encontrados fueron:

- **Tartamudeos**: interrupciones o repeticiones involuntarias de las personas a hora de grabar los audios.
- **Pauses largas**: En la Figura 12 se muestra los silencios prolongados que se encontraban a lo largo de una oración que se tuvieron que cortar.

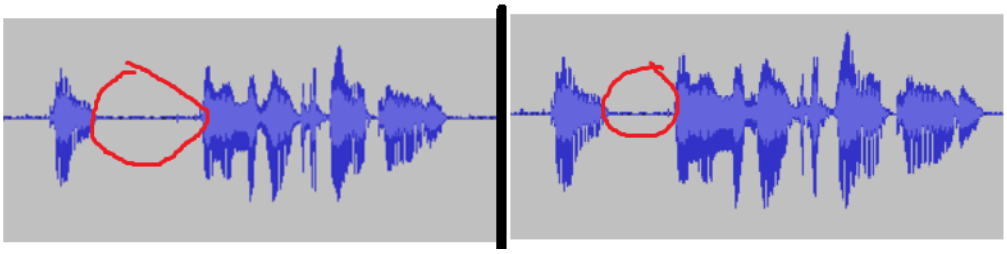


Figura 12. Audio antes y después de eliminar las pausas largas.

- **Respiraciones**: En la Figura 13 se muestra las respiraciones registradas durante la grabación de las frases, las cuales se producían al inicio o al final de cada una, las mismas que fueron eliminadas para mejorar la calidad del audio.

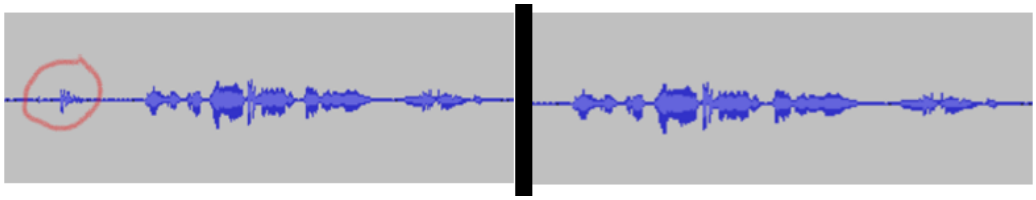


Figura 13. Audio antes y después de eliminar la respiración.

- **Velocidad de la frase**: En la Figura 14 se muestra casos en los que, durante la grabación, no se mantenía un ritmo constante en la pronunciación de las frases.

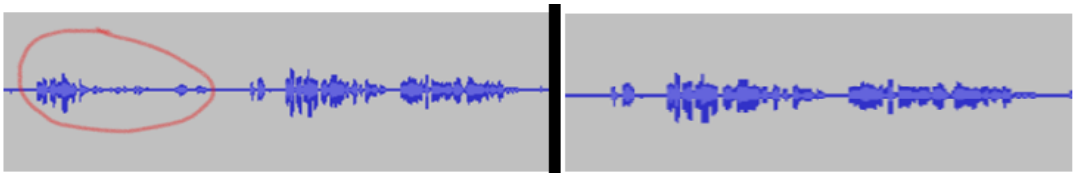


Figura 14. Audio antes y después de eliminar la parte que fue rápido.

Eliminación del ruido con Audacity

Una vez identificado los ruidos más recurrentes dentro de los audios se utilizó la herramienta Audacity donde se grabó los audios para hacer la limpieza de los mismos, por lo que se tuvo en cuenta los siguientes pasos para hacerlo:

- **Duplicar la pista de audio**: Esto se lo realizo para tener una copia en caso de cualquier falla técnica o error humano que suceda al momento de editar los audios.
- **Eliminar ruidos detectados*:*** En la Figura 15 se mostró el proceso para realizar esta parte fue escuchar audio a audio los audios grabados y mediante el uso de la herramienta se los iba eliminando cortando uno a uno.

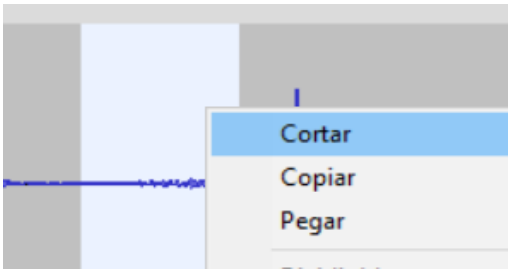


Figura 15. Uso de la función cortar para eliminar partes no deseadas del audio.

- **Normalización de los audios:** Una vez eliminados los ruidos de los audios, se procedió a realizar el proceso de normalización el cual consiste de seleccionar toda la pista de audio y seleccionar en el programa la funcióng de normalizar, el resultado de la normalización se lo presentó en la Figura 16**.**

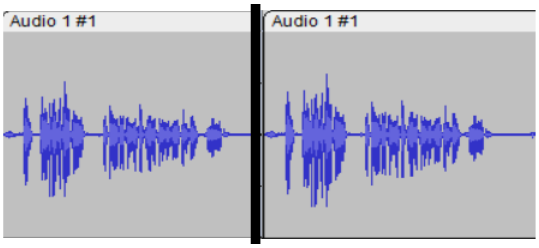


Figura 16. Antes y después de la normalización de un audio.

Parámetros de la normalización en Audacity

En la Figura 18, se observa ciertos criterios que se tuvieron en cuenta a la hora de hacer una normalización a los audios, algunos de los criterios fueron:

- **Eliminar desplazamiento DC (centrar en 0.0 verticalmente):** se utiliza para corregir problemas que pueden hacer que la forma de onda de un audio no esté centrada en la línea de 0.0 en el eje vertical. Como se observa en la Figura 17 la parte de la Onda de Desplazamiento DC se encuentra lejos y no es como la Onda Normal, esto Audacity lo corrige para mejorar la calidad del audio.

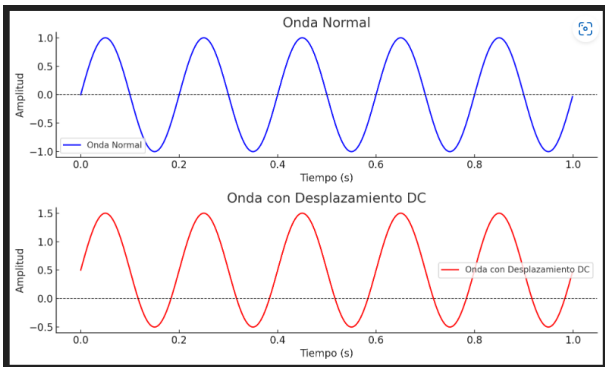


Figura 17. Onda de audio normal y una onda con desplazamiento DC**.**

- **Normalizar pico de amplitud a -1.0 dB:** El siguiente parámetro normaliza el pico de amplitud del audio, es decir, ajusta el nivel máximo de la señal de audio para que alcance un valor predeterminado, sin exceder el límite. El pico máximo que puede alcanzar una señal sin distorsión es de 0 dB, ya que superar este valor podría causar clipping, lo que distorsionaría el sonido al cortar las ondas de audio. El software Audacity tiene, por defecto, el valor de -1.0 dB para el pico de amplitud. La razón detrás de esto es que dejar un margen de 1 dB por debajo de 0 dB evita el riesgo de clipping y asegura que el audio se reproduzca de manera limpia en diferentes dispositivos y sistemas de reproducción.
- **Normalizar canales estéreo independientemente:** Finalmente, este parámetro permite ajustar el volumen de los canales izquierdo y derecho de forma separada en una grabación estéreo. Los canales en una grabación estéreo representan las dos fuentes de sonido que se escuchan por los altavoces o audífonos, uno para el oído izquierdo y otro para el derecho. Esta opción se utiliza para corregir desequilibrios cuando un canal tiene un volumen mayor que el otro, esta opción no se toma en cuenta debido a que podría desbalancear el sonido de los audios.

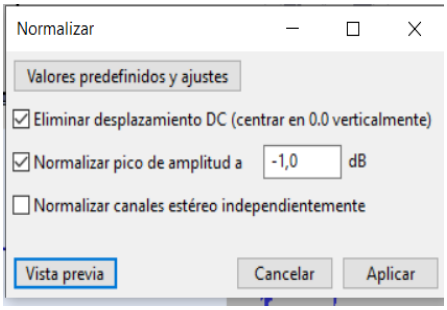


Figura 18. Parámetros de Normalización Audacity.

- **Compresor a los audios:** El compresor[7] fue una parte importante del proceso de la limpieza de los audios ya que fue de gran ayuda para que los sonidos fuertes y suaves estén más equilibrados, es decir las partes más suaves suenen más fuertes y las

[7] Referencia Compresor [Compressor - Audacity Manual]

partes más fuertes suenen más suaves como se lo observó en la Figura 19. Esto ayuda a mantener el volumen más uniforme a lo largo del tiempo y evitar que el audio se distorsione cuando llega a volúmenes muy altos.

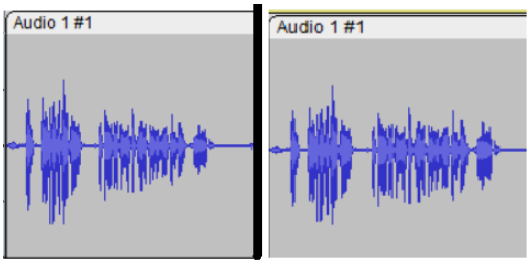


Figura 19. Audio de Normalización a Compresión

Parámetros de la compresión en Audacity

En la imagen de la Figura 20, se observa criterios que se tuvieron en cuenta a la hora de hacer una compresión a los audios, los resultados de la compresión fueron:

- **Umbral (-18 dB):** El umbral de -18 dB es equivalente al nivel típico de un diálogo normal[8], cuando cualquier sonido supera esta intensidad, el compresor reduce automáticamente su volumen, evitando así que los elementos intensos sobresalgan de manera excesiva.
- **Límite inferior de ruido (-40 dB):** El límite inferior de -40 dB corresponde a un susurro o ruido ambiental suave, esta configuración es importante para preservar los sonidos más sutiles, manteniendo la textura natural y la profundidad del audio en sus niveles más bajos.
- **Proporción (2.5:1):** Aquí se define la intensidad con la que actuará el compresor sobre los sonidos que superen el umbral de -18 dB. La presente relación significa que por cada 2.5 dB que una señal sobrepase el umbral, el compresor la reducirá a solo 1 dB por encima de este, es una relación moderada para controlar los picos de volumen sin llegar a aplastar el sonido.
- **Tiempo de ataque (1.79 seg):** Aquí se determinó cuánto tardará el compresor en empezar a reducir el volumen una vez que el audio supera el umbral de -18 dB. El ajuste es un poco lento e intencional, ya que permite que los sonidos transitorios iniciales (como golpes a los equipos, respiraciones) conserven su impacto natural antes de que el compresor comience a actuar.
- **Tiempo de decaimiento (11.1 seg):** Se indica cuánto tardará el compresor en liberar completamente su efecto una vez que el audio retorna por debajo del umbral de -18 dB, el tiempo es algo largo, pero permite que la transición entre el audio comprimido y sin comprimir sea suave y musical, evitando cambios abruptos de volumen.

[8] Audio mixing for video: What you need to know | Epidemic Sound

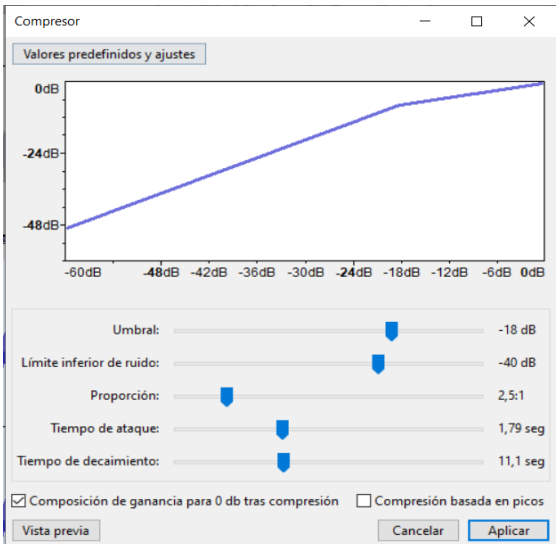


Figura 20. Compresor en Audacity.

Parámetros del ecualizador de la curva de filtros

En la imagen de la Figura 23, existen criterios que se tuvieron en cuenta a la hora de hacer una curva ecualizadora de filtros para graves y agudos, que fueron:

- **Realce de graves:** Este paso fue importante ya que permitió realizar un realce de las frecuencias graves mediante el ecualizador de audacity, lo que otorga mayor profundidad y presencia a los sonidos bajos, esto hace que los graves suenen más profundos y potentes, sin cambiar el volumen de los sonidos más agudos de manera que mantiene un equilibrio general del audio como se presentó en la Figura 21**.**

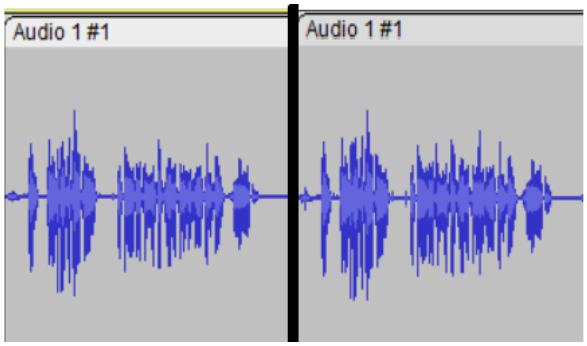


Figura 21. Audio de Compresor a la función realce agudos

- **Realce de agudos:** Este paso fue importante ya que permitió resaltar las frecuencias agudas mediante su ecualizador de audacity, lo que otorga mayor claridad y presencia a los sonidos altos, esto hace que los agudos suenen más nítidos y presentes, sin cambiar el volumen de los sonidos graves de manera que mantiene un equilibrio general del audio como se presentó en la Figura 22**.**

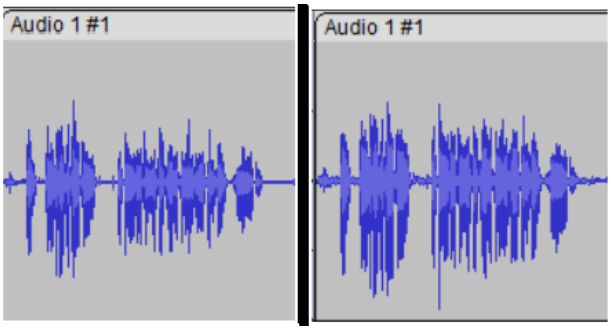


Figura 22. Audio de Realce de agudos a graves

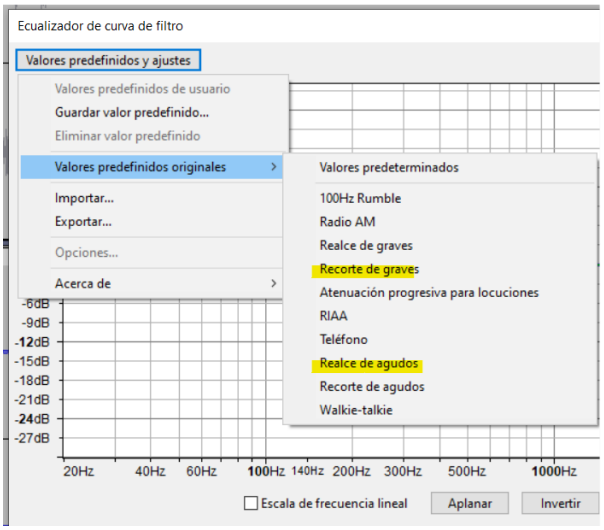


Figura 23. Curva ecualizadora de filtros para graves y agudos en Audacity.

Parámetros de la normalización de los audios

Una vez completados los pasos anteriores, se procedió a realizar nuevamente la normalización de los audios, este paso es importante porque, durante el proceso de edición de los audios, la aplicación de efectos como la normalización, compresión y la ecualización puede provocar cambios en el volumen y la onda del del audio como se lo puede observar en la Figura 16 respecto a la onda original en la Figura 19, mediante la segunda normalización, se busca restablecer un equilibrio en el volumen del audio, asegurando que las modificaciones previas no generen desajustes en la señal final como en la Figura 24**.**

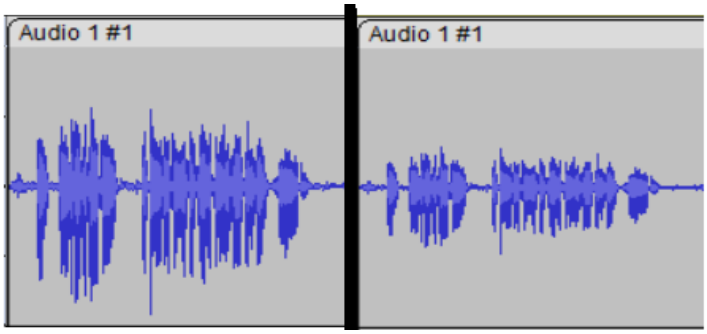


Figura 24. Audio de Realce de graves a una segunda normalización.

Obtener el perfil de ruido de los audios

Uno de los pasos finales al momento de que se realizó la limpieza de datos fue el tratamiento del ruido por lo que se tuvo en cuenta una opción bastante buena dentro de la herramienta audacity la cual fue **Reducción de Ruido[9],** esta función de la herramienta se compone de dos partes una es la obtención del perfil de ruido y la otra es reducir el ruido al audio seleccionado como se lo puede observar en la Figura 25**.**

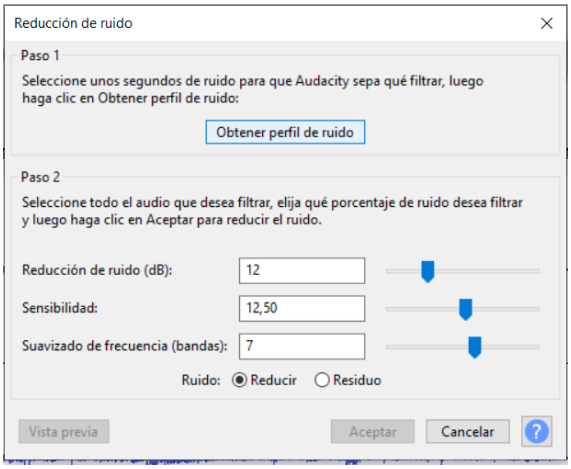


Figura 25. Función para reducir el ruido de una pista de audio

Para realizar la primera parte que es **Obtener perfil de ruido,** se debe realizar una seria de pasos previos los cuales fueron:

- Se duplicó la pista de audio para tener un respaldo de la misma.
- Una vez duplicada la pista se seleccionó la sección donde los hablantes pronunciaron las frases como la parte izquierda de la Figura 26**,** para luego recortar dichos fragmentos y que la pista quede como la parte derecha de la Figura 26, donde quedo únicamente con el segmento que contiene el ruido de los audios.

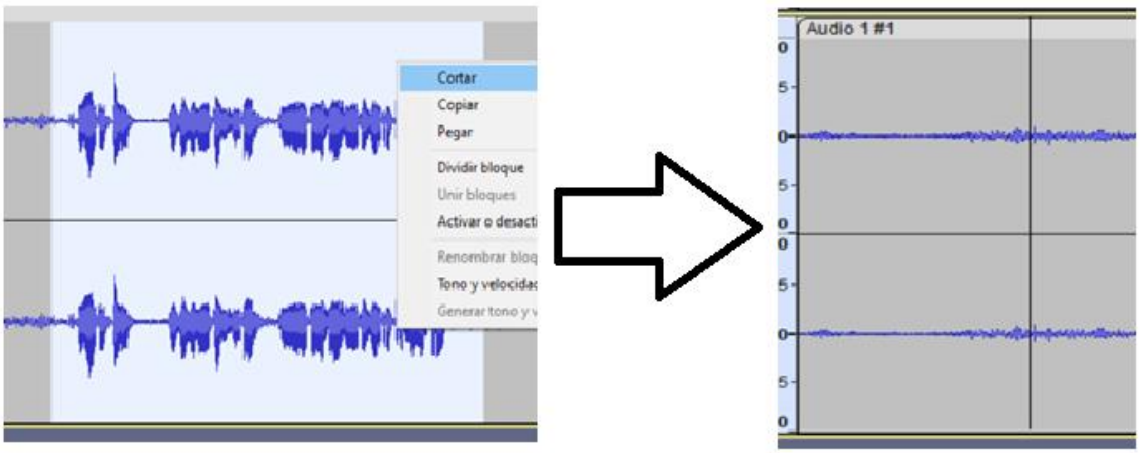


Figura 26. Sección donde los hablantes pronunciaron las frases.

[9] Noise Reduction - Audacity Manual

- Luego de obtener únicamente el ruido de las frases se seleccionó toda la pista completa y se presionó en la opción **Obtener perfil de ruido** como la Figura 26, la cual sirvió para que la herramienta identifique y almacene las características del ruido seleccionado durante el proceso de reducción.
- Luego obtener el perfil de ruido se procede a seleccionar la pista original donde se quiere eliminar el ruido y se aplicó los siguientes parámetros como la Figura 26 que fueron:
    - **Reducción de Ruido (dB) a 12:** La herramienta Audacity detecto el valor de 12 luego de obtener el perfil de ruido lo que significó que la herramienta trato de bajar un poco el volumen de fondo que considera molesto sin afectar el audio principal del mismo.
    - **Sensibilidad a 12.50:** El valor de 12.50 obtenido significa que la herramienta trata de distinguir y eliminar con cuidado el ruido y las voces de los audios, esta escala va desde 0 a 24 por lo que al estar en un término intermedio el audio no se ve tan afectado por el ruido.
    - **Suavizado de Frecuencia (bandas) a 7:** El valor de 7 obtenido para este parámetro hizo mención a que no solamente el ruido que se detecte del habla va a ser tratado sino también el ruido externo proveniente de aparatos electrónicos va a ser eliminado también sin afectar el audio principal.
    - **Ruido opción (Reducir):** Como parte final este parámetro sirvió para escuchar únicamente el ruido que se lo detecto previamente y se lo elimine en el audio principal sin que cambie mucho.
- Finalmente, en la Figura 27 se evidenció como es el resultado final de la eliminación de ruido donde en la parte izquierda se observan unas ondas que tienen cierto ruido al inicio, en la parte intermedia, final del audio y en la parte derecha se ve un audio mucho con mucha más calidad que no tiene tanto ruido.

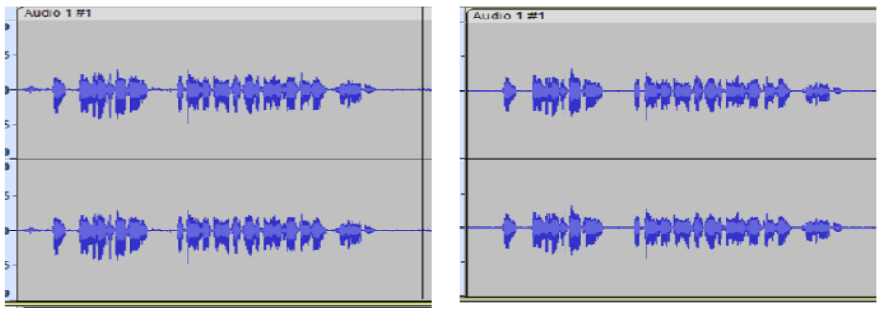


Figura 27. Antes y después de la eliminación de ruido en una pista.

Exportar los audios

El paso más importante y final del tratamiento de los datos fue la exportación de los audios según los criterios mencionados en el punto Análisis de parámetros técnicos para la generación del dataset de audios, donde se detalló más a fondo el formato de los archivos, para hacerlo se tuvieron en cuenta los siguientes pasos:

- **Selección de segmentos del audio limpio**: Se eligió uno a uno las piezas de la pista completa que fueron limpiadas como en la Figura 28, una vez se seleccionaba pieza a pieza con la ayuda del atajo de teclado Ctrl + B se indicaba al programa que segmentos se van a exportar de la pista completa como se observó en la Figura 29**.**

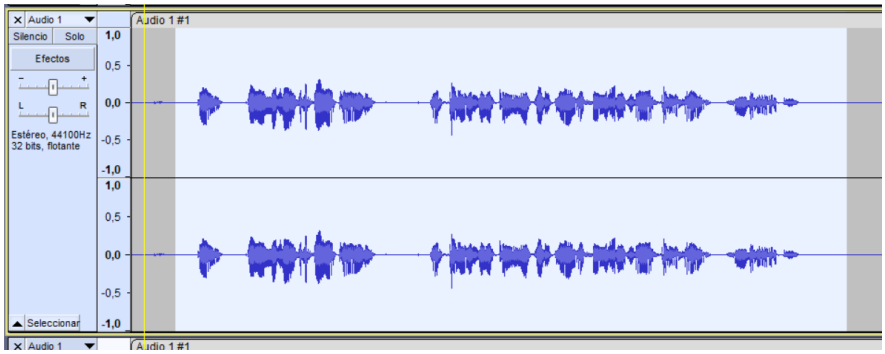


Figura 28. Selección de toda la pista de audio**.**

Figura 29. Selección de las pistas a exportar**.**

- **Exportación de los audios al formato adecuado**: Luego en el apartado Archivo → Exportar Audio, hay una pantalla como la Figura 30, donde se indica donde se van a exportar los archivos de os audios elegidos, también el formato de los audios en base a la sección Análisis de parámetros técnicos para la generación del dataset de audios, los audios se exportaron en archivos similares como la Figura 31**.**

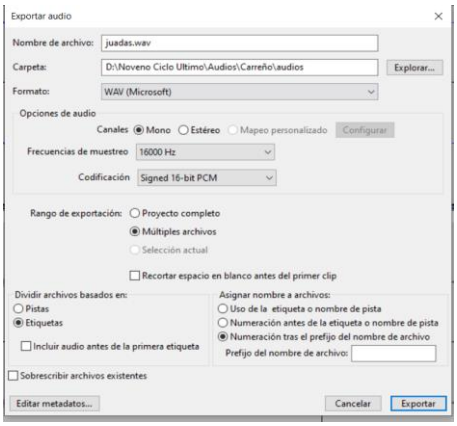


Figura 30. Parámetros para exportar los audios

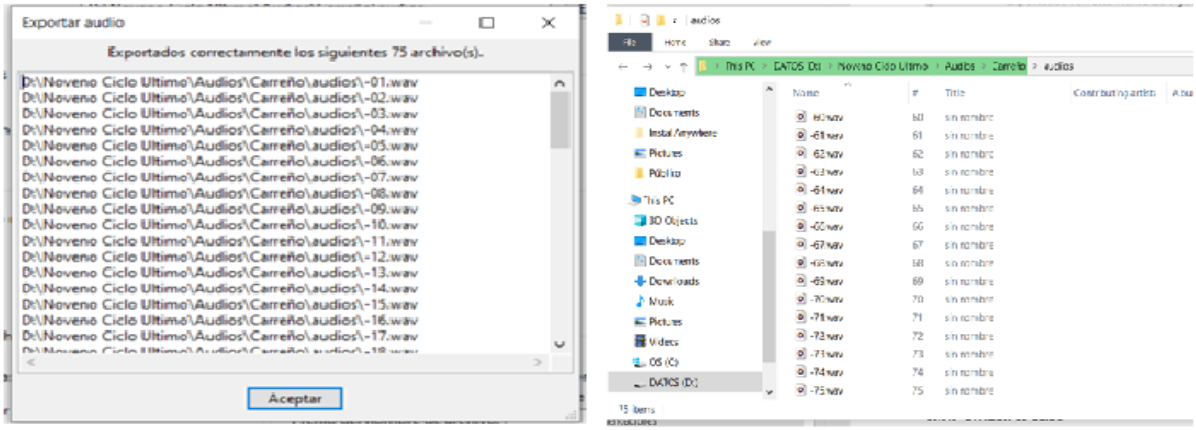


Figura 31. Archivos exportados correctamente.

- **Carpetas con audios exportados**: Una vez finalizado la exportación de los audios se obtuvo un total de 1050 audios segmentados, los cuales están en 15 carpetas distintas cada una con 70 audios.

6.1.3. Transformación de datos mediante StarGAN

Preparación de los datos

Uno de los problemas que se encontraron luego de haber generado el dataset de audios fue que a pesar de tener una numeración en la Tabla 7, en la cual se puede observar un orden, fue necesario realizar una serie de modificaciones para adaptar el código como:

Tabla 7. Algunos de los audios exportados originalmente**.**

Carpeta Almacenada	Audios exportados
audios_david	-01.wav
audios_david	-02.wav
audios_david	-03.wav
audios_jimmy	01.wav
audios_jimmy	02.wav
audios_jimmy	03.wav
audios_pablo	-01.wav
audios_pablo	-02.wav
audios_pablo	-03.wav

- **Identificación de los audios:** En este apartado los datos se deben organizar de manera adecuada para que no exista dificultad al momento de usarlos en el código.
- **Estandarización de los audios**: Una vez identificado los audios se pudo observar que tanto las carpetas como los audios no se encontraban en un formato estándar que se le puedan enviar al modelo a ser utilizado, por lo que se asignó un nombre que se

muestra en la Tabla 8, luego de que se tuvo en claro el nombre de las carpetas a renombrar.

Tabla 8. Carpetas renombradas

Carpeta Original	Carpeta Renombrada
audios_carreño	speaker01
audios_castillo	speaker02
audios_claudia	speaker03
audios_dany	speaker04
audios_david	speaker05
audios_edy	speaker06
audios_vanessa	speaker07
audios_jimmy	speaker08
audios_paulina	speaker09
audios_paula	speaker10
audios_pablo	speaker11
audios_melissa	speaker12
audios_luzuriaga	speaker13
audios_luis	speaker14
audios_jennifer	speaker15

, en el momento de asignar también un formato dentro de los audios para lo cual se tuvo en cuenta el formato speakerXX_YYY.wav, donde XX corresponde al número del hablante asignado previamente y YYY representa un número secuencial que identifica cada grabación del hablante, esta tarea se la realizo con la ayuda de un script [10]presentado en la Figura 32, donde se indica el nombre de las carpetas y el nombre de los archivos que deben seguir la numeración, también se ajustó la frecuencia de los audios a 48000 Hz .

```python
def convert_and_rename(input_dir, output_dir, prefix):
    # Asegurarse de que el directorio de salida exista
    os.makedirs(output_dir, exist_ok=True)

    # Inicializar un contador para el nombre de archivo
    counter = 1

    # Iterar sobre todos los archivos en el directorio de entrada
    for filename in sorted(os.listdir(input_dir)):  # Ordenar los archivos para asegurar secuencia
        if filename.endswith('.wav'):
            input_file = os.path.join(input_dir, filename)

            # Formatear el número del archivo con ceros a la izquierda (ej: 001, 002, ...)
            new_filename = f"{prefix}_{str(counter).zfill(3)}.wav"
            output_file = os.path.join(output_dir, new_filename)

            # Cargar el archivo de audio con pydub
            audio = AudioSegment.from_file(input_file)

            # Convertir a 48 kHz
            audio = audio.set_frame_rate(48000)

            # Guardar el archivo de audio convertido y renombrado
            audio.export(output_file, format="wav")
            print(f"Converted {input_file} to {output_file}")

            # Incrementar el contador para el próximo archivo
            counter += 1
```

Figura 32. Script para renombrar los audios y las carpetas

[10] Script Código

- **Orden de las carpetas:** Una vez realizado la estandarización de los audios se colocó los audios como la Figura 33, donde se creó dos carpetas wav48 y wav16 que van acorde a las frecuencias, para lo cual se procedió a completar la carpeta de wav48 ya que contiene los archivos renombrados y la carpeta wav16 se explicó su uso en el paso Preprocesamiento de los Audios.

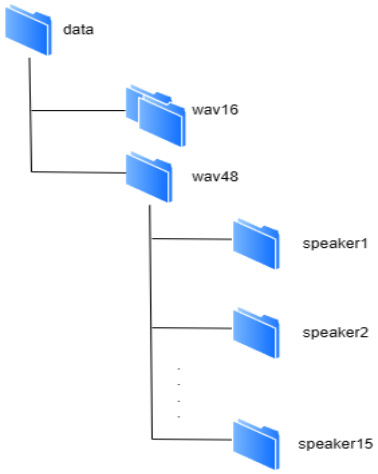


Figura 33. Orden de los archivos con los audios y carpetas generadas.

En la Figura 34 se puede observar una de las carpetas al antes en la parte izquierda de la imagen y en la parte derecha se puede observar cómo queda una vez fue renombrada, en este caso fue la carpeta del speaker02.

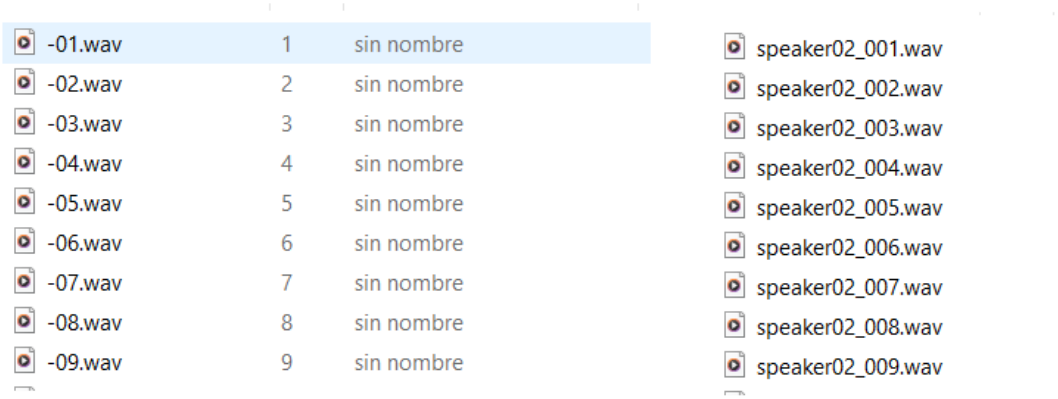


Figura 34. Audios antes y después de renombrarlos

Preprocesamiento de los Audios

Luego de haberle dado orden y formato a los audios generados se utilizó el siguiente código[11], donde se realizaron los siguientes pasos:

- Se subieron los archivos en formato wav en un archivo de extensión rar bajo el nombre de data, para luego ser ocupados dentro del proyecto.

[11] Código preprocesamiento

En la Figura 35 se cargaron los archivos con el nombre data.rar los cuales se encuentran compresos por el peso de los archivos.

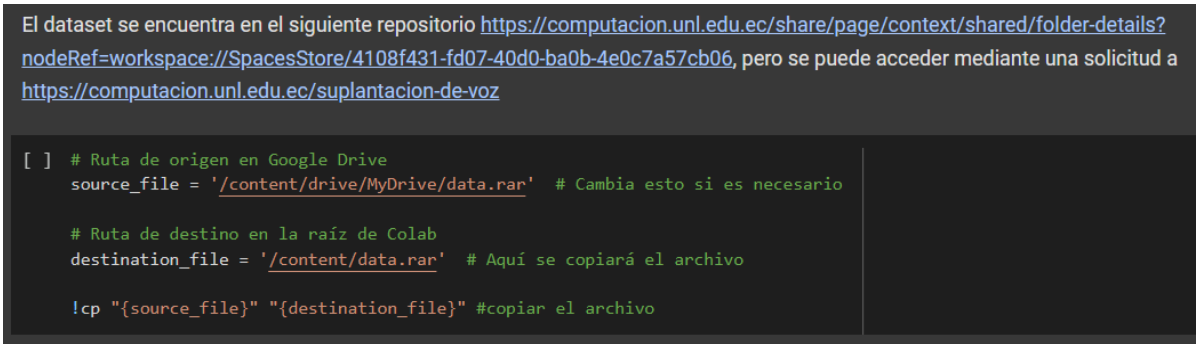


Figura 35. Carga de los archivos desde el drive

- Luego se exportó el código a ser utilizado dentro del proyecto. En la Figura 36 se clonó el repositorio del cual se va a ocupar el hcódigo para realizar el proyecto,

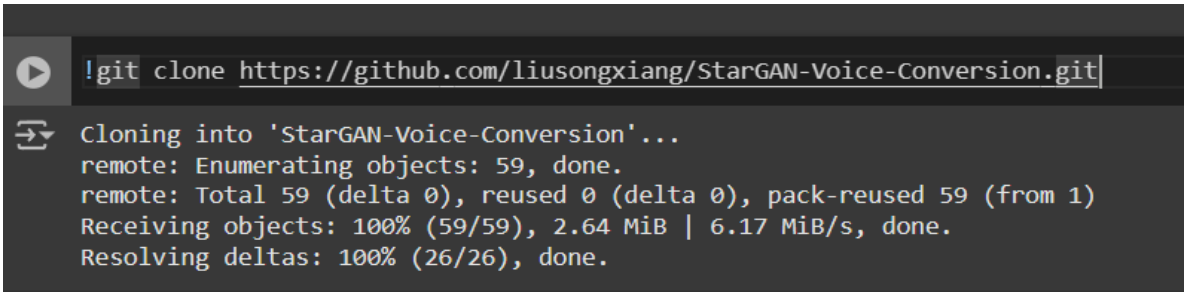


Figura 36. Clonar el repositorio del git a usar.

- Una vez hecho la importación del código fuente, se instalaron las librerías necesarias a ser utilizadas en el proyecto. En la Figura 37 se realizó la instalación de las librerías necesarias para la ejecución del proyecto clonado, las librerías fueron:
    - **Pyworld**: Es una librería para trabajar con señales de voz que se basa en el sistema WORLD, una tecnología que es utilizada en el campo del procesamiento de voz. WORLD permite analizar y sintetizar la voz.[12]
    - **TensorboardX:** Es una librería que se utiliza para visualización la cual se integra con varios frameworks de aprendizaje automático. TensorBoard puede ser usada en TensorFlow, tensorboardX y en PyTorch.[13]

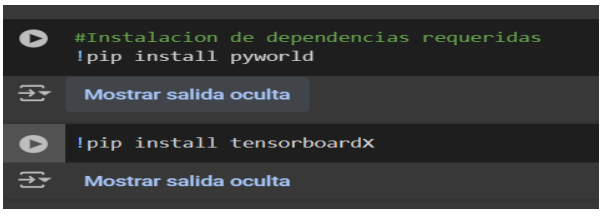


Figura 37. Importación de las dependencias del proyecto.

[12] Documentación librería pyworld pyworld · PyPI
[13] Documentación librería tensorboardX tensorboardX · PyPI

- Luego se procedió a realizar el preprocesamiento de los datos donde se ejecutó el archivo **preprocess.py.** En la imagen Figura 38 se indicó los parámetros con los que se debe ejecutar el proyecto para que se ejecute con el dataset generado, donde se tuvo hubo:
    - **– sample_rate;** Aquí se definio la muestre objetivo de los audios, para lo cúal se definio 16000 kHz.
    - **--origin_wavpath ../data/wav48:** Ruta de origen que contiene los archivos de audio originales.
    - **--target_wavpath ../data/wav16:** Ruta de destino donde se almacenaron los archivos re-muestreados
    - **--mc_dir_train ../data/mc/train:** Directorio donde se guardó las características acústicas del conjunto de entrenamiento.
    - **--mc_dir_test ../data/mc/test:** Directorio donde se guardó las características acústicas del conjunto de prueba

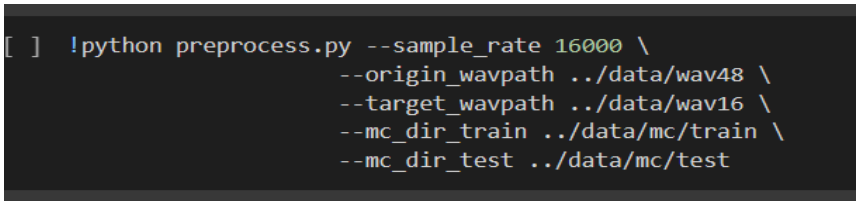


Figura 38. Ejecución del archivo preprocess.py

- En la Figura 39 se presentó los pasos que realiza el archivo proprocess.py los cuales son 4, estos pasos abarcaron desde el re-sampleo de los audios hasta el almacenamiento de las características. Por lo que se realizaron pasos como:
    - **Re-sampleo de los audios:** Los archivos cargados estaban ingresados en un formato de 48 kHz, por lo que se debieron cambiar su frecuencia a 16kHz.
    - **División de datos:** Una vez se cambió la frecuencia de los audios se dividieron los audios en dos conjuntos el 90% para entrenamiento y el 10% para pruebas todo esto de manera aleatorio.
    - **Extracción de características acústicas:** Una vez hecho la división del dataset se extraen dos características importantes como:
        - **F0:** Tono de la voz de las personas.
        - **MCEPs:** Timbre de la voz de las personas.
    - **Almacenamiento de características:** Una vez se obtuvo las características importantes se van a almacenar en un archivo con formato **npy,** ya que almacena archivos en formato binario y no ocupa mucha memoria en disco.

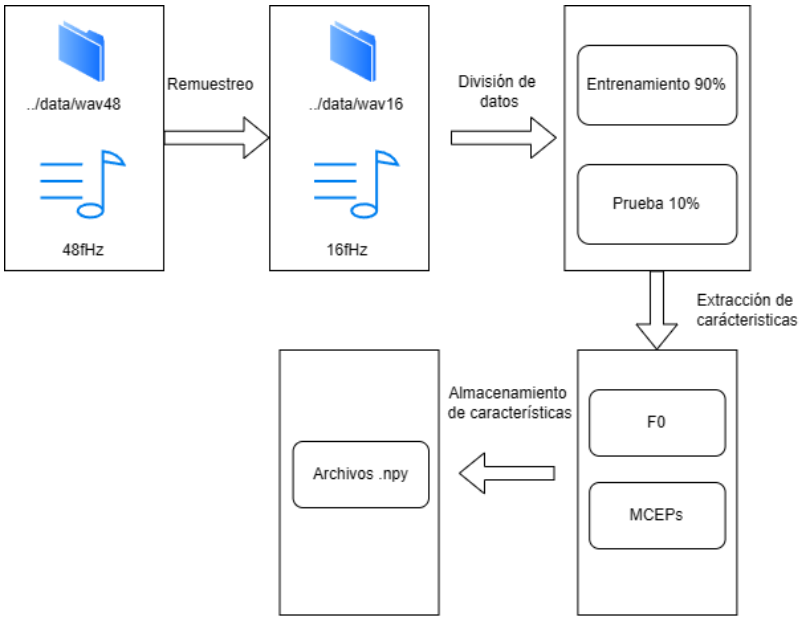


Figura 39. Flujo del Preprocesamiento de los Audios

Entrenamiento del Modelo StarGAN

Una vez se realizó el Preprocesamiento de los Audios, vino un paso importante dentro de la conversión de audios el cual fue el entrenamiento del modelo. En la Figura 41 se presentó las distintas fases del entrenamiento del modelo que van desde los hiperparametros hasta el almacenamiento de los modelos, en este proceso se realizaron los siguientes pasos:

- **Configuración de los hiperparametros:** Se mantuvieron los mismo hiperparametros del modelo original ya que no se pretendió hacer un ajuste fino, en la Tabla 9 se detallan los valores configurados, con una breve descripción de la función de cada uno de estos parámetros dentro del proceso de entrenamiento del modelo.

Tabla 9. Parámetros de Configuración para el Entrenamiento del Modelo StarGAN

Parametros	Valores	Descripción
Número de Hablantes	15	Hace referencia a la muestra total de las 15 personas de la muestra del dataset.
Número de iteraciones	200000	Total, de iteraciones para el entrenamiento del modelo.
Tasa de aprendizaje del Generador (G)	0.0001	Velocidad con la que el modelo ajusta los pesos en el generador.
Tasa de aprendizaje del Discriminador (D)	0.0001	Velocidad con la que el modelo ajusta los pesos en el discriminador.
Tamaño del batch	32	Número de muestras utilizadas en cada iteración de entrenamiento.

Lambda clasificación (λ_cls)	10	Ayuda al modelo a identificar correctamente los orígenes de los datos.
Lambda reconstrucción (λ_rec)	10	Ayuda a preservar las características esenciales de los datos.
Lambda penalización por gradiente (λ_gp)	10	Controla la estabilidad del discriminador para evitar que aprenda demasiado rápido o se vuelva demasiado sensible.

- **Ejecución del archivo main.py:** En la Figura 40 se ejecutó el código encargado de iniciar con el entrenamiento del modelo, esto lo hará con el dataset personalizado que se le cargo anteriormente.

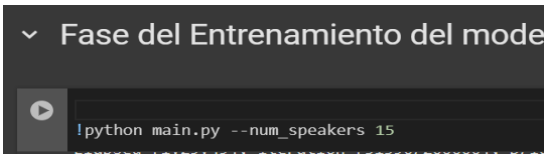


Figura 40. Ejecución del archivo main.py

- **Progreso de las iteraciones:** En la Tabla 10 se presentó el registro del progreso del entrenamiento del modelo StarGAN donde se mostraron algunas de las iteraciones del modelo, la pérdida del generador y discriminador, las métricas se entienden como:
    - **D/loss_real:** Valor encargado por parte del discriminador al clasificar correcamente las muestras reales, mientras el valor sea lo más bajo significa que está mejorando esa tarea.
    - **D/loss_fake:** Valor encargado de la parte del discriminador al identificar las muestras generadas como falsas, si el valor es lo más bajo posible significa que está aprendiendo mejor.
    - **G/loss_fake:** Valor encargado de que el generador pueda producir datos que engañen al discriminador, mientras el valor es lo más alto significa que el generador está aprendiendo.
    - **G/loss_rec:** Valor encargado de que el generador reconstruya características importantes de los audios originales, mientras en valor sea lo más bajo significa que está aprendiendo.

Tabla 10. Progreso de las Pérdidas Durante el Entrenamiento

Iteración	D/loss_real	D/loss_fake	G/loss_fake	G/loss_rec
60000/200000	-18.0252	0.1512	-1.5864	0.4757
61000/200000	-12.5936	-1.8613	4.4743	0.4628
62000/200000	-14.6281	-0.7178	-0.0442	0.4476

.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
100000/200000	-4.8214	-13.1122	12.9081	0.4132
110000/200000	-9.8889	-15.0232	23.7985	0.4204
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
150000/200000	-14.6982	6.0021	-4.9810	0.3979
151000/200000	-9.5909	-15.1443	15.4492	0.4089
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
200000/200000	-3.3437	-20.3509	21.5803	0.3774

- **Evaluación de las iteraciones:** Una vez identificadas las métricas de la Tabla 10, se explicó los valores de las iteraciones donde:
    - **Iteración 60000/200000:** Aquí en la iteración se vio que el **D/loss_real** con un valor de -18.0252 y D/loss_fake cercano a 0.1512 reflejan que el discriminador todavía enfrenta dificultades tanto para clasificar correctamente las muestras reales como para identificar las falsas, pero el generador mostró un **G/loss_fake** bajo -1.5864, lo que indica que no logra producir datos que engañen eficazmente al discriminador, asi como **G/loss_rec** en 0.4757 refleja un desempeño moderado en la reconstrucción de los audios. Por lo que esta iteración no es óptima para usarse.
    - **Iteración 61000/200000:** Aquí en la iteración se vio que el **D/loss_real** con un valor de -12.5936 y **D/loss_fake** de -1.8613 muestran que el discriminador sigue teniendo dificultades para clasificar correctamente las muestras reales y para identificar las falsas,pero el generador **G/loss_fake** con valor de 4.4743 indica que está comenzando a generar muestras que engañan mejor al discriminador, pero el valor de **G/loss_rec** 0.4628 sigue siendo moderado, lo que sugiere que el generador aún tiene margen de mejora en la reconstrucción de los audios. Aunque hay una ligera mejora, esta iteración aún no es óptima.
    - **Iteración 62000/200000:** Aquí en la iteración se vio que el **D/loss_real** con un valor de -14.6281 y **D/loss_fake** de -0.7178 reflejan que el discriminador sigue teniendo dificultades para identificar las muestras reales, pero está mejorando en identificar las falsas, pero el generador **G/loss_fake** con valor de -0.0442 muestra que el generador no está produciendo muestras de alta calidad que engañen al discriminador, también el **G/loss_rec** de 0.4476, la reconstrucción

de los audios sigue estando en un nivel aceptable, pero aún hay margen para mejorar. Esta iteración muestra avances, pero aún no es óptima.

- **Iteración 100000/200000:** Aquí en la iteración se vio que el **D/loss_real** cpm valor de -4.8214 y **D/loss_fake** con valor de **-13.1122** indican que el discriminador está mejorando su capacidad para identificar muestras falsas, aunque aún tiene dificultades con las muestras reales, también el generador **G/loss_fake** con valor de **12.9081** muestra que está mejorando en la generación de datos que engañan al discriminador, pero el **G/loss_rec** con valor de **0.4132** sigue siendo moderado. Esta iteración muestra una mejora en el rendimiento, pero aún le falta.
- **Iteración 110000/200000:** Los valores de **D/loss_real** de -9.8899 y **D/loss_fake** de -15.0232 muestran una mejora continua en el discriminador al identificar las muestras falsas, pero aún no ha alcanzado un buen rendimiento con las muestras reales. También el valor de **G/loss_fake** 23.7985 indica que el generador ha mejorado significativamente en su capacidad para engañar al discriminador, el **G/loss_rec** de valor de 0.4204 sigue siendo moderado.
- **Iteración 150000/200000:** Los valores de **D/loss_real** de -14.6982 y **D/loss_fake** de 6.0021 reflejan un buen desempeño del discriminador para identificar las muestras falsas, pero aún tiene dificultades con las reales. El valor de **G/loss_fake** -4.9810 muestra que el generador sigue teniendo dificultades para generar muestras de alta calidad que engañen al discriminador, mientras que **G/loss_rec** de 0.3979 muestra una mejora en la reconstrucción de los audios. Esta iteración no es aún óptima.
- **Iteración 151000/200000:** Los valores de **D/loss_real** -9.5909 y **D/loss_fake** -15.1443 indican que el discriminador sigue mejorando en su tarea de identificar muestras falsas, pero aún tiene espacio para mejorar con las muestras reales, el valor de **G/loss_fake** 15.4492 refleja que el generador ha avanzado en la capacidad para engañar al discriminador, aunque no está perfecto, el valor de **G/loss_rec** 0.4089 muestra que el generador sigue reconstruyendo las características de los audios pero le falta para mejorar.
- **Iteración 200000/200000:** En la iteración final los valores de **D/loss_real** -3.3437 y **D/loss_fake** -20.3509 muestran un gran desempeño del discriminador al identificar las muestras falsas y reales por lo que se encontró en un funcionando óptimo. El valor de **G/loss_fake** 21.5803 muestra que el generador está produciendo muestras de alta calidad que engañan eficazmente al discriminador. Mientras que el parámetro **G/loss_rec** con valor

de 0.3774 indica que la reconstrucción de los audios es satisfactoria, por lo que esta iteración mostró un rendimiento óptimo.

- **Almacenamiento de los modelos:** Una vez se completó con las 200,000 iteraciones, los modelos entrenados fueron almacenados en una carpeta de nombre **models**. El proceso de entrenamiento tuvo una duración aproximada de 5 horas y 30 minutos con el uso de una GPU NVIDIA L4 proporcionada por Google Colab en su versión de paga.

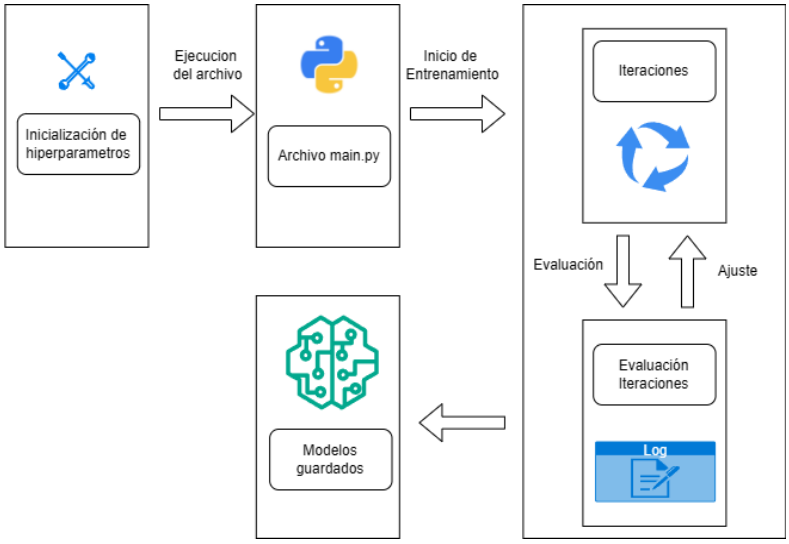


Figura 41. Flujo del Proceso de Entrenamiento

Conversión de Voz entre Hablantes

Una vez se realizó el Entrenamiento del Modelo StarGAN, se procedió con la etapa de conversión de voz, por lo que se realizaron los siguientes pasos:

- **Configuración de parámetros para la conversión de los audios**: Al momento de generar las voces se realizaron configuraciones para adaptar al dataset que se generó. En la Tabla 11 se presentó las distintas configuraciones de los parámetros que van desde la especificación del total de los hablantes hasta la definición del directorio donde se encuentran los audios fuente y los resultados de las conversiones.

Tabla 11. Configuración de parámetros para la generación de audios.

Parametros	Valores	Descripción
--num_speakers	15	Hace referencia a la muestra total de las 15 personas de la muestra del dataset.
--resume_iters	200000	Iteración del modelo que se usó para cargar los pesos entrenados y realizar la conversión.

--src_spk	speaker01 a speaker15	Hablante fuente cuya voz fue transformada en cada iteración.
--trg_spk	speaker01 a speaker15	Hablante objetivo al cual se transformó la voz del hablante fuente.
--wav_dir	./data/wav16	Directorio que contiene los audios originales utilizados en el proceso de conversión.

- **Ejecución del archivo convert.py:** En la Figura 42 se ejecutó el código encargado de iniciar con la conversión de los audios, esto lo hará con el dataset y el modelo generador con anterioridad.

```
#Conversion del Speaker01 a los distintos speakers
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker01 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker02 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker03 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker04 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker05 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker06 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker07 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker08 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker09 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker10 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker11 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker12 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker13 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker14 --wav_dir ./data/wav16
!python convert.py --num_speakers 15 --resume_iters 200000 --src_spk speaker01 --trg_spk speaker15 --wav_dir ./data/wav16
```

Figura 42. Ejecución del archivo convert.py

- **Resultados de la Conversión:** En la Tabla 12 se indicó las conversiones de la voz de 15 hablantes fuente hacia los 15 hablantes objetivo, generando un total de 225 combinaciones de audios transformados, dando un total de 1800 audios que tomo alrededor de 3 horas.

Tabla 12. Combinaciones realizadas para la generación de Audios.

Speaker Fuente	Speaker Objetivo	Numero de Conversiones	Ejemplo audio generado
speaker01	speaker01	8	speaker01_004-vcto-speaker01.wav
speaker01	speaker02	8	speaker01_024-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker02	speaker01	8	speaker02_024-vcto-speaker01.wav

Speaker Fuente	Speaker Objetivo	Numero de Conversiones	Ejemplo audio generado
speaker02	speaker02	8	speaker02_024-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker03	speaker01	8	speaker03_007-vcto-speaker01.wav
speaker03	speaker02	8	speaker03_007-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker04	speaker01	8	speaker04_002-vcto-speaker01.wav
speaker04	speaker02	8	speaker04_002-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker05	speaker01	8	speaker05_004-vcto-speaker01.wav
speaker05	speaker02	8	speaker05_004-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker06	speaker01	8	speaker06_004-vcto-speaker01.wav
speaker06	speaker02	8	speaker06_004-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker07	speaker01	8	speaker07_004-vcto-speaker01.wav
speaker07	speaker02	8	speaker07_004-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker08	speaker01	8	speaker08-speaker01

Speaker Fuente	Speaker Objetivo	Numero de Conversiones	Ejemplo audio generado
speaker08	speaker02	8	speaker08_003-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker09	speaker01	8	speaker09_010-vcto-speaker01.wav
speaker09	speaker02	8	speaker09_010-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker10	speaker01	8	speaker10_011-vcto-speaker01.wav
speaker10	speaker02	8	speaker10_011-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker11	speaker01	8	speaker11_019-vcto-speaker01.wav
speaker11	speaker02	8	speaker11_030-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker12	speaker01	8	speaker12_033-vcto-speaker01.wav
speaker12	speaker02	8	speaker12_017-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker13	speaker01	8	speaker13_004-vcto-speaker01.wav
speaker13	speaker02	8	speaker13_004-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	

Speaker Fuente	Speaker Objetivo	Numero de Conversiones	Ejemplo audio generado
speaker14	speaker01	8	speaker14_002-vcto-speaker01.wav
speaker14	speaker02	8	speaker14_002-vcto-speaker02.wav
.	.	.	
.	.	.	
.	.	.	
speaker15	speaker14	8	speaker15_026-vcto-speaker14.wav
speaker15	speaker15	8	speaker15_024-vcto-speaker15.wav
Total		1800	

- **Exportación y ordenamiento de audios:** Una vez se realizó las conversiones entre speakers, se procedió a exportarlos:
    - **Exportación de audios de Google Colab:** Los audios se generaron en una carpeta llamada convert, la cual dentro de la misma especificaba el nombre del modelo que se utilizó en este caso el 200000. En la Figura 43 se usó el comando **!zip -r "converted.zip" "./converted/"**, se comprimió la carpeta con los audios lo que facilito para su posterior descarga.

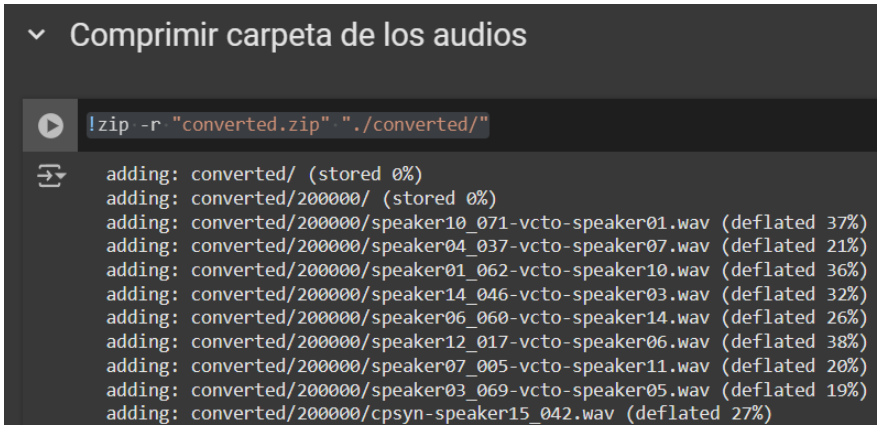


Figura 43. Comprimir los archivos de audios en una carpeta de formato zip.

    - **Ordenamiento y organización de los audios:** Una vez con la carpeta descargada, los archivos se encontraron sin ningún orden en específico, por lo que se tuvo que ordenarlos, por lo que se tuvo en cuenta un criterio jerárquico para hacerlo.

      En Tabla 13 se muestra la distribución inicial de los directorios principales, en este caso se crearon 15 speakers de acuerdo a los 15 participantes, luego se

creó subcarpetas donde se encontraban cada uno de las combinaciones realizadas entre personas.

En la Figura 44 se muestra en cambio los subdirectorios que contiene cada subcarpeta con cuantos archivos tiene en la misma, como se observa cada uno de estos subdirectorios tiene 8 archivos.

Tabla 13. Carpeta Principal.

Speaker	Subdirectorios
speaker01	• speaker01-speaker01 • speaker01-speaker02 • speaker01-speaker03 • ... • speaker01-speaker15
speaker02	• speaker01-speaker01 • speaker01-speaker02 • speaker01-speaker03 • ... • speaker01-speaker15
.	.
.	.
.	.
speaker14	• speaker01-speaker01 • speaker01-speaker02 • speaker01-speaker03 • ... • speaker01-speaker15
speaker15	• speaker01-speaker01 • speaker01-speaker02 • speaker01-speaker03 • ... • speaker01-speaker15

Tabla 14. Sub-Directorios por Speaker

Conversiones	Archivos
speaker01-speaker01	8
speaker01-speaker02	8
speaker01-speaker02	
.	.
.	.
.	.
speaker02-speaker01	8
speaker02-speaker02	8
speaker02-speaker02	8
.	.
.	.

.	.
Speaker14-speaker01	8
Speaker14-speaker02	8
Speaker14-speaker02	8
.	.
.	.
.	.
Speaker15-speaker13	8
Speaker15-speaker014	8
Speaker15-speaker015	8

En la Figura 44 se presentó el script[14] utilizado para realizar las tareas de estructuración de los audios, el script recorre toda la carpeta que contiene todos los audios exportados sin ningún orden, luego de eso crea una estructura jerárquica donde cada carpeta principal corresponde a un speaker, dentro de cada carpeta se crea subcarpetas con las distintas combinaciones de cada conversión y finalmente dentro de las carpetas convertidas se mueven todos los archivos según el orden de las transformaciones.

```python
def organize_wav_files_with_global_folders(directory):
    # Cambiar al directorio especificado
    os.chdir(directory)

    # Obtener la lista de archivos .wav
    wav_files = [file for file in os.listdir() if file.endswith('.wav')]

    for file in wav_files:
        # Dividir el nombre del archivo para extraer los identificadores
        parts = file.split('-')
        if len(parts) >= 3:
            # Extraer speakerXX y speakerYY
            speaker_from = parts[0].split('_')[0]
            speaker_to = parts[2].split('.')[0]

            # Crear el nombre de la carpeta específica
            specific_folder_name = f"{speaker_from}/{speaker_from}-{speaker_to}"

            # Crear la carpeta global para el speaker_from si no existe
            if not os.path.exists(speaker_from):
                os.makedirs(speaker_from)

            # Crear la subcarpeta específica si no existe
            if not os.path.exists(specific_folder_name):
                os.makedirs(specific_folder_name)

            # Mover el archivo a la carpeta correspondiente
            shutil.move(file, os.path.join(specific_folder_name, file))
```

Figura 44. Script en Python para la organización y estandarización de archivos de audio.

- **Organizar Final de Dataset:** En la Figura 45, se presentó la estructura del dataset organizado luego de completar la tarea de la organización, aquí el dataset consta de 4 carpetas, la carpeta de **Transcripciones** contiene todos los guiones utilizados durante las grabaciones bajo el nombre de cada uno de

[14] Código usado para estructurar carpeta audios Script

los speakers en formato txt, tanto la carpeta **Voces Reales 48kHz** y **Voces Reales 16kHz** almacenan las grabaciones originales de los hablantes organizadas en subcarpetas individuales por participante pero la carpeta de **48kHz** contiene la frecuencia en 48 hercios y la carpeta **16kHz** contiene la frecuencia en 16 hercios. Finalmente, la carpeta **Voces Convertidas Stargan** contiene todos los audios generados por la Inteligencia Artificial Generativa con sus respectivas subcarpetas con sus conversiones.

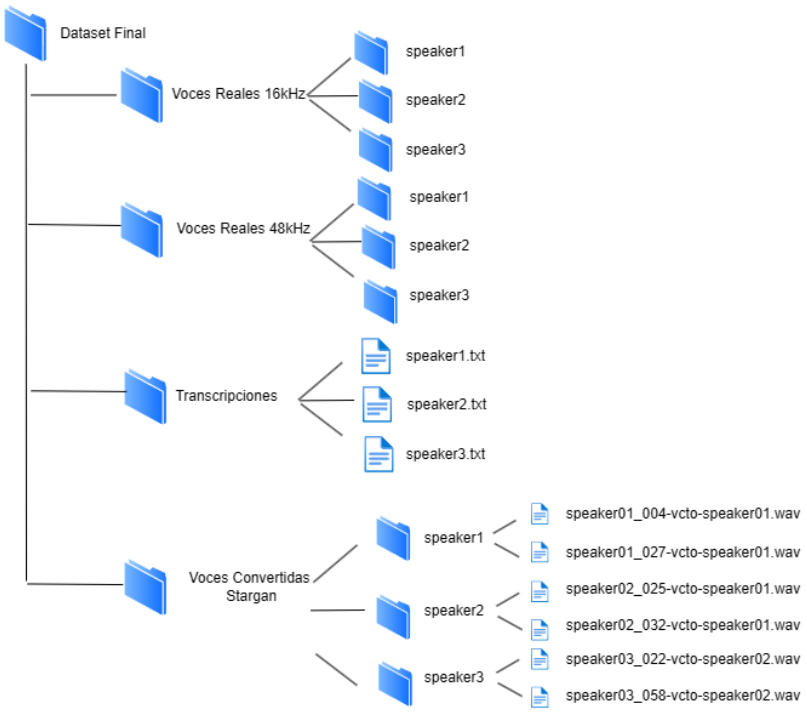


Figura 45. Estructura del Dataset Organizado

El acceso al dataset limpio se encuentra de alojado el repositorio del servidor de Alfresco de la Carrera de Computación de la UNL, donde para hacerlo se tomó en cuenta los siguientes pasos:

Acceso a Alfresco

- Se tuvo que acceder a la página https://computacion.unl.edu.ec/share/page/, donde es se encuentra el programa de Alfresco de la Carrera de Computación de la UNL.
- Luego se ingresó al servidor con el usuario y contraseña otorgadas.

Carga de Archivos

- Una vez dentro de Alfresco se va al apartado de **Mis ficheros,** luego se cargaron los archivos desde el ordenador hacia el servidor. En la Figura 46 se muestra en la parte izquierda la imagen el dataset cargado dentro del servidor en la aplicación alfresco y en la parte derecha como se encuentra estructurado la carpeta dentro de servidor.

Figura 46. Estructura y archivos dentro del servidor

Creación de un sitio informativa para presentar el dataset realizado

- Una vez realizada la carga de archivos dentro del servidor en la Figura 47 se realizó un sitio web informativo en la página web de la Carrera de Computación de la UNL https://computacion.unl.edu.ec/suplantacion-de-voz, donde se detalló el proceso de construcción, características, como descargar y el contacto con los autores del dataset.

Figura 47. Página web informativa sobre el dataset.

6.1.4. División de datos

Preparación de los Datos

En esta etapa se utilizó el codigo[15] de la sección “Código para dividir los datos aleatoriamente”. En la Figura 48 se indicó la carga del dataset y la impresión de los archivos que contiene la carpeta seleccionada.

[15] Código utilizado para dividir los datos Script

```python
import os
import random
from shutil import copy2

# Ruta donde están los audios
ruta_audios = "./Carpeta con los audios originales/"

# Listar todos los archivos en la carpeta
audios = [archivo for archivo in os.listdir(ruta_audios) if archivo.endswith(".wav")]
```

Figura 48. Carga y muestra del dataset

Aleatorización de los Datos

En la Figura 49 se muestra el fragmento de código utilizado para barajar los archivos del dataset de manera aleatoria, con esto se buscó que los datos no sigan un patrón específico al momento de dividirlos, para ello con la ayuda de la función **(`random.seed(42)`)** siempre se va a reproducir al división de datos de la misma forma, y la función **`random.shuffle(audios)`** para reordenar los archivos de forma aleatoria dentro de la lista.

```python
# Barajar los archivos para asegurarnos de que la división sea aleatoria
random.seed(42)  # Fija una semilla para reproducibilidad
random.shuffle(audios)
```

Figura 49. Código para Aleatorización de los Archivos del Dataset

Separación de Datos

En la Figura 50 se presentó el código empleado para dividir el dataset en dos conjuntos: 80% para entrenamiento y 20% para prueba. Los datos resultantes se almacenaron en las variables **train_audios** y **test_audios**, correspondientes a los conjuntos de entrenamiento y prueba,

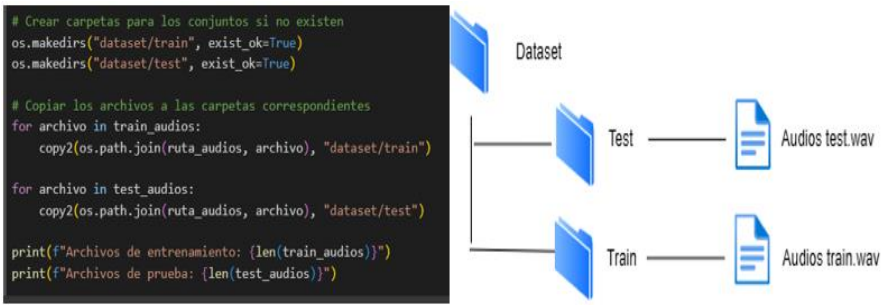
```python
# Definir proporción de train y test
proporcion_train = 0.8  # 80% entrenamiento, 20% prueba
num_train = int(len(audios) * proporcion_train)

# Dividir en train y test
train_audios = audios[:num_train]
test_audios = audios[num_train:]
```

Figura 50. División de datos en entrenamiento y pruebas

Organización en Carpetas

En la Figura 51 se presentó el código utilizar para organizar los datos en carpetas de entrenamiento y prueba, por lo que se crearon las carpetas correspondientes (train y test) dentro de un directorio principal denominado dataset.

En la parte derecha se encuentra el esquema visual con la jerarquía de las carpetas y cómo los archivos de audio fueron distribuidos en cada una de ellas.




```
# Crear carpetas para los conjuntos si no existen
os.makedirs("dataset/train", exist_ok=True)
os.makedirs("dataset/test", exist_ok=True)

# Copiar los archivos a las carpetas correspondientes
for archivo in train_audios:
    copy2(os.path.join(ruta_audios, archivo), "dataset/train")

for archivo in test_audios:
    copy2(os.path.join(ruta_audios, archivo), "dataset/test")

print(f"Archivos de entrenamiento: {len(train_audios)}")
print(f"Archivos de prueba: {len(test_audios)}")
```

Figura 51. Código y estructura de carpetas organizada de los datos de entrenamiento y prueba.

El dataset dividido se encuentra disponible en el siguiente repositorio de Google Drive [16] y para acceder al repositorio se debe realizar los pasos de la página https://computacion.unl.edu.ec/suplantacion-de-voz.

6.2. Objetivo 2: Utilizar el modelo LFCC-LCNN con el dataset generado para medir la métrica Tasa de Error de Igualación en la muestra de 15 personas de la carrera de Computación de la Universidad Nacional de Loja.

Para cumplir con el segundo objetivo planteado se adaptó mediante la adaptación de la cuarta fase de la metodología CRISP-ML(Q) donde se utilizó las fases Evaluación del Modelo, en este apartado se detallaron las tareas realizadas en cada fase.

6.2.1. Evaluación del Modelo

Preparación y ajuste del dataset

En esta etapa se tomó en consideración una serie de pasos tales como:

- **Estructuración del dataset:** En la Figura 52 se muestra la organización de las carpetas donde, la carpeta **scp** que contiene los labels de los archivos de la carpeta entrenamiento, la carpeta **eval** contiene todos los audios de la evaluación antes separados, la carpeta **train_dev** contiene los audios del entrenamiento antes separados, y el archivo **protocol.txt** es importante para que se pueda adaptar al código que se utilizó ya que corresponde a la etiquetas de los archivos del dataset, además el archivo cuenta con un formato que es **<origen> <nombre_archivo> <clasificación>** donde:
  - **<origen>:** Corresponde al participante del dataset que va desde speaker01 hasta speaker15.
  - **<nombre_archivo>:** Corresponde al nombre del archivo del audio dentro del dataset cargado.

[16] Dataset Dividido https://computacion.unl.edu.ec/share/page/folder-details?nodeRef=workspace://SpacesStore/2975e8b2-971f-41a1-95a7-be1d9646e0b7

- **<clasificación>:** Finalmente para la clasificación existen dos clases una es bonafide que identifica si el audio es real y la etiqueta spoof que identifica si el audio ha sido creado con StarGAN, un ejemplo de la etiqueta completa es **speaker01 speaker01_004-vcto-speaker15 - StarGAN spoof** para los archivos con StarGAN y speaker01 speaker01_004 - - bonafide para los archivos normales.

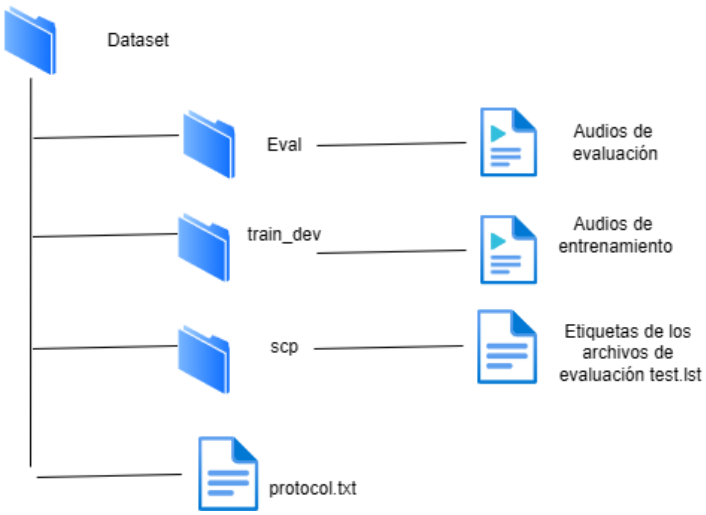


Figura 52. Estructura y organización del dataset la evaluación de audios.

Una vez se identificó la estructura para generar el archivo protocol.txt, mediante el código[17], se pudo automatizar el proceso como en la imagen Figura 53 donde se indicó que en base a los archivos test y train generados en la división de datos se generó el archivo protocol.txt con las indicaciones previas.

```
# Directorios de los datos
train_dir = "./train"  # Cambiar según tu estructura
test_dir = "./test"    # Cambiar según tu estructura
output_file = "protocol.txt"  # Archivo de salida

def generate_protocol(directories, output_file):
    with open(output_file, 'w') as f:
        for directory in directories:
            for filename in os.listdir(directory):
                if filename.endswith(".wav"):
                    # Extraer información del nombre del archivo
                    base_name = os.path.splitext(filename)[0]  # Sin extensión
                    parts = base_name.split('_')  # Separar por guiones bajos

                    # Determinar si es bonafide o spoof
                    if "-vcto-" in filename:
                        # Ejemplo: speaker01_050-vcto-speaker08.wav
                        primary_id = parts[0]  # speakerXX
                        secondary_id = f"StarGAN-{base_name}"  # Nombre completo con prefijo
                        technique = "StarGAN"
                        label = "spoof"
                        f.write(f"{primary_id} {secondary_id} - {technique} {label}\n")
                    else:
                        # Ejemplo: speaker01_054.wav
                        primary_id = parts[0]  # speakerXX
                        secondary_id = base_name  # Nombre completo sin prefijo
                        label = "bonafide"
                        f.write(f"{primary_id} {secondary_id} - - {label}\n")
```

Figura 53. Código para generar el archivo protocol.txt

[17] Código para generar el protocol.txt

El acceso al dataset de pruebas se encuentra de alojado en el servidor de Alfresco[18], de la Carrera de Computación de la UNL donde para hacerlo se tomó en cuenta los siguientes pasos:

Acceso a Alfresco

- Se tuvo que acceder a la página https://computacion.unl.edu.ec/share/page/, donde es se encuentra el programa de Alfresco de la Carrera de Computación de la UNL.
- Luego se ingresó al servidor con el usuario y contraseña otorgadas.

Carga de Archivos

- Una vez dentro de Alfresco se va al apartado de **Mis ficheros,** luego se cargaron los archivos desde el ordenador hacia el servidor. En la Figura 54 se muestra en la parte izquierda la imagen el dataset cargado dentro del servidor en la aplicación alfresco y en la parte derecha como se encuentra estructurado la carpeta dentro de servidor.

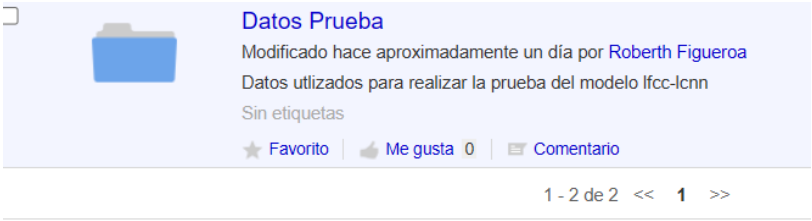


Figura 54. Código para generar el archivo protocol.txt

Carga del modelo y configuración del código

En esta etapa se tomó en consideración una serie de pasos tales como:

- **Preparación del entorno del trabajo:** Se creó un entorno dentro de colab donde se utilizó la gpu t4 de colab, para la ejecución de la práctica.
- **Carga del código:** Se utilizo el código base[19], para implementar el modelo y se realizaron configuraciones como eliminar archivos que no eran relevantes para el proyecto.

  En la Figura 55**,** se muestra el código[20] encargado de cargar el modelo preentrenado, el cual sirvió de ayuda para cumplir con el objetivo dos.

```
# arguments initialization
args = nii_arg_parse.f_args_parsed()

# Actualiza la ruta del modelo preentrenado
args.trained_model = './lfcc-lcnn-lstmsum-p2s/01/__pretrained/trained_network_finetune_lstm.pt'
```

Figura 55. Código encargado de cargar el modelo preentrenado.

[18] Repositorio del dataset https://computacion.unl.edu.ec/share/page/
[19] Repositorio del Código original
[20] Repositorio del Codigo adaptado

- **Configuración de archivos:** Al momento de realizar la prueba con el dataset creado, fue importante realizar las configuraciones en al archivo config.py que contiene las rutas de las variables que apuntan a los archivos encargados de las rutas de evaluación. En la Figura 56**,** se encuentra el código encargado de cargar la información para cargar los datos de prueba con los labels que tiene el dataset generado.

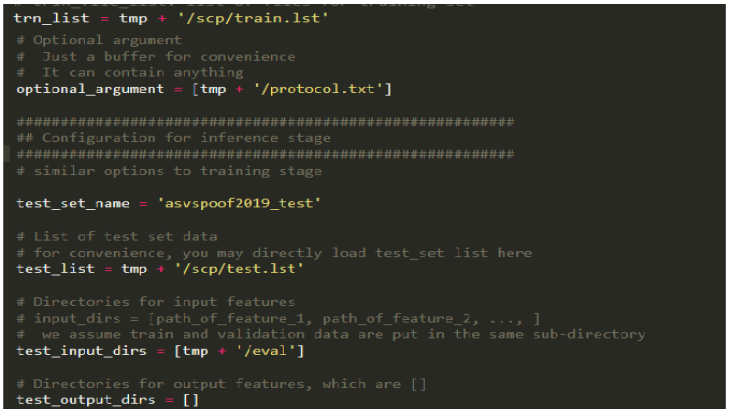


Figura 56. Configuraciones de rutas de las carpetas de evaluación

Generación de los logs de la evaluación

- **Ejecución del código de evaluación:** Luego de haber realizado la configuración de los archivos, así como la carga del modelo se ejecutó el archivo main.py que sirve para la evaluación del dataset cargado y arroja unos archivos logs que sirven para obtener la puntuación de la Tasa de Error de Igualación, en la Figura 57 se ejecutó el archivo encargado de obtener los logs para la obtención de la métrica.

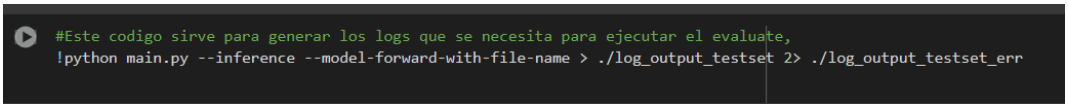


Figura 57. Obtención de los archivos logs mediante el archivo main.

Cálculo de la métrica

Durante esta etapa se llevaron a cabo los siguientes experimentos, descritos a continuación:

- **Primer experimento:** En el primer experimento lo que se realizo fue utilizar el dataset de test el cúal contiene 559 datos como se lo puede observar en la Figura 58 pero en la imagen se observó que se utilizó el dataset personal creado, mas no el dataset que utilizaron en el experimento original, ya que se buscaba probar el mismo y por ende se tomó en cuenta ese criterio, una vez realizado el experimento, se observó el resultado de los valores obtenidos de la evaluación donde de los 599 muestras 210 fueron clasificadas como muestras que no tienen alguna alteración en la voz y 349 fueron hechas con alguna red neuronal, también la tasa de error de igualación indica

un valor de 63.853% lo que significa que el modelo comete muchos errores para detectar un dataset con otros acentos pero aún tiene un porcentaje de asertividad lo que significa que no está del todo mal identificando el dataset elaborado con los datos que fue entrenado.

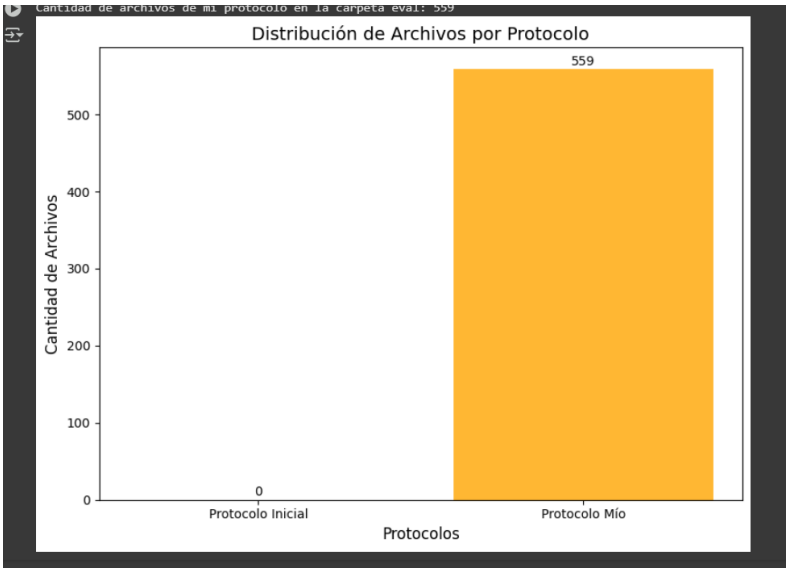


Figura 58. División de datos primer experimento.

- **Segundo experimento:** En el segundo experimento lo que se realizo fue utilizar el dataset que contiene 559 datos como se lo puede observar en la Figura 59 pero en la imagen se observó que se utilizó el dataset personal creado únicamente con los datos de train en vez del test, mas no el dataset que utilizaron en el experimento original, ya que se buscaba probar el mismo y por ende se tomó en cuenta ese criterio, una vez realizado el experimento**,** se observó el resultado de los valores obtenidos de la evaluación donde de los 1361 muestras 872 fueron clasificadas como muestras que no tienen alguna alteración en la voz y 1361 fueron hechas con alguna red neuronal, también la tasa de error de igualación indica un valor de 58.35% lo que significa que el modelo comete muchos errores para detectar un dataset con otros acentos pero aún tiene un porcentaje de asertividad, pero se puede mejorar.

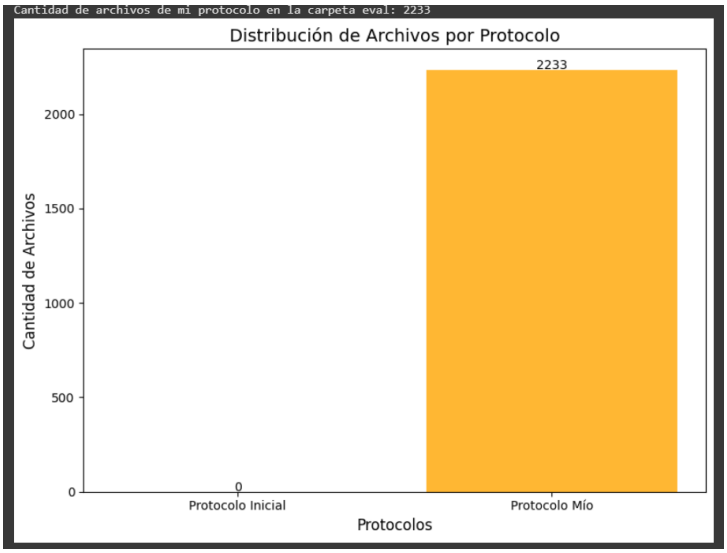


Figura 59. División de datos segundo experimento

- **Tercer experimento:** En el tercer experimento lo que se realizo fue utilizar el dataset que contiene 2792 datos como se lo puede observar en la Figura 60 pero en la imagen se observó que se utilizó el dataset personal creado ya con los datos de test y train previamente obtenidos , mas no el dataset que utilizaron en el experimento original, ya que se buscaba probar el mismo y por ende se tomó en cuenta ese criterio, una vez realizado el experimento se observó el resultado de los valores obtenidos de la evaluación donde de los 2792 muestras 1082 fueron clasificadas como muestras que no tienen alguna alteración en la voz y 1710 fueron hechas con alguna red neuronal, también la tasa de error de igualación indica un valor de 59.241% lo que significa que el modelo comete muchos errores para detectar un dataset con otros acentos y no mejoro a pesar de que se le envió todo el dataset generado completo.

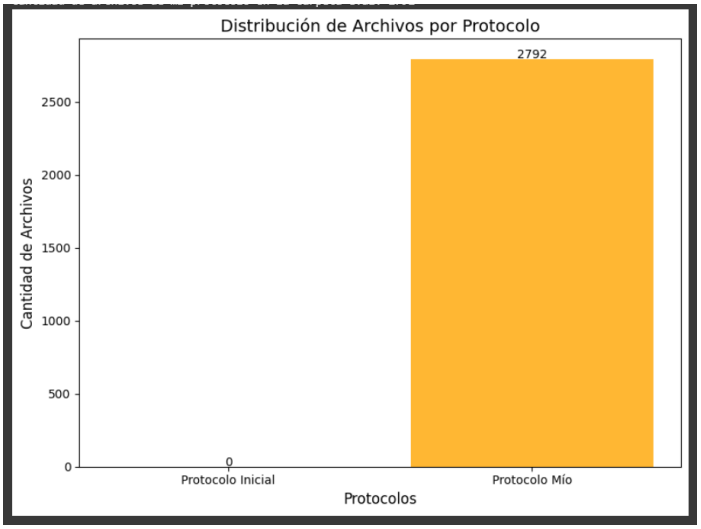


Figura 60. Distribución de datos en el tercer experimento según el protocolo utilizado.

- **Cuarto experimento:** En el cuarto experimento lo que se realizo fue utilizar el dataset que 3648 datos como se lo puede observar en la Figura 61 pero en la imagen se

observó que se utilizó el dataset personal creado con un poco de archivos del dataset original para realizar la experimentación, una vez realizado el experimento**,** se observó el resultado de los valores obtenidos de la evaluación donde de los 3648 muestras 1516 fueron clasificadas como muestras que no tienen alguna alteración en la voz y 2132 fueron hechas con alguna red neuronal, también la tasa de error de igualación indica un valor de 43.84 % lo que significa que el modelo mejoro respecto a los anteriores experimentos, pero tiene problemas bien con los datos generados.

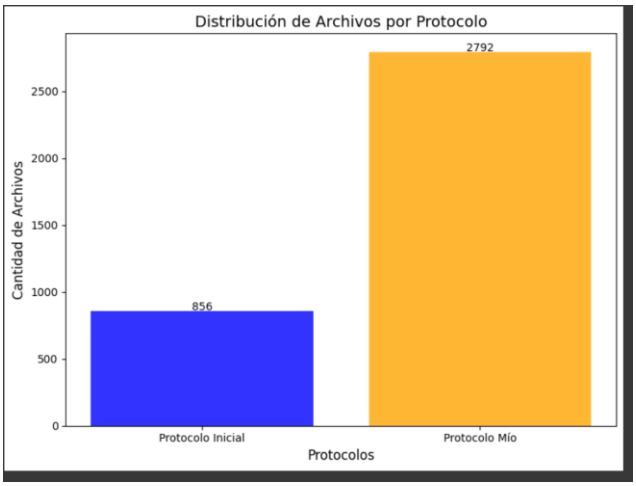


Figura 61. División de datos cuarto experimento

- **Quinto experimento:** En el quinto experimento lo que se realizo fue utilizar el dataset que 1507 datos como se lo puede observar en la Figura 62**,** pero en la imagen se observó que se utilizó el dataset personal creado con la mayoría de archivos del dataset original para realizar la experimentación, una vez realizado el experimento, se observó el resultado de los valores obtenidos de la evaluación donde la tasa de error de igualación indica un valor de 26.71 % lo que significa que el modelo identifico muchísimo mejor que los anteriores experimentos los datos evaluados ya que tiene en su mayoría datos del dataset original.

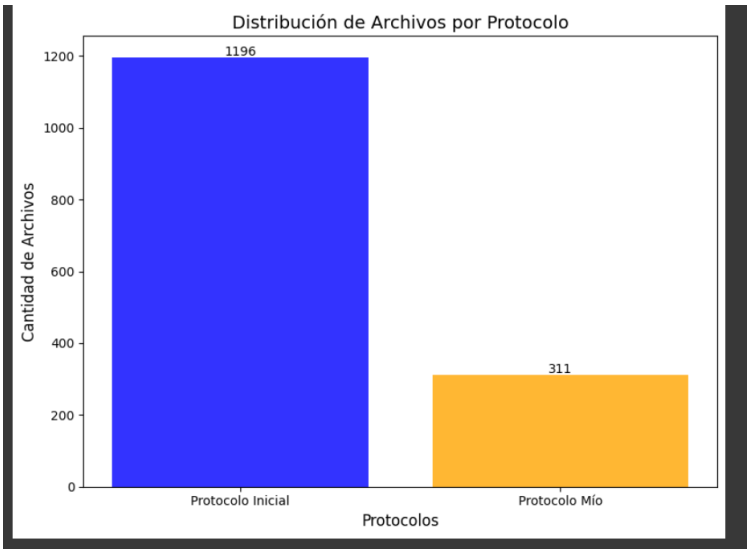


Figura 62. División de datos cuarto experimento

Análisis de los resultados obtenidos

Tabla 15. Resultados obtenidos.

Experimento	Datos Evaluados	Valor de la metricá
1	559	63.85
2	2233	58.35
3	2792	59.24
4	3648	43.84
5	1507	26.71

- En base a los resultados obtenidos de la Tabla 15**,** un modelo de audios entrenado con acentos en colombiano, chileno, peruano, venezolano y argentino no cumple con las características de un dataset con acento lojano de ecuador, ya que la fonética o la misma habla puede variar a pesar de que compartimos el mismo idioma que es el español.
- Los malos resultados obtenidos en la evaluación pueden ser debido a la construcción del dataset, ya que los datos originales fueron realizados por Google [21], donde mencionan que para la construcción del dataset lo hicieron en un estudio de grabación con equipos bastante robustos, frente al dataset que se realizó con equipos básicos y en ambientes donde no se tenía control sobre ruidos externos en su totalidad.
- Una de las consideraciones por las que el dataset realizado no pudo ser del todo bien, pudo ser debido al desbalanceo de clases que existían ya que se la muestra mas representativa fue de hombres y una pequeña muestra de mujeres, la muestra se la selecciono en base a un muestreo por conveniencia y aparte existieron problemas en

[21] Dataset Original

el país como la falta de luz eléctrica por lo que la disposición de las personas fue un poco difícil de tener,

- El dataset original también tomaba en cuenta solo una cantidad x de personas para decir que representa totalmente a un país, sin embargo, esta aproximación no es una representación completa de todas las variantes de acentos, fonéticas y estilos de habla presentes en un país, mientras que en dataset realizado se utilizó una muestra focalizada, pese a que no se obtuvo los mejores resultados igual el modelo la pudo evaluar.

7. Discusión

7.1 Primer objetivo:

La creación del dataset de audios con acento ecuatoriano fue parte importante del primer objetivo del presente trabajo de integración curricular, por lo que se basó en una metodología establecida en estudios previos y se aplicaron técnicas de procesamiento de audio y generación de datos sintéticos, dentro de esta sección se discuten los hallazgos más relevantes, la calidad del método utilizado y las limitaciones encontradas.

Los resultados obtenidos en la generación del dataset de audios fueron similares con estudios previos sobre recolección y generación de audio sintético mediante el uso de redes neuronales, la metodología elegida CRISP-ML(Q) [30] permitió una estructura organizada al momento de la recopilación, limpieza, procesamiento y generación de audios mediante la red neuronal StarGAN, este enfoque es similar a estudios [2] y [55], donde la selección de muestras fue realizada mediante un muestreo por conveniencia [56], [57], lo que fue de gran ayuda a la hora de seleccionar los participantes, pero esto puede traer consigo sesgos representativos en la muestra algo que se puede tener a consideración para futuras investigaciones. Además, el uso de modelos de lenguaje como ChatGPT para la generación de frases también ha sido reportado en estudios como[11], donde es importante utilizar frases con variabilidad fonética y expresiones coloquiales representativas del idioma objetivo. En comparación con otros datasets de una temática similar al voice spoofing, el presente trabajo logró obtener una diversidad de frases contextualizadas a un contexto ecuatoriano en especial a la provincia de Loja. Sin embargo, a diferencia de bases de datos más amplias como ASVspoof [13], el tamaño reducido de la muestra de 15 participantes representa una muestra bastante pequeña y no tan representativa, además que existió un desbalance de género en el dataset ya que en su mayoría fueron hombres frente a mujeres. Una vez se realizó la ejecución del modelo con 200,000 iteraciones, lo que permitió la conversión de las voces de los participantes en 1,800 audios de suplantación, al final de las iteraciones se observó un menor error en la pérdida del discriminador y generador, lo que indica que el modelo logró un aprendizaje efectivo, lo que significó que la calidad de la generación mejora significativamente con iteraciones prolongadas.

Los recursos computacionales fue una gran desventaja a la hora del entrenamiento del modelo StarGAN en Google Colab, por lo que se requirió una GPU pagada de colab para ejecutar el código, por lo que al usar una mayor capacidad de procesamiento se pudo ejecutar sin ningún problema, este fue un factor limitante que se puede tener a consideración en futuras investigaciones. Además, aunque se realizó la limpieza y normalización de datos, algunas grabaciones presentaron pequeñas distorsiones, por lo que se puede tener en consideración

usar equipos de una gama alta, así como usar espacios con entornos controlados para mejorar la calidad de los audios.

7.2 Segundo objetivo:

Para evaluar el desempeño del modelo LFCC-LCNN en la detección de voice spoofing con un dataset generado mediante StarGAN, se realizaron diversos experimentos utilizando una muestra de 15 personas de la Carrera de Computación de la Universidad Nacional de Loja, donde el principal objetivo fue determinar la Tasa de Error de Igualación en distintos escenarios y analizar la capacidad del modelo para generalizar en datos con acento lojano. En base a la pregunta de investigación: **¿Qué porcentaje de la métrica Tasa de Error de Igualación se obtendrá al evaluar un dataset de voice spoofing, creado mediante la técnica StarGAN con una muestra de 15 personas del cantón Loja, utilizando redes LCNN?**, se encontró que la métrica varió dependiendo de la configuración del dataset utilizado en la evaluación. En el primer experimento, donde se evaluó el dataset generado completamente con StarGAN, la Tasa de Error de Igualación alcanzó un 63.85%, indicando que el modelo presentó dificultades para distinguir audios reales y audios falsificados lo que es un alto margen de error puede explicarse por el hecho de que el modelo LFCC-LCNN fue entrenado originalmente con datos en español venezolano, chileno, peruano y argentino, mientras que el dataset evaluado en este estudio fue generado con audios de hablantes de Loja, así que la fonética y de acento entre los datos de entrenamiento y el dataset evaluado podría haber impactado en la evaluación de desempeño del modelo.

En el segundo experimento, se utilizó un dataset con más muestras de entrenamiento y menos muestras de prueba, lo que la metrica disminuyó a 58.35%, lo que indica que la cantidad de datos puede influir en la detección, en el tercer experimento, se utilizó el dataset completo previamente generado con StarGAN, sin mezclar datos originales y la metrica fue de 59.24%, mostrando que el modelo aún tenía problemas con el acento lojano, en el cuarto experimento, se comenzó a introducir una combinación de audios generados y audios originales del dataset base en menor proporción, lo que permitió reducir la tasa de error a 43.84%, finalmente, en el quinto experimento, donde la evaluación se realizó con una mayor proporción de datos originales en comparación con los generados, la metrica se redujo aún más a 26.71%, lo que da como resultado que el modelo logró mejorar su precisión a medida que se le proporcionaban datos más similares a los utilizados en su entrenamiento, reduciendo los errores de clasificación.

Los resultados del experimento tienden a depender en gran parte de las características del dataset con el que fueron entrenados, lo que explica la alta tasa de error en los primeros experimentos de este estudio, además del factor del acento, la calidad del dataset desempeñó un papel importante en los resultados obtenidos, mientras que los datasets utilizados en

investigaciones previas fueron grabados en entornos controlados y con equipos profesionales, el dataset evaluado en este estudio fue grabado en condiciones no controladas, utilizando micrófonos estándar en entornos universitarios y domésticos, lo que dio un calidad de grabación probablemente con ruido e inconsistencias en las señales de audio y por dichas razones el modelo no pudo haber tenido un buen comportamiento.

Otro aspecto relevante que pudo influir en los resultados es el desbalance en la distribución de género dentro del dataset, ya que la mayoría de los participantes en la muestra fueron hombres, lo que podría haber afectado la capacidad del modelo para reconocer variaciones en voces femeninas. En conclusión, la **Tasa de Error de Igualación obtenida en la evaluación del dataset generado mediante StarGAN con la muestra de 15 personas del cantón Loja varió entre 63.85% en el peor escenario y 26.71% en el mejor caso**, dependiendo de la composición del dataset y la cantidad de datos originales utilizados, lo demuestra la importancia de entrenar modelos con datos representativos de la población en la que se pretende aplicar la tecnología, mejorar la calidad del dataset y considerar el impacto del desbalance de género en la precisión del modelo.

8. Conclusiones

- La Tasa de Error de Igualación obtenida en la evaluación del dataset generado mediante StarGAN con la muestra de 15 personas del cantón Loja varió entre 63.85% en el peor escenario y 26.71% en el mejor caso, dependiendo de la composición del dataset y la cantidad de datos originales utilizados, el alto margen de error en los primeros experimentos no se debió a un fallo en la arquitectura del modelo LFCC-LCNN, sino a la diferencia entre los acentos del dataset de entrenamiento y el dataset evaluado, la calidad de las grabaciones y el desbalance en la muestra utilizada, los mismos que demuestran la importancia de entrenar modelos con datos representativos de la población en la que se pretende aplicar la tecnología, mejorar la calidad del dataset y considerar el impacto del desbalance de género en la precisión del modelo.
- La metodología CRISP-ML(Q) fue una herramienta importante dentro del desarrollo de este estudio, proporcionando una estructura clara y eficiente para abordar cada fase del proceso que fue desde la recolección y generación de datos, hasta la evaluación del modelo LFCC-LCNN, esta metodología permitió un enfoque organizado que facilitó la identificación de los factores que afectaron el rendimiento del modelo y permitió optimizar su desempeño a lo largo de los experimentos realizados.
- El desbalance de clases dentro del dataset pudo haber influido en la variabilidad de los resultados, ya que, al contar con una mayor proporción de una clase masculina frente a voces femeninas, la clasificación del modelo lo pudo haber tenido sesgos y afectar a los resultados de las predicciones hechas con el dataset utilizado.
- El desarrollo de este estudio evidenció la importancia de contar con recursos computacionales adecuados, ya que durante el entrenamiento del modelo y la generación de audios sintéticos mediante StarGAN, se requirió el uso de unidades de procesamiento gráfico (GPU) de alto rendimiento, debido a que los cálculos involucrados en la transformación de voz y en la extracción de características acústicas demandan una gran capacidad de procesamiento, por lo que se usó plataformas como Google Colab con GPU NVIDIA T4 y L4 permitió ejecutar los experimentos de manera eficiente.

9. Recomendaciones

- Se recomienda ampliar el dataset con grabaciones que incluyan una mayor variedad de acentos del español ecuatoriano, así como variaciones en género, edades y contextos de grabación.
- Para futuros estudios, se sugiere utilizar equipos de grabación de mayor calidad y realizar las grabaciones en entornos controlados, minimizando ruido ambiental y garantizando la uniformidad en los datos.
- Implementar scripts o herramientas automáticas para preprocesar y estructurar los datos, reduciendo el tiempo y esfuerzo manual invertido en la preparación del dataset.

Limitaciones

- Durante el estudio, las limitaciones en los recursos computacionales dificultaron la ejecución del modelo con recursos gratuitos por lo que se tuvo que adquirir un servicio pagado.
- El dataset generado presentó limitaciones relacionadas con la calidad de grabación, ya que se realizó en entornos no controlados, con micrófonos estándar y con un desbalance de género que pudo afectar los resultados del modelo.
- Aunque el estudio abordó un contexto regional específico (acento lojano), los resultados no son necesariamente aplicables a otras variantes del español ecuatoriano o latinoamericano, lo que limita su generalización a otros entornos lingüísticos.

Trabajos Futuros

- Se sugiere desarrollar un dataset más amplio y diverso que incluya grabaciones realizadas en condiciones controladas y que representen una mayor variedad de acentos y estilos de habla.
- Realizar investigaciones sobre el desempeño de arquitecturas más modernas, como modelos basados en transformers, que podrían ofrecer un mejor rendimiento en la detección de voice spoofing.
- Trabajar con expertos en lingüística, acústica y procesamiento de señales para diseñar modelos que capten con mayor precisión las particularidades fonéticas y acústicas de los datos evaluados.

10. Bibliografía

[1] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, “StarGAN-VC: Non-parallel many-to-many Voice Conversion Using Star Generative Adversarial Networks,” 2018 IEEE Spoken Language Technology Workshop, SLT 2018 - Proceedings, pp. 266–273, 2018, doi: 10.1109/SLT.2018.8639535.

[2] A. Guevara-Rukoz et al., “Crowdsourcing latin american Spanish for low-resource text-to-speech,” LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings, no. May, pp. 6504–6513, 2020.

[3] M. Dua, C. Jain, and S. Kumar, “LSTM and CNN based ensemble approach for spoof detection task in automatic speaker verification systems,” J Ambient Intell Humaniz Comput, vol. 13, no. 4, pp. 1985–2000, 2022, doi: 10.1007/s12652-021-02960-0.

[4] H. Tak, E. Modeling, S. Spoofing, and D. D. Signal, “End-to-End Modeling for Speech Spoofing and Deepfake To cite this version : End-to-End Modeling for Speech Spoofing and Deepfake Detection Dissertation Sorbonne Université,” 2023.

[5] V. Mariño and A. Tupiza, “Citas de delitos Cibernéticos,” Fiscalía General del Estado de Ecuador., Quito, Ecuador., 2024. doi: Ticket #2024061722001666.

[6] F. Sanchez, Dra. D. S. Méndez, Estado, Mtr. Mauricio Torres, Conocimiento, and Mtr. B. R. Penale, “Perfil Criminologico Ciberdelitos: Una primera aproximacion y proyeccion institucional,” Anal Biochem, vol. 11, no. 1, pp. 55–62, 2021, [Online]. Available: https://www.fiscalia.gob.ec/pdf/politica-criminal/Ciberdelitos-Perfil-Criminologico.pdf

[7] A. Nautsch et al., “ASVspoof 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech,” IEEE Trans Biom Behav Identity Sci, vol. 3, no. 2, pp. 252–265, 2021, doi: 10.1109/TBIOM.2021.3059479.

[8] X. Liu et al., “ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild,” IEEE/ACM Trans Audio Speech Lang Process, vol. 31, pp. 2507–2522, 2023, doi: 10.1109/TASLP.2023.3285283.

[9] Z. K. Wu, “ASVspoof 2015: Automatic speaker verification spoofing and countermeasures challenge evaluation plan,” Training, vol. 10, no. 15, p. 3750, 2014.

[10] N. M. Müller et al., “MLAAD: The Multi-Language Audio Anti-Spoofing Dataset,” 2024, [Online]. Available: http://arxiv.org/abs/2401.09512

[11] P. Andr, “Voice anti-spoofing data-set built from Latin American Spanish accents implementing voice conversion and text-to-speech techniques,” 2022.

[12] S. Wehnert, M. Fiorelli, D. Picca, E. W. De Luca, and A. Stellato, “Summary of the Workshop Legal Information Retrieval meets Artificial Intelligence (LIRAI’23),” CEUR Workshop Proc, vol. 3594, pp. 1–8, 2023, doi: 10.1145/3648188.3675120.

[13] Y.-T. Tien and R. Chen, “Challenges of Artificial Intelligence in Design Education,” pp. 123–126, 2024, doi: 10.1145/3670013.3670044.

[14] M. Kasinidou, S. Kleanthous, and J. Otterbacher, “‘Artificial intelligence is a very broad term’: How educators perceive Artificial Intelligence?,” pp. 315–323, 2024, doi: 10.1145/3677525.3678677.

[15] A. T. Yang and A. T. Yang, “Social Dangers of Generative Artificial Intelligence: Review and Guidelines,” ACM International Conference Proceeding Series, pp. 654–658, 2024, doi: 10.1145/3657054.3664243.
[16] L. Wang, “How Can Generative Artificial Intelligence Techniques Facilitate Intelligent Research into Ancient Books ?,” 2024, doi: 10.1145/3690391.
[17] R. Garg, J. Han, Y. Cheng, Z. Fang, and Z. Swiecki, “Automated Discourse Analysis via Generative Artificial Intelligence,” ACM International Conference Proceeding Series, pp. 814–820, 2024, doi: 10.1145/3636555.3636879.
[18] I. Y. Kwak et al., “Voice Spoofing Detection Through Residual Network, Max Feature Map, and Depthwise Separable Convolution,” IEEE Access, vol. 11, no. April, pp. 49140–49152, 2023, doi: 10.1109/ACCESS.2023.3275790.
[19] H. Dawood, S. Saleem, F. Hassan, and A. Javed, “A robust voice spoofing detection system using novel CLS-LBP features and LSTM,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, pp. 7300–7312, 2022, doi: 10.1016/j.jksuci.2022.02.024.
[20] C. Go, N. I. Park, and O. Jeon, “A Pre-Training Framework Based on Multi-Order Acoustic,” pp. 1–10, 2023.
[21] T. Gebru et al., “Datasheets for datasets,” Commun ACM, vol. 64, no. 12, pp. 86–92, 2021, doi: 10.1145/3458723.
[22] L. Omelina, J. Goga, J. Pavlovicova, M. Oravec, and B. Jansen, “A survey of iris datasets,” Image Vis Comput, vol. 108, p. 104109, 2021, doi: 10.1016/j.imavis.2021.104109.
[23] A. Alshaibi, M. Al-Ani, A. Al-Azzawi, A. Konev, and A. Shelupanov, “The Comparison of Cybersecurity Datasets,” Data (Basel), vol. 7, no. 2, 2022, doi: 10.3390/data7020022.
[24] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer,” IEEE Trans Pattern Anal Mach Intell, vol. 44, no. 3, pp. 1623–1637, 2022, doi: 10.1109/TPAMI.2020.3019967.
[25] Y. Gong, G. Liu, Y. Xue, R. Li, and L. Meng, “A survey on dataset quality in machine learning,” Inf Softw Technol, vol. 162, no. June, p. 107268, 2023, doi: 10.1016/j.infsof.2023.107268.
[26] T. Pook, J. Freudenthal, A. Korte, and H. Simianer, “Using Local Convolutional Neural Networks for Genomic Prediction,” Front Genet, vol. 11, no. November, 2020, doi: 10.3389/fgene.2020.561497.
[27] X. Wang and J. Yamagishi, “A Practical Guide to Logical Access Voice Presentation Attack Detection,” pp. 169–214, 2022, doi: 10.1007/978-981-19-1524-6_8.
[28] N. O. N. Arallel and V. O. C. Onversion, “a Da Gan : a Daptive Gan for M Any - To -M Any,” no. 2014, pp. 1–14, 2020.
[29] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, “AUTOVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss,” 2019, [Online]. Available: http://arxiv.org/abs/1905.05879
[30] S. Studer et al., “Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology,” Mach Learn Knowl Extr, vol. 3, no. 2, pp. 392–413, 2021, doi: 10.3390/make3020020.

[31] R. Alegre-veliz, P. Gaspar-ortiz, J. Gamboa-cruzado, and W. G. Pizarro, “Machine Learning for Feeling Analysis in Twitter Communications : A Case Study in HEYDRU !, Perú,” vol. 2, pp. 126–142.
[32] N. Insights and P. Edge, “Nimdzi-AI-whitepaper,” Artificial Intelligence, Localization, Winners, Losers, Heroes, Spectators, and You, 2019.
[33] A. Castellanos Sánchez and A. Martínez de la Muela, “Trabajo en equipo con Google Drive en la universidad online,” Innovación Educativa, ISSN-e 2594-0392, ISSN 1665-2673, Vol. 13, Nº 63, 2013, págs. 75-94, vol. 13, no. 63, pp. 75–94, 2013, [Online].
[34] E. Martín Roda and S. Sassano Luiz, “Posibilidades De Google Drive Para La Docencia a Distancia Y En El Aula,” Didáctica Geográfica nº, vol. 16, pp. 203–220, 2015.
[35] I. Barrios and L. Casadei, “Promoviendo el uso de Google Drive como herramienta de trabajo colaborativo en la nube para estudiantes de ingeniería.,” Revista de Tecnología de Información y Comunicación en Educación, vol. 8, no. 1, pp. 43–56, 2014.
[36] I. Challenger Pérez, Y. Díaz Ricardo, and R. A. Becerra García, “El lenguaje de programación Python/The programming language Python,” Revista Ciencias Holguín, vol. 20, pp. 1–13, 2014, [Online]. Available: http://www.redalyc.org/articulo.oa?id=181531232001
[37] Arturo Fernández Montoro, “Pyton 3 al descubierto,” p. 320, 2013.
[38] J. R. Molina Ríos, N. M. Loja Mora, M. P. Zea Ordóñez, and E. L. Loaiza Sojos, “Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,” Revista Latinoamericana de Ingenieria de Software, vol. 4, no. 4, p. 201, 2016, doi: 10.18294/relais.2016.201-207.
[39] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu, and G. Y. Li, “Deep Learning Enabled Semantic Communications With Speech Recognition and Synthesis,” IEEE Trans Wirel Commun, vol. 22, no. 9, pp. 6227–6240, 2023, doi: 10.1109/TWC.2023.3240969.
[40] R. Johary et al., “Detection of Large-Scale Floods Using Google Earth Engine and Google Colab,” Remote Sens (Basel), vol. 15, no. 22, 2023, doi: 10.3390/rs15225368.
[41] P. Kanani and M. Padole, “Deep Learning to Detect Skin Cancer using Google Colab,” Int J Eng Adv Technol, vol. 8, no. 6, pp. 2176–2183, 2019, doi: 10.35940/ijeat.F8587.088619.
[42] B. Chen, N. Mustakin, A. Hoang, S. Fuad, and D. Wong, “VSCuda: LLM based CUDA extension for Visual Studio Code,” ACM International Conference Proceeding Series, pp. 11–17, 2023, doi: 10.1145/3624062.3624064.
[43] Q. Zhao, M. Chabbi, and X. Liu, “EasyView: Bringing Performance Profiles into Integrated Development Environments,” CGO 2024 - Proceedings of the 2024 IEEE/ACM International Symposium on Code Generation and Optimization, pp. 386–398, 2024, doi: 10.1109/CGO57630.2024.10444840.
[44] R. D. Vieira and K. Farias, “CognIDE: A Psychophysiological Data Integrator Approach for Visual Studio Code,” in ACM International Conference Proceeding Series, 2020, pp. 393–398. doi: 10.1145/3422392.3422453.
[45] C. Steppa and T. L. Holch, “HexagDLy—Processing hexagonally sampled data with CNNs in PyTorch,” SoftwareX, vol. 9, pp. 193–198, 2019, doi: 10.1016/j.softx.2019.02.010.

[46] O. C. Novac et al., “Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network,” Sensors, vol. 22, no. 22, 2022, doi: 10.3390/s22228872.
[47] C. Z. Basha, B. N. L. Pravallika, and E. B. Shankar, “An efficient face mask detector with pytorch and deep learning,” EAI Endorsed Trans Pervasive Health Technol, vol. 7, no. 25, pp. 1–8, 2021, doi: 10.4108/eai.8-1-2021.167843.
[48] O. Herrera-Alcántara and J. R. Castelán-Aguilar, “Fractional Gradient Optimizers for PyTorch: Enhancing GAN and BERT,” Fractal and Fractional, vol. 7, no. 7, pp. 1–13, 2023, doi: 10.3390/fractalfract7070500.
[49] F. S. Al-Anzi and D. AbuZeina, “Synopsis on Arabic speech recognition,” Ain Shams Engineering Journal, vol. 13, no. 2, p. 101534, 2022, doi: 10.1016/j.asej.2021.06.020.
[50] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, “Deep Audio-visual Speech Recognition,” pp. 1–11.
[51] A. Widodo, M. C. Aisiyah, I. E. Ningrum, M. A. Annas, and M. Musfiana, “Analisis Percobaan Superposisi Gelombang Suara Menggunakan Software Audacity,” Yasin, vol. 2, no. 4, pp. 459–466, 2022, doi: 10.58578/yasin.v2i4.499.
[52] A. Hujatulatif, J. Jumadi, H. Kuswanto, and A. Z. Ilma, “Analyzing and Comparing Frequency of the Birds Sound Spectrum using Audacity Software in Practicum Activity,” Jurnal Penelitian Pendidikan IPA, vol. 8, no. 6, pp. 2586–2592, 2022, doi: 10.29303/jppipa.v8i6.1697.
[53] N. Washnik, C. Suresh, and C.-Y. Lee, “Using Audacity Software to Enhance Teaching and Learning of Hearing Science Course: A Tutorial,” Teaching and Learning in Communication Sciences & Disorders, vol. 7, no. 3, 2023, doi: 10.61403/2689-6443.1284.
[54] R. Jahangir, Y. W. Teh, H. F. Nweke, G. Mujtaba, M. A. Al-Garadi, and I. Ali, “Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges,” Expert Syst Appl, vol. 171, no. February 2020, p. 114591, 2021, doi: 10.1016/j.eswa.2021.114591.
[55] O. Kjartansson, A. Gutkin, A. Butryna, I. Demirsahin, and C. Rivera, “Open-Source High Quality Speech Datasets for Basque, Catalan and Galician,” Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), no. May, pp. 21–27, 2020, [Online]. Available: https://aclanthology.org/2020.sltu-1.3
[56] G. Tamayo, “Diseños muestrales en la investigación,” Revista Científica : Semestre Económico, vol. 4, no. 7, pp. 1–14, 2000, [Online]. Available: https://revistas.udem.edu.co/index.php/economico/issue/view/130
[57] T. Otzen and C. Manterola, “Técnicas de Muestreo sobre una Población a Estudio,” International Journal of Morphology, vol. 35, no. 1, pp. 227–232, 2017, doi: 10.4067/S0717-95022017000100037.

11. Anexos

Anexo 1. Entrevista realizada al experto en inteligencia artificial

Datos personales:

- **Nombre:** Josue Alejandro Sauca Pucha
- **Cargo:** Estudiante de la carrera de Computación en la Universidad Nacional de Loja
- **Email:** josue.sauca@unl.edu.ec

Información del entrevistado:

- **Nombre:** Mgtr. Oscar M. Cumbicus Pineda.
- **Cargo:** Docente investigador de la Carrera de Computación de la Universidad Nacional de Loja”

Enlace de la entrevista realizada:

https://drive.google.com/file/d/1ly1iHPhx1Tqzfk0Jdr96QkVz5Gyh6jG5/view?usp=sharing

Transcripción de la entrevista realizada:

Estudiante: Buenas Tardes Ingeniero Oscar Cumbicus, la presente entrevista actual busca ver la viabilidad sobre el uso de una derivación de la Red Neuronal Convolucional (CNN), enfocada en la clasificación de audios de suplantación de identidad que ha sido entrenado en audios en ingles con audios en español mediante el uso de la Red Neuronal Convolucional Ligera (LCNN).

Estudiante: ¿Considera que la inteligencia artificial puede llegar a incidir gravemente en algún punto la seguridad e integridad de las personas con herramientas que empleen audio?

Entrevistado: La inteligencia artificial depende de su uso, la inteligencia artificial así como a ti te dan un cuchilla que lo puedes usar para el bien o para el mal, la inteligencia artificial también se puede usar para el bien o para el mal, entonces si una persona la quiere utilizar para el mal seguramente la va a usar de una mala manera en la seguridad, pero también puede ser muy beneficioso para la seguridad de muchas cosas, por ejemplo la suplantación de identidad a través de rostros o de audios, de huellas digitales, de firmas de rubricas, entonces todo depende de cómo utilices la inteligencia artificial, para el bien o para el mal de las dos formas puede incidir.

Estudiante: En base a su experiencia ¿qué tipo de técnicas o herramientas de audio basadas en inteligencia artificial conoce y considera usted que las personas puedan usarlas con fin de hacer daño a otros individuos?

Entrevistado: Existen un montón de herramientas en el mercado, que te permiten a ti reproducir o clonar voces, una de ellas es xtss sino estoy mal, que te permite a ti con

un fragmento de tu voz reproducir otros fragmentos de texto con tu voz, en la actualidad existen herramientas gratuitas que te permiten reproducir texto con tu voz casi a nivel que sea tu voz, entonces hay muchas herramientas en el internet que te permiten hacer eso, que utilizan inteligencia artificial generativa.

Estudiante: ¿Ha tenido la oportunidad de trabajar en el desarrollo de modelos de machine learning que involucren el uso con audios, y si es así cuales considera que son los principales problemas al momento de trabajar con este tipo formato?

Entrevistado: Si he tenido la oportunidad no ha fondo, pero si he tenido la oportunidad de realizar trabajos, uno de los principales problemas es que se genera ruido del ambiente, ya que la voz se transmite mediante el aire, y dentro del medio de transporte existe mucho ruido, uno de los principales problemas es la eliminación del ruido, para que solamente quede la voz ya sea humana o animal, este es uno de los principales problemas que se encuentran en este tipo de trabajos.

Estudiante: ¿Conoce acerca de las Redes neuronales convolucionales (CNN), avances y aspectos en donde se está utilizando actualmente dichas redes empleando audios?

Entrevistado: Si las Redes Neuronales Convolucionales, principalmente trabajan con imágenes, pero para poder procesar un audio lo que se hace es sacar la imagen de un espectrograma o un mapa de frecuencia de la voz, ya sea de un animal o de una persona, tú puedes a través de varios programas ya sea de Python mismo, tu puedes a través de la voz, sacar el espectrograma de tu voz, entonces como las Redes Neuronales Convolucionales están creadas para procesamiento de imágenes se usan este tipo de redes para procesar el audio, las Redes Neuronales Convolucionales lo que hacen es transformar esa imagen del espectrograma a una matriz de caracteres, perdón a una matriz de números o pesos para ese espectrograma, luego son llevadas a un conjunto de capas donde se realiza la convolución es decir la multiplicación de matrices por filtros o matrices más pequeñas que permitan clasificar este tipo de imágenes, que en este caso son imágenes de espectrogramas de sonido o de voz.

Estudiante: A continuación, se va a indagar un poco más en lo que es el modelo de manera general, la siguiente pregunta es ¿Ha experimentado algún tipo de problema al momento de aplicar fine tuning a modelos de machine learning?

Entrevistado: Bueno uno de los principales problemas al realizar Fine Tuning es que no siempre se encuentra la arquitectura que se utilizó, es decir se creó un proyecto para detección de sonido, pero no está la arquitectura que utilizaron, es decir no hay el código donde está la arquitectura por tal motivo no se puede reproducir, por tal motivo no se puede hacer el ajuste fino de esos modelos, otro de los principales problemas es que se necesita bastante hardware para este tipo de procesamiento, es

decir para ajustar los hiperparametros muchas veces cuando ya se sube los hiperparametros como learning rate, el batch size de los datos , vamos a necesitar más hardware que no se lo encuentra en una computadora portátil, sino que hay que acudir a Google Colab pagado o a super computadoras como las de Cedia o como los que tiene la Carrera de Computación.

Estudiante: ¿Considera que la técnica Fight Tuning es adecuada para un mejoramiento de modelos de inteligencia artificial?

Entrevistado: Si de hecho está comprobado que el ajuste de modelos pre-entrenados ha servido de mucho en muchas tareas, inclusive en las Redes Neuronales Convolucionales, por el tema que te hable anteriormente de los recursos de hardware, lo que se hace es un ajuste a los modelos que han sido entrenados con millones de datos, para ajustarlos a una tarea más específica como la que tú quieres hacer de detección de sonidos en español, se puede utilizar un modelo en inglés y se lo podría ajustar para realice la detección en español.

Estudiante: ¿Qué tipo de problemas se pueden dar al momento de aplicar la técnica fine tuning en un modelo pre-entrenado?

Entrevistado: Lo que te decía anteriormente, el mayor problema son los recursos de hardware es uno de los mayores problemas que se va a dar, también la desactualización de librerías, porque muchas veces ya las librerías que se han usado para ese tipo de experimentos ya no están en esas versiones y ya no son compatibles con otras librerías, esos serían los principales problemas.

Estudiante: ¿Qué técnicas conoce que puedan ayudar a encontrar los valores adecuados al momento de realizar la configuración de los hiper parámetros en un modelo?

Entrevistado: Bueno actualmente existen muchas técnicas como SPF, LORA que te permiten buscar los parámetros adecuados automáticamente para el Fine tuning, son técnicas para que se detecten cuáles son los hiperparametros adecuados para que tu puedas finetunear un modelo y eso.

Estudiante: Muchas gracias ingeniero por la información.

Anexo 2. Invitación y respuesta del muestreo por conveniencia

Para llevar a cabo el proceso de selección de los participantes, se utilizó la técnica de muestreo por conveniencia, considerando como criterio principal la disponibilidad horaria de los voluntarios, la invitación fue enviada a través del correo institucional de la Universidad Nacional de Loja, dirigida a estudiantes de la carrera de Computación.

A continuación, se presenta el contenido del correo enviado, en el que se explica el propósito del proyecto y los requisitos necesarios para participar, así como las respuestas recibidas por parte de los voluntarios.

Invitación Mediante el Correo Institucional

8 de 171

Invitación al proyecto de integración curricular Recibidos ×

Josue Alejandro Sauca Pucha <josue.sauca@unl.edu.ec> dom, 20 oct, 19:56

para Danny, Juan, Juan, Luis, Paulina, Jimmy, Melissa, Edy, JENNIFER, Anthony, Gerardo, Vanessa, David, Anderson

Estimado/a,

Espero que este mensaje le encuentre bien. Mi nombre es Josue Alejandro Sauca Pucha y actualmente estoy desarrollando un proyecto de investigación titulado **"Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en Español"**. Este estudio forma parte de mi tic en la carrera de Ciencias de Computación.

El objetivo de este proyecto es crear un conjunto de audios grabados que nos permita avanzar en la investigación sobre la suplantación de voz, utilizando tecnologías de inteligencia artificial. Para ello, necesitamos la participación de voluntarios que estén dispuestos a grabar algunas frases en un entorno controlado.

Detalles de la grabación:

- **Duración estimada:** 15-30 minutos
- **Requisitos:** Solo necesitamos que traiga su mejor disposición para colaborar.

Si tiene alguna pregunta o desea confirmar su participación, no dude en responder a este correo o contactarme personalmente.

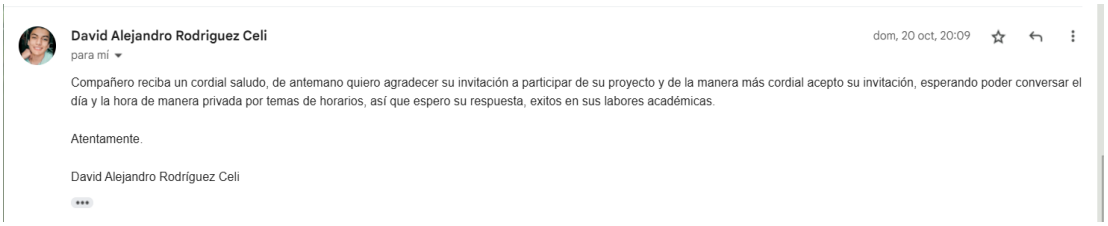
Agradezco de antemano su tiempo y apoyo, y espero contar con su valiosa participación en este proyecto.

Saludos cordiales.

9

Respuestas de las Personas a la invitación

- David Rodriguez



David Alejandro Rodriguez Celi dom, 20 oct, 20:09

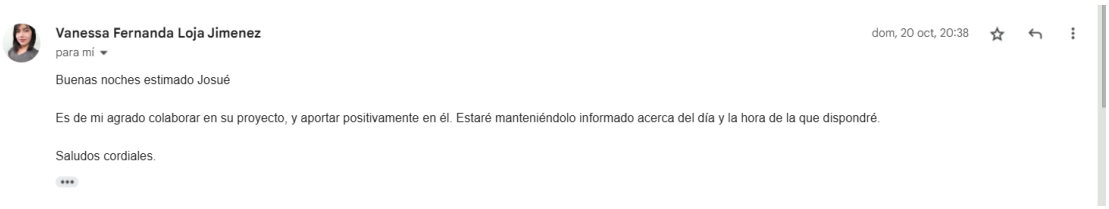
para mí

Compañero reciba un cordial saludo, de antemano quiero agradecer su invitación a participar de su proyecto y de la manera más cordial acepto su invitación, esperando poder conversar el día y la hora de manera privada por temas de horarios, así que espero su respuesta, exitos en sus labores académicas.

Atentamente.

David Alejandro Rodríguez Celi

- Vanessa Loja



Vanessa Fernanda Loja Jimenez dom, 20 oct, 20:38

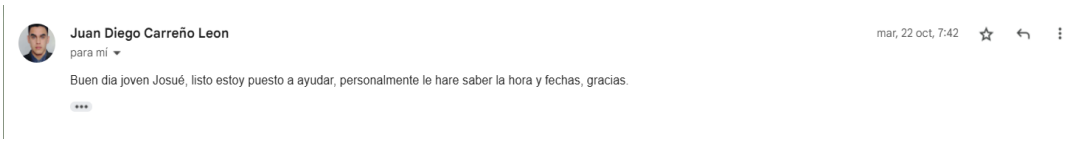
para mí

Buenas noches estimado Josué

Es de mi agrado colaborar en su proyecto, y aportar positivamente en él. Estaré manteniéndolo informado acerca del día y la hora de la que dispondré.

Saludos cordiales.

- Juan Carreño

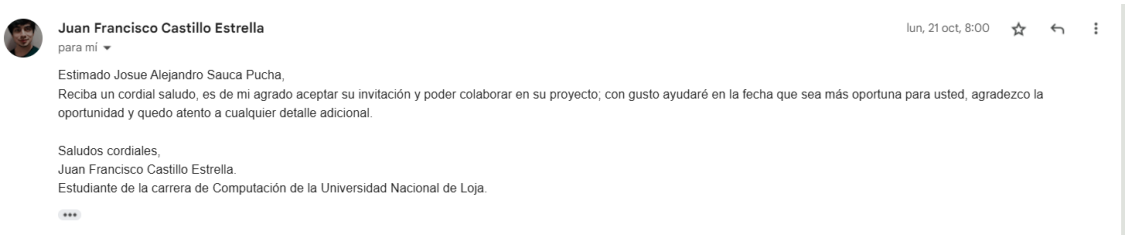


Juan Diego Carreño Leon mar, 22 oct, 7:42

para mí

Buen dia joven Josué, listo estoy puesto a ayudar, personalmente le hare saber la hora y fechas, gracias.

- Juan Castillo



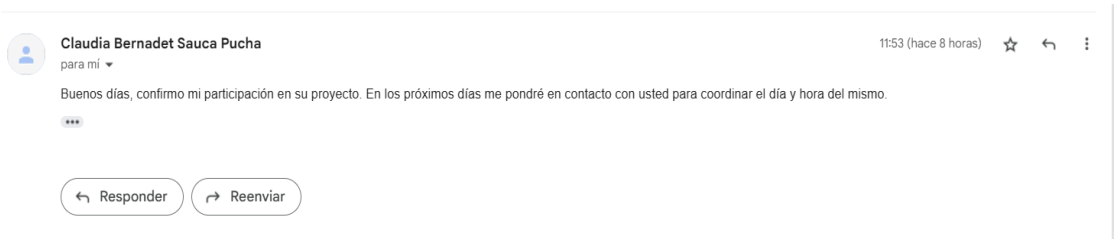
Juan Francisco Castillo Estrella lun, 21 oct, 8:00

para mí

Estimado Josue Alejandro Sauca Pucha,
Reciba un cordial saludo, es de mi agrado aceptar su invitación y poder colaborar en su proyecto; con gusto ayudaré en la fecha que sea más oportuna para usted, agradezco la oportunidad y quedo atento a cualquier detalle adicional.

Saludos cordiales,
Juan Francisco Castillo Estrella.
Estudiante de la carrera de Computación de la Universidad Nacional de Loja.

- Claudia Sauca



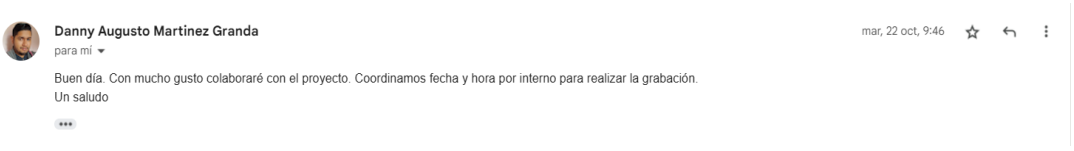
Claudia Bernadet Sauca Pucha 11:53 (hace 8 horas)

para mí

Buenos días, confirmo mi participación en su proyecto. En los próximos días me pondré en contacto con usted para coordinar el día y hora del mismo.

Responder Reenviar

- Dany Martínez

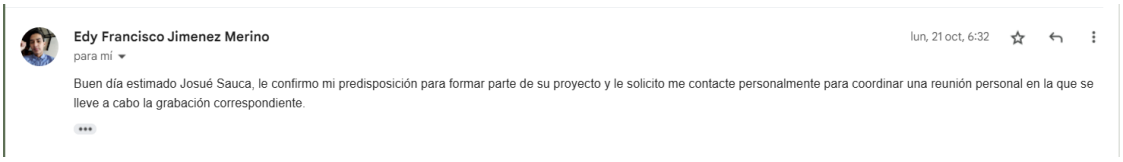


Danny Augusto Martinez Granda mar, 22 oct, 9:46

para mí

Buen día. Con mucho gusto colaboraré con el proyecto. Coordinamos fecha y hora por interno para realizar la grabación.
Un saludo

- Edy Jimenez

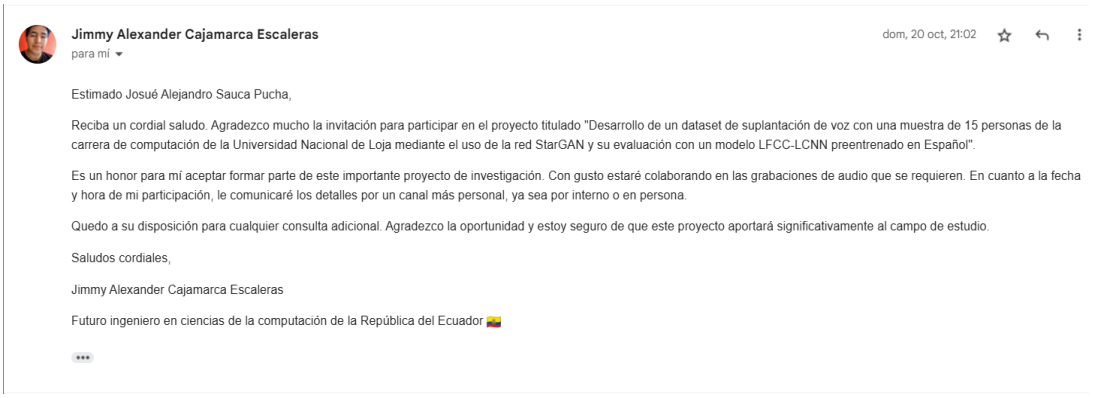


Edy Francisco Jimenez Merino lun, 21 oct, 6:32

para mí

Buen día estimado Josué Sauca, le confirmo mi predisposición para formar parte de su proyecto y le solicito me contacte personalmente para coordinar una reunión personal en la que se lleve a cabo la grabación correspondiente.

- Jimmy Cajamarca



Jimmy Alexander Cajamarca Escaleras dom, 20 oct, 21:02

para mí

Estimado Josué Alejandro Sauca Pucha,

Reciba un cordial saludo. Agradezco mucho la invitación para participar en el proyecto titulado "Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en Español".

Es un honor para mí aceptar formar parte de este importante proyecto de investigación. Con gusto estaré colaborando en las grabaciones de audio que se requieren. En cuanto a la fecha y hora de mi participación, le comunicaré los detalles por un canal más personal, ya sea por interno o en persona.

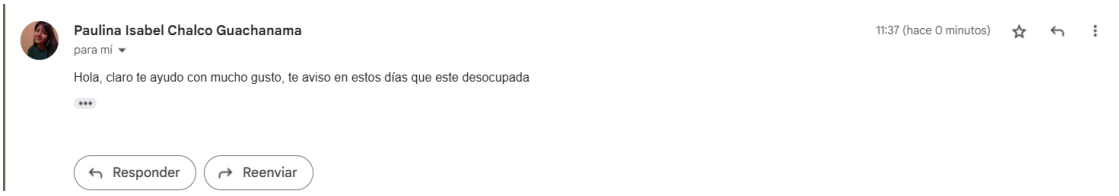
Quedo a su disposición para cualquier consulta adicional. Agradezco la oportunidad y estoy seguro de que este proyecto aportará significativamente al campo de estudio.

Saludos cordiales,

Jimmy Alexander Cajamarca Escaleras

Futuro ingeniero en ciencias de la computación de la República del Ecuador 🇪🇨

- Paulina Chalco



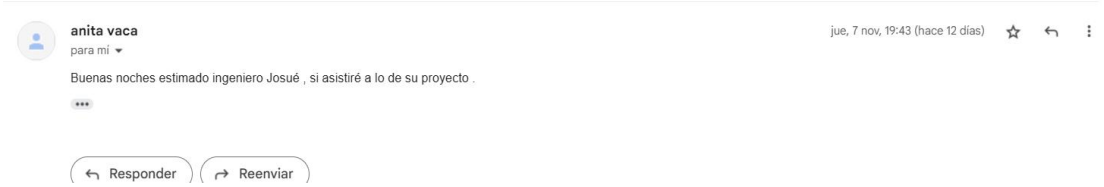
Paulina Isabel Chalco Guachanama 11:37 (hace 0 minutos)

para mí

Hola, claro te ayudo con mucho gusto, te aviso en estos días que este desocupada

Responder Reenviar

- Ana Paula Vaca

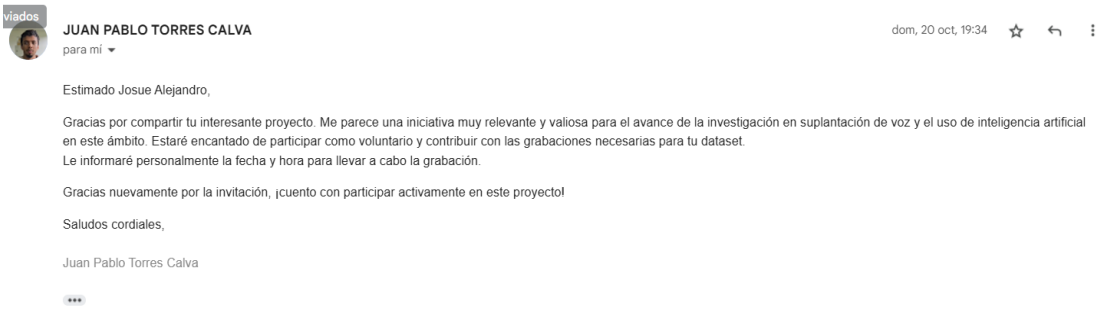


anita vaca
para mí
jue, 7 nov, 19:43 (hace 12 días)

Buenas noches estimado ingeniero Josué , si asistiré a lo de su proyecto .

Responder Reenviar

- Pablo Torres



JUAN PABLO TORRES CALVA
para mí
dom, 20 oct, 19:34

Estimado Josue Alejandro,

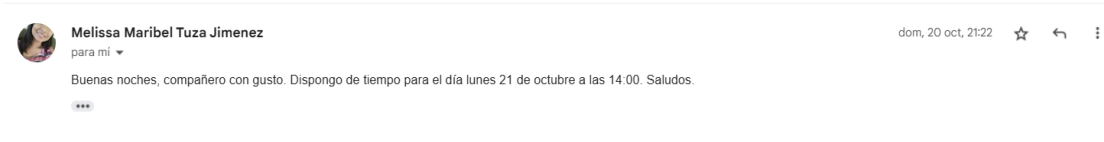
Gracias por compartir tu interesante proyecto. Me parece una iniciativa muy relevante y valiosa para el avance de la investigación en suplantación de voz y el uso de inteligencia artificial en este ámbito. Estaré encantado de participar como voluntario y contribuir con las grabaciones necesarias para tu dataset.
Le informaré personalmente la fecha y hora para llevar a cabo la grabación.

Gracias nuevamente por la invitación, ¡cuento con participar activamente en este proyecto!

Saludos cordiales,

Juan Pablo Torres Calva

- Melissa Tuza



Melissa Maribel Tuza Jimenez
para mí
dom, 20 oct, 21:22

Buenas noches, compañero con gusto. Dispongo de tiempo para el día lunes 21 de octubre a las 14:00. Saludos.

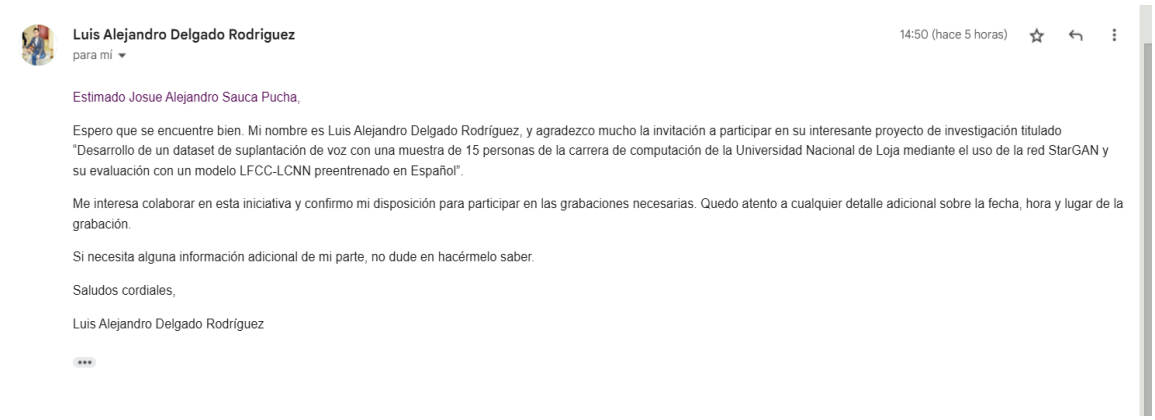
- Antony Luzuriaga

Anthony Lenin Luzuriaga Gonzalez
para mí
vie, 25 oct, 13:59

entendido joven me avisa el dia

Responder Reenviar

- Luis Delgado



Luis Alejandro Delgado Rodriguez
para mí
14:50 (hace 5 horas)

Estimado Josue Alejandro Sauca Pucha,

Espero que se encuentre bien. Mi nombre es Luis Alejandro Delgado Rodríguez, y agradezco mucho la invitación a participar en su interesante proyecto de investigación titulado "Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en Español".

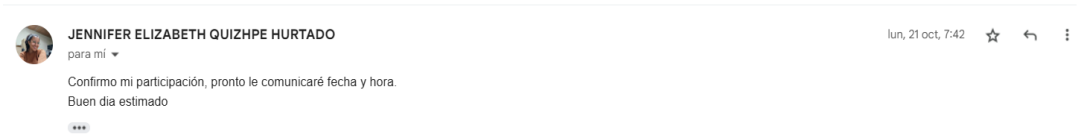
Me interesa colaborar en esta iniciativa y confirmo mi disposición para participar en las grabaciones necesarias. Quedo atento a cualquier detalle adicional sobre la fecha, hora y lugar de la grabación.

Si necesita alguna información adicional de mi parte, no dude en hacérmelo saber.

Saludos cordiales,

Luis Alejandro Delgado Rodríguez

- Jennifer Elizabeth

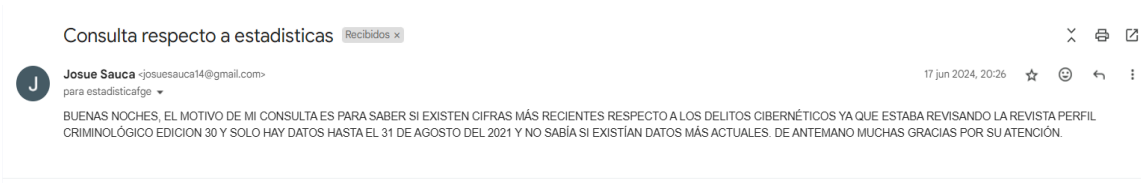


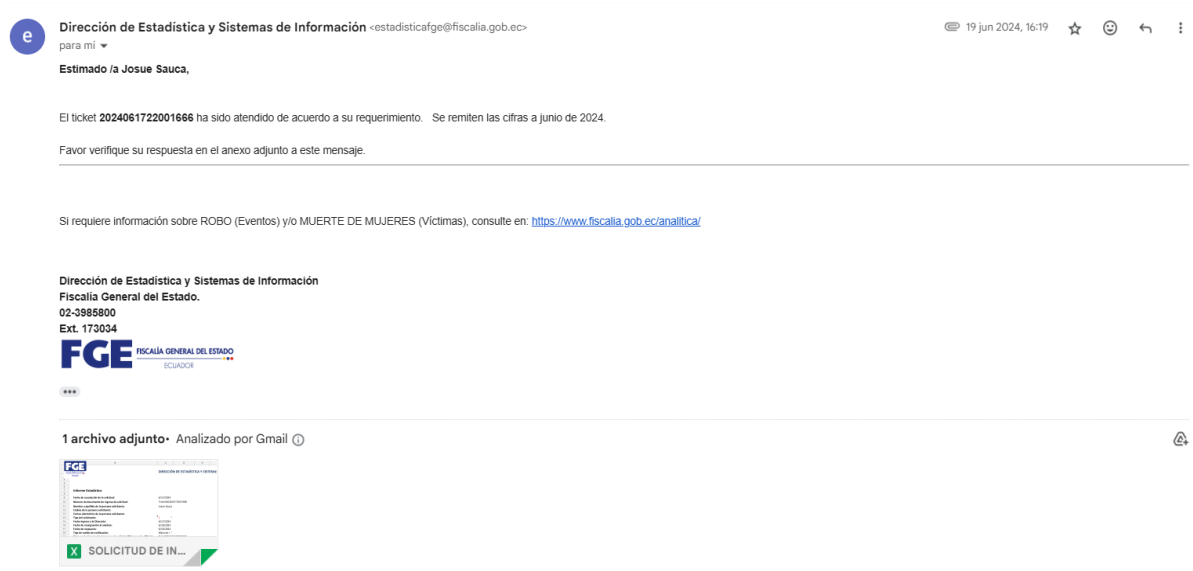
JENNIFER ELIZABETH QUIZHPE HURTADO
para mí
lun, 21 oct, 7:42

Confirmo mi participación, pronto le comunicaré fecha y hora.
Buen dia estimado

Anexo 3. Datos delitos cibernéticos consulta a la fiscalía

Sección donde se presenta una consulta realizada por correo electrónico a la Fiscalía General del Estado Ecuatoriano, donde se solicitando información actualizada sobre las estadísticas de delitos cibernéticos del 2024, el motivado fue por la revisión de la revista "Perfil Criminológico" (Edición 30), que contiene datos únicamente hasta el 31 de agosto de 2021.





Anexo 4. Certificado de Traducción

CERTIFICACIÓN DE TRADUCCIÓN

Loja, 11 de marzo de 2025

Lic. Viviana Valdivieso Loyola Mg. Sc.
DOCENTE DE INGLÉS

A petición verbal de la parte interesada:

CERTIFICA:

Que, desde mi legal saber y entender, como profesional en el área del idioma inglés, he procedido a realizar la traducción del resumen, correspondiente al Trabajo de Integración Curricular titulado **Desarrollo de un dataset de suplantación de voz con una muestra de 15 personas de la carrera de computación de la Universidad Nacional de Loja mediante el uso de la red StarGAN y su evaluación con un modelo LFCC-LCNN preentrenado en español,** de la autoría de: **Josue Alejandro Sauca Pucha,** portadora de la cédula de identidad número **1105580581**

Para efectos de traducción se han considerado los lineamientos que corresponden a un nivel de inglés técnico, como amerita el caso.

Es todo cuanto puedo certificar en honor a la verdad, facultando al portador del presente documento, hacer uso del mismo, en lo que a bien tenga.

Atentamente. -

Firmado electrónicamente por:
VIVIANA DEL CISNE
VALDIVIESO LOYOLA

Lic. Viviana Valdivieso Loyola Mg. Sc.
1103682991

N° Registro Senescyt 4to nivel **1031-2021-2296049**
N° Registro Senescyt 3er nivel **1008-16-1454771**