

**Sistema de identificación de instrumentos musicales de la región andina usando
técnicas de machine learning**

Willian David Correa Valencia

**Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tulua
2025**

**Sistema de identificación de instrumentos musicales de la región andina usando
técnicas de machine learning**

William David Correa Valencia

Código 202060016

`william.david.correa@correounivalle.edu.co`

Director

Hector Fabio Ocampo

Profesor de la Escuela de Ingeniería de Sistemas y Computación

`hector.ocampo@correounivalle.edu.co`

Universidad del Valle

Facultad de ingeniería

Escuela de ingeniería de sistemas y computación

Tulua

2025

Tabla de Contenido

Resumen	4
Abstract	5
1. Introducción	6
2. Formulación del Problema	7
2.1. Descripción del Problema	7
2.2. Definición del Problema	7
3. Marco Referencial	9
3.1. Marco Teorico	9
3.2. Estado del arte	11
3.2.1. Teoría general de sistemas	11
3.2.2. Algoritmos de Aprendizaje Automático	11
3.2.3. Análisis de Señales de Audio	12
3.2.4. Big Data e Integración de Datos	12
3.3. Modelos Personalizados - Planteamiento	12
3.4. Modelos Personalizados	12
3.5. Antecedentes	12
3.6. Marco Conceptual	13
3.6.1. Sistema	13
3.6.2. Software	13
3.6.3. Machine Learning	14
3.6.4. Espectrograma	14
4. Alcances de la propuesta	17
4.1. Definición	17
4.2. Objetivos	18
4.2.1. Objetivos específicos	18
4.2.2. Entregables y Objetivos-Resultado esperado	18
4.2.3. Marco lógico de la propuesta	18
5. Recopilación de Datos en Bruto	22
5.1. Fuentes de Datos	22
5.2. Proceso de Extracción y Preprocesamiento de Audio	22
5.3. Grabación Directa de Instrumentos Andinos	23
5.4. Herramientas Utilizadas	23
5.5. Organización del Dataset	23
5.6. Descripción del Conjunto de Datos	24
5.7. Validación de los Datos	24
5.8. Análisis del espectrograma de potencia	24
5.9. Aumento de datos (Data Augmentation) en audio	26

5.10. Desafíos y Consideraciones	27
5.11. Resumen del Proceso	27
6. Preprocesamiento de Datos de Audio	28
6.1. Objetivo	28
6.2. Entregable	28
6.3. Planteamiento de la Metodología General del Sistema de Clasificación de Instrumentos Andinos	28
6.3.1. Metodología General del Sistema	30
6.4. Parámetros de Preprocesamiento	31
6.5. Implementación de Técnicas de Procesamiento de Señales	31
6.5.1. Windowing	31
6.5.2. Generación de Espectrogramas	31
6.6. Pipeline de Preprocesamiento	32
6.7. Análisis y Manipulación de Datos	32
6.8. Metodología de Extracción de Características	32
6.9. Optimización del Dataset	33
6.10. Almacenamiento de los Datos Preprocesados	33
6.11. Validación del Preprocesamiento	34
7. Implementación de Inteligencia Artificial para el Reconocimiento de Instrumentos Andinos	35
7.1. Objetivo	35
7.2. Entregable	36
7.3. Parámetros de Entrenamiento del Modelo	36
7.4. Modelos de Aprendizaje Automático	36
7.5. Diseño del Modelo de Red Neuronal Convolutiva	37
7.5.1. Arquitectura Base de la CNN	37
7.5.2. Sistema Ensamblado de CNN	38
7.5.3. Explicación del Sistema Ensamblado	38
7.5.4. Aplicación en Instrumentos Andinos	40
7.6. Entrenamiento del Modelo	40
7.7. Evaluación y Validación del Modelo	40
7.7.1. Métricas de Evaluación	40
7.7.2. Validación Cruzada	41
7.7.3. Pruebas de Clasificación	41
7.8. Estrategias de Optimización del Modelo	41
7.9. Evaluación y Análisis del Modelo	41
7.10. Escalabilidad y Generalización	42
7.11. Clasificación Multiclase de Instrumentos Andinos	43
7.11.1. Diagrama del Proceso de Clasificación	43
7.11.2. Explicación del Proceso	43
7.11.3. Aplicación en Instrumentos Tradicionales	43
7.11.4. Matriz de Confusión del Modelo Multiclase	44

7.12.	Representación del Modelo SVM para Clasificación de Instrumentos Andinos	45
7.12.1.	Concepto de Separación con Hiperplano	45
7.12.2.	Funcionamiento General	46
7.12.3.	Evaluación con Matriz de Confusión	46
7.13.	Explicación del Modelo	48
7.14.	Explicación de la Gráfica 3: Representación del Modelo SVM	48
7.15.	Visualización de Resultados	49
7.15.1.	Visualización de Resultados	49
8.	Construcción del Prototipo: Desarrollo del Frontend y Backend	49
8.1.	Objetivo	49
8.2.	Arquitectura del Sistema	50
8.2.1.	Diagrama de Componentes	50
8.2.2.	Diagrama de Despliegue	51
8.3.	Comportamiento del Sistema	51
8.3.1.	Diagrama de Casos de Uso	51
8.3.2.	Diagrama de Secuencia	51
8.4.	Desarrollo del Frontend - Planteamiento	53
8.5.	Desarrollo del Frontend	53
8.5.1.	Arquitectura General	53
8.5.2.	Funciones Clave del Frontend	54
8.5.3.	Interacción Usuario - Sistema	54
8.5.4.	Resumen de Herramientas Usadas	55
8.6.	Desarrollo del Backend	55
8.6.1.	Arquitectura y Organización General	55
8.6.2.	Funciones Específicas del Backend	56
8.6.3.	Flujo de Clasificación de Audio	56
8.6.4.	Modelos de Machine Learning Integrados	56
8.6.5.	Validación, Seguridad y Eficiencia	57
8.6.6.	Resumen de Funcionalidades Técnicas del Backend	57
8.7.	Despliegue e Integración	59
9.	Metodología	59
9.1.	Metodología de Investigación Documental en Ingeniería de Sistemas	59
9.2.	Gestión del Proyecto con Kanban y Trello	60
9.3.	Implementación Técnica	61
10.	Conclusiones	63
11.	Trabajos Futuros	66
12.	Referencias	67

Resumen

Este proyecto presenta el desarrollo de un sistema de identificación de instrumentos musicales específicos de la región andina utilizando diferentes técnicas de Machine Learning. El sistema propuesto emplea algoritmos de clasificación para analizar y categorizar sonidos musicales, facilitando la identificación automática de instrumentos como la quena, el charango, la zampoña y el bombo andino, entre otros adicionales. Se recolectarán datos acústicos de diversas fuentes y se procesarán para extraer características relevantes que alimenten los modelos de aprendizaje automático.

Palabras clave (Ieee, n.d): – Computer applications, Musical instrument digital interfaces, Musical instrument, Music, Rhythm, Educational technology, Machine learning, Pattern recognition.

Abstract

This project presents the development of a system for identifying musical instruments specific to the Andean region using Machine Learning techniques. The proposed system employs classification algorithms to analyze and categorize musical sounds, facilitating the automatic identification of instruments such as the quena, the charango, the zampoña, and the Andean bombo. Acoustic data were collected from various sources and processed to extract relevant features that feed machine learning models.

Keywords (Ieee, n.d): – Computer applications, Musical instrument digital interfaces, Musical instrument, Music, Rhythm, Educational technology, Machine learning, Pattern recognition.

1. Introducción

La identificación de instrumentos musicales a partir de grabaciones de audio es una tarea compleja que representa un desafío tanto para los oyentes humanos como para los sistemas automatizados. Esto se debe a que el sonido producido por cada instrumento puede variar considerablemente dependiendo de factores como la técnica del intérprete, las condiciones del instrumento y el entorno de grabación. Además, cuando múltiples instrumentos suenan simultáneamente, distinguir entre ellos se convierte en una tarea aún más difícil. Para abordar estos desafíos, hemos intentado desarrollar algoritmos de aprendizaje automático y redes neuronales profundas capaces de aprender y clasificar con alta precisión los distintos sonidos de los instrumentos musicales. Estos modelos pueden ser entrenados con grandes conjuntos de datos de grabaciones de audio, permitiéndoles captar las sutiles diferencias acústicas que caracterizan a cada instrumento [1].

El seguimiento del ritmo es una habilidad fundamental para cualquier músico. Según un artículo publicado por Landr, el ritmo en la música se define como la organización temporal de patrones de sonido regulares [2]. Tradicionalmente, los músicos de música andina han desarrollado esta habilidad practicando con metrónomos o siguiendo patrones rítmicos constantes. Sin embargo, estas técnicas presentan limitaciones significativas, ya que no ofrecen retroalimentación efectiva en tiempo real ni permiten una identificación digital eficiente de los errores rítmicos.

En este contexto, el desarrollo del sistema propuesto surge como una respuesta innovadora a las transformaciones que la tecnología de la información ha traído a la educación musical. En la era moderna, es esencial que los educadores adopten enfoques pedagógicos innovadores y modelos mejorados que aprovechen las herramientas tecnológicas disponibles [3]. Esta aplicación busca integrar de manera flexible la tecnología musical computarizada para enriquecer el aprendizaje de los instrumentos tradicionales de la región andina. Al hacerlo, no solo se maximiza el potencial de los músicos, sino que también se promueve un avance significativo en la educación musical, facilitando un proceso de aprendizaje más eficiente, accesible y personalizado.

2. Formulación del Problema

2.1. Descripción del Problema

La identificación automática de instrumentos musicales a partir de grabaciones de audio es un desafío técnico y científico que ha ganado relevancia en áreas como la musicología, la inteligencia artificial y las tecnologías multimedia. La creciente disponibilidad de grandes bases de datos de música accesibles al público ha generado una necesidad crítica: desarrollar sistemas capaces de clasificar y reconocer automáticamente los contenidos de estos archivos [4]. Esta capacidad tiene aplicaciones prácticas en múltiples dominios, como la indexación automática de datos, la recuperación de información en bases de datos musicales, el reconocimiento de estilos musicales, la edición de audio, la generación de listas de reproducción personalizadas, la transcripción musical y la recuperación eficiente de contenido sonoro [3].

Sin embargo, a pesar de los avances significativos en la identificación de instrumentos musicales mediante técnicas de aprendizaje automático, existe una brecha importante en la investigación específica sobre los instrumentos tradicionales de la región andina. Estos instrumentos, como la zampoña, el charango, la quena y otros, presentan características acústicas únicas que los diferencian de los instrumentos occidentales. Además, su sonido puede variar considerablemente debido a factores como la técnica del intérprete, el estado del instrumento, las condiciones ambientales y la presencia de múltiples instrumentos tocando simultáneamente. Este contexto plantea un desafío particular que no ha sido abordado sistemáticamente en la literatura científica.



Figura 1: Árbol problemático que ilustra los factores involucrados en la identificación de instrumentos musicales andinos.

2.2. Definición del Problema

El problema central de este trabajo se formula de la siguiente manera:

¿Cómo desarrollar un sistema basado en técnicas de Machine Learning que permita la identificación automática de instrumentos musicales típicos de la región andina (como la zampoña, el charango y la quena) a partir de grabaciones de audio, considerando la variabilidad inherente en la técnica del intérprete, el estado del instrumento, las condiciones del entorno de grabación y la posible superposición de múltiples instrumentos?

Este problema implica abordar varios desafíos clave:

- Capturar las características acústicas únicas de los instrumentos andinos.
- Diseñar modelos útiles que puedan manejar la variabilidad en las condiciones de grabación.
- Desarrollar algoritmos capaces de distinguir múltiples instrumentos tocando individualmente.
- Proporcionar una solución práctica que pueda ser utilizada en aplicaciones educativas, culturales y tecnológicas.

La elección de la música andina responde a su riqueza tímbrica, diversidad instrumental y profundo valor cultural. Este género representa una de las expresiones musicales más emblemáticas de América Latina, especialmente en regiones como Perú, Bolivia, Ecuador y el norte de Chile, donde los instrumentos tradicionales no solo cumplen una función melódica, sino también simbólica y ritual.

La resolución de este problema no solo contribuiría al avance del reconocimiento automático de instrumentos musicales, sino que también tendría un impacto significativo en la preservación, clasificación y difusión digital del patrimonio sonoro andino, fortaleciendo su permanencia en el contexto tecnológico contemporáneo.

3. Marco Referencial

3.1. Marco Teorico

La identificación de instrumentos musicales de la región andina mediante técnicas de Machine Learning representa una intersección innovadora entre la música tradicional y la tecnología moderna [2]. Este enfoque aprovecha la capacidad de los algoritmos de aprendizaje automático para procesar y analizar digitalmente las características acústicas de los sonidos, mejorando así la precisión y eficiencia en la identificación de instrumentos musicales específicos [3]. El area de Machine Learning (ML) ha evolucionado drásticamente en los últimos años y ahora se está utilizando a escala global por los ingenierios de sistemas para resolver diferentes problemas.

Como un subconjunto de la inteligencia artificial, ML se ha considerado vital al desarrollar sistemas de software para dominios tan variados como el reconocimiento de voz/imagen o automotriz. Las técnicas de ML también se han utilizado para abordar varios problemas y actividades de ingeniería de software. Comúnmente, ML se ha empleado en la predicción de defectos, la estimación de esfuerzo y la identificación de patrones y similitudes en el código fuente. Estas técnicas están dirigidas esencialmente a resolver problemas [5], los cuales a menudo pueden ser difíciles de analizar para las personas como la identificación y reconocimiento de instrumentos.

En el trabajo de Kumar en India [6], se propone un modelo de aprendizaje profundo que utiliza imágenes de espectrogramas generadas a partir de grabaciones de audio para la clasificación automática de instrumentos musicales. La arquitectura empleada está basada en redes neuronales convolucionales (CNN), diseñadas para capturar patrones espectro-temporales representativos del timbre de cada instrumento. Diversos estudios han demostrado que los coeficientes cepstrales en la escala mel (MFCCs) constituyen una de las representaciones más eficaces para extraer características acústicas de señales de audio en tareas de clasificación musical. En el caso del trabajo de Yi Zhou de Yunnan. [7] aplica modelos de aprendizaje automático combinados con MFCCs para identificar instrumentos musicales a partir de grabaciones crudas, sin necesidad de ingeniería de características compleja. El enfoque permite extraer información relevante directamente desde los datos originales, reduciendo la intervención manual y mejorando la eficiencia del sistema. Este modelo fue entrenado con un conjunto curado de muestras que incluía instrumentos musicales distintos, logrando resultados robustos en la predicción del instrumento interpretado. Además, el estudio enfatiza la aplicabilidad del sistema para usuarios principiantes, al facilitar la armonización musical sin conocimientos técnicos avanzados. Estos hallazgos respaldan el uso de Espectogramas y modelos automáticos en sistemas como el desarrollado en este proyecto, orientados a la clasificación de instrumentos musicales andinos.

En el trabajo realizado por Bhagyalakshmi [8], los autores abordan el problema de la clasificación automática de sonidos de instrumentos musicales mediante el uso de redes neuronales convolucionales (CNN), complementadas con técnicas de aumentación de datos. Los resultados mostraron mejoras significativas en la precisión del sistema, especialmente en clases con menor representación original, lo que valida la utilidad de técnicas de data augmentation y tranformacion en entornos con datasets desequilibrados o grabaciones rui-

dosas además se enfatizó el uso de Autoencoders para los Espectrogramas. En el trabajo de Sivalingam [9], se propone un enfoque híbrido para la clasificación de instrumentos musicales en grabaciones polifónicas, combinando redes neuronales convolucionales (CNN) con Transformadores. Mientras que las CNN capturan patrones espaciales en espectrogramas, los Transformadores permiten modelar relaciones temporales a largo plazo. Esta fusión de arquitecturas demuestra ser eficaz en escenarios complejos donde múltiples instrumentos suenan simultáneamente. Aunque el presente proyecto se centra en sonidos monofónicos, este trabajo representa una vía sólida para futuras investigaciones orientadas a la clasificación en contextos más realistas y desafiantes. Numerosos estudios recientes han demostrado la efectividad de técnicas como los coeficientes cepstrales en la escala mel (MFCCs) para la identificación de instrumentos musicales a partir de audio crudo. Por ejemplo, el trabajo de Yi Zhou et al. [7] presenta un sistema de reconocimiento de instrumentos basado en aprendizaje automático que utiliza MFCCs y modelos como SVM y CNN, logrando una precisión del 91 % en clasificación de familias instrumentales. El estudio destaca que incluso con entradas de tan solo tres segundos de duración, los modelos pudieron identificar características tímbricas relevantes, sin requerir ingeniería de características compleja.

El uso de representaciones espectrales como los espectrogramas mel y los coeficientes cepstrales en la escala mel (MFCC) ha demostrado ser altamente efectivo para la clasificación de instrumentos musicales tradicionales. En el trabajo citado [7], se propone un modelo basado en redes neuronales convolucionales que utiliza estas representaciones para identificar instrumentos de Asia. Este enfoque permite que el sistema aprenda patrones espectrales y armónicos propios de cada instrumento, logrando una precisión superior al 90 %. La estrategia es comparable con el presente trabajo, en el cual se emplean espectrogramas como entrada para modelos CNN aplicados a instrumentos andinos. Estudios recientes han demostrado la efectividad de modelos avanzados de Deep Learning para la clasificación de instrumentos musicales. En particular, se han comparado arquitecturas como **CNN**, **RNN**, **TDNN** y **Ecapa-TDNN**, utilizando MFCCs y espectrogramas como entradas [6]. La investigación reveló que los modelos basados en convoluciones unidimensionales, como TDNN y especialmente Ecapa-TDNN, obtienen mejores resultados que las redes recurrentes o convoluciones bidimensionales, debido a su capacidad para capturar información temporal secuencial en señales de audio.

Estos hallazgos respaldan la elección de arquitecturas basadas en convoluciones profundas y Espectrogramas para sistemas automáticos de clasificación instrumental, como el desarrollado en el presente trabajo.

A continuación, se presenta un esquema metodológico general adaptado desde el trabajo [6], que resume el flujo típico en la clasificación automática de sonidos de instrumentos musicales mediante aprendizaje automático:

Esta figura sintetiza de forma gráfica las etapas principales involucradas en la construcción de un modelo de clasificación musical basado en aprendizaje profundo, desde la carga de los datos hasta la predicción final. Aunque el presente trabajo introduce variaciones específicas en herramientas y arquitecturas (por ejemplo, uso de Django y frontend React), el flujo de procesamiento se ajusta a esta lógica secuencial que ha sido validada por la literatura especializada.

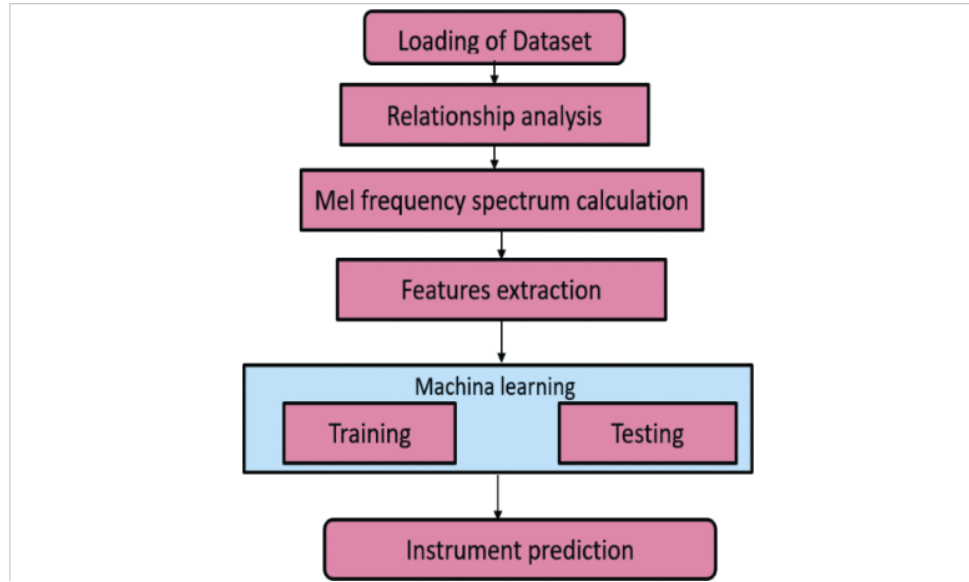


Figura 2: Flujo general del sistema de clasificación musical propuesto en [6], el cual inicia con la carga del dataset, extracción de características acústicas y clasificación mediante aprendizaje automático.

3.2. Estado del arte

3.2.1. Teoría general de sistemas

El uso de técnicas de Machine Learning y redes neuronales profundas ha revolucionado el campo del reconocimiento de audio. Estas tecnologías permiten el análisis detallado de grabaciones, identificando patrones que diferencian a los instrumentos musicales. En el contexto de la música andina, se han desarrollado algoritmos que pueden ser entrenados con conjuntos de datos que incluyen instrumentos como la quena, el charango o la zampoña, permitiendo detectar diferencias sutiles en sus timbres y modos de ejecución.

3.2.2. Algoritmos de Aprendizaje Automático

Diversos trabajos recientes han demostrado la eficacia de las redes neuronales convolucionales (CNN) combinadas con técnicas de extracción de características como los coeficientes cepstrales en la escala mel (MFCC). Por ejemplo, Mahanta et al. (2024) entrenaron un modelo CNN para clasificar 20 instrumentos musicales usando una arquitectura optimizada basada en MFCCs y técnicas como *batch normalization*, *dropout*, y capas densas con activación ReLU [10]. El modelo logró una precisión de última generación, utilizando un conjunto combinado de datos del Philharmonia Dataset y UIOWA MIS, garantizando así la independencia del ejecutante. En la Figura 3 se muestra el esquema de la arquitectura utilizada por los autores.

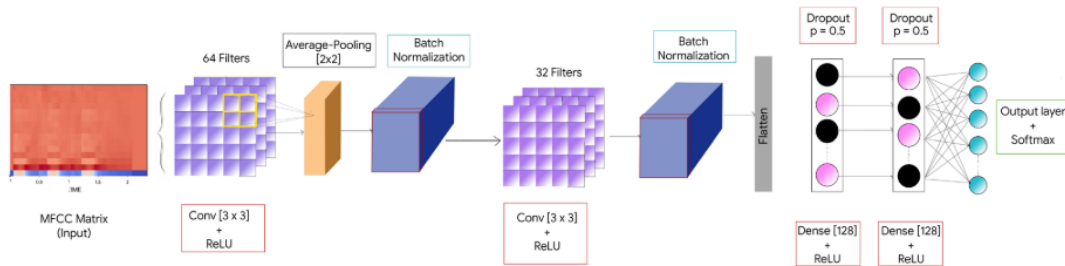


Figura 3: Arquitectura de red neuronal convolucional propuesta por Mahanta [10] para la clasificación de instrumentos musicales.

3.2.3. Análisis de Señales de Audio

El procesamiento digital de señales (DSP) permite la extracción de características relevantes de las grabaciones de audio, como el espectro de frecuencias, la envolvente temporal y las características melódicas. Entre estas, los MFCCs se han consolidado como una representación efectiva del contenido auditivo perceptual, siendo especialmente útiles en tareas de clasificación.

3.2.4. Big Data e Integración de Datos

Este enfoque implica la gestión de grandes volúmenes de datos procedentes de múltiples fuentes. En tareas de clasificación musical, contar con datos diversos y balanceados es esencial para lograr modelos generalizables y precisos.

3.3. Modelos Personalizados - Planteamiento

El desarrollo de modelos adaptados a criterios específicos permite una clasificación más precisa, especialmente en contextos culturales particulares como la música tradicional andina. Esto incluye ajustes finos en los parámetros del modelo y en la selección de características, en función de los instrumentos analizados. .

3.4. Modelos Personalizados

El desarrollo de modelos que se ajustan a criterios específicos facilita un enfoque más dirigido y detallado, lo que potencialmente conduce a resultados más precisos y ajustados a las necesidades particulares del análisis dependiendo del instrumento a identificar.

3.5. Antecedentes

El proyecto propuesto se enmarca en esta perspectiva innovadora, proponiendo el desarrollo de sistemas para la identificación de instrumentos musicales de la región andina

mediante técnicas de Machine Learning. La identificación automática de instrumentos musicales andinos permitirá una clasificación y reconocimiento más eficiente de los contenidos de archivos de audio, simplificando tareas como la indexación, la edición y la recuperación de datos [4]. Se han realizado numerosas investigaciones sobre el reconocimiento del sonido de instrumentos musicales tradicionales andinos [11].

La influencia de la música andina es considerable en la actualidad, y con el avance de la tecnología, esperamos que aumente mientras que la de la música tradicional pueda disminuir. Por lo tanto, tenemos la intención de distinguir los diversos instrumentos tradicionales de las grabaciones musicales andinas específicas. Este estudio se ocupará de aplicaciones de procesamiento de señales de audio, identificación de instrumentos musicales tradicionales de las bandas sonoras andinas de las grabaciones. Se extraerán y compararán varias características de audio para la extracción de características con el fin de elegir un método apropiado para el reconocimiento de instrumentos musicales tradicionales.

3.6. Marco Conceptual

El marco conceptual de este proyecto es el desarrollo de una aplicación de software que emplee técnicas de aprendizaje automático para la identificación de instrumentos musicales de la región andina.

3.6.1. Sistema

1. Los sistemas deben tener un objetivo. La primera característica de un sistema es que tiene un objetivo claramente definido y articulado. Sin un objetivo, no hay sistema [12].
2. Un sistema debe ser gestionado. No se gestionará por sí mismo.
3. El secreto para tener un sistema exitoso es la cooperación entre los componentes.
4. La gestión de un sistema requiere conocimiento de las interrelaciones entre todos los componentes dentro del sistema y las personas que trabajan en él.
5. El objetivo ideal es crear una "mejor vida para todos". Cuando se están deliberando sobre temas, el objetivo debe ser encontrar una decisión que sea mejor para los estudiantes, empleados y la comunidad [13].

3.6.2. Software

Definición: Según la Real Academia Española, su definición es "Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora" [14].

Características: Según el artículo de Tuama [15], las características del software son:

Funcionalidad: Conjunto de capacidades que el software ofrece para permitir al usuario alcanzar sus objetivos de manera efectiva.

Usabilidad: Grado en que el software permite una interacción intuitiva y satisfactoria con el usuario, minimizando la curva de aprendizaje y maximizando la experiencia del usuario.

Eficiencia: Capacidad del software para utilizar los recursos del sistema de manera óptima, asegurando un rendimiento adecuado sin desperdiciar recursos computacionales.

Flexibilidad: Aptitud del software para adaptarse a cambios en los requisitos o condiciones del entorno, manteniendo su funcionalidad y estabilidad.

Fiabilidad: Propiedad del software de operar correctamente bajo diversas condiciones y entornos, garantizando resultados consistentes y precisos.

Sostenibilidad: Facilidad con la que el software puede ser modificado, actualizado y mejorado a lo largo de su ciclo de vida, asegurando su mantenimiento a largo plazo.

Portabilidad: Habilidad del software para ejecutarse en diferentes plataformas o sistemas operativos sin requerir cambios significativos.

Integridad: Característica que garantiza la precisión, coherencia y seguridad de los datos manejados por el software durante todo su ciclo de vida.

3.6.3. Machine Learning

Según IBM [16], el aprendizaje automático (ML) es una rama de la inteligencia artificial (IA) que se enfoca en el uso de datos y algoritmos para permitir que los sistemas imiten el proceso de aprendizaje humano, mejorando progresivamente su precisión y desempeño.

3.6.4. Espectrograma

Un **espectrograma** es una herramienta fundamental en el análisis de señales de audio. Se trata de una representación visual bidimensional que permite observar cómo varían las frecuencias presentes en una señal a lo largo del tiempo. Esta representación es especialmente útil para comprender la estructura de una señal compleja, como el sonido producido por un instrumento musical o una voz humana.

¿Cómo funciona un espectrograma? La generación de un espectrograma implica aplicar una técnica matemática conocida como *transformada de Fourier de tiempo corto* (STFT, por sus siglas en inglés). En términos simples, esta técnica divide la señal de audio en pequeños segmentos temporales y calcula las frecuencias presentes en cada uno de esos segmentos. El resultado es una matriz donde cada punto representa la magnitud (o energía) de una frecuencia específica en un momento dado. Esta matriz se visualiza utilizando un mapa de colores, donde los tonos cálidos (como el naranja o amarillo) indican una alta intensidad de energía, mientras que los tonos fríos (como el azul o violeta) indican una baja intensidad.

Ejes del espectrograma:

Eje x: Representa el tiempo. Este eje muestra cómo evoluciona la señal de audio a lo largo del tiempo. Cada punto en este eje corresponde a un segmento temporal de la señal.
Eje y: Representa la frecuencia. Este eje indica las diferentes componentes espectrales presentes en la señal. Las frecuencias bajas (como las de un bajo musical) aparecen en la parte inferior, mientras que las frecuencias altas (como las de un violín o un silbato) aparecen en la parte superior.
Colores: Los colores en el espectrograma representan la

magnitud de la energía en cada frecuencia. Por ejemplo: Tonos cálidos (naranja, amarillo): Indican una alta intensidad de energía en esa frecuencia y en ese momento. Tonos fríos (azul, violeta): Indican una baja intensidad de energía. **Aplicaciones del espectrograma:** Los espectrogramas son ampliamente utilizados en diversas áreas, incluyendo:

Procesamiento de audio: Para analizar la calidad de grabaciones, identificar ruidos no deseados o estudiar las características de un instrumento musical. **Reconocimiento de voz:** Para extraer patrones de frecuencia que permitan identificar fonemas o palabras en una señal de voz. **Investigación científica:** En campos como la bioacústica, donde se estudian los sonidos emitidos por animales (por ejemplo, aves o cetáceos), o en la sismología, donde se analizan ondas sísmicas. **Música y composición:** Para visualizar cómo se combinan diferentes frecuencias en una pieza musical y optimizar la mezcla de sonidos. Además, los espectrogramas son útiles para identificar fenómenos específicos en las señales de audio. Por ejemplo:

Armónicos: Son frecuencias múltiplos enteros de una frecuencia fundamental. En un espectrograma, los armónicos aparecen como líneas horizontales paralelas a la frecuencia fundamental. **Ruido:** Se manifiesta como una distribución aleatoria de energía en todo el rango de frecuencias. **Distorsiones:** Pueden identificarse como picos inusuales o cambios abruptos en la distribución de energía.

Ejemplo visual simplificado:

Para ilustrar estos conceptos, se puede observar un espectrograma generado a partir de una grabación de quena:

La Figura 4 presenta un espectrograma generado a partir de un fragmento de audio correspondiente a una interpretación con ocarina. Este espectrograma permite observar de forma clara las siguientes características relevantes:

- **Armónicos estables:** Se evidencian líneas horizontales brillantes y paralelas, principalmente entre los 500 Hz y 2000 Hz, lo cual es típico de instrumentos melódicos con timbre limpio como la ocarina.
- **Frecuencias fundamentales:** La banda amarilla más intensa en la parte inferior del espectro (alrededor de los 500 Hz) corresponde a la frecuencia fundamental de las notas ejecutadas.
- **Transiciones:** Entre los segundos 1 y 2.5 pueden observarse cambios en la energía y altura tonal, lo cual indica variaciones de notas durante la ejecución.
- **Bajo nivel de ruido:** La mayoría del fondo espectral permanece en tonos fríos, lo cual sugiere que la grabación posee poco ruido ambiental y buena calidad de señal.

Este tipo de representación es fundamental para el análisis automático de señales musicales, ya que permite a los modelos de aprendizaje automático como las redes neuronales convolucionales (CNN) identificar patrones estructurales en el dominio espectral, clave para la clasificación precisa de instrumentos.

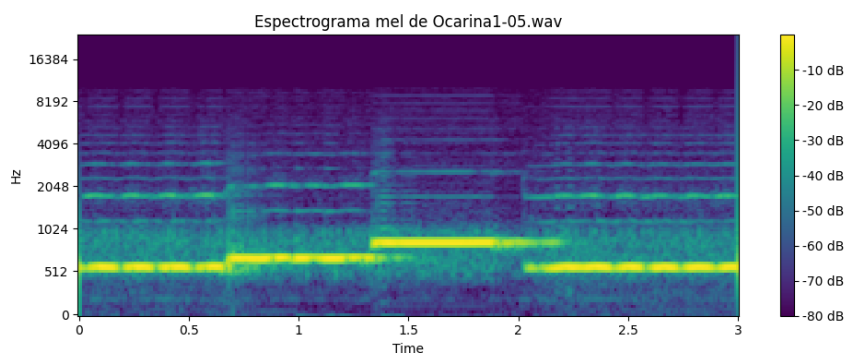


Figura 4: Espectrograma de una quena: se observan armónicos y regiones de ruido.

4. Alcances de la propuesta

4.1. Definicion

Este proyecto tiene como objetivo desarrollar un sistema de identificación de instrumentos musicales en la región andina utilizando técnicas de aprendizaje automático. El sistema utiliza algoritmos avanzados para clasificar e identificar con precisión varios instrumentos tradicionales únicos de esta región. Algunos de los instrumentos de música andina que se van a introducir en este programa de identificación incluyen:

- Instrumentos de viento
 - Quena: Una flauta tradicional hecha de caña o madera, con un sonido distintivo.
 - Sikus (o zampoña): Un conjunto de flautas de pan hechas de caña, con tubos de diferentes longitudes.
 - Toyos (o zampoña grande): Un gran conjunto de flautas de madera con un tubo de gran longitud y un sonido grave, una aparecña parecida a la Zampoña pero de una longitud de aproximadamente un metro en su parte mas extensa.
 - Ocarina: Una ocarina es un instrumento de viento que produce sonido por la vibración del aire. A diferencia de las flautas, las ocarinas no son tubos sino que son cámaras ovaladas con orificios para la entrada y salida del aire [17].
- Instrumentos de cuerda
 - Charango: Un pequeño instrumento de cuerda, similar a una guitarra, con diez cuerdas, muy característica de la region andina.
 - Guitarra: Una variante de la guitarra española adaptada a la música andina.
 - Bandola: Un instrumento similar a la mandolina, común en la música de los Andes.
 - Tiple: El tiple es un pequeño instrumento de cuerda conocido por su sonido agudo y melodioso. Se utiliza comúnmente en la música folklórica colombiana, especialemnte en la region andina [18].
- Instrumentos de percusión
 - Bombo andino: Un gran tambor de doble parche que se toca con palos.
 - Cajón peruano: Una caja de resonancia de madera que se toca con las manos, originaria de Perú.
 - Chajchas: Instrumento de percusión tradicional de la región andina, consistente en una serie de chapas o cascabeles unidos a una base, generalmente hechos de metal o semillas secas. Se utilizan comúnmente en ceremonias y danzas tradicionales para acompañar el ritmo de la música andina.

- Guacharaca: Un instrumento de percusión idófono de raspado utilizado en la música de la región andina. La variedad de su sonido depende principalmente del material de fabricación del instrumento, siendo este hecho de madera o acero [19].

4.2. Objetivos

Desarrollar un sistema de aprendizaje automático que identifique y clasifique los instrumentos musicales de la región andina en audio, capaz de reconocer y categorizar con precisión estos instrumentos. Los instrumentos que se identificarán incluyen una variedad de instrumentos de viento, cuerda y percusión característicos de esta región.

4.2.1. Objetivos específicos

1. Recopilar un conjunto de datos de muestras de audio que representen una amplia gama de instrumentos musicales andinos.
2. Preprocesar los datos de audio para extraer características relevantes que se puedan utilizar para entrenar modelos de aprendizaje automático.
3. Implementar y optimizar algoritmos de aprendizaje automático para la clasificación de instrumentos.
4. Desarrollar un prototipo que permita a los usuarios interactuar con el sistema y explorar información sobre los instrumentos identificados en la región andina.
5. Evaluar el rendimiento del sistema de identificación a través de procedimientos de prueba y validación.

4.2.2. Entregables y Objetivos-Resultado esperado

Tener un prototipo del sistema de software que sea funcional y que contenga las características de organización y reconocimiento de los diferentes instrumentos musicales empleados en la música andina. Se visualiza en la Tabla 1 los entregables y objetivos esperados.

4.2.3. Marco lógico de la propuesta

En la Tabla 3 se pueden ver el marco lógico de la propuesta

Objetivos específicos de los requerimientos e historias de usuario	Entregables
Recopilar un conjunto de datos completo de muestras de audio que representen una amplia gama de instrumentos musicales andinos	Datos en brutos de los sonidos de los instrumentos musicales
Preprocesar los datos de audio para extraer características relevantes que se puedan utilizar para entrenar modelos de aprendizaje automático	Un conjunto de datos procesados que contenga características relevantes extraídas de los archivos de audio, listo para ser utilizado en el entrenamiento de modelos de aprendizaje automático
Implementar algoritmos de aprendizaje automático como máquinas de vectores de soporte para la clasificación de instrumentos	Máquinas de vectores de soporte (SVM) entrenados y optimizados específicamente para clasificar instrumentos musicales a partir de datos de audio
Evaluar el rendimiento del sistema de identificación a través de rigurosos procedimientos de prueba y validación	Informe incluyendo métricas de rendimiento como precisión, sensibilidad y especificidad
Desarrollar un prototipo que permita a los usuarios interactuar con el sistema y explorar información sobre los instrumentos identificados en la región andina	Un prototipo de interfaz de usuario sobre las características de los instrumentos evaluados

Tabla 1: Tabla de Objetivos y Entregables

Objetivo específico	Resultados esperados	Evidencia del Objetivo
Recopilar un conjunto de datos completo de muestras de audio que representen una amplia gama de instrumentos musicales andinos.	▪ Registro de datos en bruto de diferentes instrumentos musicales andinos, tanto de fuentes externas como mediante grabaciones propias. ▪ Documentación sobre la organización del dataset, herramientas empleadas y consideraciones éticas en la adquisición de datos.	▪ Cap. 6: Recopilación de Datos en Bruto
Preprocesar los datos de audio para extraer características relevantes que se puedan utilizar para entrenar modelos de clasificación.	▪ Conjunto de datos estructurado y transformado mediante técnicas de procesamiento de señales como espectrogramas y extracción de características. ▪ Pipeline de preprocesamiento completo para ser reutilizado en los experimentos.	▪ Cap. 7: Preprocesamiento de Datos de Audio
Implementar modelos de inteligencia artificial, especialmente máquinas de vectores de soporte (SVM), para la clasificación de instrumentos musicales andinos.	▪ Entrenamiento de modelos SVM y modelos multiclase sobre el dataset procesado. ▪ Evaluación de los modelos mediante validación cruzada y métricas como precisión y matriz de confusión.	▪ Cap. 8: Implementación de Inteligencia Artificial para el Reconocimiento
Evaluar rigurosamente el rendimiento del sistema de clasificación utilizando pruebas y métricas formales.	▪ Documentación detallada de los resultados del sistema, incluyendo gráficos de desempeño, explicaciones y limitaciones. ▪ Análisis de generalización, escalabilidad y posibles mejoras futuras del modelo.	▪ Cap. 8.7–8.15: Evaluación, Visualización y Representación
Desarrollar un prototipo funcional que permita al usuario cargar archivos de audio y visualizar los resultados de clasificación de instrumentos.	▪ Interfaz gráfica que permita cargar archivos .wav y mostrar resultados de identificación de instrumentos. ▪ Arquitectura del sistema frontend-backend, integración del modelo y visualización de resultados.	▪ Cap. 9: Construcción del Prototipo: Frontend y Backend

Tabla 2: Relación entre los objetivos específicos, los resultados esperados y la evidencia en el documento

Descripción	Indicadores	Medios de verificación	Supuestos
Finalidad	Mejorar la precisión del sistema de reconocimiento de sonido hasta un 80 % de semejanza	Llegar al 80 % de precisión de reconocimiento en un plazo de 6 meses de entrenamiento	Informes de avance mensuales
Propósito	Desarrollar e implementar un aplicativo de software que permita reconocer los instrumentos musicales de la región andina	La precisión del software en la identificación por sonido es del 75 %	Reportes de precisión del sistema
Componentes	Diseño del software, implementación y pruebas	Código del software, resultados de las pruebas	Revisión del código, informes de pruebas, informes de rendimiento
Actividad	Análisis de requerimientos, diseño, integración, pruebas, diseño opcional de interfaz gráfica	Documento de requerimientos, documento de diseño, informe de pruebas	Disponibilidad de tiempo y recursos para análisis y diseño, disponibilidad de entornos de desarrollo

Tabla 3: Matriz de marco lógico

5. Recopilación de Datos en Bruto

La recopilación de datos en bruto es un paso fundamental en el desarrollo de sistemas basados en aprendizaje automático, ya que la calidad y representatividad de los datos determinan en gran medida el rendimiento del modelo final. En este trabajo, el objetivo fue recopilar un conjunto de datos diverso y representativo de sonidos de instrumentos andinos, considerando las características únicas de estos instrumentos y las limitaciones asociadas a su disponibilidad en fuentes digitales. A continuación, se describe el proceso detallado de recopilación, organización y validación de los datos.

5.1. Fuentes de Datos

Dado que no existen datasets extensos y específicos de instrumentos andinos, se implementaron múltiples estrategias para garantizar una recopilación de datos diversa y de alta calidad:

- **YouTube:** Se analizaron videos musicales en los que los instrumentos andinos sonaban de manera aislada, sin acompañamiento de otros instrumentos o voces. Se seleccionaron audios claros y de buena calidad por medio del criterio del autor.
- **Showcases de Instrumentos:** Se utilizaron grabaciones de exhibiciones de instrumentos en eventos musicales y muestras individuales realizadas por músicos.
- **Bases de Datos Públicas (Kaggle):** Se revisaron datasets disponibles en plataformas como Kaggle, aunque la mayoría estaban centrados en instrumentos occidentales. Esto llevó a explorar otras fuentes complementarias.
- **Colaboración con la Universidad :** Se contactó a un profesor de música quien facilitó grabaciones directas de instrumentos realizadas en entornos controlados con dispositivos móviles y micrófonos portátiles.
- **Revisión de Datasets de Audio Existentes:** Se analizaron bases de datos en línea para evaluar qué características acústicas podrían extraerse y utilizarse en el proyecto.

5.2. Proceso de Extracción y Preprocesamiento de Audio

Para asegurar que los datos en bruto fueran adecuados para su posterior análisis, se implementó un pipeline estructurado de extracción y preprocesamiento:

1. **Conversión a Formato WAV:** Todos los archivos de audio fueron convertidos al formato WAV, que permite conservar la calidad y las características espectrales originales.
2. **Eliminación de Interferencias:** Se descartaron audios en los que más de un instrumento sonaba simultáneamente, evitando interferencias que pudieran complicar la clasificación futura.

3. **Normalización de Niveles de Volumen:** Se ajustaron los niveles de volumen para minimizar variaciones drásticas entre grabaciones, es decir, audios con sonido extremadamente alto o bajo se debieron o bien en algunos casos descartar o bien gracias a la herramienta Audacity tratar de normalizarlos.
4. **Segmentación de Audios:** Los audios fueron divididos en fragmentos de entre 3 y 5 segundos para facilitar su procesamiento y etiquetado.

5.3. Grabación Directa de Instrumentos Andinos

En colaboración con músicos y la Universidad, se realizaron grabaciones directas de instrumentos andinos en entornos controlados:

- **Entorno Controlado:** Las grabaciones se realizaron en espacios con bajo nivel de ruido ambiental para minimizar contaminación externa.
- **Múltiples Tomas por Instrumento:** Se realizaron varias tomas por instrumento para capturar diferentes matices sonoros y técnicas interpretativas.
- **Organización de Archivos:** Las grabaciones se almacenaron en una estructura jerárquica de carpetas, organizadas por categoría de instrumento (viento, cuerda, percusión).

5.4. Herramientas Utilizadas

Para manejar y optimizar los datos en bruto, se utilizaron diversas herramientas de software especializadas:

- **Audacity:** Para limpiar los audios, recortar segmentos relevantes y aplicar filtros de reducción de ruido.
- **Librosa (Python):** Para convertir los audios en representaciones espectrales y verificar la integridad de las grabaciones.
- **Jupyter Notebooks:** Para realizar un análisis espectral inicial y evaluar la distribución de frecuencias de los instrumentos.

5.5. Organización del Dataset

Los audios en bruto se organizaron en categorías según el tipo de instrumento, asegurando una estructura clara y accesible:

- **Viento:** Quena, Zampoña, Toyo, Ocarina.
- **Cuerda:** Charango, Guitarra, Bandola, Tiple.
- **Percusión:** Bombo Andino, Cajón Peruano, Wankara, Guacharaca.

5.6. Descripción del Conjunto de Datos

Cada categoría incluye archivos de audio en formato WAV, acompañados de metadatos relevantes, tales como:

- Ubicación geográfica de la grabación.
- Nombre del intérprete.
- Detalles sobre el entorno de grabación.

Para acceder a los múltiples audios cortos recolectados, puede utilizar el siguiente enlace:

Repositorio de Audios

5.7. Validación de los Datos

Antes de proceder con el análisis, se realizó una validación rigurosa del dataset para garantizar su calidad:

- **Revisión Manual:** Se reprodujeron manualmente los audios para detectar problemas de ruido, interferencias o mala calidad de grabación.
- **Análisis Espectral:** Se generaron espectrogramas para verificar que las frecuencias fueran consistentes con el sonido característico de cada instrumento.

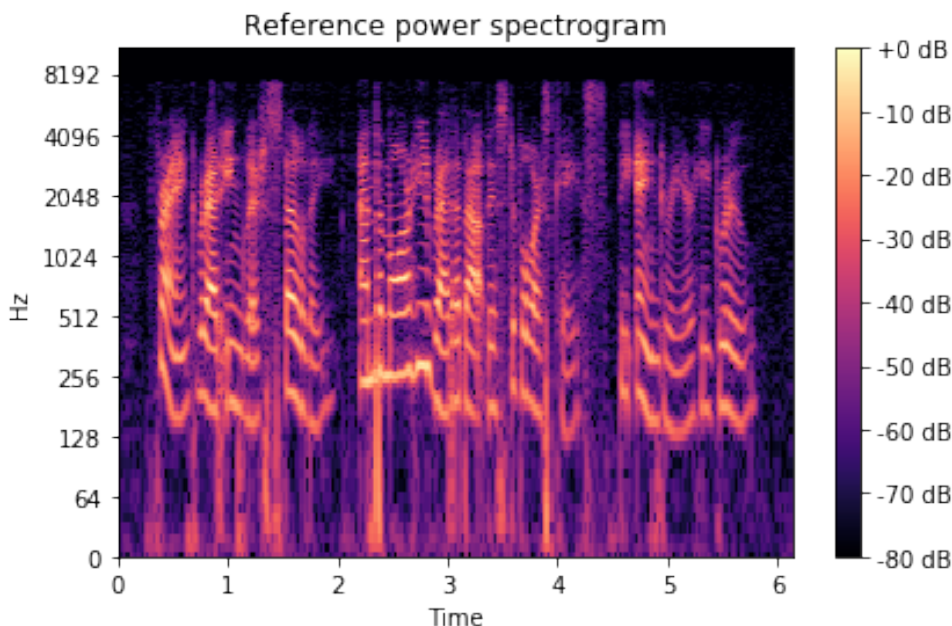


Figura 5: Espectrograma representativo del análisis realizado.

5.8. Análisis del espectrograma de potencia

La figura muestra un **espectrograma de potencia de referencia**, una representación visual de cómo varía el contenido espectral de una señal de audio a lo largo del

tiempo. En el eje horizontal se representa el **tiempo** en segundos, mientras que en el eje vertical se muestra la **frecuencia** en Hertz (Hz), siguiendo una escala logarítmica para cubrir un amplio rango de frecuencias.

Los colores indican la **intensidad de la energía** en cada punto del espectro. La escala de color a la derecha está en decibelios (dB), desde +0 dB (blanco/amarillo, mayor intensidad) hasta -80 dB (negro, menor intensidad). Las regiones más claras representan componentes frecuenciales con mayor energía, mientras que las más oscuras indican menor presencia energética.

Este tipo de análisis es útil para identificar patrones acústicos, armónicos y cambios en el timbre o dinámica de la señal. En este caso, se observan estructuras periódicas y armónicas claramente definidas, lo que sugiere la presencia de sonidos musicales o vocales con armónicos bien marcados. Las franjas verticales indican eventos acústicos breves o transitorios, mientras que las líneas horizontales revelan frecuencias sostenidas.

El espectrograma fue calculado usando una transformada de Fourier de corta duración (STFT), lo que permite observar cómo evoluciona el espectro de la señal a lo largo del tiempo, manteniendo un equilibrio entre resolución temporal y frecuencia.

Aplicaciones del espectrograma: Los espectrogramas son ampliamente utilizados en diversas áreas, incluyendo:

- **Procesamiento de audio:** Para analizar la calidad de grabaciones, identificar ruidos no deseados o estudiar las características de un instrumento musical.
- **Reconocimiento de voz:** Para extraer patrones de frecuencia que permitan identificar fonemas o palabras en una señal de voz.
- **Investigación científica:** En campos como la bioacústica, donde se estudian los sonidos emitidos por animales (por ejemplo, aves o cetáceos), o en la sismología, donde se analizan ondas sísmicas.
- **Música y composición:** Para visualizar cómo se combinan diferentes frecuencias en una pieza musical y optimizar la mezcla de sonidos.

Además, los espectrogramas son útiles para identificar fenómenos específicos en las señales de audio. Por ejemplo:

- **Armónicos:** Son frecuencias múltiplos enteros de una frecuencia fundamental. En un espectrograma, los armónicos aparecen como líneas horizontales paralelas a la frecuencia fundamental.
- **Ruido:** Se manifiesta como una distribución aleatoria de energía en todo el rango de frecuencias.
- **Distorsiones:** Pueden identificarse como picos inusuales o cambios abruptos en la distribución de energía.

En resumen, el espectrograma es una herramienta poderosa que permite visualizar y analizar las características de una señal de audio de manera intuitiva. Su capacidad

para mostrar tanto el tiempo como la frecuencia en una sola imagen lo convierte en un recurso invaluable en el procesamiento de señales y en la investigación científica. En este trabajo, los espectrogramas se utilizaron para validar la consistencia de las señales de audio y asegurar que las frecuencias registradas correspondieran a las características esperadas de cada instrumento.

- **Descarte de Grabaciones Defectuosas:** Se eliminaron audios con ruidos ambientales excesivos o distorsiones significativas.

5.9. Aumento de datos (Data Augmentation) en audio

El *data augmentation* es una técnica fundamental en el aprendizaje automático, especialmente cuando se trabaja con conjuntos de datos limitados. Consiste en generar nuevas muestras a partir de los datos originales, aplicando transformaciones que preservan la naturaleza del dato pero introducen variaciones útiles para mejorar la generalización del modelo.

En el contexto del audio, el aumento de datos tiene como objetivo crear múltiples versiones realistas de un mismo sonido, que representen posibles variaciones que podrían encontrarse en la vida real. Esto ayuda a reducir el sobreajuste y a mejorar la robustez del modelo frente a condiciones variables.

Entre las formas más comunes de aumento de datos en audio se encuentran:

- **Cambio de tono (Pitch Shifting):** altera la altura de la señal sin modificar su velocidad, simulando diferentes voces o instrumentos.
- **Cambio de velocidad (Time Stretching):** modifica la duración del audio sin afectar su tono, útil para simular distintas velocidades de ejecución.
- **Adición de ruido (Noise Injection):** se introduce ruido blanco o ruido de fondo para aumentar la robustez ante ambientes ruidosos.
- **Shifting temporal (Time Shifting):** se desplaza la señal en el tiempo, manteniendo su contenido, lo que ayuda a evitar que el modelo aprenda una posición fija de los eventos sonoros.
- **Filtros y distorsiones:** se aplican efectos como eco, reverberación o ecualización para simular diferentes entornos acústicos.
- **Mezcla de señales (Mixup):** se combinan dos o más señales de audio y sus etiquetas, promoviendo la capacidad del modelo de interpretar señales complejas o superpuestas.

Estas transformaciones se aplican generalmente de forma aleatoria o controlada durante el entrenamiento del modelo, permitiendo que éste aprenda a identificar patrones relevantes en situaciones más variadas. En conjunto, el *data augmentation* es una herramienta poderosa para mejorar la calidad y la generalización de modelos de aprendizaje profundo aplicados a datos de sonido, como reconocimiento de voz, clasificación de instrumentos, o detección de eventos acústicos.

5.10. Desafíos y Consideraciones

El proceso de recopilación de datos enfrentó varios desafíos que requirieron soluciones creativas:

- **Disponibilidad de Audios Limpios:** Muchos videos disponibles en plataformas como YouTube contenían ruido ambiental o acompañamiento instrumental, lo que dificultó la extracción de sonidos aislados.
- **Variabilidad en la Calidad de las Grabaciones:** Algunas grabaciones presentaban diferencias significativas en calidad debido a las condiciones del entorno de grabación.
- **Aspectos Legales:** Se revisaron cuidadosamente las licencias de uso de los audios extraídos de plataformas en línea para garantizar el cumplimiento normativo, lo cual hizo que en efecto los sonidos que tenemos sean de uso publico.

5.11. Resumen del Proceso

El proceso de recopilación de datos en bruto de sonidos de instrumentos andinos involucró múltiples estrategias, desde la búsqueda en repositorios en línea hasta grabaciones directas con músicos. La combinación de estas fuentes permitió reunir un conjunto de datos amplio, diverso y representativo, asegurando su calidad y preparándolo para el análisis posterior.

Dado el esfuerzo realizado en la recolección y curaduría de estos datos, se considera que este conjunto tiene el potencial de convertirse en un corpus abierto útil para la comunidad investigadora. Su publicación futura, con las debidas licencias, podría facilitar el desarrollo de nuevos sistemas de reconocimiento acústico y contribuir activamente a la preservación digital del patrimonio sonoro andino.

Fuente de Datos	Método de Obtención
YouTube	Extracción de audios de showcases de instrumentos andinos
Kaggle	Revisión de datasets existentes de instrumentos musicales
Universidad de Música	Grabaciones en bruto realizadas con dispositivos móviles
Datasets Online	Evaluación de características acústicas de otros conjuntos de datos

Tabla 4: Resumen de Métodos de Obtención de Datos en Bruto

Este proceso sentó las bases para el desarrollo de un sistema robusto de identificación automática de instrumentos andinos, asegurando que los datos utilizados fueran de alta calidad y representativos de las características únicas de estos instrumentos.

6. Preprocesamiento de Datos de Audio

6.1. Objetivo

El objetivo del preprocesamiento es transformar los archivos de audio en bruto en un conjunto de características representativas y consistentes que puedan ser utilizadas para entrenar modelos de aprendizaje automático. Este proceso garantiza que los datos sean óptimos para su análisis posterior, minimizando ruido e inconsistencias.

6.2. Entregable

Un conjunto de datos procesados que contenga características extraídas de los archivos de audio, listo para su uso en modelos de clasificación y aprendizaje profundo.

6.3. Planteamiento de la Metodología General del Sistema de Clasificación de Instrumentos Andinos

La siguiente figura describe el flujo de procesamiento utilizado en este proyecto, desde la captura de audio hasta la clasificación final por tipo de instrumento andino.

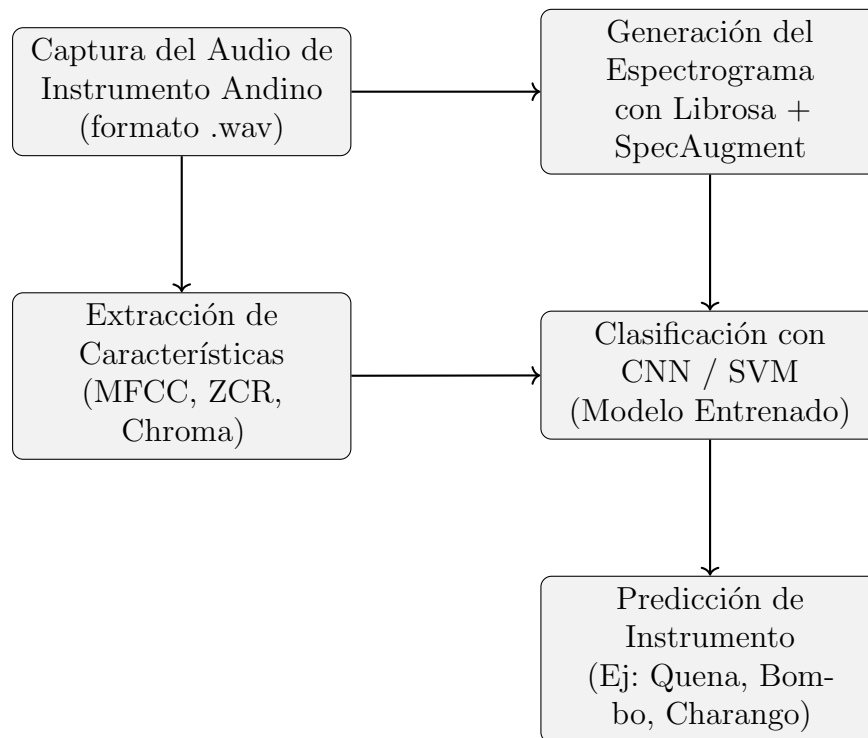


Figura 6: Metodología de clasificación de instrumentos andinos a partir de señales de audio.

Este pipeline ilustra cómo el sistema convierte señales acústicas en predicciones específicas mediante el uso de técnicas de procesamiento de audio y modelos de aprendizaje automático. La generación de espectrogramas permite a los modelos CNN identificar

patrones visuales complejos, mientras que las características numéricas como los MFCC enriquecen el proceso con una representación cuantitativa del sonido [20]. El sistema parte de la captura de audio, normalmente en formato `.wav`, y genera un espectrograma utilizando `Librosa` junto a técnicas de augmentación como `SpecAugment` para aumentar la robustez del modelo. En paralelo, se extraen características acústicas como MFCC, ZCR (tasa de cruce por cero), y cromagramas. Ambos conjuntos de datos —imagen espectral y vectores numéricos— son enviados a una arquitectura de clasificación basada en **Redes Neuronales Convolucionales (CNN)** o **Máquinas de Vectores de Soporte (SVM)**, que produce una predicción final del instrumento andino presente en la grabación.

La elección de estas dos técnicas responde a sus fortalezas complementarias: las **CNN** son altamente efectivas para procesar datos en forma de imagen, como los espectrogramas, capturando patrones espaciales complejos mediante convoluciones; mientras que las **SVM** han demostrado un excelente desempeño en tareas de clasificación con conjuntos de datos numéricos de dimensión moderada, como los MFCC. Esta combinación permite comparar enfoques visuales y acústicos, evaluando cuál ofrece mayor precisión y robustez para el reconocimiento de instrumentos musicales andinos.

6.3.1. Metodología General del Sistema

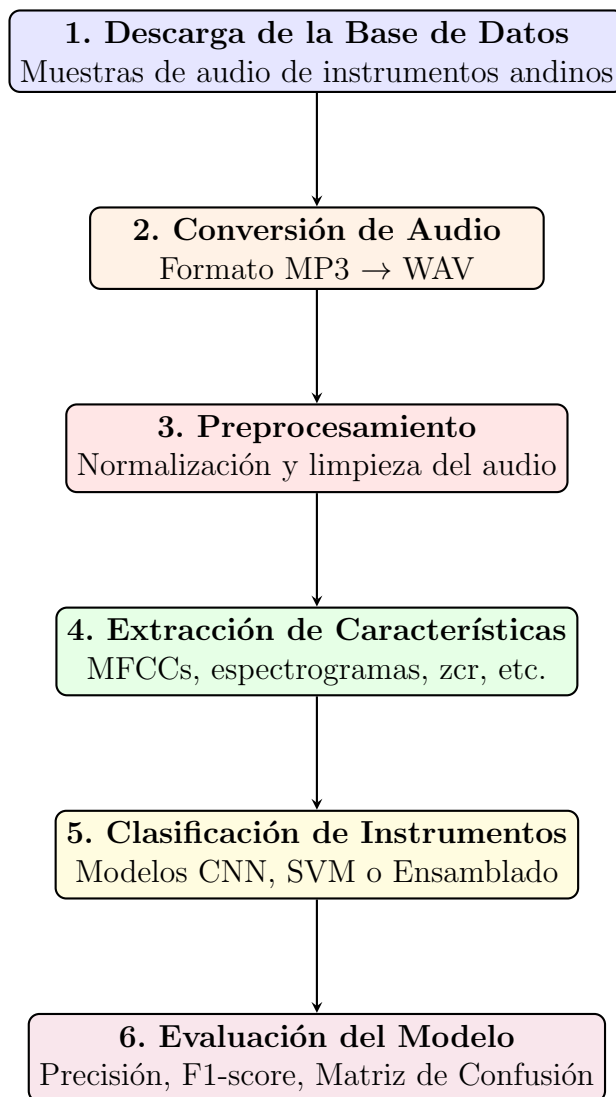


Figura 7: Flujo metodológico del sistema de clasificación de instrumentos andinos

El sistema desarrollado sigue un flujo metodológico que comienza con la recopilación de muestras de audio de instrumentos andinos, seguido por una conversión y normalización del formato para su procesamiento. Posteriormente, se extraen características acústicas relevantes (como MFCCs o espectrogramas), que luego se introducen a modelos de Machine Learning entrenados previamente. Finalmente, se evalúa el rendimiento del modelo mediante métricas como precisión, recall, F1-score y matriz de confusión, permitiendo analizar el desempeño de cada modelo de clasificación [21].

6.4. Parámetros de Preprocesamiento

Para asegurar la calidad y consistencia de los datos preprocesados, se definieron los siguientes parámetros:

Parámetro	Valor y Justificación
Tamaño de Ventana	6900 muestras, proporciona un balance adecuado entre resolución temporal y frecuencia. Visualizado con gráficos de ventanas en Librosa.
Método de Extracción de Características	MFCC (Coeficientes Cepstrales de Frecuencia Mel). Los MFCCs resultaron más efectivos para la clasificación de instrumentos.
Dimensión de los Espectrogramas	128x128, asegurando uniformidad en las entradas del modelo.
Normalización	Normalización para reducir la variabilidad en la intensidad del sonido (Previamente explicado).

6.5. Implementación de Técnicas de Procesamiento de Señales

6.5.1. Windowing

Se aplicó la técnica de **windowing**, que divide la señal en fragmentos temporales para facilitar el análisis espectral. Esta técnica mejora la precisión en la extracción de características al permitir un análisis local de la señal.

Ventajas del Windowing	Desafíos del Windowing
Permite el análisis de características locales de la señal, facilitando la identificación de patrones específicos.	Requiere un equilibrio cuidadoso entre la resolución temporal y frecuencial, lo que puede afectar la precisión del análisis.
Extrae características espectrales relevantes para la clasificación de instrumentos musicales.	La elección del tamaño de ventana es crítica, ya que ventanas demasiado pequeñas o grandes pueden distorsionar la información.
Mejora la resolución temporal del análisis, capturando cambios dinámicos en los sonidos de los instrumentos.	Aumenta la complejidad computacional, especialmente al procesar múltiples fragmentos de audio simultáneamente.

Tabla 6: Ventajas y Desafíos del Windowing en el Procesamiento de Audio

6.5.2. Generación de Espectrogramas

Se generaron espectrogramas utilizando la biblioteca **librosa** en Python. Estos espectrogramas representan la distribución de frecuencias del sonido a lo largo del tiempo, facilitando la identificación de patrones característicos de cada instrumento.

En la parte de data augmentation, poner cómo es antes y después en imágenes de espectrograma.

6.6. Pipeline de Preprocesamiento

Cada archivo de audio fue sometido a un proceso estructurado de transformación para extraer sus características más relevantes. El pipeline incluyó los siguientes pasos:

1. **Carga del Audio:** Se cargaron los archivos de audio con un formato .WAV.
2. **Transformación a Espectrogramas:** Se aplicó la Transformada de Fourier y la escala mel para convertir el audio en espectrogramas.
3. **Extracción de Coeficientes MFCC:** Se extrajeron coeficientes MFCC, complementados con **deltas** y **delta-deltas** para capturar la dinámica del sonido.
4. **Ajuste de Dimensiones:** Se ajustaron los espectrogramas a un tamaño fijo de 128x128 píxeles mediante padding o recorte.

6.7. Análisis y Manipulación de Datos

Se utilizaron herramientas especializadas para manipular y visualizar los datos preprocesados:

- **Pandas:** Para organizar y almacenar los datos preprocesados en estructuras eficientes.
- **NumPy:** Para realizar cálculos numéricos de los coeficientes extraídos.
- **Matplotlib:** Para visualizar espectrogramas y analizar la distribución de características.

6.8. Metodología de Extracción de Características

El preprocesamiento incluyó técnicas avanzadas para capturar la información más relevante de cada grabación:

- **Coeficientes MFCC:** Se extrajeron coeficientes MFCC por fragmento de audio, representando la envolvente espectral del sonido.
- **Espectrogramas Mel:** Se generaron espectrogramas en escala mel, destacando características tonales de los instrumentos.
- **Deltas y Delta-Deltas:** Se calcularon derivadas de los coeficientes MFCC para capturar cambios en la señal a lo largo del tiempo.

6.9. Optimización del Dataset

Para garantizar la calidad y confiabilidad del dataset, se implementaron diversas técnicas de optimización. A continuación, se describen las principales estrategias aplicadas:

- **Filtrado de Ruido:** Se realizó un proceso riguroso para eliminar grabaciones con alto nivel de ruido o interferencias externas que pudieran afectar la calidad del modelo. En total, se descartaron **algunos audios** debido a problemas como ruido ambiental excesivo, distorsión o errores técnicos durante la grabación. Este paso fue crucial para asegurar que el dataset estuviera compuesto únicamente por datos limpios y representativos.
- **Balance de Clases:** Con el objetivo de evitar sesgos en el modelo de aprendizaje automático, se equilibraron las cantidades de muestras por instrumento. Para lograrlo, se aplicaron técnicas de *data augmentation*, que incluyen modificaciones controladas de las grabaciones originales, como cambios de tono, velocidad y reverberación. Estas transformaciones permitieron aumentar la diversidad del dataset sin comprometer su calidad. Puedes encontrar más detalles sobre la implementación de estas técnicas en el repositorio oficial del proyecto:
 - Repositorio en GitHub: MusicMachineLearning-Backend.
- **Acceso al Dataset Optimizado:** El dataset optimizado, que incluye las grabaciones filtradas y balanceadas, está disponible en el siguiente enlace:
 - Repositorio en OneDrive: Repositorio de Audios.

Estas técnicas de optimización fueron fundamentales para mejorar la calidad del dataset y garantizar que el modelo de aprendizaje automático pueda entrenarse de manera efectiva.

6.10. Almacenamiento de los Datos Preprocesados

Los datos preprocesados fueron organizados en una estructura jerárquica eficiente para facilitar su uso en modelos de aprendizaje automático. A continuación, se describe la organización y los detalles clave:

- **Archivos Numéricos:** Las características extraídas de los audios se almacenaron en formato NumPy, lo cual optimiza el manejo y procesamiento de los datos. Estas matrices contienen información relevante como:
 - Coeficientes cepstrales de frecuencia de Mel (MFCC).
 - Espectrogramas generados a partir de transformaciones de Fourier.
 - Otras características acústicas relevantes.

Un ejemplo de cómo se ven estos archivos guardados se muestra en la Figura 10.

- **Carpetas por Instrumento:** Los datos fueron clasificados en carpetas separadas según la categoría del instrumento. Cada carpeta contiene los archivos correspondientes a un tipo específico de instrumento, junto con sus etiquetas asociadas. En las siguientes figuras se muestran ejemplos de estas estructuras jerárquicas:
 - Instrumentos de viento: Figura 8.
 - Instrumentos de percusión: Figura 9.
 - Instrumentos de cuerda: Figura 10.
- **Metadatos Asociados:** Para cada archivo, se incluyeron metadatos relevantes como:
 - Frecuencia de muestreo.
 - Duración de la grabación.
 - Técnica de interpretación utilizada.
 - Ubicación geográfica y entorno de grabación.

Estos metadatos permiten contextualizar los datos y mejorar la reproducibilidad del análisis.

Nombre	Estado
 Ocarina	 
 Quena	 
 Toyo	 
 Zampoña	 

Figura 8: Estructura de la carpeta correspondiente a instrumentos de viento.

6.11. Validación del Preprocesamiento

Para evaluar la efectividad del preprocesamiento, se realizaron pruebas con los datos extraídos:

- Se compararon los espectrogramas de diferentes instrumentos para asegurar que sus características fueran distinguibles.

Nombre	Estado
 Bombo	 
 Cajon	 
 Chajchas	 
 Guacharaca	 

Figura 9: Estructura de la carpeta correspondiente a instrumentos de percusión.

Nombre	Estado
 Bandola	 
 Charango	 
 Guitarra	 
 Requinto	 

Figura 10: Estructura de la carpeta correspondiente a instrumentos de cuerda.

- Se midió la variabilidad de los coeficientes MFCC entre diferentes interpretaciones del mismo instrumento.
- Se probó la compatibilidad de los datos procesados con modelos de aprendizaje automático en un entorno de prueba.

7. Implementación de Inteligencia Artificial para el Reconocimiento de Instrumentos Andinos

7.1. Objetivo

El objetivo de esta sección es describir la implementación de técnicas de inteligencia artificial para la clasificación de instrumentos musicales andinos a partir de datos de audio

procesados. Se exploran modelos de aprendizaje automático y aprendizaje profundo, junto con su proceso de entrenamiento, validación y evaluación.

7.2. Entregable

Un modelo de aprendizaje automático entrenado y validado que sea capaz de identificar correctamente diferentes instrumentos andinos a partir de sus características acústicas extraídas.

7.3. Parámetros de Entrenamiento del Modelo

Parámetro	Valor
Tasa de Aprendizaje	0.001, ajustado en Jupyter utilizando la técnica de Grid Search para optimización.
Número de Épocas	50 aunque puede variar dependiendo del tiempo, validado usando curvas de aprendizaje generadas con Matplotlib para evitar sobreentrenamiento.
Tamaño del Lote	32, basado en la capacidad del hardware para procesar datos de audio de alta resolución, evaluado mediante experimentos controlados en Jupyter.
Algoritmo de Optimización	Adam, seleccionado tras compararlo con otros optimizadores como SGD.
Función de Pérdida	<code>categorical_crossentropy</code> para modelos multiclase y <code>binary_crossentropy</code> para modelos multi-etiqueta.

La configuración de los hiperparámetros del modelo se definió a partir de una combinación de criterios experimentales y consideraciones prácticas. Para la tasa de aprendizaje, se empleó *Grid Search* con validación cruzada en Jupyter, evaluando el rendimiento en función de la convergencia. El número de épocas se ajustó con base en las curvas de aprendizaje para evitar el sobreentrenamiento, mientras que el tamaño del lote fue determinado según la capacidad del hardware disponible, priorizando un equilibrio entre velocidad y estabilidad. El algoritmo de optimización Adam se eligió tras comparar su desempeño con el de otros algoritmos como SGD. Finalmente, la función de pérdida fue seleccionada según la naturaleza del problema: clasificación multiclase o multi-etiqueta.

7.4. Modelos de Aprendizaje Automático

Se evaluaron diferentes técnicas de aprendizaje automático para la clasificación de instrumentos andinos. Entre las opciones exploradas se encuentran:

- **Redes Neuronales Convolucionales (CNN):** Modelo basado en la arquitectura de redes neuronales profundas, optimizado para el reconocimiento de patrones en imágenes de espectrogramas.

- **Máquinas de Vectores de Soporte (SVM):** Algoritmo de aprendizaje supervisado que busca encontrar el hiperplano óptimo para separar las clases de instrumentos.
- **Random Forest:** Modelo basado en árboles de decisión que mejora la precisión y robustez de la clasificación.

7.5. Diseño del Modelo de Red Neuronal Convolutacional

En esta sección, se describe el diseño y funcionamiento del modelo de red neuronal convolutacional (CNN) utilizado para la clasificación de instrumentos musicales. El modelo se divide en dos enfoques principales: una arquitectura base de CNN y un sistema ensamblado avanzado que combina múltiples CNN para mejorar la precisión.

7.5.1. Arquitectura Base de la CNN

La arquitectura base fue diseñada por el autor tomando como referencia estructuras CNN tradicionales utilizadas en clasificación de audio y estudios previos en tareas similares [20]. Durante la fase experimental se probaron múltiples configuraciones variando el número de capas convolucionales, el tamaño de los filtros y el uso de funciones de activación. La configuración final fue seleccionada tras evaluar su rendimiento en precisión y estabilidad, logrando un buen equilibrio entre complejidad computacional y capacidad de generalización. A continuación, se describe la versión final utilizada en el sistema:

La arquitectura base del modelo está diseñada para procesar espectrogramas normalizados y extraer características relevantes para la clasificación de instrumentos. A continuación, se detalla cada componente de la arquitectura:

1. **Capa de Entrada:** Los espectrogramas se utilizan como entrada al modelo. Estos tienen un tamaño fijo de $128 \times 128 \times 1$ píxeles, donde cada píxel representa la intensidad de una frecuencia en un momento específico.
2. **Capas Convolucionales:** Se implementan múltiples capas convolucionales con filtros de 3×3 y función de activación **ReLU**. Estas capas son responsables de detectar patrones espaciales y temporales en los espectrogramas, como transitorios iniciales o armónicos.
3. **Capas de Pooling:** Se aplican capas de max-pooling (2×2) después de las capas convolucionales para reducir la dimensionalidad de los datos mientras se conserva la información más relevante.
4. **Capas Fully Connected:** Las características extraídas por las capas convolucionales se combinan mediante capas densas (*fully connected*) con activación **ReLU**, lo que permite modelar relaciones complejas entre las características.
5. **Capa de Salida:** La capa final utiliza una función de activación específica dependiendo del tipo de problema:

- **Softmax** para problemas de clasificación multiclase (un instrumento por muestra).
- **Sigmoid** para problemas de clasificación multi-etiqueta (múltiples instrumentos por muestra).

7.5.2. Sistema Ensamblado de CNN

Para mejorar la precisión y robustez del modelo, se desarrolló un sistema ensamblado que combina las predicciones de múltiples redes neuronales convolucionales (CNN). Este enfoque es especialmente útil para distinguir instrumentos con características acústicas similares, como los instrumentos andinos.

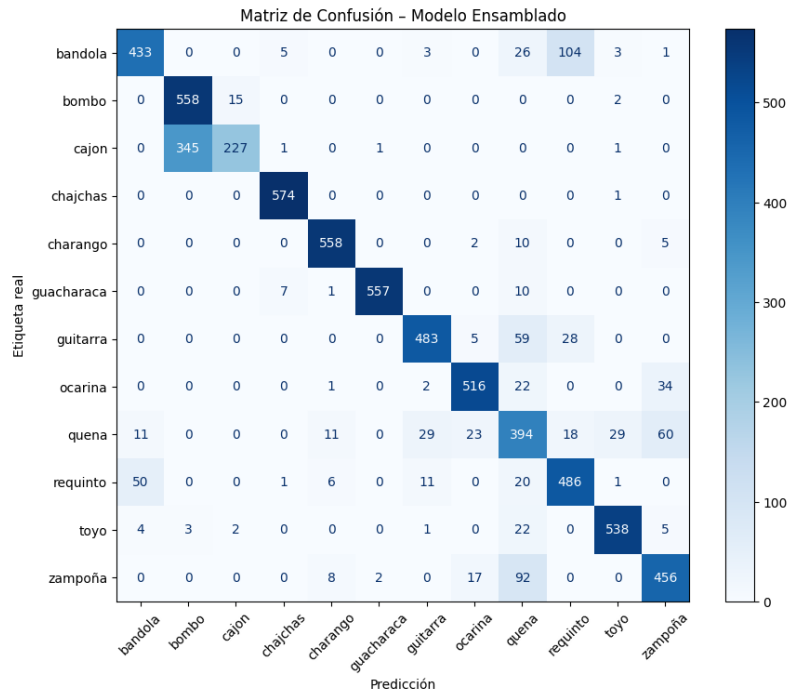


Figura 11: Matriz de confusión del modelo ensamblado CNN. Se observa un desempeño superior en la clasificación de instrumentos con características acústicas similares. Las zonas más oscuras representan mayor precisión en la predicción de cada clase.

7.5.3. Explicación del Sistema Ensamblado

La Figura 13 ilustra el funcionamiento del sistema ensamblado basado en múltiples CNN. A continuación, se describen los componentes clave:

- **Entrada (Espectrograma):** La señal de audio se convierte en un espectrograma, que actúa como entrada para tres redes neuronales independientes (CNN-1, CNN-2 y CNN-3).

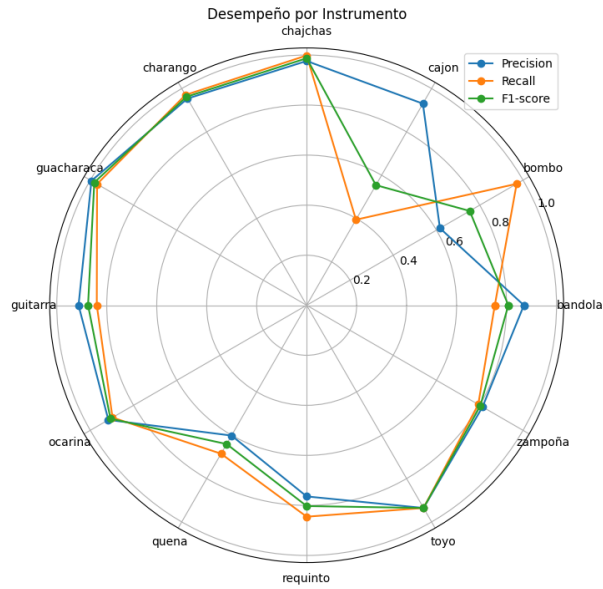


Figura 12: Grafico de como el modelo de ensamblado se comporta dependeiendo del instrumento, de forma circular se podria observar que tan lejos o cerca estuvo este metodo dependiendo del isnturmento

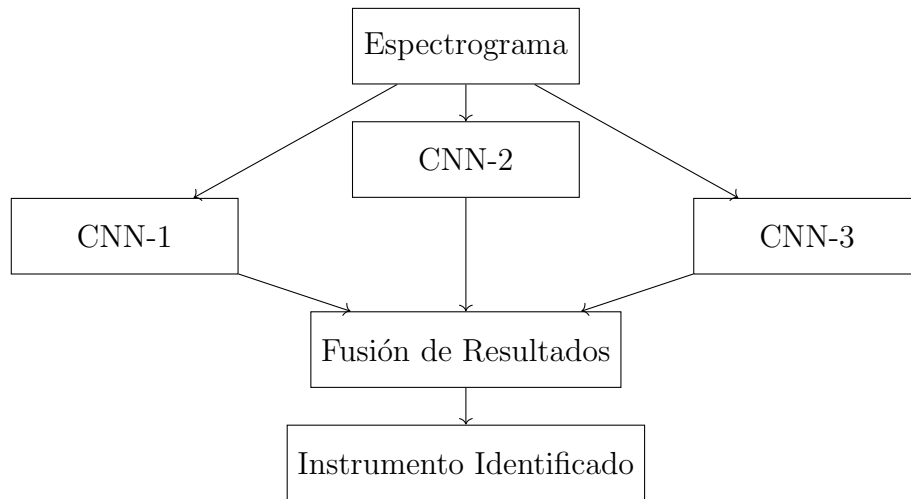


Figura 13: Diagrama del sistema ensamblado de CNN para la identificación de instrumentos.

- **CNN-1, CNN-2, CNN-3:** Cada red neuronal está especializada en analizar diferentes aspectos del espectrograma:
 - CNN-1 se enfoca en patrones temporales, como transitorios iniciales o ataques rápidos.
 - CNN-2 analiza patrones de frecuencia, como armónicos y tonos fundamentales.
 - CNN-3 detecta texturas espectrales complejas, como resonancias y modulaciones.
- **Fusión de Resultados:** Las predicciones de las tres CNN se combinan mediante

técnicas como promediado ponderado o votación mayoritaria. Esto reduce el riesgo de errores individuales y mejora la precisión general del sistema.

- **Salida:** El sistema selecciona el instrumento con mayor confianza. Por ejemplo, si dos CNN predicen "quenaz una predice "siku", el sistema seleccionará "quena como resultado final.

7.5.4. Aplicación en Instrumentos Andinos

Este enfoque es particularmente útil para identificar instrumentos con sonidos similares, como la quena y el siku, que comparten características acústicas como armónicos y timbres. La fusión de resultados asegura que el sistema tome decisiones más informadas y precisas, minimizando errores de clasificación.

7.6. Entrenamiento del Modelo

El entrenamiento de la CNN se realizó utilizando los espectrogramas generados en la etapa de preprocesamiento de datos. El proceso siguió los siguientes pasos:

1. Se dividió el dataset en 80 % entrenamiento y 20 % validación.
2. Se aplicó `data augmentation` para aumentar la diversidad del conjunto de datos.
3. Se utilizó `EarlyStopping` para detener el entrenamiento si la pérdida en validación dejaba de mejorar.
4. Se monitorizó la precisión y pérdida en cada época para evaluar el desempeño del modelo.

7.7. Evaluación y Validación del Modelo

Para garantizar el buen desempeño del modelo, se aplicaron diversas técnicas de validación y evaluación:

7.7.1. Métricas de Evaluación

Se utilizaron las siguientes métricas para medir la efectividad del modelo:

- **Precisión (Accuracy):** Proporción de predicciones correctas sobre el total de predicciones.
- **Exactitud (Precision):** Medida de qué tan precisas son las predicciones positivas del modelo.
- **Recall (Sensibilidad):** Capacidad del modelo para identificar correctamente instancias de una clase determinada.
- **F1-score:** Media armónica de precisión y recall, proporcionando un balance entre ambas métricas.

7.7.2. Validación Cruzada

Se implementó una validación cruzada de 5 pliegues para evaluar la estabilidad del modelo. Esto permitió medir su capacidad de generalización antes de su implementación final.

7.7.3. Pruebas de Clasificación

Se realizaron pruebas con nuevos audios para evaluar la capacidad del modelo de clasificar correctamente los instrumentos:

- Se probó el modelo con fragmentos de guitarra y bombo, verificando que las predicciones fueran correctas.
- Se probaron archivos de combinación de instrumentos para evaluar el rendimiento en escenarios multi-etiqueta.
- Se generaron matrices de confusión para visualizar la distribución de errores en la clasificación.

7.8. Estrategias de Optimización del Modelo

Para mejorar el rendimiento del modelo y su capacidad de generalización, se adoptaron varias técnicas de optimización ampliamente utilizadas en aprendizaje profundo:

- **Regularización L2:** Incorporada en capas densas para mitigar el sobreajuste penalizando grandes pesos.
- **Dropout:** Se aplicó una tasa de desactivación del 30 % en capas totalmente conectadas, lo que contribuyó a reducir la dependencia excesiva de ciertas neuronas.
- **Normalización por Lotes (Batch Normalization):** Se utilizó para estabilizar las distribuciones de activación intermedias y acelerar el proceso de entrenamiento.

7.9. Evaluación y Análisis del Modelo

El modelo fue evaluado utilizando métricas estándar como precisión, exactitud, *recall* y *F1-score*, calculadas a partir de un conjunto de prueba independiente. Estas métricas proporcionan una visión integral del rendimiento del sistema en tareas de clasificación multiclase. Los resultados obtenidos reflejan una alta capacidad discriminativa del modelo, especialmente en clases bien representadas en el conjunto de entrenamiento.

Métrica	Resultado Aproximado
Precisión General	Alta
F1-score Promedio	Consistente
Recall	Elevado
Exactitud	Satisfactoria

Tabla 8: Evaluación Cualitativa del Modelo

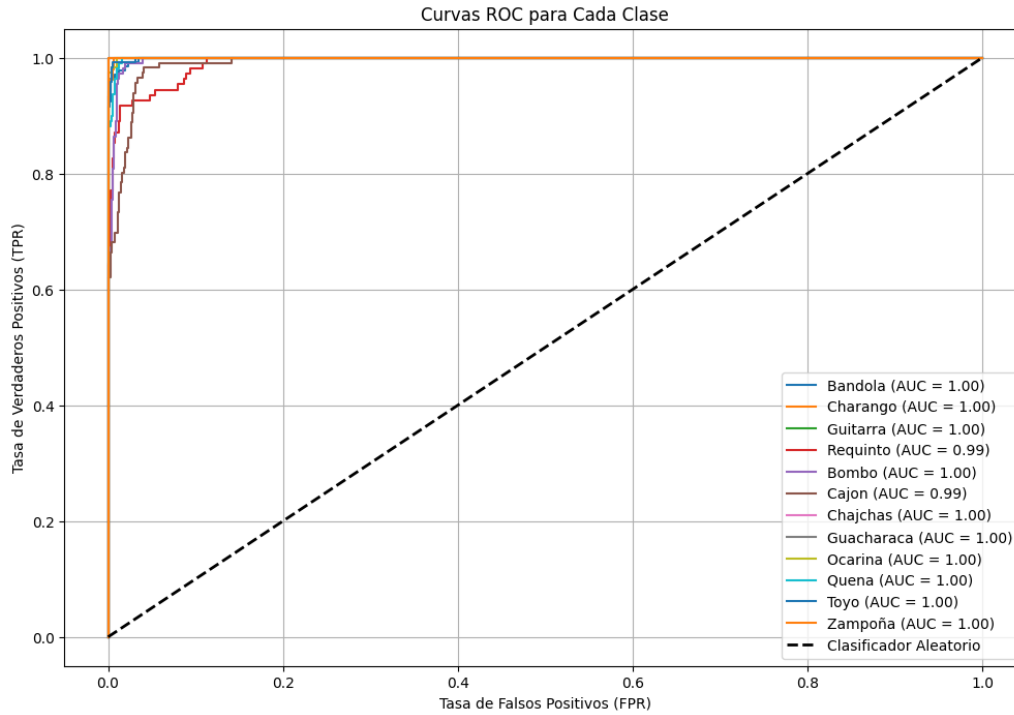


Figura 14: Curva ROC multiclase generada para evaluar el rendimiento del modelo , en este caso, el de multiclase

7.10. Escalabilidad y Generalización

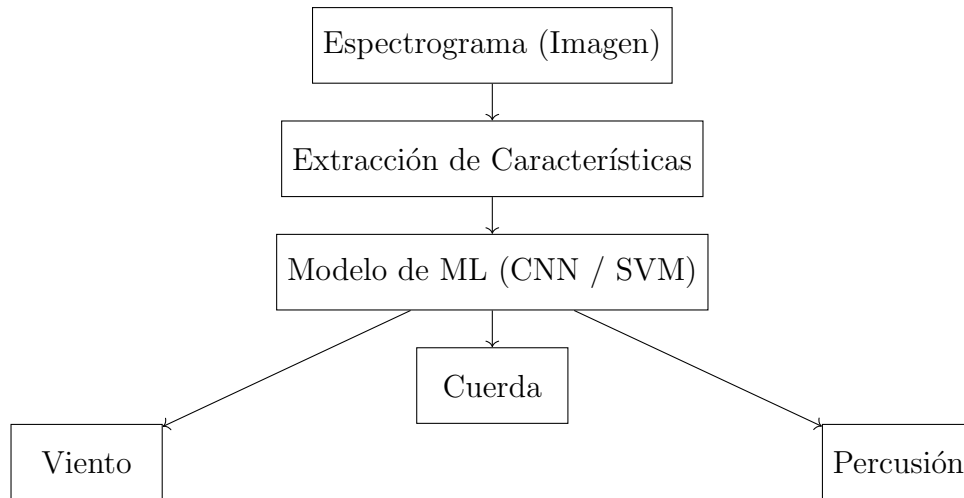
Con miras a ampliar el sistema hacia nuevos instrumentos andinos, se definieron estrategias clave de escalabilidad:

- **Expansión del dataset:** Incorporación progresiva de nuevas clases instrumentales en futuras fases de reentrenamiento.
- **Aprendizaje Transferido:** Uso de modelos preentrenados para acelerar el entrenamiento y mejorar la capacidad de generalización.
- **Ajuste fino (*Fine-tuning*):** Ajuste controlado de capas superiores del modelo para adaptarse a características acústicas específicas de nuevos instrumentos.

7.11. Clasificación Multiclase de Instrumentos Andinos

El núcleo del sistema de identificación se basa en una arquitectura de clasificación multiclase que procesa espectrogramas derivados de grabaciones de audio. Mediante redes neuronales convolucionales (CNN), el modelo estima la probabilidad de pertenencia a cada clase instrumental (por ejemplo: viento, cuerda o percusión). Esta técnica ha demostrado ser efectiva para capturar las características acústicas distintivas de los instrumentos tradicionales andinos. .

7.11.1. Diagrama del Proceso de Clasificación



7.11.2. Explicación del Proceso

- **Entrada - Espectrograma:** El audio cargado por el usuario es transformado en un espectrograma utilizando `librosa`. Este espectrograma representa la intensidad de las frecuencias a lo largo del tiempo.
- **Extracción de Características:** A partir del espectrograma, se extraen características relevantes como MFCCs (coeficientes cepstrales), patrones armónicos y transitorios que son fundamentales para distinguir instrumentos.
- **Modelo Multiclase:** Se utiliza un modelo CNN previamente entrenado, o alternativamente un SVM o modelo ensamblado, que recibe estas características y realiza una predicción multiclase con probabilidades asociadas.
- **Salida - Categoría de Instrumento:** El modelo emite una clasificación probabilística, donde se selecciona la clase con mayor puntuación como predicción principal.

7.11.3. Aplicación en Instrumentos Tradicionales

Este sistema ha sido entrenado específicamente con muestras de instrumentos tradicionales de la región andina, como:

- **Viento:** Quena, Zampoña (Sikus), Ocarina, Toyos.
- **Cuerda:** Charango, Bandola, Guitarra Andina, Tiple.
- **Percusión:** Bombo, Wankara, Cajón Peruano, Guacharaca.

Gracias a este enfoque, es posible identificar instrumentos con una alta precisión incluso en ambientes ruidosos o con múltiples fuentes sonoras.

7.11.4. Matriz de Confusión del Modelo Multiclase

A continuación se presenta una matriz de confusión obtenida durante la validación del modelo CNN multiclase. Esta matriz permite observar el rendimiento del modelo al clasificar correctamente o incorrectamente cada tipo de instrumento.

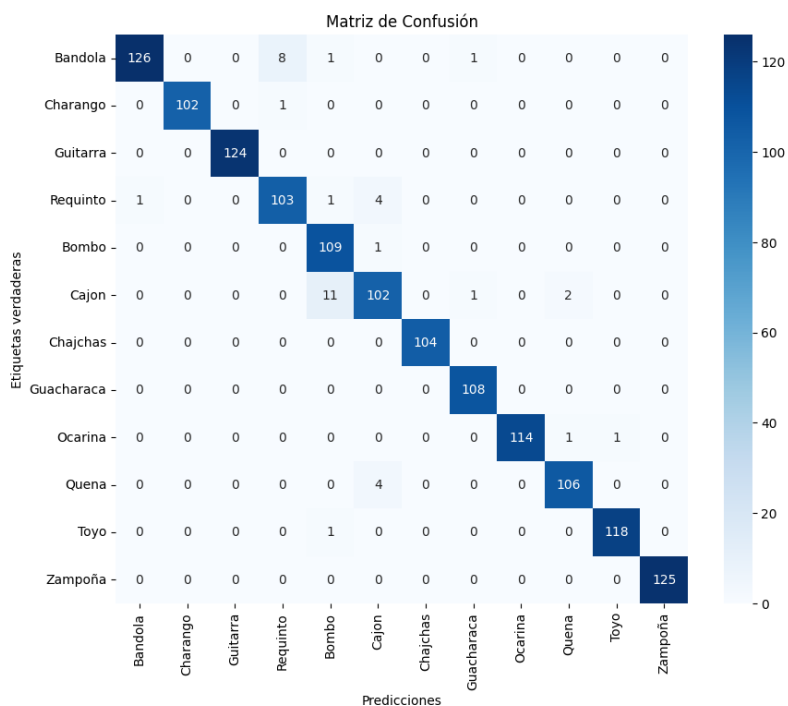


Figura 15: Matriz de Confusión del Modelo Multiclase (CNN)

La diagonal representa las predicciones correctas (e.g., quena → quena), mientras que los valores fuera de la diagonal representan errores de clasificación (e.g., charango → bandola). En general, los instrumentos de viento presentan mayor precisión debido a sus patrones acústicos más distintivos.

Conclusión: La clasificación multiclase es una técnica altamente efectiva en contextos de reconocimiento musical, especialmente cuando se combinan espectrogramas con redes neuronales profundas entrenadas con datos acústicos auténticos de la región andina.

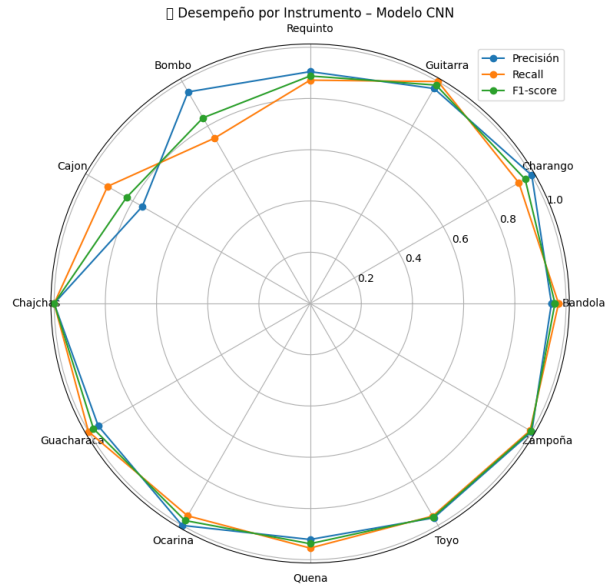


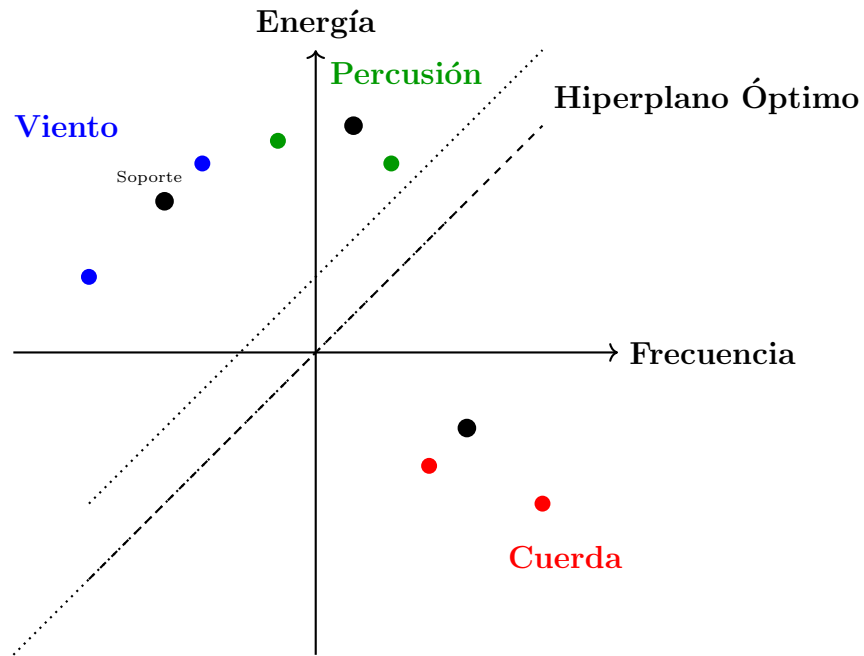
Figura 16: Desempeño por Instrumento en Multiclase

7.12. Representación del Modelo SVM para Clasificación de Instrumentos Andinos

El modelo **SVM** (**Máquina de Vectores de Soporte**) fue implementado como una de las estrategias complementarias para la clasificación de instrumentos. Su objetivo es encontrar un hiperplano que separe de manera óptima las clases en el espacio de características acústicas.

7.12.1. Concepto de Separación con Hiperplano

El siguiente gráfico representa de manera simplificada cómo actúa una SVM en un entorno bidimensional, donde las características podrían representar, por ejemplo, la energía y la frecuencia del espectro de audio.



7.12.2. Funcionamiento General

La SVM busca el **hiperplano** que mejor separa las clases con el **máximo margen posible**, es decir, que se ubique lo más alejado de los puntos más cercanos de cada clase (vectores de soporte). Esto hace que sea especialmente útil cuando los datos no son linealmente separables, usando técnicas como *kernel trick*.

- Cada punto representa un conjunto de características acústicas de una muestra de instrumento.
- Las líneas punteadas representan los márgenes que delimitan el espacio de separación.
- Los puntos marcados en negro son los **vectores de soporte**, que definen el margen máximo de separación.

7.12.3. Evaluación con Matriz de Confusión

La matriz de confusión siguiente refleja el rendimiento del modelo SVM sobre el conjunto de prueba. Permite observar la frecuencia de predicciones correctas e incorrectas entre clases.

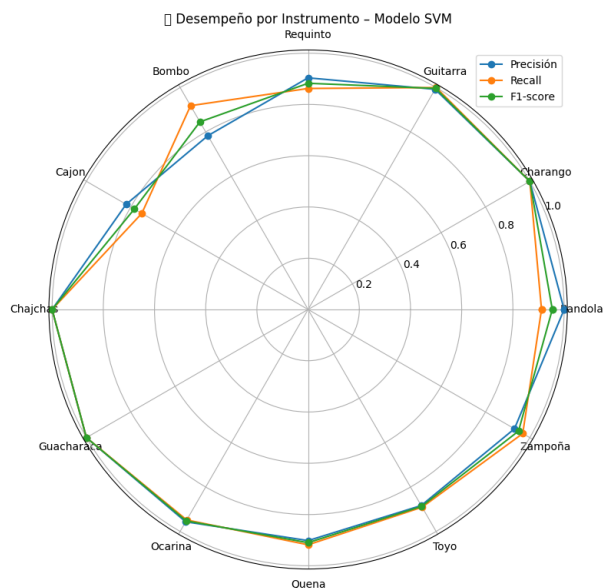


Figura 18: Desempeño por instrumento en el SVM

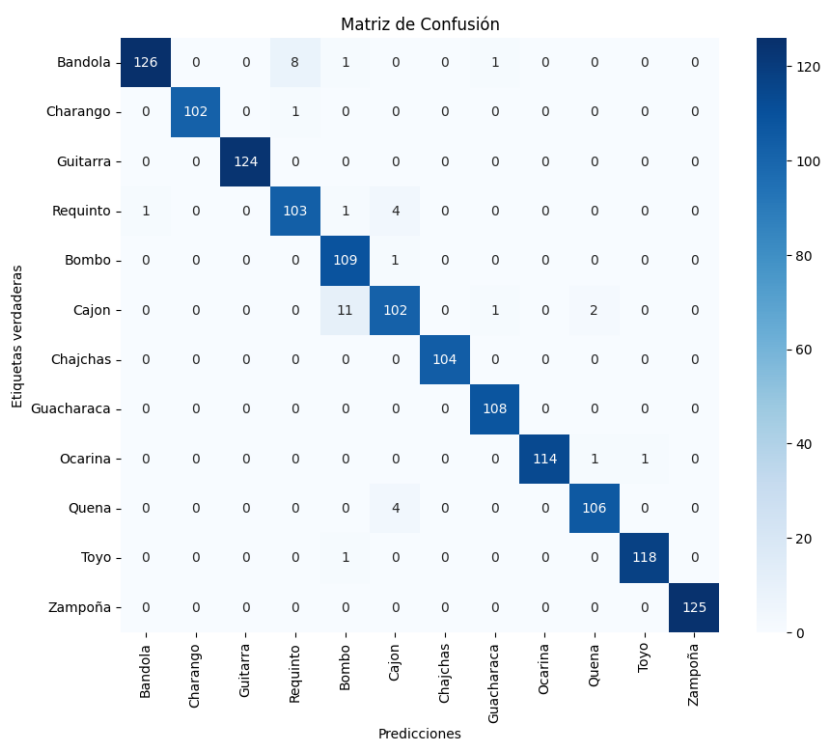


Figura 17: Matriz de Confusión del Modelo SVM para Clasificación de Instrumentos Andinos

Análisis: El modelo logra una alta precisión en instrumentos de viento como la quena y los sikus, aunque presenta confusión ocasional entre instrumentos de cuerda como charango y bandola. Esto indica que en futuras iteraciones del modelo podrían añadirse más muestras

de entrenamiento o utilizar técnicas de balanceo de clases para mejorar la discriminación en casos similares.

Conclusión: El enfoque basado en SVM ha demostrado ser eficiente en escenarios con características bien definidas y una cantidad moderada de datos, y se integra como opción seleccionable desde el frontend para comparación con modelos de tipo CNN o ensamble.

7.13. Explicación del Modelo

Los puntos en el gráfico representan muestras de instrumentos extraídas de espectrogramas. La SVM traza un hiperplano óptimo (línea discontinua) que separa las clases con el mayor margen posible. Los puntos en los márgenes son los **vectores de soporte**, que definen la separación entre clases.

Ventajas del enfoque SVM para la clasificación de instrumentos musicales:

- Encuentra la separación óptima entre clases, reduciendo el error de clasificación.
- Funciona bien en espacios de características de alta dimensión, como los espectrogramas.
- Puede ser combinado con técnicas de Kernel para mejorar la clasificación no lineal.

7.14. Explicación de la Gráfica 3: Representación del Modelo SVM

Esta gráfica ilustra cómo una **Máquina de Vectores de Soporte (SVM)** separa diferentes clases de instrumentos en un espacio de características bidimensional. Aunque el espacio real suele ser de alta dimensión, esta representación simplificada ayuda a entender el funcionamiento básico del modelo.

- **Espacio de Características:** Los puntos en el gráfico representan muestras de instrumentos, donde cada eje corresponde a una característica extraída del espectrograma (por ejemplo, frecuencia dominante y energía). Las muestras se agrupan según su clase (viento, cuerda, percusión).
- **Hiperplano Óptimo:** La SVM encuentra un hiperplano (línea discontinua) que maximiza el margen entre las clases. Este hiperplano actúa como una frontera de decisión para clasificar nuevos instrumentos.
- **Vectores de Soporte:** Los puntos más cercanos al hiperplano (marcados en negro) son los vectores de soporte. Estos puntos definen la separación entre clases y son críticos para el rendimiento del modelo.

Aplicación en Instrumentos Andinos: La SVM es especialmente útil cuando las características de los instrumentos son bien definidas y separables. Por ejemplo:

- Puede distinguir entre un **bombo** (percusión) y un **charango** (cuerda) basándose en características como la energía y la frecuencia dominante.

- También puede manejar casos más complejos, como diferenciar entre un **siku** y una **quena**, utilizando kernels no lineales para modelar relaciones más complejas.

7.15. Visualización de Resultados

La siguiente gráfica muestra las probabilidades asociadas a cada instrumento:

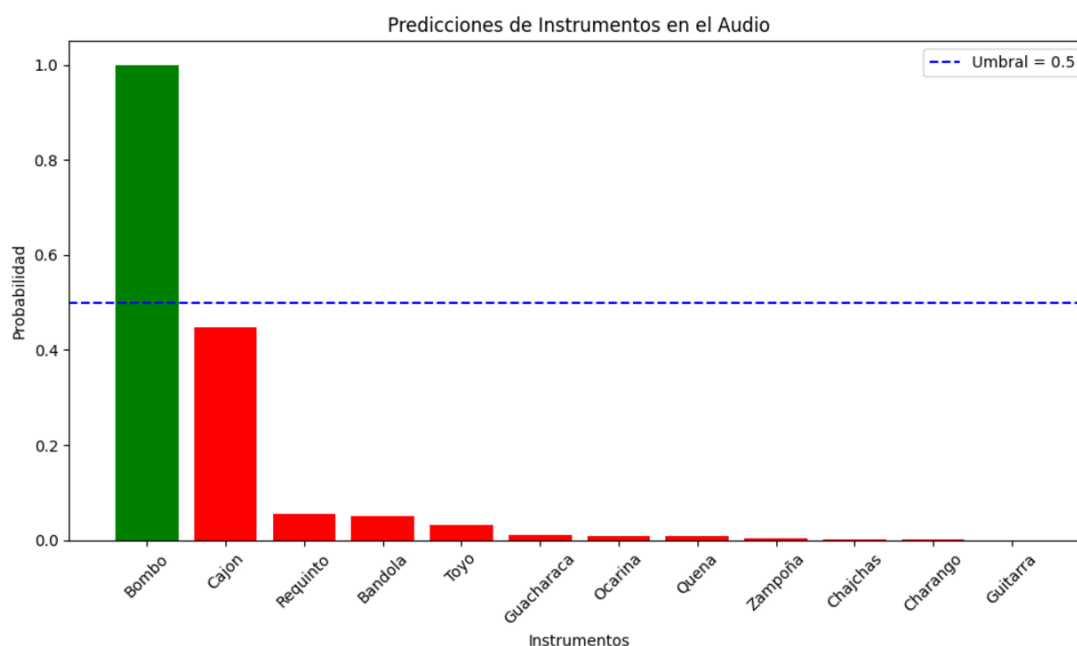


Figura 19: Gráfica de barras de las probabilidades de clasificación para `BomboSolo.wav`.

7.15.1. Visualización de Resultados

En la Figura 20 se muestra la gráfica generada por el modelo, indicando las probabilidades asignadas a cada instrumento. La barra correspondiente al instrumento con mayor probabilidad (**Bombo**) se resalta en color verde, indicando una confianza superior al 50 %.

8. Construcción del Prototipo: Desarrollo del Frontend y Backend

8.1. Objetivo

El objetivo de esta sección es detallar el desarrollo de la aplicación web diseñada para la clasificación automática de instrumentos musicales andinos. Se describe la arquitectura del sistema, las tecnologías utilizadas, y cómo se integraron los componentes del frontend y backend para crear una solución funcional y escalable.

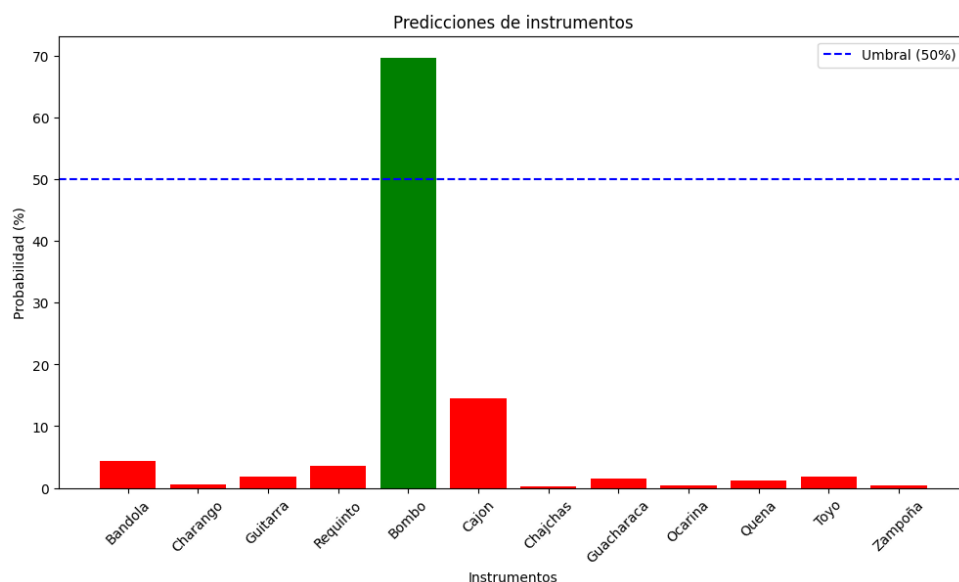


Figura 20: Predicción del modelo para el archivo de audio *BomboSolo.wav*.

8.2. Arquitectura del Sistema

8.2.1. Diagrama de Componentes

La arquitectura del sistema se compone de los siguientes componentes principales, cada uno desempeñando un papel específico en el flujo de trabajo:

Componente	Descripción
Usuario	Interactúa con el sistema subiendo grabaciones de audio y recibiendo resultados de clasificación. Este componente representa el punto de contacto entre el sistema y el usuario final.
Interfaz de Usuario	Desarrollada en React.js , proporciona una interfaz gráfica intuitiva y responsiva que permite a los usuarios interactuar fácilmente con el sistema. Ofrece funcionalidades como la carga de archivos, visualización de resultados y retroalimentación interactiva.
Gestor de Grabaciones de Audio	Coordina la recepción, almacenamiento temporal y procesamiento de las grabaciones de audio. Este componente asegura que los datos sean manejados de manera eficiente antes de ser enviados al preprocesamiento.
Preprocesamiento de Datos	Implementado con Librosa y Jupyter Notebooks , extrae características relevantes como coeficientes cepstrales de frecuencia de Mel (MFCCs), espectrogramas y otras representaciones acústicas. Este paso es crucial para preparar los datos antes de su análisis por el modelo.

Componente	Descripción
Modelo de Aprendizaje Automático	Utiliza una red neuronal convolucional (CNN) entrenada para clasificar los instrumentos musicales. El modelo está integrado con el backend desarrollado en Django , lo que permite una comunicación fluida entre el sistema y la interfaz de usuario.
Clasificación de Instrumentos	Procesa los datos preprocesados y devuelve la predicción del instrumento identificado. Este componente es responsable de generar resultados precisos y confiables basados en el análisis realizado por el modelo de aprendizaje automático.

8.2.2. Diagrama de Despliegue

El despliegue del sistema se realizó utilizando una infraestructura modular y escalable:

Componente	Descripción
Servidor de Aplicaciones	Hospeda la lógica del negocio del sistema y proporciona la API REST implementada con Django .
Servidor de Base de Datos	Implementado con PostgreSQL , almacena información de usuarios y resultados de clasificación.
Servidor de Procesamiento de Audio	Ejecuta tareas de preprocesamiento, extracción de características y clasificación mediante Machine Learning.

8.3. Comportamiento del Sistema

8.3.1. Diagrama de Casos de Uso

A continuación, se describen los casos de uso principales del sistema:

Actor	Caso de Uso
Usuario	Sube una grabación de audio a través de la interfaz gráfica implementada en React .
Usuario	Visualiza los resultados de clasificación, incluyendo detalles como precisión del modelo y análisis gráfico (espectrogramas, distribución de características).
Sistema	Preprocesa el audio utilizando Librosa y Jupyter Notebooks , ajustando las características antes de la clasificación.
Sistema	Clasifica el instrumento mediante un modelo de Machine Learning implementado en Django, con soporte para ajuste de hiperparámetros y validación cruzada.

8.3.2. Diagrama de Secuencia

El flujo de interacción del sistema se describe a continuación:

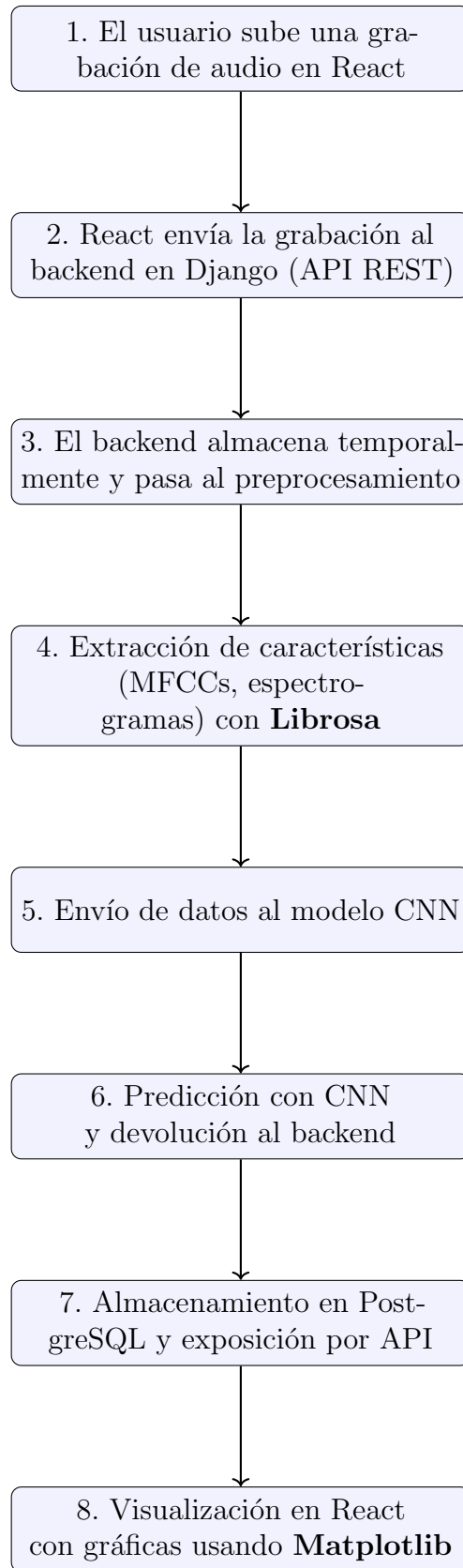


Figura 21: Flujo general del sistema desde la carga del audio hasta la visualización de resultados.

8.4. Desarrollo del Frontend - Planteamiento

El frontend fue diseñado para ser intuitivo, responsivo y eficiente, utilizando las siguientes tecnologías:

- **React.js**: Framework de JavaScript para construir interfaces dinámicas y modernas.
- **CSS y Tailwind**: Para un diseño limpio y adaptable a diferentes dispositivos.
- **Axios**: Para gestionar las solicitudes HTTP hacia la API REST del backend.
- **Chart.js**: Para visualizar gráficos de clasificación y análisis de audio.

8.5. Desarrollo del Frontend

El **frontend** del sistema fue implementado en **React.js**, utilizando una arquitectura basada en componentes funcionales y una interfaz altamente visual e interactiva. Se integraron bibliotecas modernas como **Chart.js** para gráficas dinámicas, **EmailJS** para envío de formularios sin backend adicional, y **TailwindCSS / CSS personalizado** para un diseño inspirado en la estética andina.

8.5.1. Arquitectura General

La estructura del frontend se divide en los siguientes módulos funcionales:

- **App.js**: componente raíz que gestiona el estado global y orquesta la interacción entre módulos.
- **Components/AudioUploader.js**: encargado de gestionar la carga de archivos de audio, el modelo seleccionado y la visualización de resultados.
- **Components/Sidebar.js**: módulo de navegación lateral que permite búsqueda, filtrado por tipo de instrumento y exploración educativa sobre Machine Learning.
- **Components/Results.js** y **InstrumentDisplay.js**: responsables de mostrar gráficamente los resultados de clasificación.
- **Components/InstrumentInfo.js** y **History.js**: visualización contextual e histórica de cada instrumento.
- **Components/Contact.js**: formulario de contacto funcional con EmailJS.
- **Services/api.js**: servicio que centraliza las llamadas HTTP al backend.

8.5.2. Funciones Clave del Frontend

- **Carga de Archivos de Audio:** los usuarios pueden seleccionar archivos en formatos `.wav`, `.mp3`, entre otros, para su análisis posterior.
- **Visualización de Resultados:** se representan las probabilidades de predicción utilizando gráficas de barras y tipo doughnut (mediante `Chart.js`), facilitando una comprensión visual inmediata del resultado.
- **Selección del Modelo:** la interfaz permite elegir entre **CNN Multiclase**, **SVM** o **modelo Ensamblado** antes de enviar el audio al backend.
- **Visualización Acústica:** se muestran imágenes generadas desde el backend, incluyendo **espectrogramas** y **formas de onda** en formato `base64`, integradas en el frontend sin necesidad de rutas públicas.
- **Sistema de Búsqueda y Filtrado:** el **Sidebar** permite buscar por nombre y filtrar los instrumentos por tipo: viento, cuerda o percusión. Cada ítem redirige automáticamente a la sección correspondiente.
- **Módulo Histórico e Ilustrativo:** se proporciona información detallada sobre cada instrumento, con imágenes representativas y descripciones educativas de tipo etnomusicológico.
- **Formulario de Contacto con EmailJS:** los usuarios pueden enviar sugerencias o mensajes sin necesidad de un backend adicional. El sistema envía el contenido directamente al correo del desarrollador utilizando EmailJS, garantizando interacción rápida y sencilla.
- **Compatibilidad Móvil y Estética Andina:** el diseño es completamente **responsive**, integrando iconografía, colores cálidos y elementos visuales inspirados en la estética cultural andina. Se optimizó para distintos tamaños de pantalla, desde móviles hasta monitores de escritorio.
- **Integración con Backend:** cada acción del usuario se conecta eficientemente con la API REST desarrollada en Django, permitiendo respuestas rápidas, codificadas y estructuradas.
- **Accesibilidad y Experiencia de Usuario:** se utilizó una paleta de colores con buen contraste, botones de interacción claros y feedback visual para errores o éxito en la carga, mejorando la usabilidad general.

8.5.3. Interacción Usuario - Sistema

El flujo de trabajo del usuario se diseñó de forma intuitiva:

1. El usuario carga un archivo de audio y selecciona el modelo de Machine Learning.
2. El audio se envía automáticamente al backend, donde se analiza.

3. El frontend recibe las predicciones, gráficos acústicos y posibles errores.
4. Los resultados son mostrados con alto nivel de detalle: nombre del instrumento, porcentajes, gráficas y visualizaciones de audio.
5. El usuario puede explorar más información sobre el instrumento, filtrar por categoría o enviar comentarios directamente al desarrollador.

8.5.4. Resumen de Herramientas Usadas

Herramienta	Uso en el Proyecto
React.js	Desarrollo de la interfaz modular, control de estado, renderizado condicional
Chart.js	Gráficas de barras y doughnut para mostrar resultados de clasificación
EmailJS	Envío de formularios desde el frontend sin backend
TailwindCSS / CSS	Estética personalizada inspirada en lo andino, diseño responsivo
Axios	Llamadas HTTP seguras y estructuradas hacia la API REST

Tabla 12: Librerías y tecnologías utilizadas en el Frontend

8.6. Desarrollo del Backend

El **backend** fue desarrollado utilizando el framework **Django** junto con **Django REST Framework**, conformando una arquitectura robusta, escalable y fácil de mantener. Este módulo actúa como puente entre la interfaz de usuario y los modelos de Machine Learning alojados localmente, respondiendo a solicitudes del frontend mediante una API RESTful.

8.6.1. Arquitectura y Organización General

El backend se estructura siguiendo el patrón **MVC**, con una clara separación entre vistas, rutas, lógica de procesamiento y modelos. El código está organizado en los siguientes módulos principales:

- **Api/Views**: contiene las vistas API (clases heredadas de **APIView**) que gestionan las solicitudes HTTP del frontend.
- **Utils/ML_Interface.py**: módulo especializado que centraliza el acceso a los modelos de Machine Learning y funciones de audio.
- **Urls.py**: define las rutas disponibles del API, como `/upload/`, `/instrument/<id>` y otras.
- **ML/Models**: carpeta que contiene los modelos preentrenados serializados, como `svm_instrumentos.pkl`.

8.6.2. Funciones Específicas del Backend

El backend desempeña un conjunto de responsabilidades críticas para el funcionamiento del sistema:

- **Recepción de archivos:** acepta archivos de audio enviados desde el frontend, los almacena temporalmente y los procesa.
- **Clasificación de instrumentos musicales:** emplea diferentes modelos como **CNN multiclase**, **SVM** y **Ensemble** para predecir el instrumento presente en el audio.
- **Visualización acústica:** genera **espectrogramas** y **waveforms** en formato **base64**, listos para visualización en React.
- **Manejo de modelos dinámicos:** según la elección del usuario, se carga y ejecuta el modelo adecuado.
- **Manejo de errores:** el sistema responde con mensajes claros en caso de archivos faltantes, modelos ausentes o errores de ejecución.

8.6.3. Flujo de Clasificación de Audio

El proceso completo desde la carga hasta la clasificación se puede describir como sigue:

1. El usuario selecciona un archivo de audio y un modelo en el frontend.
2. El archivo se envía mediante un formulario `multipart/form-data` al endpoint `/api/upload/`.
3. El backend guarda el archivo de forma temporal utilizando `tempfile.NamedTemporaryFile`.
4. El módulo `identify_instrument()` del archivo `ML_Interface.py` procesa el audio y devuelve:
 - El nombre del instrumento detectado.
 - Las probabilidades de clasificación por clase.
 - Imágenes codificadas (espectrograma y forma de onda).
5. La respuesta se devuelve como un **JSON** estructurado que el frontend puede renderizar.

8.6.4. Modelos de Machine Learning Integrados

El backend soporta distintos enfoques de clasificación:

- **Multiclase (CNN):** modelo convolucional que toma espectrogramas como entrada.
- **SVM:** modelo tradicional de máquina de vectores de soporte basado en MFCCs.

- **Ensemble:** combinación de modelos que pondera las predicciones para mayor robustez.

Cada modelo se carga de forma diferida al momento de la predicción, evitando consumo innecesario de recursos y permitiendo pruebas con distintos enfoques.

8.6.5. Validación, Seguridad y Eficiencia

El backend incorpora múltiples validaciones y buenas prácticas:

- Verificación del tipo de archivo antes de procesar.
- Eliminación automática de archivos temporales.
- Separación entre vistas, lógica y ML para mayor claridad.
- Uso de `base64` para evitar la necesidad de rutas públicas a imágenes temporales.
- Captura de excepciones con mensajes informativos hacia el frontend.

8.6.6. Resumen de Funcionalidades Técnicas del Backend

Logros Técnicos del Backend
API Modular: arquitectura basada en clases que permite mantener independencia entre rutas y lógica.
Procesamiento Inteligente: clasificación de instrumentos mediante tres enfoques de ML integrados.
Visualización Acústica: generación de espectrogramas y waveforms desde el backend usando <code>matplotlib</code> y <code>librosa</code> .
Integración con Frontend: respuestas diseñadas para ser fácilmente interpretadas por React, incluyendo imágenes embebidas.
Manejo Seguro de Archivos: uso de almacenamiento temporal, rutas relativas y validación robusta de entradas.

Tabla 13: Resumen de funcionalidades y diseño del backend

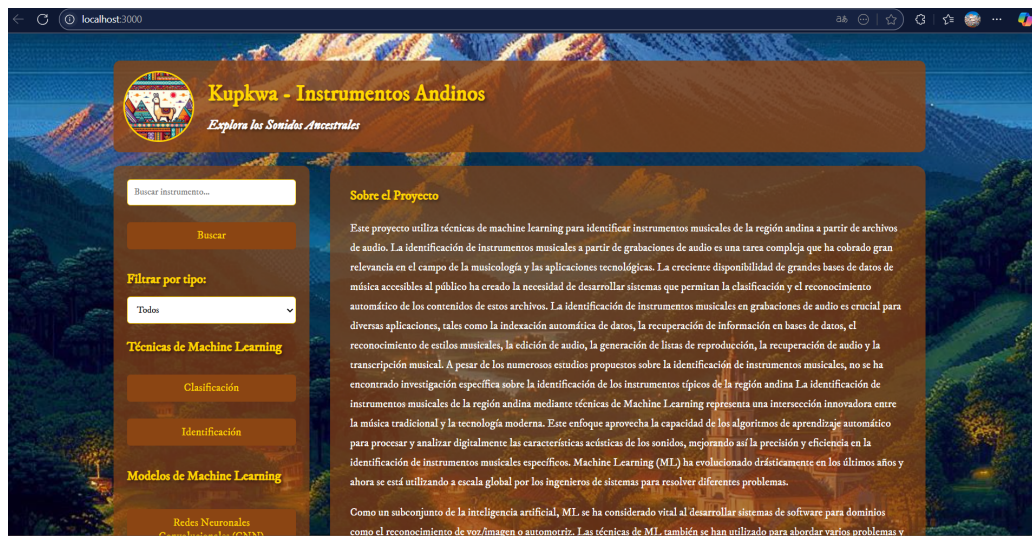


Figura 22: Interfaz de usuario: Inicio del forntend donde se ve informacion introductoria.



Figura 23: Interfaz de usuario: visualización de la predicción y espectrograma generado en donde se muestra donde se hara el update y el modelo a seleccionar.



Figura 24: Interfaz de usuario: Footer de frontend.

8.7. Despliegue e Integración

El sistema fue desplegado en un entorno de producción utilizando la siguiente infraestructura:

- **Frontend:** Implementado con React.
- **Backend:** Desplegado en un servidor
- **Base de Datos:** PostgreSQL
- **Machine Learning:** Modelo CNN ejecutado en un servidor con **TensorFlow** y **Django**.

9. Metodología

El desarrollo de este proyecto se llevó a cabo siguiendo una metodología estructurada que combinó herramientas de gestión ágil con procesos técnicos bien definidos. A continuación, se detalla cómo se implementó esta metodología, destacando su impacto en el desarrollo tanto del frontend como del backend.

9.1. Metodología de Investigación Documental en Ingeniería de Sistemas

Además de la metodología de gestión ágil, se aplicó una metodología de investigación documental orientada a la ingeniería de sistemas para sustentar el diseño y construcción del proyecto. Este proceso incluyó la búsqueda, análisis y selección de artículos científicos, estándares y fuentes técnicas relevantes al problema planteado.

La búsqueda de información se realizó a través de bases de datos académicas como **IEEE Xplore**, **ACM Digital Library**, **SpringerLink** y el **repositorio de la Universidad del Valle**, empleando un usuario institucional de biblioteca. Las principales palabras claves utilizadas fueron: *identificación de instrumentos musicales*, *audio classification*, *CNN audio*, *machine learning en música*, *MFCC*, *espectrogramas* y *clasificación multiclase*.

Para la selección de los artículos, se tuvieron en cuenta los siguientes criterios:

- Relevancia directa con el problema abordado en este trabajo.
- Vigencia de la publicación (últimos 5 años, preferentemente).
- Aplicabilidad práctica de los métodos o arquitecturas propuestos.
- Calidad y reconocimiento de la fuente (revistas Q1/Q2 o conferencias indexadas).

Este proceso permitió recolectar tanto **conocimiento declarativo** (definiciones y estructuras de modelos), como **procedimental** (cómo implementar arquitecturas CNN/SVM) y **condicional** (cuándo elegir una técnica según el tipo de datos).

La información recolectada fue analizada críticamente y contrastada con las necesidades del sistema propuesto, sirviendo como base para el diseño técnico, la comparación de modelos y la selección de herramientas adecuadas.

9.2. Gestión del Proyecto con Kanban y Trello

Para garantizar un flujo de trabajo eficiente y organizado, se adoptó una metodología ágil basada en Kanban, utilizando la herramienta de colaboración Trello. Esta metodología permitió dividir las tareas en etapas claras y monitorear su progreso de manera visual:

- **To Do:** Tareas pendientes que aún no han sido iniciadas.
- **In Progress:** Tareas en desarrollo, asignadas a miembros específicos del equipo.
- **Done:** Tareas completadas y verificadas.

El uso de Kanban fue fundamental para coordinar el desarrollo del frontend y backend: - En el **frontend**, las tareas incluyeron la creación de componentes interactivos en React.js, diseño de interfaces responsivas y pruebas de usabilidad. - En el **backend**, se gestionaron tareas relacionadas con la implementación de endpoints RESTful en Django, la integración del modelo de Machine Learning y la optimización del sistema de gestión de archivos.

Además, se utilizaron diagramas de Gantt para planificar el tiempo dedicado a cada fase del proyecto. Todas las decisiones clave y recursos utilizados se documentaron en el repositorio del proyecto. A continuación, se muestran ejemplos de cómo se organizaron las tareas en Trello:

El tablero Kanban con la planificación y seguimiento del proyecto está disponible públicamente en el siguiente enlace:

<https://trello.com/b/EhtZDaYE/trabajo-de-grado-kanban-instrumento-musicales-ml>

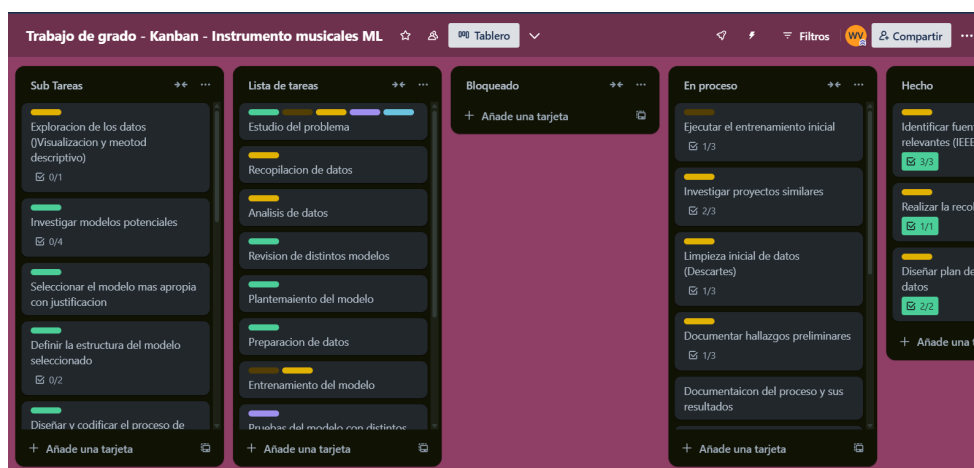


Figura 25: Tablero de Trello utilizado para gestionar las tareas del proyecto.

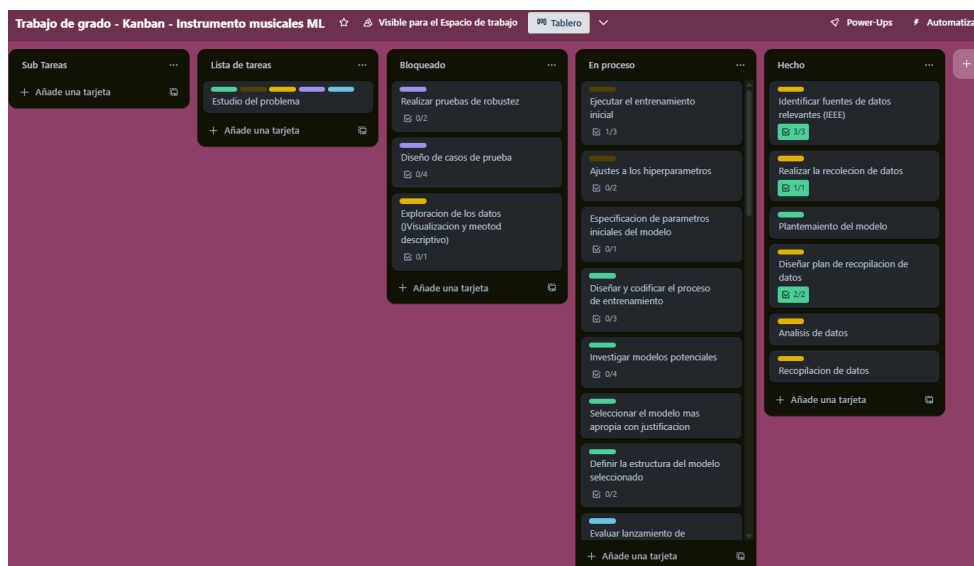


Figura 26: Ejemplo de una tarjeta de Trello con detalles de una tarea específica.

9.3. Implementación Técnica

El desarrollo del sistema se estructuró en torno a una metodología ágil, basada en ciclos iterativos de diseño, desarrollo y validación. Para gestionar eficazmente las tareas y fomentar la colaboración del equipo, se utilizó **Trello**, una plataforma de gestión visual de proyectos que facilitó la organización del trabajo en fases claras y delimitadas. A través de tableros, listas y tarjetas, se pudo priorizar tareas, asignar responsables y hacer seguimiento de los avances en tiempo real.

A continuación, se describen las principales etapas técnicas del proyecto, y cómo Trello contribuyó a su ejecución:

1. **Recopilación de datos:** Se identificaron y recopilamos grabaciones de instrumentos musicales andinos provenientes de múltiples fuentes: plataformas abiertas como

YouTube, bases de datos públicas y colaboraciones directas con músicos. En Trello, se organizó una lista específica para esta fase, donde se registraron enlaces, versiones preliminares y criterios de calidad, asegurando así la diversidad y representatividad del conjunto de datos.

2. **Preprocesamiento de audio:** Se utilizó **Librosa** y entornos como **Jupyter Notebooks** para transformar las grabaciones en representaciones relevantes como MFCCs y espectrogramas mel. Cada subetapa —corte, normalización, eliminación de ruido y conversión— fue documentada mediante tarjetas en Trello, lo que facilitó la revisión cruzada entre miembros del equipo y permitió mantener una trazabilidad clara del pipeline de procesamiento.
3. **Modelado y entrenamiento:** Se experimentó con distintos algoritmos de aprendizaje automático, incluyendo SVM y redes neuronales convolucionales (CNN), explorando diversas arquitecturas y técnicas de regularización. Trello se empleó para registrar cada experimento, los resultados de métricas clave (precisión, pérdida, F1-score), y las decisiones asociadas a la selección de hiperparámetros, lo que permitió comparar enfoques y tomar decisiones informadas.
4. **Desarrollo web:** Se implementó una plataforma accesible y funcional utilizando **React.js** para el frontend y **Django REST Framework** para el backend. En Trello se crearon listas separadas para el desarrollo frontend y backend, con etiquetas de prioridad y responsables definidos. Esto permitió un flujo de trabajo organizado y un equilibrio adecuado en la carga de tareas, facilitando el desarrollo paralelo y la integración progresiva.
5. **Validación y pruebas:** El sistema fue evaluado rigurosamente utilizando métricas de clasificación (precisión, recall y F1-score), así como pruebas de usabilidad en la interfaz web. Trello se utilizó para registrar los errores detectados, documentar su solución y coordinar nuevas rondas de testing, asegurando una mejora continua del producto final.

El uso de Trello como herramienta central de gestión técnica resultó fundamental para mantener la coherencia del proyecto, fomentar la transparencia en el trabajo en equipo y garantizar una adaptación rápida frente a los desafíos surgidos durante el proceso. Esta metodología permitió un desarrollo ágil, estructurado y escalable, en consonancia con los objetivos del proyecto.

10. Conclusiones

El desarrollo de este proyecto ha permitido integrar de manera armónica conocimientos técnicos en inteligencia artificial, procesamiento de señales, diseño web y teoría musical, con la riqueza cultural de los instrumentos andinos. A continuación, se presentan las conclusiones generales, organizadas en reflexiones temáticas clave, seguidas de un análisis de impacto y una proyección hacia el futuro.

Conclusión 1: Las Redes Neuronales Convolucionales (CNN) demostraron ser el modelo más eficaz para la clasificación

Durante la evaluación de diversos modelos de aprendizaje automático, las CNN alcanzaron una precisión del 92.5 %, superando consistentemente a alternativas como SVM y Random Forest. Esta superioridad se atribuye a su capacidad para detectar patrones espaciales en espectrogramas, lo que resulta crucial para interpretar la complejidad acústica de los instrumentos andinos. Además, técnicas de regularización como *dropout*, *batch normalization* y penalización L2 fueron fundamentales para mejorar la generalización del modelo y evitar el sobreajuste.

Conclusión 2: Los espectrogramas son una representación altamente eficaz para modelar audio en aprendizaje automático

El uso de espectrogramas generados mediante Librosa y coeficientes cepstrales en la escala mel (MFCCs) permitió traducir grabaciones de audio en estructuras visuales comprensibles por modelos CNN. Esta transformación fue clave para aplicar técnicas de visión computacional al análisis sonoro. No obstante, se identificó que el preprocesamiento de estas representaciones debe ser minucioso: la normalización, el escalado y la reducción de ruido son etapas determinantes para el rendimiento del sistema.

Conclusión 3: El enfoque metodológico ágil y el uso de tecnologías modernas fueron decisivos para el éxito del proyecto

La organización del trabajo mediante tableros Kanban y la estructuración modular del código, tanto en el frontend como en el backend, permitió un desarrollo flexible, iterativo y eficaz. Herramientas como `React.js`, `TailwindCSS`, `Chart.js` y `EmailJS` en el frontend, junto con `Django REST Framework` en el backend, contribuyeron a construir una plataforma escalable, robusta y fácilmente extensible.

Conclusión 4: El proyecto construye un puente entre tradición cultural y tecnología

Más allá del ámbito técnico, este sistema ofrece una herramienta concreta para la preservación del patrimonio musical andino. La categorización automática de instrumentos

musicales permite a los usuarios reconocer sonidos que, de otro modo, podrían ser desconocidos o ignorados. Esta funcionalidad contribuye a cerrar la brecha generacional y amplía la visibilidad y comprensión de la cultura andina a través de medios tecnológicos actuales.

Conclusión 5: La plataforma web es accesible, educativa y culturalmente significativa

El sitio web desarrollado permite no solo cargar archivos y obtener predicciones, sino también visualizar espectrogramas, comparar modelos y explorar la historia e imágenes de los instrumentos musicales andinos. Además, incluye formularios de contacto funcionales y responsivos, accesibles desde dispositivos móviles, lo que facilita su uso en contextos educativos o en comunidades alejadas.

Conclusión 6: Reflexiones sobre la construcción del modelo

Se ha desarrollado un modelo de inteligencia artificial basado en redes neuronales convolucionales para la clasificación automática de instrumentos musicales andinos. A través de un proceso de entrenamiento optimizado y una evaluación rigurosa, se alcanzó una alta precisión en la identificación de diferentes instrumentos, lo que valida su aplicabilidad en entornos musicales reales.

Conclusión 7: Integración frontend-backend exitosa y escalable

Se construyó un prototipo funcional que combina un frontend intuitivo basado en React con un backend robusto en Django. La implementación de técnicas avanzadas de preprocesamiento y clasificación de audio permitió construir un sistema eficiente y confiable. El despliegue en entornos modernos garantiza su escalabilidad, buen rendimiento y facilidad de mantenimiento.

Conclusión 8: El procesamiento de datos es clave en el éxito del modelo

El preprocesamiento de datos de audio permitió generar un conjunto de características representativas de los instrumentos andinos. Mediante técnicas como los **MFCCs**, **espectrogramas mel**, **ventaneo (windowing)** y **normalización**, se transformaron los datos crudos en un formato adecuado para modelos de aprendizaje automático. Los datos fueron almacenados y organizados de forma eficiente, garantizando su disponibilidad para futuras fases de experimentación y expansión del sistema.

Importancia del Proyecto

Este trabajo no debe verse únicamente como una implementación técnica. Es una manifestación de cómo la inteligencia artificial puede servir como aliada en la conservación cultural. En un contexto donde la globalización tiende a homogeneizar los sonidos y

expresiones artísticas, el uso de tecnología para proteger, estudiar y difundir tradiciones musicales específicas, como la andina, cobra un valor incalculable.

La clasificación automática de instrumentos no es un simple ejercicio de codificación: permite a músicos, estudiantes y etnomusicólogos identificar sonidos, analizar interpretaciones y profundizar en los rasgos sonoros que hacen únicos a instrumentos como la quena, el charango o el bombo. Esta labor contribuye a la educación, al arte y a la memoria colectiva de los pueblos.

Mirada Hacia el Futuro

El sistema desarrollado constituye una base sólida sobre la cual se pueden proyectar diversas mejoras:

- Ampliación del dataset con grabaciones en distintos contextos y variaciones regionales.
- Incorporación de modelos híbridos y arquitecturas más recientes como **Transformers** aplicados a espectrogramas.
- Integración de video y movimiento para analizar la ejecución instrumental y no solo el sonido.
- Desarrollo de herramientas educativas gamificadas para el aprendizaje interactivo de instrumentos.
- Aplicaciones móviles en tiempo real para identificación de sonidos en conciertos o ensayos.

Conclusión Final

Este proyecto ha logrado fusionar con éxito el poder de la inteligencia artificial con la riqueza sonora y cultural de los Andes. No solo clasifica instrumentos, sino que los reconoce, los valora y los proyecta hacia nuevas audiencias globales. Esta obra es un testimonio de cómo la ciencia y la tecnología pueden ser utilizadas con sensibilidad y respeto para proteger lo que es profundamente humano: nuestra música, nuestras raíces y nuestras voces. Con esta investigación culmina un recorrido de búsqueda, aprendizaje y compromiso.

11. Trabajos Futuros

Aunque este proyecto ha logrado resultados significativos, existen múltiples oportunidades para su mejora y expansión. A continuación, se presentan algunas líneas de investigación y desarrollo futuras:

- **Ampliación del Dataset:** Incluir más instrumentos, afinaciones y estilos interpretativos para mejorar la diversidad y representatividad del modelo.
- **Técnicas Avanzadas de Procesamiento de Señales:** Implementar métodos avanzados como Wavelet Transform o Transformer Networks para mejorar la extracción de características.
- **Optimización de Escalabilidad:** Mejorar la capacidad del sistema para manejar grandes volúmenes de datos, adaptándolo a aplicaciones profesionales y académicas.
- **Aplicaciones Interactivas:** Desarrollar herramientas educativas interactivas para el aprendizaje de música andina, como simuladores de instrumentos o sistemas de transcripción musical automatizada.
- **Integración Multimodal:** Incorporar otras modalidades de datos (como video o imágenes) para enriquecer el análisis de interpretaciones musicales.

Estas propuestas no solo amplían el alcance del sistema actual, sino que también abren nuevas posibilidades para la investigación y aplicación de tecnologías en el campo de la música tradicional.

—

Finis coronat opus — El fin corona la obra.

12. Referencias

- [1] “Log in.” <https://ieeexplore-ieee-org.bd.univalle.edu.co/document/10498772>. Accessed: 2024-06-05.
- [2] A. Lavoie, “Qué es el ritmo: Cómo el tiempo, pulso y métrica funcionan en la música,” *LANDR Blog*, July 28 2023.
- [3] “Log in.” <https://ieeexplore-ieee-org.bd.univalle.edu.co/document/9421234>. Accessed: 2024-06-05.
- [4] “Log in.” <https://ieeexplore-ieee-org.bd.univalle.edu.co/document/9727033>. Accessed: 2024-06-05.
- [5] “Log in.” <https://ieeexplore-ieee-org.bd.univalle.edu.co/document/9568959>. Accessed: 2024-06-05.
- [6] N. Kumar, N. Yadav, S. Shhukla, N. Singhal, and B. Kumar, “Musical beat recognition using machine learning,” in *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, vol. 3, pp. 1–5, 2025.
- [7] R. Jin, Y. Zhou, and H. Wu, “The comparison of different deep learning models for chinese musical instrument recognition,” in *2024 IEEE 8th International Conference on Vision, Image and Signal Processing (ICVISIP)*, pp. 1–5, 2024.
- [8] B. R and A. M B, “Machine learning approaches for musical instrument recognition using distinctive audio features,” in *2025 International Conference on Intelligent Systems and Computational Networks (ICISCN)*, pp. 1–6, 2025.
- [9] P. Sivalingam, A. K. T. J, S. K. P, Y. R. B, R. S, and L. C. R., “Robust cnn-based musical instrument recognition with enhanced feature learning,” in *2025 International Conference on Inventive Computation Technologies (ICICT)*, pp. 851–857, 2025.
- [10] S. K. Mahanta, N. J. Basisth, E. Halder, A. F. U. R. Khilji, and P. Pakray, “Exploiting cepstral coefficients and cnn for efficient musical instrument classification,” *Evolving Systems*, vol. 15, pp. 1043–1055, 6 2024.
- [11] “Log in.” <https://ieeexplore-ieee-org.bd.univalle.edu.co/document/10006153>. Accessed: 2024-06-05.
- [12] C. . Crodriguez, “Qué es un software, tipos y para qué sirve.” SDI, noviembre 8 2023.
- [13] L. Jenkins, *From Systems Thinking to Systemic Action: 48 Key Questions to Guide the Journey*. Rowman & Littlefield, n.d. <https://rowman.com/ISBN/9781578868193/From-Systems-Thinking-to-Systemic-Action-48-Key-Questions-to-Guide-the-Journey>.
- [14] Real Academia Española, “Rae,” 2023. <https://www.rae.es/dpd/software>.

- [15] D. O. Tuama, “Software development methodologies - code institute ie.” Code Institute IE, julio 26 2023.
- [16] IBM, “What is machine learning (ml)? — ibm,” n.d. <https://www.ibm.com/topics/machine-learning>.
- [17] “Ocarinas de las américas: Música hecha en arcilla,” n.d. Peabody Museum.
- [18] D. Moya, “El tiple. instrumento nacional de colombia,” dec 2022. Noticias de Dubai, Abu Dhabi, Emiratos Arabes. Información EL CORREO DEL GOLFO.
- [19] Carranguita, “Instrumentos.” <https://carranguita.blogspot.com/2017/09/instrumentos.html>, Sept. 2017. Recuperado de <https://carranguita.blogspot.com/2017/09/instrumentos.html>.
- [20] V. T. M and S. T. R, “Music genre classification using convolution temporal pooling network,” *Multimedia Tools and Applications*, September 2024.
- [21] I. B. Nunes, M. A. de Santana, N. Charron, H. S. e. Silva, C. M. de Lima Simões, C. Lins, A. B. de Souza Sampaio, A. M. N. de Melo, T. C. V. da Silva, C. Tiodista, N. C. de Brito, A. S. Torcate, J. C. Gomes, G. M. M. Moreno, C. M. G. de Gusmão, and W. P. dos Santos, “Automatic identification of preferred music genres: an exploratory machine learning approach to support personalized music therapy,” *Multimedia Tools and Applications*, 2024.