Reconocimiento de emociones musicales basado en características de audio dotadas de un contexto
2025 Leonardo Daniel Villanueva Medina

Universidad Autónoma de Querétaro

Facultad de Ingeniería

Reconocimiento de emociones musicales basado en características de audio dotadas de un contexto.

Tesis

Que como parte de los requisitos para obtener el Grado de

Maestro en

Ciencias en Inteligencia Artificial

Presenta

Leonardo Daniel Villanueva Medina

Dirigido por:
Dr. Efrén Gorrostieta Hurtado

Querétaro, Qro. a 28 de octubre de 2025

Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias en Inteligencia Artificial

Reconocimiento de emociones musicales basado en características de audio dotadas de un contexto

Tesis

Que como parte de los requisitos para obtener el Grado de

Maestro en Ciencias en Inteligencia Artificial

Presenta

Leonardo Daniel Villanueva Medina

Dirigido por

Dr. Efrén Gorrostieta Hurtado

Dr. Efrén Gorrostieta Hurtado
Presidente

Dr. Juan Manuel Ramos Arreguín
Secretario

Dr. Jesús Carlos Pedraza Ortega
Vocal

Dr. Marco Antonio Aceves Fernández
Suplente

Mtr. Luis Roberto García Noguez
Suplente

Centro Universitario, Querétaro, Qro. México
Fecha de aprobación por el Consejo Universitario (mes y año)

Con todo mi cariño y especial dedicación a mis padres, por su apoyo incondicional y amor constante. Este trabajo es un reflejo de su esfuerzo y dedicación.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido de alguna manera a la realización de esta tesis. En primer lugar, agradezco profundamente a mi director de tesis, el Dr. Efrén Gorrostieta Hurtado, por su orientación, confianza y paciencia.

También quiero agradecer a los miembros del sínodo por su tiempo, apoyo, conocimiento y valiosas sugerencias que enriquecieron este trabajo.

Agradezco a la institución Universidad Autónoma de Querétaro por brindarme un espacio adecuado para el desarrollo de esta investigación y por los recursos proporcionados.

A la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (Secihti), por el apoyo financiero otorgado por medio de la beca nacional de estudios de posgrado, que hizo posible la realización de este proyecto.

A mis compañeros y todos aquellos en quienes de manera directa o indirecta me motivaron a seguir adelante durante este proceso.

A mi profesor de música Raúl, quien siempre transmite su pasión y alegría por este arte.

Finalmente, agradezco profundamente a mi familia por su amor, comprensión y esfuerzo constante, este logro es en parte de ustedes.

Abreviaturas y siglas

Abreviatura/Sigla	Significado
ADFF	Attention Based Deep Feature Fusion
AMC	Audio Mood Classification
BiLSTM	Bidirectional Long Short-Term Memory
CBOW	Continuous Bag of Words
CNN	Convolutional Neural Network (Red Neuronal Convolucional)
CQT	Constant-Q Transform (Transformada Q Constante)
DEAM	Database for Emotional Analysis in Music
DL	Deep Learning (Aprendizaje Profundo)
EEG	Electroencephalogram (Electroencefalograma)
EM	Expectation-Maximization
FC	Fully Connected (Totalmente Conectada)
FFT	Fast Fourier Transform (Transformada Rápida de Fourier)
GEMS	Geneva Emotional Music Scale
GMM	Gaussian Mixture Model (Modelo de Mezcla Gaussiana)
GPR	Gaussian Process Regression (Regresión de Procesos Gaussianos)
ISMIR	International Society for Music Information Retrieval
LASSO	Least Absolute Shrinkage and Selection Operator
LSTM	Long Short-Term Memory (Memoria a Corto y Largo Plazo)
MAE	Mean Absolute Error (Error Absoluto Medio)
MER	Music Emotion Recognition (Reconocimiento de Emociones en la Música)
MEVD	Music Emotion Variation Detection
MFCC	Mel-Frequency Cepstral Coefficients (Coeficientes Cepstrales en Frecuencia Mel)
MIR	Music Information Retrieval (Recuperación de Información Musical)
MIREX	Music Information Retrieval Evaluation eXchange
ML	Machine Learning (Aprendizaje Automático)
MSE	Mean Squared Error (Error Cuadrático Medio)
NLP	Natural Language Processing (Procesamiento del Lenguaje Natural)
PCA	Principal Component Analysis (Análisis de Componentes Principales)
PMEmo	Dataset for Music Emotion Recognition
ResNet	Residual Network (Red Residual)

Continúa en la siguiente página

– Continuación de Abreviaturas y Siglas –

Abreviatura/Sigla	**Significado**
RMSE	Root Mean Squared Error (Raíz del Error Cuadrático Medio)
RSE	Relative Squared Error (Error Relativo al Cuadrado)
SAM	Self-Assessment Manikin
SE	Squeeze-and-Excitation
SGD	Stochastic Gradient Descent (Descenso de Gradiente Estocástico)
STFT	Short-Time Fourier Transform (Transformada de Fourier de Tiempo Corto)
SVM	Support Vector Machine (Máquina de Soporte Vectorial)
SVR	Support Vector Regression (Regresión de Soporte Vectorial)
t-SNE	t-Distributed Stochastic Neighbor Embedding
VA	Valence-Arousal (Valencia-Activación)

Resumen

La música, elemento fundamental en la vida cotidiana, impacta profundamente a la sociedad debido a su capacidad para transmitir y evocar emociones. El estudio de esta relación ha consolidado el campo interdisciplinario del Reconocimiento de Emociones en la Música (MER). Tradicionalmente, los sistemas MER se han centrado en el análisis de características acústicas, a menudo omitiendo aspectos teóricos cruciales como el contexto armónico de una obra, el cual está intrínsecamente ligado a la expresión emocional.

El presente trabajo aborda esta limitación mediante el desarrollo de un sistema MER multimodal que integra dos fuentes de información complementarias, utilizando los conjuntos de datos unificados de PMEmo y DEAM. Para el análisis acústico, se emplea una arquitectura ResNetSE como extractor de características a partir de espectrogramas. De forma paralela, el contexto armónico se modela codificando las secuencias de acordes con modelos Word2Vec. Finalmente, un modelo BiLSTM fusiona ambas representaciones para realizar la predicción final.

El modelo de fusión propuesto alcanza un rendimiento robusto, con un error RMSE de 0.1087 y un R^2 de 0.5087 para la dimensión de *valence*, y un RMSE de 0.1271 y un R^2 de 0.5232 para la dimensión de *arousal*. Estos resultados demuestran que un enfoque multimodal, que combina la textura acústica con el contexto armónico, simula de manera más fiel el proceso de análisis humano. Se concluye que la percepción emocional no depende de un único componente, sino de la interacción de múltiples factores como el timbre, la dinámica, el ritmo y la estructura armónica, validando así la superioridad de la estrategia de fusión.

Abstract

Music, a fundamental element of daily life, profoundly impacts society through its ability to convey and evoke emotions. The study of this relationship has established the interdisciplinary field of Music Emotion Recognition (MER). Traditionally, MER systems have primarily focused on the analysis of acoustic features, often overlooking crucial theoretical aspects such as the harmonic context of a musical piece, which is intrinsically linked to emotional expression.

This work addresses this limitation by developing a multimodal MER system that integrates two complementary sources of information, utilizing the unified PMEmo and DEAM datasets. For the acoustic analysis, a ResNetSE architecture is employed as a deep feature extractor from spectrograms. In parallel, harmonic context is modeled by encoding chord sequences using Word2Vec models. Finally, a BiLSTM model fuses both representations to perform the final emotion prediction.

The proposed fusion model achieves a robust performance, yielding an RMSE of 0.1087 and an R^2 of 0.5087 for the valence dimension, and an RMSE of 0.1271 and an

R^2 of 0.5232 for the arousal dimension. These results demonstrate that a multimodal approach, which combines acoustic texture with harmonic context, more faithfully simulates the human analysis process. We conclude that emotional perception in music does not depend on a single component, but rather on the complex interaction of multiple factors such as timbre, dynamics, rhythm, and harmonic structure, thus validating the superiority of the proposed fusion strategy.

Índice

Índice de figuras

Índice de cuadros

1. Introducción

La música es un elemento profundamente arraigado en la cotidianidad que impacta de manera notable diferentes aspectos de la sociedad, desde el apartado cultural hasta el político. La música estimula capacidades cognitivas y emocionales, de ahí que sea fuente de inspiración en múltiples investigaciones. Un ejemplo específico es la comprensión de la relación que existe entre la música y las emociones humanas [1], [2], [3].

El interés y la curiosidad en la relación musical-emocional no son algo nuevo, pues, al menos, desde el siglo pasado han existido esfuerzos por explicar cómo se relacionan ciertos componentes de la música con la activación de determinadas emociones [4], [5].

La búsqueda de determinar en qué radica el significado de una obra musical y entender cómo sus componentes provocan emociones ha involucrado áreas como la filosofía, la psicología y la teoría musical [6], [7]. Incluso, este interés se ha extendido al sector científico y tecnológico, creando así campos multidisciplinarios con el fin de abordar este problema.

Tal es el caso del campo *MER (Music Emotion Recognition)*, que en español puede entenderse como Reconocimiento de Emociones en la Música. Este campo emplea el conocimiento de áreas como las ciencias computacionales, el cómputo afectivo, la neurociencia, la psicología y la sociología para analizar características extraídas de la música e identificar qué emoción puede provocar una obra [8], [9].

En el campo MER, el eje central es el análisis de características extraídas de obras musicales, por lo general, a partir de archivos de audio. Es por ello que MER es considerado una tarea secundaria del campo *MIR (Music Information Retrieval)*,que por su traducción al español es captura (o recuperación) de Información Musical. El campo MIR se enfoca en la obtención de información de archivos de audio musicales por medio de técnicas de procesamiento y análisis de señales digitales [10], [11], [12].

1.1. Planteamiento del problema

Desde el campo MER, la comunidad científica ha identificado una serie de barreras que obstaculizan el éxito en la labor del reconocimiento de emociones.

De manera general, determinar qué emoción será transmitida por medio de la música es una labor compleja que depende de múltiples factores. Por mencionar algunos: características acústicas inherentes a la señal de audio, el contexto de la obra o factores externos propios del usuario, tales como su contexto social, cultural o emocional, así como sus gustos musicales [6], [13].

La labor de identificar qué emoción percibe alguien ante un estímulo musical es un problema multivariable. La relación entre las variables y la emoción final puede caer en la subjetividad, pues lo que para una persona resulta relevante para otra puede ser insignificante.

Ahora bien, en la tarea de identificar emociones se manejan dos vertientes: la percepción y la inducción. La inducción busca producir emociones a partir de un escenario

propicio. A menudo, los estímulos elegidos para esta labor tienen un vínculo con el usuario, mientras que la percepción solo se centra en las características propias del estímulo [14], [15].

Otra de las barreras es la elección de la taxonomía o del modelo que se emplea para representar las emociones, es decir, la manera de cuantificarlas o categorizarlas. En este sentido, existen dos visiones generales: la representación categórica y la dimensional. En los modelos categóricos, se busca representar una emoción como una variable discreta y categórica, centrándose en la asignación de adjetivos como felicidad, ira o tristeza [5], [16], [17]. Por otra parte, los modelos dimensionales se basan en la idea de entender las emociones como elementos formados por dos ejes, el *valence* (valencia) y el *arousal* (activación). De esta forma, una emoción tiene un valor numérico compuesto por un par ordenado [18], [19], [20].

La elección de una taxonomía que se adecue a los objetivos del problema es de vital importancia, pues, dependiendo de la elección, el problema puede ser abordado como una clasificación múltiple o una regresión [15]. Además, se debe tener en cuenta la desventaja de ambos modelos. Para los categóricos, la desventaja radica en la pobre capacidad de representar emociones complejas, mientras que para los modelos dimensionales se hace complejo el interpretar los valores [21].

El nivel de reconocimiento de emociones radica, en gran parte, en la taxonomía elegida y en el nivel en el que a una obra se le asigna una emoción. Es decir, dado que una canción u obra musical es un elemento temporal y que presenta variaciones de principio a fin, se suele etiquetar de manera estática o de manera dinámica.

De esta forma, se tienen los siguientes enfoques: *Song-level (categórico y dimensional)*: Asigna una emoción a partir de un solo segmento representativo de la obra. *MEVD (categórico y dimensional)*: En este enfoque la asignación de una emoción no contempla solo un segmento representativo, sino que evalúa las variaciones emocionales a lo largo de toda la obra [22].

Los conjuntos de datos también suelen estar separados según el enfoque de reconocimiento (Song-Level o MEVD) y la taxonomía. Aunado a esto, existen diferentes metodologías para generar anotaciones emocionales, pues en ocasiones suelen seguir metodologías basadas en la psicología y la neurociencia o simplemente tomar las etiquetas emocionales de rankings o listas en internet [23]. Todo esto dificulta el poder trabajar con varios conjuntos al mismo tiempo.

Finalmente, el enfoque principal de las tareas de MER se basa en la extracción y análisis de características de bajo o medio nivel, las cuales se obtienen directamente de la señal de audio. No obstante, la manera de representar estas características está ligada a otras tareas como el reconocimiento de voz o la separación de canales de audio [10], [13].

1.2. Justificación

El reconocimiento de emociones en la música ha encontrado sitio en diversas aplicaciones. Un ejemplo claro se encuentra en los servicios de streaming, los cuales han

implementado el reconocimiento de emociones en sistemas de recomendación [12]. De igual forma, se ha explorado su aplicación en beneficio de las personas con discapacidad auditiva, por ejemplo, mediante la generación de subtítulos que describen la música de las películas [24].

Además de la aplicación directa de los sistemas de reconocimiento, la música, gracias a su relación con las emociones, se usa como apoyo en tratamientos de afecciones como la depresión [25] y en la terapia de trastornos del espectro autista [26]. Del mismo modo, su capacidad para activar la memoria y evocar recuerdos [27] resulta beneficiosa en el tratamiento de pacientes con alzhéimer. La investigación sobre los efectos de la música se extiende incluso a respuestas fisiológicas directas, como su aplicación para el alivio del dolor en neonatos durante procedimientos menores [28].

Teniendo en cuenta lo antes mencionado, la música es un gran apoyo frente a los desafíos de la salud mental. La capacidad de generar una diversa gama de sentimientos afectivos ayuda a contrarrestar efectos perjudiciales del estrés, la ansiedad y la depresión. Dicha capacidad de evocar emociones convierte a la música en un recurso de apoyo valioso para enfrentar problemas de salud mental.

Ahora bien, la tarea del MER se abordó inicialmente con enfoques basados en las matemáticas, la física y la estadística. Sin embargo, el interés por aplicar nuevas estrategias ha llevado a implementar Inteligencia Artificial en las tareas de reconocimiento de emociones y tanto el *Machine Learning (ML)* [21], [29], [30], [31], [32] como el *Deep Learning (DL)*[23], [24], [33], [34], [35] se han consolidado como los enfoques principales para desarrollar sistemas MER.

Dentro del ML y el DL, el problema del reconocimiento de emociones musicales se resuelve a partir del análisis de características extraídas de archivos de audio. No obstante, no se ha llegado a un consenso sobre qué característica es la indicada para realizar la tarea. En ocasiones, una característica en concreto puede ser mejor que el resto, pero en otros experimentos la que menos éxito arrojaba de pronto es la más significativa.

Por ende, es importante contemplar conceptos como la armonía y la teoría musical, pues diferentes componentes de la estructura armónica de una obra musical juegan un papel crucial en la expresión de emociones [5], [6]. Aunado a lo anterior, se ha encontrado que la música es capaz de activar emociones por medio de inhibir y concluir las expectativas que una obra genera en el oyente [4], lo cual se ve reflejado a través de conceptos teóricos como las cadencias y la resolución de progresiones [7], [36].

Sumado a ello, la posibilidad de expresar notaciones de acordes mediante caracteres ayuda a establecer cierta similitud con el lenguaje natural, pues, dado que una forma de representar sucesiones de acordes es con conjuntos de caracteres alfanuméricos, es posible aplicar métodos de *Procesamiento del Lenguaje Natural* (por sus siglas en inglés, *NLP*) [37], [38]. De esta forma, se puede ampliar el enfoque tradicional de los sistemas MER, permitiendo la continua mejora de estos sistemas.

La continua mejora de los sistemas MER enfrenta una serie de adversidades. Por ende, es importante contemplar nuevos conceptos para solucionar estos desafíos. Por ejemplo, a partir de la importancia de las estructuras armónicas con las emociones

en la música, es necesario continuar explorando y mejorando estas opciones, pues si bien existen trabajos al respecto, no toman consideraciones como la importancia de la posición y el contexto que rodea a un acorde.

2. Antecedentes

Como se ha mencionado, el interés por parte de la comunidad científica en la música no es reciente. Durante los últimos años, este interés ha crecido y el número de publicaciones en los campos de MIR y MER ha aumentado [12].

Esto se debe, en gran parte, a que la comunidad científica ha fomentado las investigaciones en estos campos, desarrollando concursos importantes como el **AMC** o el **MediaEval**, por mencionar algunos.

En 2007 la evaluación *Audio Mood Classification (AMC)* se incluyó por primera vez en *MIREX (Music Information Retrieval Evaluation eXchange)*, organizado por la *International Society for Music Information Retrieval (ISMIR)* [39]. Su objetivo es proporcionar un punto de referencia estándar para la clasificación automática de estados de ánimo en fragmentos de audio. Desde entonces, AMC se celebra anualmente y ha ido creciendo tanto en número de participantes como en mejoras de rendimiento.

De la misma manera, en 2013, *MediaEval (Multimedia Evaluation Benchmark)*, que se ha convertido en una prestigiosa iniciativa de benchmarking para evaluar algoritmos y tecnología en la recuperación, acceso y exploración de archivos multimedia, realizó la primera convocatoria para la clasificación de emociones musicales bajo los enfoques song-level y MEVD [40]. Como resultado de esta convocatoria, se construyó un conjunto de datos con 1000 clips de audio con sus respectivas etiquetas, el cual sirve de base para el conjunto de datos elaborado por el *DEAM (Database for Emotional Analysis in Music)* [41].

Parte fundamental en las tareas de MER es la elección de un modelo de representación de emociones, los cuales suelen agruparse en dos ramas: dimensionales y categóricos. A lo largo de los años, se han propuesto diversos modelos, como el de Hevner en 1963, J. Russell en 1980 o Thayer en 1990.

Hevner planteó la representación de emociones por medio de una lista de 67 adjetivos, los cuales asoció con emociones. La lista se encuentra organizada en 8 grupos llamados: solemne, triste, soñadora, tranquila, elegante, alegre, emocionada y poderosa. Este modelo sostiene que ciertas características de la música evocan ciertas emociones [5].

Por otro lado, en 1980, J. Russell, a partir de entender los estados afectivos como elementos conformados por dos ejes, propuso un modelo con dos dimensiones [19]. La primera dimensión, denominada valencia, determina qué tan placentera o poco placentera es una emoción, mientras que la segunda, activación, indica el grado de intensidad [18].

A menudo es fácil observar cómo las anotaciones valence y arousal pueden ser discretizadas a partir de dividir el plano bidimensional en cuadrantes, en donde cada

cuadrante representa el estado del valence o arousal, dando la posibilidad de ubicar las anotaciones en los cuadrantes: Arousal Alto - Valence Positivo, Arousal Alto - Valence Negativo, Arousal Bajo - Valence Positivo y Arousal Bajo - Valence Negativo [10], [11], [21], [22].

Además de estos modelos, existe otro que se enfoca en la representación y evaluación de emociones inducidas por música. En 2008, Zentner y Scherer, a través de una serie de estudios, propusieron GEMS, escala emocional-musical de Ginebra. Esta es una herramienta para evaluar las emociones inducidas por obras musicales. GEMS se compone de una lista de 40 emociones. Estas están agrupadas en 9 factores, los cuales representan el rango de emociones que una obra musical puede evocar [1].

Existen otras representaciones, aunque menos implementadas, por ejemplo: la distribución de probabilidades, pares de antónimos [8] y el ranking de emociones [15].

En general, la tarea del reconocimiento de emociones se puede agrupar en dos grandes enfoques, el reconocimiento estático y el dinámico, los cuales a su vez se pueden clasificar en:

- reconocimiento de emociones musicales de manera *categórica a nivel canción (Song-level categorical MER)*
- reconocimiento de emociones musicales de manera *dimensional a nivel canción (Song-level dimensional MER)*
- Detección de variaciones emocionales musicales *(Music Emotion Variation Detection)*

Song-level categorical MER. Es un enfoque de clasificación, es decir, la representación de emociones es categórica. La clasificación de la obra se realiza sobre un segmento de la misma.

En 2008, Pao et al. [29] sugirieron la clasificación de clips de audio, extraídos de los coros de canciones populares, a través de un modelo KNN mejorado, añadiendo pesos discretos entre las distancias de los vecinos. Demostraron que el algoritmo superó los resultados de modelos basados en SVM y KNN, alcanzando más del 96 % de exactitud en la labor de clasificar a qué cuadrante del modelo Thayer pertenece el extracto de audio.

En 2013 [42], se propuso un modelo basado en AdaBoost y *decision stump* para realizar la clasificación de canciones en 14 categorías, alcanzando un 79 % de éxito en promedio.

En 2014, Akhilesh K Sharma et al. [30], realizaron la clasificación de ragas (música tradicional de la India) por medio de evaluaciones estadísticas, tomando como base los algoritmos de Naïve Bayes y *EM (Expectation Maximization)*.

Song-level dimensional MER. La diferencia de este enfoque con el anterior es que se considera el problema como una regresión y trabaja con datos continuos.

En [31] se presenta un modelo generativo basado en modelos de mezcla gaussiana (GMM) para la predicción de valores de valence y arousal en obras musicales, a partir

de las bases de datos de MER60 y DEAP. Los autores compararon su propuesta contra modelos SVR y mejoraron los resultados en un 71.5 % y 40.3 %.

En [32] se propuso un sistema basado en SVR para el reconocimiento de emociones musicales. Se basaron en el modelo de Thayer y obtuvieron un 94.55 % de exactitud.

En 2013, Markov y Matsui [43], [44], en el marco del taller internacional de MediaEval, desarrollaron un sistema de reconocimiento de emociones musicales utilizando procesos de regresión gaussianos GPR. Tomaron en cuenta el reconocimiento estático y dinámico, no obstante los resultados para la detección dinámica no fueron satisfactorios.

Tras no haber alcanzado su meta en el reconocimiento dinámico, Markov y Matsui, en 2014, para el taller MediaEval de ese año, propusieron un sistema basado en procesos gaussianos y filtros de Kalman [45].

Music Emotion Variation Detection. La predicción no se realiza sobre un segmento representativo, sino que se evalúan las variaciones emocionales a lo largo de toda la obra. De acuerdo con la revisión [15], la primera vez que se propuso la idea de observar las variaciones emotivas en una canción fue en [46].

L.Lu et al. [46] propusieron el reconocimiento de emociones en distintas obras de música clásica usando modelos de mezcla gaussianos GMM con un total de 16 mezclas. Los segmentos representativos fueron clasificados en 4 estados de ánimo: frenético, satisfactorio, depresivo y eufórico.

En 2016 [47], mediante un modelo basado en SVR, realizaron el reconocimiento dinámico de emociones musicales. En su experimentación, cada canción tenía 60 anotaciones de valance y 60 anotaciones de arousal. De esta manera, implementaron 2 escalas de anotaciones, es decir, una canción tenía una anotación emocional global, y a su vez cada sección de la canción también contaba con una anotación individual.

El enfoque principal de las tareas de MER se basa en la extracción y análisis de características de bajo o medio nivel, aquellas que se pueden extraer directamente de la señal de audio, como el ritmo, el color tonal o los armónicos, entre otros [10], [13]. Estas características alimentan los modelos de DL o ML, los cuales, a su vez, efectúan el análisis que permite llevar a cabo la predicción de la emoción.

En [47], como característica de entrada al modelo, usan señales de audio MFCC. De igual manera, en el trabajo [44], también se usaron MFCC. Mientras que en [29] usan anchura del timbre, volumen, centroide espectral, disonancia espectral y otras señales de audio. Por otro lado, en [43], además de MFCC, usan otras señales de audio como el factor cresta espectral y descriptores estadísticos del espectro. En [31] se utilizan descriptores armónicos. Además, en [30], [32], [42], [46], también se usan señales de audio como característica acústica de entrada.

Por su parte, Greer ha buscado la manera de contrarrestar algunos defectos que las técnicas tradicionales tienen. Para ello, ha propuesto un nuevo modelo capaz de generar características de audio y musicales, utilizando aprendizaje autosupervisado y aprendizaje por cruce de dominio, todo ello por medio de un transformer encoder bidireccional multicapa con mecanismos de autoatención [48].

Además de la información y características extraídas directamente de las señales de audio en problemas referentes al procesamiento de señales también se utiliza otra ma-

nera de representar la información de una señal de audio, por medio de espectrogramas. Los espectrogramas son una representación visual de la señal de audio que muestra la distribución de la energía de las frecuencias a lo largo del tiempo. El resultado se dibuja como un mapa de calor donde el eje horizontal es el tiempo, el vertical la frecuencia y el color indica la intensidad de la energía.

En la tabla de antecedentes, se encuentra la información relevante de investigaciones que incluyeron espectrogramas en la elaboración de sistemas MER.

La comunidad científica busca constantemente integrar el conocimiento punta de lanza, por ejemplo, en los trabajos de [34], [49] se diseñan arquitecturas basadas en *transformers*.

Los mecanismos de atención suelen estar enfocados en las características espaciales y se suelen aplicar en espectrogramas. No obstante, también se ha explorado la idea de trabajar con bloques *squeeze-and-excitation (SE)*, que son una especie de mecanismos de atención pero enfocados en la información de los canales. Esto se ha aplicado a redes neuronales convolucionales, permitiéndoles identificar y priorizar automáticamente los canales más importantes de un espectrograma.

En el trabajo de [50] se hizo uso de bloques SE en la predicción del nivel de depresión en el habla. Por otro lado, en el trabajo [51] se emplearon bloques SE para la tarea de detección y localización de eventos sonoros. Esto permitió que su modelo se enfocara tanto en los canales más importantes como en las regiones de tiempo-frecuencia más significativas del espectrograma, mejorando la identificación de las clases de sonido.

En el trabajo [52], los bloques SE se utilizaron para la tarea de MER. Los autores integraron la atención SE dentro de su Módulo de Aprendizaje de Características Temporales (TFLM). Su función era analizar las características extraídas de los espectrogramas para aprender la importancia de cada canal y así poder potenciar el peso de las características más relacionadas con la emoción, mientras suprimía las que no contribuían significativamente. Los resultados demostraron la eficacia de este enfoque, ya que su modelo (denominado ADFF) logró una mejora relativa del 10,43 % en el valence y del 4,82 % en el arousal en la puntuación R^2 en comparación con otros modelos del estado del arte.

Existen a su vez trabajos en donde se aborda el problema del reconocimiento de emociones en música mediante estrategias multimodales, los cuales suelen incorporar información contextual externa a las señales de audio en conjunto con enfoques tradicionales.

En este sentido, Panda et al. [10] han desarrollado un trabajo notable, pues se establece la importancia de encontrar características enfocadas en las emociones de la música, además de que muestra una detallada explicación de las características existentes más importantes para las tareas de MER. El estudio no se limita a esta revisión, sino que el autor desarrolla una serie de características enfocadas en el reconocimiento de emociones, y realiza varias pruebas con modelos SVM (máquinas de soporte vectorial).

En [42], además de utilizar las señales de audio como característica acústica de entrada, también usan las letras de las canciones como una entrada del modelo.

En [34] combina características de bajo y medio nivel con la letra de canciones

a través de un enfoque multimodal, en donde el análisis lo realizan modelos DL en conjunto con técnicas de NLP. Como características tradicionales utiliza espectrogramas y el análisis lo efectúan redes CNN. En el caso de la letra de canciones emplean diversos métodos NLP, obteniendo mejores resultados con BERT. En la métrica de exactitud se alcanzó un 94,58 % tras realizar la fusión de los modelos.

En [33] también combina características de bajo nivel con letras. Por medio de un sistema multimodal realiza el análisis de la información, la cual se obtiene de un conjunto de datos de 2000 canciones extraídos de la API de Last FM. Tras la fusión de los modelos se alcanzó un 78 % de exactitud.

En 2019 Greer publicó dos artículos en donde propone que mediante una representación de acordes y letras, usando vectores compartidos, las tareas de clasificación de géneros musicales pueden obtener mejores resultados [37]. No obstante, en tareas de MER, si bien logra resultados mejores que otros modelos, estos no los sobrepasan por mucho.

En cuanto a características de audio, Greer ha buscado la manera de contrarrestar algunos defectos que las técnicas tradicionales tienen, para ello, ha propuesto un nuevo modelo capaz de generar características de audio y musicales, utilizando aprendizaje autosupervisado y aprendizaje por cruce de dominio, todo esto por medio de un *transformer encoder* bidireccional multicapa con mecanismos de autoatención [48].

En [53] el autor realizó la detección de acordes tanto en archivos MIDI como en archivos de audio comunes. Una vez obtenidos los acordes, procedieron a codificarlos de acuerdo a su posición en la escala (I, II, III, etc.). De este modo se construyó una matriz de transición de acordes, en donde se almacenaban las transiciones comunes entre cada acorde. La labor se centra en la predicción de valores de valence y arousal, y para ello se utilizan modelos de machine learning de regresión, en específico SVR (Support Vector Regression) y LASSO (Least Absolute Shrinkage and Selection Operator). En el caso del conjunto de datos MIDI, la incorporación de progresiones 2-chord redujo el MSE de valencia de 0.96 a 0.71, mientras que en el conjunto de audio también se observaron mejoras (por ejemplo, de 1.39 a 1.24 con MIRToolbox).

De manera parecida, en el trabajo [54] se creó una base de datos donde se relacionó un conjunto de acordes con emociones. Luego, tras la extracción de características con la transformada rápida de Fourier y métodos estadísticos (FFT y STAT), se identificaron los acordes de cada audio. El reconocimiento de emociones se realizó por medio del cálculo de la distancia euclidiana y la correlación.

Ahora bien, los métodos de NLP no solo se han implementado en el análisis de letras. En [38] se muestra que mediante métodos de embeddings predictivos como lo son *word2vec* se pueden capturar relaciones teóricas entre acordes. Por su parte, Greer [37] propuso que mediante la representación de acordes y letras por medio de vectores compartidos las tareas de MER pueden obtener mejores resultados, aunque los resultados no sobrepasan por mucho a los ya existentes.

Además de todo lo anterior, existen investigaciones que incorporan señales mioeléctricas en sistemas MER [9], además de usar espectrogramas, acompaña el reconocimiento de emociones con imágenes EEG. Aunque este tipo de señales se utilizan cuando el

trabajo se centra en la inducción de emociones más que en la percepción [3], [14], [55], [56], [57].

Cuadro 2: Tabla de Trabajos del Estado del Arte

Aporte	**B.D.**	**Caract.**	**Modelo**	**Tax.**	**Res.**
[23] MER para IoT: optimiza la extracción de caract. locales/globales y la expresividad de MFCC (2021).	A partir de listas de Internet. 637 canciones	Bajo-Nivel: MFCC	GAN con fusión de doble canal	Cat.	93.4 % precisión (en promedio)
[11] Metodología hibrida y B.D propia para música turca para capturar simultáneamente relaciones espaciales (2021)	Propia. 124 canciones tradicionales turcas.	Bajo-Nivel: MFCC, Energías Log-Mel, caract. acústicas estándar.	CLDNN	Cat. (3 clases V-A)	99.19 % de precisión
[49] Arquitectura segmentada en dos etapas: aprendizaje no supervisado en característica y clasificación supervisada de emociones (2022).	PMEMO: 767 canciones. AllMusic: 900 clips.	Medio-Nivel: Espectrogramas Log-mel	BiLSTM y Autoencoder CNN	Cat. (D V|A)	V: 79.01 % A: 83.62 % (acc)
[21] Optimización de modelos clásicos con técnicas de metaheurística (2021).	MEMD. 1744 canciones	Bajo-Nivel: LLDs (descriptores acústicos)	NN BP (opt. ABC)	Dim.	V: RMSE 0.1066 R^2 0.4606; A: RMSE 0.1322 R^2 0.6687
[52] Arquitectura MER end-to-end con atención SE y fusión jerárquica espacio-temporal (2022).	PMEMO. 767 canciones	Medio-Nivel: Espectrogramas log-mel	VGG16 adaptado + SE attention + BiLSTM	Dim.	V: RMSE 0.2379 R^2 0.4575; A: RMSE 0.2213 R^2 0.6393

Continúa en la siguiente página...

Cuadro 2: Continuación del Cuadro 2

Aporte	**B.D.**	**Caract.**	**Modelo**	**Tax.**	**Res.**
[2] Contempla la importancia del rol de cada voz en la música mediante la separación de fuentes (2020).	PMEMO. 767 canciones	Medio-Nivel: Espectrogramas log-mel	Demucs MSS, VGG16	Dim.	V: RMSE 0.2466 R^2 0.4143; A: RMSE 0.2285 R^2 0.6100
[24] Reconocimiento de emociones musicales usando segmentos cortos y bases de datos científicas (2023)	The musical excerpts y The film music excerpts , 94 fragmentos de audio.	Medio-Nivel: Espectrograsmas STFT, MEL y CQT	CNN	Cat.	79 % (resultado general con CQT)
[22] Marco para el reconocimiento dinámico de emociones musicales (valores VA) mediante un modelo de fusión CNN-BiLSTM (2020)	The 1000 songs	Medio-Nivel: Espectrogramas Mel y Cochleogram	CNN y BiLSTM	Dim.	V: RMSE 0.07; A: 0.06

Notas:

Encabezados	**B.D.** = Base de Datos; **Caract.** = Características; **Tax.** = Taxonomía; **Res.** = Resultados.
En Tax.	**Cat.** = Categórica/o; **Dim.** = Dimensional; para formatos como (D V|A) o (3 clases V-A): **V** = Valence, **A** = Arousal, **D** = Discreto.
En Modelo	**NN BP (opt. ABC)** = Red neuronal de retropropagación optimizada con Colonia de Abejas Artificiales.

3. Fundamentación teórica

3.1. Visiones generales de las emociones

Desde un punto de vista psicológico, una emoción es una respuesta que tiene el organismo de los seres humanos ante los estímulos que nos rodean [16], teniendo como finalidad preservar la supervivencia del individuo. Estas experiencias siempre se dan de

la mano de cambios fisiológicos [17].

La representación de la emoción es parte de las bases del campo de MER. Históricamente, han surgido dos grandes marcos teóricos para abordar este tema. El primero concibe la emoción como un conjunto de estados discretos y distintos (comúnmente agrupados por adjetivos), mientras que el segundo, más contemporáneo, la describe como una estructura integrada y sistemática definida por un número reducido de dimensiones fundamentales.

El concepto de representar emociones categóricamente parte de la idea de que emociones como la felicidad o la tristeza son categorías fundamentalmente distintas. Un ejemplo de esta perspectiva teórica es el trabajo de Kate Hevner [5]. Su investigación se fundamenta en el supuesto de que existe un simbolismo sistemático en la música, donde elementos estructurales específicos son capaces de expresar emociones definidas y conceptos sentimentales.

Investigaciones posteriores comenzaron a cuestionar la idea de que los estados afectivos fueran independientes. Estos trabajos proponían que, en lugar de ser factores separados, las emociones están interrelacionadas de una manera altamente sistemática. Esta observación llevó al desarrollo de una teoría estructural del afecto, donde las emociones se definen por su posición dentro de un espacio compartido [17], [19].

La formulación más influyente de esta teoría es el Modelo Circumplejo del Afecto [19]. La tesis central de Russell propone representar la estructura cognitiva del afecto por medio de un círculo en un espacio bidimensional. Las bases de esta teoría se encuentran en el hecho de que el espacio emocional está definido por dos dimensiones bipolares, *Placer-Displacer* y *Excitación-Sueño*, además de contemplar que las emociones no son puntos aislados, sino que se organizan en el espacio circular y cada emoción no se define como una categoría, sino más bien por su ubicación dentro de este plano, como una combinación de los valores de *placer* y *excitación*.

La teoría Thayer [20], ofrece una explicación funcional y biológica para la estructura dimensional del afecto. En lugar de comenzar con un mapa cognitivo, Thayer postula que la experiencia afectiva es una manifestación consciente de sistemas biológicos fundamentales que han evolucionado para la supervivencia. Su modelo se centra en la interacción de dos sistemas de activación (arousal) principales: *Excitación Energética (Energetic Arousal)* y *Excitación Tensa (Tense Arousal)*.

El continuo trabajo en la representación emocional por medio de un plano dimensional, ha llevado no solo a la creación de herramientas de evaluación estandarizadas y no verbales, como el *Self-Assessment Manikin (SAM)*, sino también a la consolidación de una terminología convencional para sus ejes fundamentales. Si bien los trabajos fundacionales usaban términos como placer-displacer, la convención moderna, adoptada en la mayoría de los modelos dimensionales, se refiere a estos ejes como **valence** (el continuo de placer, de positivo a negativo) y **arousal** (nivel de activación, de alta a baja). Estos dos ejes, a menudo complementados por una tercera dimensión de dominancia, forman el marco estándar sobre el cual se representa y mide la respuesta emocional, solidificando el paradigma dimensional en la investigación actual [16], [17], un ejemplo gráfico de como lucen estos modelos se encuentra en la figura 1.

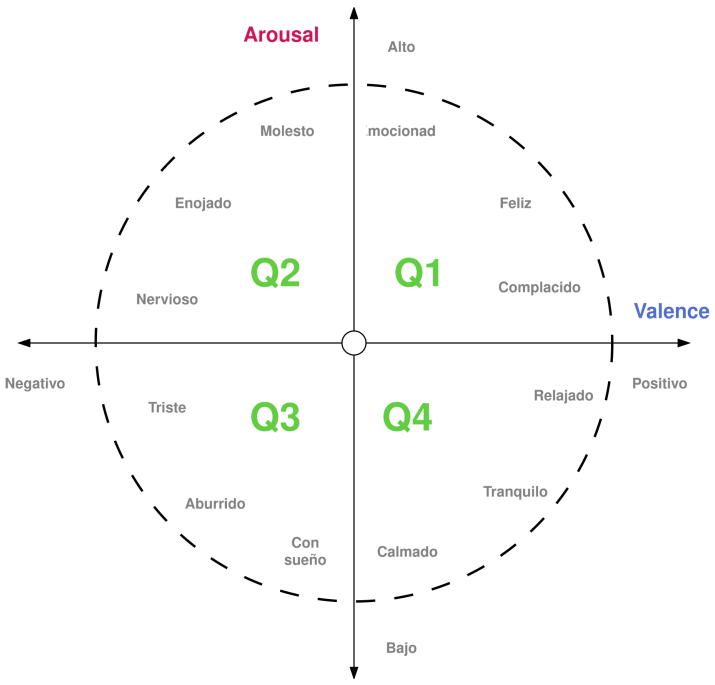

Figura 1: Modelo dimensional *Valence Arousal*; adaptada de [10], [21], [22]

Con frecuencia es complicado entender qué emoción se activa en determinados momentos, y esta tarea aumenta en complejidad cuando se trata de terceros. Aun así, existen respuestas que permiten identificar o medir una emoción, pues al efectuarse suelen también presentarse respuestas fisiológicas como el aumento de la presión, variaciones en el ritmo cardíaco, etc., cambios conductuales como tics nerviosos o en la manera con la que nos expresamos y, finalmente, cambios cognitivos. Por un lado, los cambios fisiológicos pueden ser observados usando tecnología, por ejemplo, los electroencefalogramas (EEG) para medir los cambios de la actividad eléctrica en el cerebro. Del mismo modo, los cambios conductuales pueden ser percibidos por mera observación del individuo. A su vez, medir los cambios en los procesos cognitivos solo es posible si el individuo lo indica. Comúnmente, herramientas como los tests y cuestionarios son utilizados para esta labor [16], [17].

El SAM (Self-Assessment-Manikin) es una herramienta que busca determinar qué emoción se activa en un individuo ante un evento o estímulo [16]. El SAM es compatible con la visión de representación dimensional, permitiendo al usuario expresar qué emoción percibe midiendo el grado de *arousal*, *valence* y *dominance*. De esta forma, el SAM es una encuesta no verbal y basada en imágenes. Para representar el grado de placer (positividad) se utilizan diversas figuras que representan un cambio gradual partiendo de la felicidad hasta la tristeza. Para representar la dimensión de *arousal* se representa a través de diversas figuras abrumadas. Finalmente, la dimensión de *dominance* se representa por medio de un cambio gradual en el tamaño de las figuras, partiendo de una figura pequeña hasta una grande [17].

Según Meyer en su trabajo [7] , las emociones surgen cuando se inhibe o detiene una "tendencia", entendida como un patrón de respuesta automática basado en experiencias y conocimientos previos. Frente a un estímulo inesperado, por ejemplo, un perro que se cruza en nuestro camino, el cerebro genera un escenario posible y, si la realidad difiere de lo anticipado, la tensión acumulada alivia y se activa la emoción correspondiente. Así, todas las tendencias, conscientes o no, pueden concebirse como expectativas que, al cumplirse o frustrarse, moldean nuestra respuesta emocional.

En el ámbito musical, este mecanismo de expectativas se explica por la capacidad del oyente para anticipar progresiones armónicas: cuando la resolución de un acorde coincide con lo previsto, sentimos complacencia si se desvía, percibimos tensión y emoción [7]. Steinbeis [4] refuerza esta idea al señalar que las predicciones armónicas, construidas a partir del bagaje cultural o vivencial del oyente, determinan la manera en que se experimenta una obra musical. En conjunto, Meyer y Steinbeis muestran que la percepción emocional en la música depende tanto de la inhibición de tendencias como del grado en que se satisfacen o rompen las expectativas armónicas.

3.2. Teoría musical

En el sistema de afinación temperada, predominante en la música occidental, la octava se divide en doce sonidos equidistantes, separados por intervalos iguales denominados semitonos o medios tonos [58], [59]. Dentro de este marco, surge el fenómeno de la enarmonía, que se presenta cuando dos notas diferentes en notación reciben el mismo valor acústico o altura sonora. Estas notas, conocidas como sonidos enarmónicos, representan una misma frecuencia aunque se escriban de forma distinta [36]. Este fenómeno es consecuencia tanto del sistema de afinación como de las convenciones de notación musical y permite, por ejemplo, que una misma tecla del piano pueda representar indistintamente un Do sostenido ($C\sharp$) o un Re bemol ($D\flat$).

Las escalas constituyen un elemento primordial en la teoría musical, definidas como sucesiones ordenadas de sonidos que siguen un patrón de intervalos específico. En la música occidental, la escala mayor es fundamental, caracterizándose por la secuencia de tonos (T) y semitonos (ST): $T-T-ST-T-T-T-ST$. Este patrón puede ser aplicado a cualquiera de los 12 sonidos del sistema cromático temperado, generando así su escala mayor. Cada escala presenta una jerarquía sonora centrada en la nota tónica, que actúa como el núcleo gravitacional de la tonalidad, dando contexto a las demás notas. La tonalidad organiza estas notas en grados identificados por numeración romana, siendo la tónica (primer grado) la que nombra la tonalidad, la cual puede presentarse en distintos modos, siendo los más comunes los modos mayor y menor. Cada grado, además, cumple una función armónica específica con denominaciones particulares, susceptibles a variaciones según el modo en que la tonalidad se manifiesta [36].

El anillo $\mathbb{Z}_{12}$, tal como se describe en [60], permite modelar matemáticamente las escalas musicales mediante aritmética modular. Así, cada nota se representa como un número entero módulo 12, y una escala se construye como una sucesión de intervalos. Por ejemplo, la escala mayor responde al patrón $\{2, 2, 1, 2, 2, 2, 1\}$, donde 2 equivale a

un tono y 1 a un semitono. Aplicando este patrón desde una nota base $x \in \mathbb{Z}_{12}$, se obtiene la escala correspondiente, sumando cada intervalo sucesivamente módulo 12. Así, partiendo de $x = 0$ se obtiene la escala de C mayor; desde $x = 7$, la de G mayor. La transposición, en este esquema, se reduce a una suma modular aplicada a todo el patrón.

La transposición es una operación fundamental en teoría musical que consiste en desplazar todos los elementos de una escala, acorde o melodía una misma cantidad de semitonos hacia arriba o hacia abajo [36]. En el sistema $\mathbb{Z}_{12}$, esta operación se simplifica al sumar un valor constante a cada elemento de la secuencia, aplicando la operación módulo 12. Por ejemplo, transponer cualquier escala $S = \{s_1, s_2, \ldots, s_n\}$ por un intervalo k se expresa como $S' = \{(s_1 + k) \text{ mód } 12, \ldots, (s_n + k) \text{ mód } 12\}$. Esta formalización permite implementar la transposición de forma eficiente y consistente, tanto en análisis teórico como en aplicaciones computacionales [60].

El punto principal de la teoría musical armónica es que los acordes de transición (como los subdominantes) y de resolución (como la tónica) generan significado a través de la manipulación de la tensión. Los acordes de transición nos alejan de la estabilidad, creando un movimiento que conduce a la tensión casi insoportable del acorde dominante, el cual, por su naturaleza disonante, exige regresar al reposo del acorde de tónica. Es en este ciclo de tensión y liberación, en cómo se construye, se prolonga o se resuelve esta expectativa, donde la música trasciende el sonido para convertirse en un lenguaje emocional, capaz de evocar narrativas complejas que van desde la certeza y la finalidad hasta el suspenso, la contemplación y el anhelo, pues esta es una forma de generar y resolver tendencias [7], [36].

3.3. Características acusticas de la música

Espectrogramas: Una señal de audio es la representación de las características acústicas del sonido. Este se entiende como un fenómeno de vibraciones que se propaga en el tiempo, y cuyas variaciones producen cambios en dicho sonido. Al capturar sus espectros, es posible generar una representación gráfica bidimensional que muestra la evolución de sus frecuencias a lo largo del tiempo. Estos son los espectrogramas. El espectrograma cuenta con dos dimensiones o ejes: el eje de las abscisas corresponde al tiempo y el eje de las ordenadas corresponde a la frecuencia [34], [58], [61].

Transformada de Fourier de Tiempo Corto (STFT): La *STFT* es una herramienta básica en el análisis tiempo-frecuencia que consiste en dividir la señal en pequeños segmentos de duración fija y aplicar la transformada de Fourier a cada segmento. Esto permite observar cambios en la frecuencia a lo largo del tiempo, proporcionando una representación visual denominada espectrograma. Según [58], esta técnica es fundamental para la comprensión didáctica y práctica del comportamiento espectral de señales musicales debido a su capacidad para relacionar claramente la variabilidad temporal y frecuencial de la señal.

De acuerdo con [34] la transformada de Fourier en corto plazo se define como la siguiente ecuación:

$$STFT = \{x(t)\} \equiv X(\tau, w) = \int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-iwt}dt \tag{1}$$

El proceso se entiende como:

1. Sea una señal $x(t)$ la función que se quiere transformar
2. Se multiplica esa señal por una ventana $w(t)$, que es diferente de cero solo en un intervalo corto (se suelen usar ventanas de Hann o Gaussianas).
3. A medida que la ventana se va deslizando a lo largo de la señal en el tiempo, se calcula la transformada de Fourier de la porción de la señal que queda tapada por la ventana.

En donde:

- $x(t)$ es la señal original
- $w(t-\tau)$ es la ventana centrada en τ
- w es la frecuencia angular
- e^{-iwt} corresponde al núcleo de la transformada de Fourier.

$w(t-\tau)$ es una función ventana que se traslada en el tiempo para analizar segmentos sucesivos de la señal. La elección de la ventana y su duración determinan la resolución tiempo-frecuencia del análisis [58].

En la librería `librosa` [62], los parámetros de la función `stft` controlan los elementos de esta ecuación de la siguiente manera:

1. `n_fft`: Este parámetro define la resolución en frecuencia. Está directamente relacionado con la variable de frecuencia ω en el núcleo de la transformada $e^{-i\omega t}$. Un `n_fft` mayor calcula la transformada para más puntos de frecuencia ω.
2. `hop_length`: Controla el desplazamiento de la ventana a lo largo del tiempo. Corresponde al paso discreto de la variable temporal τ. Define qué tan seguido se calcula una nueva transformada a lo largo de la señal $x(t)$.
3. `win_length`: Determina el tamaño de la función ventana $w(t-\tau)$. Define cuánta porción de la señal original $x(t)$ se analiza en cada paso τ.

Espectrograma Mel (Mel-Gram): Los espectrogramas Mel, o Mel-Grams, se basan en una escala mel que modela la percepción auditiva humana al enfatizar frecuencias que son perceptualmente relevantes. Los espectrogramas Mel han demostrado gran utilidad en tareas relacionadas con reconocimiento automático de características musicales, debido a su correlación con la manera en que los humanos perciben diferencias tonales y dinámicas en la música [63]. El trabajo de [34] enfatiza su aplicación en

la detección emocional multimodal por su capacidad de reflejar características psicoacústicas.

Técnicamente, la conversión de frecuencias lineales a frecuencias mel se realiza utilizando la ecuación:

$$mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2}$$

donde f es la frecuencia en hercios (Hz). Esto genera filtros espaciados según la percepción auditiva humana, permitiendo enfatizar rangos frecuenciales que son relevantes para el oído humano [63].

Transformada Q Constante (CQT): La Transformada Q Constante (CQT) proporciona una representación logarítmica del contenido frecuencial, donde la resolución frecuencial varía proporcionalmente a la frecuencia, generando una mejor adaptación a características musicales como las notas y sus armónicos. La CQT permite una identificación más precisa de las notas musicales y una interpretación más clara de las estructuras armónicas en comparación con métodos basados en STFT [61]. Esta característica hace que la CQT sea particularmente útil en aplicaciones musicales como identificación de notas, clasificación de instrumentos y seguimiento de modulaciones tonales, como demuestra [64].

De acuerdo con el trabajo de [61], la CQT se define como:

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} x[n] w[n] e^{-j2\pi Q n / N[k]} \tag{3}$$

donde Q es una constante que define la relación entre ancho de banda y frecuencia central, manteniéndose constante a través de todas las frecuencias analizadas, y $N[k]$ es el número variable de puntos para cada frecuencia central.

Chromagramas: Los chromagramas son espectrogramas especializados que muestran la distribución energética alrededor de las doce notas de la escala cromática, independientemente de la octava. Estos espectrogramas son altamente efectivos para capturar características emocionales en música debido a su capacidad de revelar patrones armónicos consistentes relacionados con emociones específicas [9]. La combinación de chromagramas con modelos de aprendizaje profundo ha resultado particularmente exitosa para aplicaciones como el reconocimiento automático de emociones musicales, gracias a su capacidad para destacar patrones melódicos y armónicos perceptualmente relevantes.

Desde el punto de vista técnico, los chromagramas se calculan a partir de espectrogramas convencionales mediante una agrupación energética en cada nota cromática. Matemáticamente, esto implica:

$$C(b, t) = \sum_{k \in \Omega_b} |X(k, t)|^2 \tag{4}$$

donde $C(b, t)$ es la energía en el bin cromático b en el tiempo t, y Ω_b representa el conjunto de frecuencias asociadas a la nota cromática específica b [9].

3.4. Redes Neuronales

En términos generales, una red neuronal es un modelo computacional inspirado en el cerebro humano que se compone de una colección interconectada de unidades llamadas neuronas. Cada neurona procesa la información que recibe, realiza una operación matemática en ella y produce una salida. Las neuronas se organizan en capas, donde cada capa se conecta con la siguiente mediante conexiones ponderadas. Estas conexiones y ponderaciones son ajustadas a través del entrenamiento para que la red pueda aprender y generalizar a partir de los datos de entrada [65].

La arquitectura de una red neuronal, como se ilustra en la Figura 2, se compone de tres tipos de capas. La primera, denominada **capa de entrada** (*input layer*), contiene las neuronas que reciben los datos iniciales. La última es la **capa de salida** (*output layer*), que entrega el resultado final. Entre estas dos se ubican una o más **capas ocultas** (*hidden layers*), las cuales procesan la información que fluye desde la entrada hacia la salida [65], [66].

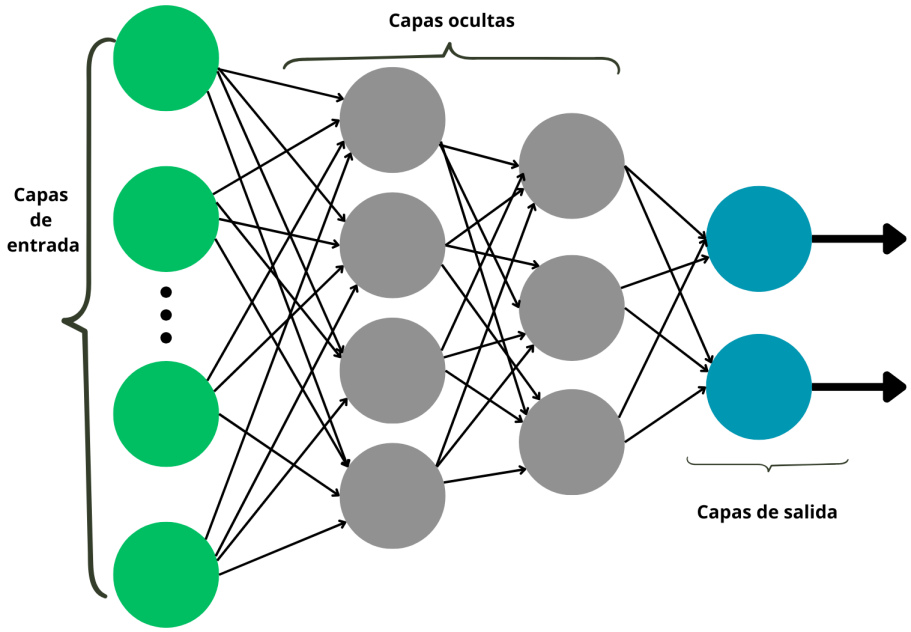

Figura 2: Arquitectura, simplificada, de una red neuronal; imagen adaptada de [65]

Un aspecto crucial en el diseño de modelos de redes neuronales es la correcta elección de sus funciones de pérdida, optimización y activación. La función de pérdida cuantifica el error del modelo durante el entrenamiento, mientras que el algoritmo de optimización es el mecanismo que actualiza los parámetros del modelo para minimizar dicho error [66]. Por su parte, la función de activación desempeña un rol central al introducir la no linealidad, una característica indispensable para que el modelo pueda aprender representaciones complejas de los datos [65].

La elección del optimizador es determinante para la eficacia y velocidad del entrenamiento de una red neuronal. Este componente se encarga de ajustar los pesos del modelo (parámetros) para minimizar la función de pérdida. A continuación, se detallan los algoritmos empleados en este trabajo.

El **Descenso de Gradiente Estocástico** o **SGD** (por sus siglas en inglés) es el algoritmo de optimización fundamental. En lugar de calcular el gradiente sobre todo el conjunto de datos, SGD lo hace para un único ejemplo o un pequeño lote (mini-batch), haciendo que el proceso sea mucho más rápido y computacionalmente eficiente. Este algoritmo de optimización suele ayudar al modelo a no caer en mínimos locales sub-óptimos [67].

Su regla de actualización es:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta J(\theta_t) \tag{5}$$

Donde:

- θ_{t+1} son los parámetros del modelo actualizados.
- θ_t son los parámetros en el paso actual.
- η (eta) es la **tasa de aprendizaje** (learning rate), que controla el tamaño del paso de actualización.
- $\nabla_\theta J(\theta_t)$ es el **gradiente** de la función de pérdida J con respecto a los parámetros θ.

RMSprop (Root Mean Square Propagation) es un optimizador adaptable que ajusta la tasa de aprendizaje de forma individual para cada parámetro. Lo logra dividiendo la tasa de aprendizaje por un promedio móvil de las magnitudes recientes de los gradientes. Esto permite amortiguar las oscilaciones en direcciones con gradientes muy grandes y acelerar el aprendizaje en direcciones donde el gradiente es pequeño, resultando en una convergencia más rápida y estable [67].

Su actualización se realiza en dos pasos:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \tag{6}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{7}$$

Donde:

- g_t es el gradiente en el paso actual t.
- $E[g^2]_t$ es el promedio móvil de los gradientes al cuadrado.
- γ (gamma) es el **factor de decaimiento** (decay rate), que controla la importancia de los gradientes pasados.

- ϵ (épsilon) es una constante de suavizado muy pequeña para evitar la división por cero.

Adam (Adaptive Moment Estimation) es otro optimizador adaptativo que combina las ventajas de dos métodos: RMSprop y Momentum. Almacena un promedio móvil no solo de los gradientes al cuadrado (segundo momento, como RMSprop), sino también de los propios gradientes (primer momento, como Momentum). Adam es conocido por su robustez y buen rendimiento en una amplia variedad de problemas, a menudo requiriendo poca configuración de hiperparámetros [68].

Las ecuaciones que rigen su actualización son:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{8}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{9}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{10}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{11}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{12}$$

Donde:

- m_t y v_t son las estimaciones del primer y segundo momento, respectivamente.
- β_1 y β_2 son los factores de decaimiento para ambos momentos.
- $\hat{m}_t$ y $\hat{v}_t$ son las estimaciones de los momentos corregidas para evitar el sesgo inicial hacia cero.
- t es el número del paso de iteración actual.

La función **Tangente Hiperbólica (Tanh)** es una de las activaciones clásicas. Comprime cualquier valor de entrada a un rango entre $[-1 - 1]$ [65].

Su ecuación es:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{13}$$

Donde:

- x es el valor de entrada a la neurona.

La función **Unidad Lineal Rectificada (ReLU)** es la activación más utilizada en las redes neuronales modernas por su simplicidad y eficiencia computacional [69]. Simplemente, devuelve el propio valor de entrada si este es positivo y cero en caso contrario. Esto ayuda a mitigar el problema del desvanecimiento del gradiente [35], [65].

Su ecuación es:

$$f(x) = \text{máx}(0, x) \tag{14}$$

Donde:

- x es el valor de entrada a la neurona.

Leaky ReLU es una variante de ReLU diseñada para solucionar el problema de la neurona muerta, que ocurre cuando una neurona se atasca en la región negativa y deja de aprender. De este modo, a diferencia de ReLU, se le asigna a αx un valor pequeño cercano a cero en lugar de usar directamente 0 [66]. Su ecuación es:

$$f(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha x & \text{si } x \leq 0 \end{cases} \tag{15}$$

Donde:

- x es el valor de entrada a la neurona.
- α (alfa) es una pequeña constante positiva, usualmente un valor como 0.01.

Unidad Lineal de Error Gaussiano (GELU) es una función de activación más moderna y suave, popular en arquitecturas avanzadas como los Transformers. Modula la salida de una neurona de forma probabilística, basándose en la función de distribución acumulada de la distribución normal estándar. Intuitivamente, decide si mantener o anular una salida de forma más suave que ReLU [70].

Su ecuación es:

$$f(x) = x \cdot \Phi(x) \tag{16}$$

Donde:

- x es el valor de entrada a la neurona.
- $\Phi(x)$ es la Función de Distribución Acumulada de la distribución gaussiana estándar.

Existen múltiples funciones de pérdida y esta se debe adaptar a la naturaleza del problema, pues su uso principal es el de reducir el error del modelo. En problemas de regresión es común encontrar funciones como *MSE* o *MAE*. No obstante, existe una opción más robusta que combina tanto el *MSE* como el *MAE*, la función de pérdida *Huber loss* o *pérdida de Huber*, que reduce la sensibilidad a valores atípicos. Por lo general, es usada para mejorar la estabilidad del modelo [49].

$$L_\delta(x) = \begin{cases} 0{,}5 \cdot x^2 & : \ if|x| \geq \delta \\ \delta \cdot |x| - 0{,}5 \cdot \delta^2 & : \ otherwise \end{cases} \tag{17}$$

En Donde:

- x: Es la diferencia entre los valores reales y los predichos.
- δ **(delta)**: Es un hiperparámetro que define el umbral. Los errores por debajo de δ son tratados como cuadráticos, mientras que los errores más grandes son tratados de forma lineal.

3.5. Redes Nueronales Convolucionales CNN

Son un tipo especial de red neuronal, estas son comúnmente utilizadas en el procesamiento de información que tiene una estructura en forma de cuadrícula. [12] [66]. De acuerdo con [65] las principales características de las CNN son: *Recepcción de campos locales*, *Pesos compartidos* y *Agrupación*.

La recepción de campos locales es un proceso que ocurre por medio de convoluciones, una convolución es un operador que permitirá extraer información de los datos ingresados en la neurona [66]. La ecuación 18 es la operación de convolución que implementaron en [35]

$$F(i,j) = (R * w)(i,j) = \sum_x \sum_y R(i-x, j-y)w(x,y) \tag{18}$$

Además de la operación de convolución, otro paso dentro la CNN es la función de agrupamiento o pooling, una técnica par agrupación es el Max Pooling. De acuerdo con [66] en el proceso de max pooling, a partir de la información entrante y saliente se extraen regiones o ventanas. Y de estas regiones se conservan solo los valores máximos. La ecuación 19 es la función para la operación de pooling usada en [35].

$$MaxPooling(x,y) = max(x,y) \tag{19}$$

3.6. Memoria a largo y corto plazo LSTM

Las redes neuronales LSTM son un tipo de redes neuronales recurrentes, en ocasiones son denotadas como LSTM-RNN, dentro de las RNN las LSTM son de las más poderosas y por ende también son de las que más recursos consumen [69].

Este tipo de RNN ofrece solución a uno de los problemas que aquejan a las RNN, el cual es el problema del gradiente inestable, en líneas generales este problema ocasiones que el aprendizaje en las primeras capas sea en extremo lento [71].

Los bloques de LSTM poseen una memoria a largo plazo, la cual se le denomina como estado de la célula (cell state). A su vez, los bloques se compone de tres puertas:

- Input Gate: Se encarga de generar los valores que se necesitan para deducir los nuevos estados.
- Forget Gate: Se encarga de controlar la información que ha sido descartada en estados previos.
- Output Gate: Se encarga de generar los valores que determinaran los siguientes estados [72].

Este tipo de arquitecturas son unidireccionales, es por esta razón que para problemas en donde es importante el contexto en ambas direcciones se emplea la arquitectura bidireccional. De esta forma existen arquitecturas que proponen unir dos bloques LSTM, así, se puede enfocar en procesar la secuencia hacia adelante (forward) y la otra procesa hacia atrás (backward) [22], [71].

3.7. Redes Residuales ResNet

Las Redes Neuronales Residuales (ResNet) son una arquitectura de redes neuronales profundas introducida para resolver el problema de la **degradación del rendimiento**, este define que, contrario a lo esperado, añadir más capas a un modelo de red neuronal puede llevar a un error de entrenamiento más alto, esto se debe a que al ser un modelo con más capas la optimización de la red se convierte en una tarea más compleja [73].

La idea fundamental de ResNet es introducir **conexiones de salto** (skip connections) que permiten que la información de una capa anterior se sume a la de una capa posterior, saltándose una o más capas intermedias.

El componente clave de una ResNet es el **bloque residual**. En lugar de esperar que un conjunto de capas apiladas aprenda directamente una función de mapeo subyacente $H(x)$, se les obliga a aprender una **función residual** $\mathcal{F}(x)$ [73].

La salida del bloque, y, se define matemáticamente como:

$$y = \mathcal{F}(x, \{W_i\}) + x$$

Donde:

- x es el vector de características de entrada al bloque.
- $\mathcal{F}(x, \{W_i\})$ es el mapeo residual aprendido por las capas del bloque, con pesos W_i.
- La operación $+x$ es la conexión de salto (skip connection) que suma la entrada original (identidad) a la salida de las capas.

3.8. Bloques Squeeze-and-Excitation (SE)

Durante el boom de las redes CNN, muchas arquitecturas de referencia, como la propia ResNet, VGG o Inception, se centraban principalmente en capturar características espaciales: bordes, formas, texturas, etc., tratando a los canales o mapas de activación por igual, sin ponderar la importancia que cada uno tuviera.

Por ende, en es que el trabajo de [74] propuso una especie de mecanismo de atención primigenio enfocados en la importancia de los mapas de cara. Su objetivo es modelar explícitamente las interdependencias entre las características de los canales. Para lograrlo, realiza una **recalibración adaptativa de características por canal**, permitiendo que la red aprenda a enfatizar las características informativas y suprimir las menos útiles, este proceso se divide en dos operaciones *Squeeze* y *Excitation*.

La operación **Squeeze** condensa la información espacial del mapa de características de entrada utilizando una agregación global por canal. A través de una operación de *Global Average Pooling* (GAP), se genera un vector descriptor [74], donde cada componente se calcula como:

$$z_c = \mathbf{F}_{GP}(textbf u_c)$$

En done $\mathbf{u}_c$ es el canal número c del vector de entrada $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_c]$ el cual fue generado por una operación de convolución, $\mathbf{F}_{GP}$ es la operación de pooling. Esto genera el vector descriptor z_c [50].

El segundo paso, *Excitation*, tiene como objetivo capturar completamente las dependencias de los canales a partir de la información agregada. Para ello, utiliza un mecanismo de compuerta (gating mechanism) con dos capas completamente conectadas (FC) alrededor de una no linealidad [50], [74]:

$$\mathbf{s} = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{z}))$$

Donde:

- δ es la función de activación ReLU.
- σ es la función de activación sigmoide.
- $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ y $\mathbf{W}_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ son los pesos de las dos capas FC. Estas capas forman un cuello de botella (bottleneck) con un ratio de reducción r para limitar la complejidad del modelo y ayudar a la generalización.

Finalmente, el mapa de características es reescalado canal a canal utilizando los valores de $\mathbf{s}$:

$$\tilde{\mathbf{x}}_c = s_c \cdot \mathbf{u}_c$$

De esta manera, el bloque SE permite que la red enfatice dinámicamente los canales más relevantes.

3.9. Embeddings y Modelos Word2Vec

Los *embeddings* son representaciones vectoriales numéricas de datos en un espacio de menor dimensión, donde los elementos similares en el contexto de los datos originales quedan cerca entre sí. En el ámbito de procesamiento del *lenguaje natural* o *NLP* por sus siglas en inglés (Natural Language Process), los embeddings son especialmente útiles para representar palabras en un espacio vectorial, facilitando que los modelos interpreten relaciones y similitudes semánticas entre palabras [37], [38].

Word2Vec es un método ampliamente utilizado para generar embeddings de palabras, desarrollado por investigadores de *Google* [75]. Este modelo emplea redes neuronales poco profundas para aprender representaciones distribuidas de palabras a partir de su contexto, logrando que términos semánticamente similares estén representados por vectores cercanos en el espacio. *Word2Vec* presenta dos arquitecturas principales: *CBOW (Continuous Bag of Words)* y *Skip-gram*.

CBOW: Este modelo predice una palabra objetivo a partir de su contexto circundante, es decir, utiliza las palabras circundantes para predecir una palabra central.

CBOW resulta útil cuando se requiere capturar una representación basada en el contexto global, pues este modelo se entrena para minimizar la probabilidad de error al predecir una palabra a partir de el conjunto de palabras que la rodean [38].

Skip-gram: A diferencia de CBOW, el modelo *Skip-gram* realiza la operación inversa: dada una palabra central, intenta predecir las palabras que la rodean en un contexto definido [37]. Este enfoque es especialmente útil para capturar relaciones y similitudes semánticas a nivel individual, ya que el modelo se entrena para maximizar la probabilidad de las palabras del contexto condicionado a una sola palabra objetivo.

$$\mathcal{L}_{\text{Skip-gram}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-c \leq j \leq c \\ j \neq 0}} \log p\big(w_{t+j} \mid w_t\big),$$

$$p\big(w_o \mid w_i\big) = \frac{\exp\big({v'_{w_o}}^{\top} v_{w_i}\big)}{\sum_{w=1}^{W} \exp\big({v'_{w}}^{\top} v_{w_i}\big)}$$

En donde:

- T Número total de palabras en el corpus de entrenamiento.
- c Tamaño de la ventana de contexto (número de palabras a cada lado de la central).
- t Índice de la palabra central en la secuencia, $t = 1, 2, \ldots, T$.
- j Desplazamiento dentro de la ventana de contexto, $-c \leq j \leq c$ y $j \neq 0$.
- w_t Palabra central en la posición t.
- w_{t+j} Palabra de contexto desplazada j posiciones respecto a la central.
- $\mathcal{L}_{\text{Skip-gram}}$ Función objetivo (average log-probability) del modelo Skip-gram.
- $v_w \in \mathbb{R}^d$ Vector de entrada ("input") de dimensión d asociado a la palabra w.
- $v'_w \in \mathbb{R}^d$ Vector de salida ("output") de dimensión d asociado a la palabra w.
- W Tamaño total del vocabulario.M.E.R.
- $p(w_o \mid w_i)$ Probabilidad de predecir la palabra de salida w_o dado el vector de entrada de la palabra central w_i, definida por la softmax.

3.10. Métricas

Dado que el problema abordado en este trabajo es de naturaleza regresiva, es fundamental emplear métricas que cuantifiquen con precisión el error entre los valores predichos por el modelo y los valores reales. Para ello, se utilizan cuatro métricas ampliamente reconocidas en tareas de regresión: el *Error Cuadrático Medio* (MSE), la *Raíz del Error Cuadrático Medio* (RMSE), el *Error Absoluto Medio* (MAE) y el coeficiente de determinación R^2. Estas métricas permiten evaluar distintos aspectos del desempeño del modelo, tales como la magnitud promedio del error, su sensibilidad a errores grandes y la proporción de la varianza explicada por el modelo.

Raíz del error cuadrático medio (RMSE): Mide la precisión de un modelo de regresión. Se calcula como la raíz cuadrada de la media de los errores cuadrados entre las predicciones del modelo y los valores reales. En la ecuación 20, n corresponde al número total de muestras, y_j corresponde al valor real de la variable dependiente de la muestra y $\hat{y_j}$ es la predicción [21].

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2} \tag{20}$$

Raíz de error relativo (RSE): Esta ecuación 21 se usa para calcular R^2. Es el residuo de la suma de los cuadrados, donde $\bar{y_j}$ representa la media del valor de y [21].

$$RSE = \frac{\sum_{j=1}^{n}(y_j - \hat{y_j})^2}{\sum_{j=1}^{n}(y_j - \bar{y_j})^2} \tag{21}$$

Puntuación R2 (score): Es usualmente usada para evaluar la exactitud de modelos de regresión. Calcula qué tan lejos se encuentran los valores de los datos de la línea de regresión [21].

$$R^2 = 1 - RSE \tag{22}$$

Aunque en ocasiones también se llega a usar el error absoluto medio. **Error absoluto medio (MAE):** Este calcula el error de la predicción del modelo. Con la ecuación 23 se puede calcular el MAE. Dicha ecuación está basada en la implementada por Yang en [21].

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y_j}| \tag{23}$$

4. Hipótesis

Mediante un sistema automático basado en machine learning, que analiza y discrimina tanto características de audio como el contexto armónico de obras musicales es capaz de encontrar patrones y relaciones de una manera natural y parecida a como un músico efectuaría el reconocimiento de emociones en obras musicales.

5. Objetivos

1. Objetivo general:

   Analizar características de bajo y alto nivel de obras musicales, por medio de un modelo múltiple de I.A basado en técnicas de deep learning, para efectuar el reconocimiento de emociones en obras musicales.

2. Objetivos específicos:

   - Preparar los datos, con técnicas de preprocesamiento, para alimentar los modelos de aprendizaje.
   - Analizar las características extraídas, por medio de modelos de aprendizaje, para realizar un primer reconocimiento de emociones.
   - Realizar la fusión de ambos modelos de I.A para obtener la clasificación final de emociones.

6. Métodos y Materiales

6.1. Introdución a la metodología

La metodología a seguir para el desarrollo del proyecto se encuentra planteada en la figura 3, la cual está compuesta por cuatro fases: la primera corresponde al proceso de recolección de los datos. La segunda, al proceso de extracción de información o características. La tercera son los modelos de análisis y aprendizaje, finalmente, la cuarta es la fusión de dichos modelos.

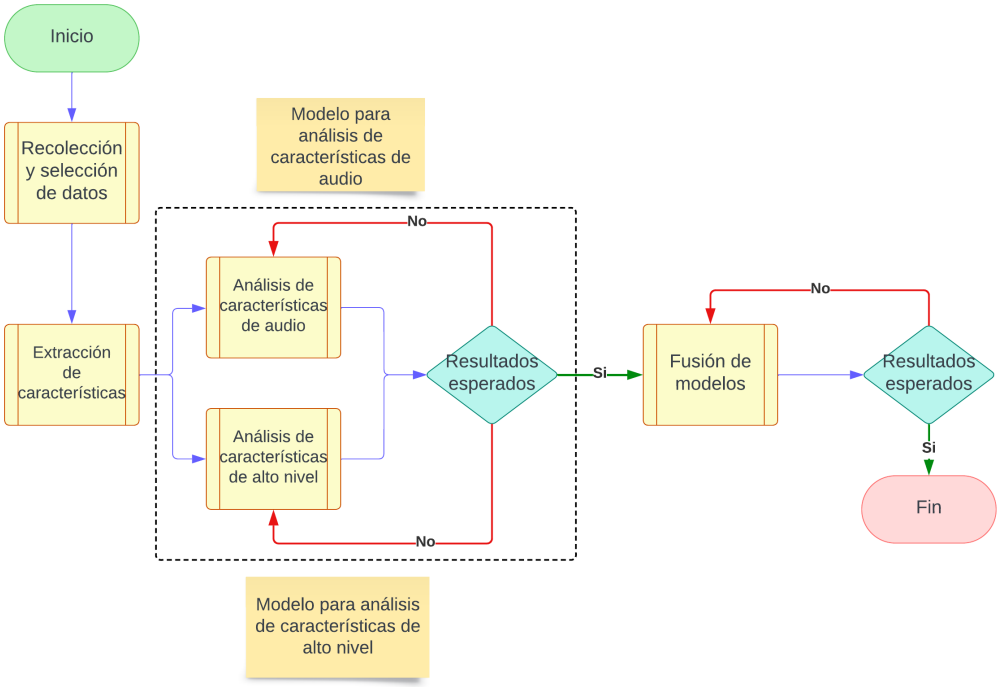

Figura 3: Metodología para el proyecto

6.2. Materiales

6.2.1. Conjunto de datos

Los conjuntos de datos existentes para la tarea de MER son variados y reflejan el panorama actual de esta tarea, pues presentan diferencias tanto ne las características como la taxonomía (dimensional o categórica), el nivel en el que las canciones son etiquetadas (estática o dinámica) o la metodología con la que las anotaciones se obtienen.

Es esta falta de homogeneidad en los conjuntos de datos resulta valioso encontrar aquellos con similitudes que permitan utilizarlos de manera conjunta. Por ello, para este trabajo se seleccionaron dos de los conjuntos de datos más utilizados en MER *PMEmo* [76] y *DEAM* [41]. Estos conjuntos siguen una metodología parecida en la generación de etiquetas, pues ambos se basan en una taxonomía dimensional. Cada anotación cuenta con valores en los ejes de *valence* y *arousal*. Por su parte, ambos conjuntos utilizaron encuestas SAM y múltiples anotadores para generar cada anotación estática global para cada una de las canciones. Tanto *PMEmo* como el conjunto de datos de *DEAM* son públicos y cuentan con archivos de audio. La tabla 3 muestra algunas de las características esenciales de los conjuntos de datos seleccionados.

Cuadro 3: Principales características de los conjuntos de datos *PMEmo* y *DEAM*

Base de datos	Año	Contenido	Formato	Tipo	Rango
PMEmo	2019	794 extractos	MP3	Dimensional (VA)	$(0-1)$
DEAM	2017	1802 extractos	MP3	Dimensional (VA)	$(1-9)$

***PMEmo*:** Contiene 794 anotaciones, recolectadas a partir de un experimento realizado a 457 sujetos. Para la generación de estas anotaciones se utilizó la escala Self-Assessment Manikin (SAM) con nueve valores, los cuales fueron posteriormente normalizados al rango $[0, 1]$. El conjunto de datos ofrece diferentes tipos de anotaciones. Para la presente investigación se seleccionaron las anotaciones emocionales dimensionales Valence-Arousal, obteniendo un total de 767 archivos de audio en formato .mp3 con sus respectivas anotaciones estáticas. Los extractos de audio tienen una frecuencia de 44.1 kHz y la duración de los mismos es variable.

DEAM: El conjunto de datos de MediaEval cuenta con un total de 1802 archivos de audio en formato .mp3 con sus respectivas anotaciones. Estos datos han sido recopilados durante un periodo de 3 años, de 2013 a 2015. Cuenta con música libre de derechos. Al igual que *PMEmo*, las anotaciones fueron obtenidas en escala SAM de nueve puntos, de $[1, 9]$. *DEAM* cuenta con anotaciones estáticas y dinámicas, ambas dimensionales, de las cuales solo se seleccionaron las estáticas. Al igual que *PMEmo*, los extractos de audio tienen una frecuencia de 44.1 kHz. La duración de los audios es de 45 segundos, no obstante, los datos recopilados en 2015 no cumplen con esta característica y la mayoría de audios contienen la canción completa.

6.2.2. Gestor de base de datos relacional

Para reunir la información de los metadatos y anotaciones de ambos conjuntos de datos en un solo lugar, se diseñó una pequeña base de datos relacional. Como gestor se utilizó SQLite, debido a su sencillez y portabilidad, pues toda la información se concentra en un único archivo, además de no requerir un servidor dedicado.

El uso de una base de datos relacional, en lugar de un archivo CSV que unifique la información de los datasets, se debe a que la estructura física de almacenamiento se encuentra separada de la parte lógica, lo cual permite modificar la estructura física sin afectar los programas que acceden a los datos.

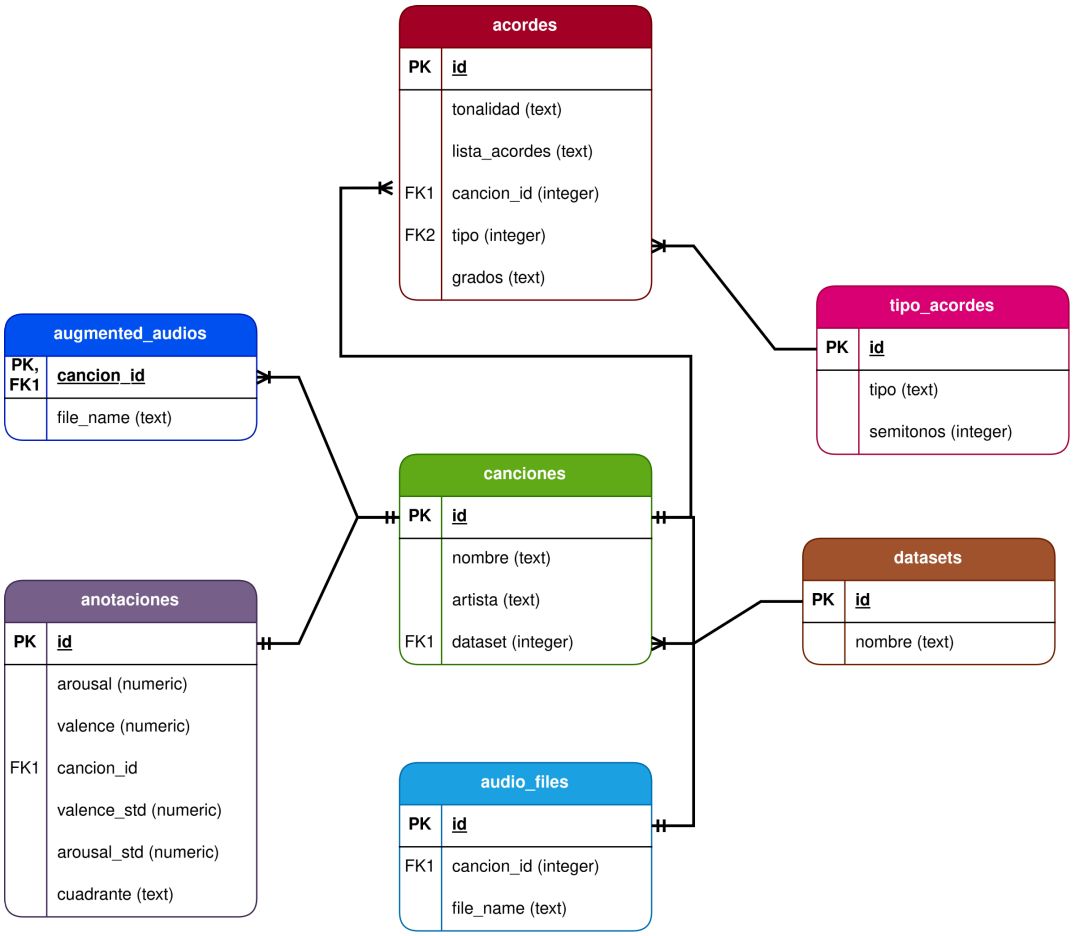

Figura 4: Diagrama relacional de la base de datos final.

La información de *PMEmo* y *DEAM* se unificó en una sola base de datos. De acuerdo con la figura 4, la tabla central, núcleo de la unificación, `canciones`, contiene la información básica de cada canción, como el nombre y el artista. La tabla de origen, `datasets`, indica de qué fuente original proviene cada canción. Por último, las tablas `audio_files`, `anotaciones`, `acordes`, `augmented_audios` y `tipo_acordes` contienen toda la información específica y técnica. La información se conecta con cada canción para detallar los nombres de los archivos de audio, las anotaciones emocionales y los acordes extraídos.

6.2.3. Entorno de Python y librerías utilizadas

Con el fin de preservar la modularidad y separar el proyecto por fases, cada tarea fue llevada a cabo en entornos virtuales de Python. Cada entorno fue configurado y creado mediante la plataforma de Anaconda. Los entornos virtuales se encuentran descritos en la tabla 4.

Cuadro 4: Entornos viruatles de python utilizados

Entorno Virtual	**Versión de Python**	**Uso**
chord_extraction_38	3.8.20	Para tareas de extracción de acordes a partir de audio, utilizando librerías compatibles con Python 3.8.
mer_prepdata	3.9.21	Para la preparación y preprocesamiento de datos destinados a modelos de Reconocimiento de Emociones en la Música (MER).
nlp-audio-env	3.10.17	Para la construcción y experimentación con los diferentes modelos tanto DL como NLP para el análisis de características acústicas y simbólicas

Para el desarrollo de este proyecto, se empleó un conjunto de librerías clave de Python. Las tareas de aprendizaje profundo y optimización de hiperparámetros se realizaron con *PyTorch* y *Optuna*, respectivamente.

El procesamiento de señales de audio y la extracción de características musicales fueron manejados principalmente por *Madmom* [77] y *Librosa* [62]. Por su parte, la manipulación de datos, el análisis numérico y la implementación de modelos de machine learning tradicionales se apoyaron en las fundamentales *Pandas, Numpy y Scikit-learn.*

6.2.4. Hardware utilizado

El desarrollo del proyecto y la experimentación se llevaron a cabo en un equipo de escritorio con el sistema operativo *Manjaro Linux*. El sistema está impulsado por un procesador *AMD Ryzen 5 5600G* con *gráficos integrados Radeon Vega* y cuenta con *64 GB de memoria RAM*, lo que facilitó el manejo eficiente de grandes volúmenes de datos y el entrenamiento de los modelos.

6.3. Tratameinto de los datos

6.3.1. Metadatos y anotaciones

Ambos conjuntos de datos comparten características importantes que posibilitan su fusión. No obstante, presentan ciertas diferencias en cuanto al formato, la organización de los datos y las anotaciones. Por ello, como paso inicial, se realizó una limpieza de datos nulos y, posteriormente, se homogeneizó la estructura y el formato de estos para

permitir el uso de ambos conjuntos. El diagrama de la figura 5 es el proceso simplificado que se implementó para la limpieza y preparación de los datos.

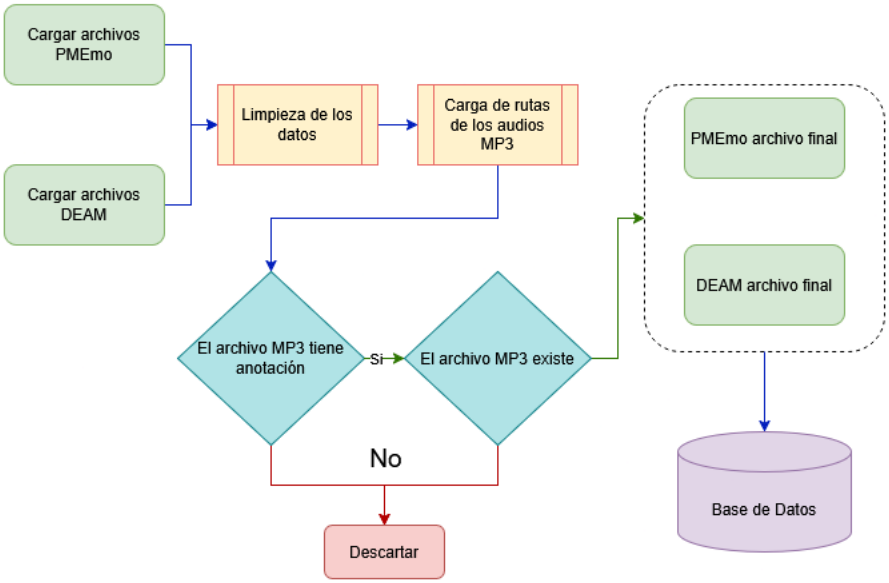

Figura 5: Proceso para el tratamiento de las anotaciones y metadatos de los conjuntos de datos

En el caso de *PMEmo*, este cuenta con un total de 4 archivos CSV. No obstante, para este trabajo, solo se tomaron en cuenta las anotaciones estáticas a nivel de canción (Song-Level). La estructura del CSV de anotaciones utilizado cuenta con 3 columnas: la primera es un id, la segunda es el valor del *arousal* y la tercera, el valor del *valence*. A partir de estas anotaciones, se obtuvieron las entradas únicas por id, lo que permitió eliminar duplicados de forma segura.

Posteriormente, se cargaron las rutas de los archivos de audio en una lista y también se cargó el archivo de metadatos. De este último solo se tomaron en cuenta las columnas del id, el nombre del artista, el nombre de la canción y el nombre del archivo de audio. De modo similar al archivo de anotaciones, se conservaron las entradas únicas por id en los metadatos. Además, se filtraron los datos para mantener solo aquellas entradas cuyo id también se encontrara en el archivo de anotaciones. De esta forma, se selecciona la información de audios que cuentan con anotaciones.

Por último, del archivo de metadatos, se comprueba que en cada entrada la ruta del archivo de audio almacenada exista en la lista de rutas para conservar solo la información de los audios que cuentan tanto con anotaciones como con un archivo de audio MP3.

Todos los archivos de audio del conjunto de datos de *PMEmo* se encuentran almacenados en la misma carpeta raíz. Esta ruta se define como una variable de entorno en un archivo `.env`. Así, al conocer el nombre del archivo de audio, es posible obtener la

ruta completa y acceder al archivo MP3 correspondiente.

En el caso del conjunto *DEAM*, como parte de la limpieza de datos, los audios del 2015 tienen duración completa, por lo tanto estos audios se separaron del resto y se segmentaron en clips de 45 segundos, los fragmentos de 45 segundos de cada canción se seleccionaron de forma aleatoria respetando la metodología del propio *DEAM*. Los segmentos de 45s fueron guardados con el resto de clips.

DEAM divide sus anotaciones estáticas a nivel de canción en dos archivos, ambos poseen las mismas columnas. El primer archivo tiene la información de la versión pre-2015. En cada archivo de anotaciones se conserva un id único. Ambos archivos se fusionaron para crear un nuevo archivo con todas las anotaciones. El archivo final conserva solo 3 columnas de los archivos de anotaciones originales: la columna del id y las columnas con los valores de *valence* y *arousal*.

Debido a que las anotaciones en el dataset de *DEAM* se encuentran en la escala de $[1-9]$, los valores se normalizan a un rango de $[0-1]$ aplicando la ecuación 24:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{24}$$

6.3.2. Control de las rutas de los archivos de audio

Para no almacenar la ruta completa del archivo de audio en la base de datos, se guardó una cadena del tipo: `$PMEMO:{nombre_archivo}` De este modo, la ruta real se construye dinámicamente sustituyendo el prefijo por la variable de entorno. Los algoritmos 1 y 2 muestran el pseudocódigo para codificar y decodificar las rutas de los archivos de audio.

Algorithm 1 EncodeRuta: Codifica el nombre de archivo en la cadena con prefijo

Require: *nombre_archivo* (cadena, por ejemplo "track123.mp3")
Ensure: *ruta_codificada* (cadena en formato "$PMEMO:track123.mp3")
1: $prefijo \leftarrow$ `$PMEMO:`
2: $ruta\_codificada \leftarrow prefijo \,\|\, nombre\_archivo$
3: **return** *ruta_codificada*

6.3.3. Fusión de los conjuntos de datos

Para este trabajo, las emociones se representan en un espacio bidimensional donde el eje horizontal corresponde al *valence* y el eje vertical al *arousal*, ambos normalizados en el rango $[0, 1]$. Para facilitar el análisis y la visualización, el espacio dimensional fue dividido en cuatro cuardantes [18], [19]:

- **Cuadrante 1 (Q1):** *arousal* $> 0{,}5$ y *valence* $> 0{,}5$. Corresponde a emociones de alta activación y valencia positiva.

Algorithm 2 DecodeRuta: Convierte la cadena codificada en ruta absoluta

Require: $ruta\_codificada$ (cadena, p. ej. "$PMEMO:track123.mp3")
Ensure: $ruta\_absoluta$ (cadena con la ruta de disco)

$prefijo \leftarrow$ `$PMEMO:`
if `startsWith`$(ruta\_codificada, prefijo)$ **then**
  $nombre\_archivo \leftarrow \text{substring}(ruta\_codificada, |prefijo| + 1, \text{fin})$
  $ruta\_base \leftarrow \text{LeerVariableEntorno}(\texttt{PMEMO\_ROOT})$
  $ruta\_absoluta \leftarrow ruta\_base \,\|\, \text{"/"} \,\|\, nombre\_archivo$
  return $ruta\_absoluta$
else
  return $ruta\_codificada$ {Ya es ruta absoluta o no usa prefijo}
end if

- **Cuadrante 2 (Q2):** *arousal* $> 0{,}5$ y *valence* $\leq 0{,}5$. Representa emociones de alta activación pero valencia negativa.

- **Cuadrante 3 (Q3):** *arousal* $\leq 0{,}5$ y *valence* $\leq 0{,}5$. Agrupa emociones de baja activación y valencia negativa.

- **Cuadrante 4 (Q4):** *arousal* $\leq 0{,}5$ y *valence* $> 0{,}5$. Indica emociones de baja activación pero valencia positiva.

En la práctica, se toma cada par (*valence*, *arousal*) y se le aplica la siguiente regla:

$$\text{cuadrante} = \begin{cases} \text{Q1}, & \text{si arousal} > 0{,}5 \ \wedge \ \text{valence} > 0{,}5, \\ \text{Q2}, & \text{si arousal} > 0{,}5 \ \wedge \ \text{valence} \leq 0{,}5, \\ \text{Q3}, & \text{si arousal} \leq 0{,}5 \ \wedge \ \text{valence} \leq 0{,}5, \\ \text{Q4}, & \text{si arousal} \leq 0{,}5 \ \wedge \ \text{valence} > 0{,}5. \end{cases}$$

De este modo, cada punto en la gráfica de dispersión se clasifica en uno de los cuatro cuadrantes, lo que permite visualizar fácilmente en qué regiones del espacio emocional se concentran las anotaciones de cada dataset antes de la fusión.

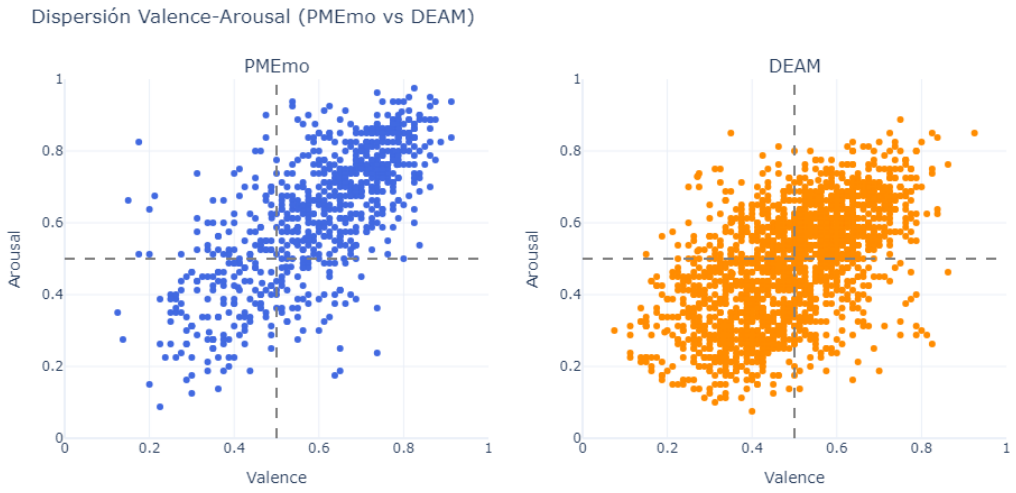

Figura 6: Comparativa de la distribución de los valores en los conjuntos de datos PMEmo(azul) Y DEAM(naranja)

Antes de efectuar la fusión de los conjuntos de datos, la dispersión en ambos varía ligeramente. En *PMEmo* se observa una mayor concentración de anotaciones en el `Q1`, como se aprecia en la figura 6. En *DEAM*, existe una mayor concentración de datos en el centro (*valence* ≈ 0.5, *arousal* ≈ 0.5), aunque cabe recalcar que en la zona del cuadrante `Q1` como en la zona de `Q3` es donde se nota una mayor densidad. No obstante, la varianza de ambos datasets es comparativamente similar (ver la tabla 6.3.3), y en ambos casos las anotaciones se concentran en torno a valores medios de cada eje, tal como muestra la gráfica de cajas 7. *PMEmo* muestra una ligera tendencia hacia valores más altos de *valence* y *arousal*. De hecho, en PMEmo aparecen un par de outliers en valores muy bajos de *arousal* y *valence*. Por su parte, *DEAM* tiende aún más al centro sin gran dispersión hacia los extremos y solo con un outlier en el eje de *valence*, próximo a 1.

Cuadro 5: Media y varianza de Valence y Arousal para PMEmo y DEAM

DS	**VM**	**VV**	**AM**	**AV**
PMEmo	0.596581	0.026239	0.622355	0.034156
DEAM	0.488018	0.021544	0.476754	0.025688

DS: Dataset; **VM**: Valence Mean; **VV**: Valence Variance; **AM**: Arousal Mean; **AV**: Arousal Variance.

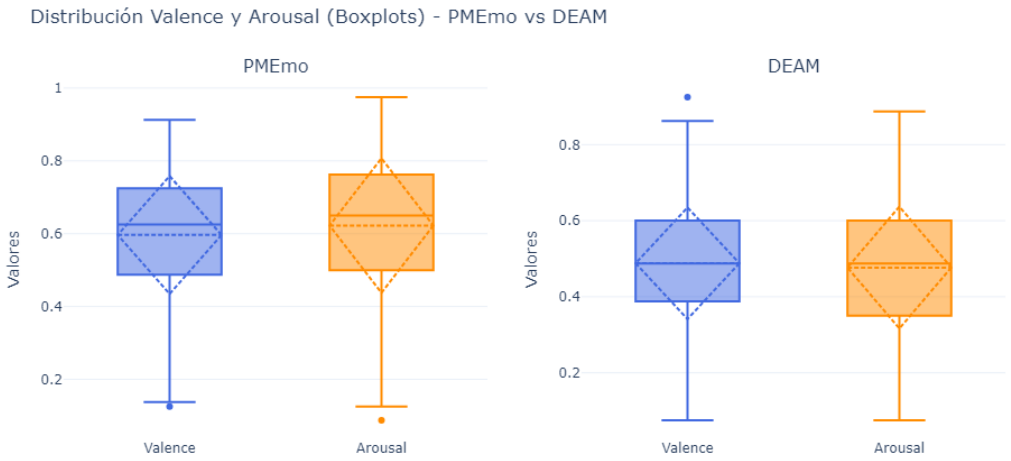

Figura 7: Comparativa distribución de los valores Valence y arousal en los conjuntos PMEmo(azul) Y DEAM(naranja)

Tras la fusión de *PMEmo* y *DEAM*, la dispersión conjunta agrupada por cuadrantes (ver figura 8) muestra que la mayoría de las canciones se agrupan alrededor del punto medio (*valence* ≈ 0.5, *arousal* ≈ 0.5). En concreto, se mantiene cierta preferencia por valores de *valence* moderadamente altos y *arousal* medios, aunque aparecen ejemplos distribuidos en todos los cuadrantes (`Q1-Q4`), lo que refleja la combinación de ambas fuentes originales.

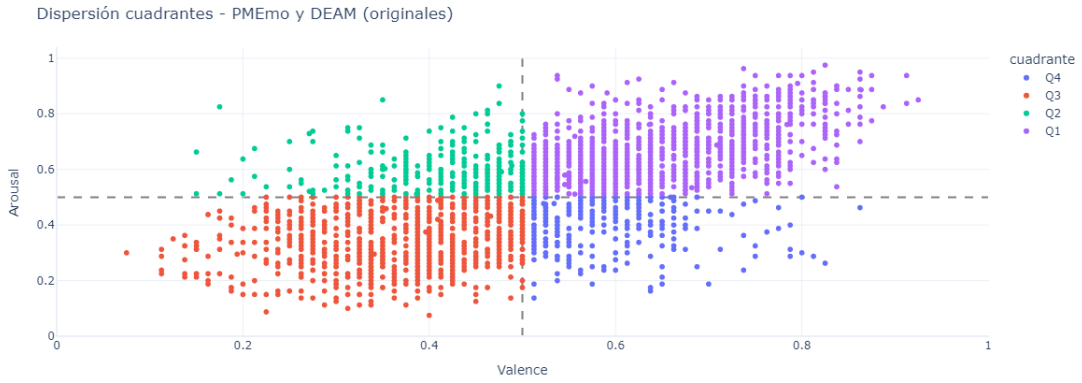

Figura 8: Dispersión de los anotaciones Valence, Arousal tras realizar la fusión de los datos.

En el boxplot resultante para el dataset fusionado (figura 9), la mediana de *valence* se ubica cerca de 0,52, mientras que la de *arousal* ronda 0,50, confirmando que las anotaciones más frecuentes se encuentran en la región central. El rango intercuartíli-

co de *valence* se extiende aproximadamente entre 0,40 y 0,65, y el de *arousal* entre 0,38 y 0,66, lo que indica una variabilidad moderada. Los bigotes sugieren que no hay valores extremos demasiado alejados de 0 o 1. En conjunto, estos gráficos evidencian que la fusión logra eliminar los datos con comportamiento lejano a la media, además de mantener la herencia de la dispersión original de *PMEmo* y *DEAM*. Sin embargo, también es notable cómo los datos tienden a concentrarse en la zona media del espacio emocional.

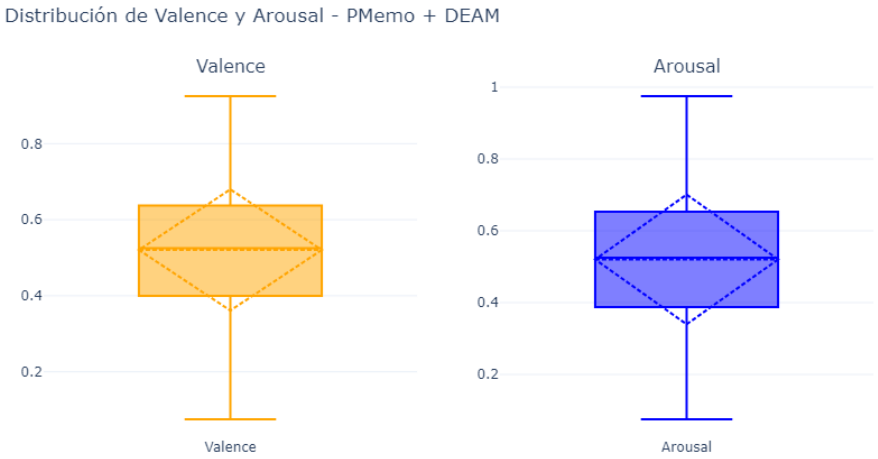

Figura 9: Comparativa distribución de los valores Valence y arousal en los conjuntos fusionados

6.3.4. Archivos de audio

Los archivos de audio se sometieron a un sencillo proceso que incluía:

- Conversión de tipo `MP3` a `WAV`.
- Down-sampling `44.100 kHz` → `22.050 kHz`.
- Aumento de datos (sobre archivos de audio)

Antes, en cada carpeta raíz (`$PMEMO:` y `$DEAM:`) de los audios se crearon tres subcarpetas para almacenar los archivos resultantes: una carpeta para los audios aumentados, otra para los audios WAV y, finalmente, una carpeta para almacenar aquellos archivos producto del aumento de datos. La figura 10 muestra un ejemplo de la distribución de las carpetas para almacenar los audios originales y aquellos producto del procesamiento.

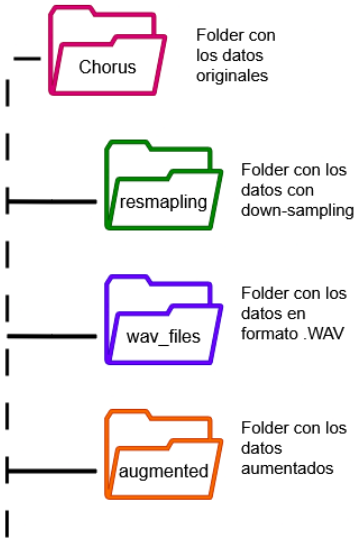

Figura 10: Distribución de folder donde se almacenan los archivos de audio procesados

Una vez que los audios son transformados, se registran las entradas en la base de datos, de los cuales se almacena el id del audio original y la ruta del archivo (la cual se construye dinámicamente).

Gracias al control de las rutas de los archivos de audio establecido, tan solo basta con modificar el proceso de construcción de rutas, sustituyendo el valor placeholder (`$PMEMO:` o `$DEAM:`) por la ruta de la carpeta de down-sampling correspondiente, y en el nombre del archivo se sustituyó el `.mp3` por `.wav`.

6.4. Obtención de las características

Para realizar el reconocimiento de emociones en una pieza musical, se diseñó un esquema de extracción de características basado en dos componentes fundamentales: las *características acústicas* derivadas de espectrogramas y las *características armónicas* extraídas a partir de progresiones de acordes. Esta doble perspectiva permite capturar tanto la dimensión temporal y espectral del audio como su estructura armónica subyacente.

6.4.1. Características basadas en espectrogramas

Para capturar la dimensión acústica de cada canción, se extrajeron cuatro tipos de espectrogramas ampliamente utilizados en tareas de análisis musical y reconocimiento emocional: **Chromagramas**, **CQT (Constant-Q Transform)**, **Espectrogramas Mel** y **Tempogramas**. Estas representaciones fueron calculadas utilizando la biblioteca `librosa` [62]. El proceso de extracción se representa visualmente el figura 11

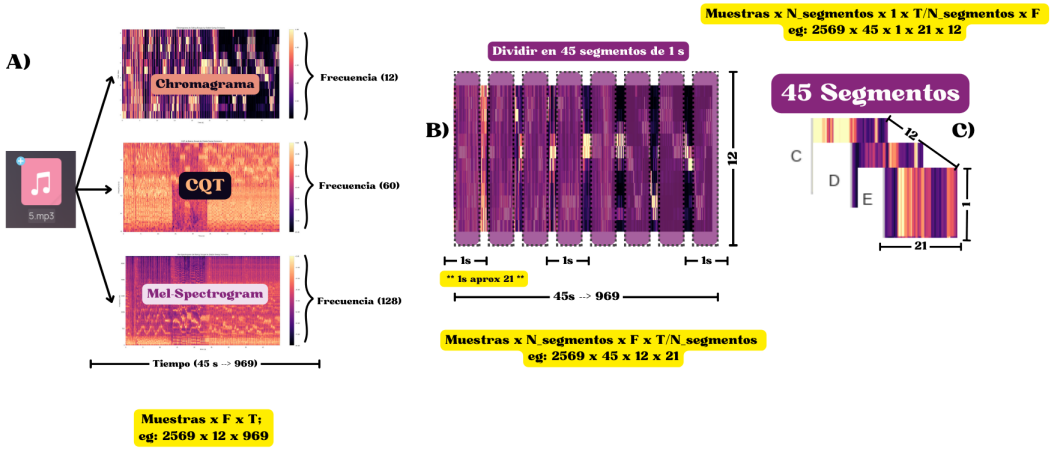

Figura 11: Fusión de modelos; A) Obtención de espectrogramas con padding; B) Segmentar en 45 sub-espectrogramas iguales; C) Redimnesionamiento de cada segmento.

Generación de espectrogramas: Cada archivo de audio (original y aumentado) fue cargado a una frecuencia de muestreo fija de 22,050 kHz. Una vez cargado el audio, se obtuvo su espectrograma correspondiente por medio de la librería `librosa`. Para estandarizar la entrada, todos los espectrogramas fueron recortados o rellenados (padding) hasta una duración total de 45 segundos. Esta longitud garantiza uniformidad en el número de frames temporales generados por los espectrogramas.

Los parámetros empleados para la obtención de espectrogramas fueron:

- **Frecuencia de muestreo:** sr = 22,050kHz
- **Tamaño de la venatana (`n_fft`):** 2048 muestras
- **Tamaño del hop (`hop_length`):** 1024 muestras

Con estos parámetros es posible calcular la cantidad de frames que el espectrograma tendrá para 45 segundos de audio, la ecuación 25 muestra el calculo de frames.

$$n_{\text{frames}} = \left\lceil \frac{\text{duration\_seconds} \times sr}{\text{hop\_size}} \right\rceil \tag{25}$$

En donde:

- duration_seconds: Duración total del audio en segundos (es decir, 45 s).
- sr: Frecuencia de muestreo, es decir, cuántas muestras por segundo toma el sistema ($sr = 22\,050$ Hz).

- hop_size: Tamaño del salto (hop size) en muestras, que indica cuántas muestras se avanza entre frames consecutivos (aquí, hop_size = 1 024).
- $\lceil \cdot \rceil$: Función "techo" o "ceiling", que siempre redondea hacia arriba al entero más cercano.

Para el ejemplo concreto:

$$\frac{\text{duration\_seconds} \times sr}{\text{hop\_size}} = \frac{45 \times 22\,050}{1\,024} = \frac{992\,250}{1\,024} \approx 968{,}9941\ldots$$

Como el resultado de esa división no es un número entero, aplicamos la función de redondeo hacia arriba:

$$n_{\text{frames}} = \lceil 968{,}994140625\ldots \rceil = 969$$

De esta forma, al tomar en cuenta 45 s, un sr de 22 050 Hz y un hop_size = 1 024, obtenemos $n_{\text{frames}} = 969$. El uso de $\lceil \cdot \rceil$ garantiza que siempre cubramos completamente la duración del audio, aunque el último frame necesite aplicarse con padding (ceros) para completarse.

Chromagramas (12×969)**:**

- **Número de filas:** 12. Cada fila corresponde a un semitono de la escala cromática, es decir, a las clases de notas musicales (C, C♯/D♭, D, . . . , B).
- **Número de columnas:** 969, igual al número de frames temporales calculados en la ecuación (25).
- **Procedimiento:**
  (a) Se calcula la STFT de la señal de audio con ventana de 2048 muestras y hop de 1024.
  (b) El espectrograma resultante se agrupa en 12 bandas logarítmicas (cada banda abarca las frecuencias correspondientes a un semitono).

Constant-Q Transform (CQT) (60×969)**:**

- **Número de filas:** 60. Cubren 5 octavas completas (de `C2` a `C7`), lo cual da $5 \times 12 = 60$ bins, cada uno correspondiente a un semitono en escala logarítmica.
- **Número de columnas:** 969,se mantienen los mismos frames temporales que en la STFT (hop de 1024 muestras).
- **Procedimiento:**
  (a) Se calcula primero la STFT con $n_{\text{fft}} = 2048$ y hop = 1024, obteniendo un espectrograma lineal de dimensión $\big(1025,\ 969\big)$.

(b) Cada columna de ese espectrograma lineal se remapea en 60 bandas logarítmicas, aplicando filtros ponderados cuya resolución relativa Q es constante.

Mel-Spectrogramas (128×969):

- **Número de filas:** 128. Se define un banco de 128 filtros mel distribuidos entre 0 y $sr/2 = 11025$ Hz, escalados según la percepción humana (escala mel).
- **Número de columnas:** 969, se emplea la misma segmentación temporal que la STFT (hop de 1024 muestras).
- **Procedimiento:**
  (a) Se calcula la STFT con $n_{\text{fft}} = 2048$ y hop $= 1024$, obteniendo un espectrograma lineal de $\big(1025,\ 969\big)$.
  (b) Se construye un banco de 128 filtros triangulares en escala mel. Por cada frame temporal (columna), se multiplica la magnitud espectral por esos 128 perfiles para obtener un vector de 128 coeficientes mel.

Tempogramas ($n_{\textbf{tempo\_bins}} \times 969$):

- Primero se extrae la *onset envelope* o envolvente de transitorios, calculada a partir de la STFT con ventana de 2048 y hop de 1024, lo que produce 969 valores de energía de onset (uno cada hop).
- Sobre esa envolvente se realiza un análisis corto en el dominio de la frecuencia de pulso por medio de autocorrelación en ventanas de M frames, desplazando cada ventana de M frames en pasos de hop igual al original (46 ms).
- El resultado es una matriz de $\big(n_{\text{tempo\_bins}},\ 969\big)$, donde $n_{\text{tempo\_bins}}$ depende de cuántas frecuencias rítmicas (BPM) se deseen cubrir.

Segmentación espacial:

Los espectrogramas son una representación visual del cambio de las frecuencias a lo largo del tiempo. En una visión simplista, cada columna del espectrograma refleja el estado del evento en un espacio temporal específico. De este modo, para que los modelos captaran la evolución temporal de los espectrogramas, cada uno fue dividido en 45 segmentos iguales. Estos segmentos se generan a lo largo del eje temporal.

De esta forma, dado que 969 frames $\approx$ 45 segundos, entonces dividir $\frac{969 \text{ frames}}{45} \approx$ 21,53 frames, logrando un redimensionamiento controlado y asegurando que cada segmento tuviese una dimensión temporal de exactamente 21 frames, pues se aplicó un recorte o truncamiento en esta cantidad de frames por segmento. Por lo tanto, la dimensión de cada segmento fue de (frecuencia $\times$ 21). Donde `frecuencia` corresponde al número de bins espectrales (filas) en cada tipo de espectrograma. Para ello se definieron las siguiente variables:

- $N_segmentos$: número de segmentos (45).
- F: dimensión frecuencia del espectrograma (12 para Chroma, 60 para CQT, 128 para Mel).
- T: dimensión temporal total del espectrograma con padding aplicado (969 frames).
- OG_T: número de frames antes de aplicar padding.
- $NT = T//N_segmentos$: número de frames por segmento (división entera).
- $FR = T\,\%N_segmentos$: número de frames residuales no asignables de forma equitativa.

El tratamiento de estos frames residuales `FR` es esencial para evitar sesgos temporales y garantizar una correcta segmentación simétrica. El algoritmo para su descarte se detalla en la figura 12

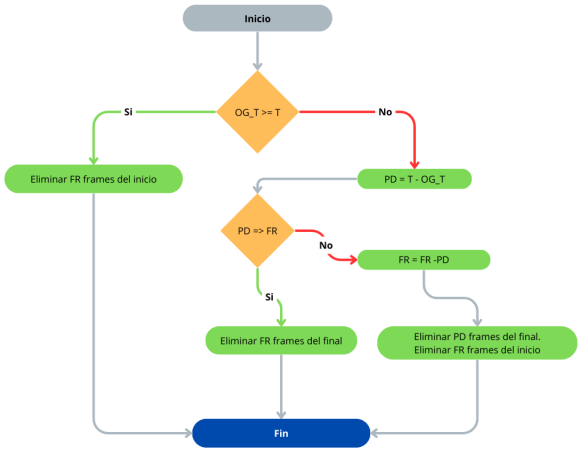

Figura 12: Diagrama de flujo para la selección de frames a descartar en el proceso de segmentación de espectrogramas.

El algoritmo sigue esta lógica:

- Si el número original de frames (OG_T) es mayor o igual a T, se eliminan FR frames desde el inicio.
- Si $OG_T < T$, se calcula el número de frames añadidos por padding:

$$PD \;=\; T - OG_T.$$

- Si $PD \geq FR$, se eliminan FR frames al final del espectrograma (sólo del padding).

- Si $PD < FR$, se eliminan primero PD frames al final (del padding) y luego $FR - PD$ frames desde el inicio (del contenido original).

Esta lógica asegura una distribución equitativa de los frames válidos en los segmentos finales, Prioriznado la eliminación de frames iniciales y cuidando mantener los frames finales del segmento, los cuales suelen contener las resoluciones musicales.

El resultado final es una matriz segmentada de forma uniforme, con dimensiones por espectrograma de $(45, 1, \texttt{NT}, F)$.

Donde NT es el número de frames por segmento (21), y F es la resolución de frecuencia. Esta estructura es la entrada directa para redes convolucionales 2D en el modelo propuesto.

En la figura 13 se muestra un caso práctico del proceso de segmentación temporal sobre un espectrograma `CQT` generado con una duración de 44 segundos. Inicialmente, la dimensión temporal del espectrograma era de 949 frames, y se aplicó padding hasta alcanzar los 969 frames necesarios. La división en 45 segmentos genera una partición entera como se ve en la ecuación 26 dejando un residuo como el de la ecuación 27.

$$NT = \left\lfloor \frac{969}{45} \right\rfloor = 21 \quad \text{frames por segmento} \tag{26}$$

$$FR = 969 \mod 45 = 24 \quad \text{frames residuales} \tag{27}$$

Dado que el padding fue de 20 frames, se procede a eliminar primero los 4 frames restantes desde el inicio del espectrograma original y luego los 20 frames del final (correspondientes al padding). Este procedimiento garantiza que la dimensión temporal final sea múltiplo de 45 y que los segmentos generados tengan exactamente 21 frames cada uno, preservando la homogeneidad temporal.

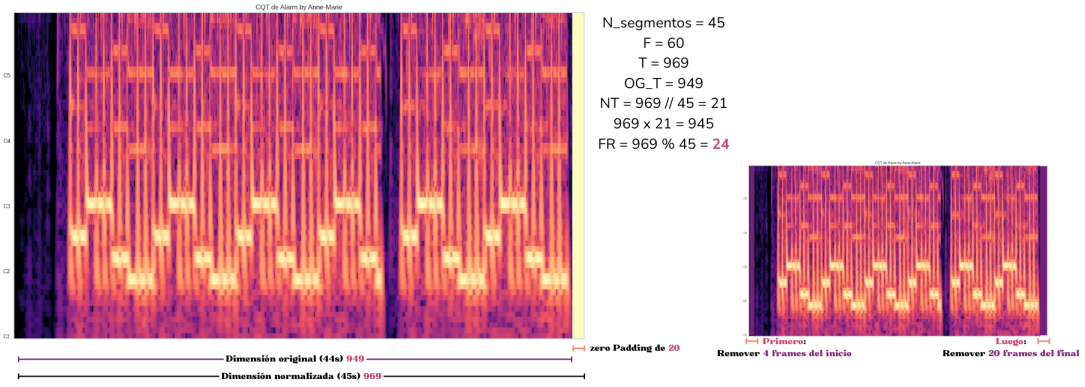

Figura 13: Proceso de segmentación de espectrogramas en 45 partes iguales; ejemplo con un espectrograma CQT.

Máscaras para los espectrogramas: Al momento de crear espectrogramas con padding, es importante identificar la información del contenido original de la información producto del rellenado con ceros. Para lograrlo, se aplica el siguiente algoritmo:

1. Partimos de un lote de espectrogramas de forma $\big(N,\ F,\ T\big)$, donde N es el tamaño del lote, F el número de bandas en frecuencia y T el número total de frames (después de haber agregado padding en el eje temporal).

2. Se elige un número fijo de segmentos temporales deseados, num_seg.

3. Se calcula cuántos frames de espectrograma corresponden, en promedio, a cada segmento:
$$\text{num\_frames} \ = \ \left\lfloor \frac{T}{\text{num\_seg}} \right\rfloor .$$
De esta forma, dividimos los T frames en num_seg bloques temporales iguales (o casi iguales).

4. Inicializamos la máscara como una matriz de ceros:
$$\text{frame\_mask} \ = \ \mathbf{0}_{\,N\times \text{num\_seg}}.$$
Cada fila i representará la máscara binaria para el espectrograma i-ésimo.

5. Disponemos de un vector og_dims $\in \mathbb{R}^N$, donde og_dims$[i] = a_i$ indica el número real de frames *originales* (sin padding) del espectrograma i.

6. Para cada ejemplo i, calculamos cuántos segmentos temporales cubre el contenido original usando

$$\text{frame\_limit}_i \;=\; \Big\lfloor \text{round}\big(a_i/\text{num\_frames}\big) \Big\rfloor.$$

Por ejemplo, si el espectrograma original tiene $a_i = 100$ frames y num_frames $= 2$, entonces $\text{frame\_limit}_i = \text{round}(100/2) = 50$.

7. Nos aseguramos de no superar el número de segmentos:

$$\text{masked\_frames}_i = \text{mín}\big(\text{frame\_limit}_i,\ \text{num\_seg}\big).$$

8. Finalmente, llenamos con 1 los primeros masked_frames_i segmentos de la fila i:

$$\text{frame\_mask}[i,\ 0 : \text{masked\_frames}_i] = 1.$$

De esta forma, los segmentos correspondientes a la parte original del espectrograma quedan marcados con valor 1, mientras que los segmentos que provienen exclusivamente del padding permanecen en 0.

Con esta máscara se logra identificar la información del espectrograma que contiene datos reales y evita que el padding sea interpretado como parte de la señal.

6.4.2. Características simbólicas (*acordes*)

Para la extracción de características basadas en la estructura armónica de las canciones se siguió un proceso que comprende los siguientes pasos: extracción de los acordes a partir de los archivos de audio, estimación de la tonalidad, creación del modelo de embeddings base, construcción y armonización de la escala, y la construcción de los embeddings con tokens estructurados. Este proceso se describe de manera gráfica en la figura 14.

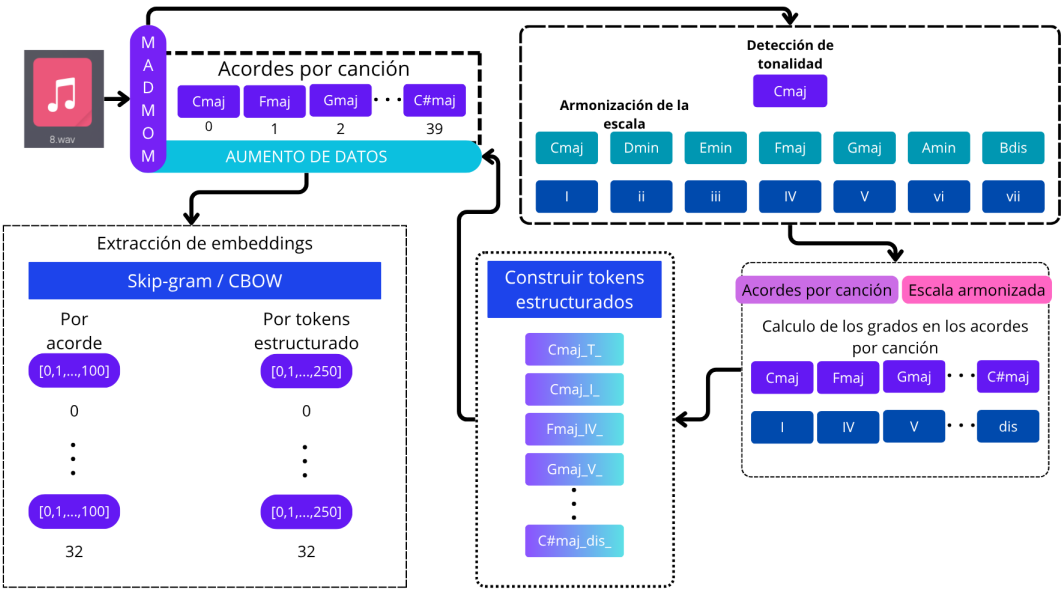

Figura 14: Diagrama del proceso en la extracción y codificación de características simbólicas basadas en acordes.

En teoría musical, la estructura armónica de una canción es clave para comprender el manejo de emociones en la música. Por ello, se implementó un proceso para la extracción de acordes, tonalidad y su representación funcional mediante grados a partir de los archivos de audio.

Para la estimación de acordes y tonalidad se utilizó la herramienta `madmom` [77], la cual permitió realizar la detección de acordes a partir de la señal de audio extraída de los archivos WAV. Además, también se obtuvo una aproximación de la tonalidad global de la canción.

La detección automática de acordes generó una serie de cadenas textuales correspondientes a la progresión armónica de la canción. Estas secuencias se transformaron en una lista de cadenas con la forma:

$$"Cmaj","Gmaj","Fmaj","Emin","Cmaj"$$

Cabe destacar que la librería de madmom tiene una gran limitación, pues solo es capaz de identificar acordes mayores y menores. Esto implica que acordes complejos como los aumentados, disminuidos o de séptima no sean contemplados en la salida del sistema. Esto simplifica las estructuras armónicas, pero al mismo tiempo limita la riqueza armónica original de una obra.

Dado que los nombres de acordes pueden estar expresados en formas enarmónicas equivalentes (por ejemplo, `D#min` y E♭min), se realizó un proceso de **normalización enarmónica** para unificar todas las notaciones a su forma con **sostenido (#)**. Esto reduce la redundancia en el vocabulario de acordes y mejora la calidad de los embeddings aprendidos. La equivalencia se basa en principios musicales estándar, como:

$$C\sharp \equiv D\flat, \quad D\sharp \equiv E\flat, \quad F\sharp \equiv G\flat, \quad G\sharp \equiv A\flat, \quad A\sharp \equiv B\flat.$$

De esta forma, se garantiza que progresiones armónicas funcionalmente idénticas no se representen con etiquetas distintas debido a notaciones alternativas.

Representación Vectorial de Acordes mediante Embeddings: Inspirado en técnicas de procesamiento de lenguaje natural (PLN) [37], [38], se aplicaron modelos *Word2Vec* basados en coocurrencias para representar las representaciones textuales de los acordes por medio de un espacio continuo. Esta técnica permite capturar relaciones sintácticas y semánticas entre acordes en un contexto musical, de forma parecida a como se modelan palabras en lenguaje natural.

Como primer acercamiento a este proceso, la representación armónica de las canciones se construyó a partir de una secuencia *lineal* de cadenas de acordes, sin contemplar información acerca de la tonalidad. Así, cada canción es representada por una progresión de acordes codificada en una lista secuencial de cadenas:

$$\{\text{Emin, Gmaj, } F\sharp\text{maj, Gmaj, } F\sharp\text{maj, Emaj, Emin, Gmaj, Emin}\}$$

Sobre este corpus se entrenaron modelos de tipo Word2Vec en sus variantes *Skip-gram* y *CBOW*. Estos modelos permiten predecir un acorde a partir de su contexto (*CBOW*) o predecir el contexto a partir del acorde central (*Skip-gram*). Formalmente, cada acorde c_i se proyecta como un vector en $\mathbb{R}^N$, donde $N \in 100, 200$:

$$\mathbf{v}_{c_i} \in \mathbb{R}^N \tag{28}$$

Estos modelos capturan regularidades contextuales basadas en la co-ocurrencia de acordes dentro de una ventana de contexto.

$$Corpus = \{C(1), C(2), \ldots, C(N)\},\ C(i) = [c_1, c_2, \ldots, c_n] \tag{29}$$

donde c_i representa el símbolo de un acorde ya normalizado. Se experimentó con tamaños de ventana de $w = 5, 10, 20$, tal como se propone en trabajos previos como el de Lahnala et al. [38].

Longitud máxima y representación por canción: Cada canción fue representada como una secuencia de vectores de embedding de dimensión d:

$$Song_i = [vc_1, vc_2, \ldots, vc_n],\ v_{ck} \in \mathbb{R}^d$$

Dado que las progresiones de acordes varían en longitud, se estableció un límite máximo de 32 acordes por canción. En caso de que la progresión tuviera una longitud menor a 32 se aplicó padding con ceros.

Grados de los acordes a partir de la tonalidad de la progresión: Para enriquecer la representación armónica más allá de la coocurrencia lineal de acordes, se diseñó un método de *embeddings estructurados* que incorpora explícitamente la *tonalidad* y los *grados armónicos* de cada acorde.

Para calcular a que *grado* pertenece cada acorde de la progresión, primero se construyó una lista con los 12 sonidos de la escala cromática, solo contemplando sostenidos:

$$C, C\sharp, D, D\sharp, E, F, F\sharp, G, G\sharp, A, A\sharp, B$$

Dada esta codificación, es posible asignar a cada sonido de la escala cromática un índice en la lista, del 0 al 11. De esta forma, avanzar una posición en la lista corresponde a avanzar un *semitono (s)*, y avanzar 2 posiciones corresponde a avanzar un *tono (T)*.

Esto nos permitió realizar cálculos matemáticos para encontrar las notas de una escala dada a partir de la nota base y el patrón de la escala que desea conseguir.

Cada escala se define por una secuencia única de *tonos* y *semitonos*. Construyendo así el patrón de construcción de la siguiente forma:

- **Escala Mayor**
  - Fórmula: T - T - S - T - T - T - S
  - Patrón Numérico: `[2, 2, 1, 2, 2, 2, 1]`
- **Escala Menor Natural**
  - Fórmula: T - S - T - T - S - T - T
  - Patrón Numérico: `[2, 1, 2, 2, 1, 2, 2]`
- **Escala Menor Armónica**
  - Fórmula: T - S - T - T - S - T$\frac{1}{2}$ - S
  - Patrón Numérico: `[2, 1, 2, 2, 1, 3, 1]`
  - *Nota: El intervalo de Tono y medio ($T\frac{1}{2}$) equivale a 3 semitonos.*
- **Escala Menor Melódica (Ascendente)**
  - Fórmula: T - S - T - T - T - T - S
  - Patrón Numérico: `[2, 1, 2, 2, 2, 2, 1]`

El proceso para construir una escala es un algoritmo simple que combina una nota de inicio (la tónica), un patrón de intervalos y aritmética. Este detalla en el algoritmo 3.

Algorithm 3 Generación de Escala Musical (Versión Robusta)

Require: nombreTonica, tipoEscala
Ensure: escalaResultante

```
escalaResultante ← {}
patronSeleccionado ← ObtenerPatron(tipoEscala)
indiceActual ← ObtenerIndice(nombreTonica)
Añadir(escalaResultante, NOTAS_CROMATICAS[indiceActual]) {añade la tónica}
for cada intervalo en patronSeleccionado do
    indiceActual ← ( indiceActual + intervalo) mód 12
    Añadir(escalaResultante, NOTAS_CROMATICAS[indiceActual])
end for
return escalaResultante
```

A continuación, se muestra una ejecución manual del algoritmo para verificar su funcionamiento.

1. **ENTRADA:** `ConstruirEscala(«E»,«Mayor»)`.
2. **INICIALIZACIÓN:**
   - `escalaResultante` ← `[]`
   - `patronSeleccionado` ← `[2,2,1,2,2,2,1]`
   - `indiceActual` ← 4 (índice de "E")
   - Se añade **E** a `escalaResultante`, que queda `[E]`.
3. **ITERACIONES DEL BUCLE:**
   - **intervalo = 2:** `indiceActual` ← $(4+2)$ mód $12 = 6$. Se añade **F♯**.
   - **intervalo = 2:** `indiceActual` ← $(6+2)$ mód $12 = 8$. Se añade **G♯**.
   - **intervalo = 1:** `indiceActual` ← $(8+1)$ mód $12 = 9$. Se añade **A**.
   - **intervalo = 2:** `indiceActual` ← $(9+2)$ mód $12 = 11$. Se añade **B**.
   - **intervalo = 2:** `indiceActual` ← $(11+2)$ mód $12 = 1$. Se añade **C♯**.
   - **intervalo = 2:** `indiceActual` ← $(1+2)$ mód $12 = 3$. Se añade **D♯**.
   - **intervalo = 1:** `indiceActual` ← $(3+1)$ mód $12 = 4$. Se añade **E** (octava).
4. **SALIDA:** La lista resultante es `[«E»,«F#»,«G#»,«A»,«B»,«C#»,«D#»,«E»]`.

Armonización de la escala: La armonización consiste en asignar una cualidad de acorde a cada una de las notas de una escala. Este proceso nos da la paleta de acordes que pertenecen a una tonalidad específica.

Cada tipo de escala genera un patrón único y predecible de acordes. A continuación se presentan las listas para las cuatro escalas principales, usando las abreviaturas **maj** (mayor), **min** (menor), **dis** (disminuido) y **aug** (aumentado).

- **Escala Mayor:**
  - Patrón: `[maj, min, min, maj, maj, min, dis]`
- **Escala Menor Natural:**
  - Patrón: `[min, dis, maj, min, min, maj, maj]`
- **Escala Menor Armónica:**
  - Patrón: `[min, dis, aug, min, maj, maj, dis]`
- **Escala Menor Melódica (Ascendente):**
  - Patrón: `[min, min, aug, maj, maj, dis, dis]`

El proceso para construir la escala armonizada es una simple concatenación uno a uno, entre la escala de notas y el patrón de armonización.

Entonces, dadas la siguiente escala y su patrón de armonización:

1. **La escala de notas calculada:**

$$A, B, C, D, E, F, G\sharp$$

2. **El patrón de armonización (menor armónica):**

`[min, dis, aug, min, maj, maj, dis]`

El proceso de `concatenación`, donde a cada nota se le asigna la cualidad de acorde en la misma posición:

Nota de la Escala	Patrón Armónico	Acorde Resultante
A	min	**A min**
B	dis	**B dis**
C	aug	**C aug**
D	min	**D min**
E	maj	**E maj**
F	maj	**F maj**
G♯	dis	**G♯ dis**

Grados de una escala: La asignación de grados es el paso final y es un mapeo directo. A cada acorde de la escala armonizada se le asigna un número romano de una lista predefinida, según el modo de la escala.

- **Mayor:** `[ I, ii, iii, IV, V, vi, vii dis ]`

- **Menor Natural (nat):** `[ i, ii dis, III, iv, v, VI, VII ]`
- **Menor Armónica (arm):** `[ i, ii dis, III aug, iv, V, VI, vii dis ]`
- **Menor Melódica (mel):** `[ i, ii, III agu, IV, V, vi dis, vii dis ]`

Estos grados se guardan en una lista paralela a la escala previamente armonizada. Si bien los grados y acordes de la escala contemplan acordes disminuidos y aumentados, en la práctica nunca se encuentran estos acordes dentro de la escala, pues `madmom` simplificará estos acordes. Sin embargo, esto conlleva a que el acorde simplificado no se encuentre dentro de la escala, así que simplemente se marcará como un `dis` que indica una disonancia, guardando así la función de este acorde dentro de la escala.

Calculo de grados en una progresión: El objetivo final de este proceso es analizar una progresión de acordes dentro de una tonalidad específica. Este proceso automatizado utiliza las escalas armonizada y de grados que generamos previamente como listas de referencia.

El método consiste en recorrer la progresión acorde por acorde. Para cada uno, se busca su posición en la escala armonizada de la tonalidad. Si se encuentra, se toma el grado que está en la misma posición en la lista de grados. Si no se encuentra, se etiqueta como una disonancia (**dis**). Generando así dos listas, una con los acordes de la progresión y otra que guarda, en la misma posición que la lista de acordes, los grados de los acordes de acuerdo con la tonalidad.

En el caso de las tonalidades menores, existen tres modos. Sin embargo, la detección automatizada no hace distinción de qué modo es el que se usa, por lo que el algoritmo de análisis se refina con una **lógica de búsqueda jerárquica** para las tonalidades menores.

Para cualquier acorde en una progresión en tonalidad menor, el algoritmo intentará encontrar una coincidencia en el siguiente orden de precedencia:

1. **Escala Menor Natural:** Es la base de la tonalidad.
2. **Escala Menor Armónica:** Si no se encuentra en la natural, se busca aquí. Es la fuente más común de acordes prestados, especialmente el V grado mayor.
3. **Escala Menor Melódica:** Si aún no se encuentra, se busca en la melódica.
4. **Disonancia (dis):** Si el acorde no existe en ninguna de las tres escalas, se marca como disonancia.

El grado que se asigna corresponderá a la primera escala en la que se encuentre el acorde.

Algorithm 4 Análisis de Progresión con Lógica Jerárquica

Require: Tonalidad, ProgresionAcordes
Ensure: analisisResultante

```
analisisResultante ← []
if Tonalidad es MAYOR then
    escalaArmonizada ← GenerarEscalaArmonizada(Tonalidad)
    escalaDeGrados ← GenerarEscalaDeGrados(Tonalidad)
    for cada acorde en ProgresionAcordes do
        {Aquí iría la lógica básica de mapeo}
        ...
    end for
else
    if Tonalidad es MENOR then
        escalaArm_Nat ← GenerarEscalaArmonizada(Tonalidad,'Natural')
        grados_Nat ← GenerarEscalaDeGrados(Tonalidad,'Natural')
        escalaArm_Armonica ← GenerarEscalaArmonizada(Tonalidad,'Armonica')
        grados_Armonica ← GenerarEscalaDeGrados(Tonalidad,'Armonica')
        escalaArm_Melodica ← GenerarEscalaArmonizada(Tonalidad,'Melodica')
        grados_Melodica ← GenerarEscalaDeGrados(Tonalidad,'Melodica')
        for cada acorde en ProgresionAcordes do
            encontrado ← falso
            indice ← BuscarIndice(acorde, escalaArm_Nat)
            if indice existe y no encontrado then
                Añadir(analisisResultante, grados_Nat[indice])
                encontrado ← verdadero
            end if
            indice ← BuscarIndice(acorde, escalaArm_Armonica)
            if indice existe y no encontrado then
                Añadir(analisisResultante, grados_Armonica[indice])
                encontrado ← verdadero
            end if
            indice ← BuscarIndice(acorde, escalaArm_Melodica)
            if indice existe y no encontrado then
                Añadir(analisisResultante, grados_Melodica[indice])
                encontrado ← verdadero
            end if
            if no encontrado then
                Añadir(analisisResultante,'dis')
            end if
        end for
    end if
end if
return analisisResultante
```

Construcción del Modelo y Tokens Estructurados:

Para generar los tokens estructurados y construir un nuevo modelo Word2Vec, se obtienen los grados de cada acorde dentro de la progresión, para posteriormente combinar ambas fuentes de información (acorde y función armónica) en cada token.

Para cada canción, se validó que la lista de acordes y la lista de grados tuvieran la misma longitud. El primer token se construye con la tonalidad, generando un *token de tonalidad* con el formato `{tonalidad}_T_`, por ejemplo, `Emin_T_`. Tras esto, para cada par (`acorde`, `grado`) se generaba un token de la forma `{acorde}_{grado}_`, como `Emin_i_`, `Gmaj_III_` o `Amaj_dis_` (cuando el acorde no pertenece a la escala). De este modo, la lista final de tokens para la progresión de una canción luce así:

$$\big[\,\texttt{tonalidad\_T\_},\ \texttt{acorde\_grado\_}_1,\ \texttt{acorde\_grado\_}_2,\ \ldots,\ \texttt{acorde\_grado\_}_L\big].$$

Al concluir, cada canción queda asociada a su secuencia completa de tokens estructurados. Generando así el corpus para entrenar el modelo Word2Vec.

Entrenamiento del Modelo Word2Vec: Utilizando el corpus de tokens generados, se entrenaron modelos Word2Vec con parámetros fijos: dimensión del embedding [150,250,350], ventana de contexto [9,18,36], 30 épocas de entrenamiento, Skipgram y CBOW, y `min_count = 1`. El resultado es un modelo que asocia cada token ("`tonalidad_T_`" o "`acorde_grado_`") a un vector en $\mathbb{R}^{250}$. Este modelo se guarda en un archivo, por ejemplo `structured_skipgram_model_250_18_30.npy`, de modo que, si existe, simplemente se carga para evitar reentrenar.

Estructuras Producto del Entrenamiento: Al finalizar el entrenamiento, el vocabulario de tokens $\mathcal{V}$ incluye todas las tonalidades con su sufijo "_T_" y cada token "acorde_grado" correspondiente. Cada token $t \in \mathcal{V}$ está representado por un vector $\mathbf{u}_t \in \mathbb{R}^N$, en el cual:

- Si t es de tipo **"`tonalidad_T_`"**, $\mathbf{u}_t$ codifica la representación de la tonalidad.
- Si t es de tipo **"`acorde_grado_`"**, $\mathbf{u}_t$ captura tanto el nombre del acorde como su función dentro de la tonalidad.

Para cada canción con L acordes, construimos una matriz

$$\mathbf{U}_S \;=\; \begin{bmatrix} \mathbf{u}_{\texttt{tonalidad\_T\_}} \\ \mathbf{u}_{c_1\_\gamma_1\_} \\ \vdots \\ \mathbf{u}_{c_L\_\gamma_L\_} \end{bmatrix} \;\in\; \mathbb{R}^{(L+1)\times N}.$$

La primera fila corresponde al vector de tonalidad, mientras que cada fila subsiguiente es el vector asociado a cada token "acorde_grado". Esta matriz $\mathbf{U}_S$ se emplea directamente en modelos secuenciales (por ejemplo, un BiLSTM), agregando padding cuando $L < L_{\text{máx}}$.

Entrada con Embeddings multiples: Los embeddings estructurados contemplan la tonalidad al inicio de la progresión de la forma: `tonalidad_T_`. Sin embargo, este token no se repite a lo largo de la progresión.

Por ello, para contemplar en todo momento el peso de la tonalidad, se obtienen el embedding base de la tonalidad junto con los embeddings de los tokens estructurados, creando así una entrada con dos diferentes tipos de embeddings, combinando así la información completa de la progresión de acordes (embeddings estructurados) y la representación vectorial de la tonalidad (embedding base). Así, la entrada de cada canción consta de:

- Una matriz de embeddings $\mathbf{X}_i \in \mathbb{R}^{L_{\text{máx}} \times N}$, que cubre toda la progresión de acordes hasta una longitud fija $L_{\text{máx}} = 32$. Cuando la progresión real tiene menos de 32 tokens, aplicamos padding con ceros. Asimismo, generamos una máscara $\text{mask}_i \in \{0,1\}^{L_{\text{máx}}}$ que indica con 1 las posiciones correspondientes a tokens válidos y con 0 las de padding.

- Un vector de tonalidad $e_i^{\text{ton}} \in \mathbb{R}^{100}$, obtenido previamente mediante un modelo base de Word2Vec entrenado únicamente sobre tokens de tonalidad.

- El par de valores $\left[v_i^{\text{val}},\ v_i^{\text{aro}}\right] \in \mathbb{R}^2$ que representa el *valence* y *arousal* objetivo para esa canción.

Resumen de Salida: Para cada canción, el preprocesamiento genera tres tensores:

$$X_i \in \mathbb{R}^{32\times 250}, \quad \text{mask}_i \in \{0,1\}^{32}, \quad e_i^{\text{ton}} \in \mathbb{R}^{100}.$$

Junto con el vector $\left[v_i^{\text{val}},\ v_i^{\text{aro}}\right] \in \mathbb{R}^2$, estos datos conforman un *minibatch* que alimenta directamente el modelo de regresión emocional, incorporando tanto la progresión completa de acordes (y sus funciones) como la representación numérica de la tonalidad.

6.5. Aumento de datos

6.5.1. Transposición de acordes

Al implementar una estructura de tokens estructurados, se adoptó un enfoque de aumento de datos basado en la transposición de la tonalidad por intervalos. Este método se fundamenta en una técnica musical común: desplazar acordes hacia arriba o abajo en el eje de alturas, conservando su estructura interna (modo mayor o menor).

Lógica básica para la transposición: Para enriquecer el conjunto de datos de entrenamiento y asegurar que el modelo aprenda a reconocer patrones armónicos independientemente de la tonalidad, se implementó una estrategia de aumento de datos basada en la transposición. Este proceso genera nuevas progresiones musicalmente coherentes al desplazar una progresión existente a diferentes tonalidades, la figura 15 muestra un ejemplo gráfico de como funciona esta técnica.

El método se basa en el análisis de grados previamente realizado y sigue un procedimiento estructurado para garantizar la correcta correspondencia armónica en la nueva tonalidad.

Dada una progresión original, su tonalidad y su análisis de grados, el proceso para generar una nueva progresión transpuesta es el siguiente:

1. **Transposición de la Tonalidad:** Se elige un intervalo de transposición (medido en semitonos) y se aplica a la tónica de la tonalidad original. Esto define la nueva tonalidad base. Por ejemplo, transponer C maj en '+2' semitonos resulta en la nueva tonalidad de D maj. Un ejemplo gráfico de transposición de un acorde se puede observar en la figura 15.

2. **Generación de la Nueva Escala de Referencia:** Utilizando los algoritmos ya definidos, se genera la escala armonizada completa para la nueva tonalidad. Esta escala servirá como "diccionario"para construir la nueva progresión.

3. **Reconstrucción de la Progresión Diatónica:** Se recorre la lista de **grados** de la progresión original. Para cada grado (ej. I, V, vi), se busca el acorde que ocupa esa misma posición en la nueva escala armonizada generada en el paso anterior. Este mapeo directo asegura que la función armónica de los acordes se preserve.

4. **Manejo de Acordes Disonantes (`dis`):** Los acordes que fueron marcados como `dis` no tienen un grado diatónico, por lo que no pueden mapearse como en el paso anterior. En su lugar, se transponen cromáticamente:

   - Se toma la nota raíz del acorde disonante original (ej. la nota 'D' del acorde 'D maj').
   - Se busca la posición de esta nota en la lista de los 12 sonidos cromáticos.
   - Se desplaza su índice según el intervalo de transposición (ej. '+2' semitonos).
   - La nueva nota raíz y la cualidad del acorde original (mayor o menor) forman el nuevo acorde disonante transpuesto.

A cada progresión se le aplicó transposición en 4 intervalos dados. Estos intervalos se encuentran detallados en la tabla 6.

Cuadro 6: Intervalos de Transposición (Estilo Minimalista)

Nombre del Intervalo	**Valor en Semitonos**
Segunda mayor	2
Tercera mayor	4
Cuarta justa	5
Quinta justa	7

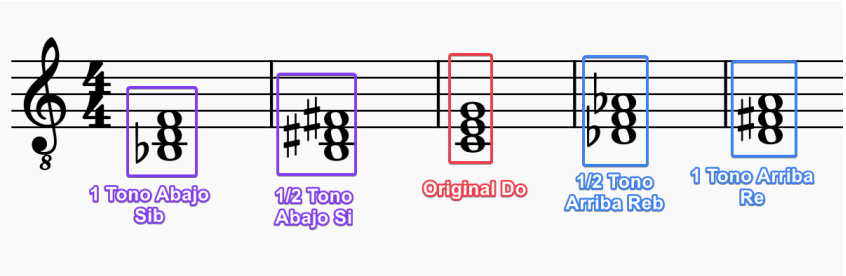

Figura 15: Transposición de un acorde de Do mayor, $\frac{1}{2}$ tono y 1 tono arriba y abajo. Notación musical clásica con pentagrama.

La tabla 7 muestra un ejemplo de cómo se aplica la transposición en una de las canciones del dataset de *PMEmo*.

Cuadro 7: Aumento de datos a la canción de Ï Have Questions"de la artista C̈amila Cabello"del conjunto de datos de Pmemo (solo los 4 primeros acordes).

Transposición	**Tonalidad**	**Progresión Resultante**
Original	C♯ maj	`A♯ min, F♯ maj, G♯ maj, A♯min,` `vi, IV, V, vi`
Segunda menor	D maj	`B min, G maj, A maj, B min` `vi, IV, V, vi`
Tercera mayor	F maj	`D min, A♯ maj, C maj, D min` `vi, IV, V, vi`
Cuarta justa	F♯ maj	`D♯ min, B maj, C♯ maj, D♯ min` `vi, IV, V, vi`
Quinta justa	G♯ maj	`F min, C♯ maj, D♯ maj, F min` `vi, IV, V, vi`

A partir de las muestras originales, este método permitió generar versiones transpuestas de cada canción. A pesar de la limitación impuesta por el número reducido de acordes posibles, el corpus aumentó de manera significativa, pasando de unas 2569 muestras originales a 12,845 progresiones únicas. Esta expansión mejoró la robustez del entrenamiento sin modificar la distribución emocional de las canciones.

6.5.2. Técnicas de aumento de datos en archivos de audio

Para incrementar la cantidad de datos disponibles y mejorar la generalización del modelo sin introducir cambios significativos en la percepción emocional de los audios, se implementaron técnicas clásicas de aumento de datos directamente sobre la señal de audio. Estas transformaciones se aplicaron antes de la extracción de espectrogramas.

Se aplicaron dos métodos principales de transformación de la señal:

Time stretching: Consiste en modificar la velocidad de reproducción del audio sin alterar su tono. Se aplicaron cuatro configuraciones:

- $0{,}81 \times$ velocidad
- $0{,}93 \times$ velocidad
- $1{,}07 \times$ velocidad
- $1{,}23 \times$ velocidad

Este método permite simular interpretaciones más lentas o rápidas de una misma pieza musical, respetando su estructura tonal.

Time shifting: En esta técnica, los últimos 5 segundos del audio se recortan y se colocan al inicio del archivo, generando una nueva versión del mismo audio con un reordenamiento temporal.

Este tipo de desplazamiento es útil para redes neuronales que exploran la evolución temporal, ya que modifica el punto de entrada sin alterar el contenido total.

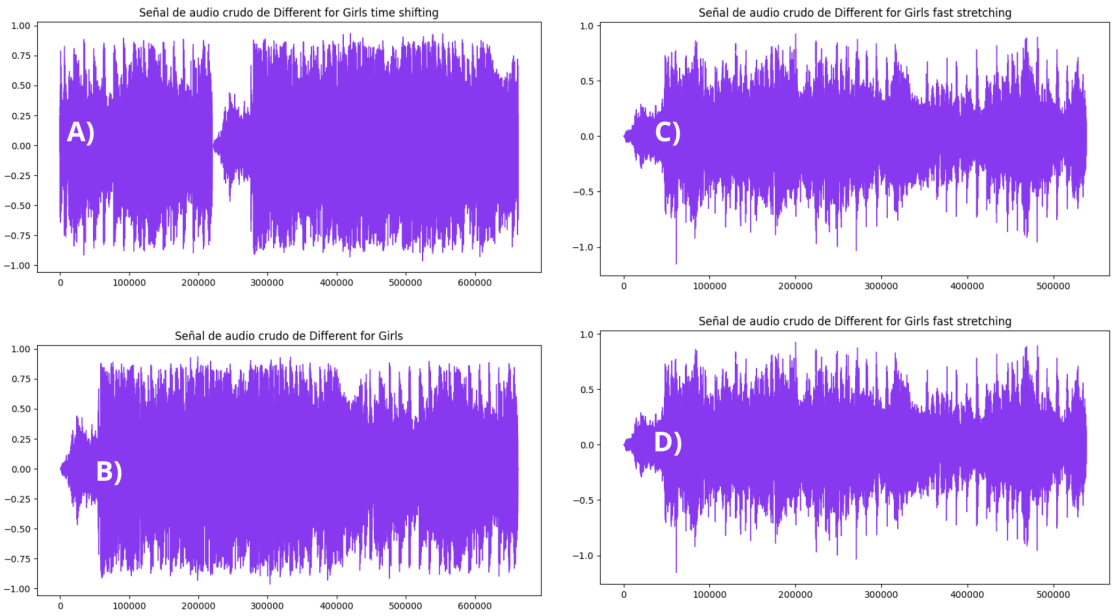

Figura 16: Ejemplo de las señales de audio de un elemento aumentado con time stretching y time shifting: A) Time shifting; B), C) y D) Time stretching $\times(0{,}81, 0{,}93, 1{,}07)$. Canción "Different for Girls".

6.6. Características profundas

Como antesala del reconocimiento de emociones, de cada tipo de características, acústicas y simbólicas (estructuras armónicas), se extraen las características profundas por medio de modelos profundos. El objetivo de este proceso es fusionar las características con diferentes dimensiones en una sola.

6.6.1. Características profundas de las estrcutras armonicas

Para la características armónicas se tiene un vector de embeddings con tokens estructurados y un vector de embeddings para representar la tonalidad de la progresión. Además de un vector máscara para identificar aquellos espacios rellenados con ceros. Obteniendo una entrada como:

$$X_i \in \mathbb{R}^{32\times 250}, \quad \text{mask}_i \in \{0,1\}^{32}, \quad e_i^{\text{ton}} \in \mathbb{R}^{100}.$$

El extractor de características armónicas recibe, para cada lote de datos, tres tensores principales:

- **X**: Secuencias de embeddings de acordes, de forma $\text{B} \times \text{T} \times D_{\text{in}}$.
- **mask**: Máscara binaria de tamaño $\text{B} \times \text{T}$ que indica qué posiciones de la secuencia son válidas (1) y cuáles son padding (0).
- **tonality**: Embedding de la tonalidad, de tamaño $\text{B} \times 100$, constante para toda la progresión de acordes.

En donde:

- $T = 32$ es la longitud máxima de la secuencia de tokens de acordes.
- $D_{\text{in}} = 250$ es la dimensión de los embeddings de cada acorde (estructurado).

Las redes *LSTM bidireccionales (BiLSTM)* procesan secuencias de acordes en ambos sentidos, permitiendo que el modelo aprenda dependencias contextuales tanto pasadas (hacia atrás) como futuras (hacia adelante). Esto es fundamental, pues en la estructura armónica un acorde no es un elemento aislado, ya que está condicionado por su contexto completo. La figura 17 muestra el diagrama de la arquitectura del modelo extractor para las características simbólicas.

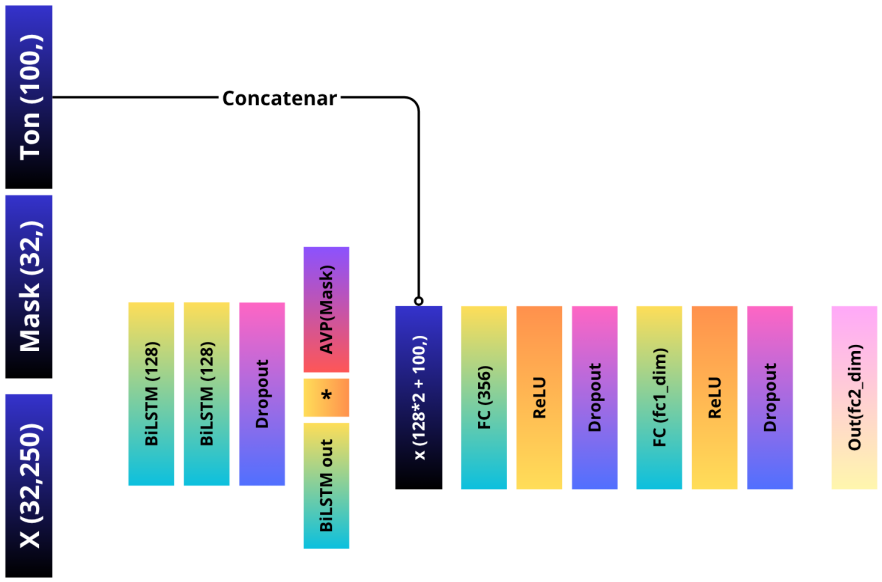

Figura 17: Extractor de características profundas y fusión de embeddings modelo BiLSTM

Descripción capa por capa: A continuación se enumeran los bloques clave del modelo, en orden de procesamiento (seis capas principales):

- **BiLSTM bidireccional (2 capas, dropout)**
  - Recibe **X** y procesa las secuencias de longitud variable (usando `mask` para ignorar padding).
  - Genera una salida de tamaño $B \times T \times (2 \cdot 128)$.
- **Pooling promedio enmascarado**
  - Se aplica la máscara sobre la salida de la LSTM para anular posiciones de padding.
  - Se calcula el promedio a lo largo de la dimensión temporal, obteniendo un vector $B \times (2 \times 128)$.
- **Concatenación con embedding de tonalidad**
  - El vector obtenido tras el pooling se concatena con **tonality** (dimensión 100), formando un tensor $B \times (2 \cdot 128 + 100)$.
- **Capa densa 1 (FC1)**
  - Proyecta la concatenación anterior desde $(2 \cdot 128 + 100)$ a fc1_dim.

- Incluye activación ReLU y dropout para evitar sobreajuste.

- **Capa densa 2 (FC2)**
  - Toma la salida de FC1 (fc1_dim) y la proyecta a fc2_dim.
  - También aplica ReLU y dropout.
  - El resultado final de FC2 se considera la "característica profunda armónica" (dimensión fc2_dim).

6.6.2. Características profundas acusticas

A continuación, se explica cómo se obtuvieron las representaciones profundas de cada tipo de espectrograma y cómo se fusionan dichas características. El extractor principal combina bloques residuales de tipo *ResNet* con bloques de atención *Squeeze-and-Excitation (SE)*. La figura 18 refleja el diagrama y la composición del modelo extractor de características acústicas.

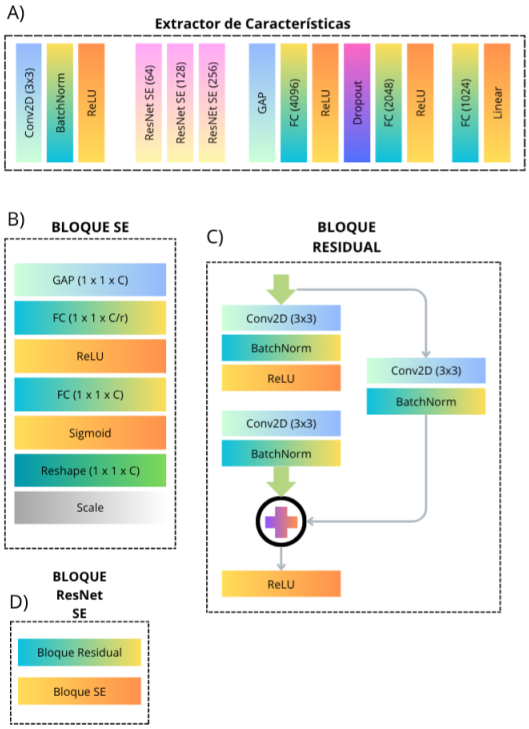

Figura 18: Extractor de características profundas ResNetSE para caracteristicas acustcias (espectrogrmas)

ResidualBlock: Cada bloque residual consta de dos convoluciones con normalización y activación, más un atajo (shortcut) que ajusta la dimensión cuando es necesario.

La estructura esencial de este bloque se puede observar en el segmento C de la figura 18.

SEBlock (Squeeze-and-Excitation): Este bloque recalibra los canales mediante atención global. De igual forma, la parte B de la figura 18 muestra el bloque completo.

ResNetSEBlock: Combina un *ResidualBlock* con un *SEBlock* tal como se muestra en la sección D de la figura 18.

En general, primero se aplica la convolución residual y luego se recalibran los canales con atención SE.

Finalmente, el extractor de características, denominado *ResNetSEExtractor*, tiene la estructura descrita en la sección A de la figura 18.

Construcción de extractores según tipo de espectrograma: Dependiendo del número de "bins" de frecuencia del espectrograma, se crea un extractor con `in_channels` equivalente a la cantidad de bins:

- **Cromagrama:** in_channels = 12.
- **CQT:** in_channels = 60.
- **Mel-spectrograma:** in_channels = 128.
- **Espectrograma temporal (TMP):** in_channels = 384.

En cada caso, el flujo es:

$$\text{Espectrograma } \big[\,(B,\ 1,\ T,\ F)\big] \xrightarrow{\text{ResNetSEExtractor}} (B,\ 1024).$$

Una vez extraídos los vectores de 1024 dimensiones para cada espectrograma, se combinan mediante una concatenación:

- **Concatenación**:

$$f_{\text{fusión}} = \big[\, f_{\text{chroma}} \,\|\, f_{\text{cqt}} \,\|\, f_{\text{mel}} \,\big] \quad \in \mathbb{R}^{3\,024}.$$

Con cuatro extractores:

$$f_{\text{fusión}} = \big[\, f_{\text{chroma}} \,\|\, f_{\text{cqt}} \,\|\, f_{\text{mel}} \,\|\, f_{\text{tmp}} \,\big] \quad \in \mathbb{R}^{4\,096}.$$

De este modo, la fusión permite aprovechar la información complementaria de cada representación espectral. Preservando la identidad de cada extractor para capas posteriores.

De manera general, el proceso de extracción y fusión de características acústicas se puede describir así:

1. **Cálculo de espectrogramas:** A partir de la señal de audio preprocesada (normalización y padding a duración fija), se obtienen:

   $$\text{Cromagrama}, \quad \text{CQT}, \quad \text{Mel-spectrograma}, \quad \text{opcionalmente TMP}.$$

   Cada uno con dimensiones $(B,\ 1,\ T,\ F)$.

2. **Extracción de características:** Para cada espectrograma se aplica el extractor correspondiente:

$$f_{\text{chroma}} = \mathcal{E}_{12}(x_{\text{chroma}}), \quad f_{\text{cqt}} = \mathcal{E}_{60}(x_{\text{cqt}}), \quad f_{\text{mel}} = \mathcal{E}_{128}(x_{\text{mel}}),$$

donde $\mathcal{E}_c$ denota un ResNetSEExtractor con c canales de entrada, y cada $f \in \mathbb{R}^{1024}$.

3. **Fusión de vectores profundos:** Se concatenan los vectores para obtener $f_{\text{fusión}}$.

4. **Entrada al modelo principal:** El vector fusionado $f_{\text{fusión}}$ se utiliza como entrada a la red que predice *arousal* y *valence*.

6.7. Modelos para el reconocimiento de emociones

La predicción final de las emociones de una canción parte de generar y cargar las características profundas (tanto acústicas como simbólicas), las máscaras asociadas a los espectrogramas y las etiquetas de *arousal* y *valence* necesarias para entrenar el modelo.

A continuación, se detalla el procedimiento para la lectura y división de los datos en conjuntos de entrenamiento, validación y prueba, empleando una partición 60 %-20 %-20 % estratificada por el identificador de cada canción.

6.7.1. Carga y división de los datos

En primer lugar, se cargan desde disco cinco conjuntos de datos principales. Las características acústicas profundas se obtienen previamente al procesar cada espectrograma (cromagrama, CQT, Mel-spectrograma y Tempogramas) con el extractor *ResNet+SE* y se almacenan en un arreglo de dimensión $N \times 1024$. De manera análoga, las características simbólicas —compuestas por los embeddings de acordes, tonalidades y grados armónicos— se guardan en otro arreglo, cuya segunda dimensión corresponde a la longitud de los vectores simbólicos. Para distinguir las posiciones de padding dentro de cada espectrograma, se dispone de un tercer arreglo binario que asigna a cada muestra una máscara de tamaño T (el número de frames temporales). Las etiquetas de emociones, es decir, los valores de *arousal* y *valence* normalizados, se encuentran reunidas en un cuarto arreglo con forma $N \times 2$. Finalmente, un archivo CSV de metadatos contiene, al menos, la columna `song_id`, que indica a qué canción corresponde cada fragmento. Al realizar la lectura, todos estos arreglos resultan tener el mismo número de filas N, de modo que para cada índice i se dispone de la tupla: (vector acústico, vector simbólico, máscara binaria, etiqueta de emociones, identificador de canción).

Una vez que los datos han sido cargados y después de verificar que comparten la misma longitud, el siguiente paso consiste en dividirlos en tres subconjuntos: entrenamiento (60 %), validación (20 %) y prueba (20 %). Para asegurar que fragmentos de una misma canción no aparezcan en más de una partición, se emplea el identificador de canción como criterio de estratificación. En la práctica, esto se logra extrayendo el vector

unidimensional de `song_id` del archivo de metadatos y pasándolo, junto con los índices de todas las muestras, a una función de partición que reserva el 60 % de las muestras para entrenamiento. El resultado es un conjunto de índices y los correspondientes fragmentos acústicos, simbólicos, máscaras y etiquetas que conforman exactamente el 60 % del total, garantizando que la proporción de canciones se mantenga equilibrada. El 40 % restante se destina a una partición temporal que servirá para generar los subconjuntos de validación y prueba. Sobre ese 40 %, se realiza una segunda división en partes iguales, de modo que cada una represente el 20 % del total original y conserve también la estratificación por canción. Al concluir el proceso, se obtienen tres conjuntos (entrenamiento, validación y prueba), cada uno con sus arreglos acústicos, símbolos, máscaras y etiquetas correspondientes, preparados para alimentar las etapas siguientes del entrenamiento del modelo.

6.7.2. Modelos predictores intermedios

Con el objetivo de probar las características extraídas, una vez obtenidas las características profundas, se ingresaron en dos modelos sencillos: un modelo denso completamente conectado y un modelo BiLSTM, cuya representación gráfica se puede observar en la figura 19.

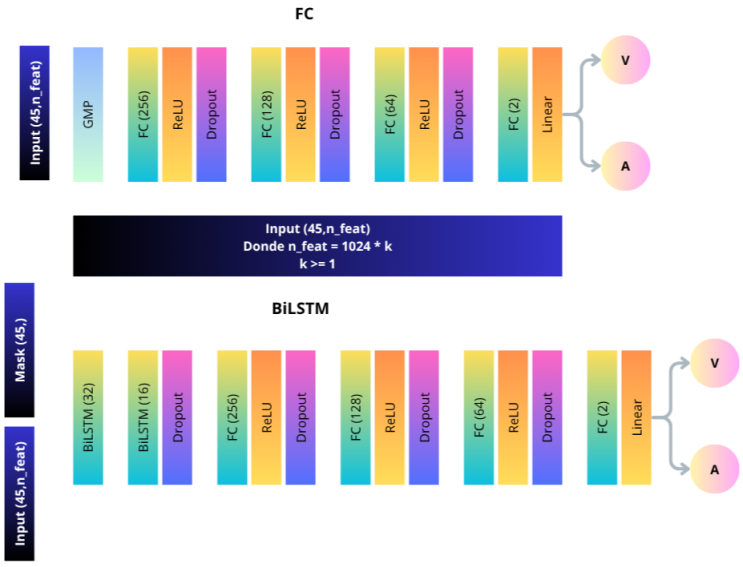

Figura 19: Modelos para la predicción de emociones valence arousal sobre caracteristicas acusticas.

En el caso de la características simbólicas solo se probaron con el modelo BiLSTM, mientras que las características acusticas se probaron con un *modelo BiLSTM* y un *modelo totalmente conectado*.

6.7.3. Modelo final y fusión de características

Como paso definitivo antes de realizar el reconocimiento de emociones, es necesario combinar las representaciones profundas obtenidas de dos fuentes: las características acústicas extraídas de los espectrogramas y las características simbólicas ya procesadas de las estructuras armónicas. Dado que los espectrogramas varían en dimensiones y los embeddings simbólicos ya se encuentran en forma de vectores profundos, cada tipo de datos se maneja con una rama específica para extraer o refinar su representación antes de fusionarlos en un bloque común que producirá las predicciones de *arousal* y *valence*.

6.7.4. Características profundas acústicas

Para las características acústicas, disponemos de un arreglo

$$X_i^{\text{ac}} \in \mathbb{R}^{T \times D_{\text{ac}}}, \quad \text{mask}_i^{\text{ac}} \in \{0,1\}^T,$$

donde T es el número máximo de segmentos temporales (frames) y D_{ac} la dimensión de cada vector producido por los extractores ResNetSE de los distintos espectrogramas. La máscara indica, para cada posición, si corresponde a señal válida (1) o a padding (0).

La rama acústica emplea dos capas de BiLSTM encadenadas, cada una con unidades bidireccionales y dropout intermedio. La primera BiLSTM recorre la secuencia completa de vectores acústicos, procesándola en ambas direcciones. Su salida se pasa a la segunda BiLSTM, que refina la representación en cada instante de tiempo. Durante el procesamiento, se usa la máscara para omitir las posiciones de padding en cada BiLSTM, de manera que las LSTM no consideren las zonas sin señal. Al concluir la segunda capa, se extraen los estados finales en las direcciones hacia adelante y hacia atrás, se concatenan y se aplica dropout adicional. Este vector concatenado, de dimensión $2H$, se proyecta mediante una capa fully-connected con activación ReLU y dropout, seguida de otra capa que reduce la dimensión a 64. De este modo, cada fragmento acústico se resume en un único vector

$$f_i^{\text{ac}} \in \mathbb{R}^{64},$$

que conserva la información temporal y espectral más relevante de todo el segmento.

6.7.5. Características profundas simbólicas

En la rama simbólica, las características ya están representadas como vectores profundos de dimensión D_{sym}, fruto de haber pasado previamente los embeddings de acordes, grados y tonalidad por un extractor BiLSTM y capas densas. Por lo tanto, no es necesario aplicar una nueva red recurrente ni utilizar máscara en esta etapa. A cada fragmento musical i le corresponde un vector simbólico

$$f_i^{\text{sym, in}} \in \mathbb{R}^{D_{\text{sym}}},$$

que ya sintetiza la progresión armónica.

La rama simbólica se limita a refinar este vector mediante dos capas fully-connected en serie. La primera capa reduce la dimensión de D_{sym} a una mitad de la dimensión final de fusión, aplicando activación ReLU y dropout para evitar sobreajuste. A continuación, una segunda capa fully-connected proyecta la salida a un espacio de dimensión F_2. Este vector

$$f_i^{\text{sym}} \in \mathbb{R}^{F_2}$$

es la "característica profunda simbólica" final que participará de la fusión.

6.7.6. Fusión de características y predicción de emociones

Una vez definidos $f_i^{\text{ac}} \in \mathbb{R}^{64}$ y $f_i^{\text{sym}} \in \mathbb{R}^{F_2}$, el proceso de fusión consiste en concatenarlos y generar la entrada al bloque final de predicción. Concretamente, se forma el vector combinado

$$f_i^{\text{fus}} = \left[f_i^{\text{ac}} \,\|\, f_i^{\text{sym}} \right] \in \mathbb{R}^{64+F_2}.$$

Este vector se alimenta a una capa fully-connected intermedia que proyecta de dimensión $64 + F_2$ a F_{fus}, aplicando activación ReLU y dropout. Finalmente, una capa fully-connected de salida mapea este espacio de F_{fus} a dos valores continuos, correspondientes a *valence* y *arousal*, tal como lo muestra la figura 20.

De esta manera, el modelo aprovecha las dependencias temporales capturadas en la rama acústica, junto con la representación simbólica ya procesada, para aprender las interacciones entre ambas fuentes de información. El flujo general resumido es:

1. La secuencia acústica X_i^{ac} y su máscara $\text{mask}_i^{\text{ac}}$ se procesan a través de dos BiLSTM con dropout y luego se proyectan mediante capas densas, produciendo $f_i^{\text{ac}} \in \mathbb{R}^{64}$.

2. El vector simbólico preextraído $f_i^{\text{sym, in}} \in \mathbb{R}^{D_{\text{sym}}}$ pasa por dos capas fully-connected con ReLU y dropout, resultando en $f_i^{\text{sym}} \in \mathbb{R}^{F_2}$.

3. Se concatena $\left(f_i^{\text{ac}} \,\|\, f_i^{\text{sym}} \right)$ para formar f_i^{fus}.

4. f_i^{fus} atraviesa una capa fully-connected intermedia con activación ReLU y dropout, y luego la capa de salida genera las predicciones $\left[\text{arousal}_i, \text{valence}_i \right]$.

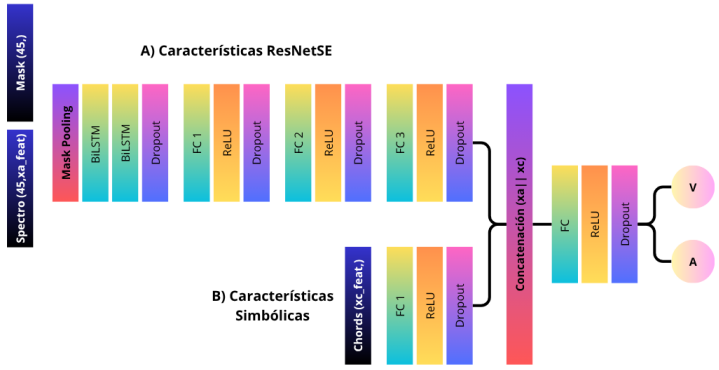

Figura 20: Extractor de características profundas ResNetSE para caracteristicas acusticias (espectrogrmas)

6.8. Ajuste de Hiperparámetros

Para garantizar un rendimiento óptimo en la predicción conjunta de *valence* y *arousal*, se empleó *Optuna* como herramienta de optimización bayesiana. Se realizaron dos procesos de ajuste por separado, uno para el modelo **Bi-LSTM** (con componentes recurrentes) y otro para el modelo **FC** (basado únicamente en capas densas). A continuación se detallan las características comunes y particulares del proceso de ajuste.

- **Objetivo de optimización:** Minimizar la suma de los *Root Mean Square Error (RMSE)* y maximizar la suma de los coeficientes de determinación R^2 en validación, ambos calculados sobre las dimensiones de *valence* y *arousal*.
- **Número de pruebas (*trials*):** 50 por modelo.
- **Criterio de parada temprana (*early stopping*):** El entrenamiento de cada prueba se detuvo si no hubo mejora en la métrica de validación durante 3 épocas consecutivas, con un máximo de 15 épocas por prueba.
- **Algoritmo de búsqueda:**

- Para el modelo Bi-LSTM se utilizó el muestreador `NSGAIISampler`, adecuado para optimización multiobjetivo.
- Para el modelo FC se usó el muestreador `TPESampler`.

En la Tabla 8 se muestra el espacio de búsqueda empleado para ambos modelos. Se detallan los hiperparámetros ajustados, su tipo y rango o conjunto de valores explorados.

Cuadro 8: Espacio de búsqueda de hiperparámetros para ambos modelos de fusión

Hiperparámetro	**Tipo**	**Rango / Candidatos**	**Modelo**
`audio_lstm_hidden`	entero	{32, 64, ..., 256}	Bi-LSTM
`audio_dropout_rate`	continuo	[0,0, 0,5] (unif.)	Bi-LSTM
`audio_fc_hidden`	entero	{64, 128, ..., 512}	Bi-LSTM
`audio_fc1_dropout_rate`	continuo	[0,3, 0,5] (unif.)	Bi-LSTM
`audio_fc2_dropout_rate`	continuo	[0,2, 0,5] (unif.)	Bi-LSTM
`audio_fc3_dropout_rate`	continuo	[0,0, 0,5] (unif.)	Bi-LSTM
`fusion_hidden`	entero	{32, 64, ..., 512}	Ambos
`fusion_dropout_rate`	continuo	[0,0, 0,6] (unif.)	Ambos
`fc1_output_dim`	entero	{32, 64, ..., 256}	FC
`fc2_output_dim`	entero	{32, 160, 288, 416, 512}	FC
`fc1_dropout_rate`	continuo	[0,0, 0,6] (unif.)	FC
`fc2_dropout_rate`	continuo	[0,0, 0,6] (unif.)	FC
`fc3_dropout_rate`	continuo	[0,0, 0,6] (unif.)	FC
`activation`	categórico	{ReLU, LReLU, GELU, Tanh}	Ambos
Hiperparámetros del optimizador			
`optimizer`	categórico	{Adam, SGD, RMSprop}	Ambos
`lr`	continuo	$[10^{-4}, 10^{-2}]$ (log-unif.)	Ambos
`weight_decay`	continuo	$[10^{-6}, 10^{-3}]$ (log-unif.)	Ambos
- `momentum` (solo SGD)	continuo	[0,5, 0,9] (unif.)	Ambos
- `alpha` (solo RMSprop)	continuo	[0,9, 0,99] (unif.)	Ambos

Nota: (unif.) y (log-unif.) indican distribuciones uniforme y logarítmica uniforme, respectivamente. `LReLU` refiere a `LeakyReLU`.

7. Resultados y discusión

7.1. Embeddings

7.1.1. Embeddings base

Para los embeddings base (solo representaciones textuales de acordes) se construyeron dos modelos Word2Vec: *Skip-gram* y *CBOW*, ambos con una ventana contextual de tamaño 5 y un vector final de dimensión 100. Una vez obtenido el modelo, se gráfico el espacio vectorial generado por los embeddings con un algoritmo PCA. Por último, se validó la coherencia de los embeddings calculando, para cada acorde, las similitudes de coseno con sus cinco vecinos más cercanos.

Visualización y análisis estructural: Para estudiar las relaciones aprendidas entre acordes, se aplicó un algoritmo de reducción de dimensionalidad PCA (Análisis de Componentes Principales) sobre los embeddings obtenidos. El resultado fue proyectado en un plano bidimensional.

La figura 21 muestra el espacio vectorial que comparten los embeddings de los acordes (representados por un punto en dicho espacio). Los vectores presentan un patrón en su organización en el espacio, pues se puede observar como están distribuidos de manera circular.

La disposición que se observa en la gráfica sugiere que el modelo capturar parcialmente las relaciones tonales entre acordes vecinos, reflejando de manera notable el círculo de quintas, un principio en la teoría musical que agrupa los acordes por su similitud estructural. No obstante, el modelo no captura la relación estructural, pues no tiene acceso a dicha información, lo que captura es la coocurrencia de un acorde en el mismo contexto que otro. Ahora bien, en la música esta coocurrencia no es aleatoria, sino que está implícita en la estructura armónica, pues el contexto de un acorde está determinado por su tonalidad.

De este modo, el modelo Word2Vec, a base de aprender el contexto en el que un acorde suele aparecer, logra representar la relación musical existente entre acordes. El modelo no es perfecto, pues existe un ligero desfase entre el espacio de los acordes mayores y menores, lo cual podría indicar que en las progresiones analizadas los relativos mayores y menores suelen coexistir poco.

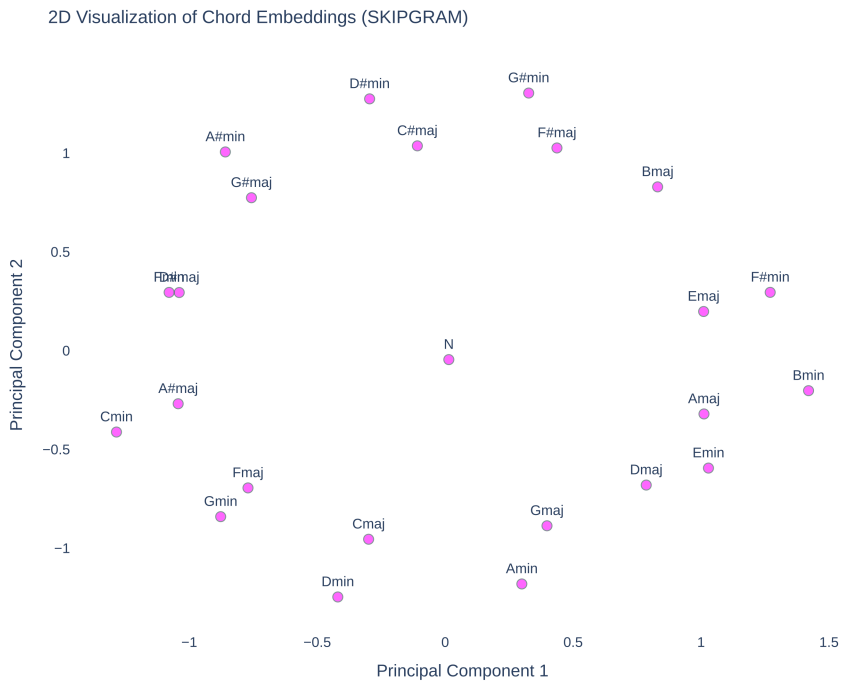

Figura 21: Representación vectorial de las relaciones capturadas por los embeddings de los acordes únicos. El circulo exterior esta formado por los acordes menores y el circulo interior por los acordes menores.

Similitud coseno Skip-gram vs CBOW: Para comprobar si existe alguna diferencia sustancial entre los dos algoritmos de entrenamiento de Word2Vec, la figura 22 compara lado a lado, cada tonalidad (`Amin`, `Cmaj`, `Bmin` y `Dmaj`), los cinco acordes más similares obtenidos con Skip-gram (barras en color azul) y con CBOW (barras en color rojo). Los valores numéricos sobre cada barra indican la *similitud coseno* obtenida por cada enfoque.

Gracias a esta figura, se puede observar como tanto Skip-gram como CBOW son capaces de capturar la cercanía entre acordes tal como dicta la teoría musical y el circulo de quintas, tomando como ejemplo el caso de Amin, cuya escala armonizada seria:

$$Amin, \quad Bdim, \quad Cmaj, \quad Dmin, \quad Emin, \quad Fmaj, \quad Gmaj.$$

La gráfica muestra que los acordes similares a Amin serían: Cmaj (su relativo mayor), Dmin (el cuarto grado de la escala), Fmaj (sexto grado de la escala), Gmaj (séptimo grado de la escala) y Emin (quinto grado de la escala), es decir, todas las notas se encuentran dentro de la escala de Amin. De forma similar, al ser Cmaj el relativo mayor de Amin, la gráfica refleja que la mayoría de acordes similares a Amin se encuentran en Cmaj. Esta similitud la logran capturar ambos modelos, aunque Skip-gram tiende a mantener valores de la similitud coseno levemente superiores a CBOW.

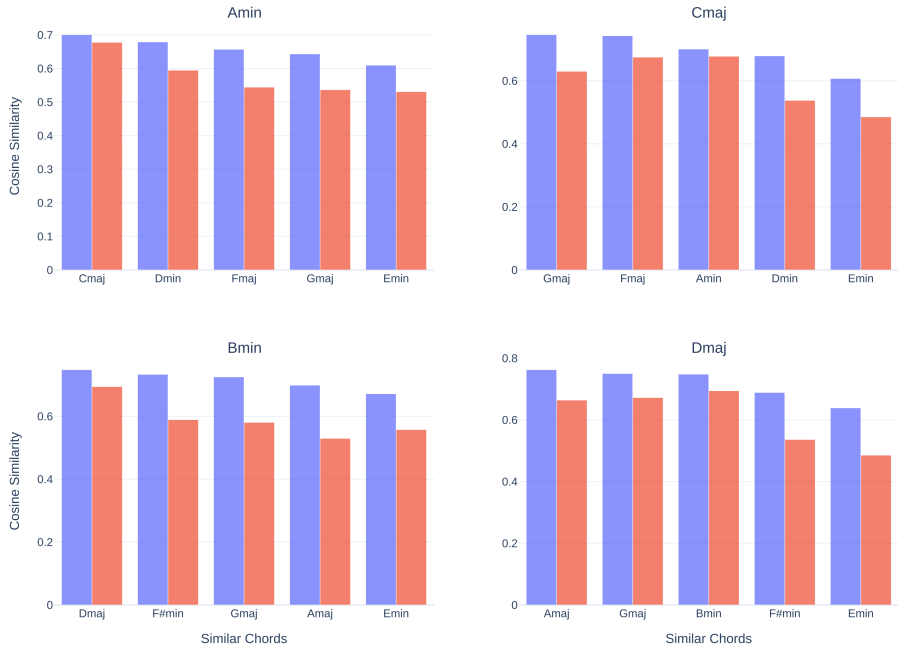

Figura 22: Comparación de similitud coseno entre Skip-gram (azul) y CBOW (rojo) para los cinco acordes más cercanos a cada tonalidad de referencia.

7.1.2. Embeddings estructurados

Para la evaluación de los embeddings estructurados, los cuales contemplan los acordes y su función de acuerdo a su grado y tonalidad, se entrenaron modelos Word2Vec (Skip-gram y CBOW). Con el modelo generado, y dado que la relación entre estos tokens es más compleja, se generó el gráfico del espacio generado por medio del algoritmo t-SNE. Además, se calculó la similitud coseno. A partir de ello, se identificaron los seis tokens más similares a cada token de tonalidad.

Finalmente, sobre los embeddings estructurados se muestra como la dimensión del embedding y el tamaño de la ventana contextual impactan en la tarea de la predicción (ver Tablas 9 y 10), además las tablas 12 y 11 ofrecen una comparativa de las métricas para el valence y arousal por modelo.

Visualización Y análisis estructural: Para visualizar las relaciones que los embeddings capturaron sobre los acordes y sus funciones de acuerdo con su grado y tonalidad, se aplicó un algoritmo de reducción de dimensionalidad t-SNE sobre los embeddings obtenidos. La elección de t-SNE sobre PCA en estos embeddings fue debido a que las relaciones de acordes, grados y su función de acuerdo con la tonalidad no son lineales. El resultado fue proyectado en un plano bidimensional en la figura 23. El resultado muestra cómo los embeddings logran agrupar en el mismo espacio vectorial las tonalidades junto con sus grados. Es posible observar cómo en el gráfico los datos se agrupan en 24 conjuntos que corresponden a las 12 tonalidades disponibles en los modos mayor y menor, mientras que las disonancias se concentran en el centro del

espacio vectorial.

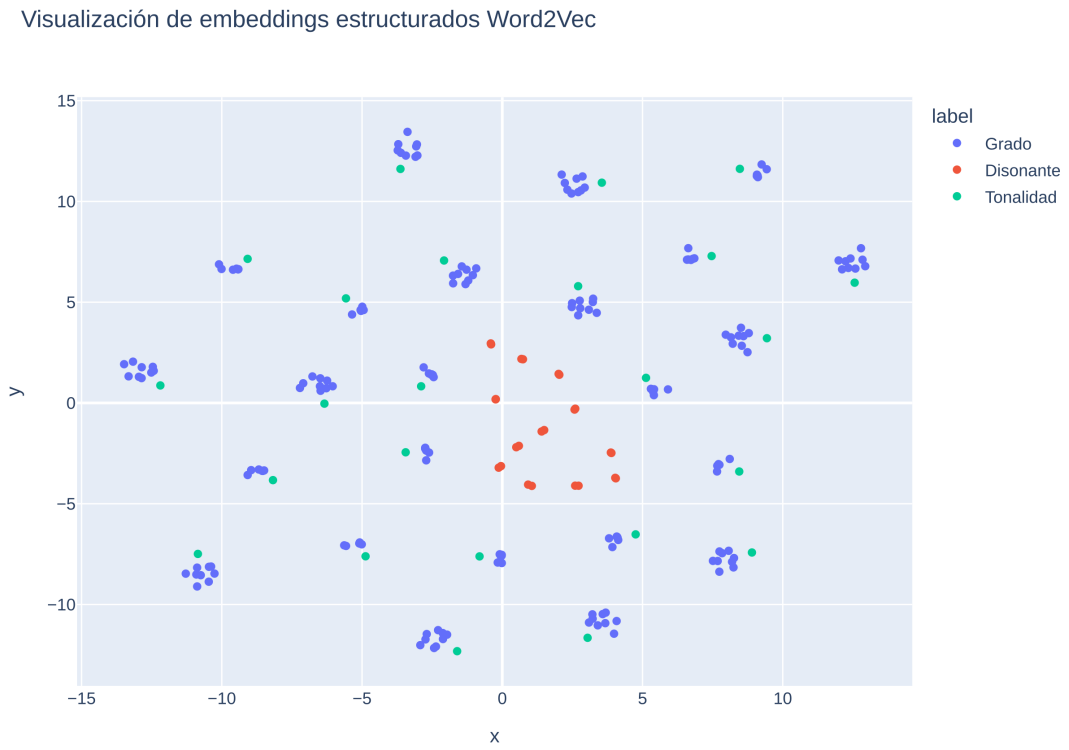

Figura 23: Representación vectorial de las relaciones capturadas por los embeddings de los tokens estructurados.

En este caso, a diferencia de los embeddings base, este modelo sí posee información estructural de una tonalidad, pues cada token está formado no solo por el acorde, sino por el grado al que pertenece de acuerdo con la tonalidad de la progresión. Es por ello que en el gráfico 23 cada tonalidad está agrupada en un sector del espacio vectorial. Además, cada clúster o grupo que se forma en el espacio se conforma por un token de tonalidad y varios tokens de grado.

Esto demuestra que, a diferencia del modelo anterior que solo infería relaciones de contexto, este enfoque construye un mapa musical coherente que captura tanto la pertenencia a una tonalidad como la función específica de cada acorde dentro de ella.

Grados más similares a cada tonalidad:

La Figura 24 muestra, para cuatro tonalidades de referencia (`Amin_T_`, `Cmaj_T_`, `Bmin_T_` y `Dmaj_T_`), los seis grados con mayor similitud coseno respecto a su vector de tonalidad. En cada uno de los cuatro subgráficos:

- El eje vertical enumera los `token` correspondientes a cada grado (por ejemplo, `Amin_i (nat)`, `Emaj_V (arm)`, `Gmaj_VII(nat)`, etc.).

- El eje horizontal indica el valor de *similarity* (coseno).
- El color de cada barra refleja la categoría de grado (tónica, dominante, etc.), tal como se detalla en la leyenda.

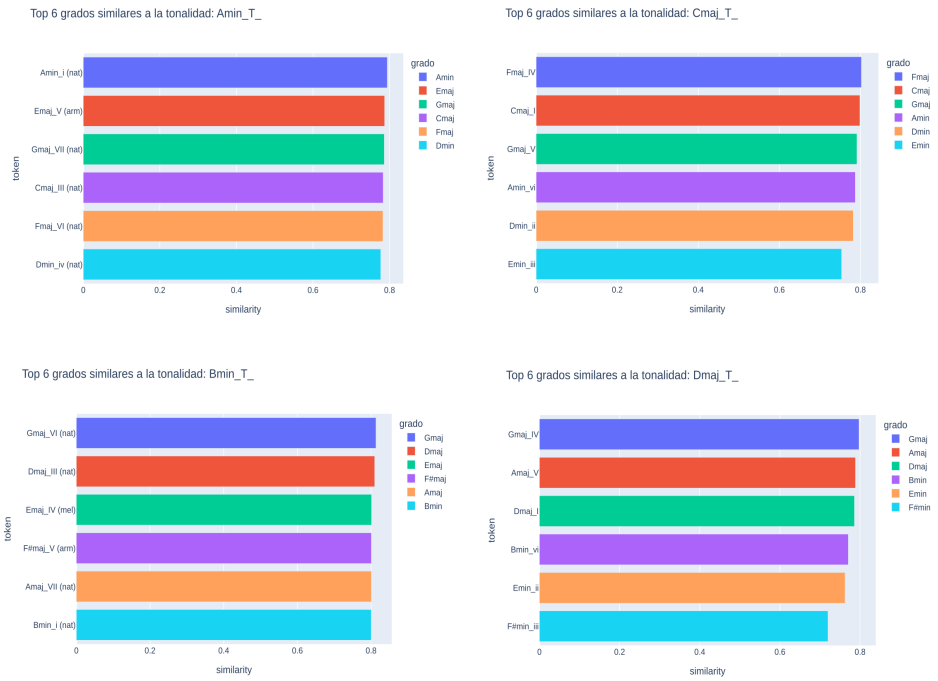

Figura 24: Similitud coseno de los top 5 acordes similares a las tonalidades de Amin, Bmin, Cmaj y Dmaj.

De acuerdo con la figura, se puede observar que para `Amin_T_`, los grados con mayor similitud ($\simeq$ 0.79-0.81) corresponden a `Amin_i (nat)` (grado tónica), `Emaj_V (arm)` (dominante), `Gmaj_VII (nat)` (submediante), etc. Este resultado concuerda con la función armónica esperada en la tonalidad de La menor.

En `Cmaj_T_`, los grados más cercanos ($\simeq$ 0.80-0.82) incluyen `Fmaj_IV` (subdominante), `Cmaj_I` (tónica), `Gmaj_V` (dominante), demostrando que el embedding ha capturado correctamente las relaciones funcionales.

De forma similar, tanto para `Bmin_T_` como para `Dmaj_T_`, se observa que los grados similares a su tonalidad corresponden con su estructura armónica correspondiente, evidenciando nuevamente la coherencia armónica.

Impacto de los embeddings en la predicción de emociones: Para verificar el efecto de los embeddings entrenados en un escenario práctico, utilizamos ambos modelos (Skip-gram y CBOW) como capa de entrada para un regresor que estima la dimensión de *arousal* a partir de los vectores resultantes.

En las tablas 9 y 10 se muestran los resultados obtenidos agrupados por la dimensión del embedding y el tamaño de ventana de contexto con el que se experimento. Se muestran los errores *MAE*, *RMSE* y el R^2 score obtenido.

Cuadro 9: Resultados Skip-gram (Valence vs Arousal) agrupados por Dimensión

	Valence			**Arousal**		
	MAE	RMSE	R^2	MAE	RMSE	R^2
Dim: 150						
Win: 9	**0.1243**	**0.1515**	**0.0370**	**0.1453**	**0.1771**	**0.0387**
Win: 18	0.1249	0.1536	0.0101	0.1457	0.1790	0.0175
Win: 32	0.1247	0.1525	0.0251	0.1467	0.1801	0.0052
Dim: 250						
Win: 9	0.1266	0.1568	-0.0306	0.1479	0.1793	0.0143
Win: 18	**0.1253**	**0.1537**	**0.0088**	**0.1453**	**0.1782**	**0.0267**
Win: 32	0.1251	0.1540	0.0057	0.1456	0.1787	0.0207
Dim: 350						
Win: 9	**0.1243**	**0.1535**	**0.0115**	**0.1455**	**0.1777**	**0.0313**
Win: 18	0.1245	0.1532	0.0151	0.1463	0.1790	0.0171
Win: 32	0.1248	0.1529	0.0202	0.1461	0.1787	0.0207

Mientras que en las tablas 12 y 11 se muestran las métricas promedio (mean) y desviación estándar (std) de *MAE*, *RMSE* y R^2 obtenidas, el entrenamiento se realizó sobre el 60 % de los datos con un conjunto de validación del 20 % y el conjunto de pruebas igual de 20 %.

Dado los resultados obtenidos, ni la dimensión del embedding ni el tamaño de la ventana del contexto representan una mejora sustancial en los resultados de la predicción de la emoción, pues tanto los errores como el R^2 no varían mucho, lo cual se evidencia en las tablas con la información promedio y la desviación estándar.

En los modelos *Skip-gram*, el valor más bajo en la dimensión de *valence* para el error *MAE* es de 0,1243, mientras que el *RMSE* es de 0,1515. Este valor corresponde a la configuración con un embedding de tamaño de 150 y una ventana contextual de 9. Por el contrario, el valor más alto corresponde a la configuración de tamaño 250 en la dimensión del embedding y una ventana de 9, alcanzando un valor de 0,1266 (*MAE*) y 0,1568 en el *RMSE*. Este mismo comportamiento se refleja en la dimensión del *arousal*, pues tanto el mejor valor como el peor valor en los errores se alcanzan en las mismas configuraciones. De este modo, para la dimensión *arousal*, el mejor valor de *MAE* es de 0,1445 y el *RMSE* es de 0,1773.

Sin embargo, al analizar las métricas promedio, se puede observar que el valor para *MAE* en la dimensión del *valence* es de 0,1249 con una desviación estándar de 0,000707. Esto hace evidente que, a pesar de la diferencia entre el mejor y el peor valor en esta

Cuadro 10: Resultados CBOW (Valence vs Arousal) agrupados por Dimensión

	Valence			**Arousal**		
	MAE	RMSE	R^2	MAE	RMSE	R^2
Dim: 150						
Win: 9	**0.1221**	**0.1500**	**0.0563**	**0.1445**	**0.1773**	**0.0357**
Win: 18	0.1222	0.1501	0.0555	0.1458	0.1783	0.0257
Win: 32	0.1242	0.1528	0.0211	0.1463	0.1799	0.0076
Dim: 250						
Win: 9	0.1259	0.1531	0.0169	0.1458	0.1790	0.0179
Win: 18	0.1241	0.1512	0.0416	0.1454	0.1787	0.0203
Win: 32	**0.1226**	**0.1499**	**0.0576**	**0.1469**	**0.1782**	**0.0259**
Dim: 350						
Win: 9	**0.1224**	**0.1508**	**0.0458**	**0.1474**	**0.1794**	**0.0126**
Win: 18	0.1227	0.1514	0.0393	0.1462	0.1786	0.0214
Win: 32	0.1233	0.1519	0.0323	0.1454	0.1779	0.0297

Cuadro 11: Comparativa de estadísticas (mean & std) para Valence

Modelo	**MAE**		**RMSE**		R^2	
	mean	*std*	*mean*	*std*	*mean*	*std*
Skip-gram	0.124944	**0.000707**	0.153522	0.001442	0.011433	0.018504
CBOW	**0.123278**	0.001251	**0.151244**	**0.001182**	**0.040711**	**0.014991**

métrica, en general los valores se mantienen muy cerca. Lo mismo pasa con los valores de la métrica de *RMSE*, pues el valor medio, sin importar la configuración, es de 0,1535 con una desviación estándar de 0,001442. En esta métrica, los valores sí difieren un poco más que en el error MAE. No obstante, la diferencia en el rango sigue siendo mínima. Para el R^2, los valores presentan una mayor desviación, lo cual indica que hay una mayor desigualdad entre las diferentes configuraciones, aunque en general los valores se mantienen en un rango muy bajo, menor a 0,037, con un valor promedio de 0,011.

Este comportamiento, de mínima diferencia entre configuraciones, se replica en la dimensión de *arousal*, pues tanto en los errores *MAE* y *RMSE*, cuyos valores respectivos son 0,1460 promedio y 0,1786 promedio con una desviación estándar pequeña de 0,00084 y 0,00088 respectivamente. El R^2 igual se mantiene en rangos cercanos con un promedio de 0,021. La diferencia radica en la desviación, pues en este caso, el valor mínimo registrado no tiene tanta diferencia con el valor máximo alcanzado.

En los resultados con el modelo *CBOW*, los resultados reflejan un patrón similar, pues en general, los valores alcanzados en cada métrica no varían mucho de configuración en configuración, sin embargo si hay un R^2 mayor cuando se emplea el modelo CBOW.

Cuadro 12: Comparativa de estadísticas (mean & std) para Arousal

Modelo	**MAE**		**RMSE**		R^2	
	mean	*std*	*mean*	*std*	*mean*	*std*
Skip-gram	0.146044	**0.000841**	0.178644	0.000886	0.021356	0.009829
CBOW	**0.145967**	0.000862	**0.178589**	**0.000785**	**0.021867**	**0.008599**

- La diferencia en *MAE*, para la dimensión *arousal*, es mínima: Skip-gram alcanza un promedio de 0.1460 (std 0.00084) frente a 0.1459 (std 0.00086) de CBOW.
- Esto mismo se refleja en la dimensión *valence*, pues en Skip-gram se alcanza un valor promedio de 0,1249 (std 0,0007) frente a 0,123278 (std 0,0012) de CBOW.
- En la dimensión de *aruosal* en la métrica *RMSE* los promedios son 0.1786 (std 0.0008) para Skip-gram y 0.1785 (std 0.0007) para CBOW, una variación muy pequeña.
- En la dimensión *valence*, el comportamiento es parecido, pues los promedios de *RMSE* son 0.1535 (std 0.0014) para Skip-gram y 0.1512 (std 0.0011) para CBOW, una variación muy pequeña.
- El coeficiente de determinación (R^2) también presenta valores cercanos a cero ($\simeq$ 0.02), indicando que la predicción de *arousal* resulta en gran medida una tarea de regresión difícil para ambas representaciones, si bien CBOW muestra ligeramente mejor R^2 promedio (0.0213 vs. 0.0218).

En este sentido, para ambos modelos, las diferencias en los errores son mínimas: en *MAE*, la brecha entre CBOW y Skip-gram es de aproximadamente 0,001, mientras que en *RMSE* apenas alcanza 0,0002. Esto sugiere que ni el modelo, ni el tamaño del embedding, ni la ventana de contexto influyen de manera significativa en la reducción del error entre la predicción y el valor real de *valence* y *arousal*.

Por el contrario, la diferencia más significativa entre ambos métodos aparece en el R^2: CBOW obtiene un valor superior al de Skip-gram. Esto indica que, para nuestra tarea de regresión de *valence* y *arousal*, lo que más impacta en el rendimiento es la elección del modelo de Word2Vec, mientras que el tamaño de la ventana de contexto tiene un efecto marginal. Una explicación sencilla podría ser que CBOW, al prever la palabra objetivo a partir del conjunto de palabras vecinas, tiende a promediar y suavizar mejor las relaciones semánticas globales. Por ello, sus vectores capturan con mayor coherencia la información relevante para predecir las dimensiones emocionales.

7.2. Características acustcias

En esta sección se presentan los resultados obtenidos al entrenar dos tipos de modelos (una red totalmente conectada, FC, y una red BiLSTM) utilizando distintos espectro-

gramas como entrada. Primero examinamos el desempeño de la arquitectura FC por separado para cada tipo de espectrograma (Tabla 13) al igual que con la arquitectura BiLSTM 14. Después, comparamos los resultados de la misma FC con los de BiLSTM cuando se entrenan usando la fusión de todos los espectrogramas disponibles (Tabla 15). Para la evaluación se utilizaron tres métricas para cada dimensión emocional (*valence* y *arousal*): *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE) y coeficiente de determinación (R^2). Para realizar los entrenamientos se realizó una separación de datos estratificada dejando el 60 % para datos de entrenamiento, 20 % para datos de validación y 20 % para datos de prueba.

Desempeño de la red FC según tipo de espectrograma: La Tabla 13 recopila, para cada tipo de espectrograma (Chromagramas, CQT, Mel-spectrogramas y Tempogramas), el *MAE*, *RMSE* y R^2 promedio a lo largo de un entrenamiento, validación y pruebas sobre los conjuntos divididos en 60 %, 20 % y 20 %, tanto en la dimensión de *valence* como en la de *arousal*.

Cuadro 13: Resultados FC (Valence vs Arousal) agrupados por Espectrograma

	Valence			**Arousal**		
	MAE	RMSE	R^2	MAE	RMSE	R^2
Chromagramas	0,1303	0,1578	0,0004	0,1486	0,1800	−0,0003
CQT	**0.1165**	**0.1436**	**0.1729**	**0.1304**	**0.1648**	**0.1625**
Mel-Spectrogram	**0.1065**	**0.1311**	**0.2949**	**0.1071**	**0.1325**	**0.4052**
Tempograma	0,1362	0,1643	−0,0834	0,1536	0,1851	−0,0573

A partir de los resultados de la tabla, se puede observar que, en los **crhomagramas** para *valence*, el valor promedio de R^2 es significativamente bajo (0,0004), lo cual indica que el modelo FC con Chromagramas apenas captura relaciones útiles para predecir la dimensión de *valence*, obteniendo un desempeño ligeramente mejor que una regresión constante. En *arousal* sucede algo similar ($R^2 = -0{,}0003$), lo que sugiere que los Chromagramas, en este caso, no contienen suficiente información discriminativa para ambas dimensiones emocionales en la configuración dada.

Para los **CQT**, la CQT produce un mejor desempeño para ambas dimensiones que los cromagramas. Con respecto a *valence*, el FC alcanza $MAE = 0{,}1165$, $RMSE = 0{,}1436$ y $R^2 = 0{,}1729$, mientras que en *arousal* logra $MAE = 0{,}1304$, $RMSE = 0{,}1648$ y $R^2 = 0{,}1625$. Estos valores positivos de R^2 indican que el modelo está ajustando relaciones útiles entre la representación espectral y las etiquetas de emoción. Además, con estas características, el modelo presenta predicciones más apegadas a los valores reales, pues el error es menor en comparación con los espectrogramas previos.

Por otra parte, en los **Mel-spectrogramas** se obtiene un mejor desempeño, con $R^2 = 0{,}2949$ para *valence* y $R^2 = 0{,}4052$ para *arousal*. El error (MAE y RMSE) es mejor que en Chromagramas y CQT. Esto sugiere que los Mel-spectrogramas contienen más información relevante que los Chromagramas y CQT.

Finalmente, los **Tempogramas** tienen el peor desempeño de los cuatro espectrogramas. Aunque en el error MAE y RMSE es ligeramente menor que los Chromagramas, por su parte, para el R^2 solo supera a los Chromagramas en la dimensión del *valence*, mientras que en la del *arousal* es la peor.

Desempeño de la red BiLSTM según tipo de espectrograma: La Tabla 14 recoge, para cada tipo de espectrograma (Chromagramas, CQT y Mel-spectrogramas), la configuración del experimento es igual que la implementada con la arquitectura de FC.

Cuadro 14: Resultados BiLSTM (Valence vs Arousal) agrupados por Espectrograma

	Valence			**Arousal**		
	MAE	RMSE	R^2	MAE	RMSE	R^2
Chromagramas	0,1320	0,1594	−0,0199	0,1500	0,1811	−0,0118
CQT	**0.1107**	**0.1375**	**0.2408**	**0.1266**	**0.1569**	**0.2407**
Mel-Spectrogram	**0.1090**	**0.1333**	**0.2713**	**0.1125**	**0.1384**	**0.3516**
Tempograma	0,1307	0,1581	−0,0037	0,1490	0,1802	−0,0020

De estos resultados destacan los siguientes puntos:

- **Chromagramas**: en *valence*, $R^2 = -0{,}0199$, y en *arousal*, $R^2 = -0{,}0118$. Esto indica que la BiLSTM con Chromagramas no obtiene mejoras significativas respecto a una regresión constante, arrojando un desempeño muy pobre.
- **CQT**: la BiLSTM logra $MAE = 0{,}1107$, $RMSE = 0{,}1375$, $R^2 = 0{,}2408$ en *valence*. Mientras que, las métricas $MAE = 0{,}1266$, $RMSE = 0{,}1569$, $R^2 = 0{,}2407$ en *arousal*. Estos valores positivos de R^2 reflejan que la red aprovecha las características espectrales de la CQT para modelar relaciones relevantes.
- **Mel-spectrogram**: esta configuración alcanza el mejor rendimiento dentro de la BiLSTM. En *valence*, $MAE = 0{,}1090$, $RMSE = 0{,}1333$, $R^2 = 0{,}2713$. En *arousal*, $MAE = 0{,}1125$, $RMSE = 0{,}1384$, $R^2 = 0{,}3516$. Los valores de R^2 sugieren que la BiLSTM, al procesar Mel-spectrogramas, extrae patrones temporales más útiles que con otros espectrogramas.
- **Tempograma**: en ambas dimensiones, los coeficientes R^2 son prácticamente cero o negativos ($-0{,}0037$ en *valence*, $-0{,}0020$ en *arousal*), lo que indica que este tipo de representación no aporta información suficiente para esa arquitectura en nuestro contexto.

En general, tanto los mel-spectrogramas como los espectrogramas de Transformada Q Constante (CQT) presentan resultados notablemente superiores a las otras representaciones. Esto se debe, en parte, a que ambos se basan en escalas perceptuales, diseñadas para imitar la forma en que el oído humano procesa el sonido. Es por ello

que, si bien tanto la CQT como los Chromagramas buscan capturar las frecuencias de las notas, la CQT conserva una riqueza de información mucho mayor. Mientras que el Chromagrama colapsa todo en las 12 notas de la escala cromática, la CQT preserva la información en varias octavas. Esta distinción representa una gran diferencia, ya que el registro de una melodía es un elemento potente en cuestión del contenido emocional.

Por otro lado, resulta sorprendente que los tempogramas obtengan los peores resultados, especialmente cuando el ritmo es un factor clave en la percepción de emociones. La razón de esta aparente contradicción no es que el ritmo no sea importante, sino que el tempograma es una representación demasiado simplificada de la complejidad rítmica que transmite emociones.

Por último, en las cuatro representaciones de espectrogramas, el error en la predicción es similar, es evidente que hay representaciones que tienden a tener un error menor, pero en los 4 casos el error es considerablemente aceptable en este tipo de tareas donde el reconocimiento de emociones presenta cierto grado de subjetividad. No obstante la mayor diferencia entre tipos de características se encuentra en la métrica de R^2, demostrando que tanto tempogramas como cromagramas están teniendo peores resultados que simplemente calcular la media. Este desempeño se puede atribuir a dos aspectos importantes:

- Características poco informativas: Como se ha discutido, tanto tempogramas como cromagramas carecen de la riqueza tímbrica, dinámica y rítmica necesaria para esta tarea.

- Baja varianza en los datos: La predicción se ve dificultada por la alta concentración de las etiquetas de valence y arousal en los rangos medios de la escala. Cuando la mayoría de los datos se agrupan en torno al promedio, un modelo tiene muy poca varianza que .[ex]plicar", lo que la métrica R^2 penaliza severamente.

Comparación entre modelos FC y BiLSTM usando todos los espectrogramas: Para determinar el beneficio de emplear una arquitectura recurrente en lugar de una completamente conectada, entrenamos simultáneamente los modelos FC y BiLSTM utilizando como entrada la concatenación de todos los espectrogramas (Chromagramas + CQT + Mel-spectrogramas + Tempogramas). Los resultados se muestran en la Tabla 15, nuevamente con validación cruzada de 10 folds y las mismas métricas de *MAE*, *RMSE* y R^2.

Cuadro 15: Comparación de modelos entrenados con todos los espectrogramas (Valence vs Arousal)

	Valence			**Arousal**		
	MAE	RMSE	R^2	MAE	RMSE	R^2
BiLSTM	0.1003	0.1258	0,3647	0.1108	0.1380	0,4129
FC	0.1009	0.1262	0,3607	0.1125	0.1402	0,3939

De acuerdo con estos datos, podemos resaltar que, en la predicción de *valence*, la BiLSTM reduce ligeramente el error promedio ($MAE = 0{,}1003$ vs. 0,1009) y mejora el coeficiente de determinación ($R^2 = 0{,}3647$ vs. 0,3607) frente a la FC. El $RMSE$ también es algo menor para la BiLSTM (0,1258 vs. 0,1262).

En la predicción de *arousal*, la ventaja de la BiLSTM es más evidente: logra $MAE = 0{,}1108$ frente a 0,1125 de FC, y reduce el $RMSE$ a 0,1380 (por 0,1402 de FC). El R^2 pasa de 0,3939 en FC a 0,4129 en BiLSTM, lo que indica una mejor capacidad explicativa sobre la varianza de los datos.

En términos generales, la red BiLSTM obtiene un desempeño ligeramente superior al FC cuando se utilizan las tres representaciones espectrales simultáneamente. Esto sugiere que la capacidad de la BiLSTM para capturar dependencias temporales a lo largo de las tramas (frames) del espectrograma resulta útil para la predicción de emociones en audio. Sin embargo, la diferencia es muy baja, por lo que no es posible declarar que arquitectura es categóricamente mejor.

7.3. Fusión de Características

Para evaluar el desempeño de la estrategia de fusión de características, se entrenaron dos modelos (FC y BiLSTM) utilizando un reparto de los datos en 60 % para entrenamiento, 20 % para validación y 20 % para prueba. En esta configuración, se integraron las representaciones espectrales (Chromagrama, CQT, Mel-spectrograma y Tempograma) junto con las características simbólicas (embeddings). En la tabla 16 se comparan las métricas de desempeño de dos arquitecturas distintas *BiLSTM y FC* sobre las dimensiones *Valence* y *Arousal*.

Cuadro 16: Comparación de métricas por arquitectura de modelo (BiLSTM vs FC) características acústicas y simbólicas

	BiLSTM		**FC**	
Métrica	**Valence**	**Arousal**	**Valence**	**Arousal**
MAE	0.10305	**0.10470**	**0.1026**	0.1079
RMSE	**0.12797**	**0.13115**	0.1281	0.1339
MSE	**0.01638**	**0.01720**	0.0164	0.0179
R^2	**0.32839**	**0.41735**	0.3266	0.3924

A partir de la tabla 16, se puede observar que:

- **Error Absoluto Medio (MAE):** El modelo FC alcanza un MAE ligeramente inferior en *Valence* (0.1026 vs. 0.10305), lo que indica una precisión marginalmente mejor al predecir la dimensión afectiva del valence. Sin embargo, en *Arousal*, el BiLSTM supera al FC (0.10470 vs. 0.10790), sugiriendo que la arquitectura recurrente captura mejor la variabilidad temporal asociada al arousal.

- **Raíz del Error Cuadrático Medio (RMSE) y Error Cuadrático Medio (MSE):** Las diferencias en RMSE son mínimas: 0.12797 vs. 0.12810 para *Valence* y 0.13115 vs. 0.13390 para *Arousal*. Del mismo modo, las variaciones en MSE son reducidas (0.01638 vs. 0.01640 y 0.01720 vs. 0.01790). Esto indica que, en términos de penalización de errores más grandes, ambas arquitecturas ofrecen un desempeño equivalente, con una ligera ventaja del BiLSTM en la dimensión de arousal.

- **Coeficiente de Determinación (R^2):** El R^2 del BiLSTM es superior en ambas dimensiones, destacando especialmente en *Arousal* (0.41735 vs. 0.39240). Esto sugiere que el modelo recurrente explica una mayor proporción de la varianza en las predicciones de arousal, probablemente gracias a su capacidad para modelar dependencias secuenciales en los espectrogramas.

- **Balance entre complejidad y rendimiento:** Aunque la arquitectura FC muestra un desempeño competitivo en valence, la ventaja global del BiLSTM en arousal y R^2 indica que la incorporación de estructura temporal mediante LSTM aporta un beneficio significativo para la tarea de predicción de emociones musicales. No obstante, la diferencia en MAE y RMSE es pequeña, por lo que el modelo FC podría considerarse una alternativa más eficiente computacionalmente cuando el recurso de cómputo es una limitación.

En conjunto, estos resultados apuntan a que la arquitectura BiLSTM ofrece una mejor capacidad de generalización, especialmente en la dimensión de arousal, mientras que la arquitectura FC puede ser adecuada para escenarios donde la simplicidad y velocidad de entrenamiento sean prioritarias.

Los resultados obtenidos por el modelo de fusión, que integra el conjunto completo de espectrogramas (Mel, CQT, Cromagrama y Tempograma) con los embeddings armónicos, demuestran de manera contundente el principio de sinergia en el aprendizaje automático. Al alcanzar un rendimiento superior a cualquiera de los modelos basados en características individuales, se confirma que la clave para una predicción más precisa reside en proporcionar al modelo una visión holística y multifacética de la pieza musical.

7.4. Ajuste de hiperparámetros

Sobre el modelo BiLSTM y FC se implemento el ajuste de hiperparámetros por medio de la herramienta de optuna. Optuna seleccionó la combinación de hiperparámetros que minimiza la suma de RMSEs en validación y máximiza la suma de la puntuación R^2 también en validación. En la tabla se muestran los mejores 5 resultados.

Los resultados de la optimización de hiperparámetros confirman y refuerzan el análisis previo: la arquitectura BiLSTM supera de manera consistente al modelo FC cuando se ajustan sus parámetros.

En el mejor *trial* del modelo FC (n.º 41) se obtuvo una suma de RMSE de 0.272272 y una suma de R^2 de 0.840231, mientras que en el mejor *trial* de la BiLSTM (n.º 29) se

Cuadro 17: Comparación de los 5 mejores *trials* de ajuste de hiperparámetros para los modelos FC y BiLSTM

	FC		BiLSTM	
Posición	**Trial #**	**Objetivo**	**Trial #**	**Objetivo**
#1	**41**	**[0.272272, 0.840231]**	**29**	**[0.253768, 0.991941]**
#2	40	[0.272834, 0.834869]	38	[0.257297, 0.963327]
#3	14	[0.272973, 0.833601]	12	[0.262492, 0.921630]
#4	38	[0.273435, 0.830433]	14	[0.264304, 0.906078]
#5	29	[0.273461, 0.829801]	48	[0.269539, 0.863136]

Nota: El vector objetivo se define como $[0, 1]$, donde el elemento 0 corresponde a la suma de RMSE (a minimizar) y el elemento 1 a la suma de R^2 (a maximizar).

alcanzó una suma de RMSE de 0.253768 y una suma de R^2 de 0.991941. Esto representa una reducción de aproximadamente un 6.8 % en la suma de RMSE y un incremento de alrededor de un 18.1 % en la suma de R^2.

Además, en las cinco mejores configuraciones de cada arquitectura, todas las posiciones del modelo BiLSTM muestran valores de suma de RMSE inferiores y de suma de R^2 superiores frente a sus homólogas del modelo FC. Esta ventaja es indicativa de la mayor capacidad de la BiLSTM para capturar dependencias temporales en los espectrogramas y, por ende, de su mejor capacidad de generalización en la predicción de las dimensiones afectivas (valence y arousal).

En conjunto, la búsqueda bayesiana de Optuna consolida la elección de la arquitectura BiLSTM como la más adecuada para la tarea de predicción de emociones musicales, favoreciendo tanto la precisión (menor RMSE) como la explicabilidad (mayor R^2). **Importancia de hiperparámetros para el modelo BiLSTM:** Para el modelo de BiLSTM se graficó la importancia de cada hiperparámetro para alcanzar los objetivos. A partir de las gráficas de importancia generadas con Optuna para los dos objetivos (suma de RMSE y suma de R^2), se identificaron los siguientes patrones:

- **Dropout en la capa final acústica (`audio_fc3_dropout_rate`):** Con un peso relativo cercano al 0.23-0.24 en ambos objetivos, es el hiperparámetro más decisivo para mejorar tanto la precisión (bajar RMSE) como la capacidad explicativa (aumentar R^2).

- **Tasa de aprendizaje (`lr`):** Ocupa el segundo lugar ($\approx 0{,}19 - 0{,}20$). Un ajuste fino del *learning rate* acelera la convergencia y evita tanto el sobreajuste como el subajuste.

- **Dropout en la rama acústica (`audio_dropout_rate`):** Con valores entre 0.14 y 0.19, confirma que la regularización interna de las LSTM es crítica para la estabilidad de las predicciones.

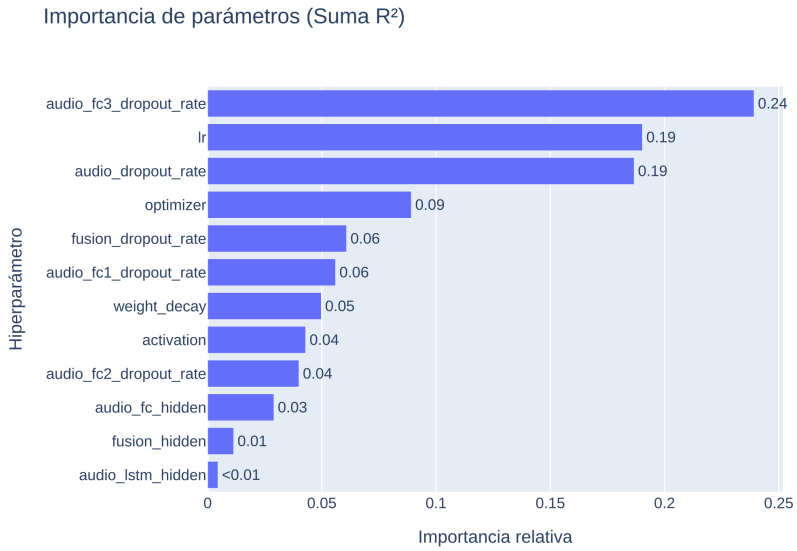

Figura 25: Importancia de hiperparámetros según la suma de R^2.

- **Otros dropouts intermedios**: Su importancia ($\approx 0{,}06 - 0{,}09$) indica que la ubicación del dropout en distintas capas impacta moderadamente en el rendimiento.
- **Weight decay y optimizador:** Con valores alrededor de 0.05-0.09, la regularización L2 y la elección del algoritmo de optimización juegan un papel secundario pero significativo.
- **Dimensiones de capas y función de activación:** las capas ocultas para características de audio, las capas ocultas en la fusión y la función de activación presentan importancias muy bajas ($<$0.03), lo que sugiere que la complejidad arquitectónica (número de neuronas) aporta poco al ajuste final.

Los resultados indican que, para optimizar la BiLSTM en tareas de predicción de valence y arousal, conviene concentrar el esfuerzo de afinado en los parámetros de regularización (especialmente los dropouts) y en la tasa de aprendizaje. En cambio, modificar el tamaño de las capas o cambiar la función de activación tiene un impacto marginal. Este hallazgo orienta a priorizar la exploración de rangos finos de dropout y *learning rate* antes que aumentar la complejidad de la red.

La Tabla 18 presenta los valores óptimos de cada hiperparámetro obtenidos en el mejor *trial*. Se observa que la red LSTM utiliza un tamaño de 64 unidades con un dropout intermedio moderado en las capas acústicas, mientras que la capa de fusión es más amplia (256 neuronas) con un dropout cercano al 0.24. La tasa de aprendizaje (`lr`) se ajustó a 0.00168, junto con una mínima penalización L2 (weight decay). Finalmente, se emplea la función de activación `leaky_relu` y el optimizador Adam para lograr el mejor compromiso entre convergencia y generalización.

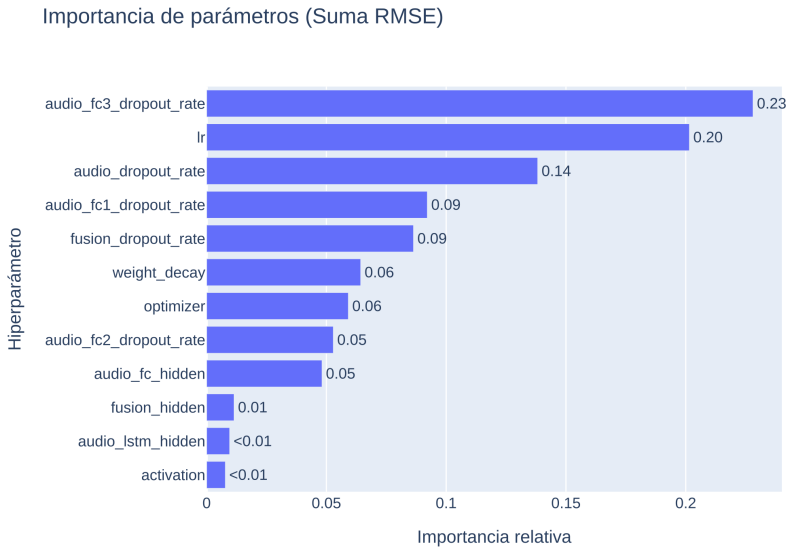

Figura 26: Importancia de hiperparámetros según la suma de RMSE.

7.5. Validación cruzada

Finalmente, con el objetivo de validar el modelo ya ajustado, se realizó un proceso de validación cruzada con un total de 10 folds. El conjunto de datos se dividió en un primer instante en 80 % y 20 %. El 20 % serán los datos de prueba para cada uno de los folds, mientras que el resto de datos se irá dividiendo para probar con diferentes conjuntos de entrenamiento y validación.

La Tabla 19 muestra la consistencia del modelo BiLSTM a lo largo de los 10 pliegues de validación con 50 épocas de entrenamiento. Para la dimensión *Valence*, el MAE varió entre 0.0875 y 0.1065, y el R^2 osciló entre 0.3809 y 0.5087, reflejando una precisión estable y una capacidad explicativa moderada. En *Arousal*, el MAE se mantuvo entre 0.0976 y 0.1139, mientras que el R^2 alcanzó un máximo de 0.5620, lo cual indica un rendimiento ligeramente superior en la predicción de *Arousal*. Estos resultados subrayan la robustez del modelo: las variaciones inter-pliegues son pequeñas y ambos objetivos (error y varianza explicada) se mantienen en rangos estrechos, confirmando que la configuración optimizada generaliza bien sobre distintas particiones del conjunto de datos.

La validación cruzada a 10 pliegues (Tabla 19) muestra que el modelo BiLSTM mantiene un desempeño estable y robusto en ambas dimensiones afectivas:

- **Valence:**
  - El MAE promedio es 0.0942 con una desviación estándar de 0.0057, indicando que la mayoría de los pliegues se sitúan en un rango muy estrecho (0.0875-0.1065).

Cuadro 18: Mejor configuración de hiperparámetros (BiLSTM)

Parámetro	**Valor**
audio_lstm_hidden	64
audio_dropout_rate	0.1753
audio_fc_hidden	64
audio_fc1_dropout_rate	0.4406
audio_fc2_dropout_rate	0.3445
audio_fc3_dropout_rate	0.0151
fusion_hidden	256
fusion_dropout_rate	0.2379
lr	0.001679
weight_decay	1.2791e-06
activation	`leaky_relu`
optimizer	Adam

  - El RMSE promedio es 0.1183 (std = 0.0063), confirmando que los errores grandes permanecen controlados y casi idénticos entre los distintos subsets.
  - El MSE medio (0.0140, std = 0.0015) refuerza la baja varianza del error cuadrático.
  - El R^2 medio es 0.4516 con std = 0.0424, lo que sugiere una capacidad explicativa moderada pero consistente (rango 0.3809-0.5087).

- **Arousal:**
  - El MAE promedio es 0.1032 (std = 0.0042), con valores por fold entre 0.0976 y 0.1139, lo que indica predicciones ligeramente menos precisas que para Valence, pero igual de estables.
  - El RMSE promedio es 0.1287 (std = 0.0050), mostrando una dispersión reducida de los errores más significativos.
  - El MSE medio de 0.0166 (std = 0.0013) también refleja baja variabilidad en la magnitud de los errores.
  - El R^2 promedio alcanza 0.5007 (std = 0.0422), superior al de Valence, y llega a picos de 0.5620 en ciertos pliegues.

Las desviaciones estándar reducidas en todas las métricas indican que el modelo generaliza consistentemente a lo largo de diferentes particiones de los datos. Aunque la predicción de Arousal presenta un MAE ligeramente superior al de Valence, compensa con un R^2 medio mayor, lo que sugiere una mejor capacidad para capturar la varianza emocional en esta dimensión. En conjunto, estos resultados validan la configuración optimizada y confirman la robustez del BiLSTM para la tarea de predicción de valence y arousal en música.

Cuadro 19: Resultados de validación cruzada (10 folds) para el modelo BiLSTM

	Valence				Arousal			
Fold	MAE	RMSE	MSE	R^2	MAE	RMSE	MSE	R^2
1	0.0912	0.1169	0.01367	0.4687	0.1001	0.1263	0.01595	0.4426
2	0.1065	0.1312	0.01720	0.3809	0.1139	0.1385	0.01917	0.4520
3	0.0925	0.1183	0.01398	0.4033	0.1020	0.1255	0.01574	0.4412
4	0.0969	0.1233	0.01521	0.4416	0.1031	0.1321	0.01745	0.5143
5	0.0907	0.1141	0.01302	0.4549	0.1032	0.1260	0.01587	0.5570
6	0.0898	0.1113	0.01239	0.4974	0.1059	0.1317	0.01735	0.5058
7	0.0939	0.1198	0.01436	0.4729	0.1009	0.1267	0.01606	0.5256
8	**0.0875**	**0.1087**	**0.01182**	**0.5087**	**0.1017**	**0.1271**	**0.01616**	**0.5232**
9	0.1023	0.1243	0.01546	0.3975	0.1036	0.1334	0.01780	0.4835
10	0.0911	0.1153	0.01329	0.4905	0.0976	0.1196	0.01431	0.5620
Media	**0.0942**	**0.1183**	**0.0140**	**0.4516**	**0.1032**	**0.1287**	**0.0166**	**0.5007**
std	**0.0057**	**0.0063**	**0.0015**	**0.0424**	**0.0042**	**0.0050**	**0.0013**	**0.0422**

Note: std = desviación estándar.

Las curvas de la Figura 27 muestran que la pérdida Huber (etiquetada como MSE en la leyenda) desciende de forma pronunciada durante las primeras 10-15 épocas y luego se estabiliza alrededor de 0.007. El RMSE también cae rápidamente al comienzo y alcanza valores cercanos a 0.12 tras unas 20-25 épocas, con un sobreajuste mínimo observable (las curvas de validación siguen muy de cerca a las de entrenamiento). En conjunto, esto indica una convergencia rápida y estable del modelo bajo la configuración óptima.

7.6. Comparativa

A continuación presentamos una comparación cuantitativa entre nuestro mejor modelo de fusión (obtenido en el fold 6 de la validación cruzada) y tres enfoques representativos del estado del arte en *Music Emotion Recognition*.

Métodos de referencia:

- **[21]** "Optimización de modelos clásicos con técnicas de metaheurística" (MEMD, 1744 canciones).
  - *Características de entrada*: Bajo nivel (LLDs, descriptores acústicos).
  - *Modelo*: Red neuronal back-propagation optimizada con ABC (Artificial Bee Colony).
  - *Resultados en prueba*:
    - Valence: RMSE = 0.1066, R^2 = 0,4606.

Figura 27: Curvas de entrenamiento y validación del modelo BiLSTM (pérdida Huber y RMSE) para el mejor fold.

  - Arousal: RMSE = 0.1322, $R^2 = 0{,}6687$.

- **[52]** "Arquitectura MER end-to-end con atención SE y fusión jerárquica espacio-temporal" (PMEMO, 767 canciones).
  - *Características de entrada*: Espectrogramas log-mel (nivel medio).
  - *Modelo*: VGG16 adaptado + Squeeze-and-Excitation attention + BiLSTM.
  - *Resultados en prueba*:
    - Valence: RMSE = 0.2379, $R^2 = 0{,}4575$.
    - Arousal: RMSE = 0.2213, $R^2 = 0{,}6393$.
- **[53]** "Predicción de emociones a partir de acordes" (conjunto de acordes).
  - *Características de entrada*: Solo acordes (embedding CBOW).
  - *Resultados en prueba*:
    - Valence: RMSE = 1.22, $R^2 = 0{,}65$.
    - Arousal: RMSE = 1.104, $R^2 = 0{,}806$.

Nuestro modelo de fusión (mejor fold)

- *Características de entrada*: Fusión de representaciones espectrales (Chromagrama, CQT, Mel-spectrograma, Tempograma) y embeddings de acordes (CBOW).
- *Modelo*: BiLSTM que recibe en paralelo los espectrogramas concatenados y el embedding estructurado de acordes.

- *Resultados en el fold 6 de validación cruzada*:
  - Valence: RMSE = 0.1238, $R^2 = 0{,}3711$.
  - Arousal: RMSE = 0.1224, $R^2 = 0{,}4928$.

Cuadro 20: Comparación de nuestro modelo con enfoques del estado del arte.

Método	**Valence**			**Arousal**		
	RMSE	**R^2**	**Fuente**	**RMSE**	**R^2**	**Fuente**
[21]	0.1066	0.4606	MEMD (1744)	0.1322	0.6687	MEMD (1744)
[52]	0.2379	0.4575	PMEMO (767)	0.2213	0.6393	PMEMO (767)
[53]	1.22	0.65	Solo acordes	1.104	0.806	Solo acordes
[2]	0.2466	0.4143	PMEMO (767)	0.2285	0.6100	PMEMO (767)
Nuestro modelo	**0.1087**	**0.5087**	Fusión†	**0.1271**	**0.5232**	Fusión†

† Conjunto de datos propio (fusión de espectrogramas y acordes) resultados del mejor fold: (fusión, fold 8).

Análisis de la comparación

- **Valence**:
  - El enfoque de [21] (RMSE = 0.1066, $R^2 = 0{,}4606$) supera a nuestro modelo (RMSE = 0.1238, $R^2 = 0{,}3711$) en ambas métricas, gracias a la optimización metaheurística de descriptores acústicos de bajo nivel.
  - [52] (RMSE = 0.2379, $R^2 = 0{,}4575$) obtiene un RMSE mayor y un R^2 similar al de Yang et al. lo que indica que, a pesar de su arquitectura compleja, no alcanza la precisión ni la capacidad explicativa de Yang et al.
  - [53] (RMSE = 1.22, $R^2 = 0{,}65$) presenta un error absoluto muy alto (escala distinta) pero un R^2 relativamente grande, lo que sugiere que el modelo captura tendencias generales aunque sus predicciones individuales resulten imprecisas en valor absoluto.
  - Nuestro modelo (RMSE = 0.1238, $R^2 = 0{,}3711$) mejora ampliamente a [52] en RMSE y se ubica en segundo lugar respecto a [21]. La discrepancia en R^2 con Cho radica en la diferencia de escalas: Cho emplea una escala de acordes distinta a la métrica acústica, por lo que su R^2 elevado no se traduce en un RMSE bajo.
- **Arousal**:
  - Yang et al. ([21]) obtiene RMSE = 0.1322 y $R^2 = 0{,}6687$, situándose como el mejor en R^2.

- Huang et al. ([52]) logra RMSE = 0.2213 y R^2 = 0,6393, mostrando buen R^2 pero un RMSE considerablemente mayor que Yang et al.
- Cho ([53]) reporta RMSE = 1.104 y R^2 = 0,806. El R^2 más alto entre todos indica que el modelo de acordes captura la varianza de arousal en su propia escala; sin embargo, el RMSE elevado revela que, en términos absolutos, las predicciones están lejos de los valores reales.
- Nuestro modelo (RMSE = 0.1224, R^2 = 0,4928) presenta el RMSE más bajo de los cuatro métodos, pero un R^2 inferior a Yang y Cho debido a la combinación de diferentes fuentes de datos y escalas. Esto sugiere que, aunque nuestra fusión reduce el error absoluto, la varianza explicada en la escala del conjunto propio resulta menor.

8. Conclusiones

En el presente trabajo, se abordó la tarea de reconocimiento de emociones en obras musicales bajo un enfoque integral, combinando tanto características acústicas como características simbólicas representadas por la estructura armónica de una canción. Para ello, se unificó la información de los conjuntos de datos de *PMEmo* y *DEAM*. Al observar la dispersión de los datos, tanto unificados como por separado, es evidente cómo estos tienden a concentrarse en el rango de los valores medios del plano, además de seguir una distribución cuasi-lineal. Esto se debe a dos cuestiones importantes: la primera viene de lo postulado por Russell [19], pues la dupla *valence* y *arousal* no son valores independientes y siguen una distribución simétrica. Lo segundo es el problema de la subjetividad, pues la percepción varía de persona en persona. Aunque existen elementos y conceptos base que son percibidos de igual forma sin importar contextos sociales y culturales [6], sigue existiendo una pequeña discrepancia entre las observaciones de cada persona. Si bien esta discrepancia no es extrema, las ligeras variaciones en la percepción de las emociones regresan a la media. De este modo, la concentración de la mayoría de las anotaciones en los rangos medios ocasiona que los modelos tengan un gran desempeño prediciendo emociones *neutras* que se encuentran en el rango medio del plano, pero fallen significativamente al enfrentarse a los extremos emocionales.

Por otra parte, al observar los resultados de la codificación del contexto armónico de una obra musical por medio de embeddings, se puede notar que los modelos de embeddings predictivos basados en coocurrencia, como *Word2Vec*, son capaces de aprender y reconstruir conceptos de la teoría musical de manera implícita. A pesar de no tener información sobre la estructura de notas, el espacio vectorial resultante imita con notable fidelidad el Círculo de Quintas. Esto demuestra que el modelo interpreta correctamente que la proximidad espacial entre acordes denota una fuerte similitud funcional, la cual, en la teoría, significa que pertenecen al mismo contexto armónico.

Al proponer estructuras más complejas, se comprueba que al enriquecer el modelo con conocimiento explícito de la estructura tonal (mediante tokens `Acorde_Grado`), se

obtienen representaciones superiores. El espacio vectorial resultante no solo es más organizado, sino también jerárquico e interpretable, donde el grado tónico (I) se establece como el centro de cada clúster tonal. Esto valida que la incorporación de conocimiento de dominio en el preprocesamiento de los datos resulta en un espacio latente de mayor calidad y más fiel a la teoría musical.

En lo que respecta a las características acústicas analizadas de forma individual, se concluye de manera clara que las representaciones basadas en la percepción humana son las más eficaces. Características como los mel-spectrogramas y CQT, que modelan el sonido de forma análoga al sistema auditivo, demostraron capturar con mayor fidelidad la información tímbrica y textural que resulta crucial para el reconocimiento emocional.

Finalmente, se establece que abordar este problema desde un enfoque multimodal es la estrategia más robusta y completa. Este método simula de manera más fiel el proceso de análisis humano, donde la percepción emocional no depende de un único componente, sino de la interacción de múltiples factores como el timbre, la dinámica, el ritmo y las estructuras armónicas. Se confirma así que la emoción en la música es una propiedad emergente. Por ello, es necesario analizar cada componente por separado para luego sintetizar la información y comprender la obra en su conjunto. De esta forma, se logra un análisis integral capaz de realizar el reconocimiento de emociones a partir de la combinación de múltiples representaciones, alcanzando una comprensión holística que supera las limitaciones de cada perspectiva individual.

9. Referencias bibliográficas

[1] M. Zentner, D. Grandjean y K. R. Scherer, «Emotions evoked by the sound of music: Characterization, classification, and measurement.,» *Emotion*, vol. 8, págs. 494-521, 4 ago. de 2008, ISSN: 1931-1516. DOI: 10.1037/1528-3542.8.4.494. dirección: https://doi.apa.org/doi/10.1037/1528-3542.8.4.494.

[2] J. de Berardinis, A. Cangelosi y E. Coutinho, «The multiple voices of musical emotions: source separation for improving music emotion recognition models and their interpretability,» en *International Society for Music Information Retrieval Conference*, 2020, págs. 310-317.

[3] H. He, Y. Tan, J. Ying y W. Zhang, «Strengthen EEG-based emotion recognition using firefly integrated optimization algorithm,» *Applied Soft Computing*, vol. 94, pág. 106 426, sep. de 2020, ISSN: 15684946. DOI: 10.1016/j.asoc.2020.106426. dirección: https://linkinghub.elsevier.com/retrieve/pii/S1568494620303665.

[4] N. Steinbeis, S. Koelsch y J. A. Sloboda, «The Role of Harmonic Expectancy Violations in Musical Emotions: Evidence from Subjective, Physiological, and Neural Responses,» *Journal of Cognitive Neuroscience*, vol. 18, págs. 1380-1393, 8 ago. de 2006, ISSN: 0898-929X. DOI: 10.1162/jocn.2006.18.8.1380. dirección:

https://direct.mit.edu/jocn/article/18/8/1380/4198/The-Role-of-Harmonic-Expectancy-Violations-in.

[5] K. Hevner, «Experimental Studies of the Elements of Expression in Music,» *The American Journal of Psychology*, vol. 48, pág. 246, 2 abr. de 1936, ISSN: 00029556. DOI: 10.2307/1415746. dirección: https://www.jstor.org/stable/1415746?origin=crossref.

[6] G. Athanasopoulos, T. Eerola, I. Lahdelma y M. Kaliakatsos-Papakostas, «Harmonic organisation conveys both universal and culture-specific cues for emotional expression in music,» *PLOS ONE*, vol. 16, M. Sadakata, ed., e0244964, 1 ene. de 2021, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0244964. dirección: https://dx.plos.org/10.1371/journal.pone.0244964.

[7] L. Meyer, *Emotion and Meaning in Music* (ACLS Humanities E-Book). University of Chicago Press, 1956, ISBN: 9780226521398. dirección: https://books.google.com.mx/books?id=HuWCVGKhwy0C.

[8] X. Cui, Y. Wu, J. Wu, Z. You, J. Xiahou y M. Ouyang, «A review: Music-emotion recognition and analysis based on EEG signals,» *Frontiers in Neuroinformatics*, vol. 16, pág. 997 282, oct. de 2022, ISSN: 16625196. DOI: 10.3389/FNINF.2022.997282/BIBTEX.

[9] M. B. Er e I. B. Aydilek, «Music Emotion Recognition by Using Chroma Spectrogram and Deep Visual Features,» *International Journal of Computational Intelligence Systems*, vol. 12, pág. 1622, 2 dic. de 2019, ISSN: 1875-6883. DOI: 10.2991/ijcis.d.191216.001. dirección: https://link.springer.com/10.2991/ijcis.d.191216.001.

[10] R. Panda, R. Malheiro y R. P. Paiva, «Novel Audio Features for Music Emotion Recognition,» *IEEE Transactions on Affective Computing*, vol. 11, págs. 614-626, 4 oct. de 2020, ISSN: 1949-3045. DOI: 10.1109/TAFFC.2018.2820691. dirección: https://ieeexplore.ieee.org/document/8327886/.

[11] S. Hizlisoy, S. Yildirim y Z. Tufekci, «Music emotion recognition using convolutional long short term memory deep neural networks,» *Engineering Science and Technology, an International Journal*, vol. 24, págs. 760-767, 3 jun. de 2021, ISSN: 22150986. DOI: 10.1016/j.jestch.2020.10.009. dirección: https://linkinghub.elsevier.com/retrieve/pii/S2215098620342385.

[12] L. Moysis et al., «Music Deep Learning: Deep Learning Methods for Music Signal Processing—A Review of the State-of-the-Art,» *IEEE Access*, vol. 11, págs. 17 031-17 052, 2023, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3244620. dirección: https://ieeexplore.ieee.org/document/10043650/.

[13] J. S. Gomez-Canon et al., «Music Emotion Recognition: Toward new, robust standards in personalized and context-sensitive applications,» *IEEE Signal Processing Magazine*, vol. 38, págs. 106-114, 6 nov. de 2021, ISSN: 1053-5888. DOI: 10.1109/MSP.2021.3106232. dirección: https://ieeexplore.ieee.org/document/9591555/.

[14] K. MIYAMOTO, H. TANAKA y S. NAKAMURA, «Online EEG-Based Emotion Prediction and Music Generation for Inducing Affective States,» *IEICE Transactions on Information and Systems*, vol. E105.D, 2021EDP7171, 5 mayo de 2022, ISSN: 0916-8532. DOI: 10.1587/transinf.2021EDP7171. dirección: https://www.jstage.jst.go.jp/article/transinf/E105.D/5/E105.D_2021EDP7171/_article.

[15] D. Han, Y. Kong, J. Han y G. Wang, «A survey of music emotion recognition,» *Frontiers of Computer Science*, vol. 16, pág. 166 335, 6 dic. de 2022, ISSN: 2095-2228. DOI: 10.1007/s11704-021-0569-4. dirección: https://link.springer.com/10.1007/s11704-021-0569-4.

[16] T.-M. Bynion y M. T. Feldner, «Self-Assessment Manikin,» en *Encyclopedia of Personality and Individual Differences*, V. Zeigler-Hill y T. K. Shackelford, eds. Cham: Springer International Publishing, 2017, págs. 1-3, ISBN: 978-3-319-28099-8. DOI: 10.1007/978-3-319-28099-8_77-1. dirección: https://doi.org/10.1007/978-3-319-28099-8_77-1.

[17] M. M. Bradley y P. J. Lang, «Measuring emotion: the self-assessment manikin and the semantic differential,» *Journal of behavior therapy and experimental psychiatry*, vol. 25, n.º 1, págs. 49-59, 1994.

[18] J. POSNER, J. A. RUSSELL y B. S. PETERSON, «The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology,» *Development and Psychopathology*, vol. 17, págs. 715-734, 03 sep. de 2005, ISSN: 0954-5794. DOI: 10.1017/S0954579405050340. dirección: http://www.journals.cambridge.org/abstract_S0954579405050340.

[19] J. A. Russell, «A circumplex model of affect.,» *Journal of Personality and Social Psychology*, vol. 39, págs. 1161-1178, 6 dic. de 1980, ISSN: 1939-1315. DOI: 10.1037/h0077714. dirección: https://doi.apa.org/doi/10.1037/h0077714.

[20] R. E. Thayer, *The Biopsychology of Mood and Arousal*. Oxford University Press-New York, NY, sep. de 1990, ISBN: 9780195068276. DOI: 10.1093/oso/9780195068276.001.0001. dirección: https://academic.oup.com/book/54447.

[21] J. Yang, «A Novel Music Emotion Recognition Model Using Neural Network Technology,» *Frontiers in Psychology*, vol. 12, pág. 760 060, sep. de 2021, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2021.760060. dirección: https://www.frontiersin.org/articles/10.3389/fpsyg.2021.760060/full.

[22] P. Du, X. Li e Y. Gao, «Dynamic Music emotion recognition based on CNN-BiLSTM,» en *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, jun. de 2020, págs. 1372-1376, ISBN: 978-1-7281-4323-1. DOI: 10.1109/ITOEC49072.2020.9141729. dirección: https://ieeexplore.ieee.org/document/9141729/.

[23] I.-S. Huang, Y.-H. Lu, M. Shafiq, A. A. Laghari y R. Yadav, «A Generative Adversarial Network Model Based on Intelligent Data Analytics for Music Emotion Recognition under IoT,» *Mobile Information Systems*, vol. 2021, A. Nayyar, ed., págs. 1-8, nov. de 2021, ISSN: 1875-905X. DOI: 10.1155/2021/3561829. dirección: https://www.hindawi.com/journals/misy/2021/3561829/.

[24] M. J. Lucia-Mulas, P. Revuelta-Sanz, B. Ruiz-Mezcua e I. Gonzalez-Carrasco, «Automatic music emotion classification model for movie soundtrack subtitling based on neuroscientific premises,» *Applied Intelligence*, vol. 53, págs. 27 096-27 109, 22 nov. de 2023, ISSN: 0924-669X. DOI: 10.1007/s10489-023-04967-w. dirección: https://link.springer.com/10.1007/s10489-023-04967-w.

[25] F. A. Baker et al., «Clinical effectiveness of music interventions for dementia and depression in elderly care (MIDDEL): Australian cohort of an international pragmatic cluster-randomised controlled trial,» *The Lancet Healthy Longevity*, vol. 3, e153-e165, 3 mar. de 2022, ISSN: 26667568. DOI: 10.1016/S2666-7568(22)00027-7. dirección: https://linkinghub.elsevier.com/retrieve/pii/S2666756822000277.

[26] M. Sharda, G. Silani, K. Specht, J. Tillmann, U. Nater y C. Gold, «Music therapy for children with autism: investigating social behaviour through music,» *The Lancet Child & Adolescent Health*, vol. 3, págs. 759-761, 11 nov. de 2019, ISSN: 23524642. DOI: 10.1016/S2352-4642(19)30265-2. dirección: https://linkinghub.elsevier.com/retrieve/pii/S2352464219302652.

[27] J. Pillay, «Music as a trigger for specific memories,» *Alzheimer's & Dementia*, vol. 19, e065259, S6 jun. de 2023, ISSN: 1552-5260. DOI: 10.1002/alz.065259. dirección: https://alz-journals.onlinelibrary.wiley.com/doi/10.1002/alz.065259.

[28] S. Anbalagan, J. H. Velasquez, D. S. Gutierrez, S. Devagiri, D. Nieto y P. Ankola, «Music for pain relief of minor procedures in term neonates,» *Pediatric Research*, vol. 95, págs. 679-683, 3 feb. de 2024, ISSN: 0031-3998. DOI: 10.1038/s41390-023-02746-4. dirección: https://www.nature.com/articles/s41390-023-02746-4.

[29] T.-L. Pao, Y.-M. Cheng, J.-H. Yeh, Y.-T. Chen, C.-Y. Pai e Y.-W. Tsai, «Comparison between Weighted D-KNN and Other Classifiers for Music Emotion Recognition,» en *2008 3rd International Conference on Innovative Computing Information and Control*, IEEE, 2008, págs. 530-530, ISBN: 978-0-7695-3161-8. DOI: 10.1109/ICICIC.2008.679. dirección: http://ieeexplore.ieee.org/document/4603719/.

[30] A. KSharma, A. Panwar y P. Chakrabarti, «Analytical Approach on Indian Classical Raga Measures by Feature Extraction with EM and Naive Bayes,» *International Journal of Computer Applications*, vol. 107, págs. 41-46, 6 dic. de 2014, ISSN: 09758887. DOI: 10.5120/18759-0035. dirección: http://research.ijcaonline.org/volume107/number6/pxc3900035.pdf.

[31] J.-C. Wang, Y.-H. Yang, H.-M. Wang y S.-K. Jeng, «Modeling the Affective Content of Music with a Gaussian Mixture Model,» *IEEE Transactions on Affective Computing*, vol. 6, págs. 56-68, 1 ene. de 2015, ISSN: 1949-3045. DOI: 10.1109/TAFFC.2015.2397457. dirección: http://ieeexplore.ieee.org/document/7029060/.

[32] B. Han, S. Rho, R. B. Dannenberg y E. Hwang, «SMERS: Music Emotion Recognition Using Support Vector Regression,» en *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, K. Hirata, G. Tzanetakis y K. Yoshii, eds., International Society for Music Information Retrieval, 2009, págs. 651-656. dirección: http://ismir2009.ismir.net/proceedings/PS4-13.pdf.

[33] C. Chen y Q. Li, «A Multimodal Music Emotion Classification Method Based on Multifeature Combined Network Classifier,» *Mathematical Problems in Engineering*, vol. 2020, págs. 1-11, ago. de 2020, ISSN: 1024-123X. DOI: 10.1155/2020/4606027. dirección: https://www.hindawi.com/journals/mpe/2020/4606027/.

[34] K. Pyrovolakis, P. Tzouveli y G. Stamou, «Multi-Modal Song Mood Detection with Deep Learning,» *Sensors*, vol. 22, pág. 1065, 3 ene. de 2022, ISSN: 1424-8220. DOI: 10.3390/s22031065. dirección: https://www.mdpi.com/1424-8220/22/3/1065.

[35] M. B. Er, H. Çiğ y İbrahim Berkan Aydilek, «A new approach to recognition of human emotions using brain signals and music stimuli,» *Applied Acoustics*, vol. 175, pág. 107 840, abr. de 2021, ISSN: 0003682X. DOI: 10.1016/j.apacoust.2020.107840. dirección: https://linkinghub.elsevier.com/retrieve/pii/S0003682X20309452.

[36] E. Herrera, *Teoría Musical y Armonía Moderna*. Antoni Bosch Editor, S.A., 1990, vol. I, ISBN: 9788485855315.

[37] T. Greer, K. Singla, B. Ma y S. Narayanan, «Learning Shared Vector Representations of Lyrics and Chords in Music,» en *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2019-May, IEEE, mayo de 2019, págs. 3951-3955, ISBN: 978-1-4799-8131-1. DOI: 10.1109/ICASSP.2019.8683735. dirección: https://ieeexplore.ieee.org/document/8683735/.

[38] A. Lahnala et al., «Chord Embeddings: Analyzing What They Capture and Their Role for Next Chord Prediction and Artist Attribute Prediction,» en *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Cham, 2021, vol. 12693 LNCS, págs. 171-186, ISBN: 978-3-030-72914-1. DOI: 10.1007/978-3-030-72914-1_12. dirección: https://link.springer.com/10.1007/978-3-030-72914-1_12.

[39] X. Hu, J. S. Downie, C. Laurier, M. Bay y A. F. Ehmann, «The 2007 MIREX Audio Mood Classification Task: Lessons Learned,» en *ISMIR 2008, 9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, PA, USA, September 14-18, 2008*, J. P. Bello, E. Chew y D. Turnbull, eds., 2008, págs. 462-467. dirección: http://ismir2008.ismir.net/papers/ISMIR2008_263.pdf.

[40] M. M. B. ©, *The 2013 Emotion in Music Task (A new .ᴬffect Task")*, inglés, http://www.multimediaeval.org/mediaeval2013/emotion2013/, Accedido en Junio de 2025, 2013.

[41] A. Aljanaki, Y. H. Yang y M. Soleymani, «Developing a benchmark for emotional analysis of music,» *PLOS ONE*, vol. 12, e0173392, 3 mar. de 2017, ISSN: 1932-6203. DOI: 10.1371/JOURNAL.PONE.0173392. dirección: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0173392.

[42] D. Su, P. Fung y N. Auguin, «Multimodal music emotion classification using AdaBoost with decision stumps,» en *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, mayo de 2013, págs. 3447-3451, ISBN: 978-1-4799-0356-6. DOI: 10.1109/ICASSP.2013.6638298. dirección: http://ieeexplore.ieee.org/document/6638298/.

[43] K. Markov, M. Iwata y T. Matsui, «Music emotion recognition using Gaussian Processes,» *MediaEval 2013 Multimedia Benchmark Workshop*, vol. Vol-1043, págs. 18-19, 2013, ISSN: 1613-0073. dirección: https://ceur-ws.org/Vol-1043/.

[44] K. Markov y T. Matsui, «Dynamic Music Emotion Recognition Using State-Space Models,» *MediaEval 2014 Multimedia Benchmark Workshop*, vol. 1263, págs. 16-17, 2014, ISSN: 1613-0073. dirección: https://ceur-ws.org/Vol-1263/.

[45] K. Markov y T. Matsui, «Music Genre and Emotion Recognition Using Gaussian Processes,» *IEEE Access*, vol. 2, págs. 688-697, 2014, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2014.2333095. dirección: http://ieeexplore.ieee.org/document/6843353/.

[46] L. Lu, D. Liu y H.-J. Zhang, «Automatic mood detection and tracking of music audio signals,» *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, págs. 5-18, 1 ene. de 2006, ISSN: 1558-7916. DOI: 10.1109/TSA.2005.860344. dirección: http://ieeexplore.ieee.org/document/1561259/.

[47] H. Xianyu et al., «SVR based double-scale regression for dynamic emotion prediction in music,» en *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2016-May, IEEE, mar. de 2016, págs. 549-553, ISBN: 978-1-4799-9988-0. DOI: 10.1109/ICASSP.2016.7471735. dirección: http://ieeexplore.ieee.org/document/7471735/.

[48] T. Greer, X. Shi, B. Ma y S. Narayanan, «Creating musical features using multi-faceted, multi-task encoders based on transformers,» *Scientific Reports*, vol. 13, pág. 10 713, 1 jul. de 2023, ISSN: 2045-2322. DOI: 10.1038/s41598-023-36714-z. dirección: https://www.nature.com/articles/s41598-023-36714-z.

[49] N. He y S. Ferguson, «Music emotion recognition based on segment-level two-stage learning,» *International Journal of Multimedia Information Retrieval*, vol. 11, págs. 383-394, 3 sep. de 2022, ISSN: 2192-6611. DOI: 10.1007/s13735-022-00230-z. dirección: https://link.springer.com/10.1007/s13735-022-00230-z.

[50] C. Fan, Z. Lv, S. Pei y M. Niu, «Csenet: Complex Squeeze-and-Excitation Network for Speech Depression Level Prediction,» en *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, págs. 546-550. DOI: 10.1109/ICASSP43922.2022.9746011.

[51] J. W. Yeow, E.-L. Tan, J. Bai, S. Peksi y W.-S. Gan, *Squeeze-and-Excite ResNet-Conformers for Sound Event Localization, Detection, and Distance Estimation for DCASE 2024 Challenge*, 2024. arXiv: 2407.09021 [eess.AS]. dirección: https://arxiv.org/abs/2407.09021.

[52] Z. Huang, S. Ji, Z. Hu, C. Cai, J. Luo y X. Yang, *ADFF: Attention Based Deep Feature Fusion Approach for Music Emotion Recognition*, 2022. arXiv: 2204.05649 [cs.SD]. dirección: https://arxiv.org/abs/2204.05649.

[53] Y.-H. Cho, H. Lim, D.-W. Kim e I.-K. Lee, «Music emotion recognition using chord progressions,» en *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, oct. de 2016, págs. 002 588-002 593, ISBN: 978-1-5090-1897-0. DOI: 10.1109/SMC.2016.7844628. dirección: http://ieeexplore.ieee.org/document/7844628/.

[54] F. Zhang, H. Meng, M. Li, R. Cui y C. Liu, «Music Emotion Recognition Based on Chord Identification,» en *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, H. Meng, T. Lei, M. Li, K. Li, N. Xiong y L. Wang, eds., Cham: Springer International Publishing, 2021, págs. 956-963, ISBN: 978-3-030-70665-4.

[55] X. Li, D. Song, P. Zhang, Y. Zhang, Y. Hou y B. Hu, «Exploring EEG features in cross-subject emotion recognition,» *Frontiers in Neuroscience*, vol. 12, pág. 294 333, MAR mar. de 2018, ISSN: 1662453X. DOI: 10.3389/FNINS.2018.00162/BIBTEX.

[56] B. Geethanjali, K. Adalarasu, M. Jagannath y N. P. G. Seshadri, «Music-Induced Brain Functional Connectivity Using EEG Sensors: A Study on Indian Music,» *IEEE Sensors Journal*, vol. 19, págs. 1499-1507, 4 feb. de 2019, ISSN: 1530-437X. DOI: 10.1109/JSEN.2018.2873402. dirección: https://ieeexplore.ieee.org/document/8478764/.

[57] F. Hasanzadeh, M. Annabestani y S. Moghimi, «Continuous emotion recognition during music listening using EEG signals: A fuzzy parallel cascades model,» *Applied Soft Computing*, vol. 101, pág. 107 028, mar. de 2021, ISSN: 15684946. DOI: 10.1016/j.asoc.2020.107028. dirección: https://linkinghub.elsevier.com/retrieve/pii/S1568494620309674.

[58] C. J. Ortiz-Echeverri, J. Rodríguez-Reséndiz y M. Garduño-Aparicio, «An approach to STFT and CWT learning through music hands-on labs,» *Computer Applications in Engineering Education*, vol. 26, n.º 6, págs. 2026-2035, 2018. DOI: https://doi.org/10.1002/cae.21967. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.21967. dirección: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.21967.

[59] O. A. Agustín Aquino y E. Lluis Puebla, «Una invitación a la teoría matemática de la música. I. Introducción y teoría de la interpretación,» *Ciencias*, vol. 101, n.º 101, ago. de 2011, ISSN: 0187-6376. dirección: https://www.revistas.unam.mx/index.php/cns/article/view/26597.

[60] O. A. Agustín Aquino y E. Lluis Puebla, «Una invitación a la teoría matemática de la música. II. Armonía y contrapunto,» *Ciencias*, vol. 102, n.º 102, feb. de 2012, ISSN: 0187-6376. dirección: https://www.revistas.unam.mx/index.php/cns/article/view/30135.

[61] J. C. Brown, «Calculation of a constant Q spectral transform,» *The Journal of the Acoustical Society of America*, vol. 89, n.º 1, págs. 425-434, ene. de 1991, ISSN: 0001-4966. DOI: 10.1121/1.400476. eprint: https://pubs.aip.org/asa/jasa/article-pdf/89/1/425/12135575/425_1_online.pdf. dirección: https://doi.org/10.1121/1.400476.

[62] B. McFee et al., *librosa/librosa: 0.11.0*, ver. 0.11.0, mar. de 2025. DOI: 10.5281/zenodo.15006942. dirección: https://doi.org/10.5281/zenodo.15006942.

[63] K. Egodawele y R. Ranasinghe, «2023 International Research Conference of Sri Lanka Technology Campus Colombo,» dic. de 2023.

[64] H. Purwins, B. Blankertz y K. Obermayer, «A new method for tracking modulations in tonal music in audio data format,» en *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 6, 2000, 270-275 vol.6. DOI: 10.1109/IJCNN.2000.859408.

[65] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination press San Francisco, CA, USA, 2015. dirección: http://neuralnetworksanddeeplearning.com/.

[66] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016, ISBN: 9780262337373. dirección: http://www.deeplearningbook.org.

[67] S. Ruder, «An overview of gradient descent optimization algorithms,» *arXiv preprint arXiv:1609.04747*, 2016.

[68] D. P. Kingma, «Adam: A method for stochastic optimization,» *arXiv preprint arXiv:1412.6980*, 2014.

[69] F. Chollet, *Deep learning with Python*. Simon y Schuster, 2021.

[70] D. Hendrycks y K. Gimpel, «Gaussian Error Linear Units (GELUs),» *arXiv preprint arXiv:1606.08415*, 2016.

[71] A. Graves y J. Schmidhuber, «Framewise phoneme classification with bidirectional LSTM networks,» en *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4, 2005, 2047-2052 vol. 4. DOI: 10.1109/IJCNN.2005.1556215.

[72] H. Mukherjee et al., «Music chord inversion shape identification with LSTM-RNN,» *Procedia Computer Science*, vol. 167, págs. 607-615, ene. de 2020, ISSN: 1877-0509. DOI: 10.1016/J.PROCS.2020.03.327.

[73] K. He, X. Zhang, S. Ren y J. Sun, *Deep Residual Learning for Image Recognition*, 2015. arXiv: 1512.03385 [cs.CV]. dirección: https://arxiv.org/abs/1512.03385.

[74] J. Hu, L. Shen y G. Sun, «Squeeze-and-Excitation Networks,» *CoRR*, vol. abs/1709.01507, 2017. arXiv: 1709.01507. dirección: http://arxiv.org/abs/1709.01507.

[75] T. Mikolov, K. Chen, G. Corrado y J. Dean, *Efficient Estimation of Word Representations in Vector Space*, 2013. arXiv: 1301.3781 [cs.CL]. dirección: https://arxiv.org/abs/1301.3781.

[76] K. Zhang, H. Zhang, S. Li, C. Yang y L. Sun, «The PMEmo Dataset for Music Emotion Recognition,» en *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ACM, jun. de 2018, págs. 135-142, ISBN: 9781450350464. DOI: 10.1145/3206025.3206037. dirección: https://dl.acm.org/doi/10.1145/3206025.3206037.

[77] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs y G. Widmer, «madmom,» en *Proceedings of the 24th ACM international conference on Multimedia*, ACM, oct. de 2016, págs. 1174-1178, ISBN: 9781450336031. DOI: 10.1145/2964284.2973795. dirección: https://dl.acm.org/doi/10.1145/2964284.2973795.

Anexos

A. Documentos

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS

A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

VILLANUEVA MEDINA LEONARDO DANIEL

Presentó y acreditó el **Examen de Comprensión de Textos en Inglés** efectuado el día cinco de febrero de dos mil veinticinco.

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día cinco de junio de dos mil veinticinco.

A t e n t a m e n t e,
"Enlazar Culturas por la Palabra"

DRA. MA. DE LOURDES RICO CRUZ

MLRC/mgoa*CL*FLL-C.-1145

UAQ CRECER EN LA DIVERSIDAD | fll.uaq.mx | 442 192 12 00 EXT. 61010 | Campus Aeropuerto, Anillo Vial Fray Junípero Serra s/n, Santiago de Querétaro, Qro. México. C.P. 76140.

FOLIO: 2099

Figura 28: Constancia de comprensión de textos en inglés

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS

A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

VILLANUEVA MEDINA LEONARDO DANIEL

Presentó el **Examen de Manejo de la Lengua** efectuado el día veintisiete de noviembre de dos mil veintitrés, en el cual obtuvo la siguiente calificación:

8-

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día ocho de agosto de dos mil veinticuatro.

A t e n t a m e n t e,
"Enlazar Culturas por la Palabra"

DRA. MA. DE LOURDES RICO CRUZ

MLRC/idos*CL*FLL-C.-38245

UAQ CRECER EN LA DIVERSIDAD
fll.uaq.mx
442 192 12 00 EXT. 61010
Campus Aeropuerto, Anillo Vial Fray Junípero Serra s/n, Santiago de Querétaro, Qro. México. C.P. 76140.
FOLIO: 63731

Figura 29: Constancia de manejo de la lengua inglesa

ISSN 1870-4069

NLP Applied to Musical Harmonic Structures for Music Emotion Recognition

Leonardo Daniel Villanueva Medina, Efrén Gorrostieta Hurtado

Universidad Autónoma de Queretaro,
Mexico

lvillanueva@uaq.mx

Abstract. Music emotion recognition (M.E.R.) is a multidisciplinary field that integrates computer science, affective computing, and neuroscience elements to analyze musical features to detect emotions. Most research in this field has focused on low and mid-level features, often ignoring theoretical and harmonic aspects of music. In this work, we propose using regression-based machine learning models applied to word embeddings in harmonic structures (chords). The results indicate an RMSE of 0.0252 and an R^2 score of 0.9751 for the valence dimension, in comparison with the arousal, an RMSE of 0.1319 and an R^2 score of 0.4676. These findings indicate that incorporating theoretical and harmonic concepts enhances the performance of M.E.R models, particularly in the valence dimension, reflecting improved detection of the positivity of emotions.

Keywords: mer, word embeddings, machine learning, musical features.

1 Introduction

Music has remarkably impacted social, cultural, and political aspects. For this reason, it has been the target of many studies, one of them being the relationship between emotions and music [13] since music is a means of expression capable of evoking emotions [6].

Music Emotion Recognition (M.E.R.) has incorporated knowledge from several fields, such as computer science, affective computing, and neuroscience. It aims to analyze musical features extracted from audio signals (low and mid-level) and abstract features such as song lyrics (high-level) [13, 9, 15, 7].

Within M.E.R.'s works, two approaches for linking emotions and songs predominate. The first one attaches a general emotion to the whole work (song-level), a static approach. The second, dynamic approach, focuses on detecting the music emotion variations that occur through the song, namely MEDV (music emotion variation) [9, 6].

Emotional perception is complex because it involves multiple variables, such as the song or external information, such as the listener's social, cultural, and emotional context [17, 7].

Selecting the appropriate taxonomy is crucial for clearly delineating the problem as either a multi-class classification or a regression task [9]. In this

Figura 30: Artículo publicado en la revista Research in Computing Science 154(5), 2025

Sociedad Mexicana de Inteligencia Artificial, A. C.
www.smia.org.mx

Metepec, Estado de México, 20 de abril de 2025

Leonardo Daniel Villanueva Medina
Efrén Gorrostieta Hurtado
PRESENTES

Por este conducto les informo que su artículo titulado “NLP Applied to Musical Harmonic Structures for Music Emotion Recognition” sometido a XVIII Congreso Mexicano de Inteligencia Artifial 2025, COMIA 2025, ha sido aceptado para su publicación en la revista Research in Computing Science. Agradecemos su participación y lo esperamos para su presentación en modalidad poster del 12 al 16 de mayo.

Agradeciendo la atención brindada a la presente y poniéndome a su disposición para cualquier aclaración, me despido de ustedes enviándoles un cordial saludo.

Dra. Bella Citlali Martínez Seis
Presidente de Comité de Programa de COMIA 2025

Figura 31: Carta de aceptación del artículo

Figura 32: Constancia de participación en el XVII Congreso Mexicano de Inteligencia Artificial - Comia 2025 por la presentación del artículo

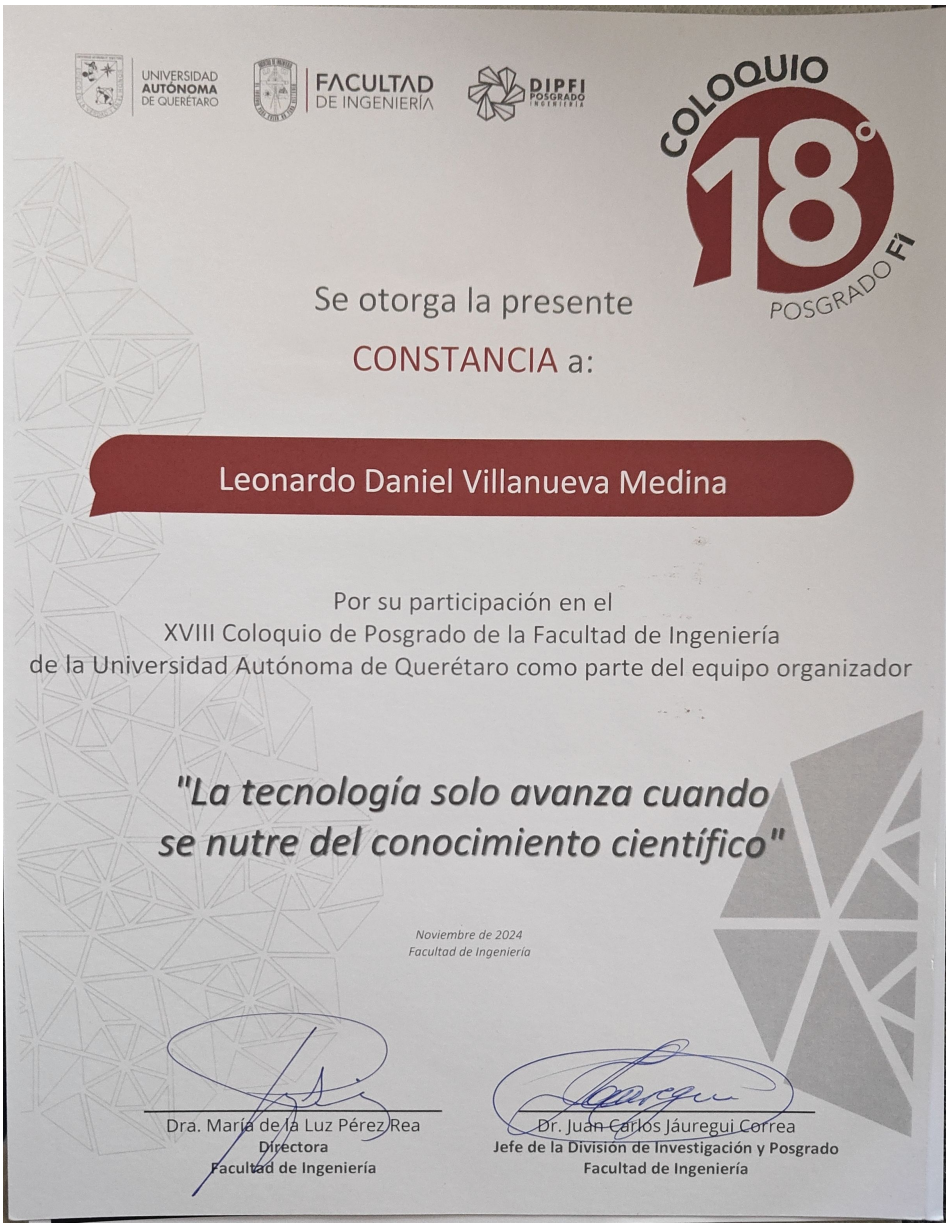

Figura 33: Constancia de participación en el XVIII Coloquio de Posgrado de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro por formar parte del staff del evento

FACULTAD DE INGENIERÍA
DIRECCIÓN DE INVESTIGACIÓN Y POSGRADO

Constancia de actividades de retribución social

Santiago de Querétaro a 10 de septiembre de 2025

A quien corresponda:
P r e s e n t e.

En cumplimiento a lo establecido en el *Artículo 20, Capítulo VII. De la Conclusión de la Beca o Apoyo*, del *Reglamento de Becas de la Secretaría de Ciencia, Humanidades Tecnología e Innovación*, y en el marco de la Convocatoria **Becas nacionales para estudios de posgrado 2023-2**, hago constar que el **C. Leonardo Daniel Villanueva Medina** con número de **CVU 1313344**, **beneficiado** con una beca para obtener el grado de **Maestría** en el programa **005351 – Maestría en Ciencias en Inteligencia Artificial**, que se imparte en la **Universidad Autónoma de Querétaro - Campus Aeropuerto**, realizó las actividades de retribución social durante el periodo de vigencia de la beca en el tiempo en el que fue **alumno** regular de esta Institución.

Asimismo, hago constar que, conforme a lo establecido en la Ley General de Archivos, la coordinación del posgrado organiza y conserva la evidencia documental de dichas actividades en caso de que la SECIHTI o cualquier otra instancia la requiera.

Sin más por el momento, le envío un cordial saludo.

Dr. Saúl Tovar Arriaga

Coordinador del programa Maestría en Ciencias en Inteligencia Artificial

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
CERRO DE LAS CAMPANAS S/N, COL. LAS CAMPANAS
FACULTAD DE INGENIERÍA, EDIFICIO G.
C.P. 76010 QUERÉTARO, QRO. TEL. 192 12 00 EXT. 6023

Figura 34: Carta de retribución social