EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación y Desarrollo Tecnológico

Tesis de Doctorado

Modelo computacional de detección de emociones basado en frecuencias cardiacas y estímulos audio-visuales

presentada por

M.C. Alvaro Abraham Colunga Rodriguez

como requisito para la obtención del grado de

Doctor en Ciencias de la Computación

Director de tesis

Dra. Alicia Martínez Rebollar

Codirector de tesis

Dr. Hugo Estrada Esquivel

Cuernavaca, Morelos, México. Agosto de 2025.

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

cenidet® Centro Nacional de Investigación y Desarrollo Tecnológico	ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL	Código: CENIDET-AC-006-D20
		Revisión: O
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4	Página 1 de 1

Cuernavaca, Mor., a 10 de junio de 2025

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
P R E S E N T E

AT'n: DR. JUAN GABRIEL GONZÁLEZ SERNA
PRESIDENTE DEL CLAUSTRO DOCTORAL DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Los abajo firmantes, miembros del Comité Tutorial del estudiante **M.C. Álvaro Abraham Colunga Rodríguez** manifiestan que después de haber revisado el documento de tesis titulado **"Modelo Computacional de Detección de Emociones Basado en Frecuencias Cardíacas y Estímulos Audio-Visuales"**, realizado bajo la dirección de la **Dra. Alicia Martínez Rebollar** y la codirección del **Dr. Hugo Estrada Esquivel,** el trabajo se ACEPTA para proceder a su impresión.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"

DRA. ALICIA MARTÍNEZ REBOLLAR
TECNM/CENIDET

DR. HUGO ESTRADA ESQUIVEL
TECNM/CENIDET

DR. EDDIE HELBERT CLEMENTE TORRES
TECNM/CENIDET

DR. JAVIER ORTIZ HERNÁNDEZ
TECNM/CENIDET

DR. DANTE MÚJICA VARGAS
TECNM/CENIDET

DRA. MARTA LILIA ERAÑA DÍAZ
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS (UAEM)

c.c.p: C Verónica Sotelo Boyas/ Jefa del Departamento de Servicios Escolares
c.c.p: C Nóe Alejandro Castro Sánchez / Jefe del Departamento de Ciencias Computacionales
c.c.p: Expediente

Centro Nacional de Investigación y Desarrollo tecnológico
Subdirección Académica

Cuernavaca Mor, 11/junio/2025
Oficio No. SAC/136/2025
Asunto: Autorización de impresión de tesis

ÁLVARO ABRAHAM COLUNGA RODRÍGUEZ
CANDIDATO AL GRADO DE DOCTOR
EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Modelo Computacional de Detección de Emociones Basado en Frecuencias Cardiacas y Estímulos Audio-Visuales"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

c.c.p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/lmz

Resumen

La presente tesis propone el desarrollo de un modelo computacional híbrido para la detección automática de emociones humanas utilizando señales fisiológicas, particularmente el ritmo cardíaco, y estímulos audiovisuales. El trabajo se enmarca dentro del área de la computación afectiva, cuyo objetivo es dotar a los sistemas inteligentes de la capacidad para percibir, interpretar y responder a emociones humanas. Para ello, se diseñó y ejecutó un experimento controlado en el que se expuso a participantes a estímulos audiovisuales con el fin de inducir emociones como calma, felicidad, tristeza y enojo, mientras se registraban sus señales de ritmo cardíaco. Además, se utilizó el conjunto de datos RAVDESS para desarrollar un modelo complementario de detección de emociones a partir de la voz.

Se emplearon técnicas de aprendizaje automático, redes neuronales y programación genética. En particular, se propuso un modelo híbrido que combina la capacidad de abstracción de las redes neuronales con la flexibilidad estructural de la programación genética multiárbol. Esta combinación permitió transformar y seleccionar automáticamente características relevantes a partir de señales fisiológicas y acústicas, generando representaciones optimizadas para la clasificación emocional. Se extrajeron características tanto estadísticas como espectrales, incluyendo transformadas wavelet y coeficientes cepstrales (MFCC).

Los modelos fueron evaluados utilizando métricas estándar como precisión, sensibilidad y F1-score, y se demostró que la estrategia híbrida mejora significativamente el rendimiento de la clasificación en comparación con modelos convencionales. Los resultados obtenidos validan la viabilidad de emplear señales fisiológicas y sensoriales para la detección emocional, abriendo la puerta a futuras aplicaciones en contextos como salud, educación y sistemas interactivos. Finalmente, se identificaron líneas de trabajo futuro orientadas a mejorar la interpretabilidad, generalización y combinación de múltiples fuentes fisiológicas en entornos reales.

Abstract

This thesis proposes the development of a hybrid computational model for the automatic detection of human emotions using physiological signals, specifically heart rate, and audiovisual stimuli. The work is framed within the field of affective computing, which aims to equip intelligent systems with the ability to perceive, interpret, and respond to human emotions. To achieve this, a controlled experiment was designed and executed, in which participants were exposed to audiovisual stimuli intended to induce emotions such as calmness, happiness, sadness, and anger, while their heart rate signals were recorded. Additionally, the RAVDESS dataset was used to develop a complementary emotion detection model based on speech.

Machine learning techniques, neural networks, and genetic programming were employed. In particular, a hybrid model was proposed that combines the abstraction capabilities of neural networks with the structural flexibility of multi-tree genetic programming. This combination enabled automatic transformation and selection of relevant features from physiological and acoustic signals, generating optimized representations for emotional classification. Both statistical and spectral features were extracted, including wavelet transforms and Mel-Frequency Cepstral Coefficients (MFCCs).

The models were evaluated using standard metrics such as accuracy, recall, and F1-score, demonstrating that the hybrid strategy significantly improves classification performance compared to conventional models. The results validate the feasibility of using physiological and sensory signals for emotion detection, paving the way for future applications in contexts such as healthcare, education, and interactive systems. Finally, future work is proposed to improve model interpretability, generalization, and the integration of multiple physiological signals in real-world environments.

AGRADECIMIENTOS

Al Tecnológico Nacional de México por permitirme continuar con mi formación académica.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el apoyo brindado para desarrollar este trabajo de investigación.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por darme la oportunidad de integrarme a su programa de doctorado.

A mis directores de Tesis por todo su apoyo, consejos y guía durante mi estancia en el CENIDET.

A mis revisores que con sus comentarios nutrieron éste trabajo de investigación.

A mi esposa e hijo por todo su apoyo incondicional.

A mi familia en general por sus buenos deseos y oraciones.

A mis compañeros que me apoyaron siempre que lo necesité.

Contenido

Lista de figuras

Lista de tablas

Capítulo 1

Introducción

1.1. Introducción

Las emociones juegan un papel importante en la vida de las personas y pueden definirse como una reacción a estímulos externos. Las emociones tienen una función adaptativa que nos permite actuar para sobrellevar alguna situación determinada. Las emociones positivas están ligadas al placer o bienestar y pueden tener un efecto positivo en la calidad de vida. Las emociones negativas por su parte son desagradables y aunque se asocian con un papel de adaptación o supervivencia pueden tener un impacto en la salud de las personas (Xiao, Zhang, Lin, y Cai, 2021).

Las computadoras no pueden entender las emociones de un usuario y reaccionar de acuerdo al estado emocional como lo hacen las personas. Es así como surge la computación afectiva que es el campo de investigación que tiene como objetivo proveer a sistemas inteligentes la habilidad de percibir, interpretar y reaccionar a emociones humanas. Sus aplicaciones desde el área de la salud, educación y ámbito laboral (Daily y cols., 2017). Las personas usualmente expresan sus emociones mediante formas externas (visualmente, voz, gestos) y de forma interna (ritmo cardíaco, respiración, presión arterial, temperatura corporal, actividad cerebral) (Guanghui y Xiaoping, 2021). Un sistema que puede adaptarse o responder de acuerdo a las emociones de una persona puede ofrecer una mejor experiencia de uso. La adaptabilidad basada en emociones no solo mejora la experiencia del usuario al hacer que la tecnología sea más sensible y receptiva, sino que también contribuye a la eficiencia y eficacia del sistema. La respuesta adecuada a las necesidades emocionales de las personas, no solo proporcionan soluciones, sino que generan una interacción que más auténtica, comprensiva y más satisfactoria para el usuario.

En este trabajo se desarrolla un modelo que pueda aplicarse para la detección de emociones de una persona mediante interacciones con un sistema inteligente, de tal forma que pueda aplicarse en diferentes contextos en donde se requiera como por ejemplo:

interacción con un robot (Pham, Do, Su, Bishop, y Sheng, 2021), desarrollo de tutores inteligentes (Khadimallah, Abdelkefi, y Kallel, 2020), entre otros.

1.2. Estado del arte

La detección de emociones de una persona mediante un sistema informático necesita adquirir información mediante interacciones con ella. Uno de los datos utilizados para la detección de emociones es la imagen del rostro de una persona de la cual se extraen sus expresiones faciales, los métodos para extraer las características están basados en: detección de bordes, características locales y globales, geométricas y de textura (Chandraprabha, Shwetha, Kavitha, y Sumathi, 2021) (Ramos y Dadiz, 2018). Otros trabajos como (Bhadangkar, Pujari, y Yakkundimath, 2020) utilizan el análisis de componente principal (PCA, por sus siglas en inglés) y el análisis lineal discriminante (LDA, por sus siglas en inglés) en donde son empleados para reducir la dimensionalidad de los vectores de características, realizando una comparación de varios algoritmos de clasificación: Redes Neuronales Artificiales (ANN, por sus siglas en inglés), Bosque aleatorio (RF, por sus siglas en inglés) y Máquinas de Soporte Vectorial (SVM, por sus siglas en inglés). Una aplicación de la detección de emociones usando expresiones faciales [16], propone una técnica pedagógica de acuerdo a la emoción de los estudiantes. Otra aplicación hace uso de la cámara web de una computadora para tomar imágenes en intervalos de tiempo para poder detectar las emociones de los empleados de una empresa y poder realizar un seguimiento de su estado emocional (Chandraprabha y cols., 2021).

Otros métodos utilizados para la detección de emociones son los que emplean las señales acústicas de la voz. En el reconocimiento de emociones, la creación de conjuntos de datos adecuados para entrenar modelos de clasificación es crucial. Las características acústicas extraídas de los archivos de audio deben captar elementos clave del habla o del sonido, como el tono, la intensidad, la duración y otros parámetros relevantes para identificar emociones. Los conjuntos de datos, diseñados cuidadosamente basados en estas características, han sido fundamentales para entrenar modelos que buscan mejorar la precisión en la clasificación de emociones. Los siguientes trabajos relacionados exploran varias metodologías y enfoques utilizados para construir y emplear estos conjuntos de datos en la detección de emociones en el habla. En el trabajo de (Singh y Prasad, 2023), se construyó una red convolucional para el reconocimiento de emociones en el habla dependiente del género. Los datos utilizados para esta red se formaron a partir de características MFCC (Coeficientes Cepstrales de Mel) y sus variantes delta MFCC y delta–delta MFCC. El reconocimiento de emociones en el habla alcanzó una precisión promedio del 72.07 %. Sin embargo, este trabajo se basa en el género de la persona, lo que puede limitar la generalización del reconocimiento de emociones cuando la información de género no está disponible.

Otro ejemplo de reconocimiento de emociones en el habla fue desarrollado por (Abdulmohsin,

Abdul Wahab, y Abdul Hossen, 2021), donde se utilizó una red neuronal perceptrón para detectar emociones, mostrando un resultado de clasificación del 86.1 %. Este trabajo emplea características estadísticas como la tasa de cruce por cero (ZCR), entropía, energía, desviación del ZCR, desviación de la energía, Haar, función de Fourier, función de aptitud de MATLAB, función de ruido, función de tono, MFCC, coeficientes de cepstrum gammatone (GTCC) y la proporción armónica. La gran variedad de características utilizadas puede hacer que el modelo sea complejo y difícil de replicar. Otra red neuronal convolucional (Zisad, Hossain, y Andersson, 2020), diseñada para el reconocimiento de emociones en el habla de personas con trastornos neurológicos, utiliza únicamente la característica MFCC para formar tres conjuntos de datos: uno para entrenamiento, otro para prueba y uno para validación. Los resultados obtenidos en el reconocimiento de emociones, medidos en entrenamiento, prueba y validación, son 0.841, 0.740 y 0.744, respectivamente. El trabajo muestra buenos resultados en el entrenamiento; sin embargo, la diferencia significativa con los otros conjuntos de datos puede indicar que el modelo no se generaliza bien a datos nuevos.

El aprendizaje por transferencia es una técnica que utiliza una red convolucional preentrenada para aprovechar el conocimiento adquirido. El trabajo realizado por (Luna-Jiménez y cols., 2021) utiliza datos obtenidos de imágenes y audio, y los introduce en dos redes preentrenadas para clasificar emociones. Los archivos de audio se procesan mediante una red convolucional para extraer características, y luego se utiliza el algoritmo de máquina de vectores de soporte para la clasificación, obteniéndose una precisión del 80.08 %. El aprendizaje por transferencia puede ser una buena alternativa para evitar entrenar una red neuronal desde cero; sin embargo, usar una red entrenada con imágenes puede limitar su capacidad para reconocer datos de audio de manera efectiva. Otro tipo de red neuronal, denominada red neuronal probabilística (Deshmukh y Gupta, 2023), se empleó utilizando datos obtenidos del tono, croma, roll-off, centroide espectral, ancho de banda y ZCR. Los resultados muestran una precisión del 84.64 % en la clasificación de emociones en el habla. La limitada cantidad de características utilizadas puede no captar toda la información emocional del audio; este trabajo podría incluir características más variadas para mejorar su precisión. Las redes profundas son un subconjunto de las redes convolucionales que cuentan con más capas en su arquitectura.

El trabajo desarrollado por (Bhattacharya, Borah, Mishra, y Mondal, 2022) utiliza una red neuronal convolucional que emplea un conjunto de datos construido con espectrograma mel, croma, MFCC, contraste y características tonnetz. El resultado es una precisión del 77.60 % en la clasificación. El número limitado de características utilizadas puede influir en los resultados, por lo que este trabajo podría explorar el uso de otras características para mejorar su rendimiento.

El trabajo de (Paul, Bera, Dey, y Phadikar, 2024) compara varios algoritmos para la detección de emociones en el habla. Entre estos algoritmos se encuentran las máquinas de vectores de soporte, K-vecinos más cercanos, árboles de decisión y el análisis discriminante lineal. El conjunto de datos utilizado se construye combinando características como MFCC,

LPC, tono, energía y ZCR. Con el algoritmo de K-vecinos más cercanos se logró una precisión del 95 %. Aunque estos resultados son buenos, este algoritmo suele ser criticado por ser ineficiente con conjuntos de datos grandes, lo que limita la escalabilidad del modelo. La fusión de datos se puede aplicar tanto a nivel de características como a nivel de decisiones en la clasificación.

El trabajo en (Mishra, Warule, y Deb, 2024) demuestra la fusión de características y el uso de redes neuronales convolucionales y profundas en un conjunto de datos construido con características MFCC, espectrograma mel y espectrograma. El resultado es una precisión del 80.42 % en la clasificación de emociones. Este trabajo podría explorar métodos de selección de características para optimizar la combinación de las mismas.

El uso de dispositivos para la detección de emociones ha crecido significativamente en los últimos años debido a su capacidad para recopilar datos fisiológicos de manera continua y no invasiva. Entre las variables más utilizadas en estos estudios destacan el ritmo cardíaco (HR), la respuesta galvánica de la piel (GSR) y la temperatura de la piel (SKT), las cuales han sido ampliamente exploradas en el contexto del reconocimiento de emociones. En esta sección se presentan algunos trabajos de investigación relevantes en la detección de emociones.

El estudio de (Pepa, Capecci, y Ceravolo, 2019) emplea un reloj inteligente para monitorear las emociones en pacientes con enfermedad de Parkinson, utilizando datos obtenidos de una banda inteligente en un entorno controlado. Las emociones se clasifican en términos de valencia y activación, siguiendo el modelo de (Russell, s.f.), alcanzando una precisión del 93.4 % en valencia y 78.6 % en activación.

Por otro lado, (Francisti y cols., 2023) analiza el impacto de las emociones en el rendimiento académico durante la pandemia de COVID-19. Aunque su trabajo no realiza una clasificación explícita de emociones, sus hallazgos sugieren que los cambios en la activación emocional, medidos a través del ritmo cardíaco, están relacionados con el desempeño de los estudiantes. Este estudio refuerza la viabilidad del uso de dispositivos inteligentes para el monitoreo emocional.

Otras propuestas de investigación han explorado la clasificación de emociones empleando datos obtenidos de sensores de relojes inteligentes. (Wang y cols., 2020) propone un sistema adaptativo para la detección de felicidad, tristeza, enojo, miedo y estado neutral, utilizando volumen sanguíneo, actividad electrodermal y temperatura de la piel, con una efectividad del 74.3 %. (Shu y cols., 2020) desarrolla un modelo basado en el ritmo cardíaco, logrando una precisión del 84 % en la clasificación de neutral, felicidad y tristeza. (Takeshita, Shoji, Hossain, Yokokubo, y Lopez, 2021) enfoca su investigación en la detección de miedo y no miedo a partir de datos obtenidos de un reloj inteligente durante la visualización de videos de diferentes géneros, alcanzando una efectividad del 90 %. Estos estudios destacan la importancia de los dispositivos inteligentes en la detección de emociones y subrayan las diferencias en las métricas de precisión según las emociones evaluadas y las metodologías utilizadas. El modelo propuesto en esta investigación busca optimizar el proceso de recono-

cimiento de emociones, combinando la capacidad adaptativa de la Programación Genética con la robustez en el aprendizaje de patrones de las Redes Neuronales en la detección de emociones a partir de señales fisiológicas.

1.3. Motivación

La inteligencia artificial ha permitido el desarrollo de aplicaciones que interactúan con las personas de manera más activa. Los sistemas inteligentes que operan detrás de plataformas como redes sociales y servicios de streaming adaptan su contenido según el uso y las preferencias de los usuarios (Renugadevi y cols., 2024) (Stoynov, 2023).

La detección de emociones como parte de la computación afectiva (Picard, 1995) es un campo de investigación que ha cobrado un interés creciente en los últimos años debido a su potencial para transformar la interacción entre humanos y máquinas. Integrar las emociones en los sistemas inteligentes puede mejorar significativamente la experiencia del usuario. Aplicaciones en marketing, plataformas educativas y videojuegos podrían adaptar su contenido a partir del reconocimiento de las emociones de los usuarios.

Uno de los indicadores fisiológicos que reflejan las emociones es el ritmo cardíaco (Pham y cols., 2021). Dispositivos como relojes inteligentes y otros accesorios que miden y registran este dato generan información valiosa que puede utilizarse como entrada para la detección de emociones. Esta tecnología permitiría desarrollar sistemas capaces de responder no solo a las acciones, sino también a las emociones de las personas.

La expresión de emociones se ve afectada por factores como la cultura, las experiencias personales y otros aspectos influyen en la manera en que cada persona manifiesta sus emociones. Un modelo automático diseñado para detectar emociones en un individuo puede no ser aplicable a otro. El monitoreo de las emociones de una persona es fundamental porque permite desarrollar sistemas inteligentes que adapten sus respuestas de manera personalizada. Comprender y responder a las emociones no solo optimiza el funcionamiento de la aplicaciones, sino que también facilita la regulación emocional en tiempo real, lo que puede contribuir al bienestar psicológico de los usuarios (Yu, Bai, y Li, 2023).

1.4. Objetivos

Objetivo general

Desarrollar y evaluar un modelo computacional híbrido utilizando técnicas de programación genética y redes neuronales para la detección automática de emociones humanas a partir del ritmo cardíaco y estímulos audiovisuales, así como modelos complementarios con voz basados en aprendizaje automático.

Objetivos específicos

- Diseñar y ejecutar un experimento controlado, para construir un conjunto de datos etiquetado que relacione señales de ritmo cardíaco con respuestas emocionales, con el uso de estímulos audiovisuales seleccionados para inducir emociones específicas en condiciones experimentales reproducibles.
- Desarrollar un modelo de detección automática de emociones en la voz, con el fin de identificar el estado emocional de una persona a partir de sus expresiones acústicas, utilizando técnicas de aprendizaje automático y redes neuronales entrenadas con características extraídas de un conjunto de datos de audio previamente etiquetado.
- Desarrollar un modelo que utilice estímulos audio visuales y el ritmo cardíaco de una persona para identificar sus emociones utilizando programación genética y redes neuronales.
- Desarrollar un modelo híbrido que combine programación genética y redes neuronales para detectar emociones a partir del ritmo cardíaco y estímulos audiovisuales.
- Evaluar y comparar el desempeño de los modelos desarrollados, para determinar su capacidad de detección emocional en la voz y ritmo cardíaco bajo estímulos audiovisuales con el uso de métricas estándar de clasificación aplicadas a los datos obtenidos en el experimento controlado.

1.5. Hipótesis

El uso conjunto de aprendizaje automático y programación genética permite desarrollar modelos computacionales capaces de identificar de forma automática las emociones de una persona, a partir del análisis combinado de estímulos audiovisuales, señales de voz y la variabilidad del ritmo cardíaco.

1.6. Contribuciones

En el campo de las ciencias computacionales las aportaciones de este trabajo se resumen a continuación.

Se creó un conjunto de datos que combina ritmo cardíaco, características de audio y etiquetas emocionales, usando un experimento controlado con estímulos emocionales para provocar emociones.

Se desarrolló un modelo para detectar emociones en la voz usando redes neuronales y técnicas de aprendizaje automático. Este modelo se entrenó con el conjunto de datos RAVDESS, usando características espectrales y wavelets, y se evaluó con validación cruzada.

Se desarrolló un modelo híbrido que combina programación genética y redes neuronales para detectar emociones a partir del ritmo cardíaco, integrando ambas técnicas para mejorar la precisión en la clasificación.

Se empleó la técnica de programación genética multiárbol para crear un método de transformación de datos multimodales (audio, imagen y señales fisiológicas), optimizado para generar representaciones útiles en una red neuronal utilizada como clasificador, con una cantidad reducida de variables.

1.7. Organización de la tesis

La estructura en la que está organizado este trabajo de investigación es la siguiente:

El capítulo 2 define los conceptos necesarios para el desarrollo y fundamento del trabajo.

El capítulo 3 muestra la metodolgía seguida para la obtención de los datos, y en el entrenamiento de los modelos de clasificación.

El capítulo 4 presenta los resultados obtenidos por el modelo de detección de emociones en ritmo cardíaco utilizando estímulos audio visuales.

El capítulo 5 presenta algunas conclusiones y trabajos futuros.

Capítulo 2

Marco teórico

2.1. Emociones

Las emociones son una parte crucial de nuestra actividad mental y juega un papel importante en la forma en que pensamos y nos comportamos. Las emociones también son un complejo conjunto de interacciones con factores objetivos y subjetivos, mediados por los sistemas neuro-homonales, los cuales pueden: (a) convertirse experiencias afectivas tales como los sentimientos de placer/disgusto y excitación; (b) generan procesos cognitivos como pueden ser efectos en la percepción emocionalmente relevantes, procesos de etiquetado, evaluaciones; (c) activar ajustes psicológicos a las condiciones de evaluación y (d) conducir a un comportamiento en generalmente, pero que no siempre es así, expresivo, enfocado y adaptativo (Kleinginna y Kleinginna, s.f.).

Existen seis emociones básicas: felicidad, tristeza, enojo, miedo y disgusto (Ekman, s.f.). Las emociones positivas son reacciones emocionales que expresan afectividad positiva, como la felicidad cuando se alcanza una meta, alivio cuando un peligro ha sido evitado o contento cuando se está satisfecho con situaciones actuales (APA, s.f.).

Las emociones negativas por su lado son desagradables, normalmente disruptivas y esta diseñadas para expresar afectividad negativa. Una emoción negativa no contribuye al logro de una meta. Ejemplos pueden ser el enojo, miedo, envidia y tristeza (APA, s.f.).

El modelo de russell (Russell, s.f.) es un modelo de dos dimensiones en donde ubica a las emociones dentro de una circunferencia se muestra en la Figura 2.1. El eje horizontal representa la valencia que se refiere al grado de placer de la emoción; el eje vertical representa la activación que se entiende como la intensidad de la emoción.

El modelo se divide en cuatro cuadrantes en donde:

- 1er cuadrante: valencia positiva (0°) y activacion alta (90°).

- 2do cuadrante: valencia negativa (180°) y activacion alta (90°).
- 3er cuadrante: valencia negativa (180°) y activacion baja (270°).
- 4to cuadrante: valencia positiva (0°) y activacion baja (270°).

El modelo define 28 estados afectivos de acuerdo sus valores de valencia y activación como resultado de las investigaciones de Russell.

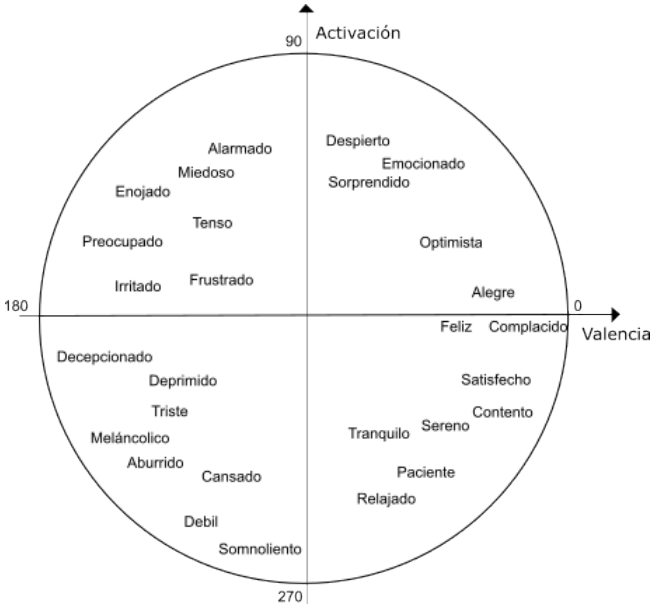

Figura 2.1 Modelo circunflejo.

2.2. Modelo predictivo

Un modelo predictivo (Kuhn y Johnson, 2013) es una herramienta matemática que usa datos históricos para estimar la probabilidad de eventos futuros sin enfocarse en comprender necesariamente las causas detrás de ellos. Su principal objetivo es la precisión en la predicción, optimizando su capacidad de anticipar resultados con base en patrones identificados en los datos.

Características de un modelo predictivo:

1. Uso de datos históricos: Se basa en datos previos para identificar tendencias y relaciones.
2. Enfoque en precisión: Su prioridad es hacer predicciones certeras más que interpretar el porqué de los resultados.

3. Automatización del proceso: Puede operar sin intervención humana directa, como en filtros de spam o sistemas de valoración de inmuebles.
4. Aplicación en diversas áreas: Se usa en campos como medicina, economía y tecnología para tomar decisiones basadas en datos.

Componentes de un modelo predictivo:

- Datos de entrada: Información relevante recopilada para construir el modelo.
- Preprocesamiento: Limpieza y transformación de los datos para eliminar ruido y mejorar la calidad del análisis.
- Modelo matemático: Algoritmo o conjunto de reglas utilizadas para hacer predicciones.
- Evaluación y validación: Uso de métricas para medir la efectividad del modelo y asegurar su precisión en datos nuevos.
- Balance entre precisión e interpretabilidad: Modelos más complejos suelen ser más precisos pero menos comprensibles, mientras que modelos más simples pueden ser interpretables pero menos efectivos.

Un modelo predictivo eficaz combina la capacidad de análisis computacional con el conocimiento experto del problema que aborda, asegurando que las predicciones sean confiables y útiles en la toma de decisiones.

2.3. Métricas de evaluación de modelos

Las métricas de evaluación (Zheng, 2015) se utilizan para medir qué tan bien funciona un modelo predictivo. El desempeño de un modelo predictivo se refiere a su capacidad para predecir correctamente la categoría a la que pertenece un nuevo dato o un dato no visto antes. Elegir las métricas adecuadas es muy importante, ya que influyen en la forma en que se mide y compara el rendimiento de los algoritmos de aprendizaje.

Cuando un modelo predictivo clasifica un dato, pueden darse cuatro posibles resultados:

- Verdadero Positivo (TP): Cuando el dato realmente es positivo y el modelo lo clasifica como positivo.
- Falso Negativo (FN): Cuando el dato es positivo, pero el modelo lo clasifica como negativo.

- Verdadero Negativo (TN): Cuando el dato es negativo y el modelo lo clasifica correctamente como negativo.
- Falso Positivo (FP): Cuando el dato es negativo, pero el modelo lo clasifica como positivo.

Una forma de visualizar los resultados es mediante una matriz de confusión. La matriz de confusión como de puede ver en la Figura 2.2 , organiza y resume los resultados de un modelo de clasificación, mostrando la cantidad de predicciones correctas e incorrectas en diferentes categorías.

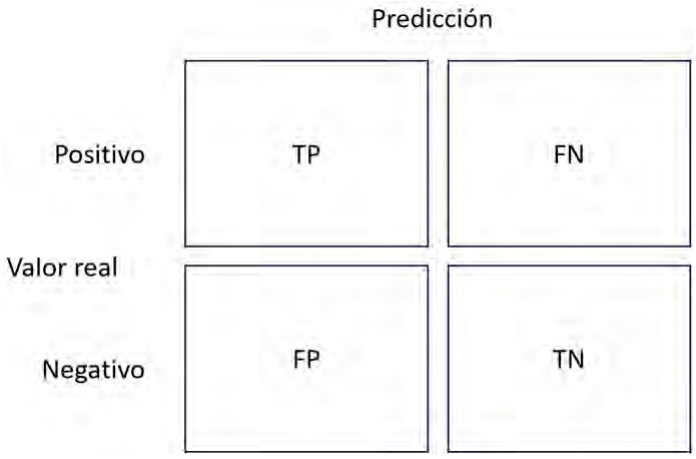

Figura 2.2 Matriz de confusión.

Para el cálculo de las métricas se utilizan las siguientes formulas:

Accuracy o exactitud: es la proporción de instancias que el modelo clasifica correctamente tanto positivas como negativas.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.1}$$

Precisión: es la proporción de instancias que el modelo clasifica como positivas y que en realidad son positivas.

$$precision = \frac{TP}{TP + FP} \tag{2.2}$$

Recall: mide la proporción de instancias positivas reales que el modelo identifica correctamente.

$$recall = \frac{TP}{TP + FN} \tag{2.3}$$

F1-score: Combina las métricas de precisión y recall para definir la capacidad de clasificación de un modelo.

$$F1 = 2 * \frac{precicion * recall}{precision + recall} \tag{2.4}$$

2.4. Programación genética

Las técnicas de cómputo evolutivo (Gestal, Cebrián, Rabuñal, Dorado, y Pazos, 2010) se basan en la teoría de la evolución de Darwin. Los algoritmos genéticos, optimización por enjambre de partículas, programación genética son algunas de las técnicas utilizadas en computo evolutivo. La programación genética tiene la ventaja de construir soluciones que tengan una representación flexible usualmente en forma de árbol.

La programación genética comienza con una población generada aleatoriamente compuesta por individuos o programas. Los nodos terminales son seleccionados aleatoriamente de un conjunto de terminales que contiene variables o constantes. Los nodos internos son seleccionados de un conjunto de funciones que contiene operaciones aritméticas y operadores lógicos. Los individuos son expresiones que incluyen a las características. Los individuos son evaluados por una función de aptitud y los que tengan mayor aptitud son seleccionados para la siguiente generación. Los operadores genéticos de cruce y mutación son aplicados para producir nuevos individuos. El proceso de programación genética continúa seleccionando individuos para la siguiente generación y generando individuos mediante los operadores genéticos hasta que se cumple un criterio de paro. El proceso de programación genética tiene como salida al mejor individuo de la población, la Figura 2.3 muestra un diagrama de este proceso.

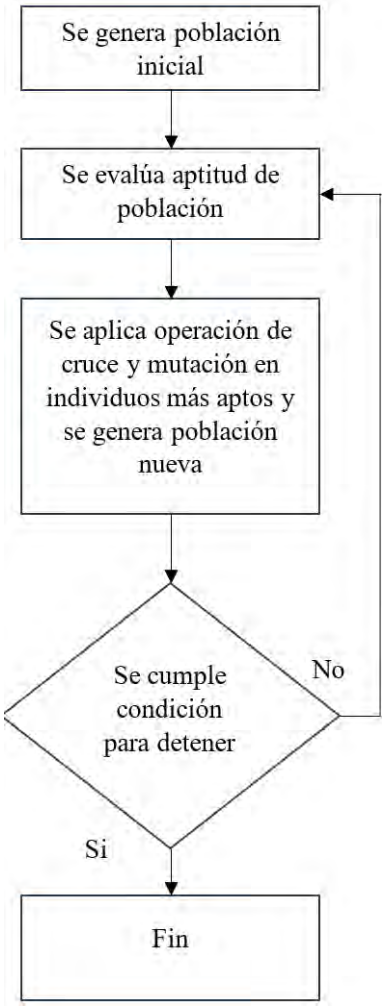

Figura 2.3 Diagrama general de proceso evolutivo.

La programación genética representa a los individuos mediante árboles sintácticos en donde las terminales son las variables de entrada y los nodos son funciones. En la Figura 2.4 se muestra una representación del árbol de una solución.

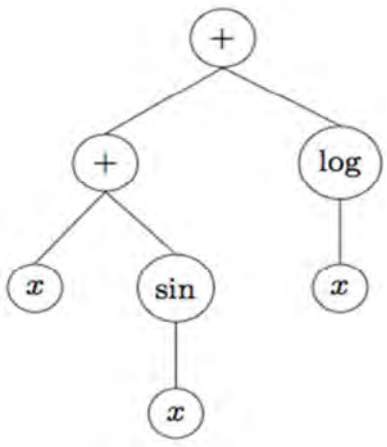

Figura 2.4 Representación de solución en programación genética.

Terminales

Las soluciones que se desarrollaron con la técnica de programación genética se representan en forma de árbol sintáctico de operaciones. Las terminales se definen en este caso como características con las que se harán las operaciones.

Funciones

Las soluciones desarrolladas por la técnica de programación genética se definen como un árbol sintáctico en el que los nodos intermedios son funciones matemáticas y los nodos terminales son las características antes mencionadas. Las funciones debe definirse de acuerdo al problema que se esta resolviendo, en algunos casos se defininen como operaciones matemáticas (suma, resta, etc), en otras pueden ser operaciones especiales dentro del dominio del problema.

Operación de cruce

La operación de cruce que se realiza sobre los árboles consiste en seleccionar un punto en 2 árboles e intercambiar sus ramas para asi generar dos nuevas soluciones como se puede observar en la Figura 2.5.

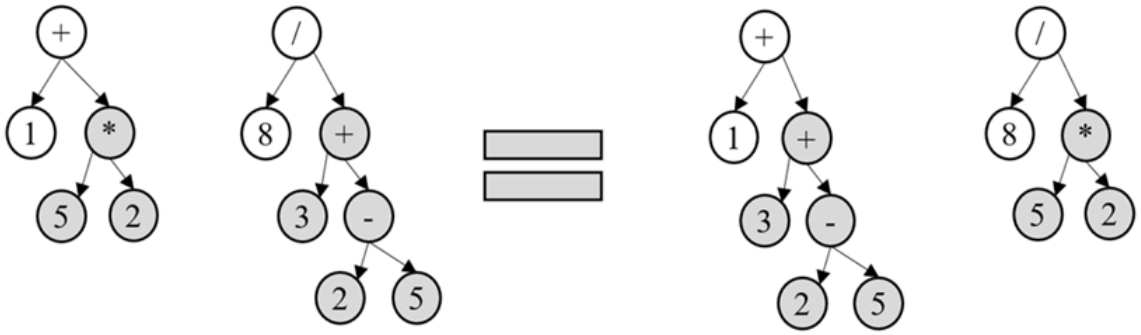

Figura 2.5 Operación de cruce.

Operación de mutación

La operación de mutación se realiza seleccionando un nodo al azar en donde se cambia la función generando asi una nueva solución, se muestra en la Figura 2.6.

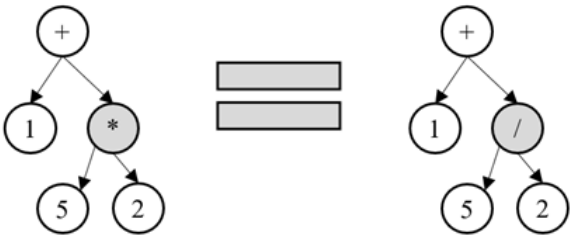

Figura 2.6 Operación de mutación.

Función de aptitud

Función de aptitud: Los individuos con mayor aptitud son seleccionados para aplicar las operaciones de cruce y mutación. Esta selección se basa en el valor asignado a cada individuo. La función de aptitud debe elegirse cuidadosamente según las características del problema que se busca resolver.

2.5. Redes neuronales

Una red neuronal artificial (RNA) es un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano (Fausett, 1994). Consiste en un conjunto de nodos o neuronas artificiales (Figura 2.7), organizadas en capas que procesan información mediante conexiones ponderadas. Estas redes son utilizadas para resolver problemas complejos de

aprendizaje automático, como clasificación, regresión, reconocimiento de patrones y toma de decisiones.

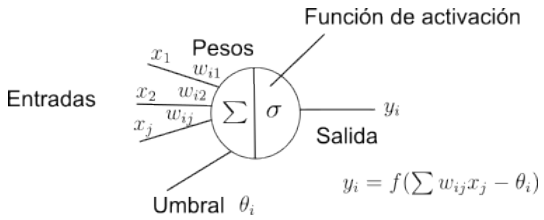

Figura 2.7 Ejemplo de de neurona artificial.

Las redes neuronales artificiales están compuestas por tres tipos principales de capas:

- Capa de entrada: recibe los datos de entrada y los transmite a las capas ocultas.
- Capas ocultas: realizan cálculos y transformaciones mediante funciones de activación, ajustando los pesos de las conexiones en función de un algoritmo de aprendizaje.
- Capa de salida: genera la salida final del modelo, que puede ser una predicción o clasificación.

La funcion de activación es una función matemática que se aplica a la salida de cada neurona para introducir no linealidad en el modelo. Su propósito es permitir que la red neuronal aprenda representaciones más complejas y pueda capturar patrones no lineales en los datos. Estas funciones transforman la suma ponderada de las entradas de la neurona en una salida que luego se transmite a la siguiente capa de la red.

Algunas funciones de activacion son:

- Sigmoide: su fórmula es $\sigma(x) = \frac{1}{1+e^{-x}}$ tiene un rango de salida entre $0,1$. Se usa principalmente en problemas de clasificación binaria.
- Tangente Hiperbólica: su fórmula es $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ tiene un rango de salida de $(-1,1)$. Es similar a la sigmoide, pero con una salida centrada en cero, lo que puede mejorar la convergencia.
- Rectified Linear Unit (ReLU): su fórmula es $f(x) = max(0,x)$ tiene un rango de salida de $[0,\infty)$. Es la más utilizada en redes profundas debido a su eficiencia computacional.

Una red neuronal artificial (ANN), ante un conjunto de datos de entrada, su propósito es que la aplicación genere las salidas esperadas. Este proceso implica el uso de distintos valores de entrada con el fin de ajustar los pesos de las conexiones entre neuronas siguiendo un método preestablecido. A lo largo del entrenamiento, estos pesos se van modificando progresivamente hasta alcanzar valores óptimos que permitan que cada entrada produzca la salida deseada.

Capítulo 3

Modelo de detección de emociones utilizando el ritmo cardíaco bajo estímulos audio visuales y la voz

Este trabajo muestra la construcción de modelos capaces de identificar emociones a partir del ritmo cardíaco y voz. Las emociones se tratan de inducir mediante estímulos audio visuales en un experimento controlado. La metodología seguida se estructura en tres fases principales que muestran desde la recopilación de datos hasta la validación del desempeño de los modelos propuestos.

3.1. Metodología de solución

La metodología para desarrollar un modelo de detección de emociones se divide en tres fases que se muestran en la Figura 3.1. La primera fase llamada "Desarrollo de conjuntos de datos para detección de emociones", muestra los pasos para generar los datos que seran utilizados para el entrenamiento de los modelos. La segunda fase denominada "Desarrollo de modelos para detección de emociones", muestra los algoritmos de aprendizaje de máquina, redes neuronales y programación genética utilizados para el entrenamiento de los modelos de clasificación de emociones. La tercer fase llamada "Pruebas y resultados obtenidos" presenta los resultados obtenidos despues de probar los modelos en los conjuntos de datos de prueba para detección de emociones con la voz y ritmo cardíaco.

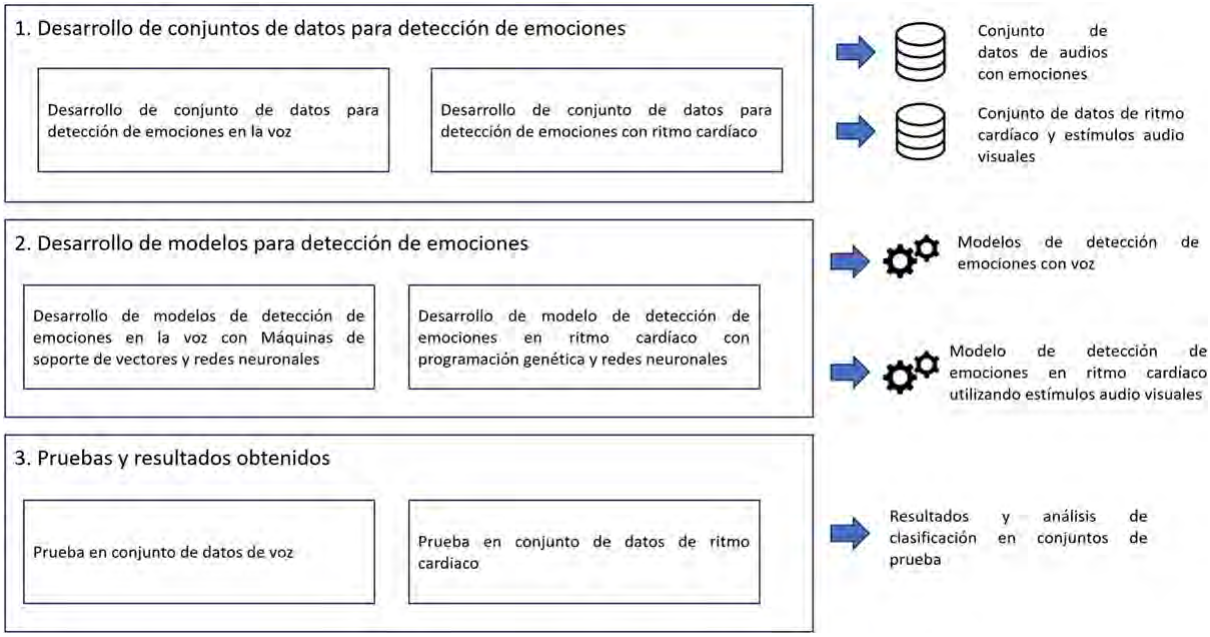

Figura 3.1 Metodología de solución para el desarrollo del modelo de detección de emociones.

3.2. Desarrollo de conjuntos de datos para detección de emociones

En esta sección se desarrollan dos conjuntos de datos para la detección de emociones; el primer conjunto se construye a partir de archivos de audio grabados por actores profesionales etiquetados con una emoción; el segundo conjunto de datos se desarrolla a partir de estímulos audiovisuales utilizados en un experimento de emociones y contiene información del ritmo cardíaco de las personas participantes en dicho experimento.

Un conjunto de datos es una colección estructurada de información utilizada para entrenar, validar y probar modelos de aprendizaje de máquina. Está compuesto por muestras (también llamadas instancias o observaciones), donde cada muestra tiene múltiples características (también llamadas atributos o features) que describen el fenómeno que se desea modelar.

3.2.1. Desarrollo de conjunto de datos para modelar la detección de emociones en la voz

El desarrollo de conjunto de datos para modelo de detección de emociones en la voz se presenta en esta sección. El proceso utilizado para crear el conjunto de datos a partir de los archivos de audio se muestra en la Figura 3.2.

Los audios utilizados para construir el conjunto de datos para el modelo de detección de emociones en la voz se obtuvieron del conjunto de datos RAVDESS que contiene un total de 1440 archivos de audio y están etiquetados con las emociones de calma, felicidad, tristeza, enojo, miedo, sorpresa, y disgusto. Las grabaciones fueron realizadas con 24 actores (12 hombres y 12 mujeres). El objetivo de este proceso es generar el conjunto de características a partir de los archivos de audio de RAVDESS. La técnica de validación cruzada se empleó con el objetivo de identificar el modelo con mejor capacidad de generalización, es decir, aquel que logra un buen desempeño al clasificar datos no vistos durante el entrenamiento. En este proceso también se estandarizan los valores de los conjuntos de datos a una escala consistente, de modo que las variables con valores grandes no afecten de manera desproporcionada el entrenamiento del modelo.

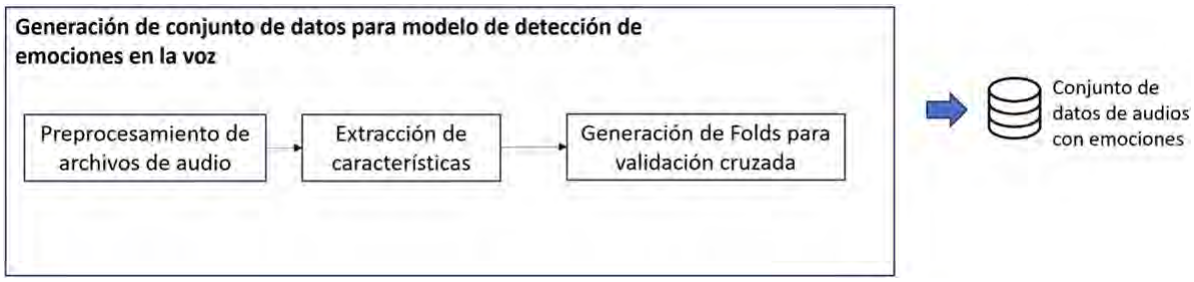

Figura 3.2 Proceso de desarrollo de conjunto de datos.

Preprocesamiento de archivos de audio

El preprocesamiento de los archivos de audio es una etapa importante en la generación de los datos que seran utilizados para el entrenamiento del modelo. El preprocesamiento ayuda a resultar partes importantes en los datos y tambien información innecesaria para el modelo. La técnica utilizada para preprocesar los archivos de audio consistió en eliminar los segmentos de silencio al inicio y al final de los archivos de audio. Si bien el silencio a veces puede contener información útil para la detección de emociones, en este caso, los archivos de audio contienen silencios al comienzo y al final que no se espera que contribuyan al conjunto de características. Por lo tanto, aplicamos una técnica para eliminar los valores de la señal de audio que estén por debajo de un umbral de 30 dB. Un ejemplo de eliminación de silencio en una señal de audio se muestra en la Figura 3.3.

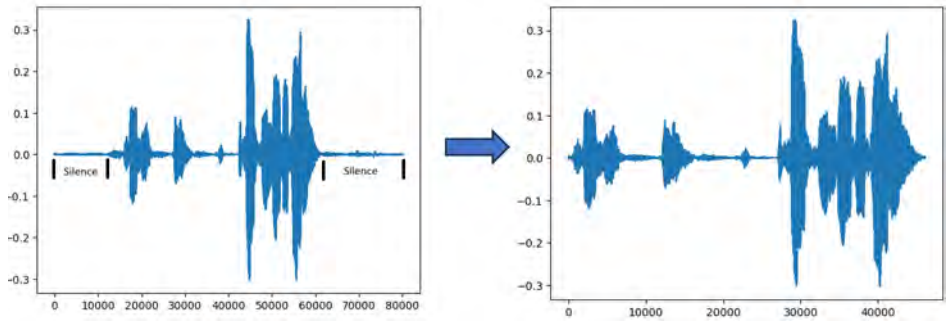

Figura 3.3 Proceso de eliminación de silencio de un archivo de audio.

Extracción de características

La extracción de características es el proceso de convertir archivos de audio en datos para que los modelos puedan ser entrenados. Las características se obtienen de los archivos de audio preprocesados dividiendolos en fragmentos más pequeños. Los fragmentos de audio se procesan mediante funciones que calculan datos tanto en el dominio del tiempo como en el dominio de la frecuencia, utilizando funciones del paquete libROSA versión 0.10 (McFee y cols., s.f.) y la transformada wavelet discreta.

La señal de audio se divide en segmentos o marcos de un tamaño predefinido en milisegundos. El enmarcado permite capturar la información temporal de la señal, donde pueden ocurrir cambios en el tono y otras características. Sin embargo, dividir la señal en cuadros también puede omitir información importante; por lo tanto, la división se realiza superponiendo una parte del cuadro anterior en milisegundos. La porción de información compartida entre cuadros se denomina solapamiento y también se define en milisegundos. De este modo, cada cuadro, a pesar de tener un tamaño fijo, no contiene información aislada de una parte de la señal, como se muestra en la Figura 3.4. Los cuadros obtenidos se formaron dividiendo la señal cada 25 milisegundos con un solapamiento de 15 milisegundos.

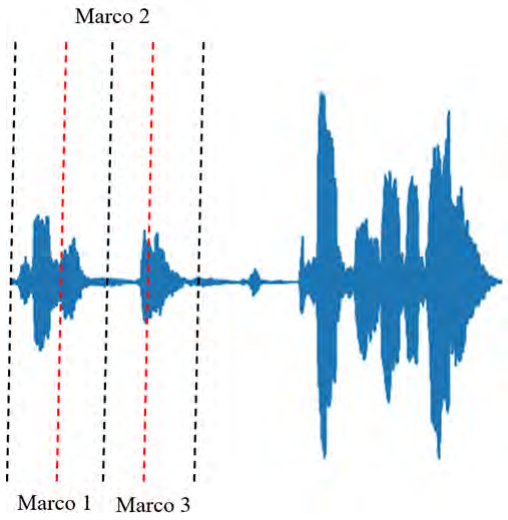

Figura 3.4 Proceso de enmarcado de un archivo de audio.

La extracción de características a partir de las señales obtenidas mediante los marcos puede resultar en la pérdida de información debido a la discontinuidad de la señal en los bordes de cada marco. La pérdida de información durante la aplicación puede minimizarse mediante el uso de una función de ventana matemática en cada marco. En este proyecto, se utilizó la función de ventana de Hamming (Blackman y Tukey, s.f.) en los marcos obtenidos en el paso anterior, como se muestra en la Figura 3.5.

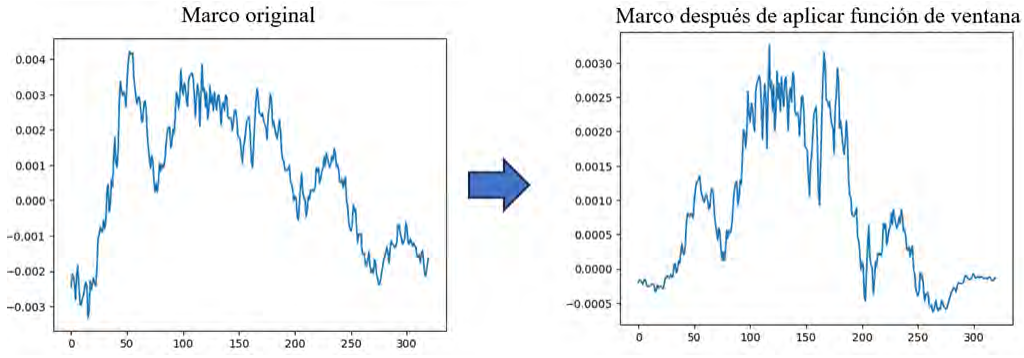

Figura 3.5 Proceso de ventaneo sobre un marco de archivo de audio.

El cálculo de las características se realizó utilizando los marcos de cada archivo de audio y luego aplicando las funciones del paquete libROSA. Los resultados obtenidos de cada marco se usaron para calcular la media y la desviación estándar de las características de la señal de audio. La Figura 3.6 ilustra el proceso descrito anteriormente.

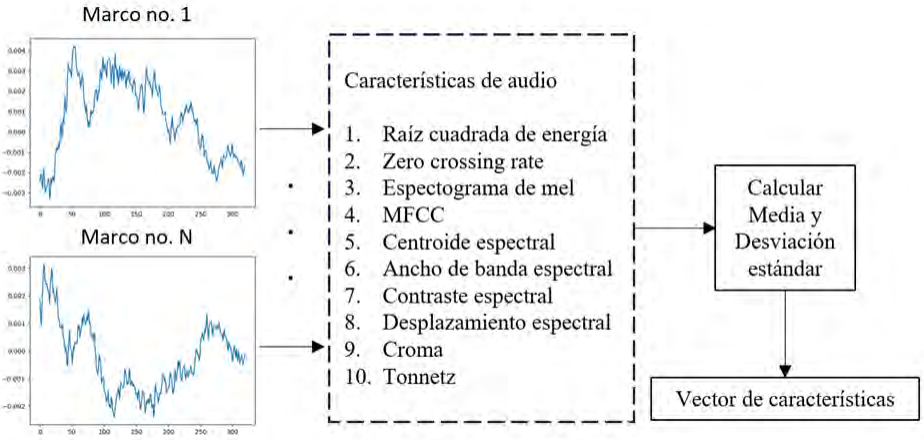

Figura 3.6 Proceso de extracción de características de audio.

A continuación se definen las características extraidas de cada marco de la señal se audio.

1. La característica de Energía Cuadrática Media (RMSE, por sus siglas en inglés) en el caso de una señal de audio corresponde aproximadamente a la intensidad de la señal en términos de volumen.

2. La Tasa de Cruces por Cero (Zero crossing rate en inglés) es la característica que indica el número de veces que la señal cruza el eje horizontal en cero y puede utilizarse para la segmentación de palabras y silencios.

3. El Espectrograma Mel, utilizando la escala Mel, relaciona la frecuencia percibida o tono de un sonido puro con su frecuencia medida real. Esta representación gráfica se usa porque toma en cuenta la percepción auditiva humana.

4. Los Coeficientes Cepstrales en la Frecuencia Mel (MFCCs, por sus siglas en inglés) son coeficientes utilizados para la representación del habla basados en la percepción auditiva humana, derivados del Espectrograma Mel.

5. El centroide espectral indica la frecuencia alrededor de la cual se concentra la energía de un espectro como una media ponderada y se usa para extraer características relacionadas con la prosodia y la expresión emocional en el habla.

6. El ancho de banda espectral se refiere a la extensión o rango de frecuencias abarcadas por una señal en el dominio de la frecuencia. Puede utilizarse para identificar la calidad en los tonos de un archivo de audio y detectar cambios en el contenido de una señal.

7. El contraste espectral es una medida que evalúa la diferencia en energía entre regiones de frecuencia específicas en el espectrograma de una señal de audio y puede utilizarse para identificar patrones en el habla.

8. El desplazamiento espectral es una medida que indica la tendencia central de las frecuencias presentes en una señal de audio y puede utilizarse para identificar cambios en el contenido sonoro de una señal a lo largo del tiempo.

9. Chroma es la representación cromática de una señal de audio que se centra en la información relacionada con la tonalidad y la relación entre notas musicales.

10. Tonnetz se utiliza en el análisis musical y el procesamiento de audio para representar la información tonal de una señal; ayuda a caracterizar las relaciones entre notas musicales.

Las características wavelet se basan en el uso de funciones wavelet para analizar señales de audio. Las wavelets son funciones matemáticas que pueden descomponer una señal en diferentes escalas y resoluciones. Estas características son útiles para capturar patrones en distintas escalas de tiempo y frecuencia en una señal de audio, permitiendo la detección de cambios rápidos y lentos en la señal.

La transformada wavelet discreta (DWT, por sus siglas en inglés) es una técnica para el análisis de señales. Las wavelets tienen la ventaja de capturar información tanto en el dominio del tiempo como en el dominio de la frecuencia, una ventaja sobre la transformada de Fourier. A diferencia de la transformada de Fourier, la transformada wavelet proporciona alta resolución temporal y baja resolución en frecuencia para frecuencias altas, así como alta resolución en frecuencia y baja resolución temporal para frecuencias bajas.

En este sentido, es similar al oído humano, que presenta características similares de resolución en el tiempo y la frecuencia. La transformada wavelet discreta proporciona una representación compacta de una señal en ambos dominios, donde una señal $x[k]$ se filtra utilizando filtros de paso bajo y paso alto, separando la señal en componentes de baja y alta frecuencia. La transformada wavelet discreta se puede definir como:

$$D(j,k) = 2^{\frac{-j}{2}} \sum x(i)\psi * (2^{-j}i - k) \tag{3.1}$$

Donde $\psi(t)$ es una función temporal con energía finita y decaimiento rápido, denominada wavelet madre. El análisis de la transformada wavelet discreta puede realizarse mediante un algoritmo piramidal asociado con bancos de filtros. En el algoritmo, la señal se analiza en diferentes bandas de frecuencia y resoluciones, descomponiéndola en información aproximada e información detallada. La información aproximada se descompone aún más utilizando el mismo procedimiento. Esta información se obtiene aplicando sucesivamente filtros de paso alto y paso bajo a la señal en el dominio del tiempo, lo cual puede definirse mediante las siguientes ecuaciones:

$$Y_{high}[k] = \sum_n x(n)g[2k - n] \tag{3.2}$$

$$Y_{low}[k] = \sum_{n} x(n)h[2k-n] \tag{3.3}$$

Donde $Y_{high}[k]$ y $Y_{low}[k]$ son las salidas de los filtros de paso alto (g) y paso bajo (h), respectivamente, después de realizar el submuestreo de la señal por un factor de 2. Los archivos de audio del conjunto de datos fueron sometidos a la función de transformada wavelet discreta db8 hasta el nivel 8 para obtener 8 coeficientes de aproximación, los cuales consisten en vectores de la señal que representan las altas frecuencias, y 1 coeficiente de detalle, que es un vector que representa las bajas frecuencias, con el fin de extraer sus características, como se muestra en la Figura 3.7.

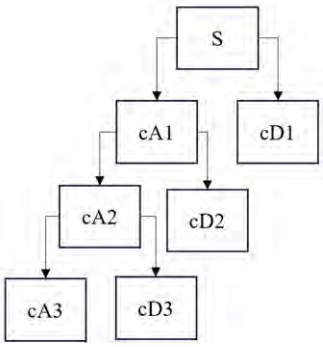

Figura 3.7 Descomposición de tranformada de wavelet discreta.

Los coeficientes obtenidos a través de la función wavelet se utilizaron para calcular las siguientes características estadísticas, como se muestra en la Figura 3.8. Para cada coeficiente, se calcula lo siguiente:

- La media del valor absoluto
- La moda
- La mediana
- La desviación estándar
- La varianza
- La curtosis
- La asimetría
- La energía

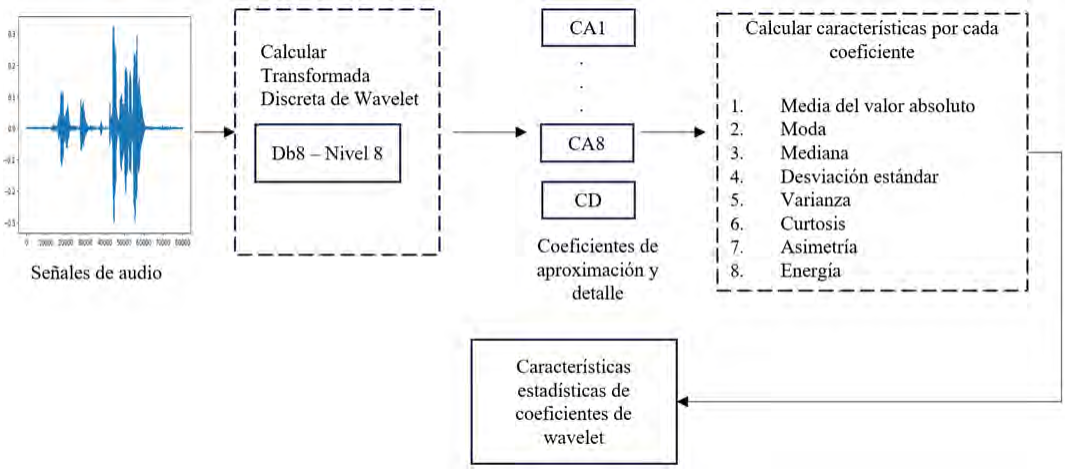

Figura 3.8 Extracción de características estadísticas de wavelet.

El conjunto de características obtenido se formó al combinar las características de audio y las estadísticas de wavelet de cada archivo. El conjunto de características tiene un total de 1,440 filas y 428 columnas o variables.

Generación de particiones para validación cruzada

La división del conjunto de datos de características es una técnica que consiste en separar el conjunto de datos en varias partes con el objetivo de entrenar y validar el modelo que se está construyendo. La división se realiza utilizando la técnica de validación cruzada y, posteriormente, se aplica una función de estandarización de datos a cada partición para garantizar que las características tengan una escala uniforme, mejorando así el rendimiento y la precisión del modelo durante el proceso de entrenamiento y validación. La validación cruzada consiste en dividir el conjunto de datos en k partes. Las partes resultantes utilizan un total de $k-1$ para entrenar el modelo y 1 para la validación. La parte de validación se alterna para entrenar el modelo con las k partes. El resultado se obtiene promediando la evaluación del modelo a lo largo de las k partes. La Figura 3.9 muestra un ejemplo de la división en 5 partes ($k = 5$).

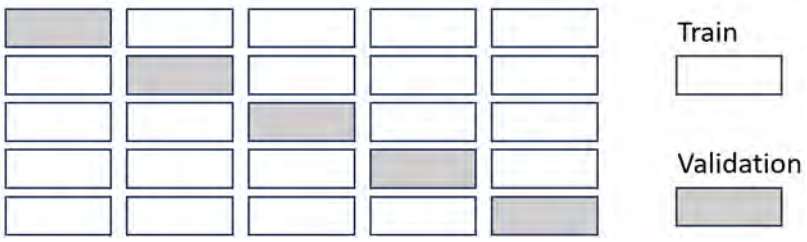

Figura 3.9 Ejemplo de validación cruzada con 5 particiones.

El conjunto de datos de características de tamaño 1440 x 428 se dividió en 5 partes utilizando la función StratifiedKFold del paquete de Python scikit-learn versión 1.2.2, empleando el parámetro stratify para indicar que la división debe mantener la misma cantidad de datos para cada clase. La Tabla 3.1 muestra un extracto del conjunto de datos.

Tabla 3.1 Ejemplo de conjunto de datos de características.

	1_c0_mean	**2_c0_mode**	**3_c0_median**	**...**	**428_tonnetz_5_std**
1	0.03260697	0.994764255	0.343952065	...	0.060766204
2	0.031042535	0.993540307	0.349692219	...	0.095447981
3	0.041163097	0.994547902	0.349505242	...	0.089243978
4	0.025004974	0.996293042	0.351764576	...	0.087259648
5	0.03627768	0.99522758	0.348607574	...	0.083980832
6	0.054186207	0.992952848	0.367127011	...	0.121817398
7	0.056793051	0.99176338	0.323824959	...	0.099802614
8	0.036685049	0.993271052	0.335927114	...	0.100369336
9	0.03131778	0.994909573	0.347986156	...	0.110979818
10	0.041359181	0.992372333	0.345193904	...	0.113798574
11	0.013856337	0.997056363	0.344266154	...	0.134094245
12	0.024565761	0.995189912	0.363523663	...	0.107523496
13	0.007212581	0.995810414	0.34449072	...	0.2382349
14	0.013813184	0.989054693	0.351012158	...	0.218710507
15	0.00941033	0.996556064	0.346752746	...	0.193657683
16	0.007900601	0.997387807	0.350739522	...	0.20664556
17	0.020613458	0.995190859	0.347693878	...	0.070204522
18	0.032944678	0.994875843	0.346705087	...	0.069820654
19	0.029466183	0.995789873	0.354694826	...	0.074418909
20	0.025535562	0.994442672	0.347978531	...	0.067303884
21	0.014910281	0.995401179	0.348866583	...	0.116524966
22	0.01541462	0.995792206	0.339884358	...	0.077745228
23	0.018334393	0.993447942	0.350599581	...	0.121271439
24	0.029407396	0.983693169	0.353022807	...	0.186772099
25	0.015497266	0.995263396	0.352209603	...	0.155462614
26	0.020343996	0.99582568	0.341054782	...	0.179589828
	...	...	...	...	...
1428	0.004462466	0.997577257	0.346222381	...	0.80543409
1429	0.001560001	0.999249675	0.346999469	...	0.315858926
1430	0.003482282	0.998797402	0.345853474	...	0.328083471
1431	0.001699915	0.999230282	0.346719489	...	0.363052002
1432	0.002855875	0.998755625	0.347075995	...	0.225514065
1433	0.004083173	0.997826276	0.347118658	...	0.459133997
1434	0.006516211	0.990872323	0.348477711	...	0.366262323
1435	0.002246214	0.998988774	0.346762212	...	0.272684906
1436	0.004372704	0.998595318	0.342049993	...	0.309726783
1437	0.005294366	0.997037202	0.346656299	...	0.322107398
1438	0.007585385	0.997422398	0.349771089	...	0.42345583
1439	0.017806088	0.990279002	0.34782234	...	0.262172329
1440	0.021480996	0.966654615	0.354002547	...	0.234965921

La estandarización es un proceso utilizado para transformar un conjunto de datos de manera que tenga una escala común. La estandarización ayuda a que los algoritmos tengan un mejor rendimiento, ya que evita que valores muy grandes o muy pequeños afecten la identificación de patrones en los datos. La estandarización se realizó utilizando la función MinMaxScaler del paquete de Python scikit-learn. La función MinMaxScaler utiliza la Ecuación 3.4 para escalar los datos a un rango específico, típicamente entre 0 y 1. La estandarización se aplica a todas las variables que forman el conjunto de datos. La Tabla 3.2 muestra un ejemplo de una variable con datos estandarizados.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.4}$$

Tabla 3.2 Ejemplo de valores originales y estandarizados.

c0_mean	**c0_mean**
0.00779002	0.03260697
0.00742269	0.03104253
0.00979901	0.0411631
0.01232359	0.02500497
0.00600506	0.03627768
0.00865191	0.05418621

3.2.2. Desarrollo de conjunto de datos para detección de emociones con ritmo cardíaco

El conjunto de datos para el modelo de detección de emociones con ritmo cardíaco como se muestra en la Figura 3.10 se generó a partir de la información obtenida mediante un experimento controlado de emociones.

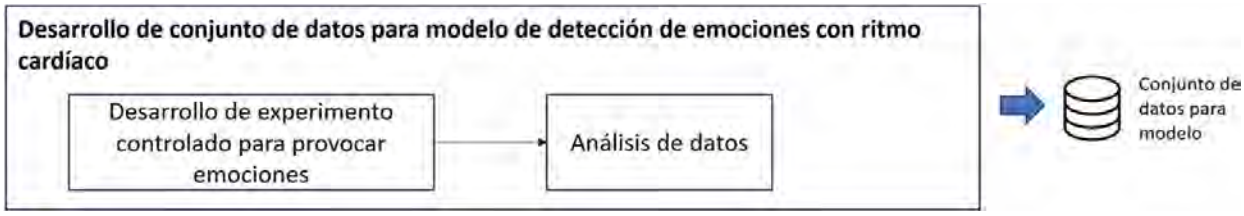

Figura 3.10 Proceso de desarrollo de conjunto de datos para modelo de detección de emociones.

Desarrollo de estímulos audio visuales

El desarrollo de estímulos audiovisuales requiere de imágenes y sonidos que provoquen emociones. Los estímulos audiovisuales se construyen forma de video en el que se presentan imágenes por un tiempo de 10 segundos por cada imagen para la emoción de calma se muestran imágenes neutrales de flores, paisajes, la emoción de enojo muestra imágenes de situaciones de injusticias y maltrato de personas, la emoción de felicidad muestra personas sonriendo y situaciones de personas felices, la emoción de tristiza muestra imágenes de personas en duelo y llanto. El audio del video esta compuesto por sonidos molestos como zumbidos, gritos y llanto en el caso del video para la emoción de enojo y en las emociones de calma, felicidad y tristeza se utilizan canciones acordes a cada una de esas emociones. Las emociones de una persona se ven afectadas por el entorno en el que se desenvuelve. En esta sección se presenta el proceso (Figura 3.11) para desarrollar estímulos que pueden ser utilizados para provocar emociones en un experimento.

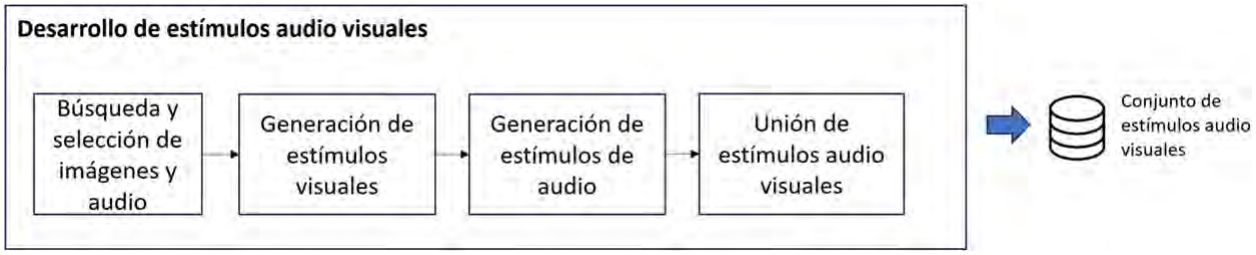

Figura 3.11 Proceso de desarrollo de estimulos audio visuales.

Búsqueda y selección de imágenes y audio

Los estímulos audiovisuales utilizados en este trabajo tienen la finalidad de inducir en los participantes las emociones de: calma, enojo, tristeza y felicidad. Los estímulos fueron seleccionados de un conjunto de imágenes, sonidos y música que fueron desarrollados con este propósito. Los archivos de audio e imágenes utilizados fueron agrupados de acuerdo a los valores de valencia y activación según el modelo de emociones de Russel (Russell, s.f.). La valencia representa el grado de placer de una emoción y la activación se refiere a la intensidad. El modelo de emociones de Russel puede dividirse en 4 secciones de acuerdo a las dimensiones de valencia y activación: Valencia Baja - Activación Alta (VBAA), Valencia Alta - Activación Alta (VAAA), Valencia Baja - Activación Baja (VBAB) y Valencia Alta - Activación Baja (VAAB). En cada uno de las cuatro secciones definidas se encuentran las emociones de calma, enojo, tristeza y felicidad; como se observa en la Figura 3.12. Las imágenes y los audios que se prepararon para inducir las emociones en los participantes se ubicaron en el modelo de emociones en cada una de estas secciones. Los estímulos utilizados son videos compuestos por las imágenes, sonidos y música seleccionada para cada emoción.

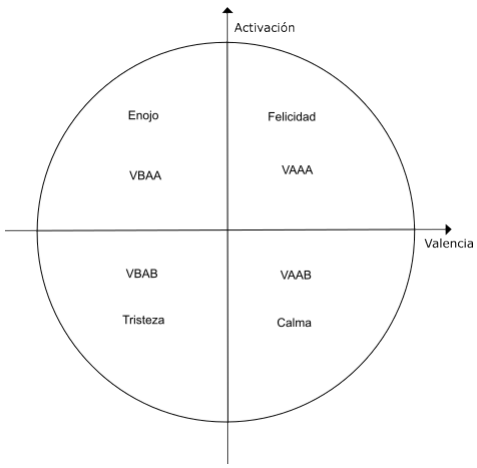

Figura 3.12 Zonas y emociones en modelo circunflejo.

Generación de estímulos visuales

El conjunto de datos “International affective picture system"(IAPS) (Lang, Bradley, y Cuthbert, s.f.) es un conjunto validado para la provocación de emociones mediante imágenes y contiene un total de 1183 imágenes. La información contenida en el conjunto de datos además de las imágenes es el valor de cada una de ellas en términos de un modelo afectivo de 3 dimensiones: placer (valencia), intensidad (activación) y dominio de la emoción. Algunos ejempos de las imágenes seleccionadas del conjunto de datos para la emoción de calma se pueden ver en la Tabla 3.3, la Tabla 3.4 para la emoción de enojo, la Tabla 3.5 para la emoción de tristeza y la Tabla 3.6 para la emoción de felicidad donde muestra la descripción de la imagen, el número de imagen, la media de la valencia, la media de la activación.

Tabla 3.3 Ejemplo de imágenes IAPS para emoción de calma. Se muestra la descripción, el ID del conjunto IAPS y la valencia y activación promedio.

Desc	IAPS ID	Val prom	Act prom	Desc	IAPS ID	Val prom	Act prom
Gannet	1450.0	6.37	2.83	Nature	5201.0	7.06	3.83
GirlMakeup	2308.0	5.22	3.82	Garden	5202.0	7.25	3.73
Girl	2320.0	6.17	2.9	Seaside	5210.0	8.03	4.6
Woman	2374.0	6.29	3.86	Nature	5220.0	7.01	3.91
ManW/Fish	2392.0	6.15	3.85	Nature	5250.0	6.08	3.64
Medicalworker	2394.0	5.76	3.89	Galaxy	5300.0	6.91	4.36
Musician	2488.0	5.73	3.91	Boat	5390.0	5.59	2.88
Harvest	2515.0	6.09	3.8	Violinist	5410.0	6.11	3.29
Bakers	2579.0	5.53	3.85	Mushroom	5530.0	5.38	2.87
Chess	2580.0	5.71	2.79	Sky	5593.0	6.47	3.98
City	2594.0	6.05	3.84	Mountains	5631.0	7.29	3.86
Woman	2620.0	5.93	2.72	Cave	5661.0	5.96	4.15
Balloons	2791.0	6.64	1.7	Field	5711.0	6.62	3.03
Teenager	2870.0	5.31	3.01	Farmland	5720.0	6.31	2.79
Couple	4605.0	5.59	3.84	Field	5725.0	7.09	3.55
Sunflower	5001.0	7.16	3.79	Grain	5726.0	6.23	2.84
Flower	5010.0	7.14	3.0	Flowers	5731.0	5.39	2.74
Flower	5020.0	6.32	2.63	Nature	5760.0	8.05	3.22
Flower	5030.0	6.51	2.74	Nature	5780.0	7.52	3.75
Garden	5199.0	6.93	4.7	Lake	5781.0	7.13	3.82
Flowers	5200.0	7.36	3.2	Flowers	5811.0	7.23	3.3
Mountain	5814.0	7.15	4.82	Mountains	5820.0	7.33	4.61
Seagulls	5831.0	7.63	4.43	Beach	5836.0	7.25	4.28
Clouds	5870.0	6.78	3.1	Candlestick	7053.0	5.22	2.95
FireHydrant	7100.0	5.24	2.89	Bus	7140.0	5.5	2.92
Clock	7190.0	5.55	3.84	AbstractArt	7237.0	5.43	3.88
Building	7242.0	5.28	3.83	Fruit	7283.0	5.5	3.81
Tomatoes	7285.0	5.67	3.83	Ferrry	7489.0	6.54	4.49
Window	7490.0	5.52	2.42	Store	7495.0	5.9	3.82
House	7530.0	6.71	4.0	Violin	7900.0	6.5	2.6

Las imágenes que se utilizaron fueron seleccionadas de acuerdo a los valores de valencia y activación promedio. El modelo de emociones se aplicó calculando la mediana de estos valores para encontrar el punto medio. Los valores de la mediana de la valencia (5.22) y activación (4.86) se tomaron como referencia para ubicar las imágenes en los cuatro

cuadrantes del modelo de emociones descrito anteriormente. La selección de las imágenes para construir los videos se realizó tomando en cuenta su contenido y su ubicación en el modelo de acuerdo a las emociones de calma($valprom >= 5.22, actprom <= 4.86$), enojo ($valprom <= 5.22, actprom >= 4.86$), tristeza($valprom <= 5.22, actprom <= 4.86$) y felicidad($valprom >= 5.22, actprom >= 4.86$). La distribución de las imágenes puede verse en la Figura 3.13.

Tabla 3.4 Ejemplo de imágenes IAPS para emoción de enojo. Se muestra la descripción, el ID del conjunto IAPS y la valencia y activación promedio.

Desc	**IAPS ID**	**Val prom**	**Act prom**	**Desc**	**IAPS ID**	**Val prom**	**Act prom**
MaleFace	2220.0	5.03	4.93	StarvingChild	9040.0	1.67	5.82
Hunters	2688.0	2.73	5.98	Matador	9150.0	4.54	5.31
Riot	2691.0	3.04	5.85	Soldiers	9163.0	2.1	6.53
DrunkDriving	2751.0	2.67	5.18	Garbage	9295.0	2.39	5.11
Harassment	4621.0	3.19	4.92	Mob	9402.0	4.48	5.07
Prison	6000.0	4.04	4.91	Handicapped	9415.0	2.82	4.91
BeatenFem	6315.0	2.31	6.38	Assault	9427.0	2.89	5.5
Attack	6350.0	1.9	7.29	Kids	9520.0	2.46	5.41
Attack	6360.0	2.23	6.33	DuckInOil	9560.0	2.12	5.5
Military	6825.0	2.81	5.36	Police	6840.0	3.63	5.95
Police	6838.0	2.45	5.8				

Tabla 3.5 Ejemplo de imágenes IAPS para emoción de tristeza. Se muestra la descripción, el ID del conjunto IAPS y la valencia y activación promedio.

Desc	IAPS ID	Val prom	Act prom	Desc	IAPS ID	Val prom	Act prom
Hospital	2205.0	1.95	4.53	ScaredChild	9041.0	2.98	4.64
SadFace	2230.0	4.53	4.13	Boy	9070.0	5.01	3.63
CryingFamily	2456.0	2.84	4.55	Fisher	9171.0	4.01	4.72
ElderlyMan	2520.0	4.13	4.22	HungMan	9265.0	2.6	4.34
Woman	2700.0	3.19	4.77	Smoke	9280.0	2.8	4.26
Alcoholic	2752.0	4.07	4.3	Garbage	9290.0	2.88	4.4
Boy	2795.0	3.92	4.7	BurntBldg	9471.0	3.16	4.48
Gun	2811.0	2.17	6.9	SickKitty	9561.0	2.68	4.79
CryingBoy	2900.0	2.45	5.09	Flood	9926.0	3.85	4.83
DisabledChild	3300.0	2.74	4.55	Hospital	7520.0	3.83	4.57
Memorial	9002.0	3.39	4.55				

Tabla 3.6 Ejemplo de imágenes IAPS para emoción de felicidad. Se muestra la descripción, el ID del conjunto IAPS y la valencia y activación promedio.

Desc	IAPS ID	Val prom	Act prom	Desc	IAPS ID	Val prom	Act prom
Gorilla	1659.0	6.57	4.89	Brownie	7200.0	7.63	4.87
Puppies	1710.0	8.34	5.41	Cupcakes	7405.0	7.38	6.28
Jellyfish	1908.0	5.28	4.88	Parachute	8163.0	7.14	6.53
Family	2340.0	8.03	4.9	Sailboat	8170.0	7.63	6.12
Romance	4603.0	7.1	4.89	Surfers	8206.0	6.43	6.41
Wedding	4628.0	7.23	5.19	Pilot	8300.0	7.02	6.14
Mountains	5600.0	7.57	5.19	TennisPlayer	8350.0	7.18	5.18
Flowers	5849.0	6.65	4.89	Gymnast	8470.0	7.74	6.14

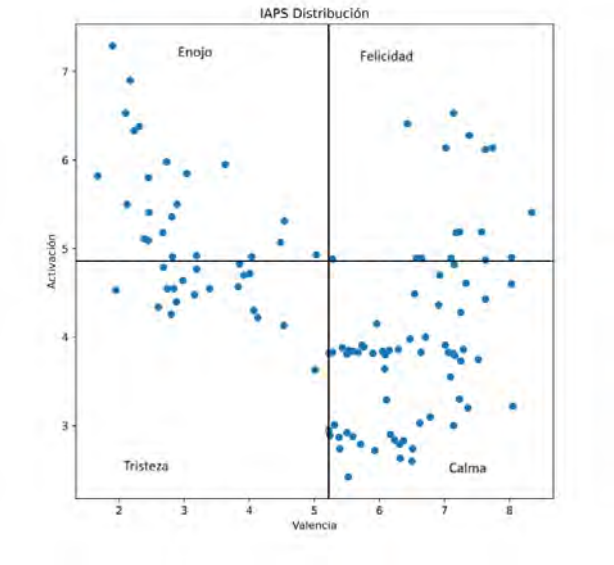

Figura 3.13 Distribución de imágenes en el modelo de emociones

Generación de estímulos de audio

La generación de los estímulos de audio se hizo combinando dos conjuntos de datos de audio para provocar emociones. El primer conjunto de datos “International Affective Digitized Sounds"(IADS) (Bradley y Lang, 2007) es un conjunto de datos validado para provocar emociones mediante audios y contiene un total de 167 archivos de audio de duración variable. Los datos asociados a emociones de este conjunto de datos se encuentran en términos de un modelo de 3 dimensiones de placer (valencia), intensidad (activación) y dominio de la emoción. Los valores que se observan en la Tabla 3.7 los elementos utilizados. Los sonidos de este conjunto de datos se utilizaron solo para crear el video de la emoción de enojo tomando como referencia la mediana de valencia ($valprom < 4.88$) y activación ($Actprom > 5.89$). La distribución de los sonidos puede verse en la Figura 3.14.

Tabla 3.7 Ejemplo de sonidos IADS para emoción de enojo. Se muestra el nombre del sonido, el número del conjunto IADS y la valencia y activación promedio.

Sonido	**Número**	**Val prom**	**Act prom**
Bees	115	2.16	7.03
Buzzing	116	3.02	6.51
BabyCry	261	2.75	6.51
Crowd1	310	3.89	6.78
Office2	319	3.56	6.08
JackHammer	380	3.7	6.33
CarHorns	420	2.34	7.08
EngineFailure	502	3.15	6.32
AirRaid	624	2.82	7.1
AlarmClock	709	2.78	7.54
Cuckoo	710	4.27	6.24
Siren1	711	2.61	7.39
Buzzer	712	2.42	7.98
Siren2	714	3.1	6.94
Alarm	715	4.3	6.99
DentistDrill	719	2.89	6.91
GlassBreak	730	3.22	6.23
Electricity	910	3.86	6.18

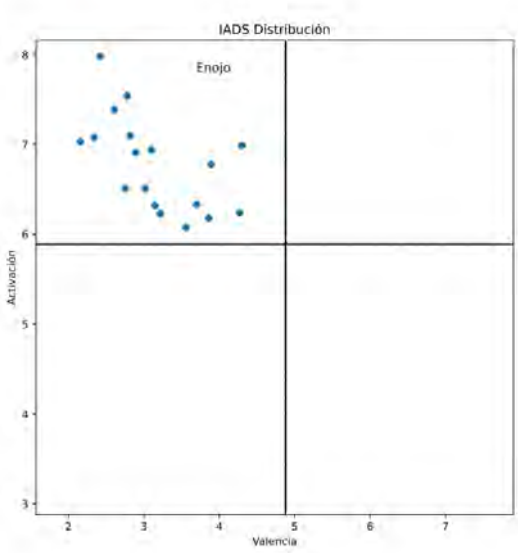

Figura 3.14 Distribución de sonidos de enojo en el modelo de emociones

El segundo es el conjunto de datos de música generado por el trabajo (Griffiths, Cunningham, Weinel, y Picking, 2021), este conjunto de datos consta de música organizada por géneros musicales. Los archivos seleccionados para cada emoción se muestran como ejemplos en las Tablas 3.8 y 3.9.

Tabla 3.8 Música utilizada para provocar emociones

ID	Canción	Artista
1	Oxygen Part 4	Jean Michel Jarre
2	Take Five	Dave Brubeck
3	King Tubby Meets Rockers Uptown	Augustus Pablo
4	Phenomenon	LL Cool J
5	The Model	Kraftwerk
6	9 to 5	Dolly Parton
7	Summertime	Louis Armstrong

Tabla 3.9 Música utilizada para provocar emociones

ID	Género	Emoción
1	Electronic	Calma
2	Jazz	Calma
3	Reggae	Calma
4	Hip Hop	Calma
5	Electronic	Calma
6	Country	Felicidad
7	Jazz	Tristeza

Unión de estímulos audio visuales

Las imágenes, sonidos y canciones seleccionados fueron utilizados para crear 5 videos para emoción de calma, 1 video de emoción de felicidad, 1 video de emoción de enojo y 1 video de emoción de tristeza. Los videos tienen diferente duración en función de la canción seleccionada. Las canciones en algunos videos fueron recortadas en su parte final para ajustarse a un tiempo aproximado de 3 minutos. Las imágenes en cada video se mostraron por un tiempo de 10 segundos cada una. La Tabla 3.10 muestra los videos y la duración de cada uno.

Tabla 3.10 Videos utilizados para provocar emociones

Video	Duración (min)
Calma 1	3:27
Calma 2	3:21
Calma 3	2:33
Calma 4	3:01
Calma 5	3:40
Enojo	3:40
Felicidad	2:33
Tristeza	3:31

Desarrollo de experimento controlado para provocar emociones

El experimento se diseñó en 4 procesos (Figura 3.15): (i) inducción a calma, (ii) inducción a enojo, (iii) inducción a tristeza y (iv) inducción a felicidad. Los procesos se realizaron en

2 partes que incluyen los siguientes videos mostrados en la Tabla 3.11. El primer video se utilizó para tener una linea base y después tener la posibilidad de comparar los datos con la emoción que se estaba induciendo.

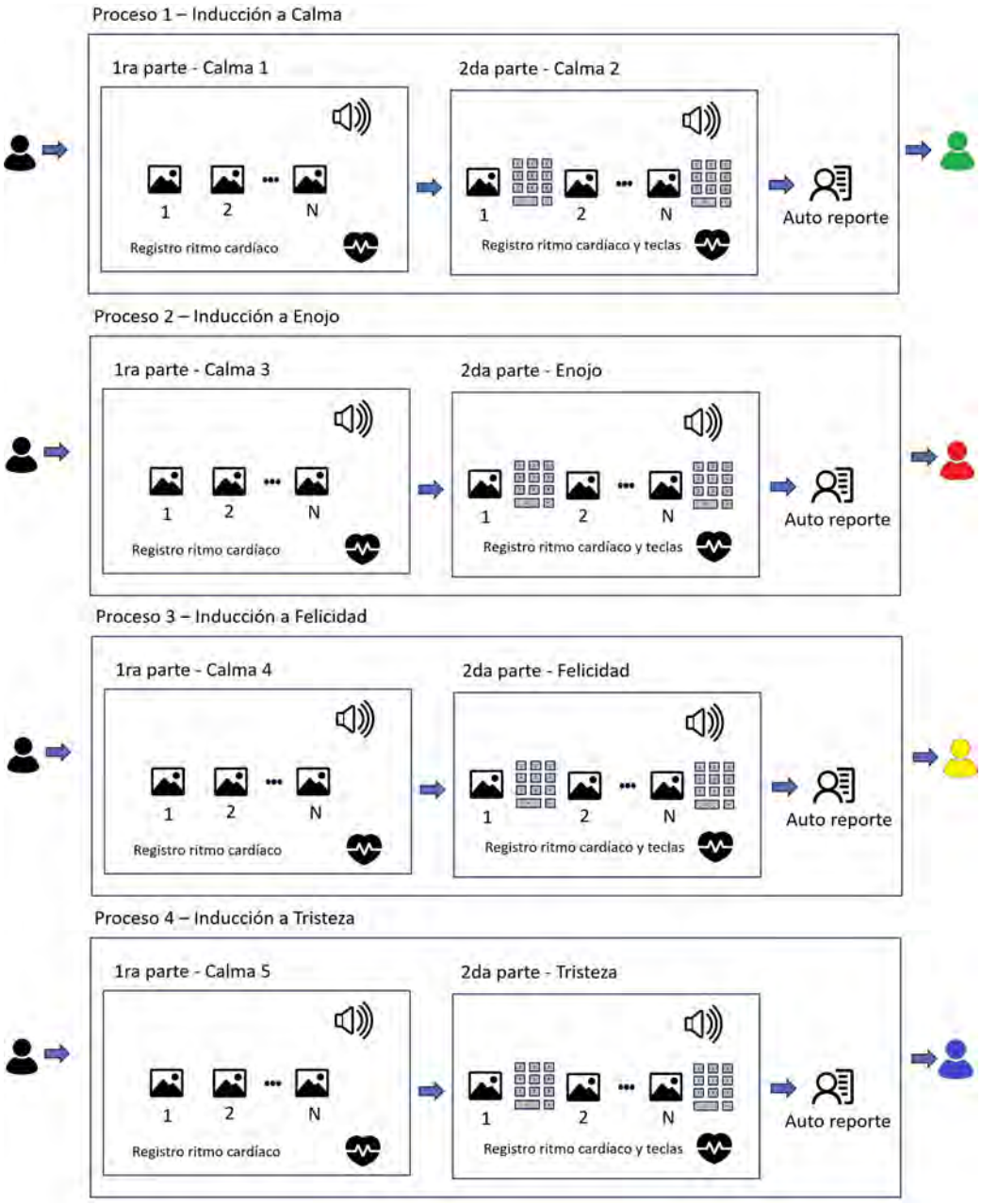

Figura 3.15 Procesos de experimento de emociones.

Tabla 3.11 Secuencia de videos de experimento

Video 1	**Video 2**
Calma 1	Calma 2
Clama 3	Enojo
Calma 4	Felicidad
Calma 5	Tristeza

Los participantes del experimento fueron estudiantes del Tecnológico Nacional de México campus Ensenada con una muestra (n=11) de 8 hombres y 3 mujeres en edades entre 21 y 39 años (edad media=24.55 desviación estándar=5.76). Las personas participaron de manera voluntaria, se les informó sobre el objetivo del experimento y las partes en que se realizaría.

Los materiales y equipo utilizado para la realización del experimento de emociones. El equipamiento utilizado en el experimento nos permitió registrar los datos fisiológicos de los participantes.

Los equipos utilizados fueron:

- Una computadora portátil y audífonos con cancelación de ruido.
- Un sensor OH1 para registrar el ritmo cardíaco.
- Imágenes IAPS e IADS para la creación de estímulos.
- Software psycopy versión 2022.2.5.
- Un teclado numérico.
- Carta de consentimiento para uso de los datos de los participantes.
- Autoreporte sobre emociones (Figura 3.16).

El experimento se llevó a cabo en un lugar privado, con el fin de evitar distracciones. Este espacio permitía controlar la iluminación mediante apagadores de luz y cortinas en las ventanas. Además, se encontraba lo más alejado posible de fuentes de ruido externas.

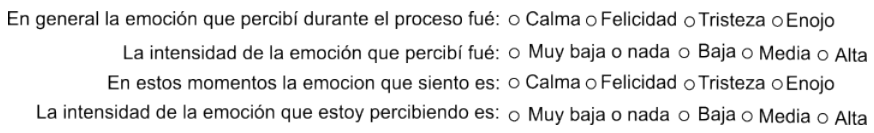

Figura 3.16 Preguntas en reporte de emociones

Se informó a los participantes que durante la segunda parte del proceso debían presionar algunas teclas para registrar la intensidad de la emoción después de observar cada imagen en los videos. Las cuatro teclas seleccionadas para el registro se muestran en la Tabla 3.12. Un auto reporte que se puede ver en la Figura 3.16 contiene 4 preguntas fue utilizado al final de cada proceso. El reporte nos permitió obtener la emoción de los participantes después de haber sido sometidos a los estímulos audiovisuales.

Tabla 3.12 Ejemplo de teclas y tiempo registrado de los participantes

Tecla	**Intensidad**
Num 7	Muy baja o nada
Num 8	Baja
Num 9	Media
Num +	Alta

El experimento siguió el siguiente protocolo:

1. Se explica al participante de manera verbal y por escrito en que consiste el experimento y se le proporciona una carta de consentimiento para que la lea y firme para autorizar el uso de sus datos.
2. Se le indica al participante que en la segunda parte del proceso después de ver cada imagen en el video debe presionar alguna de las teclas seleccionadas para indicar la intensidad de la emoción.
3. Se coloca el sensor para ritmo cardíaco y se verifica que registre los datos correctamente.
4. El participante realiza una serie de respiraciones para relajarse y se coloca los audífonos.
5. Se inicia el primer proceso de emoción de calma.
6. Se contesta el reporte del experimento.
7. Se realiza el proceso 2 de inducción a enojo.
8. Se contesta el auto reporte del experimento.
9. Se comienza el proceso 3 de inducción a tristeza.
10. Se contesta el auto reporte del experimento.
11. Se inicia el proceso 4 con el estímulo de felicidad.

Análisis de datos

Los datos obtenidos del experimento se analizaron para crear un conjunto de datos que pudiera utilizarse el entrenamiento de algoritmos de clasificación de emociones. Los datos obtenidos del experimento fueron:

1. El ritmo cardíaco de los participantes.
2. Las teclas que el participante presionó y el momento en que fueron presionadas.

El ritmo cardíaco de los participantes se obtuvo de la plataforma del sensor OH1. El archivo como se muestra en la Tabla 3.13 contiene el registro del ritmo cardíaco del proceso de calma que tiene un tamaño de 410 x 2. El achivo en formato CSV contiene en la primera columna el tiempo en segundos desde que inicio el monitoreo del ritmo cardíaco y en la segunda columna el valor del ritmo cardíaco del participante.

El registro del inicio y fin de cada proceso del experimento se realizó de forma manual y de esa forma se obtuvieron los valores del ritmo cardíaco de cada uno de los participantes.

Tabla 3.13 Ejemplo de archivo CSV obtenido de sensor OH1

Time	HR (bpm)
00:12:00	71
00:12:01	71
00:12:02	72
00:12:03	72
00:12:04	72
00:12:05	72
00:12:06	72
...	... ...
00:15:15	74
00:15:16	74
00:15:17	74
00:15:18	75
00:15:19	75
00:15:20	76
00:15:21	76

El software Pyscopy se utilizó para registrar las teclas presionadas por los participantes después de mostrar los estímulos audio visuales. El archivo con la información de las teclas

presionadas por participante se formó utilizando la nomenclatura $P1\_1\_K$ para el proceso 1 y la parte 1, $P1\_1\_T$ para registrar el tiempo en segundos del proceso 1 y parte 1. Las columnas $P1\_2\_K$ y $P1\_2\_T$ muestran los datos de la parte 2 y así sucesivamente como se puede observar en la Tabla 3.14.

Tabla 3.14 Teclas y tiempo utilizados para registrar intensidad de la emoción

$P1\_1\_K$	$P1\_1\_T$	$P1\_2\_K$	$P1\_2\_T$	**...**	$P4\_1\_K$
num_add	3.1982228	num_8	3.3299071	...	
num_8	12.3501446	num_add	11.9682138	...	
num_9	23.0128359	num_7	21.4725897	...	
num_7	33.1327648	num_add	31.4980192	...	
num_7	41.8057778	num_add	42.2891519	...	
num_8	51.517571	num_9	52.7921626	...	
num_7	61.3251057	num_add	61.4648016	...	

La cantidad total de tecla presionadas por participante se muestra en la Tabla 3.15 cada una de esas teclas está asociada con la intensidad muy baja o nada con valor 0, baja con valor 1, media con valor 2 y alta con valor 3.

La cantidad de teclas presionadas por los participantes se analizaron para revisar cuantas corresponden a los valores 2 y 3 de intensidad, con la finalidad de obtener valores en los que los participantes reportaron la emoción buscada. El análisis mostró que los participantes 01 y 02 no registraron partes con intensidad media y alta. Los participantes 05, 06 y 07 no registraron datos debido a una falla técnica. Los datos del participante 09 se encuentran incompletos debido a que no se registró información en la emoción de enojo. Los participantes 03, 04, 08, 10 y 11, cuentan con segmentos de ritmo cardíaco asociados en cada una de las emociones utilizadas en el experimento.

Tabla 3.15 Cantidad de teclas presionadas por los participantes y emoción

Participante	Emoción	# de teclas presionadas			
P01	CALMA	1	P01	ENOJO	7
P01	FELICIDAD	2	P01	TRISTEZA	2
P02	CALMA	5	P02	ENOJO	18
P02	FELICIDAD	12	P02	TRISTEZA	12
P03	CALMA	21	P03	ENOJO	33
P03	FELICIDAD	21	P03	TRISTEZA	25
P04	CALMA	14	P04	ENOJO	27
P04	FELICIDAD	29	P04	TRISTEZA	17
P05	CALMA	0	P05	ENOJO	0
P05	FELICIDAD	0	P05	TRISTEZA	0
P06	CALMA	0	P06	ENOJO	0
P06	FELICIDAD	0	P06	TRISTEZA	0
P07	CALMA	0	P07	ENOJO	0
P07	FELICIDAD	0	P07	TRISTEZA	0
P08	CALMA	20	P08	ENOJO	22
P08	FELICIDAD	16	P08	TRISTEZA	22
P09	CALMA	19	P09	ENOJO	21
P09	FELICIDAD	15	P09	TRISTEZA	20
P10	CALMA	19	P10	ENOJO	20
P10	FELICIDAD	16	P10	TRISTEZA	21
P11	CALMA	15	P11	ENOJO	14
P11	FELICIDAD	14	P11	TRISTEZA	17

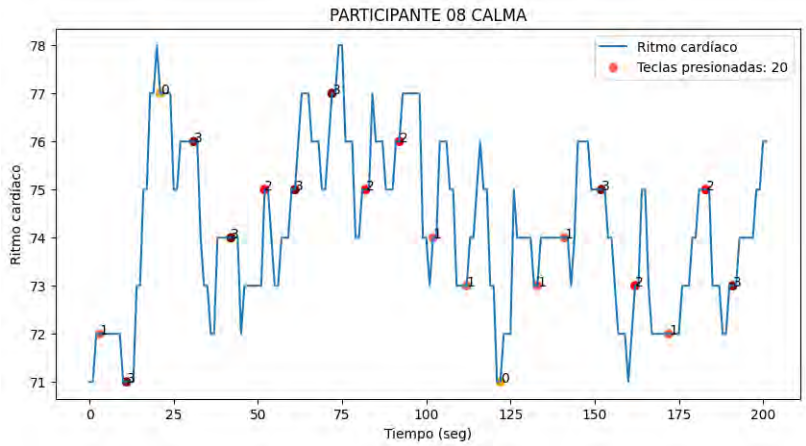

Figura 3.17 Ritmo cardíaco y momentos en que se presionaron las teclas para indicar la intensidad de la emoción

El análisis de los datos recopilados reveló inconsistencias en algunos registros: algunos participantes no reportaron las emociones inducidas de la manera esperada. Otros no registraron datos debido a una falla técnica. Los datos del experimento al estar incompletos en el caso de algunos participantes se decide no incluirlos en el conjunto de datos y utilizar solo la información obtenida del participante 08. A pesar de los inconvenientes en la recolección de datos, fue posible contar con registros completos de todas las emociones inducidas, por lo que se extrajo la información del ritmo cardiaco en cada uno de los procesos del experimento.

El conjunto de datos se formó utilizando los datos registrados en la segunda parte de cada proceso del experimento debido a que es la parte que contiene la información asociada a la emoción que se estaba tratando de provocar. El ritmo cardíaco se extrajo utilizando la duración de cada canción como se muestra en la Tabla 3.10. Los datos se obtienen de los archivos mostrados en las tablas 3.13 y 3.14. El tiempo total del proceso de calma y el registro de las teclas presionadas por el participante 08 se muestran en la gráfica de la Figura 3.17.

Los momentos en los que se presionaron las teclas se utilizaron como marca para obtener un fragmento del ritmo cardíaco tomando 5 segundos antes y 5 segundos después para formar segmentos de 10 segundos por cada marca asociada con una emoción y su intensidad. En la Figura 3.18 se muestra un ejemplo del participante 08 con los segmentos que se extraen en color rojo.

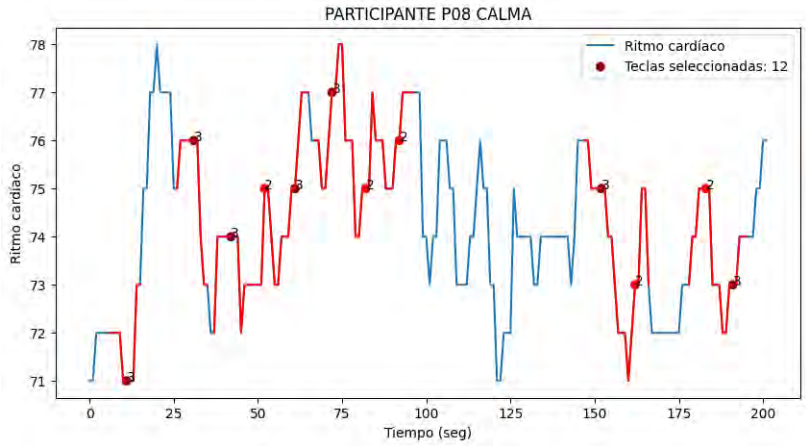

Figura 3.18 Ritmo cardíaco y momentos en que se presionaron las teclas para indicar la intensidad de la emoción.

El archivo del conjunto de datos que se construyó inicialmente a partir de los datos del experimento se muestra en la Tabla 3.16 como ejemplo del participante 08. El conjunto de datos esta formado por las variables: Participante; Ritmo Cardíaco; 1ra Did que representa la primera diferencia del ritmo cardíaco; 2da Dif que representa la segunda diferencia del ritmo cardíaco y emoción. La primera y segunda diferencia se calculó como una forma de

Tabla 3.16 Ejemplo conjunto de datos con ritmo cardíaco y 1ra y 2da diferencia

Part.	R. Cardíaco (HR)	1ra Dif & 2da Dif	Emoción
P08	73,74,74,74,72,71,72,72	0,1,0,0,-2,-1,1,0 0,1,-1,0,-2,1,2,-1	Enojo
P08	77,77,77,77,76,77,77,77	-1,0,0,0,-1,1,0,0 -1,1,0,0,-1,2,-1,0	Enojo
P08	70,70,70,70,70,70,70,70	0,0,0,0,0,0,0,0 0,0,0,0,0,0,0,0	Felicidad
P08	73,74,74,74,72,71,72,72,72,73	0,1,0,0,-2,-1,1,0,0,1 0,1,-1,0,-2,1,2,-1,0,1	Enojo
P08	77,77,77,77,76,77,77,77,78,78	-1,0,0,0,-1,1,0,0,1,0 -1,1,0,0,-1,2,-1,0,1,-1	Enojo
P08	70,70,70,70,70,70,70,70,70,70	0,0,0,0,0,0,0,0,0,0 0,0,0,0,0,0,0,0,0,0	Felicidad
P08	80,78,77,77,77,77,75,75,76,76	1,-2,-1,0,0,0,-2,0,1,0 1,-3,1,1,0,0,-2,2,1,-1	Enojo
P08	76,76,75,75,76,77,77,78,78,76	0,0,-1,0,1,1,0,1,0,-2 1,0,-1,1,1,0,-1,1,-1,-2	Tristeza
P08	77,77,77,77,77,79,79,79,79,76	0,0,0,0,0,2,0,0,0,-3 2,0,0,0,0,2,-2,0,0,-3	Calma
P08	84,81,81,79,77,75,75,75,76,76	-3,-3,0,-2,-2,-2,0,0,1,0 -2,0,3,-2,0,0,2,0,1,-1	Tristeza
P08	73,75,75,75,73,72,72,71,71,70	0,2,0,0,-2,-1,0,-1,0,-1 0,2,-2,0,-2,1,1,-1,1,-1	Felicidad
P08	72,74,74,74,72,72,72,71,71,72	0,2,0,0,-2,0,0,-1,0,1 -1,2,-2,0,-2,2,0,-1,1,1	Enojo
P08	72,74,74,74,74,74,74,74,72,73	0,2,0,0,0,0,0,0,-2,1 1,2,-2,0,0,0,0,0,-2,3	Enojo

obtener los cambios del ritmo cardíaco durante el tiempo en que se aplicaron los estímulos audiovisuales. El archivo del conjunto de datos tiene un total de 78 registros.

La distribución de los registro por cada emoción del conjunto de datos 3.17.

Tabla 3.17 Cantidad de registros de cada emoción del conjunto de datos para entrenar el modelo.

Emoción	**Cantidad de registros**
Calma	20
Enojo	22
Feliz	19
Tristeza	17
Total	**78**

Este primer conjunto de datos mostrado en la Tabla 3.17 como se puede observar contiene pocos datos para el entrenamiento de un modelo de detección de emociones. Además, no contempla la información proveniente de los audios y las imágenes utilizadas como estímulos audiovisuales durante el experimento. El conjunto de datos formado a partir de la información obtenida del experimento de emociones se compone de: el ritmo cardíaco del participante, los valores de la valencia y activación de las imágenes mostradas en los videos y las caraceterísticas del audio utilizado.

La información de ritmo cardíaco, imágenes y audio obtenida del experimento se procesó como una serie de tiempo para así poder obtener más datos para el entrenamiento del modelo. Los datos obtenidos fueron preprocesados. El preprocesamiento de datos es una etapa fundamental en el desarrollo del modelo de clasificación de emociones, ya que permite transformar los datos brutos en un formato óptimo para su análisis y procesamiento. En esta investigación, las señales de ritmo cardíaco obtenidas del experimento fueron preprocesadas con el objetivo de eliminar el ruido presente en los registros. Para este proceso, se aplicó la técnica de filtro de media móvil, la cual suaviza la señal y reduce fluctuaciones no deseadas. Esta técnica se define mediante la siguiente fórmula:

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[x-k] \tag{3.5}$$

Donde M es el tamaño de la ventana, $x[n]$ es la señal de entrada y $y[n]$ la señal suavizada. La técnica de filtro de media móvil nos permitió suavizar la señal eliminando el ruido que presentaba inicialmente. La señal suavizada se puede ver en el ejemplo de la Figura 3.19 . Esta técnica se aplicó a los datos de ritmo cardíaco, imagen y características de audio.

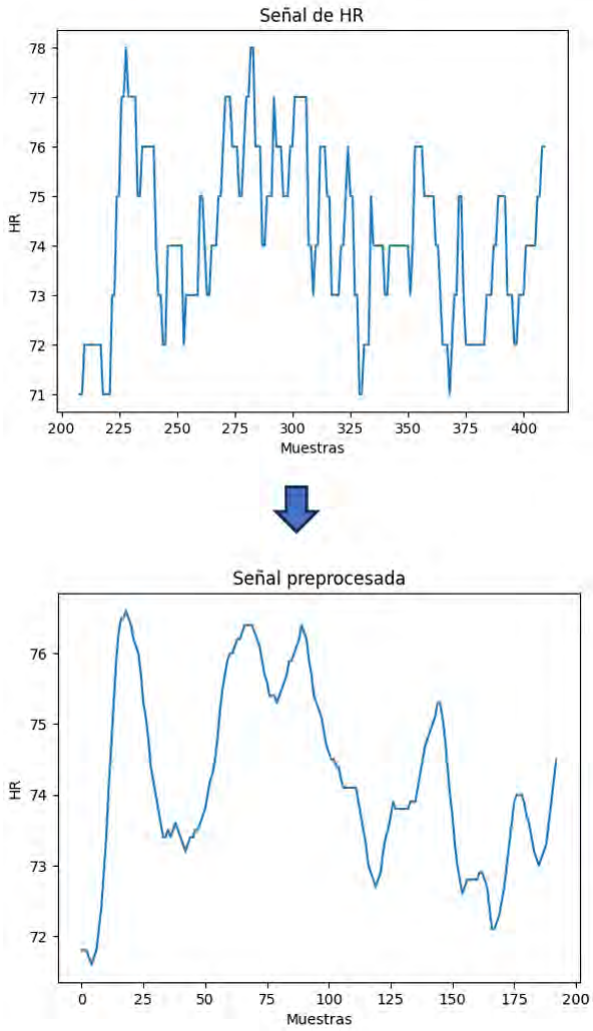

Figura 3.19 Señal preprocesada mediante Filtro de media móvil.

La extracción de características es una etapa crucial en el proceso de reconocimiento de emociones, ya que permite obtener los atributos más representativos de las señales capturadas, facilitando así el entrenamiento de los modelos de aprendizaje automático.

La extracción de características de realizó empleando una técnica de ventana deslizante para segmentar las señales en fragmentos de tamaño predefinido. Este enfoque permite analizar la variabilidad temporal de las señales y capturar patrones dinámicos en la evolución del ritmo cardíaco. La ventana deslizante facilita la detección de transiciones emocionales al proporcionar información en intervalos sucesivos, lo que resulta esencial para el reconocimiento preciso de estados afectivos. Para cada fragmento obtenido mediante la ventana deslizante, se calcularon un conjunto de características relevantes, las cuales fueron seleccionadas en función de su capacidad para discriminar entre diferentes emociones. Estas características incluyen métricas estadísticas, frecuencia y dominio

temporal, entre otras. La Figura 3.20 ilustra el proceso de segmentación y la extracción de características a partir de los fragmentos generados.

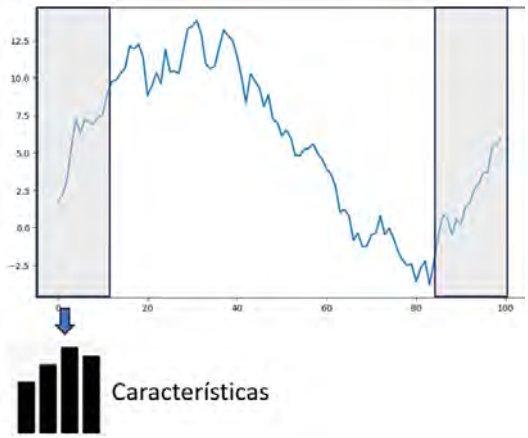

Figura 3.20 Extracción de características con Ventana deslizante

Las características del ritmo cardíaco fueron:

Valor mínimo

Representa el valor más bajo dentro de ese rango de rimo cardíaco.

$$min(x) = min(x_1, x_2, ..., x_n) \tag{3.6}$$

Rango (máximo – mínimo)

Representa la amplitud de los valores del ritmo cardíaco

$$rango = max(x) - min(x) \tag{3.7}$$

Media

Indica el valor general del ritmo cardíaco dentro de ese rango

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.8}$$

Desviación estándar

Indica que tanto fluctúa el ritmo cardíaco dentro de ese rango de valores

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{3.9}$$

Varianza

Representa otra medida de variabilidad de los valores

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{3.10}$$

Energía

Es una medida de la actividad de la señal acumulada en un periodo de tiempo

$$E = \sum_{i=1}^{n} x_i^2 \tag{3.11}$$

Potencia

Permite observar cómo se mantiene la energía de la señal a lo largo del tiempo

$$P = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \tag{3.12}$$

Los archivos de la música utilizada en los estímulos fueron procesados mediante el paquete de lenguaje Python pyAudioAnalysis versión 0.3.14. La función ShortTermFeatures extrae las características por cada segundo de cada archivo de música utilizado en el experimento. El total de las características de audio es de 68 y son: zcr, energy, energy entropy, spectral centroid, spectral spread, spectral entropy, spectral flux, spectral rolloff, mfcc 1 a mfcc 13, chroma 1 a chroma 12, chroma std, delta zcr, delta energy, delta energy entropy, delta spectral centroid, delta spectral spread, delta spectral entropy, delta spectral flux, delta spectral rolloff, delta mfcc 1 a delta mfcc 13, delta chroma 1 a delta chroma 12, delta chroma std. La Tabla 3.18 muestra un ejemplo de los datos.

Tabla 3.18 Ejemplo de archivo con datos de audio

zcr	**energy**	**energy_entropy**	**...**
0.020182321	4.96E06	2.69062057	...
0.027574947	5.01E05	3.137089981	...
0.046940904	0.000177845	3.294968515	...
...	...	...	...

Los datos de las imágenes utilizadas en cada estímulo también se ordenaron de acuerdo al orden en que se fueron presentando, formando así un archivo por cada emoción. Los datos corresponden a los valores de valencia y activación de cada imagen que fueron tomados como sus características y se muestran en la Tabla 3.19.

Tabla 3.19 Ejemplo de archivo con datos de imágenes

Valencia	**Activación**	**Emoción**
6.09	3.8	Calma
6.09	3.8	Calma
6.09	3.8	Calma
6.09	3.8	Calma
4.04	4.91	Enojo
4.04	4.91	Enojo
4.04	4.91	Enojo
4.04	4.91	Enojo
7.63	4.87	Felicidad
7.63	4.87	Felicidad
7.63	4.87	Felicidad
7.63	4.87	Felicidad
3.85	4.83	Tristeza
3.85	4.83	Tristeza
3.85	4.83	Tristeza
3.85	4.83	Tristeza
...	...	...

Los archivos del participante seleccionado para el desarrollo del modelo de clasificación de emociones que contienen los datos de ritmo cardíaco, imágenes y característicos de

audio forman un total de 12; tres por cada proceso. Las características del ritmo cardíaco, imágenes y audio se etiquetaron con las emociones de calma, enojo, felicidad y tristeza.

El conjunto de datos utilizado para el entrenamiento del modelo de clasificación de emociones se construyó a partir de la unión de los datos de características de ritmo cardíaco calculadas por las ecuaciones 3.6 a la ecuación 3.12, los datos de audio de la Tabla 3.18 y los datos de las imágenes de la Tabla 3.19 con un total de 78 columnas y 749 renglones.

De esta forma al utilizar la técnica de ventana deslizante se construypó un nuevo conjunto de datos con mas registros para el entrenamiento del modelo. La Tabla 3.20 muestra una comparación de ambos conjuntos de datos.

Tabla 3.20 Comparativa de cantidad de registros de cada emoción del conjunto de datos inicial y final para entrenar el modelo.

Emoción	Cantidad inicial	Cantidad final
Calma	20	192
Enojo	22	211
Feliz	19	144
Tristeza	17	202
Total	**78**	**749**

Como se puede ver en la Tabla 3.20 la cantidad de registros para entrenamiento se aumentó considerablemente. El conjunto se dividió en dos partes la primera de tamaño 599x78 para entrenamiento y la segunda de tamaño 150x78 para prueba.

3.3. Desarrollo de modelos de detección de emociones

En esta sección se describe el desarrollo de los modelos de detección de emociones. En la primera parte, se construyó un modelo enfocado en la identificación de emociones a partir de señales de voz, dado que este fue el primer tipo de información disponible para el presente trabajo de investigación. Para la creación de este modelo, se utilizaron datos de voz provenientes del conjunto de datos RAVDESS. Posteriormente, se entrenaron dos algoritmos: una máquina de soporte vectorial (SVM) y una red neuronal, con el objetivo de comparar su desempeño en la tarea de clasificación de emociones.

En la segunda parte del desarrollo se construyó un modelo basado en información obtenida a partir de un experimento de emociones, en el cual se registraron datos del ritmo cardíaco como respuesta a estímulos audiovisuales. Para este modelo, se empleó la técnica de programación genética con el propósito de procesar los datos de ritmo cardíaco, así

como las imágenes y el audio de los estímulos. Los datos procesados fueron utilizados para la clasificación de emociones mediante una red neuronal.

3.3.1. Desarrollo de modelos de detección de emociones con voz utilizando machine learning y redes neuronales

El desarrollo de los modelos para detección de emociones en la voz de presenta en esta sección. Los modelos se desarrollaron utilizando el algoritmo de máquinas de soporte vectorial y redes neuronales perceptrón multicapa como se muestra en la Figura 3.21.

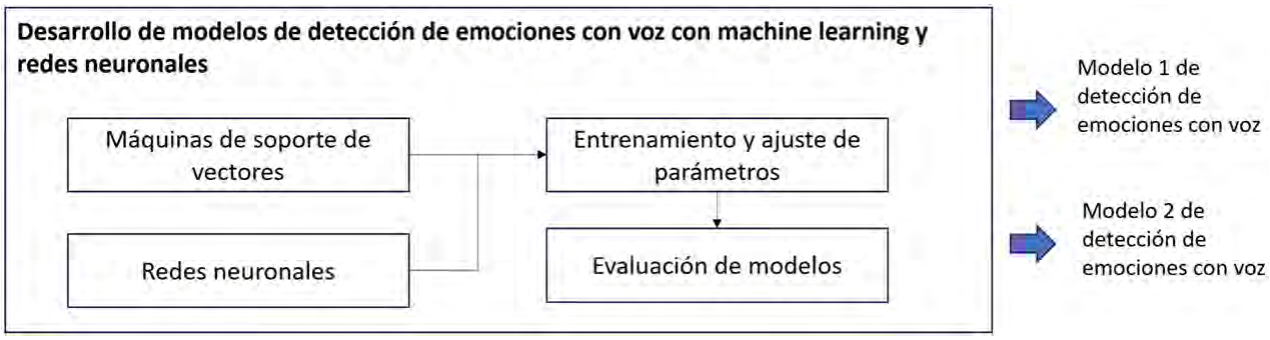

Figura 3.21 Proceso de desarrollo de conjunto de datos.

El entrenamiento de los modelos es el proceso en el cual un algoritmo utiliza los datos obtenidos de las señales de audio como entrada y realiza un ajuste para obtener una salida asociada a una emoción. Los modelos entrenados para la detección de emociones en la voz utilizan redes neuronales y máquinas de soporte vectorial (SVM) como estrategias para clasificar los archivos de audio en una de las siguientes emociones: neutral, calma, felicidad, tristeza, enojo, miedo, disgusto y sorpresa.

Las redes neuronales son estructuras que pueden aprender relaciones complejas entre los datos extraídos de las señales de voz y las emociones representadas en ellas, lo que les permite generar modelos capaces de clasificar estas señales en distintas emociones. Por otro lado, las máquinas de soporte vectorial (SVM), incluso con pocos datos, pueden encontrar un hiperplano que separa las instancias de diferentes emociones en el conjunto de datos, proporcionando un modelo de clasificación robusto para esta tarea.

Las redes neuronales y las máquinas de soporte vectorial son técnicas ampliamente utilizadas que han demostrado buenos resultados; por esta razón, hemos decidido emplearlas y comparar los resultados obtenidos en el conjunto de datos RAVDESS.

El entrenamiento se llevó a cabo utilizando un paquete de Python llamado Optuna versión 3.1.1 (Akiba, Sano, Yanase, Ohta, y Koyama, 2019). El objetivo principal de Optuna es encontrar de manera eficiente los valores óptimos de los hiperparámetros que maximizan o minimizan una función objetivo. A diferencia de técnicas tradicionales como la búsqueda

en malla (grid search) o la búsqueda aleatoria (random search), que exploran el espacio de hiperparámetros de manera estática y frecuentemente ineficiente, Optuna emplea un enfoque basado en optimización bayesiana y técnicas de paro anticipado automático (pruning). Esto le permite adaptar dinámicamente la exploración del espacio de búsqueda en función de los resultados obtenidos en iteraciones anteriores, centrándose en las regiones más prometedoras. Además, Optuna permite definir la función objetivo de forma flexible y manejar estudios paralelos, lo que lo convierte en una herramienta poderosa y escalable para la optimización de modelos en tareas de aprendizaje automático.

Máquinas de soporte vectorial

Las máquinas de soporte vectorial (SVM) son un clasificador estadístico que puede clasificar datos de manera binaria o en múltiples clases (Bishop, 2006). En las máquinas de soporte vectorial, se construyen hiperplanos (Figura 3.22) sobre el espacio multidimensional de los datos, los cuales pueden utilizarse para tareas de clasificación o regresión. Un componente importante en las máquinas de soporte vectorial es el kernel, una función matemática que permite la construcción de hiperplanos para realizar la clasificación de los datos.

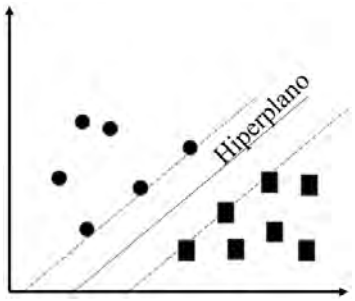

Figura 3.22 Hiperplano de máquinas de soporte vectorial.

El kernel polinomial emplea funciones polinomiales donde x e y son vectores de entrada, c es un término constante y d es el grado del polinomio. El kernel polinomial tiene la ventaja de detectar tanto correlaciones lineales como no lineales en los datos.

$$K(x, y) = (x * y + c)^d \tag{3.13}$$

La implementación del algoritmo de Máquinas de Soporte Vectorial (SVM) se llevó a cabo utilizando el paquete de Python Scikit-learn versión 1.2.2 con los siguientes parámetros:

C: Este es el parámetro de costo, donde un valor bajo puede permitir clasificaciones incorrectas, mientras que un valor alto crea un límite de decisión más ajustado para clasificar correctamente las instancias.

Coef0: Se utiliza en el kernel polinomial y controla la influencia de los términos de mayor grado en la función del kernel.

Degree: Es el grado del polinomio utilizado; un valor más alto puede ajustarse a relaciones más complejas.

Kernel: Es el tipo de kernel que se utilizará.

Gamma: Define el grado de influencia de una instancia de entrenamiento.

El modelo generado utilizó los datos de entrenamiento y el algoritmo de Máquinas de soporte vectorial con los parámetros mostrados en la Tabla 3.21.

Tabla 3.21 Rangos de valores utilizados por Optuna para ajustar los paramétros del modelo SVM.

Parámetro	**Valores**
C	0.01 a 1000
coef0	0.001 a 1.0
Degree	1 a 10
Gamma	0.1 a 10
Kernel	linear, poly, rbf, sigmoid

Los valores de los parámetros del algoritmo mostrados en la Tabla 3.21 se utilizaron para encontrar la mejor combinación en la que el modelo lograra los mejores resultados en la clasificación de emociones. El algoritmo de Máquinas de Soporte Vectorial (SVM) demostró ser efectivo en la clasificación de emociones en audio.

La optimización de parámetros, como el tipo de kernel, la constante de regularización (C) y el parámetro gamma, logró mejorar la precisión del modelo. Estos parámetros se definieron después de ejecutar el proceso de de búsqueda partiendo de los valores de la tabla 3.21. El proceso de búsqueda se realizó durante 2000 iteraciones encontrandose los valores que mejor resultado en la metrica de exactitud en la iteración 845 como se puede observar en la Figura 3.23.

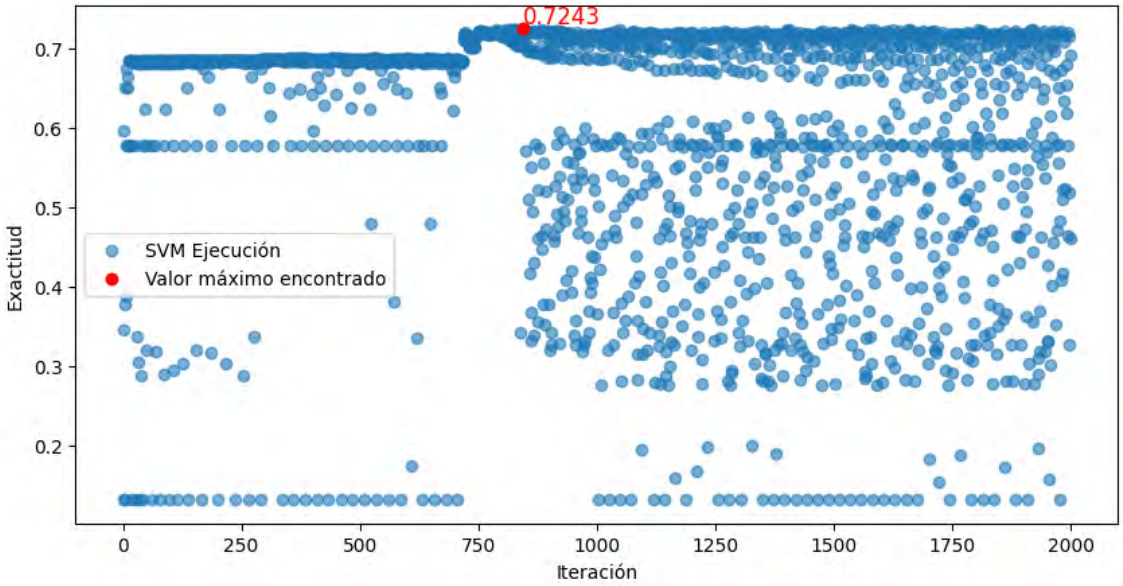

Figura 3.23 Gráfica de valores de exactitud en cada iteración durante el proceso de búsqueda de parámetros para la máquina de soporte vectorial.

El algoritmo de máquinas de soporte vectorial (SVM), después de realizar el proceso de búsqueda durante 2000 iteraciones, muestra sus parámetros en la Tabla 3.22. Los parámetros encontrados para la red neuronal después de 2000 iteraciones se presentan en la Tabla 3.24.

Tabla 3.22 Parámetros finales de máquinas de soporte vectorial

Parámetro	**Valores**
C	276.08207104667605
coef0	0.061262500520705684
Degree	5
Gamma	0.549180099605578
Kernel	rbf

El modelo construido con el algoritmo de máquinas de soporte vectorial se resume de la siguiente manera:

Parámetros de entrenamiento:

- 'C': 276.08207104667605
- 'coef0': 0.061262500520705684

- 'decision_function_shape': 'ovr'
- 'degree': 5
- 'gamma': 0.549180099605578
- 'kernel': 'rbf'
- 'max_iter': 100000
- 'probability': False'
- 'random_state': 42
- 'tol': 0.001

Número de características: 428

Número total de vectores soporte: 1134

Número de vectores soporte por clase: [77 145 153 153 154 154 150 148]

Resumen estadístico de los vectores soporte:

- Media global: 0.1546
- Desviación estándar global: 0.2417
- Valor mínimo: 0.0000
- Valor máximo: 1.0000

Resumen de los coeficientes duales:

- Media: -0.0000
- Desviación estándar: 1.2010
- Mínimo: -7.4870
- Máximo: 7.3294

Red neuronal perceptrón multicapa

El perceptrón multicapa es una versión evolucionada del perceptrón simple. Esta versión del perceptrón incorpora capas ocultas de neuronas, lo que le permite representar funciones no lineales.

El perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y n capas ocultas. La capa de entrada es donde se reciben las variables de entrada. Las capas ocultas forman un conjunto donde cada salida está conectada a la siguiente capa, y, en última instancia, se obtiene una suma ponderada, incluyendo sus umbrales o pesos.

La capa de salida proporciona los resultados esperados de la red, como se muestra en la Figura 3.24.

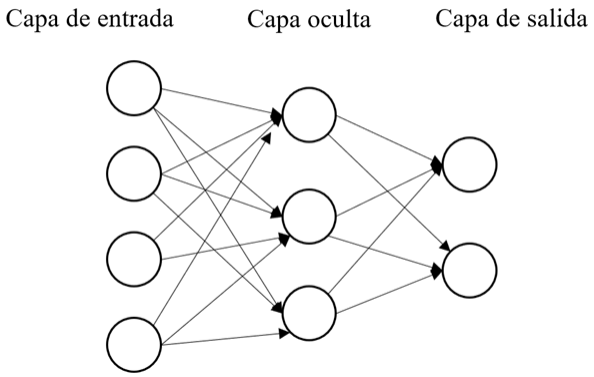

Figura 3.24 Red neuronal perceptrón multicapa.

La implementación de la red neuronal perceptrón multicapa se llevó a cabo utilizando el paquete de Python Scikit-learn versión 1.2.2, donde se definieron los siguientes parámetros:

Activation: Define la función de activación utilizada.

Alpha: Controla la penalización de la red neuronal y ayuda a prevenir el sobreajuste (overfitting).

Batch_size: Especifica el número de instancias que se utilizarán en cada iteración del entrenamiento.

Hidden_layer_sizes: Define el número de neuronas en cada capa oculta.

Learning_rate: Controla la tasa de aprendizaje de la red; el valor constant mantiene una tasa de aprendizaje constante, invscaling reduce la tasa de aprendizaje a medida que avanza el entrenamiento, y adaptive ajusta automáticamente la tasa de aprendizaje en función de la convergencia.

Learning_rate_init: Valor inicial de la tasa de aprendizaje.

Solver: Especifica qué algoritmo ajustará los pesos de la red; puede tomar los valores sgd, adam y lbfgs.

Los parámetros utilizados por Optuna para generar el modelo con la red neuronal perceptrón multicapa se muestran en la Tabla 3.23.

Tabla 3.23 Perceptron Neural Network Parameters.

Parámetro	**Valores**
Activation	logistic, tanh, relu
Alpha	0.0001 a 0.99
batch_size	32,64,128,256,512,1024
hidden_layer_sizes	8 a 1000
learning_rate	adaptive, constant, invscaling
learning_rate_init	0.001 a 0.99
Solver	sgd, adam

De forma similar se utiizó el paquete Optuna para la búsqueda de los mejores parámetros para la red neuronal. El proceso de búsqueda de parámetros se realizó en 2000 iteraciones, encontrandose el mejor resultado en la iteración 844 como se puede ver en la Figura 3.25.

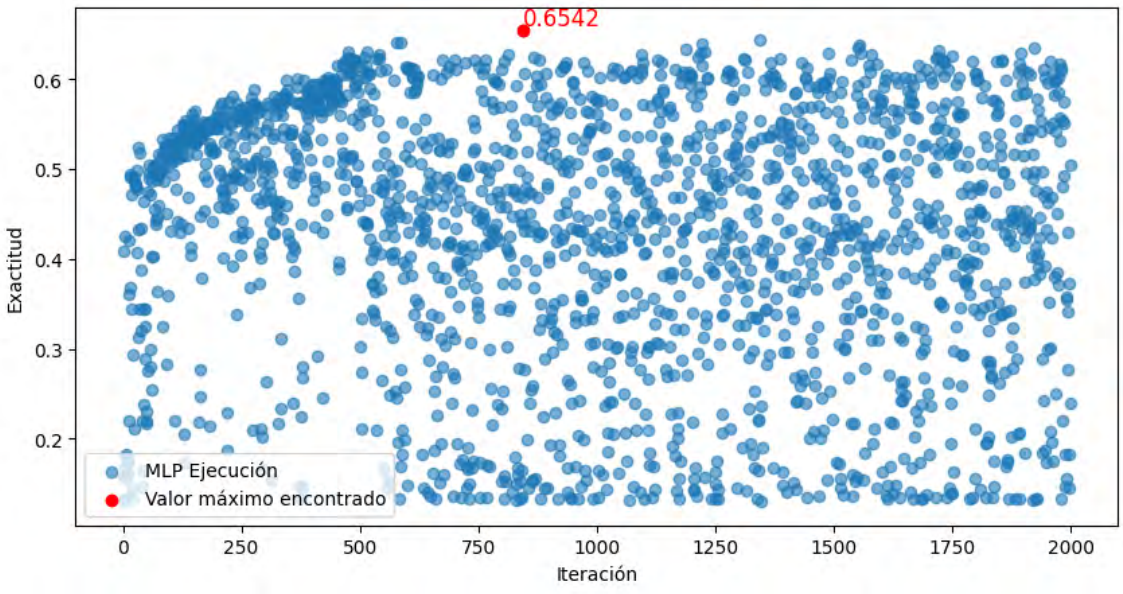

Figura 3.25 Gráfica de valores de exactitud en cada iteración durante el proceso de búsqueda de parámetros para la red neuronal.

Tabla 3.24 Parámetros de la red neuronal perceptrón multicapa.

Parámetro	**Valores**
Activation	relu
Alpha	0.00011725999683111915
batch_size	32
hidden_layer_sizes	421
learning_rate	adaptive
learning_rate_init	0.055495929262743944
Solver	sgd

La arquitectura de la red neuronal utilizada se muestra en la Figura 3.26 y fue entrenada con los parámetros especificados en la Tabla 3.24. La capa de entrada cuenta con un total de 428 neuronas, conforme al número de variables. La capa oculta contiene 421 neuronas, mientras que la capa de salida tiene 8 neuronas, correspondientes a las emociones de: neutral, calma, enojo, felicidad, tristeza, disgusto y miedo.

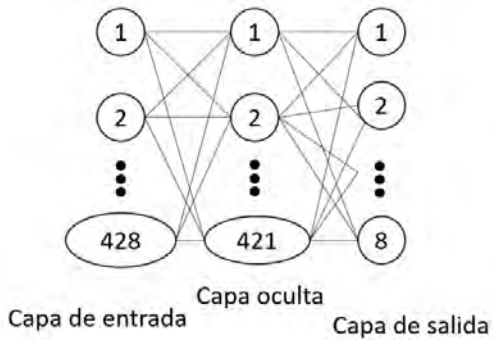

Figura 3.26 Arquitectura de la red neuronal perceptrón multicapa.

El modelo construido por la red neuronal se resume de la siguiente manera:

Número de capas (input + ocultas + output): 3

Número de neuronas en la capa oculta: 421

Capa 1:

Pesos

- Forma: (428, 421)
- Media: -0.0008
- Desviación estándar: 0.0810

Sesgo o Bias

- Forma: (421,)
- Media: -0.0194
- Desviación estándar: 0.0659

Capa 2:

Pesos

- Forma: (421, 8)
- Media: -0.0015
- Desviación estándar: 0.2343

Sesgo o Bias

- Forma: (8,)
- Media: -0.0287
- Desviación estándar: 0.2690

Parámetros de entrenamiento:

- Función de activación: relu
- Algoritmo de optimización: sgd

3.3.2. Desarrollo de modelo de detección de emociones con ritmo cardíaco

El modelo desarrollado para la detección de emociones con ritmo cardíaco siguió el proceso que se muestra en la Figura 3.27. El modelo utiliza los datos obtenidos del experimento después de ser preprocesados y después de haber obtenido las características de los estímulos utilizados y el ritmo cardíaco.

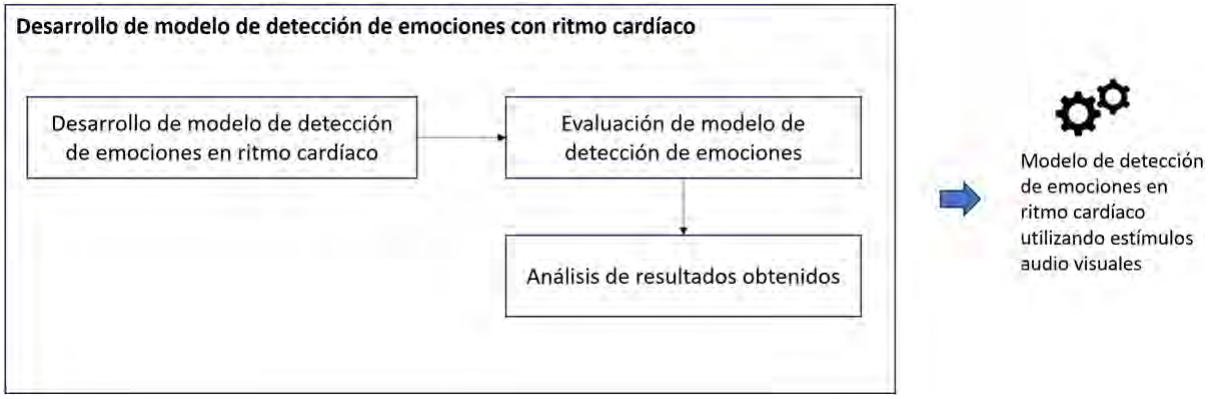

Figura 3.27 Proceso de desarrollo de modelo de detección de emociones.

El desarrollo del modelo de detección de emociones utiliza un enfoque híbrido en el que se combinan la técnica de programación genética y redes neuronales como se observa en la Figura 3.28. La primera parte utiliza el conjunto de datos de ritmo cardíaco, imágenes y audio como entrada para la técnica de programación genética y se obtienen datos en un espacio de valores que puedan ser separables. La siguiente parte entrena una red neuronal con los datos transformados para realizar la clasificación de las emociones de calma, enojo, felicidad y tristeza.

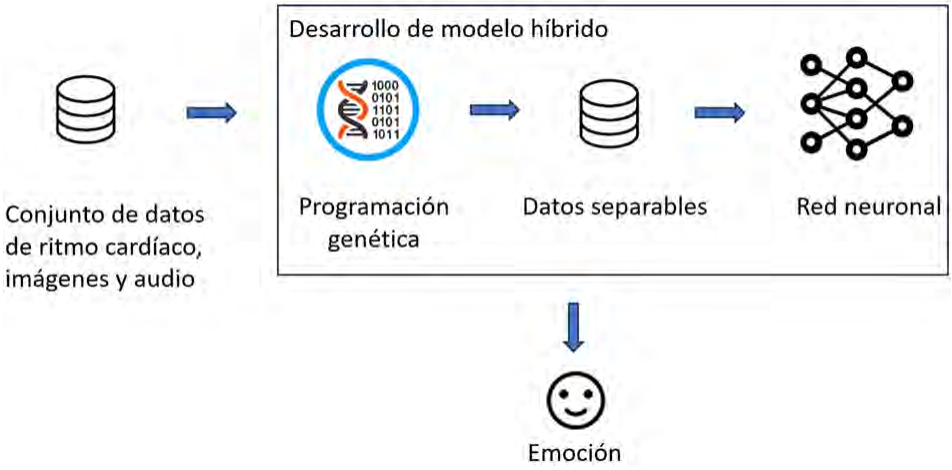

Figura 3.28 Desarrollo de modelo híbrido de clasificación de emociones.

Programación genética

La programación genética se utilizó en este modelo híbrido para generar individuos que se representan mediante un árbol sintáctico en el que las hojas son las variables o características y los nodos intermedios son funciones aritméticas. El individuo, al representar una

función, transforma las características de entrada en un valor de salida. La función a pesar de ser no lineal y al representar un solo valor puede no ser suficiente para ser utilizada como un clasificador. El enfoque multi árbol en programación genética permite que cada individuo pueda contener varias funciones. Un individuo multi árbol puede generar más de una función para que de esa forma se obtenga como salida varios valores. Los valores de salida de las funciones del individuo pueden utilizarse como entrada en una red neuronal para realizar la separación de las instancias y crear un clasificador de las emociones. Un ejemplo de los individuos generados con la técnica de programación genética y el enfoque multi árbol se muestra en la Figura 3.29.

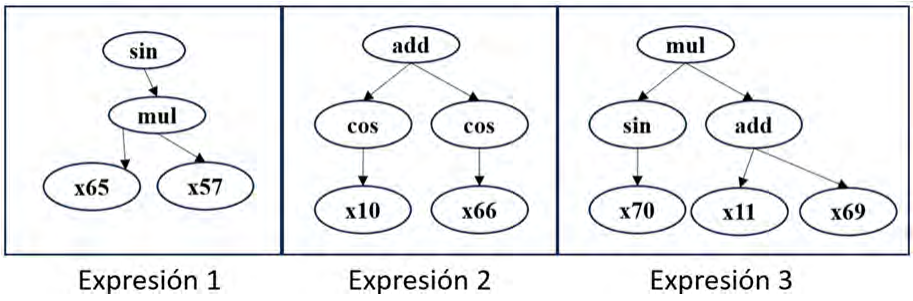

Figura 3.29 Ejemplo de individuo multiárbol.

Terminales

Las terminales se definen en este caso como características con las que se harán las operaciones. Las variables y su correspondiente característica se muestran en la Tabla 3.25.

Operadores genético de cruce

Los operadores genéticos de cruce y mutación operan sobre cada uno de los árboles que componen un individuo. La operación de cruce genera dos nuevos individuos en los que se intercambian nodos de manera aleatoria por cada sub árbol, esta operación permite heredar características de individuos con buen desempeño. Por ejemplo, en la Figura 3.30 se muestran dos individuos de los cuales se seleccionan algunos nodos para intercambiarse.

Tabla 3.25 Definición de variables utilizadas por los individuos.

Variable	**Característica**	**Variable**	**Característica**
x0	Zcr	x40	delta spectral flux
x1	Energy	x41	delta spectral rolloff
x2	energy entropy	x42	delta mfcc 1
x3	spectral centroid	x43	delta mfcc 2
x4	spectral spread	x44	delta mfcc 3
x5	spectral entropy	x45	delta mfcc 4
x6	spectral flux	x46	delta mfcc 5
x7	spectral rolloff	x47	delta mfcc 6
x8	mfcc 1	x48	delta mfcc 7
x9	mfcc 2	x49	delta mfcc 8
x10	mfcc 3	x50	delta mfcc 9
x11	mfcc 4	x51	delta mfcc 10
x12	mfcc 5	x52	delta mfcc 11
x13	mfcc 6	x53	delta mfcc 12
x14	mfcc 7	x54	delta mfcc 13
x15	mfcc 8	x55	delta chroma 1
x16	mfcc 9	x56	delta chroma 2
x17	mfcc 10	x57	delta chroma 3
x18	mfcc 11	x58	delta chroma 4
x19	mfcc 12	x59	delta chroma 5
x20	mfcc 13	x60	delta chroma 6
x21	chroma 1	x61	delta chroma 7
x22	chroma 2	x62	delta chroma 8
x23	chroma 3	x63	delta chroma 9
x24	chroma 4	x64	delta chroma 10
x25	chroma 5	x65	delta chroma 11
x26	chroma 6	x66	delta chroma 12
x27	chroma 7	x67	delta chroma std
x28	chroma 8	x68	HR
x29	chroma 9	x69	VAL
x30	chroma 10	x70	ACT
x31	chroma 11	x71	HR max
x32	chroma 12	x72	HR min
x33	chroma std	x73	HR std
x34	delta zcr	x74	HR var
x35	delta energy	x75	HR range
x36	delta energy entropy	x76	HR power
x37	delta spectral centroid	x77	HR energy
x38	delta spectral spread		
x39	delta spectral entropy		

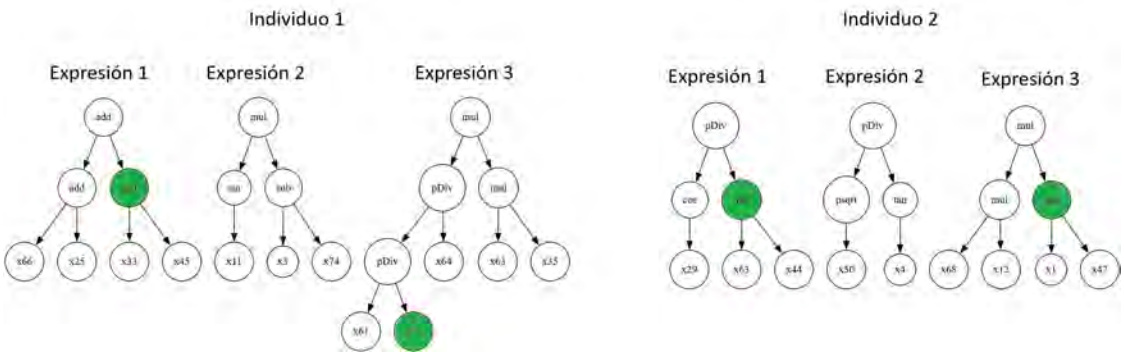

Figura 3.30 Ejemplo de individuos con nodos seleccionado para aplicar operación de cruce.

La operación de cruce para los individuos mostrados en la Figura 3.30 da como resultados dos nuevos individuos en donde se intercambiaron los nodos seleccionados como se muestra en la Figura 3.31. Se puede observar que en la expresión 3 del individuo 1 en la Figura 3.31 no tiene una operación que se incluya en alguno de los individuos padre; esto se debe que en la generación de los individuos existe una restricción que limita la altura y en caso de exceder la altura definida se genera un nuevo arbol de manera aleatoria.

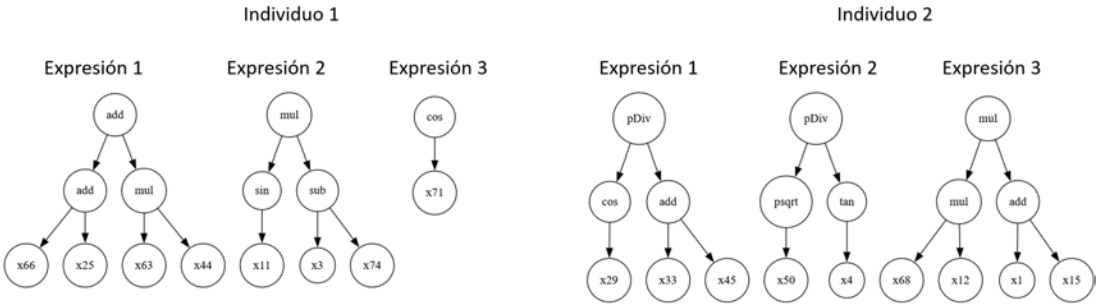

Figura 3.31 Ejemplo de individuos generados a partir de la operación de cruce.

Operadores genético de mutación

La operación de mutación sobre un individuo selecciona alguno de los nodos de cada subárbol y genera una expresión a partir de ese punto. El resultado de la operación de mutación también está limitada en cuanto a la altura del árbol resultante y, en caso de excederse, se genera una nueva expresión. El objetivo de esta operación es introducir variabilidad en la población y permitir la exploración de nuevas regiones del espacio de soluciones. El individuo que resulta de aplica la operación de mutación sobre el individuo de la Figura 3.30 se puede observar en la Figura 3.32.

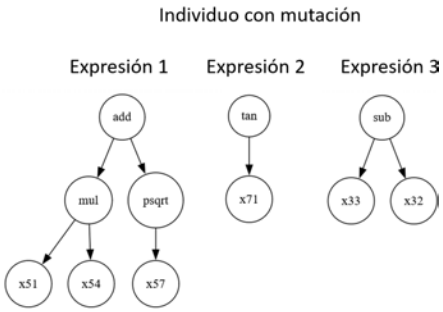

Figura 3.32 Ejemplo de después de aplicar la operación de mutación sobre el individuo 1 de la Figura 3.30.

Selección de individuos

Las operaciones de cruce y mutación se realizan sobre los individuos que son seleccionados, esta selección se realiza mediante la técnica de torneo. La selección por torneo es una técnica utilizada en programación genética para elegir los individuos que participarán en la reproducción. Consiste en seleccionar aleatoriamente un subconjunto de la población (en este caso, de tamaño 5) y evaluar su desempeño según la función de aptitud. De los cinco individuos seleccionados, se elige el que tenga la mejor aptitud para ser padre de la siguiente generación. Este método introduce una presión selectiva controlada, ya que favorece a los mejores individuos dentro del torneo, pero aún permite que individuos menos aptos tengan una pequeña probabilidad de ser seleccionados, lo que contribuye a mantener la diversidad genética en la población.

En la selección de los individuos que componen cada generación se utiliza además la técnica de elitismo que consiste en preservar los mejores individuos de una generación (en este caso un individuo) para asegurarse de que su calidad no se pierda debido a las operaciones aleatorias de cruce o mutación. Esto se logra copiando directamente uno o varios individuos con mejor aptitud a la siguiente generación sin modificarlos, garantizando así que el rendimiento global de la población no empeore con el tiempo. Esta estrategia complementa la selección por torneo, ya que mientras el torneo favorece la selección de buenos individuos de forma estocástica, el elitismo asegura que los mejores realmente sobrevivan, manteniendo una presión evolutiva positiva constante a lo largo de las generaciones. Combinadas, ambas técnicas permiten un equilibrio entre la exploración de nuevas soluciones y la explotación de soluciones óptimas ya encontradas.

Funciones

Las funciones utilizadas (Tabla 3.26) por las soluciones son operaciones aritméticas básicas: suma, resta, multiplicación, división y raíz cuadrada; funciones aritméticas: seno, coseno, tangente. La división se encuentra protegida para que en el caso de que el denominador

sea cero devuelva el valor de 1. La raíz cuadrada se redefine también como la raíz cuadrada del valor absoluto pasado como parámetro para proteger la función en caso de error.

Tabla 3.26 Definición de funciones utilizadas por los individuos.

Función	**Nombre**
Sum	Suma
Sub	Resta
Mul	multiplicación
pDiv	división
Sqrt	raíz cuadrada
Sin	Seno
Cos	Coseno
Tan	tangente

Función de aptitud

Los individuos con mayor aptitud son seleccionados para aplicar las operaciones de cruce y mutación. Esta selección se basa en el valor asignado a cada individuo. La función de aptitud debe elegirse cuidadosamente según las características del problema que se busca resolver. El desempeño del modelo de clasificación de emociones depende de su capacidad para cumplir con la tarea asignada. En este contexto, la métrica de evaluación F1 resulta especialmente útil, ya que permite medir el equilibrio entre precisión y exhaustividad. Esta métrica proporciona una visión clara sobre la cantidad de errores cometidos y la calidad de la clasificación obtenida. Durante el proceso evolutivo, la métrica F1 se empleó para evaluar el rendimiento de los individuos. A continuación, se presentan las métricas utilizadas en el proceso de evaluación.

Parámetros del proceso evolutivo

Los parámetros del proceso evolutivo utilizados permiten controlar la búsqueda de soluciones mediante programación genética y se definen como: las probabilidades de que los individuos de la población se crucen y muten, el tamaño de la población y el número de generaciones. Los parámetros utilizados se muestran en la Tabla 3.27.

Tabla 3.27 Parámetros utilizados en el proceso evolutivo.

Parámetro	**Valor**
Probabilidad de cruce	70 %
Probabilidad de mutación	30 %
Tamaño de la población	100
Número de generaciones	20

La técnica de programación genética realiza una búsqueda de funciones que transforman el conjunto de datos de entrada en valores que puedan ser utilizados por la red neuronal para ser clasificados dentro de las emociones de calma, enojo, tristeza y felicidad. La Tabla 3.28 muestra las soluciones encontradas después de realizar 10 ejecuciones del proceso evoutivo.

Tabla 3.28 Mejores soluciones encontradas con programación genética después de 10 ejecuciones.

Número	Solución	Aptitud
1	$\frac{x_{15}-x_{20}}{\sin(x_7)}$ $\cos(x_{70})+\cos(x_{70})$ $\cos(x_7)\cdot\cos\left(\sqrt{x_{69}}\right)$	0.9219
2	$cos\,(x_{32}\cdot x_{63}+x_{31}+x_{63})$ $\sqrt{x_{64}+x_{69}}$ $sin\,(29+x_{70})$	0.9349
3	$\cos\,(x_5-x_{19})$ $\sqrt{x_{56}}+\left(\frac{x_{69}+x_{69}}{x_{76}-x_{40}}+x_{69}\right)$ $\frac{\sin(x_1)+(\cos(x_{70})-\sin(x_1)-x_{58})}{\cos(\cos(\tan(x_{58})))}$	0.9458
4	$\sqrt{x_{11}}-\left(\sin\left(\sqrt{x_{64}}\right)-x_{55}\right)$ $\sqrt{x_{69}}-\left(\sqrt{x_{61}-x_{68}}-x_{69}\right)$ $\frac{\tan(x_{31})-(\sqrt{\sin(x_{70})}+x_1)}{\sqrt{\sin(x_1)}}$	0.8743
5	$\left(x_{69}-\left((x_{51}-x_{11})+\tan(x_{31})+\sqrt{\cos(x_{23})}\right)\right)\cdot\cos(x_{37})$ $\cos\left(\cos\left(\cos\left(x_{41}+x_0\right)\right)\right)\cdot\sqrt{\sqrt{\cos(x_0)}}$ $\tan\left(\sqrt{x_{70}}\right)$	0.9698
6	$\tan\left(\frac{x_{71}}{x_{71}}\right)$ $\frac{\sin(x_9)\cdot\sqrt{x_9}}{\sin(x_{72}\cdot x_{32})}$ $\tan(x_{70})-(x_{14}+x_{70})$	0.8841
7	$\sqrt{x_{10}-x_{70}}$ $\sqrt{x_{69}+x_{66}}$ $(x_{51}\cdot x_{43}+\sin(x_{43}))-\tan\left(x_{70}+x_{60}\right)$	0.9357
8	$\cos\left(\sin(x_{71})\right)$ $\tan\left(x_4+x_{39}+x_{39}+x_4+x_{46}+x_4\right)-(x_{69}+x_{39})$ $\frac{x_{41}-x_9}{\sqrt{x_5}}+\cos(x_8)$	0.8934
9	$\sin\left(\sqrt{x_{75}}\right)-(x_3+x_3)$ $\sqrt{x_{10}}$ $\cos\left(\tan\left(\sqrt{x_{67}+x_{70}}\right)\right)$	0.8483
10	$\tan\left(\frac{\cos(x_{23})}{\cos(x_{23})}\right)-x_{61}-(x_{69}+x_{30})$ $\frac{\frac{\cos(x_{70})}{\cos(x_{28})}}{\cos(x_{14})+\frac{x_{25}}{x_{14}}}$	0.9580

De acuerdo a las soluciones encontradas que se muestran en la Tabla 3.28 seleccionamos la solución número 3 que se muestra en tres partes en la Figura 3.33 debido a que

involucra datos del ritmo cardíaco, imagenes y audio como se puede ver en la Tabla 3.29.

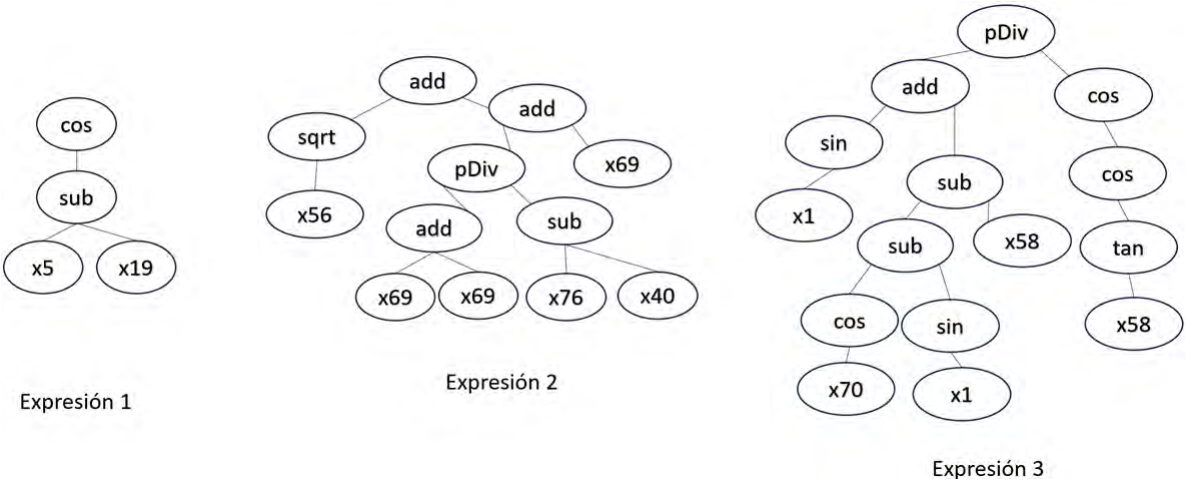

Figura 3.33 Solución multiárbol seleccionada.

La solución forma un sistema de ecuaciones en el que los datos son transformados en 3 valores. Los valores obtenidos de las ecuaciones son clasificados por la red neuronal. Las ecuaciones de la solución son las siguientes:

$$cos(x_5 - x_{19}) \tag{3.14}$$

$$\sqrt{x_{56}} + \frac{2x_{69}}{x_{76} - x_{40}} + x_{69} \tag{3.15}$$

$$\frac{sin(x_1) + (cos(x_{70}) - sin(x_1) - x_{58})}{cos(cos(tan(x_{58})))} \tag{3.16}$$

Las ecuaciones 3.14 , 3.15 y 3.16 muestran las variables utilizadas que corresponden a las características de las imágenes, audio y ritmo cardíaco del conjunto de datos. Las características incluidas en el conjunto de datos como se mencionó anteriormente forman un total de 78 pero como se muestra en la Tabla 3.29 sólo se utilizan 9 en la solución. La técnica de programación genética además de encontrar una solución en la que la red neuronal pudiera clasificar de manera correcta los datos también encontró una solución que utilizara menos características.

La ecuación 3.16 puede reescribirse eliminando el término redundante $sin(x_{x1})$.

$$\frac{cos(x_{70}) - x_{58}}{cos(cos(tan(x_{58})))} \tag{3.17}$$

Tabla 3.29 Características relevantes encontradas con programación genética en la solución.

Variable	Característica
x1	Energy
x5	spectral entropy
x19	mfcc 12
x40	delta spectral flux
x56	delta chroma 2
x58	delta chroma 4
x69	VAL
x70	ACT
x76	HR power

Redes neuronales

La red neuronal utilizada para el entrenamiento del modelo se implementó utilizando la clase MLPClassifier integrada en paquete de Python Scikit-learn versión 1.3.2 utilizando los parámetros que se muestran en la Tabla 3.30. Los parámetros de la Tabla 3.30 definen la arquitectura de la red neuronal que se muestra en la Figura 3.34 integrada por tres capas. La capa de entrada con 78 valores correspondientes a las características del conjunto de datos. La capa oculta con 7 neuronas. La capa de salida con 4 neuronas que corresponden a las emociones de calma, enojo, tristeza y felicidad del modelo de clasificación de emociones.

Tabla 3.30 Parámetros utilizados en la red neuronal.

Parámetro	Valor
activation	relu
solver	adam
hidden layer sizes	7
alpha	0.0001
batch size	16
learning rate init	0.001
learning rate	Adaptive
max iter	100000
early stopping	True
Verbose	False

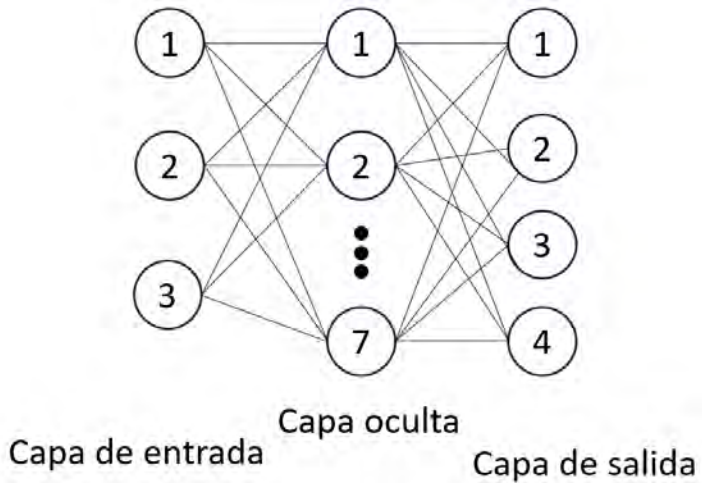

Figura 3.34 Arquitectura de la red neuronal.

La primera parte del modelo de detección de emociones con ritmo cardíaco utiliza las ecuaciones 3.14 , 3.15 y 3.16 para transformar las características de ritmo cardíaco, audio e imagen en datos que puedan clasificarse. La segunda parte utiliza la red neuronal para realizar la clasificación de las emociones. Los datos que componen la arquitectura de la red neuronal entrenada en el modelo son: en la capa de entrada con 3 neuronas, capa oculta con 7 neuronas y la capa de salida con 4 neuronas. Los pesos de la Capa de entrada a la capa oculta se muestran en la Tabla 3.31. Los valores de sesgo (bias en inglés) permiten desplazar la función de activación, lo que incrementa la flexibilidad del modelo para representar relaciones no lineales. Estos valores para la capa oculta de la red neuronal se muestran en la Tabla 3.33 y para la capa de salida en la Tabla 3.34.

Resumen del modelo de clasificación de emociones con la red neuronal:

Número de capas (input + ocultas + output): 3

Número neuronas de la capa oculta: 7

Función de activación: relu

Algoritmo de optimización: adam

Tabla 3.31 Pesos de la red neuronal de la Capa de entrada hacia la Capa oculta

Peso	Neurona 1	Neurona 2	Neurona 3
1	0.64405733	-0.14208533	-0.0977711
2	-0.24284489	0.10092873	-1.35324943
3	0.03965248	-0.12024884	0.03855283
4	-0.02197124	-0.08366381	0.01602587
5	-0.28596627	0.31922585	0.59422849
6	0.75126126	0.29798675	-1.08084747
7	-0.65468053	-0.28302555	0.75391099

Tabla 3.32 Pesos de la red neuronal de la Capa oculta hacia la Capa de salida

Peso	Neurona 1	Neurona 2	Neurona 3	Neurona 4
1	-0.86781783	-0.44895387	-0.84128081	0.56002964
2	1.01039109	-0.14865216	-0.73730856	0.27657281
3	-1.21374017	0.520387	-1.30183425	0.87971143
4	-0.09312881	-0.00507135	0.04669369	-0.02785335
5	-0.66350989	0.26376089	0.43333401	-0.4004606
6	0.05632388	-1.07785027	-0.38647497	-0.44462713
7	-0.06290107	0.89685162	-1.10986615	-0.88035337

Tabla 3.33 Sesgo de la red neuronal de la Capa oculta

Sesgo	Valor
1	0.88397924
2	0.11559708
3	0.83349834
4	0.10521434
5	0.56641224
6	0.21797508
7	1.06211503

Tabla 3.34 Sesgo de la red neuronal de la Capa de salida

Sesgo	**Valor**
1	-0.35236751
2	-0.40097197
3	-0.00841133
4	0.14402895

Capítulo 4

Pruebas y resultados obtenidos

En esta sección se presentan los resultados de las pruebas realizadas a los modelos en los conjuntos de prueba para detección de emociones con voz y el conjunto de prueba para detección de emociones con el ritmo cardíaco.

4.1. Modelo de detección de emociones en la voz

Los resultados de las pruebas realizadas sobre el modelo que utiliza el algoritmo de máquinas de soporte vectorial y el modelo con la red neuronal usando los datos de prueba se muestran en la Tabla 4.1 con las métricas correspondientes. Estas pruebas fueron realizadas sobre un conjunto de datos de características de audio de tamaño 1440 por 428. El conjunto de datos se dividió en 5 partes utilizando la técnica de validación cruzada.

Tabla 4.1 Evaluación de validación cruzada de Máquinas de soporte vectorial y red neuronal

	SVM			**Red neuronal**		
	Exactitud	**Precisión**	**Sensitividad**	**Exactitud**	**Precisión**	**Sensitividad**
Partición 1	0.694	0.691	0.692	**0.680**	0.673	0.661
Partición 2	**0.739**	**0.739**	**0.739**	0.614	0.619	0.606
Partición 3	0.729	0.729	0.718	0.673	**0.676**	**0.670**
Partición 4	0.739	0.732	0.736	0.652	0.651	0.631
Partición 5	0.718	0.718	0.720	0.649	0.639	0.645
Media	0.724	0.722	0.721	0.654	0.652	0.642

Tabla 4.2 Métricas Máquinas soporte vectorial sobre cada emoción

Emoción	Presición	Sensitividad
Neutral	0.583	0.736
Calm	**0.846**	**0.846**
Happy	0.675	0.710
Sad	**0.742**	0.666
Angry	0.690	0.763
Fearful	0.651	0.736
Disgust	**0.833**	**0.769**
Surprised	**0.896**	**0.684**

Las métricas, en la Tabla 4.2 brindan información sobre qué tan bien el clasificador identifica cada emoción, resaltando su efectividad en la diferenciación entre ellas. El modelo con el algoritmo de máquinas de soporte vectorial muestra mejor rendimiento en el Fold 2 y sobre estos datos se calcula su matriz de confusión que se muestra en la Figura 4.1. El clasificador desarrollado con el algoritmo de máquinas de soporte vectorial tiene un buen desempeño en Precisión al identificar las emociones Calma, Disgusto y Sorpresa. Sin embargo, la emoción Neutral presenta resultados relativamente bajos, lo que creemos que podría deberse a una confusión con Calma. Si bien las emociones Felicidad, Miedo y Enojo muestran puntuaciones moderadas, ajustes adicionales podrían mejorar su clasificación. La emoción "Tristeza"presenta un desempeño razonablemente bueno.

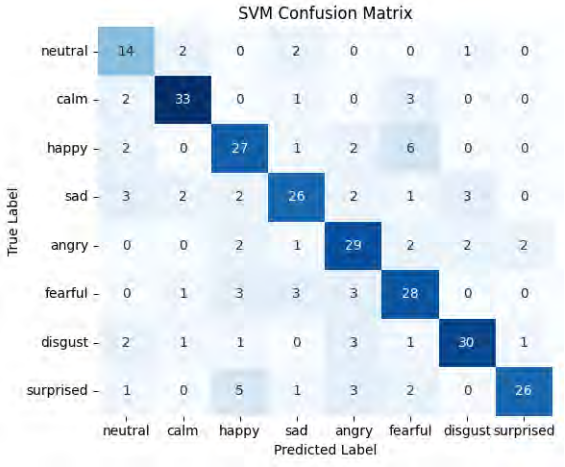

Figura 4.1 Matriz de confusión para máquinas de soporte vectorial.

Tabla 4.3 Métricas de la Red neuronal sobre cada emoción

Emoción	Presición	Sensitividad
Neutral	0.538	0.35
Calm	0.680	0.842
Happy	0.577	0.684
Sad	0.676	0.589
Angry	**0.764**	0.684
Fearful	0.625	0.641
Disgust	**0.731**	**0.789**
Surprised	**0.794**	**0.710**

Las métricas que se calcularon sobre el Fold 1 mostradas en la Tabla 4.3 presentan que las emociones Neutral y Felicidad obtuvieron las puntuaciones más bajas, mientras que Calma, Tristeza y Miedo también fueron inferiores en comparación con Enojo, Disgusto y Sorpresa. Aunque los resultados mostrados en la matriz de confusión de la Figura 4.2 generales no son óptimos, el clasificador basado en red neuronal perceptrón muestra un patrón de rendimiento distinto en comparación con las máquinas de soporte vectorial (SVM), demostrando una mejor precisión en un conjunto diferente de emociones.

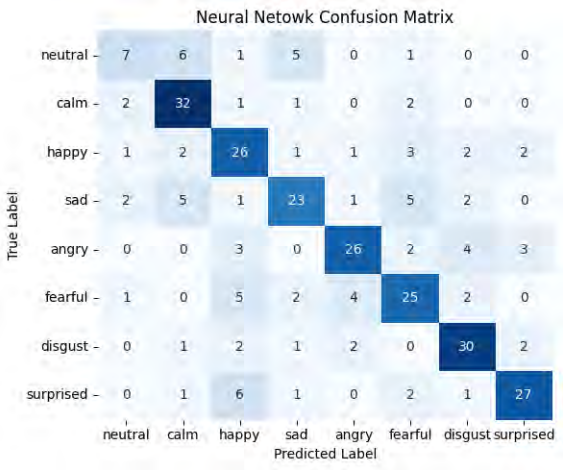

Figura 4.2 Matriz de confusión para la red neuronal.

Los resultados obtenidos muestran que el modelo construido con el algoritmo de máquinas de soporte vectorial (SVM) tuvo un mejor desempeño en comparación con el modelo de la red neuronal. Esto sugiere que este algoritmo es más efectivo cuando se aplica a un conjunto de datos con un gran número de variables. Sin embargo, la Figura 4.2 sugiere que podría ser interesante explorar la posibilidad de aplicar alguna técnica de selección de

características para reducir la dimensionalidad y verificar si los resultados mejoran aún más.

Por otro lado, aunque la red neuronal también muestra un buen rendimiento con instancias de alta dimensión y demuestra su capacidad para encontrar modelos que clasifiquen señales de audio, su desempeño no alcanzó al de las máquinas de soporte vectorial. Para mejorar los resultados de la red neuronal, se podría considerar aumentar el número de capas, sin convertirla en una red profunda, ya que este tipo de redes requiere más datos y es computacionalmente costoso de entrenar.

Los modelos desarrollados con el conjunto de datos RAVDESS nos sirvieron como una primera aproximación para la detección de emociones de una persona. Sin embargo, los datos utilizados para su desarrollo provienen de un conjunto de datos con voces actuadas y esto resulta dificil de replicar con personas que no son actores. El trabajo realizado hasta este momento nos sirvió de guía para entender el proceso de entrenamiento y evaluación de modelos de clasificacion de emociones.

4.2. Modelo de detección de emociones en ritmo cardíaco

En la detección de emociones con ritmo cardíaco y estímulos audiovisuales los resultados obtenidos tras aplicar la técnica de programación genética al conjunto de datos y entrenar la red neuronal para clasificar las emociones de una persona se presentan en la siguiente parte. El conjunto de datos, compuesto por información de ritmo cardíaco y estímulos audiovisuales, se dividió en un tamaño de 78×599 para el conjunto de entrenamiento y 78×150 para el conjunto de prueba. Los datos fueron obtenidos de una persona que participó en el experimento de identificación de emociones. El modelo desarrollado logró una precisión del 95 % en los datos de prueba y 94 % datos de prueba para la clasificación de las emociones de calma, enojo, felicidad y tristeza. La cantidad de instancias correctamente clasificadas por el modelo puede observarse en las Figuras 4.3 y 4.4, lo que permite evaluar su desempeño.

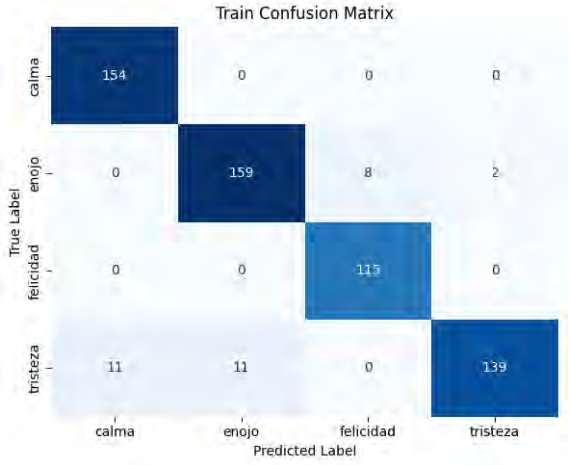

Figura 4.3 Matriz de confusión con datos de entrenamiento.

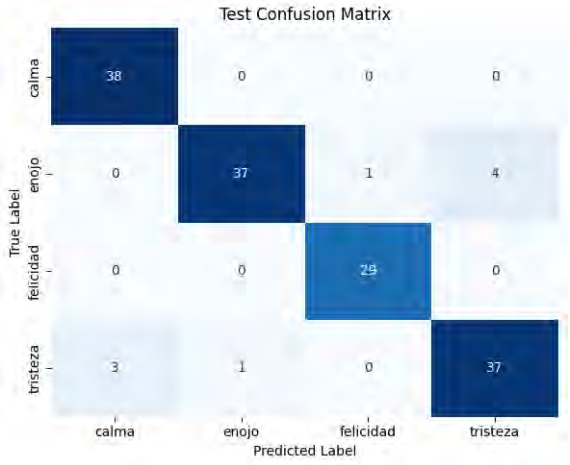

Figura 4.4 Matriz de confusión con datos de prueba.

Las emociones de enojo y tristeza resultaron con algunos fallos en su clasificación como se muestra en las Figuras 4.3 y 4.4. Los datos que se utilizan para el cálculo de las funciones utilizan características de audio, imagen y ritmo cardíaco. El clasificador construido a partir de estos datos nos indica que para el caso de las emociones de enojo y tristeza los valores obtenidos son similares después de realizar los cálculos. Las imágenes y audio pueden tener valores similares entre estas dos emociones. El ritmo cardíaco de la persona utilizado en el conjunto de datos también puede tener valores similares entre las emociones de enojo y tristeza. El resultado obtenido por el clasificador nos da un indicio de que para construir un modelo que pueda clasificar más emociones sea necesario involucrar más datos.

El modelo desarrollado con un enfoque híbrido entre programación genética y redes neuronales nos da como resultado no solo una disminución de variables involucradas como

ya se habia mencionado, si no que también el espacio al que transforma las instancias hacen que una red neuronal sencilla como la construida aquí pueda realizar la clasificación de manera correcta, las Figuras 4.5 y 4.6 muestran como quedaron las instancias de cada emoción despues de aplicar la técnica de programación genética.

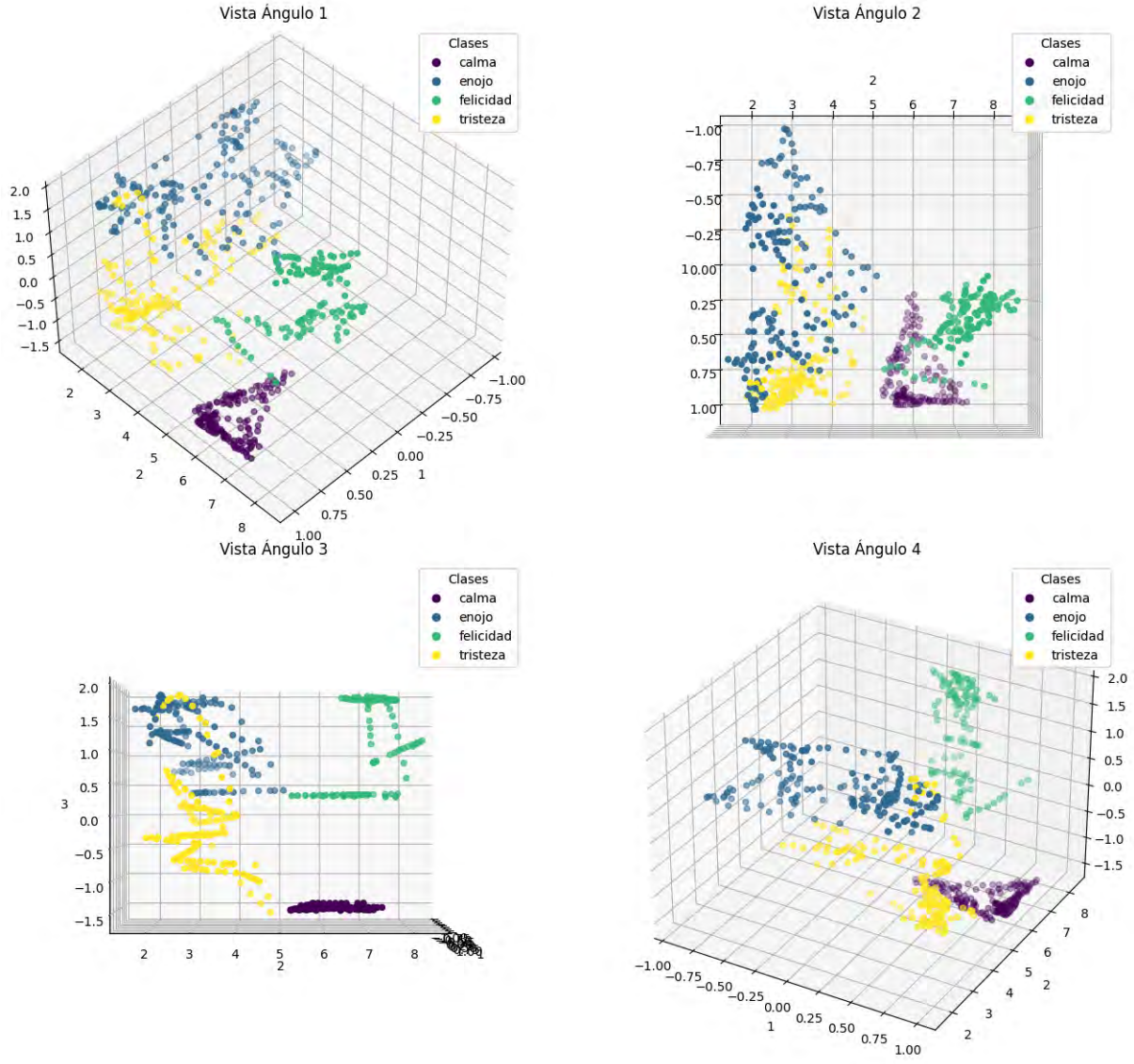

Figura 4.5 Representación gráfica de instancias de entrenamiento.

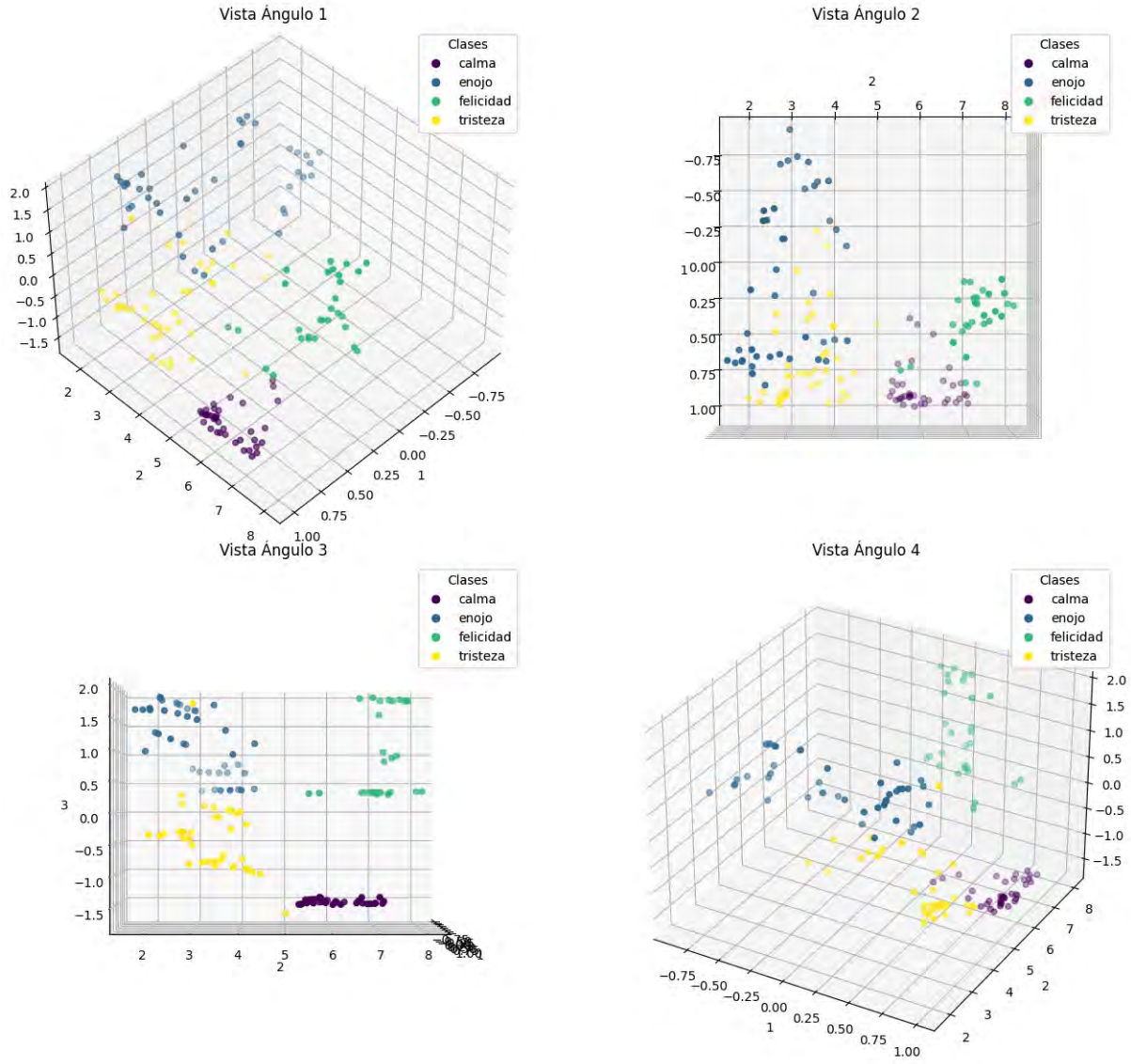

Figura 4.6 Representación gráfica de instancias de prueba.

La Figura 4.7 muestra cómo fue evolucionando el modelo basado en programación genética a lo largo de las generaciones. En la gráfica, el eje X representa el número de generaciones y el eje Y muestra los valores de aptitud. Se incluyen tres elementos clave: la aptitud del mejor individuo en cada generación, el promedio de los mejores individuos obtenidos en distintas corridas, y la desviación estándar asociada a ese promedio. Esta representación permite observar no solo cómo mejora la calidad de las soluciones con el tiempo, sino también qué tan consistentes fueron los resultados entre corridas. En general, se aprecia una tendencia ascendente tanto en el mejor individuo como en el promedio, lo que indica que el proceso evolutivo fue efectivo al refinar las soluciones generación tras

generación.

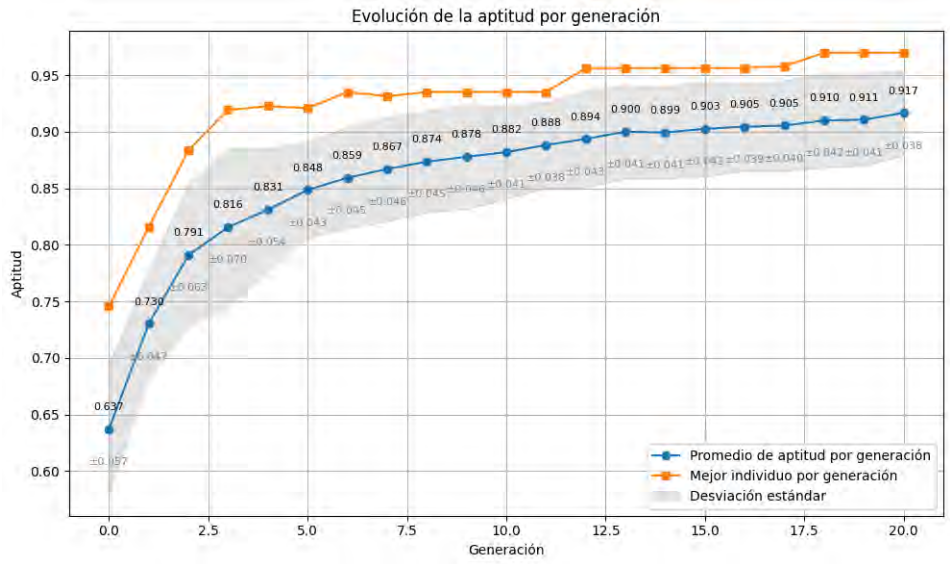

Figura 4.7 Gráfica de convergencia del proceso evolutivo.

El modelo de clasificación se evaluó aplicando la técnica de validación cruzada, la cual permitió evaluar su desempeño de manera más robusta al dividir el conjunto de datos en múltiples particiones de entrenamiento y prueba. Para cada partición, se calcularon las métricas estándar de evaluación: exactitud, precisión, sensibilidad y F1-score. Estas métricas proporcionan una visión integral del comportamiento del modelo, permitiendo identificar tanto su capacidad para predecir correctamente las clases como su consistencia al manejar falsos positivos y falsos negativos. En general, los resultados que se pueden observar en la Tabla 4.4 mostraron un buen equilibrio entre las métricas, lo que indica que el modelo es confiable y tiene un rendimiento estable al enfrentarse a datos no vistos durante el entrenamiento.

Tabla 4.4 Resultado de validación cruzada utilizando el modelo de programación genética con redes neuronales.

Partición	**Exactitud**	**Presición**	**Sensitividad**	**F1**
Partición 1	**0.9667**	**0.9677**	**0.9667**	**0.9665**
Partición 2	0.9200	0.9237	0.9200	0.9188
Partición 3	0.9333	0.9347	0.9333	0.9319
Partición 4	0.9600	0.9620	0.9600	0.9595
Partición 5	0.9396	0.9410	0.9396	0.9386

Interpretación del modelo

La interpretabilidad de un modelo (Gao y Guan, s.f.) se refiere a la capacidad de comprender, explicar y justificar el comportamiento interno del modelo y las decisiones que toma a partir de los datos de entrada. Un modelo interpretable permite a los usuarios, desarrolladores y expertos del dominio entender cómo se generan las predicciones, identificar qué características influyen más en los resultados, y detectar posibles sesgos o errores en el proceso de aprendizaje.

Un modelo que utiliza programación genética es considerado interpretable debido a la forma en que representa sus soluciones: expresiones simbólicas o árboles sintácticos compuestos por operaciones matemáticas y funciones definidas explícitamente. A diferencia de modelos de caja negra, las soluciones generadas por programación genética pueden analizarse y entenderse directamente, ya que muestran de manera clara cómo se combinan las variables de entrada para generar una salida. Esta estructura transparente permite identificar patrones, relaciones y reglas relevantes en los datos.

Retomando las expresiones que seleccionamos como solución de programación genética para el modelo y que se muestran a continuación.

$$x = \cos(x_5 - x_{19}) \tag{4.1}$$

$$y = \sqrt{x_{56}} + \left(\frac{2x_{69}}{x_{76} - x_{40}} + x_{69} \right) \tag{4.2}$$

$$z = \frac{\cos(x_{70}) - x_{58}}{\cos(\cos(\tan(x_{58})))} \tag{4.3}$$

La Figura 4.5 y la Figura 4.6 muestran como quedan transformados los datos de entrada en un espacio de 3 dimensiones. Los valores del eje x se obtienen mediante la expresión 4.1, el eje y esta dado por la expresión 4.2 y el eje z se calcula mediante la expresión 4.3. Los rangos de los valores obtenidos por las expresiones mencionadas al analizarse se pueden resumir en la Tabla 4.5 en donde se muestra que rangos corresponden a cada emoción.

Tabla 4.5 Rangos de valores en los ejes x, y y z para cada emoción de acuerdo a las expresiones encontradas con programación genética.

Emoción	Min x	Max x	Min y	Max y	Min z	Max z
Calma	0.2135	0.9982	5.2472	7.3129	-1.4641	-1.3301
Enojo	-0.9253	0.9578	1.6829	5.0902	0.3630	1.8515
Felicidad	0.0791	0.8330	5.2850	8.3523	0.2900	1.8510
Tristeza	-0.3330	0.9783	1.9764	5.0214	-1.6344	1.8507

Los valores de Tabla 4.5 nos indican que si al aplicar las expresiones a los datos de entrada la red neuronal propuesta clasificará la emoción de la persona como calma, enojo, felicidad o tristeza.

Las variables de entrada utilizados para el mapeo de los datos en tres dimensiones, según las variables empleadas por las expresiones 4.1, 4.2 y 4.3 se describen a contiuación:

x1 - Energía (Energy): La energía del sonido es una medida de qué tan fuerte o intensa es una señal de audio en un periodo de tiempo. Cuanto mayor sea la energía, más fuerte o potente se percibe el sonido; si la energía es baja, el sonido es suave o casi silencioso. Durante el grito, la energía del sonido es alta porque las ondas son grandes. Durante el susurro, la energía es baja porque las ondas son pequeñas. Lo mismo sucede con la musica la energía del audio cambia a lo largo del tiempo según los instrumentos, el ritmo y la intensidad del momento.

x5 - Entropía espectral (spectral entropy): La entropía espectral mide cuán desordenado o impredecible está el contenido de frecuencias de un sonido. Cuando se tiene baja entropía espectral el sonido suena más puro o tonal (como una nota de piano). La alta entropía espectral suena más ruidoso o caótico (como una multitud o ruido blanco).

x19 - mfcc 12: Los Coeficientes Cepstrales de Mel (MFCC) son una forma de representar cómo percibe el oído humano el sonido. Cada coeficiente captura una característica diferente de la forma del espectro del sonido. Los primeros MFCCs capturan la energía global y forma general del espectro. Los coeficientes medios (como el 12) capturan detalles más finos del timbre. El MFCC 12 puede cambiar, ayudando a distinguir entre emociones solo por el tono y matices de la voz.

x40 - Cambio en el Flujo Espectral (delta spectral flux): El flujo espectral mide qué tanto cambian las frecuencias de un sonido de un instante al siguiente. El cambio en el flujo espectral es útil para detectar momentos de transición rápida o emoción intensa, como sustos, gritos, o tensión en una grabación. Ayuda a capturar la dinámica del cambio emocional en el audio.

x56 - Cambio en el Segundo Componente de Chroma (delta chroma 2): La representación chroma indica cuánta energía tiene cada nota musical (Do, Do#, Re, etc.) en un fragmento de audio, sin importar la octava. El Chroma 2 corresponde a la nota Re (D). El delta chroma 2 mide qué tanto cambia la intensidad de la nota Re a lo largo del tiempo. Es útil para detectar cuándo esa nota aparece, se refuerza o desaparece en la música.

x58 - Cambio en el Cuarto Componente de Chroma (delta chroma 4): El Chroma 4 representa la nota Mi (E). El delta chroma 4 mide cuánto cambia la presencia o intensidad de la nota Mi con el tiempo. Es decir, detecta si la nota Mi aparece, desaparece o cambia su fuerza en la música o el habla.

x69 - Valencia (VAL): La valencia es una medida que indica qué tan positiva o negativa es una emoción. Una valencia alta significa que la emoción es positiva (como alegría o amor). Una valencia baja significa que la emoción es negativa (como tristeza o enojo). La

valencia puede ser útil para clasificar emociones en voz o música o imágenes, permitiendo diferenciar entre sonidos o imágenes positivas y agradables vs. negativos o desagradables.

x70 - Activación (ACT): La activación mide qué tan intensa o energética es una emoción, sin importar si es positiva o negativa. La activación es alta con emociones intensas como enojo, euforia, miedo. La activación baja con emociones suaves como tristeza, calma o relajación.

x76 - Potencia del ritmo cardíaco (HR power): La potencia del ritmo cardíaco se refiere a la cantidad de energía contenida en la señal del ritmo cardíaco. Es una forma de medir qué tan fuertes o pronunciadas son las variaciones en los latidos del corazón en un período de tiempo. Una mayor potencia puede reflejar una actividad fisiológica intensa, y una menor potencia puede indicar un estado de reposo o relajación. Este valor es útil para detectar reacciones emocionales fisiológicas, ya que emociones como miedo, estrés o entusiasmo afectan el corazón, y eso se refleja en la potencia de la señal.

La Tabla 4.6 muestra los rangos de valores de cada variable.

Tabla 4.6 Rangos de valores de cada variable.

Variable	Nombre	Valor mínimo	Val máximo
x1	Energy	0.0003	0.0743
x5	spectral entropy	0.0980	2.7626
x19	mfcc 12	-0.6691	0.4293
x40	delta spectral flux	-0.1000	0.0005
x56	delta chroma 2	-0.0116	0.0124
x58	delta chroma 4	-0.0015	0.0015
x69	VAL	1.67	8.34
x70	ACT	3.63	7.29
x76	HR power	4720.1	7564.3

Eje X $x = \cos(x_5 - x_{19})$

Este eje utiliza la diferencia entre la entropía espectral (x_5) y el coeficiente 12 de la frecuencia de Mel (x_{19}). La entropía espectral se puede ver por la Tabla 4.6 que sus valores son más altos, por lo tanto, domina la expresión. Los valores altos de x nos indican que hay una baja diferencia entre x_5 y x_{19}, lo cual está asociado con sonidos estables y armónicos. Esto se refleja particularmente en la emoción calma, donde la entropía espectral tiende a ser baja. Por el contrario, valores bajos de x se observan en la emoción de enojo, en donde los sonidos pueden ser más ruidosos. En el caso de la emoción de felicidad los valores tienen una entropía espectral media, que representa sonidos agradables, pero sin ser caóticos.

Eje Y $y = \sqrt{x_{56}} + (\frac{2x_{69}}{x_{76} - x_{40}} + x_{69})$

Esta dimensión está fuertemente influida por la variable x_{69} correspondiente a la valencia emocional, apareciendo de forma aditiva y racional. El impacto de otras variables, como la potencia del ritmo cardíaco (x_{76}) y la variable del cambio del componente 2 de croma (x_{56}), es pequeño. Así, valores altos de y corresponden a emociones positivas como felicidad y calma, mientras que valores bajos están asociados a emociones negativas como enojo y tristeza, caracterizadas por valencia baja, independientemente de la activación fisiológica. Esto quiere decir también que apesar de que esta dimensión toma en cuenta la potencia del ritmo cardíaco, al hacer el cálculo esta variable no es tan relevante como la valencia.

La expresión de esta dimensión puede simplificarse. El primer término, $\sqrt{x_{56}}$, alcanza un valor máximo de 0.11, al calcularse la raíz cuadrada de 0.0124; por lo tanto, esta variable, al aportar un valor reducido, puede omitirse. En cuanto a los valores de la segunda parte de la expresión, puede decirse que la diferencia $x_{76} - x_{40}$ es prácticamente igual al valor de x_{76}, ya que el rango en el que varía x_{40} es limitado. El cociente $\frac{2x_{69}}{x_{76}}$ también toma valores bajos, como 0.00044 (resultado de $\frac{2 \cdot 1.67}{7564.3}$) y 0.0035 (resultado de $\frac{2 \cdot 8.34}{4720.1}$), por lo que esta operación también puede descartarse. Después de realizar este análisis, la nueva expresión se presenta en la ecuación 4.4.

$$y = x_{69} \tag{4.4}$$

Eje Z $z = \frac{cos(x_{70}) - x_{58}}{cos(cos(tan(x_{58})))}$

La variable dominante en esta expresión es la activación (x_{70}), ya que el resultado del cálculo del coseno determina en gran medida el valor del numerador. La variable de la diferencia del cuarto valor del croma (x_{58}) modifica levemente el resultado debido a que el rango de valores que tiene son pequeños. Este eje distingue claramente entre estados calmados/tristes (valores negativos de z) y estados de felicidad/enojo (valores positivos), reflejando la variabilidad emocional respecto a la activación fisiológica.

La variable x_{58} tiene valores muy pequeños, entre -0.0015 y 0.0015. Cuando un número es tan pequeño, su tangente es prácticamente el mismo número $tan(x_{58}) \approx x_{58}$. Por lo tanto $cos(tan(x_{58})) \approx cos(x_{58})$, el coseno también se puede aproximar como $cos(x_{58}) \approx 1$. Finalmente, si tomamos el coseno de ese resultado, obtenemos algo muy cercano a un número constante, aproximadamente $cos(cos(tan(x_{58}))) \approx cos(1) \approx 0.54$. El denominador cambia muy poco, sin importar el valor exacto de x_{58}. Por eso, podemos tratarlo como una constante.

$$z = cos(x_{70}) - x_{58} \tag{4.5}$$

Modelo simplificado

El modelo que transforma los datos de características al espacio de tres dimensiones ahora queda de la siguiente forma:

$x = \cos(x_5 - x_{19})$

$y = x_{69}$

$z = cos(x_{70}) - x_{58}$

La relación entre los valores de las variables del modelo y las diferentes emociones se muestran en la Tabla 4.7.

Tabla 4.7 Resumen de valores de las variables y su relacion con las emociones.

Emoción	**Variables**	**Características de la emoción**
Calma	x_5 bajo, x_{69} alto, x_{70} alto	Sonidos poco caóticos (entropía baja), alta valencia emocional, pero con activación física alta y estable (z negativo).
Enojo	x_5 alto, x_{69} bajo, x_{70} medio	Sonidos caóticos, valencia baja, pero activación media, lo que representa emociones con poco agrado.
Felicidad	x_{69} alto, x_{70} medio-bajo, x_5 medio	Alta valencia, activación estable, y estructuras acústicas balanceadas. Sonidos estimulantes pero agradables.
Tristeza	x_{69} bajo, x_{70} bajo, x_5 medio-bajo	Baja valencia, alta activación (pero sin agrado), con sonidos poco ruidosos, planos o apagados.

En la Figura 4.8 se muestra una comparación visual entre los datos proyectados en el espacio de tres dimensiones utilizando la versión original y la versión simplificada del sistema de ecuaciones generado mediante programación genética. Esta representación permite observar cómo la simplificación de las expresiones no afecta la distribución de los datos en el espacio tridimensional, conservando en gran medida la estructura general del modelo, lo que valida la efectividad de la reducción sin pérdida significativa de información relevante.

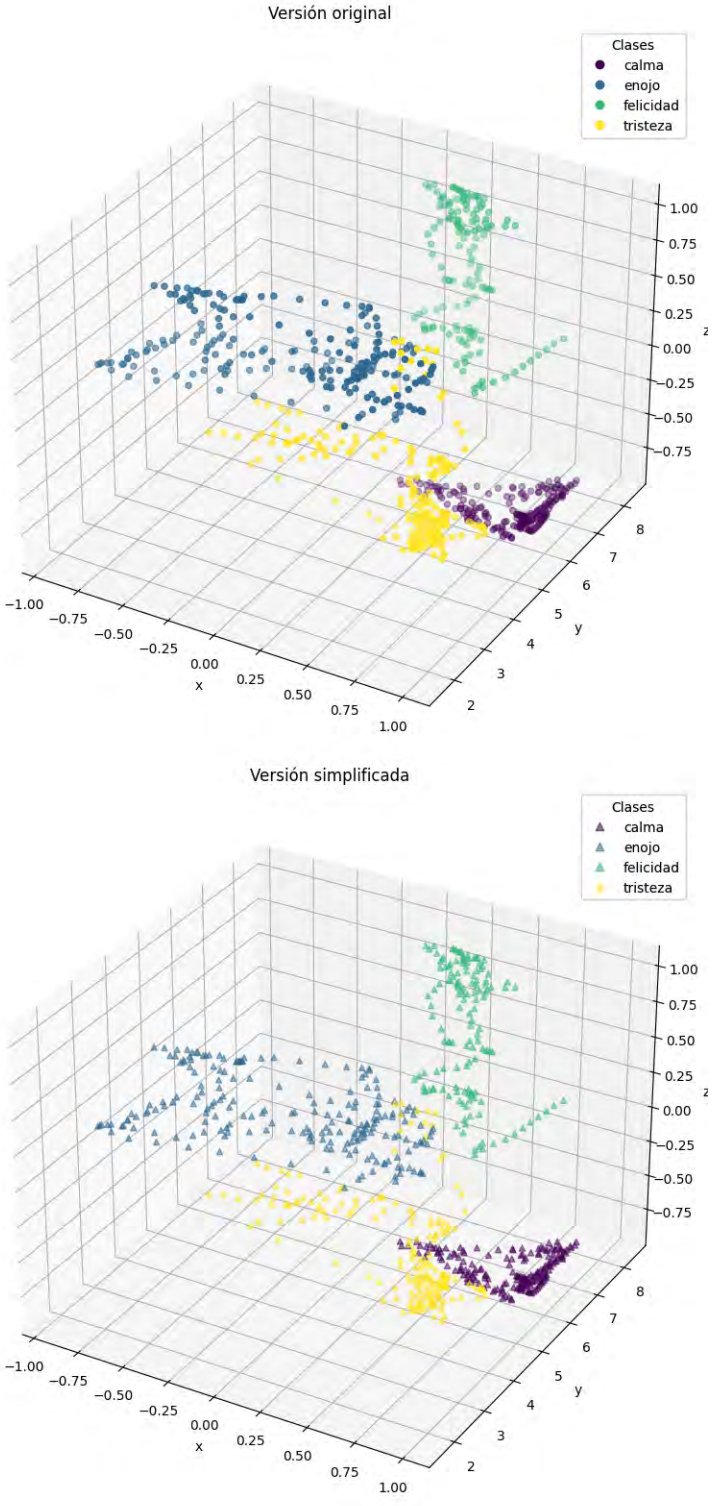

Figura 4.8 Espacio de 3 dimensiones formado por versión original y versión simplificada del sistema de ecuaciones.

En la Tabla 4.8 se presentan los resultados de validación cruzada para ambas versiones del modelo: la original y la simplificada. Como se puede observar, la versión simplificada conserva un desempeño muy similar al modelo original en todas las particiones y métricas evaluadas (exactitud, precisión, sensibilidad y F1). Esta similitud en los resultados confirma que la simplificación de las expresiones no compromete significativamente la calidad del modelo, lo que valida su uso como una alternativa más compacta y eficiente sin pérdida notable de precisión.

Tabla 4.8 Comparación de resultados de aplicar validación cruzada entre el modelo original y el modelo simplificado.

Partición	**Original**				**Simplificado**			
	Exact.	Prec.	Sens.	F1	Exact.	Prec.	Sens.	F1
Partición 1	**0.9667**	**0.9677**	**0.9667**	**0.9665**	**0.9600**	**0.9615**	**0.9600**	**0.9598**
Partición 2	0.9200	0.9237	0.9200	0.9188	0.9467	0.9471	0.9467	0.9458
Partición 3	0.9333	0.9347	0.9333	0.9319	0.9200	0.9193	0.9200	0.9187
Partición 4	0.9600	0.9620	0.9600	0.9595	0.9333	0.9393	0.9333	0.9314
Partición 5	0.9396	0.9410	0.9396	0.9386	0.9329	0.9354	0.9329	0.9312

Con base en los resultados obtenidos y presentados en las gráficas y tablas comparativas, se puede concluir que el modelo propuesto logró clasificar adecuadamente las emociones inducidas a partir de estímulos audiovisuales. La comparación entre la versión original y la versión simplificada del sistema de ecuaciones mostró diferencias mínimas en las métricas de desempeño, lo cual valida el uso de expresiones reducidas sin comprometer la precisión del modelo. Además, el análisis de las variables asociadas a cada emoción (Tabla 4.7) revela patrones consistentes entre las características acústicas y las respuestas emocionales, como el vínculo entre la entropía del sonido (x_5) y la valencia emocional, o entre x_{69} y el nivel de activación. Estos resultados no solo respaldan la validez del enfoque híbrido basado en programación genética y redes neuronales, sino que también ofrecen interpretaciones significativas sobre cómo ciertos atributos del estímulo se relacionan con estados emocionales específicos, fortaleciendo así la utilidad del modelo tanto a nivel predictivo como explicativo.

Capítulo 5

Conclusiones y trabajos futuros

El objetivo general de este trabajo de investigación fué el de desarrollar un modelo computacional basado en aprendizaje automático y programación genética para la detección de emociones en una persona a partir de su frecuencia cardíaca, con el fin de mejorar la interacción con sistemas inteligentes en distintos contextos como educación, salud y tecnología asistiva.

5.1. Conclusiones generales

A partir del desarrollo del presente trabajo, se lograron alcanzar los objetivos planteados, los cuales permitieron explorar distintas aproximaciones para la identificación de emociones humanas a partir de señales fisiológicas y de voz. Los resultados obtenidos muestran que tanto los métodos de aprendizaje automático como los enfoques evolutivos aplicados a datos experimentales controlados pueden facilitar la detección de emociones con niveles de precisión aceptables. A continuación, se presentan las principales conclusiones de esta investigación:

1. **Diseñar y ejecutar un experimento controlado, para construir un conjunto de datos etiquetado que relacione señales de ritmo cardíaco con respuestas emocionales, con el uso de estímulos audiovisuales seleccionados para inducir emociones específicas en condiciones experimentales reproducibles.**

   Se diseñó y ejecutó con éxito un experimento controlado que permitió inducir emociones mediante estímulos audiovisuales, registrando señales de ritmo cardíaco. Como resultado, se construyó un conjunto de datos etiquetado que capturó de manera consistente las variaciones fisiológicas asociadas a distintas respuestas emocionales. Este conjunto sirvió como base para entrenar y evaluar modelos de aprendizaje automático orientados a la clasificación de emociones, confirmando la viabilidad de emplear señales fisiológicas en estudios de detección de emociones.

2. **Desarrollar un modelo de detección automática de emociones en la voz, con el fin de identificar el estado emocional de una persona a partir de sus expresiones acústicas, utilizando técnicas de aprendizaje automático y redes neuronales entrenadas con características extraídas de un conjunto de datos de audio previamente etiquetado.**

   El análisis demostró que los modelos de inteligencia artificial aplicados a datos de voz permiten identificar emociones con un grado de precisión aceptable. A través de la comparación entre una red neuronal y un clasificador SVM, se evidenció que el desempeño puede variar en función de las emociones a clasificar y la complejidad del modelo. Aunque el reconocimiento emocional por voz mostró limitaciones en ciertas categorías, ofreció un enfoque complementario que aporta valor en contextos donde los datos fisiológicos no están disponibles.

3. **Desarrollar un modelo que utilice estímulos audio visuales y el ritmo cardíaco de una persona para identificar sus emociones utilizando programación genética y redes neuronales.**

   El modelo desarrollado integró programación genética y redes neuronales para identificar emociones a partir del ritmo cardíaco y estímulos audiovisuales. La programación genética permitió identificar patrones relevantes entre los estímulos aplicados y las respuestas fisiológicas, generando representaciones que reflejan el estado emocional del sujeto. Estas representaciones fueron utilizadas como entrada para una red neuronal, lo que permitió que el modelo tuviera un buen desempeño en la clasificación de emociones, validando así la efectividad del enfoque híbrido propuesto.

4. **Desarrollar un modelo híbrido que combine programación genética y redes neuronales para detectar emociones a partir del ritmo cardíaco y estímulos audiovisuales.**

   El modelo híbrido desarrollado combinó programación genética y redes neuronales para procesar señales fisiológicas junto con estímulos audiovisuales. Esta integración permitió transformar los datos de manera eficiente, destacando las características más relevantes para la clasificación de emociones. La combinación de ambas técnicas no solo mejoró el rendimiento del sistema, sino que también demostró ser una alternativa robusta para detectar emociones a partir de datos multimodales.

5. **Evaluar y comparar el desempeño de los modelos desarrollados, para determinar su capacidad de detección emocional en la voz y ritmo cardíaco bajo estímulos audiovisuales con el uso de métricas estándar de clasificación aplicadas a los datos obtenidos en el experimento controlado.**

   El enfoque propuesto, que combina programación genética con modelos de inteligencia artificial, logró transformar adecuadamente los datos de entrada para facilitar su clasificación. Esta metodología híbrida demostró ser eficaz al reducir la dimensionalidad de los datos y resaltar las variables más relevantes, permitiendo que incluso redes

neuronales sencillas alcanzaran altos niveles de precisión. Los resultados obtenidos validan la utilidad de la programación genética como técnica de preprocesamiento orientada a la optimización del rendimiento en tareas de clasificación de emociones.

Los resultados obtenidos en esta investigación permiten concluir que es posible identificar emociones humanas de forma automática mediante el análisis de señales fisiológicas como el ritmo cardíaco utilizando inteligencia artificial. La construcción de un conjunto de datos controlado, el análisis del reconocimiento por voz, y el uso de programación genética como técnica de transformación de datos, contribuyeron al desarrollo de modelos de clasificación robustos y precisos. La propuesta metodológica basada en programación genética no solo optimiza el proceso de aprendizaje de los modelos, sino que también abre nuevas posibilidades para el diseño de sistemas inteligentes capaces de interpretar estados emocionales en contextos reales. Este trabajo sienta las bases para futuras investigaciones orientadas a mejorar la detección multimodal de emociones y su aplicación en entornos interactivos, de salud o de asistencia personalizada.

Las emociones varían de persona a persona. Los estímulos que pueden utilizarse para provocarlas afectan de manera variada ya que las personas se ven influenciadas por factores como la cultura, edad y experiencia. Por lo tanto, el desarrollo de un modelo que se pueda aplicar de manera general es difícil de generar. Sin embargo, lo que se puede generar es una metodología para poder desarrollar estos modelos de manera personalizada.

El modelo desarrollado permite utilizar diferentes audios e imágenes que presenten valores similares en las características identificadas como relevantes por la programación genética, lo cual abre la posibilidad de integrar nuevos estímulos al sistema sin necesidad de rediseñar completamente el modelo. Esta capacidad de generalización se debe a que el modelo no depende de estímulos específicos, sino de las propiedades cuantificables de los mismos, como la entropía del sonido, la valencia emocional o el nivel de activación. Gracias a ello, es posible ampliar gradualmente el conjunto de estímulos utilizados para la detección de emociones, manteniendo la coherencia con los patrones ya aprendidos y favoreciendo la adaptabilidad del sistema a distintos contextos y poblaciones.

5.2. Trabajos futuros

Las emociones de las personas, al ser muy variadas, plantean un desafío adicional para su análisis automático; por ello, un trabajo futuro podría centrarse en mejorar la interpretabilidad de los modelos de detección de emociones, especialmente aquellos que utilizan programación genética. Comprender cómo el modelo llega a una determinada clasificación emocional permitiría aumentar la confianza en su uso, particularmente en contextos sensibles como la salud o la educación.

El trabajo presentado muestra que los dispositivos inteligentes no invasivos pueden ser utilizados eficazmente para la detección de emociones en tiempo real. Sin embargo,

la generalización del modelo a diferentes contextos y usuarios sigue siendo un desafío, ya que el modelo fue entrenado y evaluado con un número limitado de participantes en condiciones controladas. La incorporación de datos de diferentes fuentes fisiológicas podría mejorar la efectividad y robustez del sistema.

Este trabajo representa el inicio de un proyecto más amplio cuyo objetivo es desarrollar un modelo capaz no solo de detectar, sino también de inducir cambios en las emociones de una persona. Para lograrlo, es fundamental avanzar hacia modelos personalizados que se adapten a las respuestas fisiológicas individuales (como rl ritmo cardíaco), permitiendo seleccionar audios e imágenes con características similares a las identificadas por la programación genética como relevantes para cada emoción. Esta capacidad de adaptación facilitaría la integración de nuevos estímulos diseñados específicamente para provocar cambios emocionales en función del estado actual del usuario. Además, en futuras etapas se contempla la incorporación de múltiples señales fisiológicas, como la respuesta galvánica de la piel (GSR), la variabilidad de la frecuencia cardíaca (HRV) y la temperatura corporal, con el fin de mejorar la precisión del sistema y permitir una diferenciación más clara entre emociones similares, como el enojo y la tristeza.

5.3. Publicaciones

A continuación, se presentan las publicaciones derivadas del trabajo desarrollado en esta investigación. Estos trabajos han sido sometidos a evaluación y contribuyen a la difusión del conocimiento generado, así como al fortalecimiento de la comunidad científica en el área de inteligencia artificial aplicada al reconocimiento de emociones.

- Alvaro A Colunga-Rodriguez, Alicia Martínez-Rebollar, Hugo Estrada-Esquivel, Eddie Clemente, Rosa Olivia Maquinay Díaz, “Emotion experiment design to build a dataset” (2024), Tecnología y Ciencia Aplicadas, Vol. 7, Número 2.
- Colunga-Rodriguez, A.A.; Martínez-Rebollar, A.; Estrada-Esquivel, H.; Clemente, E.; Pliego-Martínez, O.A. Developing a Dataset of Audio Features to Classify Emotions in Speech (2025). Computation, Vol. 13, Num. 39.

  doi: https://doi.org/10.3390/computation13020039
- Se encuentra en proceso de publicación el siguiente artículo. Alvaro A Colunga-Rodriguez, Alicia Martínez-Rebollar, Hugo Estrada-Esquivel, Eddie Clemente. Modelo híbrido de Programación Genética y Redes Neuronales para el reconocimiento de emociones. Computación y sistemas.

Apéndice A
Código fuente de programa para programación genética

```python
import pandas as pd
import numpy as np
from itertools import combinations
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler, MaxAbsScaler, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
from sklearn.metrics import f1_score
from deap import algorithms
from deap import base
from deap import creator
from deap import tools
from deap import gp
import operator
import random
import sys
import os
import datetime
import multiprocessing
import math
import copy
import traceback


def pDiv(left, right):
    try:
        if right != 0:
            return left / right
        else: return 1
    except ZeroDivisionError:
        return 1

def psqrt(x):
    return math.sqrt(abs(x))

def sin(x):
    return math.sin(x)

def cos(x):
    return math.cos(x)

def tan(x):
    return math.tan(x)

def cargar_datos(archivo):
    df = pd.read_csv(archivo)

    return df

def square(x):
    return x * x

# Guardar el logbook en un archivo de texto
def guardar_logbook_txt(logbook, filename):
    with open(filename, mode='w') as file:
        file.write(",".join(logbook.header) + "\n")  # Escribir la cabecera
        for record in logbook:
            ind1 = f"{str(record['ind1'])}"
            ind2 = f"{str(record['ind2'])}"
            ind3 = f"{str(record['ind3'])}"

            file.write(f"{record['gen']},{record['nevals']},{record['best_fit']},\"{in
            d1}\",\"{ind2}\",\"{ind3}\"" + "\n")


def my_ephimeral():
    #return round(random.uniform(-10, 10),2)
    return random.randint(1,100)

def evaluate_ind(individual,**args):
    func_0 = toolbox.compile(expr=individual[0])
```

```
        func_1 = toolbox.compile(expr=individual[1])
        func_2 = toolbox.compile(expr=individual[2])

        try:
            result_1 = np.apply_along_axis(lambda row: func_0(*row), 1, args['datos_in'])
            result_2 = np.apply_along_axis(lambda row: func_1(*row), 1, args['datos_in'])
            result_3 = np.apply_along_axis(lambda row: func_2(*row), 1, args['datos_in'])

            result = np.vstack((result_1,result_2,result_3)).T

            params = {
            'activation': 'relu',
            'solver': 'adam',
            #'hidden_layer_sizes':(layer_1,layer_2),
            'hidden_layer_sizes':7,
            'alpha': 0.0001,
            'batch_size': 16,
            'learning_rate_init':0.001,
            'learning_rate': 'adaptive',
            'max_iter': 100000,
            'early_stopping': True,
            'verbose':False
            }

            model = MLPClassifier(**params, random_state = 51)
            model.set_params(**params)
            model.fit(result,args['datos_out'])

            f1 = f1_score(args['datos_out'], model.predict(result), average='weighted')

            if f1 >= 0.97:
                return -1e-10,

            return f1,
        except Exception as e:
            print(traceback.format_exc())
            return -1e-10,



toolbox = base.Toolbox()

# Cantidad de columnas
columns = 78
# Conjunto de funciones
pset = gp.PrimitiveSet("main", columns)
pset.addPrimitive(operator.add,2)
pset.addPrimitive(operator.sub,2)
pset.addPrimitive(operator.mul,2)
pset.addPrimitive(pDiv,2)
pset.addPrimitive(sin, 1)
pset.addPrimitive(cos, 1)
pset.addPrimitive(psqrt, 1)
pset.addPrimitive(tan, 1)

#Terminales
# Renombra las terminales
pset.renameArguments(**{f"ARG{i}": f"x{i}" for i in range(columns)})
pset.addEphemeralConstant("rand1", my_ephimeral)

creator.create("FitnessMulti", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMulti)

toolbox = base.Toolbox()

# Crear un árbol individual
toolbox.register("expr", gp.genHalfAndHalf, pset=pset, min_=2, max_=3)
#toolbox.register("tree", tools.initIterate, creator.Individual, toolbox.expr)
toolbox.register("tree", tools.initIterate, gp.PrimitiveTree, toolbox.expr)


# Crear un individuo con 3 árboles
```

```python
def init_individual():
    return creator.Individual([toolbox.tree() for _ in range(3)])

toolbox.register("individual", init_individual)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

toolbox.register("select", tools.selTournament, tournsize=5)
toolbox.register("compile", gp.compile, pset=pset)

def limit_tree_height_crossover(ind1, ind2, max_height=2):

    ind1 = copy.deepcopy(ind1)
    ind2 = copy.deepcopy(ind2)

    for idx in range(3):
        # Seleccionar un punto de cruce en ind1
        idx1 = random.randint(1, len(ind1[idx]) - 1)
        # Seleccionar un punto de cruce en ind2
        idx2 = random.randint(1, len(ind2[idx]) - 1)

        # Obtener los nodos donde se realizará el intercambio
        node1 = ind1[idx][idx1]
        node2 = ind2[idx][idx2]

        # Validar que las aridades de los nodos son compatibles
        if node1.arity != node2.arity:
            # Si no son compatibles, intenta nuevamente
            continue

        # Intercambiar subárboles
        slice1, slice2 = ind1[idx].searchSubtree(idx1), ind2[idx].searchSubtree(idx2)

        # Hacer copias profundas de los subárboles antes de intercambiarlos
        subtree1 = copy.deepcopy(ind1[idx][slice1])
        subtree2 = copy.deepcopy(ind2[idx][slice2])

        ind1[idx][slice1], ind2[idx][slice2] = subtree2, subtree1

        # Limitar la altura de los árboles cruzados
        if ind1[idx].height > max_height:
            # Generar un nuevo subárbol con altura limitada
            new_subtree = gp.genFull(pset=pset, min_=1, max_=max_height)
            # Reemplazar el árbol completo con el nuevo subárbol truncado
            ind1[idx] = gp.PrimitiveTree(new_subtree)

        if ind2[idx].height > max_height:
            # Generar un nuevo subárbol con altura limitada
            new_subtree = gp.genFull(pset=pset, min_=1, max_=max_height)
            # Reemplazar el árbol completo con el nuevo subárbol truncado
            ind2[idx] = gp.PrimitiveTree(new_subtree)

    return ind1, ind2

# Registrar el operador
toolbox.register("mate", limit_tree_height_crossover)

def limit_tree_height_mutation(ind,pset,max_height=2):
    # Realizar la mutación en cada árbol

    for idx in range(3):
        mutated_tree, = gp.mutUniform(ind[idx], expr=toolbox.expr, pset=pset)

        ind[idx] = copy.deepcopy(mutated_tree)

        # Limitar la altura del árbol mutado
        if ind[idx].height > max_height:
            # Generar un nuevo subárbol con altura limitada
            new_subtree = gp.genFull(pset=pset, min_=1,
            max_=max_height)
            # Reemplazar el árbol completo con el nuevo subárbol truncado
            ind[idx] = gp.PrimitiveTree(new_subtree)
    return ind,
```

```
214
215 # Registrar el operador
216 toolbox.register("mutate", limit_tree_height_mutation,pset=pset)
217
218
219 if __name__ == '__main__':
220     """
221     # crear un individuo
222     indi = toolbox.individual()
223     indi2 = toolbox.individual()
224     print('ind1')
225     for t in indi:
226         print(t)
227
228     print('--------------------')
229     print('ind2')
230     for t in indi2:
231         print(t)
232
233
234     print('crossover')
235     indi3,indi4 = toolbox.mate(indi, indi2)
236     print('--------------------')
237     for t in indi3:
238         print(t)
239     print('--------------------')
240     for t in indi4:
241         print(t)
242
243     print('--------------------')
244     print('mutate')
245     for t in indi:
246         print(t)
247     print('--------------------')
248     indi5, = toolbox.mutate(indi)
249     print(len(indi5))
250     for t in indi5:
251         print(t)
252
253     sys.exit(0)
254     """
255
256     # Process Pool
257     cpu_count = multiprocessing.cpu_count()
258     pool = multiprocessing.Pool(cpu_count)
259     toolbox.register("map", pool.map)
260
261     # leer archivo de datos
262     dataset_dir = 'features_full'
263     datos_in = pd.read_csv(f'{dataset_dir}/audio_img_hr_train.csv').to_numpy()
264     datos_out =
        pd.read_csv(f'{dataset_dir}/audio_img_hr_train_out.csv').to_numpy().ravel()
265
266
        toolbox.register("evaluate",evaluate_ind,datos_in=datos_in,datos_out=datos_out)

267
268     today_dt = datetime.datetime.now().strftime("%Y-%m-%d-%H-%M-%S")
269     main_dir = f"run-mtree-{today_dt}"
270
271     if not os.path.exists(main_dir):
272         os.mkdir(main_dir)
273
274     cxProb = 0.7
275     mutProb = 0.3
276     numGenerations = 50
277     popSize = 200
278     verbose = True
279
280     with open(os.path.join(main_dir,f"GP-data.txt"), "w") as gp_data_file:
281         gp_data_file.write(f"Individuos:{popSize}\n")
282         gp_data_file.write(f"Generaciones:{numGenerations}\n")
```

```
283             gp_data_file.write(f"cxprob:{cxProb}\n")
284             gp_data_file.write(f"mutprob:{mutProb}\n")
285 
286         # Crear logbook para guardar estadísticas
287         logbook = tools.Logbook()
288         #logbook.header = ['gen', 'nevals'] + (stats.fields if stats else []) +
            ['best_fit','precision','recall','best_ind']
289         logbook.header = ['gen', 'nevals'] + ['best_fit','ind1','ind2','ind3']
290 
291         population = toolbox.population(n=popSize)
292         halloffame = tools.HallOfFame(1)
293 
294         # Proceso evolutivo
295 
296         # Evalua la población
297         invalid_ind = [ind for ind in population if not ind.fitness.valid]
298         fitnesses = toolbox.map(toolbox.evaluate, invalid_ind)
299         for ind, fit in zip(invalid_ind, fitnesses):
300             ind.fitness.values = fit
301 
302         if halloffame is not None:
303             halloffame.update(population)
304 
305         #record = stats.compile(population) if stats else {}
306 
307         # Mejor individuo generacion incial
308         best_ind = tools.selBest(population, 1)[0]
309         best_fit = best_ind.fitness.values
310 
311 
312         #logbook.record(gen=0, nevals=len(invalid_ind),
            **record,best_fit=best_fit,precision=precision,recall=recall,best_ind=str(best_ind
            ))
313         logbook.record(gen=0, nevals=len(invalid_ind),
            best_fit=best_fit,ind1=best_ind[0],ind2=best_ind[1],ind3=best_ind[2])
314         if verbose:
315             print(logbook.stream)
316 
317         # Generaciones
318         for gen in range(1, numGenerations + 1):
319             # Selecciona la siguiente generación
320             offspring = toolbox.select(population, len(population))
321 
322             # Variar el grupo de individuos
323             offspring = algorithms.varAnd(offspring, toolbox, cxProb, mutProb)
324 
325             # Evaluar los individuos con fitness inválido
326             invalid_ind = [ind for ind in offspring if not ind.fitness.valid]
327             fitnesses = toolbox.map(toolbox.evaluate, invalid_ind)
328             for ind, fit in zip(invalid_ind, fitnesses):
329                 ind.fitness.values = fit
330 
331             # Actualiza el Hall of Fame con la nueva población
332             if halloffame is not None:
333                 halloffame.update(offspring)
334 
335             # Reemplaza la población con la nueva
336             population[:] = offspring
337 
338             # Mejor individuo generacion
339             best_ind = tools.selBest(population, 1)[0]
340             best_fit = best_ind.fitness.values
341 
342 
343             # Agrega las estadísticas de la generación al logbook
344             #record = stats.compile(population) if stats else {}
345             #logbook.record(gen=gen, nevals=len(invalid_ind),
                **record,best_fit=best_fit,best_ind=str(best_ind))
346             #logbook.record(gen=gen,
                nevals=len(invalid_ind),best_fit=best_fit,best_ind=",".join(str(best_ind)))
347             logbook.record(gen=gen, nevals=len(invalid_ind),
                best_fit=best_fit,ind1=best_ind[0],ind2=best_ind[1],ind3=best_ind[2])
```

```
348             if verbose:
349                 print(logbook.stream)
350 
351         # Fin del proceso evolutivo
352 
353         # Mejor individuo
354         best_ind = tools.selBest(halloffame, 1)[0]
355         best_fit = best_ind.fitness.values
356         print(f'Mejor fitness: {best_fit}')
357         print(f'Mejor individuo: \n{best_ind[0]}\n{best_ind[1]}\n{best_ind[2]}')
358 
359         guardar_logbook_txt(logbook, f'{main_dir}/logbook.csv')
360         print()
361         #sys.exit(0)
362         #break
363 
364         pool.close()
365 
```

Bibliografía

Abdulmohsin, H. A., Abdul Wahab, H. B., y Abdul Hossen, A. M. J. (2021, julio). A new proposed statistical feature extraction method in speech emotion recognition. *Computers & Electrical Engineering*, *93*, 107172. doi: 10.1016/j.compeleceng.2021.107172

Akiba, T., Sano, S., Yanase, T., Ohta, T., y Koyama, M. (2019, julio). Optuna: A Next-generation Hyperparameter Optimization Framework. En *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2623–2631). Anchorage AK USA: ACM. doi: 10.1145/3292500.3330701

APA. (s.f.). *APA dictionary of psychology.* Descargado de `https://dictionary.apa.org`

Bhadangkar, Dasharath.K., Pujari, J. D., y Yakkundimath, R. (2020, octubre). Comparison of Tuplet of Techniques for Facial Emotion Detection. En *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (pp. 725–730). Palladam, India: IEEE. doi: 10.1109/I-SMAC49090.2020.9243439

Bhattacharya, S., Borah, S., Mishra, B. K., y Mondal, A. (2022, noviembre). Emotion detection from multilingual audio using deep analysis. *Multimedia Tools and Applications*, *81*(28), 41309–41338. doi: 10.1007/s11042-022-12411-3

Bishop, C. M. (2006). *Pattern recognition and machine learning.* New York: Springer.

Blackman, R. B., y Tukey, J. W. (s.f.). The measurement of power spectra from the point of view of communications engineering — part i. , *37*(1), 185–282. Descargado 2025-06-05, de `http://ieeexplore.ieee.org/document/6768513/` doi: 10.1002/j.1538-7305.1958.tb03874.x

Bradley, M., y Lang, P. (2007). *The International Affective Digitized Sounds (2nd Edition; IADS-2).* University of Florida, Gainesville, FL.: Technical Report B-3.

Chandraprabha, K. S., Shwetha, A. N., Kavitha, M., y Sumathi, R. (2021, febrero). Real time-Employee Emotion Detection system (RtEED) using Machine Learning. En *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 759–763). Tirunelveli, India: IEEE. doi: 10.1109/ICICV50876.2021.9388510

Daily, S. B., James, M. T., Cherry, D., J. Porter, J., Darnell, S. S., Isaac, J., y Roy, T. (2017). Chapter 9 - affective computing: Historical foundations, current applications, and future trends. En M. Jeon (Ed.), *Emotions and affect in human factors and human-computer interaction* (p. 213-231). San Diego: Academic Press. Descargado de `https://`

www.sciencedirect.com/science/article/pii/B9780128018514000094 doi: https://doi.org/10.1016/B978-0-12-801851-4.00009-4

Deshmukh, S., y Gupta, P. (2023, septiembre). Application of probabilistic neural network for speech emotion recognition. *International Journal of Speech Technology*. doi: 10.1007/s10772-023-10037-w

Ekman, P. (s.f.). An argument for basic emotions. , *6*(3), 169–200. Descargado 2024-09-26, de https://www.tandfonline.com/doi/full/10.1080/02699939208411068 doi: 10.1080/02699939208411068

Fausett, L. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*. USA: Prentice-Hall, Inc.

Francisti, J., Balogh, Z., Reichel, J., Benko, Ľ., Fodor, K., y Turčáni, M. (2023, octubre). Identification of heart rate change during the teaching process. *Scientific Reports*, *13*(1), 16674. doi: 10.1038/s41598-023-43763-x

Gao, L., y Guan, L. (s.f.). Interpretability of machine learning: Recent advances and future prospects. , *30*(4), 105–118. Descargado 2025-06-19, de https://ieeexplore.ieee.org/document/10114634/ doi: 10.1109/MMUL.2023.3272513

Gestal, M., Cebrián, D., Rabuñal, J., Dorado, J., y Pazos, A. (2010). *Introducción a los algoritmos genéticos y la programación genética*.

Gratz, K. L., y Roemer, L. (s.f.). Multidimensional assessment of emotion regulation and dysregulation: Development, factor structure, and initial validation of the difficulties in emotion regulation scale. , *26*(1), 41–54. Descargado 2021-12-03, de http://link.springer.com/10.1023/B:JOBA.0000007455.08539.94 (Number: 1) doi: 10.1023/B:JOBA.0000007455.08539.94

Griffiths, D., Cunningham, S., Weinel, J., y Picking, R. (2021, agosto). A multi-genre model for music emotion recognition using linear regressors. *Journal of New Music Research*, *50*(4), 355–372. doi: 10.1080/09298215.2021.1977336

Gross, J. J. (s.f.). Emotion regulation: Affective, cognitive, and social consequences. , *39*(3), 281–291. Descargado 2021-12-03, de http://doi.wiley.com/10.1017/S0048577201393198 (Number: 3) doi: 10.1017/S0048577201393198

Gross, J. J., y Thompson, R. (2007). Emotion regulation: Conceptual foundations. , 3–27.

Guanghui, C., y Xiaoping, Z. (2021). Multi-modal emotion recognition by fusing correlation features of speech-visual. *IEEE Signal Processing Letters*, *28*, 533-537. doi: 10.1109/LSP.2021.3055755

Khadimallah, R., Abdelkefi, M., y Kallel, I. (2020). Emotion regulation in intelligent tutoring systems: A systematic literature review. En *2020 IEEE international conference on teaching, assessment, and learning for engineering (TALE)* (pp. 363–370). IEEE. Descargado 2023-11-13, de https://ieeexplore.ieee.org/document/9368372/ doi: 10.1109/TALE48869.2020.9368372

Kleinginna, P. R., y Kleinginna, A. M. (s.f.). A categorized list of emotion definitions, with suggestions for a consensual definition. , *5*(4), 345–379. Descargado 2021-12-03, de http://link.springer.com/10.1007/BF00992553 (Number: 4) doi: 10.1007/BF00992553

Kuhn, M., y Johnson, K. (2013). *Applied predictive modeling*. Springer New York. Descargado de https://books.google.com.mx/books?id=xYRDAAAAQBAJ

Lang, P., Bradley, M., y Cuthbert, B. (s.f.). *International affective picture system (IAPS): Affective ratings of pictures and instruction manual*. University of Florida, Gainesville, FL.: Technical Report A-8.

Luna-Jiménez, C., Griol, D., Callejas, Z., Kleinlein, R., Montero, J. M., y Fernández-Martínez, F. (2021, noviembre). Multimodal Emotion Recognition on RAVDESS Dataset Using Transfer Learning. *Sensors*, *21*(22), 7665. doi: 10.3390/s21227665

McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., y Nieto, O. (s.f.). librosa: Audio and music signal analysis in python. En (pp. 18–24). Descargado 2024-05-08, de https://conference.scipy.org/proceedings/scipy2015/brian_mcfee.html doi: 10.25080/Majora-7b98e3ed-003

Mishra, S. P., Warule, P., y Deb, S. (2024, abril). Speech emotion classification using feature-level and classifier-level fusion. *Evolving Systems*, *15*(2), 541–554. doi: 10.1007/s12530-023-09550-9

Paul, B., Bera, S., Dey, T., y Phadikar, S. (2024, enero). Machine learning approach of speech emotions recognition using feature fusion technique. *Multimedia Tools and Applications*, *83*(3), 8663–8688. doi: 10.1007/s11042-023-16036-y

Pepa, L., Capecci, M., y Ceravolo, M. G. (2019, junio). Smartwatch based emotion recognition in Parkinson's disease. En *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)* (pp. 23–24). Ancona, Italy: IEEE. doi: 10.1109/ISCE.2019.8901033

Pham, M., Do, H. M., Su, Z., Bishop, A., y Sheng, W. (2021). Negative emotion management using a smart shirt and a robot assistant. , *6*(2), 4040–4047. Descargado 2022-04-19, de https://ieeexplore.ieee.org/document/9384222/ doi: 10.1109/LRA.2021.3067867

Picard, R. (1995). Affective Computing. *M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 321*.

Ramos, A. L. A., y Dadiz, B. G. (2018, noviembre). A Facial Expression Emotion Detection using Gabor Filter and Principal Component Analysis to identify Teaching Pedagogy. En *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)* (pp. 1–6). Baguio City, Philippines: IEEE. doi: 10.1109/HNICEM.2018.8666274

Renugadevi, R., Selvamuthukumar, M., Manikanta, K., Jeyaprakash, K., Kalaiarasi, G., y Arul Edwin Raj, A. (2024, diciembre). A Deep Learning Approach for YouTube Music Recommendation Based on Facial Emotion. En *2024 2nd International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC)* (pp. 195–199). Hyderabad, India: IEEE. doi: 10.1109/ICMACC62921.2024.10894134

Russell, J. A. (s.f.). A circumplex model of affect. , *39*(6), 1161–1178. Descargado 2022-04-07, de http://content.apa.org/journals/psp/39/6/1161 (Number: 6) doi: 10.1037/h0077714

Shu, L., Yu, Y., Chen, W., Hua, H., Li, Q., Jin, J., y Xu, X. (2020, enero). Wearable Emotion

Recognition Using Heart Rate Data from a Smart Bracelet. *Sensors*, *20*(3), 718. doi: 10.3390/s20030718

Singh, V., y Prasad, S. (2023). Speech emotion recognition system using gender dependent convolution neural network. *Procedia Computer Science*, *218*, 2533–2540. doi: 10.1016/j.procs.2023.01.227

Stoynov, V. (2023, mayo). A Novel Emotion-Aware Networking Model for Enhanced User Experience in 5G networks. En *2023 33rd Conference of Open Innovations Association (FRUCT)* (pp. 296–308). Zilina, Slovakia: IEEE. doi: 10.23919/FRUCT58615.2023.10143069

Takeshita, R., Shoji, A., Hossain, T., Yokokubo, A., y Lopez, G. (2021, noviembre). Emotion Recognition from Heart Rate Variability Data of Smartwatch While Watching a Video. En *2021 Thirteenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)* (pp. 1–6). Tokyo, Japan: IEEE. doi: 10.23919/ICMU50196.2021.9638844

Wang, Z., Yu, Z., Zhao, B., Guo, B., Chen, C., y Yu, Z. (2020, octubre). EmotionSense: An Adaptive Emotion Recognition System Based on Wearable Smart Devices. *ACM Transactions on Computing for Healthcare*, *1*(4), 1–17. doi: 10.1145/3384394

Xiao, H., Zhang, Y., Lin, X., y Cai, H. (2021). Exercise intervention framework of emotion regulation based on heart rate variability. En *2020 IEEE international conference on e-health networking, application & services (HEALTHCOM)* (pp. 1–6). IEEE. Descargado 2021-12-03, de `https://ieeexplore.ieee.org/document/9399001/` doi: 10.1109/HEALTHCOM49281.2021.9399001

Yu, M., Bai, Y., y Li, Y. (2023, julio). Emo-regulator: An emotion-regulation training system fusing virtual reality and EEG-based neurofeedback. En *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (pp. 1–4). Sydney, Australia: IEEE. doi: 10.1109/EMBC40787.2023.10340975

Zheng, A. (2015). *Evaluating machine learning models: A beginner's guide to key concepts and pitfalls*. O'Reilly Media. Descargado de `https://books.google.com.mx/books?id=OFhauwEACAAJ`

Zisad, S. N., Hossain, M. S., y Andersson, K. (2020). Speech Emotion Recognition in Neurological Disorders Using Convolutional Neural Network. En M. Mahmud, S. Vassanelli, M. S. Kaiser, y N. Zhong (Eds.), *Brain Informatics* (Vol. 12241, pp. 287–296). Cham: Springer International Publishing. doi: 10.1007/978-3-030-59277-6_26