# 

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

DEPARTAMENTO DE TECNOLOGÍAS DE

INFORMACIÓN Y TELECOMUNICACIONES

PROYECTO DE GRADO

**MODELAMIENTO DE LA PER****CEPCIÓN DEL PULSO EN LA MÚSICA SALSA**

INTEGRANTE

JESÚS HERNANDO PAZ BOLAÑOS

DIRIGIDO POR

DANIEL GÓMEZ MARÍN, PhD

JAVIER GUSTAVO DIAZ CELY, PhD

CARLOS ALBERTO ARCE LOPERA, PhD

SANTIAGO DE CALI, DICIEMBRE DE 2019

Tabla de contenido

[Resumen 6](#_Toc26965926)

[Abstract 7](#_Toc26965927)

[Lista de acrónimos 8](#_Toc26965928)

[Glosario de términos 9](#_Toc26965929)

[Lista de símbolos 10](#_Toc26965930)

[Índice de figuras 11](#_Toc26965931)

[Índice de tablas 12](#_Toc26965932)

[Motivación y antecedentes 13](#_Toc26965933)

[Contexto 13](#_Toc26965934)

[Antecedentes del problema 13](#_Toc26965935)

[Justificación 13](#_Toc26965936)

[Descripción del problema 15](#_Toc26965937)

[Hipótesis y restricciones 16](#_Toc26965938)

[Objetivos 17](#_Toc26965939)

[Objetivo general 17](#_Toc26965940)

[Objetivos específicos 17](#_Toc26965941)

[Marco teórico 18](#_Toc26965942)

[Salsa 18](#_Toc26965943)

[Beat, Pulso o Tactus 18](#_Toc26965944)

[Beat detection 19](#_Toc26965945)

[Deep Learning 20](#_Toc26965946)

[Estado del arte 22](#_Toc26965947)

[Joint beat and downbeat tracking with recurrent neural networks. 22](#_Toc26965948)

[Robust downbeat tracking using an ensemble of convolutional networks. 22](#_Toc26965949)

[Convolutional recurrent neural networks for music classification 23](#_Toc26965950)

[Context-dependent beat tracking of musical audio 23](#_Toc26965951)

[Resumen 24](#_Toc26965952)

[Metodología 26](#_Toc26965953)

[Esquema de trabajo 26](#_Toc26965954)

[Fases de desarrollo del proyecto 26](#_Toc26965955)

[Análisis de riesgos y limitaciones 29](#_Toc26965956)

[Cronograma del anteproyecto 31](#_Toc26965957)

[Presupuesto 32](#_Toc26965958)

[Contribución y resultados del proyecto de grado 34](#_Toc26965959)

[Aportes relacionados con el objeto del proyecto 34](#_Toc26965960)

[Aportes relacionados con el desarrollo de capacidades del investigador 34](#_Toc26965961)

[Resultados y entregables 34](#_Toc26965962)

[Resultados 35](#_Toc26965963)

[Creación de la base de datos 36](#_Toc26965964)

[Selección de las canciones 36](#_Toc26965965)

[Creación de las tablas 36](#_Toc26965966)

[Despliegue de las tablas 37](#_Toc26965967)

[Desarrollo del cliente y el servidor 38](#_Toc26965968)

[Requerimientos funcionales 38](#_Toc26965969)

[Requerimientos no funcionales 39](#_Toc26965970)

[Diseño 39](#_Toc26965971)

[Despliegue 40](#_Toc26965972)

[Resultados 40](#_Toc26965973)

[Sesiones con los usuarios 42](#_Toc26965974)

[Modelado y evaluación de los datos 44](#_Toc26965975)

[Entendimiento del negocio 44](#_Toc26965976)

[Entendimiento de los datos 44](#_Toc26965977)

[Preparación de los datos 44](#_Toc26965978)

[Eliminación del ruido 45](#_Toc26965979)

[Evaluación de los usuarios 48](#_Toc26965980)

[Unificación de las series de tiempo 50](#_Toc26965981)

[Creación de los datos de entrada 52](#_Toc26965982)

[Modelado 55](#_Toc26965983)

[Evaluación 56](#_Toc26965984)

[Análisis de los sujetos 58](#_Toc26965985)

[Conclusiones 61](#_Toc26965986)

[Trabajo futuro 62](#_Toc26965987)

[Anexos 63](#_Toc26965988)

[Anexo 1. Árbol del problema. 63](#_Toc26965989)

[Anexo 2. Árbol de objetivos. 64](#_Toc26965990)

[Anexo 3. Análisis de participación. 65](#_Toc26965991)

[Referencias bibliográficas 66](#_Toc26965992)

## 

## Resumen

Los modelos de detección del pulso tienen problemas al enfrentar canciones de salsa. Este inconveniente se genera ya que al entrenar este tipo de modelos se utilizan canciones de géneros “sencillos”, la salsa al ser un género tan complejo presenta un gran reto para estos modelos. Para desarrollar modelos que detecten el pulso en las canciones de salsa, primero se deben generar los datos apropiados ya que estos no existen.

Por estos motivos, el proyecto brinda una herramienta capaz de ayudar a un usuario a marcar el pulso en las canciones y almacenar los resultados en una base de datos. Los datos resultantes se procesan para entrenar de una manera adecuada modelos que sean capaces de predecir el pulso en las canciones de salsa. Además, con los datos recolectados de los usuarios se realiza un análisis para encontrar cuales son las características de este para que posea una mejor calidad en sus datos.

**Palabras claves:** Salsa, detección pulso, redes neuronales, música.

## Abstract

Pulse detection models have problems tracking salsa songs. This inconvenience is generated because models are trained with songs of "simple" genres, salsa music is a complex genre that presents a great challenge for these models. In order to develop models that detect the pulse in salsa songs, the appropriate data must first be generated since they do not exist.

For these reasons, the project provides a tool that can help a user track the pulse on songs and store the results in a database. The resulting data is processed to properly train models that can predict the pulse in salsa songs. In addition, with the collected data of the users an analysis is made to find which are the characteristics of this one so that it possesses a better quality in its data.

**Keywords:** Salsa, beat tracking, neural networks, music.

## Lista de acrónimos

KDE **K**ernel **d**ensity **e**stimation

MDS **M**ulti**d**imensional **s**caling

Cnn **C**onvolucional **n**eural **n**etwork

Rnn **R**ecurrent **n**eural **n**etwork

Hz **H**ert**z**

sr **S**harpe **r**atio

srp **S**harpe **r**atio **p**rocesado

ioi**I**nter **o**nset **I**nterval

## Glosario de términos

**Pulse o Pulso:** Sensación de marca y periodicidad sobrepuesta a estímulos cíclicos en diversas modalidades (sonido, luz, habla, etc.). En la música comúnmente se manifiesta como la indicación periódica de un ciclo a veces marcado con los pies, la cabeza o las manos. En ámbitos no científicos y musicales también se le conoce al pulso como beat.

**Data set o Base de datos:** Conjunto de datos organizado de tal modo que permita obtener con rapidez diversos tipos de información (RAE, 2010a).

**Ground-Truth:** Para confirmar o validar directamente (información o datos derivados indirectamente), especialmente (en sensores remotos) mediante observación directa en tierra, en lugar de mediante la interpretación de datos obtenidos de forma remota; para hacer observaciones de (tierra, un área, etc.) directamente sobre el terreno, especialmente para confirmar o validar los datos obtenidos indirectamente (“Definition of ground-truth in English by Oxford Dictionaries,” n.d.)**.**

**Pure Data:** Es un entorno de programación visual de código abierto que se ejecuta en cualquier dispositivo, desde computadoras personales a dispositivos integrados y teléfonos inteligentes (IEM, 2014).

**Ritmo:** Proporción guardada entre los acentos, pausas y repeticiones de diversa duración en una composición musical (RAE, 2010b).

## Lista de símbolos

Desviación estándar.

Media aritmética, promedio o media.

## Índice de figuras

[Figura 1. Cronograma del proyecto de grado. 32](#_Toc26973186)

[Figura 2. Estructura de la base de datos. 37](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973187)

[Figura 3. Diagrama de despliegue. 39](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973188)

[Figura 4 Pantalla de ingreso 40](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973189)

[Figura 5 Pantalla de calibración 41](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973190)

[Figura 6 Mensaje final de la calibración 41](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973191)

[Figura 7 Pantalla para marcar el pulso en las canciones 42](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973192)

[Figura 8 Estructura de datos para almacenar valores del retraso. 46](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973193)

[Figura 9 Histograma global para un usuario 47](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973194)

[Figura 10 Histograma de una sola fecha para un usuario 47](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973195)

[Figura 11 Comparación de dos usuarios usando su KDE 48](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973196)

[Figura 12 Multidimensional scaling para la matriz de comparaciones. 49](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973197)

[Figura 13 Valores del sharpe ratio asociados a un color. 50](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973198)

[Figura 14 Visualización del ioi para una canción 52](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973199)

[Figura 15 Espectrograma mel de una canción 53](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973200)

[Figura 16 Ejemplo de matriz de espectrograma 53](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973201)

[Figura 17 Proceso para la creación de los datos 54](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973202)

[Figura 18 Verificación existencia un pulso en la ventana. 55](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973203)

[Figura 19 Estructura de la red neuronal 56](file:///C:\Users\jeher\OneDrive\Escritorio\Icesi\19-1\PDG\Paz_Sistemas-Proyecto_V2.1.docx#_Toc26973204)

# Índice de tablas

[Tabla 1. Resumen comparativo del estado del arte. 24](#_Toc26966034)

[Tabla 2. Actividades y entregables para cada una de las fases de desarrollo en el modelo incremental. 26](#_Toc26966035)

[Tabla 3. Actividades y entregables para cada una de las fases del modelo de desarrollo CRISP-DM. 27](#_Toc26966036)

[Tabla 4. Análisis de riesgos y limitaciones que han ocurrido en el proyecto. 29](#_Toc26966037)

[Tabla 5. Análisis de riesgos y limitaciones que pueden ocurrir en el transcurso del proyecto. 29](#_Toc26966038)

[Tabla 6. Descripción y justificación de presupuesto del proyecto. 31](#_Toc26966039)

## Motivación y antecedentes

### Contexto

Los algoritmos de detección automática del pulso tienen grandes conflictos al enfrentarse a las canciones de salsa. Winter (2018) afirma: “Este ritmo se originó en África; consiste en cuatro ritmos principales repartidos en dos compases a la vez, junto con un conjunto de instrumentos que incluyen el tambor mambo, la trompeta y la guaracha, una calabaza musical”. Por ende, el ritmo en la salsa es el factor que aumenta en complejidad comparándolo con otros géneros. De manera que los algoritmos fallan al no estar entrenados ni diseñados para este tipo de canciones.

Es importante mencionar que este proyecto de grado hace parte de un proyecto más grande denominado “Deep salsa” que pertenece a la facultad de ingeniería y al departamento de diseño.

### Antecedentes del problema

El área del análisis musical en la que se utiliza la detección del pulso en las canciones se ve sesgada a utilizar música de ritmo sencillo. Pongamos por caso el uso de una red neuronal recurrente convolucional para el etiquetado de música, donde se utiliza un data set proveniente de last.fm que provee canciones del género pop, rock, entre otros (Choi et al., 2017). El anterior es uno de los cientos que ejemplos que se pueden encontrar en el cual se emplea un data set en el cual domina la música de ritmo sencillo, lo que tiene por consecuencia la ausencia o poca investigación en otros géneros musicales.

### Justificación

En este momento, existen multitud productos en el mercado relacionados con el manejo de bases de datos en donde se almacenan en promedio más de 30 millones de canciones, algunos ejemplos pueden ser Spotify, Apple Music, Pandora, Rdio, entro otros (Pullen, 2015); además, la interacción entre los millones de usuarios que posee cada una de estas plataformas con las canciones que éste escucha generan una gran cantidad de datos que sirven para mejorar la aplicación y la experiencia de usuario. Por esta razón, se han creado diversidad de modelos que procesan canciones para la detección del pulso, de similitud tímbrica, de clasificación por estilo, de análisis de armonía, entre otras características; en especial para los géneros de música rock, pop, clásica y electrónica. Así mismo, se permite generar investigación por medio del análisis de diversos géneros para poder estudiar a fondo su estructura, rasgos característicos y los aportes que genera cada uno de los diferentes compositores.

Las repercusiones que el proyecto busca, serán generar facilidades para que futuros investigadores tengan las herramientas básicas para afrontar la detección del pulso en las canciones de salsa, y a través de esto poder profundizar en la estructura de este género musical de una manera sin precedentes. Esto no solo aportará a la investigación, sino que también influirá en el desarrollo de nuevos productos que producirán un resultado positivo para Latinoamérica. Todo esto permitirá un mayor reconocimiento para la cultura, debido a que se podrá aumentar la investigación y el conocimiento en este sector.

## Descripción del problema

Actualmente, se presenta una gran dificultad en el estudio del pulso en las canciones del género salsa. Lo anterior se debe a que los algoritmos de análisis musicales son entrenados con música de ritmo sencillo, por ejemplo: rock, pop, clásica, electrónica, entre otras. A diferencia de estos géneros, la salsa cuenta con un ritmo muy complejo (Winter, 2018), lo que provoca que los algoritmos fallen al enfrentar este tipo de canciones. Además, no existen bases de datos que se usen para entrenar algoritmos de detección del pulso que contengan canciones del género salsa, aunque existen bases de datos en donde se encuentran canciones con el mismo nivel de complejidad (Böck et al., 2016). La dificultad en el estudio del pulso tiene como efectos la carencia de investigaciones enfocadas al análisis de las canciones del género salsa, la ineficiencia en los análisis actuales por poca existencia de bases de datos adecuadas y la poca existencia de aplicaciones y desarrollos tecnológicos que se basen en el análisis de la salsa.

## Hipótesis y restricciones

La hipótesis en este proyecto consiste en la realización de una herramienta que posea la funcionalidad de recopilar la sensación del pulso de los usuarios en las canciones del género salsa. Todo esto para obtener una cantidad suficiente de datos que servirán como insumo para entrenar y validar modelos de analítica de datos que sean capaces de predecir el pulso en canciones de salsa.

En el proyecto se presentan las siguientes restricciones:

* Realizar lo más pronto posible la aplicación que se va a encargar de la recopilación de los datos. La herramienta debe contar con todas las funcionalidades en correcto estado, ya que la recopilación de los datos es una de las partes más importantes del proyecto debido a que a partir de los datos recopilados se procederá a entrenar los modelos de analítica de datos.
* Ceñirse a un cronograma en donde se establecieron ya las fechas de cierre de cada uno de los entregables del proyecto.

## Objetivos

### Objetivo general

Entrenar y evaluar modelos que procesen una canción de salsa y que puedan generar eventos temporales que señalen la posición de la sensación de pulso, donde las locaciones del pulso deben estar alineadas con las sensaciones de los seres humanos.

### Objetivos específicos

* Crear una base de datos de salsa con la posición del pulso anotada por expertos (también conocida como ground truth).
* Comparar las tendencias temporales de anotación de pulso entre sujetos provenientes de diferentes contextos musicales: sujetos sin entrenamiento musical formal, sujetos con entrenamiento musical formal, sujetos entrenados en el baile de la Salsa.
* Aplicar diversas metodologías analíticas para procesar archivos de audio junto con archivos de anotaciones y entrenar modelos capaces de predecir las posiciones de los pulsos en la música salsa.

## Marco teórico

### Salsa

El género salsa es un tipo de música bailable que tuvo su origen en Cuba y Puerto Rico. Cuando sus representantes emigraron a Estados Unidos, hicieron popular esta música entre los años de 1940 y 1950 (Winter, 2018). De ahí que la salsa sea el resultado de la combinación entre el son cubano, diversos tipos de música latinoamericana y otros ritmos estadounidenses.

“Para la gran mayoría de músicos caribeños que ya habían logrado nombre y prestigio en la década de los 50, la salsa (como género musical) no existe, es tan solo música cubana vieja tocada con ciertos arreglos novedosos.” (Rondón, 2017: 29). De modo que la salsa es un género complicado de definir, debido a que es un movimiento musical que nace a partir de la combinación de varios estilos musicales y culturales que se unieron en un lugar y momento históricos. Para resumir, la Salsa es un género musical multifacético, conformado en lo estrictamente musical por ritmos de origen cubano como el Son, Son Montuno, Guaracha, Mambo, Guajira, Pachanga, Jazz afrocubano, Bolero, Bolero Son, Rumba y Danzón.

### Beat, Pulso o Tactus

El pulso en la música son los eventos temporales que se generan de manera asincrónica y repetitiva, llegando a producir un patrón que puede ser regular y predecible. Por lo tanto, el beat o pulso es la unidad básica de medición de tiempo en la música.

En la mayoría de los casos, para seguir una canción, ya sea para bailar, acompañarla con las palmas o los pies, los oyentes siguen la marcación del pulso en las canciones. En algunas composiciones, por periodos de tiempo, no se tiene ningún sonido, lo que conlleva a que el oyente infiera dónde puede estar la ubicación del pulso. Este es un fenómeno que ocurre por una de las características que tiene el pulso, la repetitividad, de ahí que los oyentes puedan predecir en qué momento va a ocurrir el pulso en la composición sin la presencia de sonidos que marquen el pulso en ese instante (Lerdahl & Jackendoff, 1983).

### Beat detection

El término *beat detection* hace referencia al procesamiento y análisis de una composición musical y predecir en dónde está la ubicación del pulso, tal y como lo haría un ser humano. Existe un gran grupo internacional que abarca temas como *beat detection*, es denominado *Music Information Retrieval* (MIR). Anualmente, se realizan congresos como el *International Society of Music Information Retrieval* (ISMIR), el cual es un foro donde se comparte la investigación sobre procesamiento, algoritmos y avances relacionados con la música, en el que también abarca el área de *beat detection* (ISMIR, 2015)*.* Además, existe un concurso llamado *Music Information Retrieval Evaluation eXchange* (MIREX) que desde el año 2005, busca la evaluación formal de los sistemas y algoritmos propuestos por la comunidad de recuperación de información de música (MIR), todo esto basándose en la evaluación a partir de bases de datos públicas (Downie, 2008).

La gran mayoría de las investigaciones en el área de *beat detection* utilizan para su desarrollo el Ballroom dataset que contiene 698 canciones con una duración aproximada de 30 segundos y una duración total de más o menos 20.940 segundos, abarca géneros como Cha Cha, Jive, Rumba, entre otros (“Ballroom,” 2004). En el año 2016, se dio la publicación del *Extended Ballroom dataset* que amplia 6 veces la cantidad de canciones (es decir 4.180 canciones) y agrega 5 nuevos géneros (para un total de 13 géneros), junto con mejoras en la calidad de audio (Marchand & Peeters, 2016). Este *dataset* incluye el género Salsa, pero solamente agrega 47 canciones que equivale a menos del 1,2% de la base de datos. De ahí que los algoritmos desarrollados para la detección del pulso no tengan una gran relación con el género salsa, aunque los resultados en la evaluación sean positivos. Dado que la cantidad de canciones de salsa es insignificante en comparación con la cantidad total de canciones en la base de datos, lo que genera un mal funcionamiento de los algoritmos en el momento de evaluar el género salsa.

La evaluación de un algoritmo de detección de pulso se basa en la continuidad de los pulsos detectados correctamente, las métricas más utilizadas son CMLc, CMLt, AMLc, AMLt (M. Davies, Degara, & Plumbley, 2009). CMLc es la proporción entre la sección más larga correctamente detectada y la longitud del archivo, con pulsos en el nivel métrico correcto. CMLt es el número total de pulsos correctos en el nivel métrico correcto (no necesariamente contiguos). AMLc es la proporción entre la sección más larga correctamente detectada y la longitud total del archivo, con pulsos en niveles métricos permitidos (no necesariamente el pulso correcto). AMLt es el número total de pulsos correctamente encontrados en diferentes niveles métricos (no necesariamente el pulso correcto). Otras métricas de evaluación útiles son el *p-score* y *f-measure*. Estas serán las métricas para evaluar la efectividad del algoritmo desarrollado en este proyecto.

### Deep Learning

El *deep learning* o aprendizaje profundo (DL) sigue el modelo de la ciencia del cerebro. Ya que tiene como base las redes artificiales tradicionales que gracias a los avances tecnológicos del momento pueden procesar inmensas cantidades de datos. Estas redes pueden realizar tareas inimaginables en el pasado, como pueden ser, la traducción automática de texto, vehículos inteligentes, reconocimiento de objetos en imágenes, entre otros.

El objetivo del DL es partir de una base de datos para encontrar posibles patrones y regularidades que sirvan para enseñar a un software a realizar tareas sin que esté programado para realizarlas, por ejemplo, la identificación de objetos como el lenguaje, imágenes, audio, entre otros. El DL hace parte del área de aprendizaje automático o *machine learning,* debido a que la maquina aprende a partir de la experiencia y no por la programación(Dunn, 2019)*.* Existen muchos modelos de DL, para el proyecto se van a necesitar los que estén relacionados con el manejo de estructuras secuenciales, por ejemplo, los modelos convolucionales o los modelos recurrentes ya que estos permiten la extracción de datos a partir de una estructura de datos con una relación temporal.

Las redes neuronales tradicionales están compuestas por tres capas: la capa de entrada, capa intermedia y la capa de salida. La capa de entrada contiene las neuronas por las que ingresan los datos que se quieren procesar. La capa intermedia se compone de varios niveles de neuronas, en donde cada vez que se cambia de nivel se aplica una transformación a los valores. La capa de salida posee las neuronas de salida, estas son las encargadas de recibir la información de la capa anterior y mostrarla, las salidas pueden ser numéricas o categorías (Urbas, 2018). Estas redes no permiten agregar mucha complejidad a los datos entre las capas de entrada y salida, por lo tanto, no son aptas para procesar datos complejos como audio o imágenes. Existen otros tipos de modelos de DL en donde se puede tener un mejor procesamiento tipos de datos más complejos.

De acuerdo con lo publicado en (Nvidia, 2018), las redes neuronales convolucionales o *convolutional neural networks* (CNNs) son un tipo de redes neuronales artificiales en donde se pueden procesar datos complejos (como imágenes, la voz, audio, entre otros), ya que estas permiten representar los datos de entrada en varias dimensiones con relaciones complejas entre ellos. Las CNNs, para transformar los datos, utilizan operaciones de convolución, consisten en la combinación de los datos de entrada (también conocido como mapa de características) con un kernel de convolución que actúa como filtro, todo esto para generar un mapa de características transformado. Los filtros en las capas convolucionales se modifican basándose en los parámetros aprendidos, para lograr una buena extracción de información para una tarea dada. Es por esto por lo que las CNNs pueden emplearse para generar una buena extracción de características de los datos, lo que puede utilizarse para conectarse con modelos de *machine learning* tradicionales y así tener la capacidad de generar una producción a partir de las salidas que éstas generan.

Según (Nvidia, 2018), las redes neuronales recurrentes o *recurrent neural networks* (RNNs) son también un tipo de redes neuronales artificiales que tienen memoria y ciclos de retroalimentación que permiten tener un mejor reconocimiento de patrones en los datos. Las RNNs son buenas para procesar datos con relaciones secuenciales, debido a esto tienen un uso común para tratar series de tiempo. En las RNNs se agregan conexiones recurrentes, estas consisten en conexiones que alimentan las capas ocultas de la red neuronal de nuevo en sí mismas. Una RNN puede visualizarse como la múltiple instanciación de redes neuronales en donde la salida de cada red se encuentra conectada a la entrada de la siguiente red. Todo esto permite a las RNNs la capacidad de realizar tareas como reconocimiento del habla, procesamiento del lenguaje natural, clasificación de imágenes, entre otras.

## Estado del arte

Para exponer el estado del arte del proyecto, se tratarán diferentes artículos en los cuales se publicaron propuestas para la detección del pulso en canciones:

### Joint beat and downbeat tracking with recurrent neural networks.

En el artículo (Böck et al., 2016), se propone un método para el seguimiento del *beat* y el *downbeat* con la utilización de una red neuronal recurrente (RNN) junto con una red bayesiana dinámica (RBD). El propósito de estos modelos es el siguiente: la RNN es la encargada de modelar la estructura de la obra musical, ya que divide en múltiples niveles interrelacionados y clasifica si a ese nivel corresponde un beat, downbeat o un no beat. La RBD, se ocupa de post procesar cada una de las funciones de probabilidad de la RNN, todo esto para alinear los *beats* y *downbeats* con la mejor solución global al inferir el compás, el tempo y la fase de secuencia.

Esta solución podría ser utilizada para el proyecto, debido a que Böck afirma que su sistema muestra un buen rendimiento para la detección del beat y downbeat en un amplio rango de diferentes géneros musicales y estilos. Lo anterior tendría que evaluarse, puesto que para la realización del artículo no se utilizaron datasets con canciones del género salsa, lo que podría provocar inconvenientes a causa de la complejidad de la misma.

### Robust downbeat tracking using an ensemble of convolutional networks.

En este artículo (Durand, Bello, David, & Richard, 2017), se presenta un sistema para realizar el seguimiento automático del downbeat en canciones. Para lograr esto, se divide la señal de audio en cuadros que se sincronizan a nivel de **tatum** de la música. Luego, se procesan estos datos para generar funciones basadas en el ritmo, armonía, melodía y contenido de bajos, que servirán de insumo para un conjunto de redes neuronales convolucionales que se entrenan para cada función, que culmina en la predicción de los *downbeats* por **tatum**. Por último, se combinan los resultados de este conjunto de redes para producir la predicción de downbeat por **tatum**.

Esta solución también podría ser utilizada para el proyecto, pero es necesario tener en cuenta que este algoritmo presenta ciertos fallos: primero, el modelo no maneja bien las canciones con 2 *beats* por compás, ya que las confunde con canciones de 4 *beats* por compás, por ejemplo, la música Creta. Segundo, es posible adaptar el algoritmo a otros tipos de música que no son con los que se entrenó, pero es necesario tener un *dataset* adaptado a este algoritmo para obtener un mejor desempeño. En conclusión, este algoritmo es posible utilizarlo en las canciones del género salsa, pero no se tiene ninguna certeza de que funcione con eficacia debido a que no fue entrenado con este género, lo que produce cierta incertidumbre y probabilidad de que falle por la complejidad rítmica de la salsa.

### Convolutional recurrent neural networks for music classification

En el artículo (Choi et al., 2017), se hace pública una red neuronal recurrente convolucional (CRNN) para la clasificación de música, esta consiste en una red neuronal convolucional (CNN) en donde se reemplazan las últimas capas por una red neuronal recurrente (RNN). Lo anterior se realiza debido a que: primero, las RNN son más flexibles para procesar las características locales que las CNN, esto es útil ya que algunos de los grupos de clasificación pueden verse más definidos por las características globales de la canción, mientras que otros pueden verse afectados por las características locales y de segmentos cortos. Segundo, las RNN pueden procesar una señal de longitud variable mientras que las CNN necesitan una señal de tamaño fijo.

El artículo reveló que las CRNN tienen un buen rendimiento respecto al número de parámetros y al tiempo de entrenamiento, esto indica que esta CRNN tiene efectividad para la extracción de características de la música. Para el proyecto, es posible experimentar en la implementación de una CRNN para que cumpla la función de la detección del pulso, pero no se tiene certeza de que funcione para las canciones de salsa, debido a que, nuevamente, no se utilizaron canciones de este género para el desarrollo de este artículo.

### Context-dependent beat tracking of musical audio

En este artículo (M. E. P. Davies & Plumbley, 2007), se publica un sistema para la detección del pulso en una señal de audio musical. Davies y Plumbey proponen una aproximación utilizando un modelo de dos estados: el primer estado, llamado estado general, se utiliza para inferir el periodo inicial de un pulso y detectar cambios en el tempo. El segundo estado, llamado dependiente del contexto, es usado para mantener la continuidad dentro de una hipótesis de pulso en un periodo dado. Estos tiempos de pulso se recuperan cuando pasan por una función que realiza un filtro para separar el periodo del pulso y la alineación.

El algoritmo demostró tener un buen rendimiento y un costo computacional reducido, que resulta ser su mayor característica. Para el proyecto podría pensarse la implementación de esta solución, pero su mayor defecto es la no utilización de canciones del género salsa para el desarrollo del algoritmo. En otras palabras, el algoritmo propuesto funciona bien para canciones de género Rock, Pop, Jazz, Clásica, entre otros, pero no se tiene ninguna garantía de que procese correctamente canciones del género salsa.

### Resumen

La tabla 1, corresponde al resumen de los artículos descritos en el estado del arte. En esta es posible encontrar que ninguno de los proyectos diferente a este utiliza canciones del género salsa. Además, es común la utilización de DL para dar solución a una efectiva detección de pulsos en las canciones.

Tabla 1. Resumen comparativo del estado del arte.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Joint beat and downbeat tracking with recurrent neural networks.** | **Robust downbeat tracking using an ensemble of convolutional networks.** | **Convolutional recurrent neural networks for music classification** | **Context-dependent beat tracking of musical audio** | **Deep salsa** |
| **Base de datos contiene salsa** | No | No | No | No | Si |
| **Utiliza Deep learning** | Si | Si | Si | No | Si |
| **Base de datos contiene diferentes géneros** | Si | Si | Si | Si | No |
| **Detecta Beats y/o Downbeats** | Si | Si | No | Si | Si |

## Metodología

### Esquema de trabajo

Para la realización del proyecto se llegó al acuerdo de tener reuniones semanales con los tutores, esta se presenta una vez a la semana con una duración entre 30 y 60 minutos. En el caso de que sucediera algún imprevisto o la validación de alguna entrega del proyecto, entonces se pactaron reuniones adicionales con los tutores. También, fue necesario requerir los servicios de la oficina de SYRI Operaciones para proceder con la instalación del software necesario en los computadores de la universidad. Se realizó una solicitud, para que posteriormente el equipo de SYRI asigne a una persona la labor de brindar los permisos de administrador para la instalación del software necesario para la realización de la toma de los datos a través de la herramienta, todo esto sucedió en la sala 309 del edificio C en la universidad Icesi.

### Fases de desarrollo del proyecto

Para la realización de este proyecto se implementarán dos metodologías distintas. El proyecto consiste en la realización de un software que tenga la capacidad de recolectar los datos proporcionados por un usuario de la sensación del pulso, esta tarea es posible solucionarla aplicando un modelo de ciclo de vida incremental. Posteriormente, se busca entrenar y validar modelos de analítica de datos que sean capaces de predecir la posición del pulso en una canción de salsa, estos modelos serán entrenados con los datos recopilados por la herramienta, para esta labor la mejor opción es la metodología CRISP-DM (IBM, 2012).

Se seleccionó el ciclo de vida incremental ya que cuenta con las siguientes ventajas: primero, permite la realización de cambios del tamaño de los incrementos si es necesario. Segundo, al efectuar el desarrollo de un software dividido en diferentes incrementos, lo que se logra es obtener una rápida y constante retroalimentación de cómo está encaminado el proyecto. Tercero, se realizan entregas constantes a el usuario, éste al utilizar y probar cada nuevo incremento se involucra más con el proyecto.

La metodología CRISP-DM fue elegida ya que aporta una guía completa para usuarios principiantes en el área de la analítica de datos. Es de gran ayuda por lo que establece cuales pueden ser las etapas de la metodología y provee una lista de posibles actividades y entregables para cada una de estas.

Para la metodología incremental se definieron cuatro incrementos, el cuarto hace uso de la metodología CRISP-DM, ya que hace referencia a una tarea de analítica de datos. Los incrementos detalladamente son:

1. Base de datos: consiste en la realización del modelo y posterior despliegue de la base de datos para almacenar toda la información recopilada por el sistema.
2. Servidor: corresponde al desarrollo de un servidor que cuente con la capacidad de comunicarse con varios clientes, además de realizar consultas y modificaciones a la base de datos.
3. Cliente: desarrollo de una interfaz gráfica capaz de reproducir audio y capturar las interacciones de los usuarios al marcar la sensación del pulso, además cuenta con la capacidad de comunicación con el servidor para enviar todo lo registrado en la interfaz.
4. Modelos de analítica de datos: consiste en el entrenamiento y posterior evaluación de modelos de inteligencia artificial para la predicción del pulso en las canciones de salsa, todo esto utilizando los datos recolectados por la herramienta desarrollada en los primeros tres incrementos.

Es importante mencionar que al acabar el incremento número tres, se procederá a realizar las sesiones de marcación del pulso con los estudiantes voluntarios, consisten en citar a los estudiantes con conocimientos musicales y hacer que usen la herramienta desarrollada. Se realizarán las sesiones necesarias para realizar la marcación de aproximadamente 500 canciones, donde cada una de estas debe ser marcada por tres estudiantes diferentes. Al culminar estas sesiones se procederá con el desarrollo del incremento número cuatro.

En la tabla 2, se pueden encontrar las fases de del ciclo de vida incremental (análisis, diseño, implementación y pruebas, despliegue), cada una con sus actividades y entregables correspondientes.

Tabla 2. Actividades y entregables para cada una de las fases de desarrollo en el modelo incremental.

|  |  |  |
| --- | --- | --- |
| **Fases de desarrollo.** | **Actividades** | **Entregables** |
| **Análisis** | - Elicitación de requerimientos.  - Definición de los requerimientos a realizar en el incremento. | - Especificación de requerimientos de software.  - Definición del alcance de cada incremento. |
| **Diseño** | - Diseño arquitectónico del incremento.  - Diseño detallado del incremento. | - Diagrama de despliegue. |
| **Implementación y Pruebas** | - Desarrollo del código correspondiente al incremento.  - Pruebas unitarias. | - Código correspondiente al incremento.  - Resultados de las pruebas. |
| **Despliegue** | - Integración con los demás incrementos.  - Despliegue en la sala de ejecución del experimento. | - Resultados de la integración de los incrementos. |

En la tabla 3, se encuentra cada una de las fases del modelo de desarrollo CRISP-DM, con sus debidas actividades y entregables. Para este proyecto no se realizará una fase de despliegue ya que el modelo de analítica de datos, en este caso solo será evaluado y publicado, más no entrará en ninguna fase de producción.

Tabla 3. Actividades y entregables para cada una de las fases del modelo de desarrollo CRISP-DM.

|  |  |  |
| --- | --- | --- |
| **Fases de desarrollo.** | **Actividades** | **Entregables** |
| **Entendimiento del negocio** | - Determinar los objetivos del negocio y del minado de datos.  - Definir un plan de proyecto. | - Objetivos del negocio.  - Criterios de éxito según el negocio. |
| **Entendimiento de los datos** | - Adquisición inicial de los datos.  - Descripción de los datos.  - Exploración de los datos.  - Verificar la calidad de los datos. | - Reporte de la recolección inicial de los datos.  - Reporte con la especificación de los datos. |
| **Preparación de los datos** | - Seleccionar los datos.  - Limpiar los datos.  - Construir los datos.  - Integrar los datos.  - Formatear los datos. | - Reporte con la justificación de la selección de los datos.  -Reporte de limpieza de los datos.  - Registros de los nuevos datos generados.  - Integración de los datos.  - Datos con un formato correcto. |
| **Modelado** | - Selección de las técnicas de modelado.  - Diseño de la evaluación.  - Construir el modelo.  - Evaluación de la calidad del modelo. | - Configuración de los parámetros para el modelo.  - Modelos con su debida descripción.  - Estimación de la calidad de los modelos. |
| **Evaluación** | - Evaluar resultados.  - Realizar la revisión del proceso.  - Determinar los siguientes pasos. | - Evaluación de los resultados. |

### Análisis de riesgos y limitaciones

Debido a que el proyecto ya se ha desarrollado en una parte considerable, entonces se han presentado ciertos riesgos y limitaciones que se evidencian en la tabla 4, para los cuales ya fue aplicada su debida contingencia.

Tabla 4. Análisis de riesgos y limitaciones que han ocurrido en el proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgos y limitaciones.** | **Efecto** | **Mitigación** | **Contingencia** |
| Dificultad en instalación de software en equipos de la universidad. | Retraso en el desarrollo. | Reducir la cantidad de software a instalar. | Instalar todo el software necesario en una petición. |
| Tecnología utilizada presente problemas. | Retraso en el desarrollo. | Investigar sobre la tecnología a utilizar. | Solucionar errores o cambiar de tecnología. |
| Solo pueden participar en el experimento estudiantes de la universidad. | Reducción del número de participantes. | Brindar una buena experiencia en el evento. | Aumentar la publicidad del evento en la universidad. |
| Falta de participantes para recopilación de los datos. | Retraso en la recopilación de los datos | Ofrecer incentivo monetario. | Publicitar de una mejor manera la convocatoria. |

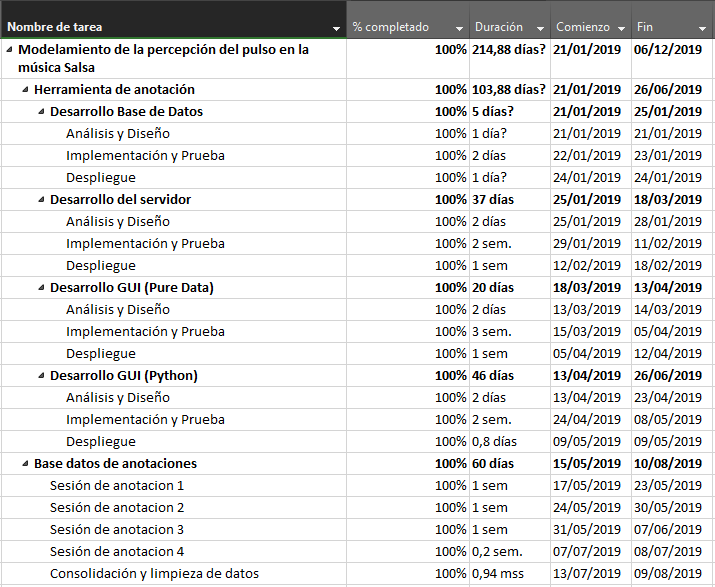
Como todavía no se ha terminado el cronograma propuesto, los riesgos y limitaciones presentes en la tabla 5 pueden presentarse en lo que resta de tiempo de trabajo.

Tabla 5. Análisis de riesgos y limitaciones que pueden ocurrir en el transcurso del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgos y limitaciones.** | **Efecto** | **Mitigación** | **Contingencia** |
| Los resultados del experimento no cuenten con la calidad necesaria. | Retraso en el entrenamiento de los modelos. | Realizar pruebas en la herramienta antes de realizar el experimento. | Encontrar el error en la toma de los datos y solucionarlo. |
| La cantidad de canciones marcadas no sea suficiente para entrenar el modelo. | Retraso en el entrenamiento de los modelos. | Realizar el experimento para una gran cantidad de canciones. | Agregar más canciones y repetir el experimento para las nuevas canciones. |
| El modelo entrenado no sea capaz de detectar correctamente el pulso en las canciones de salsa. | Retraso en la finalización del proyecto. | Asegurarse de la calidad de los datos ingresados por los estudiantes en el experimento. | Buscar el motivo por el cual el modelo no funciona correctamente y solucionarlo. |

### Cronograma del anteproyecto

En la figura 1 es posible apreciar cada una de las actividades que se encuentran estipuladas para realizar en el proyecto, cada una de ellas cuenta con su indicador de porcentaje completado, la duración planeada, su fecha de comienzo y de fin.



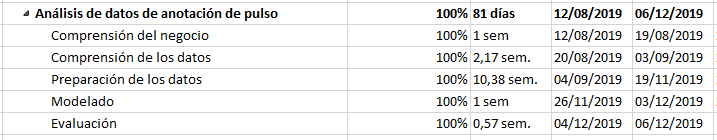


Figura 1. Cronograma del proyecto de grado.

### Presupuesto

Para la ejecución de este proyecto se realizó el siguiente presupuesto:

Tabla 6. Descripción y justificación de presupuesto del proyecto.

|  |  |  |
| --- | --- | --- |
| **Descripción** | **Precio** | **Justificación** |
| Pagos a los estudiantes participantes en la toma de los datos | 1’200.000 COP | Para la exitosa realización de la base de datos es necesario un equipo de estudiantes con experiencia en la música, ya que estos son los encargados de realizar la marcación del pulso en cada una de las canciones de salsa. Para esto se planea que por estudiante se puedan marcar 10 canciones por hora, la sala tiene capacidad de 20 estudiantes y tenemos aproximadamente 500 canciones, que deben ser escuchadas por 3 estudiantes diferentes. Entonces llegamos a la conclusión de que es necesario pagar a 20 estudiantes por aproximadamente 7.5 horas, cuando la hora se paga a 8.000 COP. |
| Compra de 20 auriculares para la realización del experimento. | 1’000.000 COP | Debido a que el experimento consiste en que diferentes personas escuchen canciones de salsa para realizar la marcación del pulso, lo que se busca es tener una escucha homogénea para todos los participantes. Por lo tanto, es necesario que todos los participantes cuenten los con mismos medios para el experimento. El precio de los auriculares de es 50.000 COP, y es necesario realizar la compra de 20 unidades. |

## Contribución y resultados del proyecto de grado

### Aportes relacionados con el objeto del proyecto

* Publicación de la base de datos con los resultados del experimento de la marcación del pulso en las canciones de salsa seleccionadas. Lo que permitirá a futuros investigadores contar con los datos para continuar con el análisis en las canciones de salsa.
* Publicación del software para la recolección de datos. Esto proporcionará a las personas interesadas la capacidad de replicar o realizar experimentos parecidos con las modificaciones que deseen.
* Publicación de modelos para la detección del pulso en las canciones de salsa.
* Proveer la capacidad de aumentar la investigación del género salsa, debido a que se publicará una base de datos, una herramienta y modelos de analítica con la capacidad de continuar investigando en este género.

### Aportes relacionados con el desarrollo de capacidades del investigador

* Aumento en las capacidades de comunicación y escritura.
* Desarrollo de conocimientos en lenguaje de programación Python, inteligencia artificial y música.
* Incremento en habilidades investigativas.

### Resultados y entregables

Para cada uno de los objetivos específicos se tienen los siguientes entregables:

Objetivo específico 1.

* Documento de especificación de requerimientos.
* Diagrama de despliegue.
* Base de datos en donde se encuentra la marcación del pulso en canciones de salsa.
* Aplicación que se compone de un servidor y cliente para realizar la toma de datos de los usuarios.

Objetivo específico 2.

* Resultados de la comparación entre los diferentes sujetos que participaron en el experimento.

Objetivo específico 3.

* Especificación de los objetivos del negocio.
* Resultados de los modelos y su evaluación.
* Análisis de los resultados y los pasos a seguir.

### Resultados

Los resultados de este proyecto se presentan de acuerdo con cada uno de los implementos definidos en la metodología. Las etapas se encuentran en el orden cronológico en el cual se realizaron y van acompañadas de sus respectivos entregables.

## Creación de la base de datos

Los datos son la parte más importante en la creación de los modelos, por esto mismo, es necesario tener un riguroso proceso en la recolección de estos para este proyecto. Se establecieron una serie de etapas que se llevaran para una correcta creación y recopilación de los datos, los pasos fueron: selección de las canciones, creación de las tablas y despliegue de las tablas.

### Selección de las canciones

Para este proceso se delega el trabajo a él magister Mateo de los Ríos Camacho, quien es el director de la carrera de música de la universidad Icesi. Se le encomienda la recopilación de la mayor cantidad de canciones de salsa que tuvieran origen en el periodo comprendido entre los años 1955 y 1995. Como resultado fueron entregadas 424 canciones que se encontraron en diferentes formatos, por lo tanto, para unificarlas, todas las canciones fueron convertidas en el formato mp3. Se debe tener en cuenta que es de suprema necesidad conservar la mayor calidad de audio posible en cada pieza, por esto mismo, al convertirlas se establece una frecuencia de muestreo de 48000 Hertz (Hz) y una velocidad de bits de 128 kbps.

### Creación de las tablas

Al contar con el insumo de las canciones, se procede con el diseño de la base de datos. Como se aprecia en la figura 2 fueron creadas 4 tablas. La primera de estas es llamada *despacho\_cancion,* es la encargada de almacenar cada una de las canciones por: su identificador y nombre, el número de veces que esta se ha escuchado (donde 0 es el mínimo y 3 el máximo) y cada uno de los 3 usuarios que escucharon y marcaron la canción junto con la fecha en que se realizó dicha actividad. La segunda, tiene el nombre de *usuarios*, en esta tabla se almacena toda la información relacionada con los participantes del experimento, es decir, su cédula, nombre y apellido, profesión y las fechas en las cuales usó el software. La tercera, se llama *delay* en esta tabla encontramos el identificador correspondiente generado por la base de datos, la cédula del usuario al cual corresponde ese retraso, cada una de las pulsaciones que marcó el usuario para encontrar su retraso y la fecha en la cual lo hizo. Cabe aclarar que un usuario al iniciar la sesión debe escuchar y marcar el pulso a una canción de 30 segundos, de la cual se poseen los valores exactos de las pulsaciones. A partir de esto, es posible calcular la diferencia entre lo que el usuario marca al escuchar la canción y lo que ocurre realmente. La última, tiene el nombre de *databeats*, en esta tabla se almacena cada una de las marcaciones de las canciones generadas por los usuarios. Se compone de las claves foráneas de la canción marcada y el usuario que la marcó, la lista de los pulsos que se generaron, el retraso Imagen que contiene captura de pantalla

Descripción generada automáticamentede la canción y la fecha en la cual se crearon los datos.

Figura 2. Estructura de la base de datos.

### 

### Despliegue de las tablas

Después de generarse el diseño de la base de datos y lo que esta iba a contener en cada una de sus tablas, se procede con la realización del despliegue de esta base de datos relacional en algún motor de base de datos. Debido a que era necesario que la base de datos fuera creada en el menor tiempo posible, la prioridad fue encontrar un gestor que permita la rápida creación y despliegue de una base de datos. La opción seleccionada fue XAMPP, que es una distribución de apache de fácil y rápida instalación, al instalarse el equipo queda preparado para desarrollar una aplicación en PHP (Apache Friends, 2017). La base de datos se diseña y despliega en MySQL (incluida al instalar XAMPP) de manera local. No se buscaron otras alternativas, ya sea en la nube o local, debido a que se contaba con cierta experiencia utilizando este gestor, lo que permite realizar esta tarea en el menor tiempo posible.

## Desarrollo del cliente y el servidor

En el proceso de recopilación de los datos, era necesario crear una herramienta de software que permita a los usuarios transmitir la sensación del pulso en la canción al computador en forma de datos. Para realizar esta tarea se siguieron ciertos requerimientos funcionales y no funcionales que se van a presentar a continuación.

### Requerimientos funcionales

El servidor debe permitir:

R1. Buscar si el documento de identidad ingresado existe en la base de datos, en caso de que exista debe asignar una canción al usuario y en caso contrario se debe notificar que no tiene permitido acceder al sistema.

R2. Asignar una canción a un usuario. La canción asignada no debe ser escuchada más de tres veces y el usuario no debe de haberla escuchado antes. Debe retornar el nombre de esta canción para que el servidor pueda cargarla fácilmente.

R3. Recibir y guardar en la base de datos la lista de pulsaciones de las canciones generadas en el cliente.

El cliente debe permitir:

R4. Ingresar el documento de identidad del usuario y enviarlo al servidor para que éste valide si existe en la base de datos, con la respuesta debe permitir o denegar el acceso a la aplicación.

R5. Cargar y reproducir una muestra de una canción en donde ya se conocen los datos de las pulsaciones, para generar una manera en la que los usuarios se puedan calibrar. Esto para permitir encontrar un valor asociado al retraso respecto a los datos reales comparado con los que genera cada usuario.

R6. Cargar y reproducir una canción almacenada en el computador por medio del nombre que envía el servidor.

R7. Recibir la pulsación de la barra espaciadora siempre y cuando en el momento se encuentre presente la reproducción de una canción. Adicionalmente, se le indica al usuario que pulse la barra espaciadora cada vez que sienta el pulso en la canción, estos datos de tiempo se deben almacenar en una lista para que al culminar la canción sean enviados al servidor.

R8. Reproducir continuamente 10 canciones asignadas por el servidor, para que al culminar estas canciones se muestre un mensaje al usuario en el que se le alerte que necesita tomar un descanso. Posterior al mensaje, se debe cerrar la sesión al usuario para que tome un descanso y en caso de que quiera continuar debe volver a iniciar su sesión.

### Requerimientos no funcionales

El software debe:

RNF1. Ser desarrollado en el lenguaje de programación Python, ya que es un lenguaje fácil de entender y compilar en cualquier computador.

RNF2. Desplegarse en los equipos de las salas de cómputo de la universidad.

RNF3. Desarrollarse en el menor tiempo posible, todo esto para iniciar rápidamente la recolección de los datos.

### Diseño

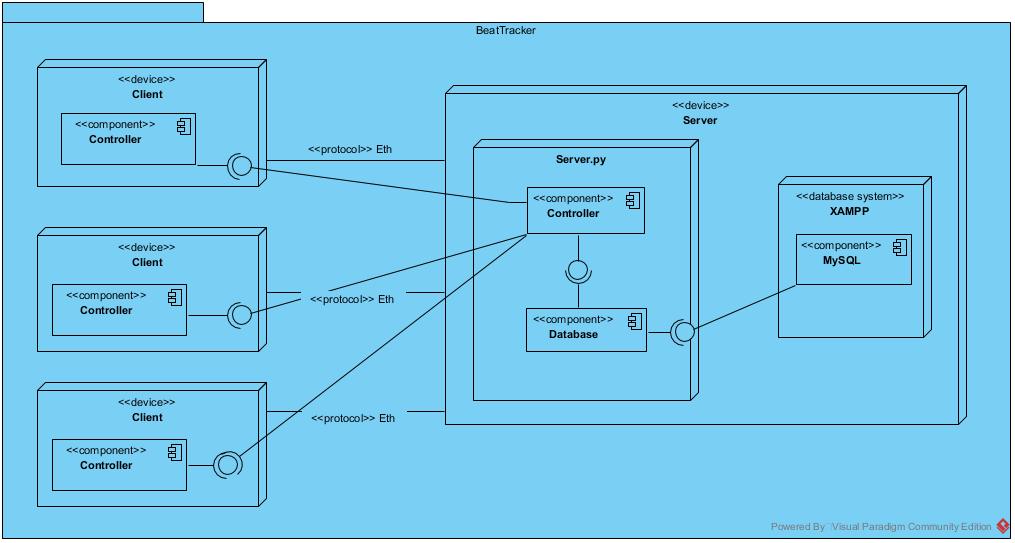
Para el desarrollo de este trabajo se desarrolló y siguió el diagrama de despliegue que corresponde a la figura 3. En este diagrama podemos observar que se encuentra distribuido en dos diferentes tipos de dispositivos. El primero, es el llamado *Server*, este tipo de dispositivo solo se encuentra una vez en la arquitectura, es el encargado de recibir todas las peticiones de los clientes además de consultar, procesar e insertar filas en la base de datos. El segundo, son los clientes, este tipo de dispositivo puede ir desplegado en varios computadores diferentes, es el encargado de recibir los datos por parte del usuario y enviarlos al servidor.

Figura 3. Diagrama de despliegue.

### Despliegue

Para la etapa de despliegue fue necesario pedir la ayuda del equipo de Syri Infraestructura, ya que este grupo es el encargado de dar permisos de administrador para la instalación del software en los equipos de la universidad. Por lo tanto, fue necesario hacer una petición formal que posteriormente culmino con la ayuda de algunos de los integrantes de este grupo. En los computadores fue necesario instalar Python y algunas librerías necesarias para la ejecución del software, ya que estos fueron los encargados de cumplir el rol de cliente en la arquitectura y son los usados por los usuarios. Mientras el servidor y la base de datos fueron desplegados en otro equipo diferente a los propios de la universidad, ya que los datos son almacenados de manera local y siempre es necesario para ejecutar el servidor contar con permisos de administrador.

### Resultados

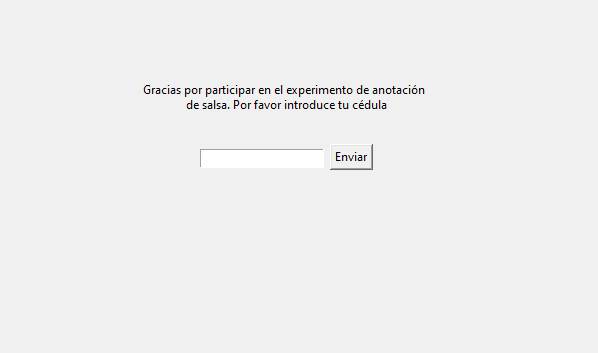
El cliente cuenta con una interfaz gráfica que le permite al usuario iniciar sesión y marcar cada vez que sienta el pulso en una canción de salsa. Lo primero que observa el usuario al ingresar es una pantalla donde se debe ingresar el documento de identidad para iniciar sesión, este debe estar previamente agregado en la base de datos. En la figura 4 es posible apreciar la pantalla anteriormente descrita.

Figura 4 Pantalla de ingreso

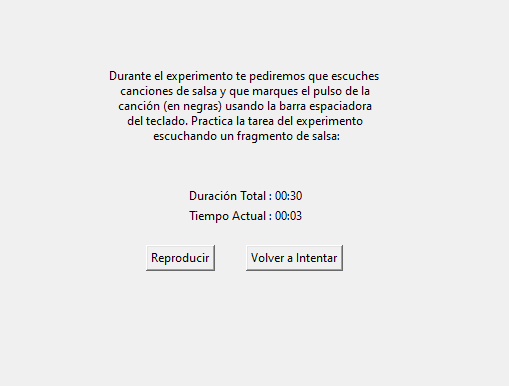
El usuario después de entrar pasa a otra pantalla (figura 5) donde debe pulsar el botón reproducir, todo esto para escuchar un fragmento de una canción de salsa donde sabemos exactamente en qué posición se encuentran los pulsos de la canción, para de esta manera, procesar los datos que se poseen con lo que el usuario genera y generar métricas para evaluar el desempeño de cada uno de los usuarios y asignarles un peso de importancia, proceso que se aclara más adelante.

Figura 5 Pantalla de calibración

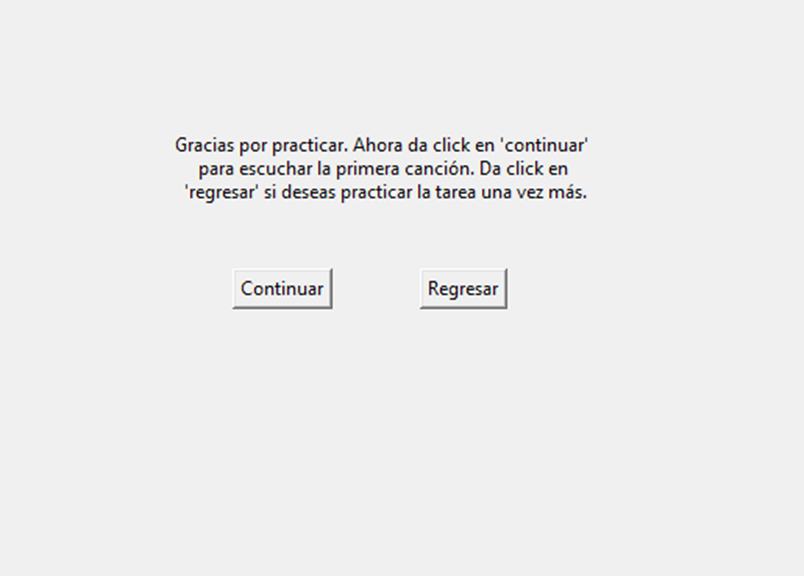
Al acabar de escuchar la canción, se muestra una pantalla con un mensaje donde le pregunta al usuario si desea continuar con el experimento o prefiere seguir practicando (figura 6). Si el usuario desea seguir practicando, pulsa el botón regresar y la canción de muestra vuelve a aparecer. En caso de que desee continuar se avanza a la siguiente pantalla.

Figura 6 Mensaje final de la calibración

El usuario al elegir continuar puede ver una pantalla (figura 7) donde se muestra la duración de la canción que se va a reproducir y las instrucciones que se brindan a los usuarios. Cuando el usuario desee empezar, debe pulsar el botón reproducir y la canción va a sonar, por lo cual debe marcar el pulso con la barra espaciadora. En caso de que desee repetir la canción debe pulsar el botón volver a intentar, y la canción inicia de nuevo. Al acabar de reproducir la canción, automáticamente saldrá una pantalla parecida a la tercera donde le pregunta al usuario si desea continuar, este ciclo seguirá hasta que el usuario escuche diez canciones diferentes. Al acabar, el usuario puede elegir si toma un descanso y continua o si desea terminar la sesión.

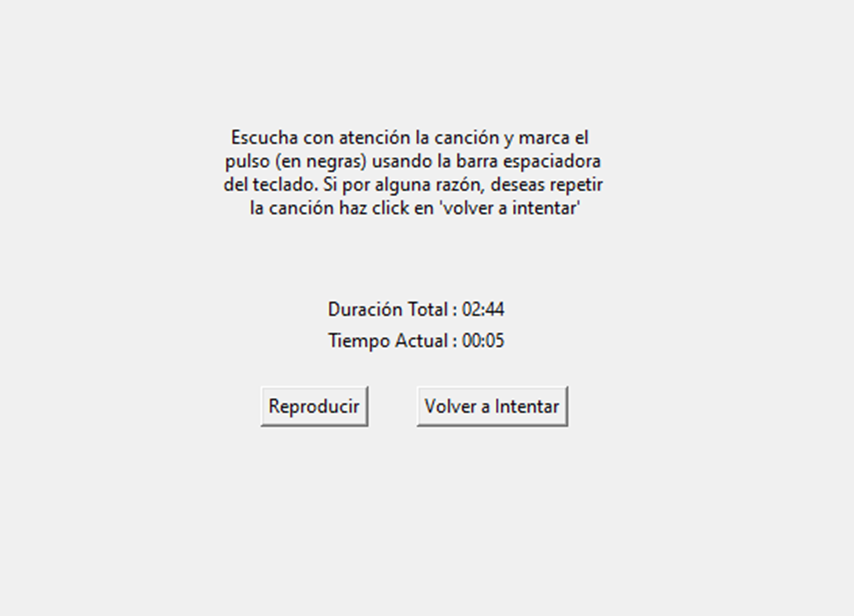


Figura 7 Pantalla para marcar el pulso en las canciones

### 

### Sesiones con los usuarios

El software al encontrarse desarrollado en su totalidad y debidamente desplegado en las salas de computo de la universidad, debe ser utilizado por los participantes para de esta manera obtener los datos necesarios para cada canción. Por estos motivos, se realizaron convocatorias por medio del correo institucional a los estudiantes de música y también se realizaron afiches que se ubican en lugares estratégicos de la universidad. En la duración del proyecto se presentaron más de 20 estudiantes diferentes a lo largo de las más de 12 sesiones que se realizaron. A los estudiantes se les realiza un pago por hora asistida y cada sesión puede tener una duración de 2 hasta 8 horas, con pausas entre cada ronda de canciones y tiempos libres para descansar.

La asistencia durante el semestre fue muy baja, debido a que los estudiantes deben realizar trabajos y exámenes, además de que es una tarea difícil encontrar horarios que los participantes tengan libres y que encajen con una sala disponible. Por estos motivos, las sesiones donde más participación se presentó y contaron con mayor duración fueron al finalizar las clases y exámenes, es decir, cuando los participantes se encontraban en época de vacaciones. Por estos motivos, al finalizar todas las sesiones, no se alcanzó el objetivo de completar todas las canciones seleccionadas debido a los problemas de asistencia, pero si se logró la marcación de la sensación del pulso de 364 de 424 canciones, es decir, el 85,85% de las canciones seleccionadas.

## Modelado y evaluación de los datos

### Entendimiento del negocio

El objetivo principal para el proyecto es partir de una canción de salsa y encontrar cuales son las posiciones de los pulsos en la canción. Este tipo de proyectos ya se han implementado antes pero no con canciones de salsa. Por lo tanto, para que un modelo se considere exitoso es necesario que identifique los pulsos en una canción de salsa y que los datos con los cuales está siendo entrenado y evaluado sean generados por la herramienta.

### Entendimiento de los datos

Los datos generados por la herramienta consisten en un identificador de canción, el identificador del usuario que realizó los datos y una lista en donde se ubica el tiempo en el que los participantes marcaron que sintieron un pulso. También se cuenta con las canciones en formato mp3, donde el identificador es el mismo nombre de la canción.

Al momento de desarrollar la aplicación se tuvo en cuenta que era necesario encontrar el valor del retraso que posee cada uno de los usuarios en el momento de marcar los pulsos. Este retraso es el valor de tiempo que corresponde a lo que se demora el dato en llegar desde que se presiona la barra espaciadora hasta que la aplicación la guarda el dato. Para realizar el proceso de encontrar este valor, el usuario al comenzar a usar la herramienta debe realizar la marcación del pulso de una canción de 30 segundos de la cual sabemos dónde se encuentran exactamente sus pulsos.

Por estos motivos en la base de datos tenemos dos tipos de series de tiempo de pulsos: el primer tipo, son los datos que provienen de la canción inicial de 30 segundos. El segundo tipo, son los datos que provienen de las canciones de las cuales no sabemos dónde exactamente se encuentran los pulsos.

### Preparación de los datos

La preparación de los datos es el proceso para limpiar todo el ruido y generar los datos necesarios para entrenar el modelo. En este caso de compone de las fases de eliminación del ruido, evaluación de los usuarios, unificación de las series de tiempo y la creación de los datos de entrada. La salida de este proceso es el insumo para las fases de modelado y evaluación.

### Eliminación del ruido

Para poder entrenar los modelos es necesario procesar toda la información recopilada por la aplicación y de esta manera crear un conjunto de datos que será el insumo para entrenar el modelo. Cabe tener en cuenta que lo más importante para la base de datos es contar con una ubicación del pulso lo más exacta posible. Para esto se procede con la eliminación del ruido en las series de tiempo que contienen las ubicaciones del pulso. Este proceso utiliza los datos provenientes de la canción de 30 segundos que ya conocemos sus pulsos, de esta manera es posible aplicar un tratamiento a estos datos para encontrar el valor del retraso correspondiente a cada persona.

Para realizar este proceso se recorre cada uno de los usuarios y se agrupan sus datos por fecha, es decir, cada usuario posee las fechas en las cuales realiza la percepción del pulso en las canciones y a estas fechas se adicionan los datos que el usuario generó ese día. Al tener estos datos se aplica la fórmula 1.

En la fórmula 1, corresponde al valor retraso en la posición actual. Este es el resultado de la resta entre el valor del pulso en la posición actual de la muestra de 30 segundos () donde ya se conoce el valor y el pulso en la posición actual de la muestra de 30 segundos de los datos que genero el usuario (). Cada resultado de estos datos se almacena en una lista que contiene los valores del retraso correspondientes a la fecha, es decir, la fórmula se aplica a todas las veces que se repite la canción de muestra y se almacenan por fecha, donde la fecha tiene a un usuario. Se puede apreciar la estructura de cómo se almacenan los datos en la figura 8.

Figura 8 Estructura de datos para almacenar valores del retraso.

Con estos datos es posible realizar diferentes tipos de operaciones, debido a que se pueden procesar ya sea de una manera global, es decir, realizar un conglomerado de todos los datos por usuario sin importar la fecha o realizar el mismo proceso teniendo en cuenta las fechas, tal y como se ilustra en la figura 9. A continuación, se realiza un conteo de la cantidad de veces que aparece un valor del retraso para posteriormente generar un histograma, donde las valores son los diferentes datos de retraso y la frecuencia con la que aparecen. En la figura 9 es posible apreciar un histograma global de los datos generados por un usuario, es decir, todos los datos generado sin importar la fecha. El eje x corresponde al valor del retraso, mientras que el eje y tiene la cantidad de veces que aparece el valor en el eje x. La línea azul ubicada sobre el histograma corresponde al *kernel density estimation* (KDE). El KDE es un estimador de densidad del núcleo que depende de todos los datos para alcanzar una estimación, en este caso se emplea para realizar una suavización de cada punto de datos observado sobre una vecindad local de este punto de datos (The University of Edinburgh, 2008) y de esta manera obtener una función de lo que representa el histograma de los datos.

Para calcular el valor del ruido, el proceso a seguir es encontrar el máximo valor que posee la función del KDE en su eje y, para hallar a qué valor corresponde del histograma en el eje x. En el caso de la figura 9, el valor corresponde a 251,72 milisegundos, que es el ruido para este usuario si no se tuviera en cuenta la fecha en la que se crearon los datos.

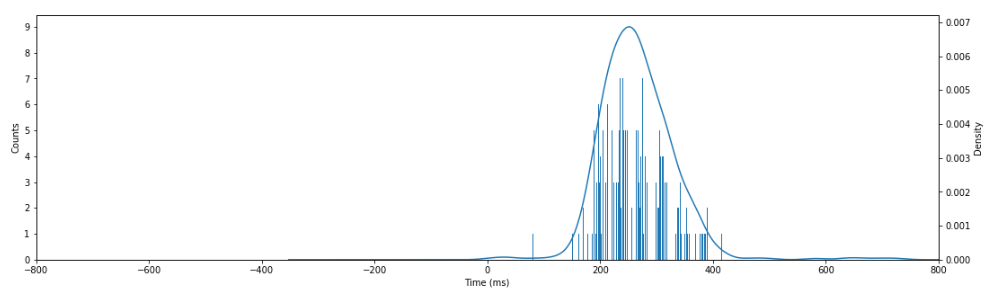


Figura 9 Histograma global para un usuario

Para eliminar el ruido no se utiliza el dato que corresponde al ruido global por usuario, se realiza por fecha ya que al descubrir que en cada fecha el valor es diferente, no es posible asumir este ruido como una constante. Por lo tanto, se procede a realizar los mismos cálculos hechos para el cálculo del ruido global, es decir aplicar la fórmula, crear el histograma y encontrar el punto máximo en el KDE. Al realizar este procedimiento se obtiene una gráfica como la que se presenta en la figura 10, que corresponde al mismo usuario de la figura 9 pero realizando un filtro para que solo se encuentren datos creados en esta fecha. En este caso el valor del ruido para la fecha en la cual el usuario marcó los pulsos fue de 207,6 milisegundos.

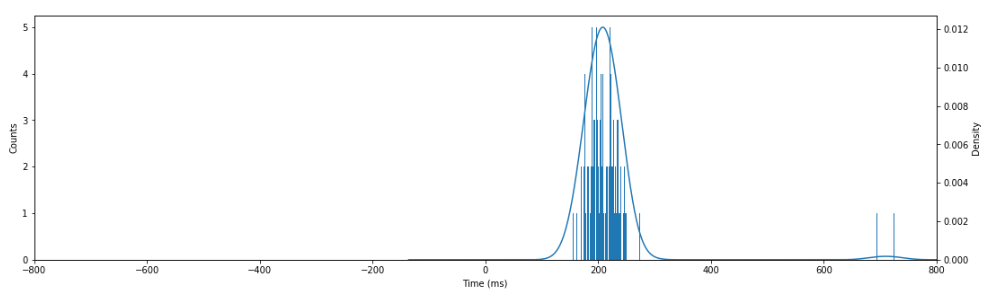


Figura 10 Histograma de una sola fecha para un usuario

Ahora, como se cuenta con los valores correspondientes al ruido de todos los usuarios en cada fecha que realizaron datos, se puede comenzar a eliminar este ruido de las canciones. Para esto, se recorren todos los datos de la ubicación de los pulsos teniendo en cuenta en qué fecha fueron realizados y que usuario los marcó, para de esta manera buscar a qué valor de retraso corresponde y restarlo a todos los datos de la serie de tiempo de los pulsos.

### Evaluación de los usuarios

Es necesario encontrar una manera de evaluar el desempeño de cada uno de los usuarios participantes, para esto se pueden crear diferentes tipos de métricas. Una de las formas de evaluación consiste en partir de la función del KDE perteneciente al histograma global de cada usuario y compararlo con los demás. Para compararlos, se agregan las dos funciones de KDE a una gráfica y se encuentra la intersección de sus funciones (figura 11), de esta intersección se calcula su área bajo la curva para posteriormente agregar este valor a una matriz de comparaciones. Esta matriz tiene como ejes los identificadores de los usuarios, ya que es necesario tener la capacidad de acceder a la comparación de dos usuarios específicos en cualquier momento.

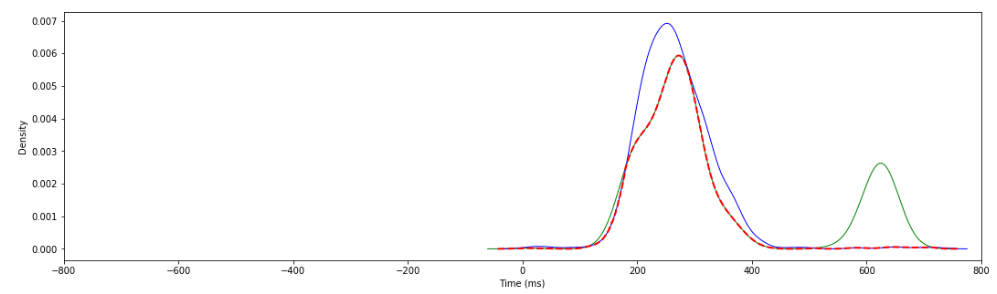


Figura 11 Comparación de dos usuarios usando su KDE

En la figura 11, se pueden apreciar 3 funciones de diferentes colores. La función de color azul corresponde al KDE generado a partir de los datos del usuario A, mientras la función de color verde pertenece al KDE de los datos del usuario B. La función roja punteada, corresponde a la intersección entre estos dos usuarios, a partir de esta función se encuentra su área bajo la curva y se agrega a la matriz de comparaciones anteriormente mencionada.

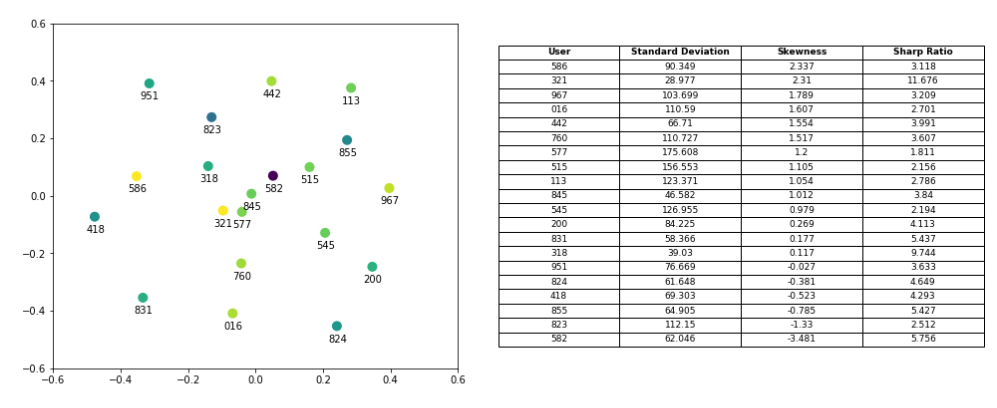
Cuando la matriz está completa, se procede a realizar un método llamado *multidimensional scaling* (MDS), este brinda una manera de representar un conjunto de datos en un entorno espacial, permite de esta manera mayor facilidad de interpretación de los datos en comparación con los datos en crudo (Pigden, Young, & Hamer, 1988). Lo que se busca con este resultado es poder ubicar a los usuarios en una gráfica en donde se pueda apreciar las distancias entre cada uno de ellos. En la figura 12 se puede encontrar el gráfico correspondiente al MDS.

Figura 12 Multidimensional scaling para la matriz de comparaciones.

Las otras métricas que se obtuvieron de las series de tiempo de los pulsos fueron la desviación estándar, media y sharpe ratio. Después realizar el análisis de cada una de las métricas se toma la decisión de que para evaluar el desempeño de un usuario se va a utilizar el sharpe ratio. Esta es una métrica utilizada especialmente en finanzas para calcular el retorno de una inversión comparado con su riesgo (Sharpe, 1994), donde la persona con un mayor valor va a ser considerada mejor que una con menor valor. La fórmula 2 corresponde al cálculo del sharpe ratio para este proyecto, donde es el valor del sharpe ratio para el usuario, es el promedio de los datos del retraso que poseen los usuarios y corresponde a la desviación estándar de estos mismos datos.

(2)

Los colores que se aprecian en la figura 12, corresponden al valor del sharpe ratio que posee dicho usuario. La relación entre color y usuario se puede encontrar en la figura 13, donde el eje x corresponde al valor del sharpe ratio, mientras el eje y posee los últimos 3 dígitos de los documentos de identidad de los usuarios.

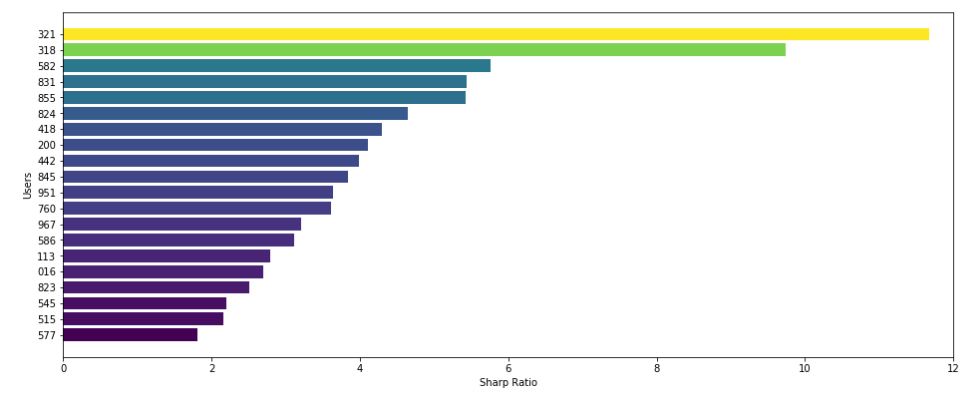


Figura 13 Valores del sharpe ratio asociados a un color.

Con el desarrollo del MDS, es posible validar que los usuarios tienen unos datos muy parecidos entre ellos. Debido a que en la figura 12, todos los usuarios se encuentran agrupados en el centro y no se presenta ningún usuario atípico, es decir, que se aleje mucho del grupo. Por estos motivos no se procede con la eliminación de los datos de alguno de estos.

### Unificación de las series de tiempo

Como cada una de las canciones fue marcada en tres ocasiones por usuarios diferentes, es decir, por cada canción existen tres diferentes listas de tiempos en donde se encuentran los pulsos. Esto genera que sea necesario proceder con la unificación de las tres listas. Para esto es necesario utilizar las métricas generadas en la sección anterior, en este caso solo se utiliza el valor asociado al sharpe ratio de cada usuario.

Al contar con 3 listas de datos provenientes de diferentes usuarios para una sola canción, es claro que, al tener un factor humano, los datos de las listas van a ser muy diferentes, por ejemplo, puede que los usuarios no comienzan siempre al mismo tiempo las canciones o no marcan los pulsos que los otros si marcaron, entre otras razones. Por estos motivos, para recorrer los datos se comienza iterando la lista con los valores del usuario con un nivel más alto en su sharpe ratio. Con el valor de la iteración se busca en las otras dos listas si alguno de los otros usuarios posee un valor que encuentre a una distancia de 200 milisegundos del valor actual. Estos datos parecidos se utilizarán para aplicar los siguientes procedimientos, en este caso solo basta con que dos usuarios posean un valor del pulso parecido para poder continuar, en caso de que no sean parecidos, se sigue iterando hasta que se encuentre algún valor que los usuarios tengan similar.

Además, es necesario calcular un valor adicional denominado sharpe ratio procesado (), este valor se calcula para cada uno de los usuarios y cambia dependiendo de la canción. Este valor permite asignar un mayor peso a los usuarios que tengan un valor de sharpe ratio más alto, y penaliza a los usuarios con valores más bajos. En la fórmula 3, el valor corresponde al cálculo del sharpe ratio procesado para un usuario y canción. es el valor del sharpe ratio de un usuario específico. El resultado es el un el porcentaje de importancia que se le asigna a los datos del respectivo usuario para toda una canción.

(3)

Posteriormente, se establece la creación de un diccionario que posee como llave el identificador de la canción y su valor corresponde a una lista con cada uno de sus tres diferentes arreglos de tiempo en donde se encuentra el pulso. Con estos datos se procede a iterar cada una de las canciones y a aplicar la fórmula 4.

(4)

En la fórmula 4, es el valor del pulso unificado en la posición del arreglo. corresponde al valor del sharpe ratio procesado, para generar este dato se tiene en cuenta los valores del sharpe ratio que poseen los otros usuarios que marcaron esa misma canción, se calcula para cada canción utilizando la fórmula 3. es el valor del pulso del usuario en la posición del arreglo .

La fórmula 4 consiste en la multiplicación del valor actual del pulso de cada uno de los usuarios por su valor de sharpe ratio. Por lo tanto, permite crear una ponderación en los datos, en otras palabras, permite asignar un peso mayor a los usuarios que tengan un valor más elevado en su sharpe ratio, mientras que, si el valor es menor, los datos de este usuario tienen menos peso para el valor final.

Para apreciar de una manera gráfica los resultados de esta lista unificada y de la misma manera evaluar estos, se define la fórmula 5. Esta fórmula tiene como resultado el valor del *inter onset interval* (ioi), se genera a partir de la diferencia entre el valor actual del pulso y el siguiente ), para de esta manera obtener la diferencia que existe entre cada dato. Esto se hace ya que por lo general las canciones de salsa tienen un ritmo constante, lo que genera que el ioi tienda a ser muy parecido para todos los valores del arreglo. En la figura 14, se puede apreciar de una manera gráfica los valores del ioi para una canción, en el eje x se cuenta con una línea de tiempo, mientras en el eje y se presenta el valor del ioi en ese momento. Como se aprecia en la figura 14 al inicio los valores no cambian mucho, es claro que los datos no son perfectos, por lo tanto, se procede a eliminar los valores de ioi demasiado altos. Cuando se presenta un valor muy alto significa que hay un vacío en el intervalo de tiempo siguiente como se puede apreciar entre los 190 y 210 segundos, esto se debe a que en ese intervalo los datos de los diferentes usuarios no cuentan con características en común, en otras palabras, no son similares.

(5)

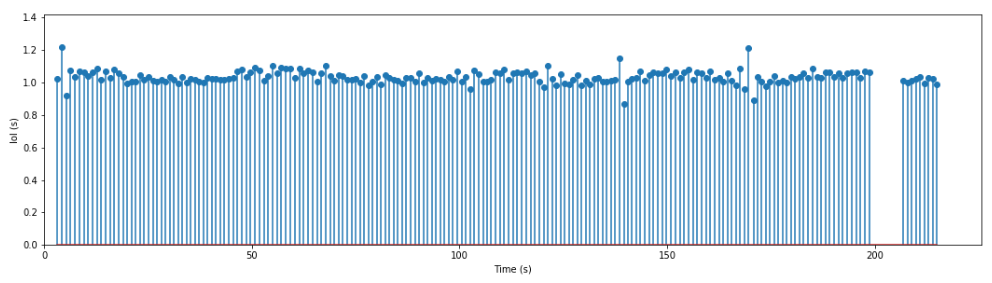


Figura 14 Visualización del ioi para una canción

### Creación de los datos de entrada

Al contar con listas unificadas, es necesario establecer una manera de suministrar los datos al modelo para que este tenga la capacidad de analizar la información de la canción y las posiciones de los pulsos. Como las canciones se encuentran en formato mp3, es necesario extraer las características más importantes de estos archivos, para este proceso se crean espectrogramas mel para cada canción. Un espectrograma mel, es una manera de representar una canción en la forma más cercana a la que los seres humanos perciben los sonidos, además de que en el proceso de generación del espectrograma de aplican una serie de filtros y parámetros para obtener una representación más condensada, por lo tanto se reduce el tamaño de la matriz del espectrograma (Arumugam & Shanmugamani, 2018). Al aplicar este tipo de espectrogramas a los datos de una canción, gráficamente se ve como en la figura 15.

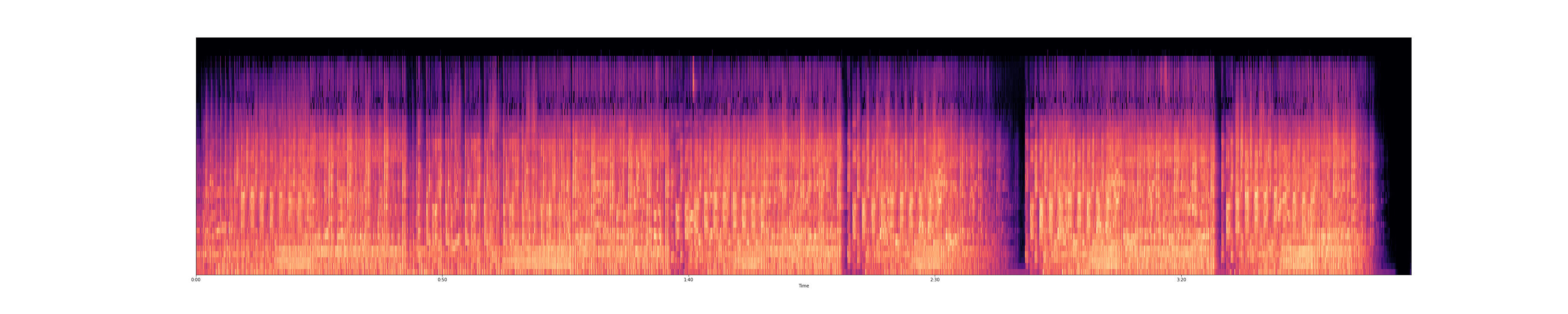


Figura 15 Espectrograma mel de una canción

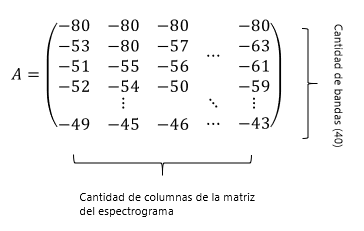
Para la creación de estos espectrogramas se utilizan los siguientes parámetros: *sample rate* de 48.000 Hz, 40 bandas, tamaño de ventana de 512 y un *overlap* de 0,5. Con estos parámetros se genera una matriz que se almacena para su posterior uso, donde el tamaño de las filas es siempre 40, ya que este es el valor asociado a la cantidad de bandas que posee el mismo. Mientras la cantidad de columnas siempre es variable ya que depende de la duración de la canción, la duración de una sola columna tiene un valor aproximado de 10 milisegundos. En la figura 16, es posible apreciar la matriz A que es un ejemplo de cómo se puede ver un espectrograma en forma de matriz para una canción.

Figura 16 Ejemplo de matriz de espectrograma

Para la creación del conjunto de datos se procede canción por canción. Con los datos de la matriz del espectrograma se procede a crear una ventana de tiempo de 230 milisegundos y esta avanza moviéndose 10 milisegundos (una columna de la matriz) hasta llegar al final de la canción. Donde cada una de estas ventanas se almacena para de esta forma crear una matriz de 3 dimensiones, en la figura 17 se puede apreciar el proceso de una manera gráfica. Donde primero se realiza la traspuesta de la matriz A para de esta manera acceder a los datos de una manera más sencilla, para de esta manera proceder con la creación de las ventanas como se aprecia en la figura.

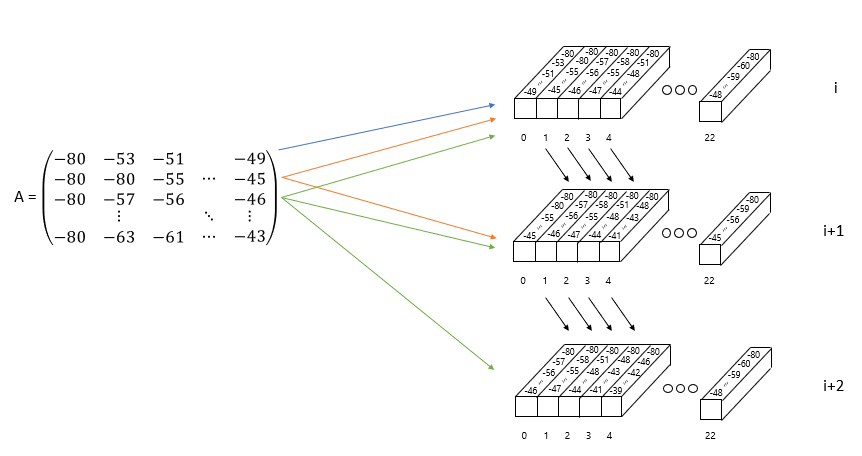


Figura 17 Proceso para la creación de los datos

Cómo estas listas de pulsos son una sucesión de valores de tiempo, es necesario asignar este valor a una columna del espectrograma. Para esto primero se calcula cual es la duración en tiempo de una columna, este cálculo corresponde a la fórmula 6, donde es la duración de la columna en segundos, corresponde a la duración de la canción en segundos y es el número total de columnas que tiene la matriz del espectrograma generado para esa canción. Ya contando con el valor de la duración en tiempo de una columna del espectrograma, se aplica la fórmula 7 para encontrar a qué columna corresponde ese valor, en dónde corresponde al número de la columna a la que corresponde ese tiempo, en otras palabras, la posición que ocupa en la matriz, es la ubicación del pulso en segundos y es la duración de la columna en segundos. Estos cálculos se realizan para todas las canciones y se almacenan en una lista.

Ahora, para agregar las posiciones de los pulsos a cada ventana solo se tienen en cuanta los primeros 30 milisegundos de la ventana de 230 milisegundos. Es decir, como ya se posee una lista con el número de la columna y un 1 o un 0 si posee o no un pulso, se revisa si alguna de las columnas que se encuentran en estos primeros 30 milisegundos corresponden al valor de un 1. En caso de que alguna de estas columnas posea el valor de un uno, se le asigna a toda la ventana de 230 milisegundos el valor de 1, es decir, que posee un pulso. Este proceso se realiza para todas las ventanas, por este motivo siempre en el data set se van a encontrar 3 valores de 1 seguidos. Este proceso se puede ver de una manera gráfica en la figura 18.

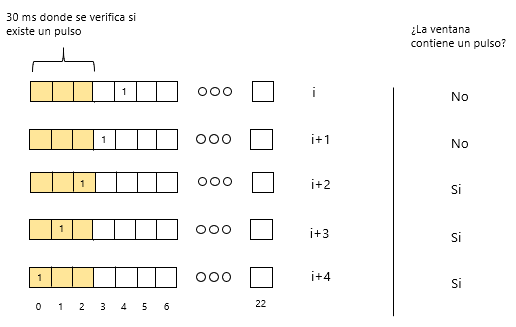


Figura 18 Verificación existencia un pulso en la ventana.

### Modelado

Para la creación del modelo se establece que se van a utilizan redes neuronales convolucionales. Para esta de define un modelo secuencial al cual se le agregan dos capas convolucionales que aplican un filtro de una dimensión, estas tienen los tamaños de 12 y 24 neuronas respectivamente y un *kernel size* de 3. Luego se establece un *flatten* que lo que hace es eliminar todas las dimensiones excepto por una, es decir, convierte la entrada a una salida de una sola dimensión. Para después agregar una capa completamente conectada de 16 neuromas, donde su salida entra en una capa de *dropout* con un valor de 0.4, esto quiere decir que asigna un peso de cero a un 40% de las neuronas, para reducir la probabilidad de sobreaprendizaje*.* Para culminar con una neurona con el peso de si el dato ingresado es un pulso o no. La estructura de este modelo se aprecia de una mejor manera en la figura 19.

Los datos resultantes de la etapa anterior se dividen en un 75% destinado exclusivamente para el entrenamiento del modelo, mientras el 25% restante para se utiliza para realizar la evaluación del modelo. Posteriormente, se procede a entrenar el modelo por 2800 épocas.

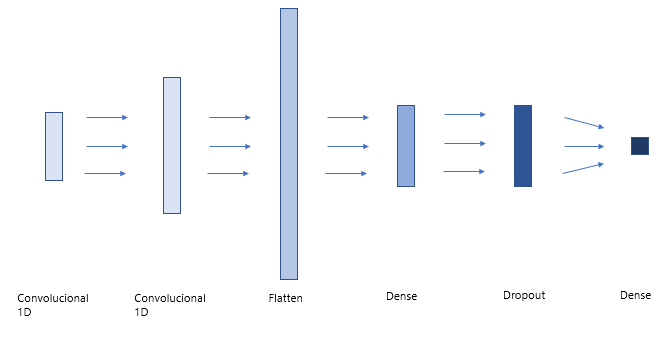


Figura 19 Estructura de la red neuronal

### Evaluación

Después de encontrar el mejor modelo con la mejor configuración de estructura y parámetros, se procede a evaluar con la otra sección del conjunto de datos, que desde el inicio fue destinado para la evaluación. Al realizar la validación se puede apreciar que la métrica del *Accuracy* (calculado con la fórmula 8) presenta un incremento, ya que el base line es de 97,13% mientras que al evaluar los datos con el modelo se presenta un resultado del 99,89%.

De la misma manera, al recopilar los valores que se generan en cada época de entrenamiento del modelo, es posible generar las gráficas que se presentan en las figuras 20, 21, 22 y 23. En las figuras 21, 22 y 23, es posible encontrar las comparaciones de los resultados del modelo para el entrenamiento y la validación de las métricas auc, precisión y recall. Donde en el eje x se encuentran cada una de las épocas, mientras en el eje y se ubica el valor de la métrica correspondiente a ese instante y figura. Se puede observar que a medida que transcurren las épocas, los valores de auc, precisión y recall del modelo aumentan mientras las pérdidas disminuyen y se estabilizan, como se aprecia en la figura 20, donde el eje x corresponde a las épocas y el eje y representa el valor de la pérdida en ese momento. Los valores finales de las métricas son un accuracy de 99,89%, precisión de 98,48%, auc de 99,19%, recall de 97,46% y un kappa de 97,91%, que indica que el modelo mejora en gran manera respecto a la línea base. Por lo tanto, es posible afirmar que el modelo es capaz de predecir el beat en las canciones de salsa y lo hace de una manera eficaz.

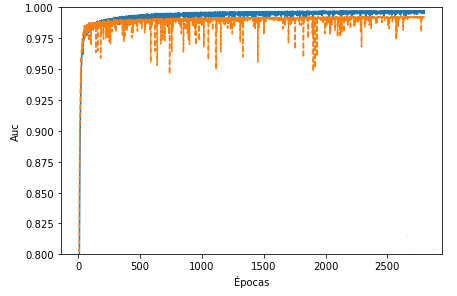
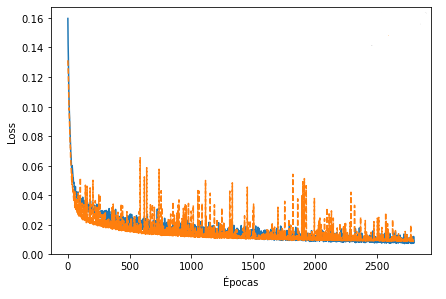
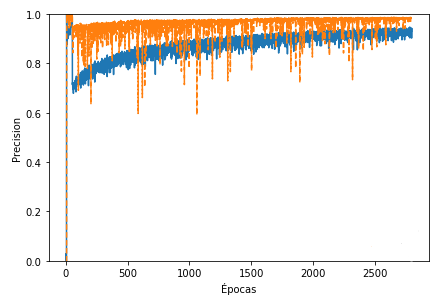
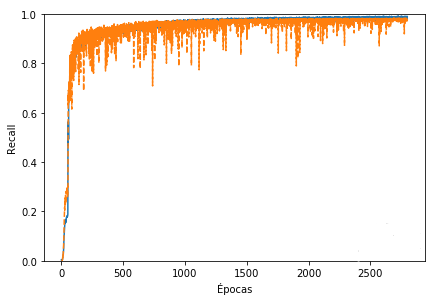


Figura 23 Evolución del Recall a través de las épocas

Figura 22 Evolución de la precisión a través de las épocas

Figura 21 Evolución del Auc a través de las épocas

Figura 20 Evolución de las pérdidas a través de las épocas

## Análisis de los sujetos

Al contar con diferentes tipos de usuarios que marcaron canciones de salsa, es posible contar con diversas maneras de interpretar una canción. Los usuarios resolvieron un breve cuestionario que cuenta con las siguientes preguntas:

1. ¿Por cuántos años usted ha estudiado música?
2. ¿Cuántos años lleva escuchando música salsa?
3. ¿Por cuántos años usted ha estudiado música salsa?
4. ¿Cuántos años lleva bailando música salsa?

Se procede a comparar si alguna de estas variables se encuentra relacionada con obtener mejores valores en el sharpe ratio de cada usuario. Para esto de realiza una correlación como prueba de hipótesis, donde las hipótesis para cada una de las variables evaluadas son las siguientes:

Tabla 7 Hipótesis clasificadas por variable.

|  |  |
| --- | --- |
| **Variable** | **Hipótesis** |
| Años estudiando música | : Los usuarios con más años estudiando música no tienen un mayor valor del sharpe ratio.  : Los usuarios con más años estudiando música tienen un mayor valor del sharpe ratio. |
| Años escuchando música salsa | : Los usuarios con más años escuchando música salsa no tienen un mayor valor del sharpe ratio.  : Los usuarios con más años escuchando música salsa tienen un mayor valor del sharpe ratio. |
| Años estudiando música salsa | : Los usuarios con más años estudiando música salsa no tienen un mayor valor del sharpe ratio.  : Los usuarios con más años estudiando música salsa tienen un mayor valor del sharpe ratio. |
| Años bailando música salsa | : Los usuarios con más años bailando música salsa no tienen un mayor valor del sharpe ratio.  : Los usuarios con más años bailando música salsa tienen un mayor valor del sharpe ratio. |

El software que se utiliza para hacer la correlación de Pearson entre las variables seleccionadas y el sharpe ratio se llama *IBM SPSS Statistics.* Esta herramienta permite realizar este tipo de pruebas de manera sencilla y rápida. Se establece un nivel de significancia (alfa) α = 5%. En la tabla 8, es posible apreciar los valores de la media, error estándar, índice de confianza del 95% límite superior e inferior y el valor p, que es el resultado de realizar la prueba de Pearson sobre los datos seleccionados.

Tabla 8 Resultados de las pruebas para las variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Años estudiando música** | **Años escuchando música salsa** | **Años estudiando música salsa** | **Años bailando música salsa** |
| **Media** | 5,92 | 7,23 | 0,92 | 1,92 |
| **Error estándar** | 1,37 | 2,05 | 0,42 | 0,55 |
| **IC 95% límite inferior** | 3,25 | 3,21 | 0,11 | 0,85 |
| **IC 95% límite superior** | 8,60 | 11,25 | 1,74 | 3,00 |
| **Valor P** | 66% | 27% | 33% | 6% |

Gracias al valor p, es posible realizar la toma de decisiones sobre cada una de las variables. Como ninguno de los valores p fue menor al nivel de significancia establecido, todas las hipótesis se rechazan. En la tabla 9, se establecen las conclusiones para cada una de las variables de acuerdo con las hipótesis y las pruebas realizadas.

Tabla 9 Resultados de la comparación

|  |  |
| --- | --- |
| **Variable** | **Conclusión** |
| Años estudiando música | Los usuarios con más años estudiando música no tienen un mayor valor del sharpe ratio. |
| Años escuchando música salsa | Los usuarios con más años escuchando música salsa no tienen un mayor valor del sharpe ratio. |
| Años estudiando música salsa | Los usuarios con más años estudiando música salsa no tienen un mayor valor del sharpe ratio. |
| Años bailando música salsa | Los usuarios con más años bailando música salsa no tienen un mayor valor del sharpe ratio. |

## Conclusiones

La música salsa es un género que resalta en nuestra cultura latinoamericana, por este motivo es importante que se generen herramientas y aportes que puedan ser de utilidad para futuros investigadores en esta área.

Por medio de la herramienta desarrollada en este proyecto, es posible replicar este tipo de experimentos y adaptarlos de acuerdo con lo que se necesite. Gracias a la herramienta fue posible la creación de una base de datos de canciones del género salsa con sus correspondientes pulsos debidamente anotados, esto puede generar un incentivo a todos los futuros investigadores para explorar este género. Es importante para nuestra cultura que se presenten en mayor cantidad este tipo de aportes que nos den un tipo de visualización adicional, todo esto para que sea más atractivo para personas interesadas en esta área poder iniciar su investigación, ya que cuentan con suficientes insumos para una rápida y efectiva implementación.

Igualmente, se presenta una manera diferente de entregar los datos a un modelo. Para que, por medio del entrenamiento y evaluación del modelo con nuestra base de datos, sea posible identificar el pulso en las canciones de salsa.

El objetivo es generar interés en los investigadores de todas partes del mundo, puesto que para ellos la primera opción no es una base de datos de música latinoamericana. Los investigadores extranjeros prefieren entrenar sus modelos con la música de la cultura pop, géneros conocidos y de mucho éxito, es por este motivo que en los artículos descritos en el estado del arte todos usan bases de datos comunes con géneros muy conocidos. Por estas razones, los modelos entrenados con este tipo de data sets van a fallar al tratar de reconocer el pulso en canciones de música salsa, dado que cuentan con múltiples de diferencias.

Es por esto por lo que los investigadores, no solo deben entrenar sus modelos con datos comunes y cotidianos. Sino también deben de tener en cuenta otras culturas, dado que estas generan datos que no son tan comunes y pueden ser de utilidad. Por lo tanto, es un llamado para que los investigadores piensen en entrenar y evaluar sus modelos no solo con los data sets a los que están acostumbrados, sino también con datos provenientes de otras culturas y lugares geográficos.

## Trabajo futuro

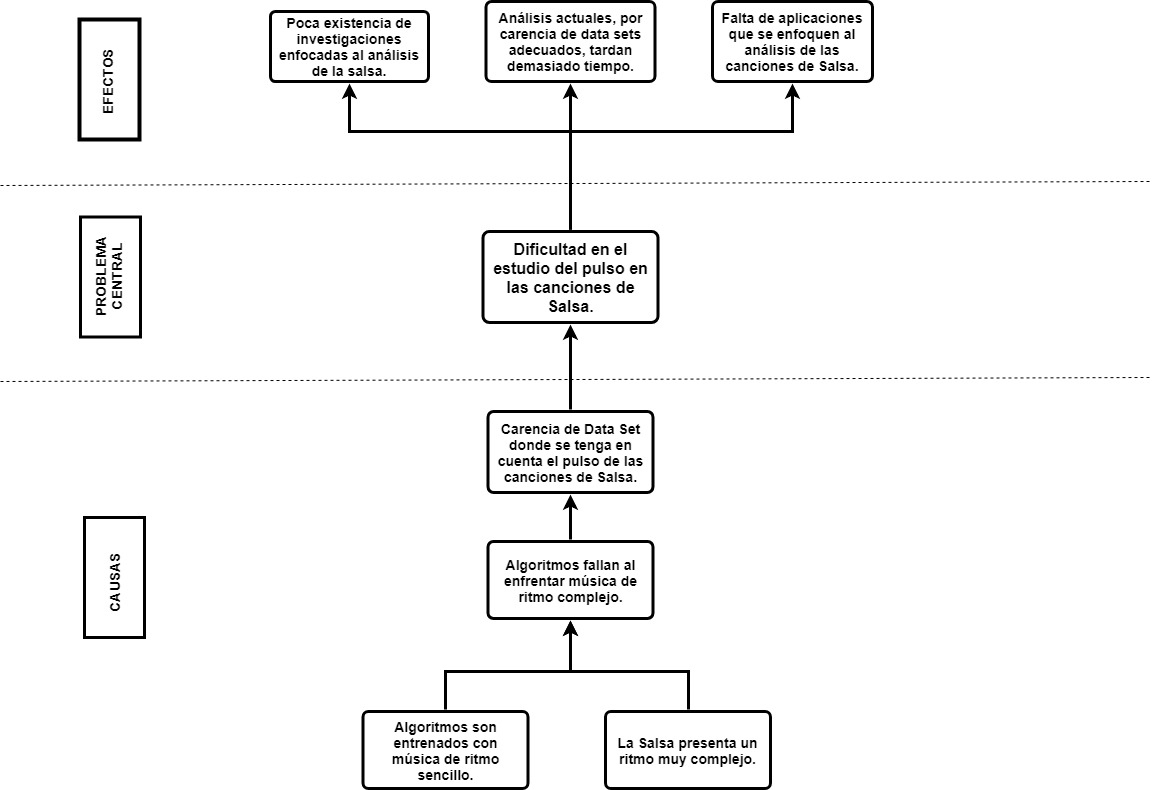
En la duración del proyecto lastimosamente se presentó una baja participación de personas interesadas en realizar la marcación del pulso en las canciones utilizando la herramienta. Por lo tanto, será de mucha importancia aumentar la cantidad de canciones marcadas para el desarrollo de una base de datos con mayor calidad y que sea más atractiva de usar. Para aumentar el nivel de participación sería interesante el desarrollo de la herramienta en una versión web, ya que al tener una versión en línea será más fácil el acceso para los estudiantes y personas interesadas en participar en este experimento, al contrario de cómo está establecido para este proyecto que es una participación presencial. De la misma manera, puede ser interesante contar con una mayor cantidad de personas. Puesto que permitiría contar con puntos de vista diferentes y abarcar una mayor cantidad de canciones.

Debido a que el proyecto solo se centra en entrenar y evaluar modelos para la detección del pulso en canciones de salsa, no se implementaron otro tipo de géneros en la base de datos. Puede ser interesante evaluar el impacto de esta adición en el momento del desarrollo de los modelos. Además, se puede pensar en otros tipos de modelos diferentes y en la forma en cómo se procesan los datos, todo esto para determinar cuál es la manera más efectiva de encontrar el pulso en las canciones de salsa.

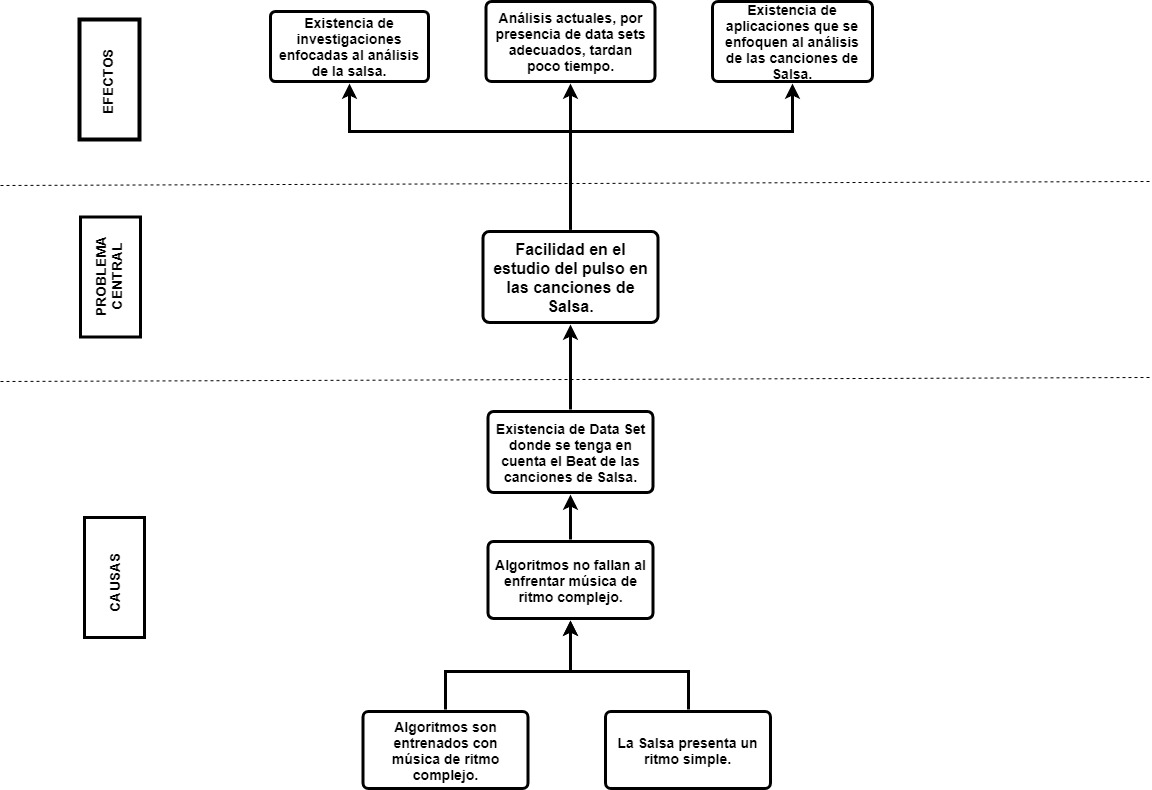
Por otro lado, la elaboración de los datos de entrada del modelo es una laboriosa tarea que requiere de un tiempo para procesarse. Por estos motivos, es de suma importancia encontrar la manera de optimizar esta creación o de generar otro tipo de entradas para optimizar este tiempo de procesado. Todo esto para tener la capacidad de ingresar los datos de una canción al modelo y que este prediga las ubicaciones del pulso en tiempo real.

## Anexos

### Anexo 1. Árbol del problema.

****

### Anexo 2. Árbol de objetivos.

****

### Anexo 3. Análisis de participación.

* Investigadores de carácter musical interesados en el género salsa: Son beneficiarios directos, dado que el proyecto genera herramientas para que puedan hacer sus investigaciones con mayor facilidad.
* Estudiantes de música: son beneficiarios directos, por lo tanto, se podrá generar mayor conocimiento en cuanto al género salsa.
* El género salsa: Es un beneficiario indirecto, debido a que con las herramientas generadas es posible generar un mayor conocimiento sobre este género.
* Los profesores de música: Son beneficiarios indirectos, de manera que se aumenta el conocimiento de la salsa, se pueden llegar a brindar mejores clases y proyectos.
* Los desarrolladores de software enfocado a aplicaciones en las que se utilice música: Son beneficiarios directos, de modo que la generación de estos algoritmos puede servir como base para futuros desarrollos.

## Referencias bibliográficas

Apache Friends. (2017). XAMPP Installers and Downloads for Apache Friends. https://doi.org/10.2212/spr.2011.1.4

Arumugam, R., & Shanmugamani, R. (2018). *Hands-On Natural Language Processing with Python: A Practical Guide to Applying Deep Learning Architectures to Your NLP Applications*. Retrieved from https://books.google.com.co/books?id=ipplDwAAQBAJ&pg=PA243&lpg=PA243&dq=mel-scaled+spectrogram.&source=bl&ots=nXFOR5vO-2&sig=ACfU3U2WyxuNtJIhmfQ9W36QpGLTHSj87g&hl=es-419&sa=X&ved=2ahUKEwj1hdnS-ZrmAhWC1FkKHcYsDUQ4ChDoATASegQICRAE#v=onepage&q=mel-scaled spectrogram.&f=false

Ballroom. (2004). Retrieved April 10, 2019, from http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html

Böck, S., Krebs, F., & Widmer, G. (2016). Joint Beat and Downbeat tracking with recurrent neural networks. In *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference* (pp. 255–261). Retrieved from https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/186\_Paper.pdf

Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). *CONVOLUTIONAL RECURRENT NEURAL NETWORKS FOR MUSIC CLASSIFICATIONChoi, K., Fazekas, G., Sandler, M., & Cho, K. (n.d.). CONVOLUTIONAL RECURRENT NEURAL NETWORKS FOR MUSIC CLASSIFICATION. Retrieved from https://github.com/keunwoochoi/icassp\_2017*. Retrieved from https://github.com/keunwoochoi/icassp\_2017

Davies, M., Degara, N., & Plumbley, M. (2009). *Evaluation methods for musical audio beat tracking algorithms, Technical Report C4DM-TR-09-06*. *… University, Centre for Digital Music …*. Retrieved from https://www.academia.edu/1099851/Evaluation\_methods\_for\_musical\_audio\_beat\_tracking\_algorithms

Davies, M. E. P., & Plumbley, M. D. (2007). Context-Dependent Beat Tracking of Musical Audio. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, *15*(3). https://doi.org/10.1109/TASL.2006.885257

Definition of ground-truth in English by Oxford Dictionaries. (n.d.). Retrieved March 7, 2019, from https://en.oxforddictionaries.com/definition/ground-truth

Downie, J. S. (2008). The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, *29*(4), 247–255. https://doi.org/10.1250/ast.29.247

Dunn, T. (2019). Deep learning: Discovery Service para Universidad ICESI. *Salem Press Encyclopedia of Science, 2019. 2p.* Retrieved from http://eds.a.ebscohost.com/eds/detail/detail?vid=3&sid=f08f6fee-84dd-4088-b6cd-8815ae3ef498%40sdc-v-sessmgr06&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbGl2ZSZzY29wZT1zaXRl#AN=119214353&db=ers

Durand, S., Bello, J. P., David, B., & Richard, G. (2017). Robust downbeat tracking using an ensemble of convolutional networks. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *25*(1), 72–85. https://doi.org/10.1109/TASLP.2016.2623565

IBM, I. B. M. (2012). Manual CRISP-DM de IBM SPSS Modeler. *IBM Corporation*, 56. Retrieved from ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/15.0/es/CRISP-DM.pdf

IEM. (2014). Pure Data - PD Community Site. Retrieved March 7, 2019, from https://puredata.info/

ISMIR. (2015). The International Society of Music Information Retrieval. Retrieved April 9, 2019, from https://www.ismir.net/

Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*. MIT Press.

Marchand, U., & Peeters, G. (2016). The Extended Ballroom Dataset. In *ISMIR 2016 Late-Breaking Session*. Retrieved from http://anasynth.ircam.fr/home/media/ExtendedBallroom.

Nvidia. (2018). Convolutional Neural Network (CNN) | NVIDIA Developer. Retrieved April 13, 2019, from https://developer.nvidia.com/discover/convolutional-neural-network

Pigden, N., Young, F. W., & Hamer, R. M. (1988). Multidimensional Scaling: History, Theory and Applications. *The Statistician*, *37*(1), 90. https://doi.org/10.2307/2348396

Pullen, J. P. (2015). Streaming Showdown: Apple Music vs. Spotify vs. Pandora vs. Rdio: Discovery Service para Universidad ICESI. Retrieved March 6, 2019, from http://eds.a.ebscohost.com/eds/detail/detail?vid=0&sid=e7fea650-2236-45fe-bdd6-b0419be7db5d%40sessionmgr4010&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbGl2ZSZzY29wZT1zaXRl#db=bsu&AN=103138827

RAE. (2010a). Definición de base - Diccionario de la lengua española - Edición del Tricentenario. Retrieved from https://dle.rae.es/?id=5ASmP2Z

RAE. (2010b). Definición de ritmo - Diccionario de la lengua española - Edición del Tricentenario. Retrieved March 7, 2019, from https://dle.rae.es/?id=WWXSc4r

Rondón, C. M. (2017). *El Libro de la Salsa: Cronica de la Musica del Caribe Urbano*. Ediciones B.

Sharpe, W. F. (1994). The Sharpe Ratio. *The Journal of Portfolio Management*, *21*(1), 49–58. https://doi.org/10.3905/jpm.1994.409501

The University of Edinburgh. (2008). Kernel Density Estimators (pp. 125–193). https://doi.org/10.1002/9780470316849.ch6

Urbas, J. V. (2018). Neural Networks: Discovery Service para Universidad ICESI. *Salem Press Encyclopedia of Science*. Retrieved from http://nebulosa.icesi.edu.co:2118/eds/detail/detail?vid=8&sid=051a7f42-897b-42f4-ac7b-727af108d0c5%40pdc-v-sessmgr02&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbGl2ZSZzY29wZT1zaXRl#db=ers&AN=89317103

Winter, M. (2018). Salsa (music). Retrieved March 6, 2019, from http://eds.a.ebscohost.com/eds/detail/detail?vid=3&sid=97c452e9-acbb-40e7-bc75-5a3c2890671d%40sessionmgr4010&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbGl2ZSZzY29wZT1zaXRl#AN=87322775&db=ers