



UNIVERSIDADE  
LUSÓFONA

# Classificação de géneros musicais com integração de bases de dados

## Trabalho Final de curso

Relatório Intercalar 1º Semestre

Nome do Aluno: Eduardo Colaço Marques

Nome do Orientador: professor Fernando Teodósio

Trabalho Final de Curso | LEI | 23 de Julho 2021

[www.lusofona.pt](http://www.lusofona.pt)

## Direitos de cópia

*(Classificação de géneros musicais com integração de bases de dados)*, Copyright de *(Eduardo Colaço Marques)*, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

## Resumo

Neste relatório está descrito o projeto desenvolvido no âmbito do trabalho de final de curso atribuído pelo professor Fernando Teodósio.

Após várias iterações este trabalho obteve o objetivo do desenvolvimento em python de uma prova de conceito de um sistema de classificação de géneros musicais com recurso a uma rede neural convolucional[1] e inserção dinâmica em bases de dados .

A classificação no seu estado atual consegue analisar ficheiros do tipo “.mp3” e “.wav” através dos coeficientes cepstrais de frequência de mel[2] e passando esses valores por uma rede neural convolucional treinada com recurso ao conjunto de dados GTZAN[3] devolve uma previsão do género musical do ficheiro.

No que toca à inserção dos dados obtidos, não foi possível automatizar o processo nem criar compatibilidade para diferentes tipos de base de dados, no entanto é possível ao utilizador selecionar as qualquer tabela numa base de dados SQL e colocar uma entrada na mesma inclusive de ficheiros através da conversão dos mesmos para “*binary large objects*” ou *BLOBS*.

No futuro sinto que este projeto poderia continuar a ser desenvolvido tanto para melhorar a precisão da rede neural através de melhorias no algoritmo tanto para criar integração com diferentes tipos de bases de dados como “*MongoDB*” , “*MariaDB*” entre outras.

**Palavras-chave:** trabalho de final de curso, rede neural convolucional, bases de dados

## Abstract

This document describes the final work for the course of computing engineering from the university Lusófona de Humanidades e Tecnologias.

After several iterations this project obtained its final objective in the form of the development in python of a proof-of-concept app that classifies musical genres through a convolutional neural network [1] and allows dynamic database insertions.

The classification in its current state can analyse *“.mp3” and “.wav” type files* by passing their mel frequency cepstral coefficients[2] through a neural network which was trained with aid from the GTZAN dataset[3] to return the audio file's musical genre

When it comes to database insertions it wasn't possible to automate the process nor add compatibility with different types of databases, however it's possible for the user to select any table in the database and make an entry including files by converting them to binary large objects or BLOBS

In the future I feel like this project could see continued development both to increase the neural network precision through algorithm improvements, but also to create integration with different types of databases such as MongoDB, MariaDB amongst many others

**keywords:** final project, convolutional neural network, databases

# Índice

Resumo.....	iii
Abstract.....	iv
Índice.....	v
Lista de Figuras.....	vi
Lista de Tabelas.....	vii
1 Identificação do Problema.....	1
2 Viabilidade e Pertinência.....	3
3 Levantamento e análise de requisitos.....	4
Tabela de requisitos.....	4
4 Solução Desenvolvida.....	6
Tabela de Tecnologias usadas.....	6
4.1.1 Tratamento de Dados.....	7
4.1.2 Rede Neural Convolucional.....	8
4.1.3 Bases de dados.....	9
4.1.4 Interface.....	10
5 Benchmarking.....	13
6 Conclusões.....	14
7 Calendário.....	15
Bibliografia.....	16
Glossário.....	17

## Lista de Figuras

<b>Figura 1</b> -cotas do mercado de streaming de música no 1º quartil de 2021	1
<b>Figura 2</b> -Obtenção de MFCC's	7
<b>Figura 3</b> -Tabela de testes criada na base de dados	9
<b>Figura 4</b> -User Interface da aplicação	10
<b>Figura 5</b> -Função de classificação de músicas	11
<b>Figura 6</b> -Função de criação de entradas na base de dados	12

# Lista de Tabelas

tabela de requisitos funcionais----- 4

tabela de requisitos não funcionais----- 5

tabela de tecnologias usadas-----6

# 1 Identificação do Problema

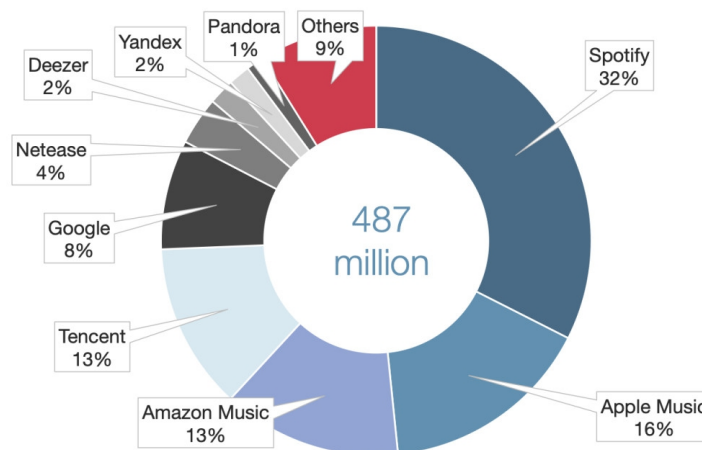
Este capítulo relata o enquadramento do problemas que identificamos para num capítulo seguinte desenvolver-mos a sua solução.

O desenvolvimento deste trabalho começou com o simples tema da classificação automática de géneros musicais mas após um breve estudo da arte foi nos possível concluir que a área de estudo já se encontrava bastante saturada sendo que grandes empresas já empregam tecnologias derivadas destes estudos como é o caso do *spotify* e do seu *echo nest*[4] que é um sistema que entre outras funções identifica e organiza as musicas presentes na base de dados por géneros musicais.

Decidimos após a conclusão anterior alterar e perspectiva com que abordáramos o trabalho, e após um estudo do mercado foi nos chegou-se á conclusão que mais de 50% do mercado de streaming de música pertence a grandes empresas:

## Spotify maintains subscriber market share lead but Amazon, Tencent and Google made big gains

Streaming music subscriber market shares, Q1 2021, global



Source: MIDiA Research Music Model (06/21)

MIDiA.

**Fig.1-cotas do mercado de streaming de música no 1º quartil de 2021**

Empresas que como o exemplo analisado anteriormente, têm os seus sistemas privados de classificação e organização géneros musicais, sistemas demasiado dispendiosos para serem criados[5] para novas empresas que queiram entrar no mercado. Uma alternativa que seria possível para estas empresas seria a utilização de um dos vários projetos já existentes no entanto cada entrada na base de dados teria que ser feita à mão.

É neste contexto que surge a solução proposta por este relatório, isto é, um sistema que agrupe o processo de classificação e de inserção de dados nas bases de dados através de uma aplicação open-source desenvolvida em python com recurso a diversas bibliotecas pré-existentes de modo a que possa ser utilizado por empresas que se queiram estabelecer no mercado mas não tenham o mesmo capital da competição.



No entanto após desenvolvimento e conclusão do projeto devo salientar que o projeto existe num estado de prova-de-conceito e que carece ainda de bastante desenvolvimento sendo que n me foi possível implementar alguns do requisitos planeados inicialmente quer por falta de conhecimentos como por limitações relacionadas com o tempo para o desenvolvimento.

## 2 Viabilidade e Pertinência

Neste capítulo verificamos a pertinência e viabilidade de mercado do trabalho em desenvolvimento através de um estudo da bibliografia

Como foi possível analisar no capítulo anterior O mercado de streaming de música está dominado por grandes empresas[4] que por sua vez desenvolveram sistemas privados ou proprietários de organização e classificação de músicas dentro das suas próprias bases de dados, sistemas muitas vezes desenvolvidos pela própria empresa ou por subsidiárias cujo o capital é na ordem dos milhões[4].

Este trabalho encontra a sua viabilidade na existência de pequenas empresas que não consigam ou não estejam dispostas desenvolver sistemas próprios de classificação embora estas consistam menos de 9% do mercado sinto que é justificável o desenvolvimento de ferramentas que estas possam usar.

No entanto o ponto anterior não justificado o trabalho por si só, sinto que uma das mais valias deste projeto é o seu desenvolvimento em open-source num repositório github permitindo a qualquer pessoa desenvolve-lo posteriormente inclusive futuros alunos.

Após testes preliminares a versão está funcional como protótipo ou prova-de-conceito, garantindo a análise relativamente precisa de músicas assim como a integridade de dados com as bases de dados usadas sendo que por enquanto apenas aceita ligações a bases de dados SQL o que demonstra a sua possível utilidade no futuro.

Hoje em dias existem aplicações em larga escala, tal como o *spotify* que por possuírem largas bibliotecas de música tiveram a necessidade de desenvolver aplicações de classificação automática de músicas sempre que estas são adicionadas á sua biblioteca de faixas sonoras, neste caso o foi desenvolvido o *Echo nest*[6], desenvolvido em 2005 com código proprietário.

Tendo o *spotify* e a *apple music* uma cota do mercado de *streaming* de música superior a 50% é difícil aplicações sem o mesmo nível de capital destacaram-se, portanto muitas fazem uso de programas open-source semelhantes ao que está a ser desenvolvido sendo que muitas vezes estes não todas as capacidades que um sistema de relativamente grande escala necessitaria, o que nos prova que este tipo de programas têm alguma procura a nível de mercado.

Planeando desenvolver uma aplicação classificadora de música automática por si só não teria pertinência quase nenhuma, no entanto o facto da aplicação que está planeada ir ser desenvolvida em open-source viabiliza o projeto no sentido em que outro investigador ou entidade possa toma-lo como base para o desenvolvimento de uma aplicação mais completa que até possa ser disponibilizada em *app stores*.

### 3 Levantamento e análise de requisitos

Este tópico é responsável pela listagem de requisitos necessários para o desenvolvimento do projeto.

Após uma reflexão sobre funções necessárias ao funcionamento da aplicação proposta foi necessário criar uma lista de requisitos para a mesma.

**Tabela 1- tabela de requisitos funcionais**

Tabela de requisitos			
Titulo	Descrição	Critério de aceitação	Implementação
<i>Input de ficheiros na aplicação</i>	A aplicação deve aceitar a introdução de ficheiros de som por parte do utilizador	A)O ficheiro tem formato correto	Concluída
Análise de ficheiros	A aplicação deve automaticamente analisar o conteúdo do ficheiro e se possível classificá-lo	A) O ficheiro tem de ter uma duração mínima de 1 minuto	Parcialmente
<i>Input de dados numa base de dados</i>	Após a análise do ficheiro permitir ao utilizador colocar os dados associados dentro de uma base de dados	A) Conexão à base de dados a funcionar corretamente B) Todos os dados obrigatórios estão presentes	Concluída
<i>Visualização de itens presentes na base de dados</i>	Permite ao utilizador verificar que musicas e que géneros já estão presentes na base de dados	A)Conexão à base de dados a funcionar corretamente	Não implementado

**Tabela 1- tabela de requisitos não funcionais**

Tabela de requisitos		
Titulo	Descrição	Implementação
Verificação de duplicados	A aplicação deverá automaticamente verificar se existem itens duplicados dentro da base de dados	Não implementado
Navegabilidade	A aplicação deve ter um <i>UI</i> simples e intuitivo	Parcialmente
Garantir integridade de dados	O sistema deve garantir que os dados estão de acordo com os restantes da base de dados	Concluída

E após o desenvolvimento do projeto foi possível concluir que alguns dos requisitos não seriam possíveis de implementar por dificuldades técnicas ou por conflito com outros requisitos

## 4 Solução Desenvolvida

Neste capítulo iremos apresentar a forma como desenvolvemos a solução para o projeto planeado, delineando as tecnologias e plataformas que iremos usar.

Tabela de Tecnologias usadas	
Nome	Descrição
Python	Linguagem de programação de alto nível
<i>keras</i>	Keras é uma biblioteca de Python que permite a criação de redes neuronais e a implementação de algoritmos de <i>machine learning</i>
Librosa	Esta é uma biblioteca de Python que permite a leitura e tratamento de ficheiros de som assim como extração “features” dos mesmos
<i>Numpy</i>	É uma biblioteca vocacionada á realização de operações matemáticas de maior grau de complexidade
<i>Sklearn</i>	Biblioteca que permitiu fazer a divisão do conjunto de testes
<i>json</i>	Uma biblioteca que permite a leitura e tratamento de dados do tipo JSON
<i>Mysql connector</i>	Permite ao programa estabelecer ligações com uma base de dados e inclusive executar operações na mesma
<i>TKinter</i>	Biblioteca que permite a criação interfaces gráficas
<i>GTZAN dataset</i>	É um conjunto de dados que contem ficheiros de som de cerca de 30 segundos separados por género musical que será usado para treinar o algoritmo de deep learning

No estado corrente de desenvolvimento do trabalho realizamos o tratamento de dados que surgem na forma de um ficheiro de formato WAV , infelizmente não é possível trabalhar com os dados neste especifico formato sendo é necessário o seu tratamento..

### 4.1.1 Tratamento de Dados

O primeiro passo que foi necessário tomar neste projeto foi realizar o tratamento de dados temos na nossa posse o conjunto de dados GTZAN[] no entanto não podemos dar simples ficheiros “.wav” á nossa rede e esperar resultados é necessário decompor os mesmos para se obter “features” que possam ser analisadas pela nossa base de dados.

Primeiro passo foi determinar que “feature” analisar e após o estudo de outros trabalhos já existentes uma feature utilizada de forma bastante regular são MFCC’s.

MFCC’s ou coeficientes cepstrais da frequência de mel, explicados de uma forma sucinta são os valores numéricos que definem um “cepstrum” de frequência de mel que podem ser obtidos através da biblioteca librosa e que irão permiti rà nossa rede neural identificar padrões .

```
#MFCC feature a analisar
MFCC = librosa.feature.mfcc(signal,n_fft=numero_de_amostras_fft, hop_length=hop_length,n_mfcc=15)
```

**Fig.2-Obtenção de MFCC’s**

Onde signal é o nosso sinal obtido após se carregar o ficheiro de som, n\_fft é o numero de transformadas de fourier de curto termo que serão utilizadas e n\_mfcc é o número de coeficientes obtidos .

A escala de mel é uma escala que relaciona a frequência de um som interpretada por um ser humano com a verdadeira frequência do som.

É necessário ainda carregar estes dados para um ficheiro json juntamente com o género musical do sinal original percorrendo a biblioteca de sons GTZAN

- 1) Aplicamos a transforma de fourier discreta neste sinal para obtermos o espectro de energia do sinal

O primeiro passo foi criar um dicionário de dados vazio que iria conter os géneros possíveis de músicas, uma lista com conjuntos de MFCC’s de cada amostra

### 4.1.2 Rede Neural Convolucional

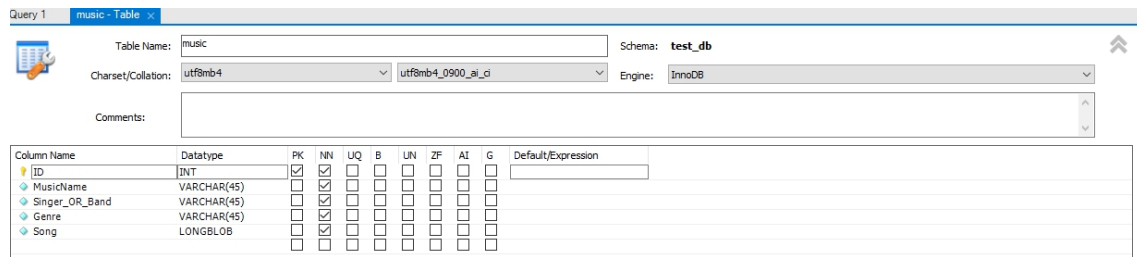
Para a análise de ficheiros de audio o primeiro passo executado foi a criação uma rede neural convolucional através da biblioteca keras foi escolhida uma rede neural convolucional ao invés de um perceptron de múltiplas camadas(MLP) pois cnn permite uma melhor análise de imagens e no futuro o método de análise do som poderá ser mudado da analise coeficientes cepstrais de frequencia de mel para a análise do espectrograma dos coeficientes cepstrais da frequencia de mel.

Foi criado um modelo sequencial 3 camadas convolucionais com 128,64 e 32 neurónios por esta ordem com método de ativação “relu” cada uma seguida de uma camada de max pooling e uma camada de “batch normalization” estas camadas estão presentes com o intuito de impedir “overfitting” na nossa rede neural.

Overfitting ocorre quando o a rede neural começa a aprender padrões de “barulho” nos conjuntos de teste que não irão existir em amostras que queiramos que esta avalie, diminuindo a probabilidade de previsões erradas.

### 4.1.3 Bases de dados

Para a testagem do funcionamento foi criada uma base de dados simples através de MYSQL workbench onde criámos também uma tabela com o seguinte formato:



Query 1 music - Table

Table Name: music Schema: test\_db

Charset/Collation: utf8mb4 utf8mb4\_0900\_ai\_ci Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MusicName	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Singer_OR_Band	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Genre	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Song	LONGBLOB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

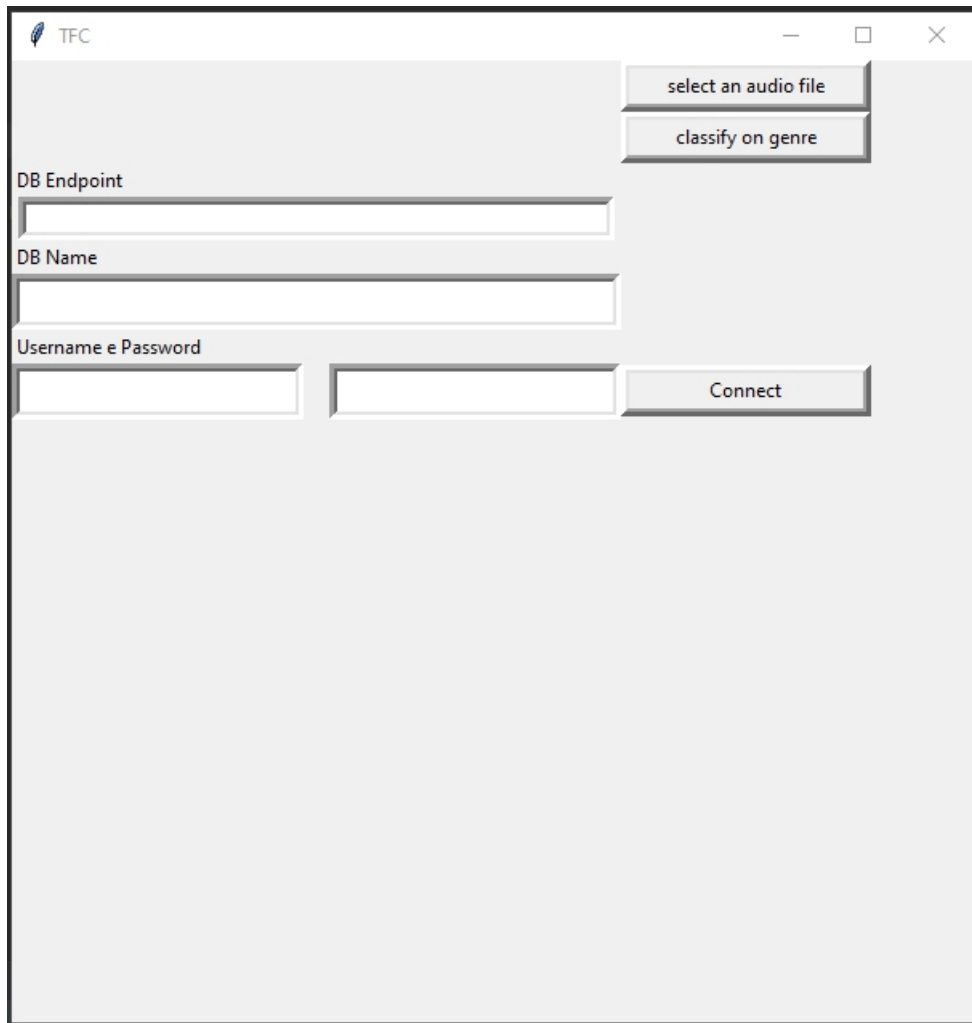
**Fig.3-**Tabela de testes criada na base de dados

Foi escolhido o tipo de dados LONGBLOB que recebe dados binários até 4GB para armazenar cada ficheiro de música, sendo poderiam existir melhor alternativas tais como guardar o ficheiro diretamente na máquina e ter uma coluna que guardasse o caminho do ficheiro.



#### 4.1.4 Interface

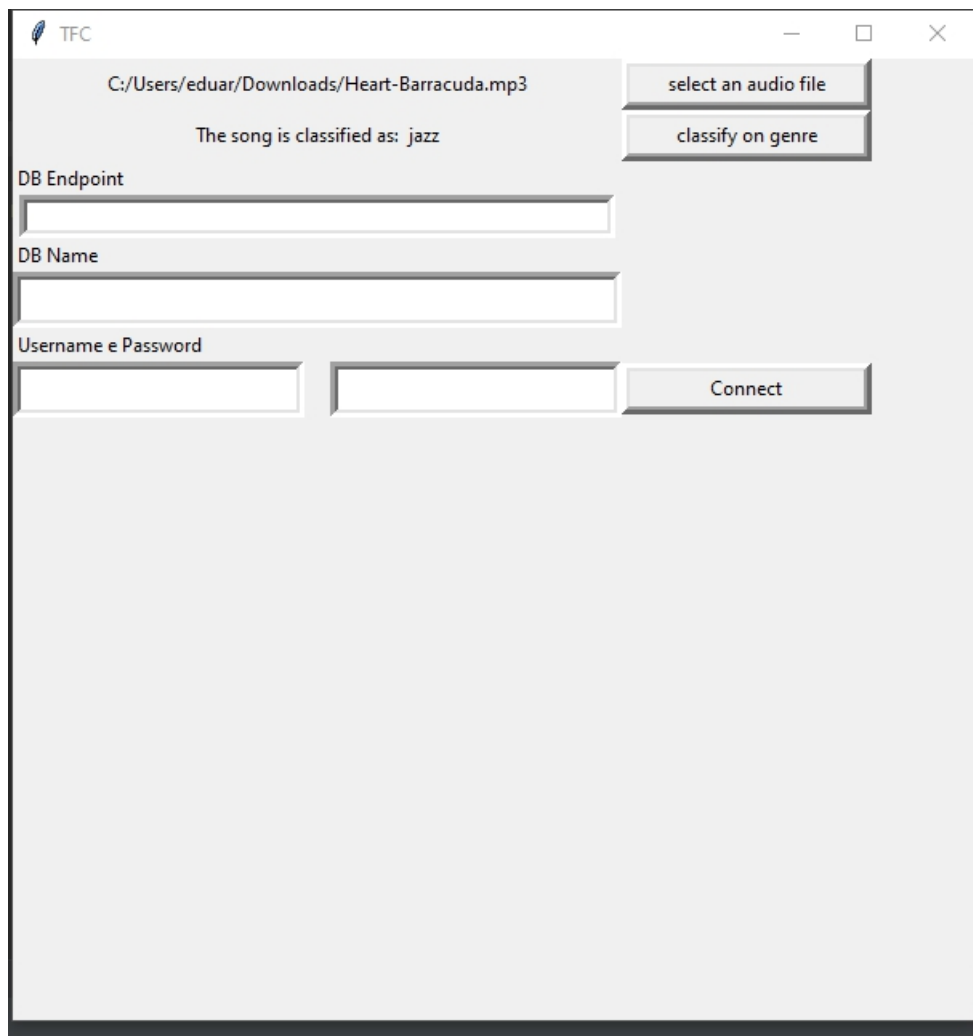
Foi criada por fim uma interface gráfica com recurso á biblioteca Tkinter sendo que é bastante simples penso que também torna a sua utilização mais intuitiva:



**Fig.4-***User Interface da aplicação*

Onde numa fase inicial o utilizador poderá seleccionar um ficheiro de som para ser analisado imediatamente mesmo sem fazer ligação à base de dados.

Sendo que a precisão da mesma precisa de algumas melhorias:



**Fig.5**-Função de classificação de músicas

Fazendo a conexão à nossa base de dados inserido o url ou Endpoint correspondente o nome da base de dados e as credenciais como deparados com um menu dropdown que nos irá permitir escolher qualquer uma das tabelas presentes na base de dados:

(password censurada porque usei uma password pessoal sem considerar o relatório)

The screenshot shows a Java Swing window titled 'TFC'. At the top, it displays the file path 'C:/Users/eduar/Downloads/Heart-Barracuda.mp3' and a status message 'The song is classified as: jazz'. There are two buttons: 'select an audio file' and 'classify on genre'. Below these are input fields for 'DB Endpoint' (containing 'localhost') and 'DB Name' (containing 'test\_db'). A section for 'Username e Password' has a 'root' username and a masked password, with a 'Connect' button. A dropdown menu is set to 'music'. At the bottom, there are labels for 'ID', 'MusicName', 'Singer\_OR\_Band', 'Genre', and 'Song', followed by buttons 'mudar o ficheiro' and 'Make database entry'.

**Fig.6-Função de criação de entradas na base de dados**

O programa deteta ainda o dataType de cada coluna e gera uma entrada correspondente sendo que datatypes Int só irão aceitar valores numéricos ,por exemplo, e temos ainda que se o programa deteta um Datatype do tipo BLOB e já tivermos classificado uma música este irá converter o ficheiro automaticamente para dados binários.

Este processo está testado e funcional para qualquer tipo de base de dados SQL no entanto é suscetível à existência de bugs que não tenham sido detetados durante o seu desenvolvimento.

## 5 Benchmarking

Neste capítulo fazemos uma comparação com a *concorrência* e outros trabalhos já existentes que possam ter o mesmo tema.

Após ter sido efetuada uma pesquisa foi possível denotar a existência de vários trabalhos já integrados nesta temática com diferentes abordagens inclusive alguns com já alguns anos o que torna a inovação de métodos algo complicada.

Um exemplo muito semelhante ao que estava planeado inicialmente para este projeto é o trabalho “Features for the Classification and Clustering of Music in Symbolic Format”[4] que estuda a classificação de música em apenas 4 géneros musicais classificados como “melodia, harmonia, baixo e bateria” com recurso a machine learning.

Um estudo que efetua uma abordagem bastante diferente está intitulado “CLASSIFYING MUSIC BY GENRE USING THE WAVELET PACKET TRANSFORM AND A ROUND-ROBIN ENSEMBLE”[5] que usa a transformada de wavelet discreta e uma classificação round-robin para determinar o género de música a partir de um ficheiro de som.

No entanto este trabalho separa-se dos restantes pela integração de um sistema de comunicação com bases de dados juntamente com o sistema de classificação de géneros algo que não me foi possível encontrar disponível no mercado.

## 6 Conclusões

Sinto que este trabalho poderia ter sido muito melhor realizado especialmente no entanto sinto que cumpri o objetivo inicial mesmo que não tenha sido da melhor forma.

Este trabalho no seu estado final terá valor apenas como prova de conceito de que o que é proposto é possível e que para ter viabilidade de mercado seria necessário desenvolvimento por alguém com mais experiência na área

No entanto sinto que também foi uma oportunidade de aprendizagem pois trabalhei com tecnologias que nunca tinha explorado anteriormente nomeadamente a API de deep learning keras e a linguagem Python em si que até ao início deste semestre não tinha utilizado, sinto no entanto que a maior dificuldade encontrada neste trabalho foi os conceitos em volta da análise de som que mesmo agora no final do trabalho não sinto que estejam completamente integrados.

## 7 Calendário

### Estudo sobre Teoria Musical:

-Início: 1 de Dezembro de 2020;

-Fim:20 de Dezembro de 2020;

### Criação de diagramas:

-Início:21 de Dezembro de 2020;

-Fim:1 de Janeiro de 2021;

### Início do desenvolvimento da aplicação:

-Início:7 de Janeiro de 2021;

-Fim:30 de Março de 2021;

### Criação de um base de dados:

-Início:1 de Abril de 2021;

-Fim:13 de Abril de 2021;

### Integração da aplicação e base de dados:

-Início:14 de Abril de 2021;

-Fim:6 de Maio de 2021;

### Entregas:

-1º relatório intermédio: 22 de Janeiro de 2021;

-relatório intercalar de 2º semestre: 23 de Maio de 2021;

-Entrega final: 23 de Julho de 2021;

## Bibliografia

- [1] Pt.wikipedia.org, 2021, Rede neural convolucional, [online] available at:  
[https://pt.wikipedia.org/wiki/Rede\\_neural\\_convolucional](https://pt.wikipedia.org/wiki/Rede_neural_convolucional)
- [2] Pt.wikipedia.org, 2021, Mel-frequency cepstrum, [online] available at:  
[https://en.wikipedia.org/wiki/Mel-frequency\\_cepstrum](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum)
- [3] kaggle.com, 2021, GTZAN dataset, [online], available at:  
<https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification>
- [4] craft.co, 2021, echoNest, [online], available at:  
[https://en.wikipedia.org/wiki/The\\_Echo\\_Nest](https://en.wikipedia.org/wiki/The_Echo_Nest)

## Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso