

## Cenários de teste:

1. Inserir álbum com nome, ano de lançamento e artista/banda.
2. Inserir músicas com nome, duração e possibilidade de favoritar ou não.
3. Pesquisar álbum por nome, ano de lançamento ou artista/banda.
4. Pesquisar música por nome ou artista/banda.
5. Gerar playlist quando a soma da duração das músicas é menor que 1h.
6. Gerar playlist quando a soma da duração das músicas é igual a 1h.
7. Gerar playlist quando a soma da duração das músicas é maior que 1h.
8. Gerar playlist quando todas as músicas são favoritas.
9. Gerar playlist quando não há músicas favoritas.
10. Gerar playlist quando a soma das músicas favoritas é inferior à 30 minutos ou duração da playlist.

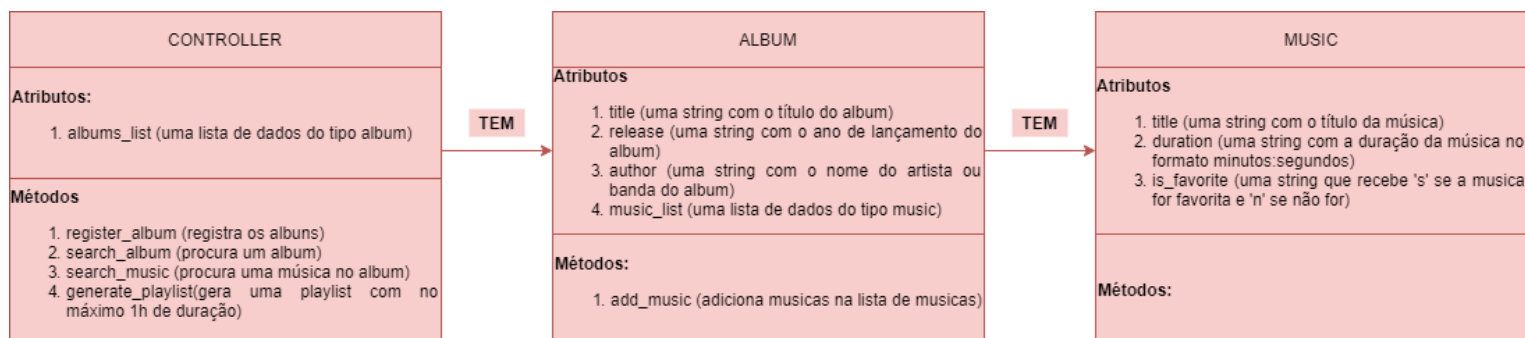
## Premissas assumidas:

- Assumiu-se que não haverão álbuns com o mesmo nome.
- Assumiu-se que o usuário sempre irá inserir a duração da música no formato “minutos:segundos”
- Assumiu-se que o usuário sempre irá inserir um dado com um tipo válido no menu de opções

## Decisões de projeto:

No que diz respeito à linguagem, optei pelo uso da linguagem python por ser uma linguagem com muitas bibliotecas úteis e que permite fazer muitas operações complexas com pouca linha de código. Optei por usar orientação a objetos também, por ser uma maneira melhor e mais eficiente de trabalhar os dados dentro do código, permitindo o reaproveitamento do mesmo e deixando os dados mais parecidos com o mundo real. Além disso, utilizei a arquitetura MVC, para que a conexão entre as camadas (dados, lógica de negócio e interação com usuário) fosse feita de uma forma melhor. Das bibliotecas de python, utilizei apenas a *random* para utilizar a função *sample* com o objetivo de embaralhar as músicas a fim de gerar uma playlist aleatória

## Modelagem:



## Instruções para executar o sistema:

Com o python instalado no computador, abra o prompt de comando, dirija-se até o diretório onde o arquivo “view.py” está localizado e execute “python view.py”