



Guia do Analisador de Bandas e Músicas



Visão Geral do Projeto

Este projeto é uma aplicação web construída com **Streamlit** que permite analisar bandas e músicas através de diferentes APIs. O sistema oferece análise de sentimento, nuvem de palavras, tradução e informações detalhadas sobre artistas e suas músicas.



Estrutura do Projeto

Arquivos Principais:

- `bandas_streamlit_app.py` - Interface do usuário (frontend)
- `bandas_backend.py` - Lógica de negócio e APIs (backend)
- `.env` - Credenciais das APIs



Objetivo Principal

Fornecer uma ferramenta completa para análise musical que combina:

- **Dados do Spotify** (informações de músicas e artistas)
- **Letras de músicas** (via Lyrics.ovh API)
- **Informações enciclopédicas** (Wikipedia)
- **Análise de texto** (sentimento, frequência de palavras, tradução)



Detalhamento dos Arquivos

1. `bandas_streamlit_app.py` (Frontend)

Função: Interface principal da aplicação web

Principais Seções:



Configuração Inicial

- Configuração da página Streamlit
- Importação das funções do backend
- Verificação de dependências



Interface do Usuário

- Seletor de modo: "Analisar Banda" ou "Analisar Música Específica"
- Campos de entrada para dados do usuário

- Layout responsivo com colunas

Modo "Analisar Banda"

- Busca informações gerais da banda
- Exibe resumo da Wikipedia com imagem
- Mostra estimativas de vendas
- Gera nuvem de palavras das letras
- Traduz palavras mais frequentes
- Análise de sentimento das músicas

Modo "Analisar Música Específica"

- Busca letra de uma música específica
 - Exibe informações detalhadas da música
 - Análise focada em uma única música
 - Métricas de popularidade (quando disponível)
-

2. `bandas_backend.py` (**Backend**)

Função: Contém toda a lógica de negócio e conexões com APIs

Funções Principais:

Funções de Busca

```
python
```

```
buscar_musicas(banda, limite=10)
```

- **Objetivo:** Busca músicas de uma banda no Spotify
- **Retorna:** Lista com nome e artista das músicas

```
python
```

```
buscar_musica_especifica(nome_musica, artista=None)
```

- **Objetivo:** Busca informações detalhadas de uma música específica
- **Retorna:** Dados completos da música (nome, artista, álbum, popularidade)

```
python
```

```
buscar_letras(banda, limite=10)
```

- **Objetivo:** Coleta letras de múltiplas músicas de uma banda
- **Retorna:** Lista de letras coletadas

python

```
buscar_letra_musica_especifica(nome_musica, artista=None)
```

- **Objetivo:** Busca a letra de uma música específica
- **Retorna:** Letra e informações da música

🔪 Funções de Processamento de Texto

python

```
limpar_letra(letra)
```

- **Objetivo:** Remove metadados e limpa letras para análise
- **Remove:** Timestamps, créditos, números de linha, etc.

python

```
extrair_palavras_principais(texto_input, limite=15)
```

- **Objetivo:** Identifica palavras mais frequentes no texto
- **Filtra:** Stopwords em português e inglês
- **Retorna:** Lista das palavras mais comuns com frequência

python

```
traduzir_palavras(palavras_freq)
```

- **Objetivo:** Traduz palavras do inglês para português
- **Usa:** Google Translate API
- **Retorna:** Lista com palavra original, tradução e frequência

🌈 Funções de Análise

python

```
gerar_nuvem(texto_input)
```

- **Objetivo:** Cria visualização de nuvem de palavras
- **Biblioteca:** WordCloud + Matplotlib

- **Retorna:** Figura matplotlib pronta para exibição

python

```
analise_sentimento(texto_input)
```

- **Objetivo:** Calcula polaridade emocional do texto
- **Biblioteca:** TextBlob
- **Retorna:** Valor entre -1 (negativo) e +1 (positivo)

python

```
interpretar_sentimento(valor)
```

- **Objetivo:** Converte valor numérico em interpretação textual
- **Retorna:** Emoji + descrição + cor

Funções de Informação

python

```
buscar_dados_wikipedia(banda)
```

- **Objetivo:** Coleta informações enciclopédicas sobre a banda
- **Retorna:** Resumo, imagem e dados de vendas

3. `.env` (Configuração)

Função: Armazena credenciais das APIs de forma segura

Credenciais Incluídas:

- `SPOTIPY_CLIENT_ID` - ID do cliente Spotify
- `SPOTIPY_CLIENT_SECRET` - Chave secreta Spotify
- `GENIUS_ACCESS_TOKEN` - Token do Genius (backup)

APIs e Bibliotecas Utilizadas

APIs Externas:

- **Spotify API** - Dados de músicas e artistas
- **Lyrics.ovh API** - Letras de músicas
- **Wikipedia API** - Informações enciclopédicas

- **Google Translate API** - Tradução de textos

Bibliotecas Python:

- `streamlit` - Interface web
 - `spotipy` - Cliente Spotify
 - `wordcloud` - Nuvem de palavras
 - `textblob` - Análise de sentimento
 - `googletrans` - Tradução
 - `matplotlib` - Visualizações
 - `pandas` - Manipulação de dados
 - `requests` - Requisições HTTP
-



Fluxo de Funcionamento

Para Análise de Banda:

1. **Entrada:** Nome da banda
2. **Busca:** Músicas no Spotify
3. **Coleta:** Letras via Lyrics.ovh
4. **Processamento:** Limpeza e análise de texto
5. **Visualização:** Nuvem de palavras + tradução
6. **Análise:** Sentimento médio das músicas
7. **Complemento:** Informações da Wikipedia

Para Música Específica:

1. **Entrada:** Nome da música + artista
 2. **Busca:** Letra específica
 3. **Processamento:** Análise focada
 4. **Visualização:** Nuvem + palavras traduzidas
 5. **Análise:** Sentimento da música
 6. **Métricas:** Popularidade e informações
-



Funcionalidades Principais



Análise de Sentimento

- Determina se as letras são positivas, negativas ou neutras

- Usa algoritmos de processamento de linguagem natural
- Fornece interpretação em português com emojis

Nuvem de Palavras

- Visualização das palavras mais frequentes
- Remove stopwords automaticamente
- Design colorido e responsivo

Tradução Inteligente

- Traduz palavras do inglês para português
- Mantém frequência das palavras
- Exibe em formato de tabela organizada

Dados Ricos

- Informações do Spotify (popularidade, álbum)
- Resumos da Wikipedia com imagens
- Estimativas de vendas quando disponíveis

Casos de Uso

Para Fãs de Música:

- Descobrir temas recorrentes nas letras de bandas favoritas
- Comparar sentimentos entre diferentes artistas
- Explorar vocabulário usado por músicos

Para Pesquisadores:

- Análise acadêmica de tendências musicais
- Estudos de linguística aplicada à música
- Comparação de estilos entre épocas/gêneros

Para Educadores:

- Ferramenta didática para aulas de inglês
- Análise cultural através da música
- Demonstração de APIs e processamento de texto

Tratamento de Erros

O sistema inclui tratamento robusto de erros:

- **Fallbacks** quando APIs não respondem
 - **Validação** de entrada do usuário
 - **Mensagens claras** de erro
 - **Alternativas** quando recursos não estão disponíveis
-



Métricas e Indicadores

Análise de Sentimento:

- **+1.0:** Muito positivo
- **0.0:** Neutro
- **-1.0:** Muito negativo

Frequência de Palavras:

- Contagem absoluta de ocorrências
- Filtros por relevância
- Ordenação por importância

Popularidade (Spotify):

- Escala de 0-100
- Baseada em streams e engajamento
- Atualizada em tempo real