






# Shorts Generator

**Gerador de Shorts de Lives** - Uma versão simplificada para identificar momentos engraçados usando IA e criar shorts automaticamente.

## Visão Geral

Este projeto é uma versão simplificada do [shorts\\_maker](#) original, focada especificamente em:

-  Processar vídeos de lives ou arquivos locais
-  Extrair e transcrever áudio automaticamente
-  Identificar momentos engraçados usando GPT-5 nano (ou modelos compatíveis)
-  Criar shorts automaticamente com os melhores momentos
-  Executar tudo em Docker para facilitar a instalação

## Funcionalidades

- **Processamento Automático:** Baixa vídeos de URLs ou processa arquivos locais
- **Transcrição Inteligente:** Usa Whisper para transcrever áudio com timestamps precisos
- **IA para Identificação:** Utiliza LLM para identificar momentos engraçados e virais
- **Edição Automática:** Cria shorts individuais e compilações automaticamente
- **Formato Otimizado:** Gera vídeos no formato vertical ideal para redes sociais
- **Docker Ready:** Configuração completa com Docker para fácil instalação

## Instalação Rápida

### Opção 1: Docker (Recomendado)

Bash

```
# 1. Clone o repositório
git clone <repository-url>
cd shorts_generator

# 2. Configure as variáveis de ambiente
cp .env.example .env
# Edite o arquivo .env com sua chave da OpenAI
```

```
# 3. Execute com Docker Compose
docker-compose up -d

# 4. Acesse o container
docker-compose exec shorts-generator bash

# 5. Execute o gerador
python src/main.py "https://youtube.com/watch?v=VIDEO_ID"
```

## Opção 2: Instalação Local

Bash

```
# 1. Clone o repositório
git clone <repository-url>
cd shorts_generator

# 2. Instale as dependências
pip install -r requirements.txt

# 3. Configure FFmpeg (se necessário)
# Windows: Defina FFMPEG_PATH no .env
# Linux/Mac: Instale via package manager

# 4. Configure a API
export OPENAI_API_KEY="sua_chave_aqui"

# 5. Execute
python src/main.py "caminho/para/video.mp4"
```



## Configuração

### Variáveis de Ambiente

Crie um arquivo `.env` baseado no `.env.example` :

Plain Text

```
# Obrigatório
OPENAI_API_KEY=sua_chave_openai_aqui

# Opcional
OPENAI_API_BASE=https://api.openai.com/v1
```

```
FFMPEG_PATH=/caminho/para/ffmpeg # Windows:  
D:\Projetos\Git\Python_Util\ffmpeg\bin\ffmpeg.exe
```

## FFmpeg no Windows

Se você tem o FFmpeg compilado em uma pasta específica (como mencionado:

D:\Projetos\Git\Python\_Util\ffmpeg\bin ), configure:

Plain Text

```
FFMPEG_PATH=D:\Projetos\Git\Python_Util\ffmpeg\bin\ffmpeg.exe
```



## Como Usar

### Comando Básico

Bash

```
python src/main.py "fonte_do_video"
```

### Exemplos de Uso

Bash

```
# Processar vídeo do YouTube  
python src/main.py "https://youtube.com/watch?v=VIDEO_ID"  
  
# Processar arquivo local  
python src/main.py "/caminho/para/video.mp4"  
  
# Criar apenas shorts individuais  
python src/main.py "video.mp4" --no-compilation  
  
# Criar apenas compilação  
python src/main.py "video.mp4" --no-individual  
  
# Manter arquivos temporários  
python src/main.py "video.mp4" --keep-temp  
  
# Diretório de saída personalizado  
python src/main.py "video.mp4" --output-dir "/caminho/saida"
```

## Usando Docker

Bash

```
# Executar com Docker
docker run -v $(pwd)/output:/app/output \
  -e OPENAI_API_KEY="sua_chave" \
  shorts_generator \
  python src/main.py "https://youtube.com/watch?v=VIDEO_ID"

# Ou usando docker-compose
docker-compose run shorts-generator python src/main.py "video.mp4"
```



## Arquitetura

O projeto é dividido em módulos especializados:

### Módulos Principais

- **video\_ingestion.py** : Baixa vídeos de URLs ou processa arquivos locais
- **audio\_processor.py** : Extrai áudio e gera transcrições com Whisper
- **moment\_identifier.py** : Usa LLM para identificar momentos engraçados
- **video\_editor.py** : Cria e edita os shorts finais
- **main.py** : Orquestra todo o processo

### Fluxo de Trabalho

1. **Ingestão** → Baixa/processa o vídeo de entrada
2. **Transcrição** → Extrai áudio e gera texto com timestamps
3. **Análise IA** → Identifica momentos engraçados com LLM
4. **Edição** → Cria shorts individuais e compilações
5. **Saída** → Salva vídeos otimizados para redes sociais



## Exemplos de Saída

O gerador cria automaticamente:

### Shorts Individuais

- **short\_01.mp4** - Melhor momento identificado

- short\_02.mp4 - Segundo melhor momento
- short\_03.mp4 - Terceiro melhor momento
- ...

## Compilação

- compilation\_short.mp4 - Compilação dos melhores momentos

## Arquivos de Análise

- transcription.txt - Transcrição completa com timestamps
- funny\_moments.json - Análise detalhada dos momentos identificados

## Configurações Avançadas

### Personalizar Modelos

Edite config/settings.py para ajustar:

Python

```
# Modelo do Whisper para transcrição
WHISPER_MODEL = "base" # tiny, base, small, medium, large

# Modelo LLM para análise
LLM_MODEL = "gpt-4o-mini" # ou gpt-4, gpt-3.5-turbo

# Duração dos shorts
SHORT_DURATION = 60 # segundos

# Duração mínima de momentos
MIN_MOMENT_DURATION = 10 # segundos
```

## Otimização de Performance

Para vídeos longos:

- Use WHISPER\_MODEL = "tiny" para transcrição mais rápida
- Limite MAX\_VIDEO\_DURATION = 1800 (30 minutos)
- Use CHUNK\_DURATION = 300 para processar em pedaços

## Docker

## Build da Imagem

Bash

```
docker build -t shorts_generator .
```

## Executar Container

Bash

```
# Execução simples
docker run -it --rm \
  -v $(pwd)/output:/app/output \
  -v $(pwd)/videos:/app/videos \
  -e OPENAI_API_KEY="sua_chave" \
  shorts_generator

# Com docker-compose (recomendado)
docker-compose up -d
docker-compose exec shorts-generator bash
```

## Volumes Importantes

- `/app/data` - Arquivos temporários (áudio, transcrições)
- `/app/output` - Shorts gerados
- `/app/videos` - Vídeos de entrada (opcional)

## Troubleshooting

### Problemas Comuns

#### Erro: FFmpeg não encontrado

Bash

```
# Linux/Mac
sudo apt install ffmpeg # Ubuntu/Debian
brew install ffmpeg      # macOS

# Windows
# Baixe do https://ffmpeg.org e configure FFMPEG_PATH
```

## Erro: OpenAI API Key

Bash

```
export OPENAI_API_KEY="sua_chave_aqui"  
# ou configure no arquivo .env
```

## Erro: Memória insuficiente

Bash

```
# Reduza a qualidade do Whisper  
WHISPER_MODEL = "tiny"  
  
# Ou processe vídeos menores  
MAX_VIDEO_DURATION = 1800
```

## Erro: Dependências Python

Bash

```
pip install --upgrade pip  
pip install -r requirements.txt
```

## Logs e Debug

Para debug detalhado:

Bash

```
python src/main.py "video.mp4" --keep-temp  
# Manterá arquivos temporários para análise
```



## Performance

### Tempos Estimados

Para um vídeo de 1 hora:

- **Transcrição:** 5-15 minutos (dependendo do modelo Whisper)
- **Análise IA:** 1-3 minutos (dependendo do modelo LLM)
- **Edição:** 2-5 minutos (dependendo do número de shorts)

## Requisitos de Sistema

### Mínimo:

- 4GB RAM
- 2GB espaço livre
- Python 3.11+

### Recomendado:

- 8GB+ RAM
- 10GB+ espaço livre
- GPU (para Whisper acelerado)



## Contribuição

Este projeto é uma versão simplificada focada em facilidade de uso. Para contribuir:

1. Fork o repositório
2. Crie uma branch para sua feature
3. Faça commit das mudanças
4. Abra um Pull Request



## Licença

Este projeto é baseado no [shorts\\_maker](#) original e mantém a mesma licença.



## Créditos

- Baseado no projeto [shorts\\_maker](#) por @guinalcantara
- Usa [Whisper](#) para transcrição
- Usa [MoviePy](#) para edição de vídeo
- Usa [yt-dlp](#) para download de vídeos



## Suporte

Para dúvidas ou problemas:

1. Verifique a seção [Troubleshooting](#)
2. Consulte os logs de erro



3. Abra uma issue no repositório

---

 **Divirta-se criando shorts incríveis com IA! 🚀**