

Embarcadero Conference 2024

Inovação faz parte do nosso DNA!

Interação por voz com seu ERP: IA mudando a experiência do seu software

Daniel Fernandes



Assuntos

- O que são as LLMS
- O que é o Whisper
- Sobre o GPT
- BASS
- Libs Necessária
- Explicando o código
- Na prática

Sobre mim

- Daniel Fernandes.
- Desenvolvedor de Software Sênior na AquaSoft.
- Mais de 9 anos de experiência em Delphi.
- Graduado em Gestão da Tecnologia da Informação (FATEC Bragança Paulista).
- Pós-graduado em Engenharia de Software (Estácio).
- Criador do @DinosDev no Instagram.
- Faço parte da comunidade Delphi na Veia.
- Instrutor dos cursos oficiais Embarcadero.
- Desenvolvedor do componente DinosOffices.
- Palestrante.

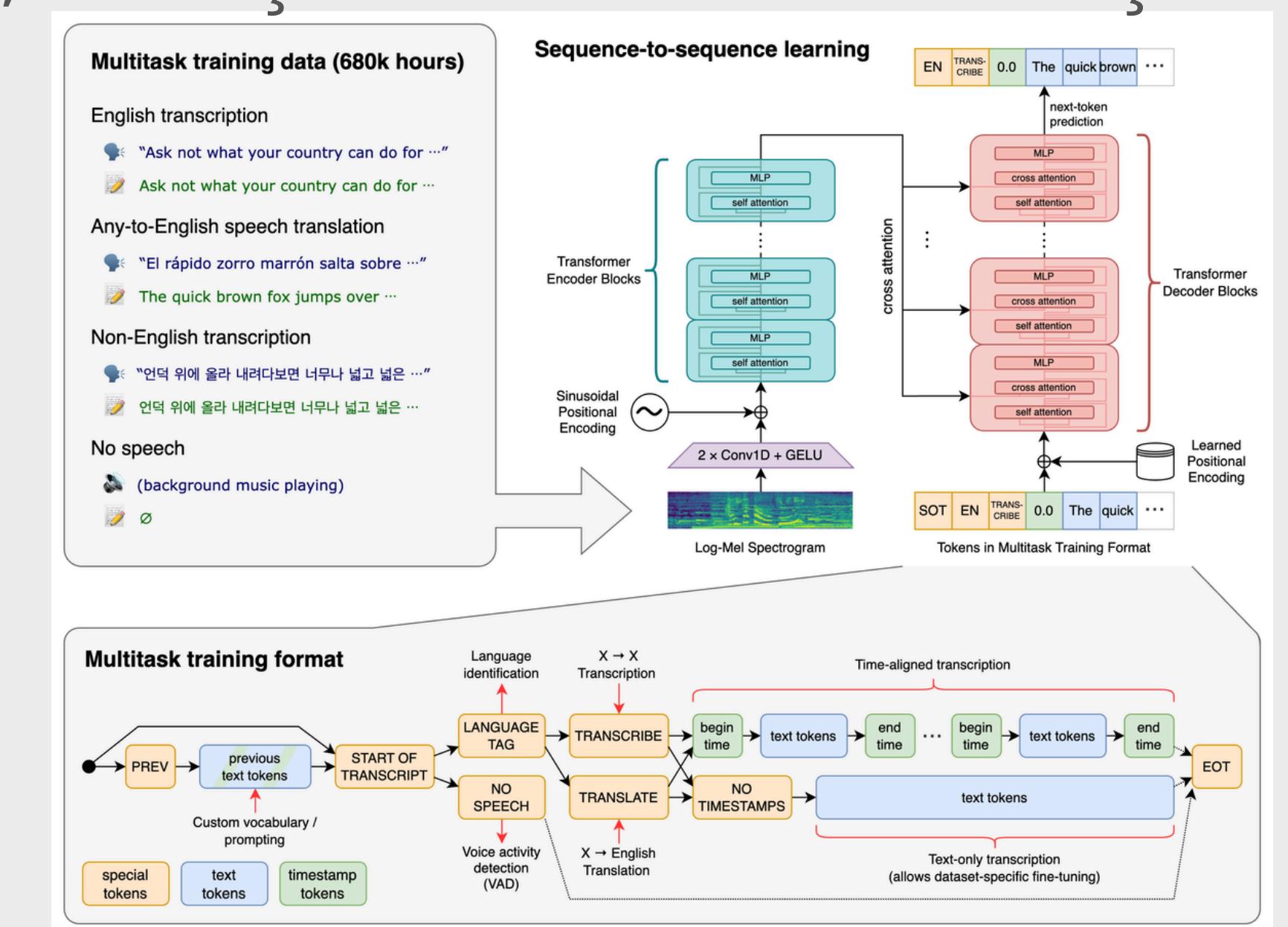
O que são as LLMS

Os Modelos de Linguagem de Grande Escala (Large Language Models ou LLMs) são um tipo de modelo de **inteligência Artificial** criado para entender e gerar texto.



O que é o Whisper

O Whisper é um modelo de reconhecimento de fala de uso geral. Ele é treinado em um grande conjunto de dados de áudio diversificado e também é um modelo multitarefa que pode realizar reconhecimento de fala multilíngue, tradução de fala e identificação de idioma.



Sobre o GPT

Chat GPT teve seu desenvolvimento pautado em redes neurais e machine learning, tendo sido criado com foco em diálogos virtuais. A ideia é que ele pudesse aprimorar a experiência e os recursos oferecidos por assistentes virtuais, como Alexa ou Google Assistente.



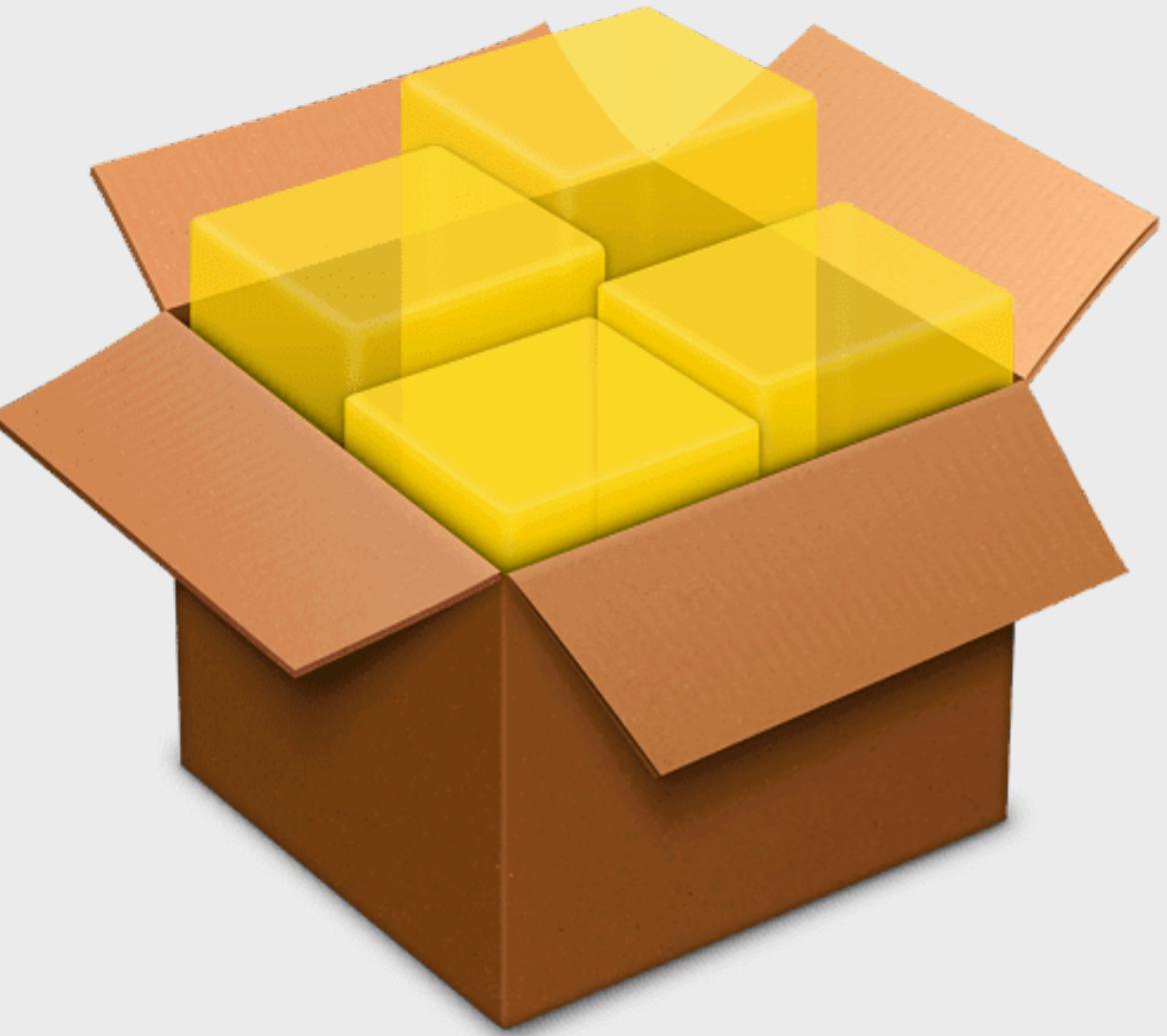
Sobre o BASS

BASS é uma biblioteca de áudio para uso em software em várias plataformas. Seu propósito é fornecer aos desenvolvedores samples poderosos e eficientes, stream (MP3, MP2, MP1, OGG, WAV, AIFF, gerados de forma personalizada e mais por meio de codecs e complementos do S.O)



Libs Necessária

- Git
- OpenAI - Whisper
- Conda
- PyTorch
- Chocolaty
- ffmpeg
- DelphiOpenAI

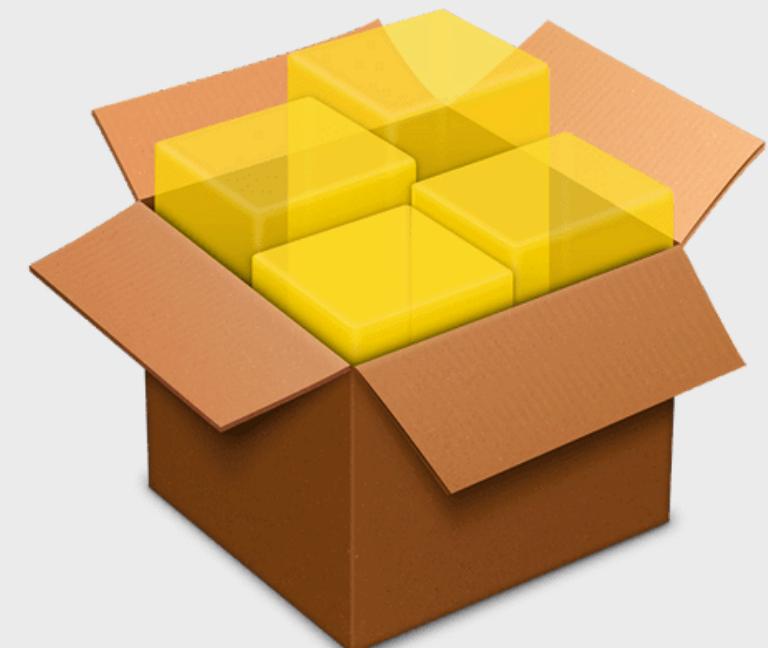


Libs Necessária - instalação

- Git - <https://git-scm.com/downloads>
- OpenAI - Whisper - <https://github.com/openai/whisper?tab=readme-ov-file>
 - .Antes de instalar o Whisper
 - .Instalar o conda <https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html>
 - Anaconda Distribution
 - .PyTorch <https://pytorch.org/get-started/previous-versions/>

Linux and Windows

```
# CUDA 11.8
conda install pytorch==2.3.0 torchvision==0.18.0 torchaudio==2.3.0 pytorch-cuda=11.8 -c pytorch -c nvidia
# CUDA 12.1
conda install pytorch==2.3.0 torchvision==0.18.0 torchaudio==2.3.0 pytorch-cuda=12.1 -c pytorch -c nvidia
# CPU Only
conda install pytorch==2.3.0 torchvision==0.18.0 torchaudio==2.3.0 cpuonly -c pytorch
```



Libs Necessária - instalação

Chocolatey <https://chocolatey.org/install>

With PowerShell, you must ensure `Get-ExecutionPolicy` is not Restricted. We suggest using `Bypass` to bypass the policy to get things installed or `AllSigned` for quite a bit more security.

- Run `Get-ExecutionPolicy`. If it returns `Restricted`, then run `Set-ExecutionPolicy AllSigned` or `Set-ExecutionPolicy Bypass -Scope Process`.

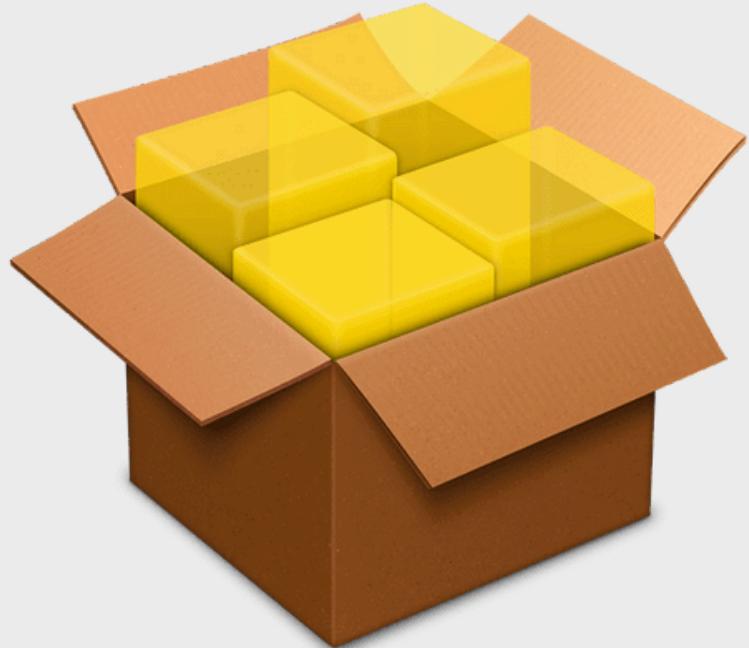
Now run the following command:

```
> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::Sec
```

3. Paste the copied text into your shell and press Enter.

4. Wait a few seconds for the command to complete.

5. If you don't see any errors, you are ready to use Chocolatey! Type `choco` or `choco -?` now, or see Getting Started for usage instructions.



Libs Necessária - instalação

ffmpeg

Usamos o chocolatey para instalar

- No Powershell, rode o comando:

choco install ffmpeg

Whisper

No **Powershell**,

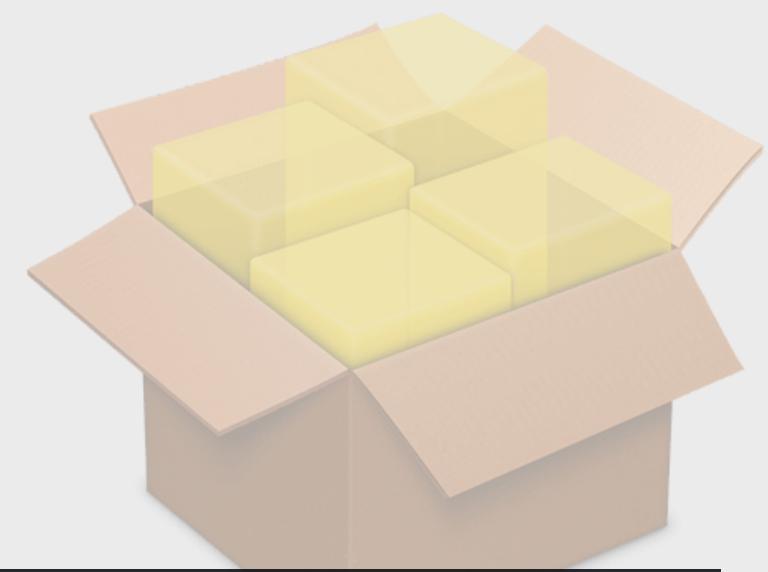
- Para instalar: **pip install -U openai-whisper**
- Para puxar as atualizações:

pip install --upgrade --no-deps --force-reinstall git+https://github.com/openai/whisper.git

BASS

Apos download, incluir o fonte bass.pas no projeto e a dll bass.dll junto com o executavel

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39 M	tiny.en	tiny	~1 GB	~32x
base	74 M	base.en	base	~1 GB	~16x
small	244 M	small.en	small	~2 GB	~6x
medium	769 M	medium.en	medium	~5 GB	~2x
large	1550 M	N/A	large	~10 GB	1x

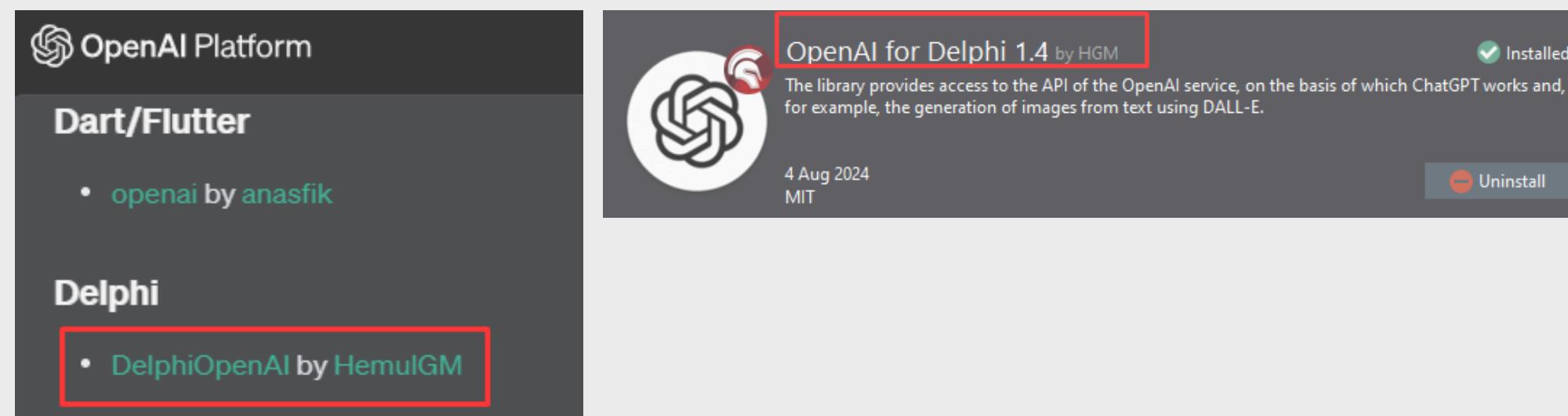


Libs Necessária - instalação

Delphi OpenAI

Usar a lib para Delphi, disponível: <https://platform.openai.com/docs/libraries/community-libraries>

O Link a baixo, levara ao GIT, você pode pegar e instalar manualmente ou pelo GetIt,
recomendo o GetIt se tiver suporte para seu Delphi.

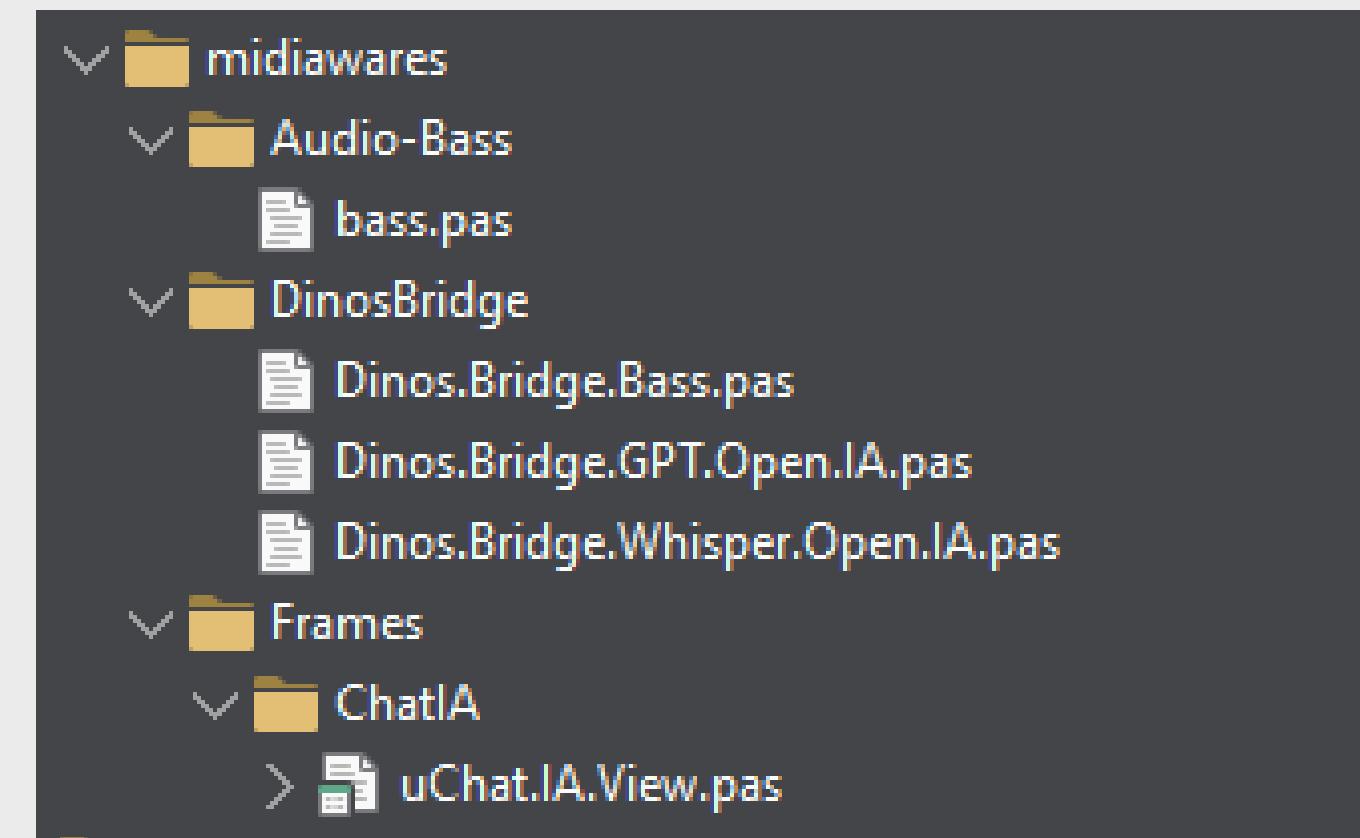


Gerar a APIKEY

<https://platform.openai.com/api-keys>

Libs Necessária - instalação

Adicionar os middleware ao projeto



Adione a API_KEY gerada no slide anterior no fonts 'Dinos.Bridge.GPT.Open.IA'



```
Dinos.Bridge.GPT.Open.IA ×
Search for a type Search for a method
end;
TDinosChatGPT = class(TOpenAI)
private
  Const
    API_KEY  = 'xxxxx';
    BASE_URL = 'https://api.openai.com/v1';
    CONNECTION_TIME_OUT = 60000;
var
  FModelsAvailability: TStringList;
  FVoice: TypeGPTVoice;
public
  function TextToSpeech(const AText: String): TMemoryStream;
```

Vamos ao código

Fluxo do projeto



O Aplicativo





Embarcadero Conference 2024

Inovação faz parte do nosso DNA!



Quer me ver na
#ECON25?
Acesse o QRCode
e avalie minha palestra!



Daniel Fernandes

@Dinosdev

<https://www.linkedin.com/in/daniel-fernandes-124393b0/>

<https://github.com/Daniel09Fernandes/DinosLib-IA-Integration>

