

IA Conversacional en Ubuntu

Implementación Local con Vocode.

Presiona Espacio →



UbuCon
Latinoamérica 9na Edición

AGOSTO 30 y 31-2024 BARRANQUILLA, COL.



Sebastian Rojo



Open Source Engineer at Vocode



Fundador y CTO de ArtisanLabs



Comunicación en Tiempo Real, Voice ML,
ML locales y distribuidos.



Contribuidor Activo en la Comunidad
Python y de Código Abierto



linkedin.com/in/arpagon



[@arpagon](https://twitter.com/@arpagon)

- ① ¿Quién está interesado en escuchar sobre IA conversacional?
- ① Dejen la mano levantada los que usan ChatGPT...
pero copian y pegan sin revisar... 😂⚡
- ① ¿Quién ha usado asistentes de voz como Siri, Alexa o Google Assistant?
- ① ¿Quién ha usado IAs conversacionales?
- ① ¿Quién ha implementado algo con inteligencia artificial?

Por qué es esto importante

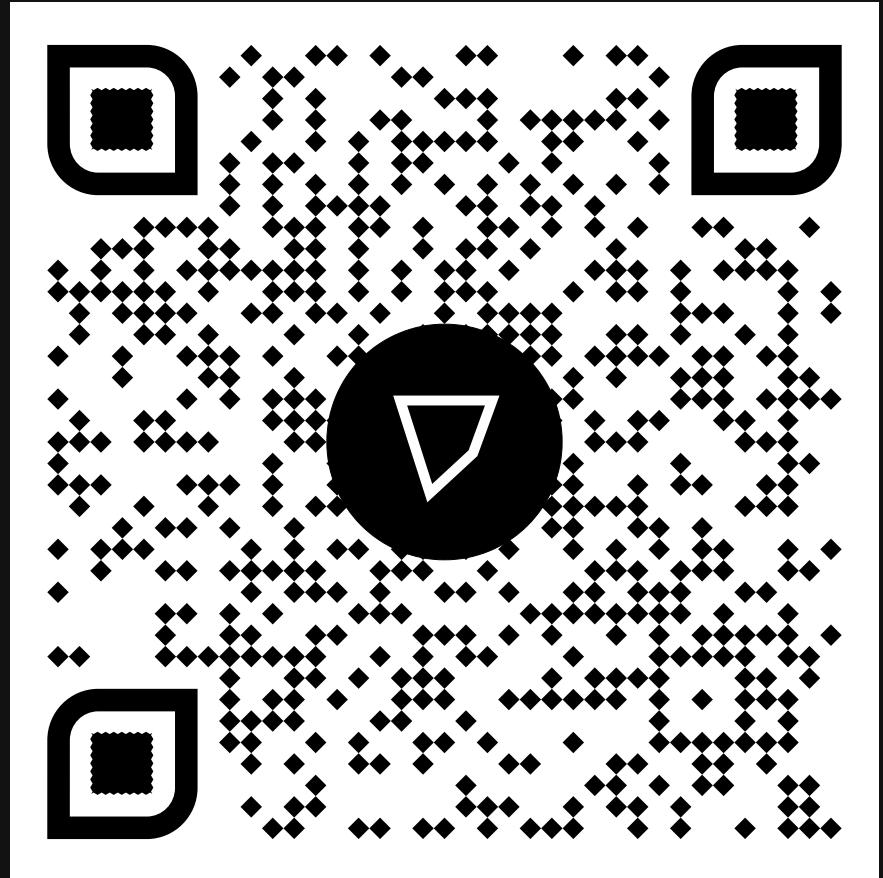
⑤ ¿Cuántos trabajadores "deskless" hay en el mundo?

¡Es más del 80%! Hay más de 3.5 mil millones de trabajadores **deskless** en todo el mundo.
Sin embargo, solo el 1% del software está diseñado específicamente para ellos.



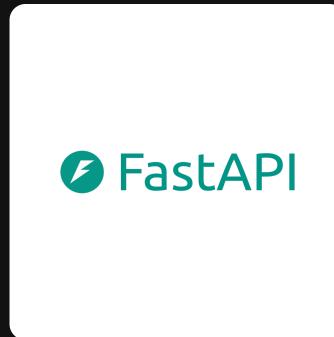
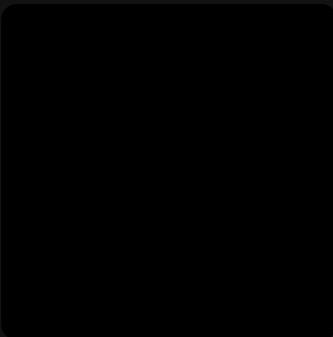
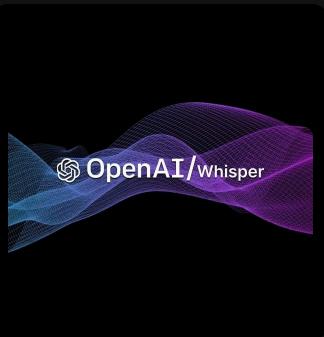
¿Qué es Vocode?

- ⌚ Es una librería modular y open source que facilita la creación de aplicaciones de voz basadas en modelos de lenguaje (LLM).
- 💡 Respaldado por Y Combinator, Google Ventures y otros inversores de renombre.
- 🌐 <https://github.com/vocodedev/vocode-core>



Demo

Componentes



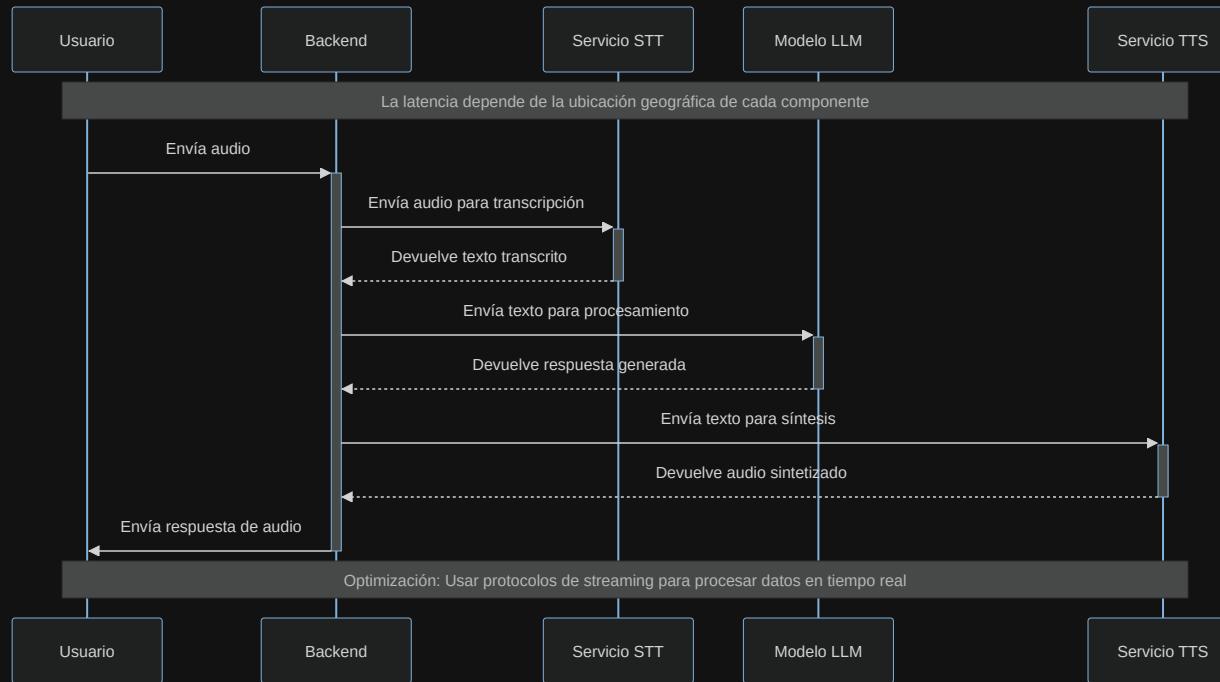
Speech to Text (STT)
Voz a Texto

Large Language Model
(LLM)

Text to Speech (TTS)
Síntesis de Voz por Texto

Backend de Orquestacion

Arquitectura: Orquestación multi-modelo



Code

```
async def main():
(
    microphone_input,
    speaker_output,
) = create_streaming_microphone_input_and_speaker_output(
    use_default_devices=False,
)
# config STT, LLM, TTS
await conversation.start()
print("Conversation started, press Ctrl+C to end")
signal.signal(signal.SIGINT, lambda _0, _1: asyncio.create_task(conversation.terminate()))
while conversation.is_active():
    chunk = await microphone_input.get_audio()
    conversation.receive_audio(chunk)
```

Configure STT

```
conversation = StreamingConversation(  
    output_device=speaker_output,  
    transcriber=DeepgramTranscriber(  
        DeepgramTranscriberConfig.from_input_device(  
            microphone_input,  
            endpointing_config=PunctuationEndpointingConfig(),  
            api_key=settings.deepgram_api_key,  
        ),  
    ),
```

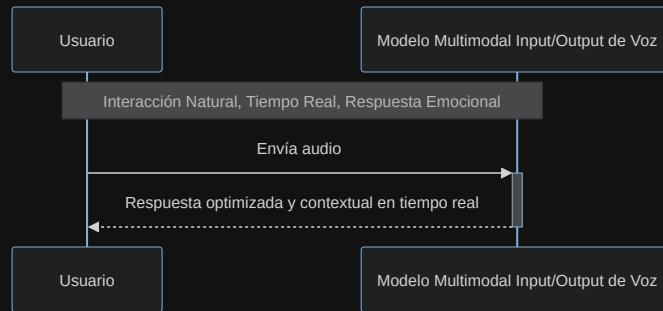
Configure LLM

```
agent=ChatGPTAgent(  
    ChatGPTAgentConfig(  
        openai_api_key=settings.openai_api_key,  
        initial_message=BaseMessage(text="What up"),  
        prompt_preamble="""The AI is having a pleasant conversation about life""",  
    )  
,
```

Configure TTS

```
synthesizer=AzureSynthesizer(  
    AzureSynthesizerConfig.from_output_device(speaker_output),  
    azure_speech_key=settings.azure_speech_key,  
    azure_speech_region=settings.azure_speech_region,  
,  
)
```

Arquitectura: multimodal model (e2e)



Learn More

[Documentación](#) · [GitHub](#) · [Ejemplos](#)

Powered by  Sliddev