

Documentación Técnica: Proyecto Voicebot

1. Arquitectura del Sistema

El proyecto está basado en una arquitectura cliente-servidor con comunicación bidireccional en tiempo real mediante WebSockets. La aplicación está desarrollada con FastAPI en el backend y JavaScript puro en el frontend.

Estructura del Proyecto

```
voicebot/
├── app/
│   ├── api/
│   │   └── chat_endpoint.py    # Endpoint WebSocket para la comunicación
│   ├── config/
│   │   ├── global_config.py   # Configuración global
│   │   └── websocket_config.py # Configuración de WebSockets
│   └── tools/
│       ├── chatgpt.py         # Integración con ChatGPT
│       └── transcription.py   # Servicio de transcripción de audio
├── templates/
│   └── index.html             # Interfaz de usuario
├── .env                       # Variables de entorno
├── main.py                    # Punto de entrada de la aplicación
├── requirements.txt           # Dependencias
└── run.sh                     # Script de ejecución
```

2. Componentes Principales

Backend

1. **ConnectionManager** (websocket_config.py):

- Gestiona las conexiones WebSocket activas
- Mantiene el historial de conversaciones para cada conexión
- Proporciona métodos para enviar mensajes y reiniciar conversaciones

2. **ChatGPService** (chatgpt.py):

- Gestiona la comunicación con la API de OpenAI
- Mantiene el contexto de las conversaciones
- Procesa los mensajes y devuelve las respuestas

3. **Transcription Service** (transcription.py):

- Convierte el audio a texto utilizando la API Whisper de OpenAI

- Gestiona los errores de transcripción
- Devuelve el resultado al cliente

4. **Chat Endpoint** (chat_endpoint.py):

- Maneja la comunicación WebSocket
- Coordina el flujo de datos entre los componentes
- Gestiona errores y desconexiones

Frontend

1. **Interfaz de Usuario** (index.html):

- Proporciona controles para grabación de audio y gestión de la conexión
- Muestra la conversación y los estados del sistema
- Gestiona la captura de audio del micrófono

3. Flujo de la Petición

Establecimiento de la Conexión

1. El usuario accede a la aplicación web
2. El cliente JavaScript crea una conexión WebSocket al servidor
3. El servidor acepta la conexión y genera un ID único para la conversación

Captura y Procesamiento de Audio

1. El usuario inicia la grabación presionando el botón "Iniciar Grabación"
2. El navegador captura el audio del micrófono utilizando la API MediaRecorder
3. Al detener la grabación, el audio se convierte en un blob
4. El blob se envía al servidor a través de la conexión WebSocket

Procesamiento en el Servidor

1. El servidor recibe los bytes de audio
2. Envía un mensaje de estado "processing" al cliente
3. El audio se envía a la API de OpenAI Whisper para transcripción
4. El texto transcrito se envía de vuelta al cliente con un mensaje "transcribed"
5. El texto se envía a la API de ChatGPT para obtener una respuesta
6. La respuesta se envía al cliente con un mensaje "completed"

Presentación al Usuario

1. El cliente recibe los mensajes de estado y actualiza la interfaz

2. La transcripción del audio se muestra como un mensaje del usuario
3. La respuesta de ChatGPT se muestra como un mensaje del asistente
4. El sistema vuelve al estado "listo" para una nueva interacción

4. Tecnologías Utilizadas

- **Backend:**
 - FastAPI: Framework web de alto rendimiento
 - WebSockets: Para comunicación bidireccional en tiempo real
 - OpenAI API: Para transcripción (Whisper) y generación de texto (ChatGPT)
 - Python: Lenguaje de programación principal
- **Frontend:**
 - HTML/CSS/JavaScript: Para la interfaz de usuario
 - WebSocket API: Para la comunicación con el servidor
 - MediaRecorder API: Para captura de audio en el navegador