1. Introducción

Este documento describe un **asistente de voz inteligente** que funciona en **Edge AI** sin necesidad de conexión a internet. Se combinan varias tecnologías de procesamiento de voz y lenguaje natural para crear un sistema eficiente y preciso.

2. Objetivo

Diseñar un **asistente de voz offline** que pueda:
✅ Escuchar y transcribir preguntas habladas.
✅ Comprender el significado profundo de las palabras.
✅ Buscar información en bases de datos de documentos.
✅ Generar respuestas precisas y verificadas.
✅ Convertir las respuestas en voz.
✅ Funcionar en dispositivos de Edge AI sin depender de la nube.

---

3. Tecnologías Utilizadas

Componente	Tecnología utilizada	Función principal
ASR (voz a texto)	Whisper.cpp (GGML)	Convierte voz en texto
Embeddings	ELMo	Captura significado del texto
Búsqueda (RAG)	FAISS / ChromaDB	Recupera información relevante
Generación de respuesta	LLaMA (Llama.cpp, GGML)	Redacta respuestas precisas
Síntesis de voz (TTS)	Coqui-AI / Piper TTS	Convierte texto en voz
Edge AI hardware	Raspberry Pi, Jetson Nano, Intel NUC	Corre todo sin internet

---

4. Flujo del Sistema

1 Entrada de voz (Whisper.cpp)

- **Tecnología:** Whisper.cpp es un modelo de reconocimiento automático de voz (ASR) optimizado para CPU y compatible con GGML, lo que permite transcribir audio a texto con alta precisión en dispositivos de Edge AI.
- **Ejemplo:** El usuario habla: "¿Cómo reseteo un motor Siemens modelo X200?"
- **Proceso:** Whisper.cpp convierte la voz en texto sin necesidad de internet.

2 Procesamiento del lenguaje (ELMo + embeddings)

- **Tecnología:** ELMo (Embeddings from Language Models) genera representaciones de palabras basadas en contexto mediante redes neuronales bidireccionales (BiLSTM).
- **Ejemplo:** "Reseteo" se asocia a términos como "reinicio" o "reconfiguración", mejorando la comprensión de la consulta.

3 Búsqueda de información (RAG + FAISS/ChromaDB)

- **Tecnología:** Se usa un sistema de recuperación de información basado en FAISS o ChromaDB, donde se indexan manuales y documentos técnicos como embeddings vectoriales.
- **Ejemplo:** La búsqueda devuelve un fragmento relevante del manual de Siemens X200 con las instrucciones precisas.

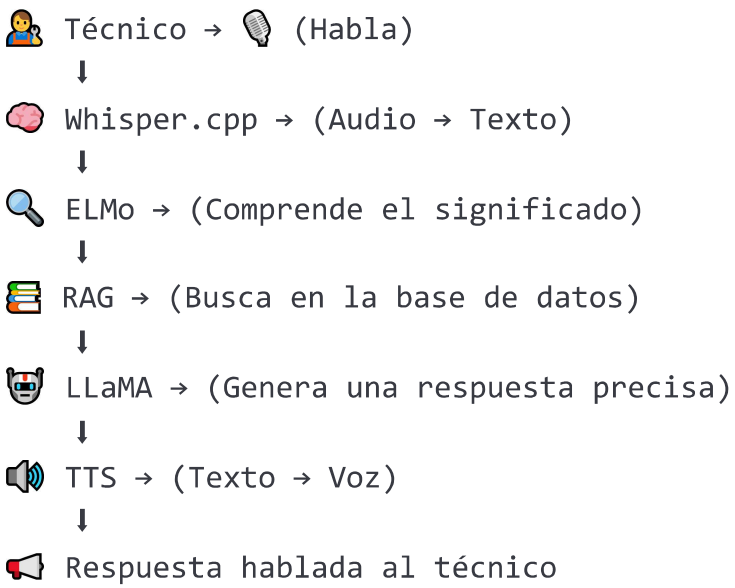
4 Generación de respuesta (LLaMA con contexto del RAG)

- **Tecnología:** LLaMA, un modelo de lenguaje ligero y eficiente, genera respuestas basadas en la información recuperada por RAG, minimizando alucinaciones.
- **Ejemplo:** "Para resetear el modelo X200, mantén presionado el botón Reset durante 5 segundos."

5 Salida en voz (Text-to-Speech - TTS)

- **Tecnología:** Se emplea Coqui-AI o Piper TTS, modelos de síntesis de voz ligeros y optimizados para Edge AI.
- **Ejemplo:** El texto generado por LLaMA se convierte en una respuesta hablada.

5. Arquitectura del Sistema



---

6. Beneficios del Sistema en Edge AI

✅ **Funciona sin internet** → Ideal para fábricas y entornos industriales.
✅ **Responde rápido** → No depende de servidores en la nube.
✅ **Reduce alucinaciones** → Usa RAG para obtener información precisa.
✅ **Optimizado para CPU** → Gracias a GGML, puede ejecutarse en hardware limitado.

---

7. Conclusión

Este asistente de voz permite a técnicos e ingenieros acceder a información confiable sin depender de la nube, utilizando tecnologías optimizadas para dispositivos de **Edge AI**.

🚀 **¿Siguiente paso?** Implementar este sistema en un hardware específico y ajustar la base de datos según las necesidades del usuario.