

ERS

Fepro Voice to Ticket

Date: March 19th, 2026

Contents

Definición del proyecto	3
Título del reto.....	3
Descripción del problema	3
Objetivo general	3
Diseñar e implementar una solución automatizada basada en Inteligencia Artificial que permita la gestión multilingüe de incidentes, facilitando la extracción de información clave a partir de reportes en idiomas como alemán, inglés o español para su registro sistemático en plataformas como Otobo.	3
Requerimientos funcionales	3
Requerimientos no funcionales	4
Restricciones y entorno técnico	5
Tecnologías.....	5
Entregables técnicos	5
Cronograma de Desarrollo	6
Calendario de mentoría y seguimiento	7
Sincronización técnica	7
Checkpoints y entregas.....	7
Canal de comunicación	7
Criterios de evaluación	7

Definición del proyecto

Título del reto

Voice to ticket

Descripción del problema

Las empresas dedicadas a la gestión de aplicaciones se enfrentan al reto constante de recibir incidentes reportados en una gran variedad de idiomas, lo que añade una capa de complejidad técnica y logística a su operación diaria. Para mantener la continuidad del servicio, estos reportes deben ser procesados mediante un flujo de trabajo rápido y eficiente que elimine las barreras lingüísticas de forma inmediata. Solo a través de una gestión ágil de estos datos es posible que el equipo de resolución reciba la información precisa y sin demoras, permitiéndoles intervenir a tiempo para mitigar riesgos y cumplir con los niveles de servicio acordados.

Objetivo general

Diseñar e implementar una solución automatizada basada en Inteligencia Artificial que permita la gestión multilingüe de incidentes, facilitando la extracción de información clave a partir de reportes en idiomas como alemán, inglés o español para su registro sistemático en plataformas **como Otobo**.

Requerimientos funcionales

- **RF1: Recepción Multilingüe:** El sistema debe ser capaz de recibir y procesar texto en al menos tres idiomas iniciales: alemán, inglés y español.
- **RF2: Identificación de Entidades (IA):** La IA debe extraer datos clave del mensaje (nombre del usuario, descripción del problema, urgencia y tipo de servicio afectado).

- **RF3: Traducción Automática:** El sistema debe traducir la descripción del incidente al idioma predeterminado de los técnicos (e.g. español) sin perder el contexto técnico.
- **RF4: Integración con APIs:** La solución debe enviar automáticamente los datos extraídos a la API de la herramienta de gestión de incidentes para crear el ticket de soporte.

Requerimientos no funcionales

- **RNF1: Tiempo de Respuesta:** El procesamiento de la IA y la creación del ticket no debe exceder los 1 minuto tras la recepción del incidente.
- **RNF2: Precisión de la IA:** El modelo de extracción de información debe tener un nivel de confianza (accuracy) mínimo del 90% para evitar registros erróneos.
- **RNF3: Seguridad y Privacidad:** Todos los datos procesados deben estar encriptados (TLS/SSL) durante el tránsito.
- **RNF4: Escalabilidad:** La arquitectura debe soportar un incremento en el volumen de tickets (e.g. de 10 a 500 diarios) sin degradar el rendimiento.
- **RNF5: Mantenibilidad:** El código debe estar documentado y modularizado para permitir la adición de nuevos idiomas o sistemas de ticketing en el futuro.
- **RNF6: Acuse de recibo:** La IA debe confirmar al usuario la información que obtuvo del ticket al mostrarle un mensaje con todo lo recolectado al final de cada conversación.
- **RNF7: Confirmación de datos:** La IA solicitará datos de los que no tenga más de 90% de precisión al usuario.

Restricciones y entorno técnico

Tecnologías

AWS, C#, JavaScript, Python (con flask), Agentic AI y herramientas colaborativas como Git.

Entregables técnicos

Documentación de Arquitectura y Diseño

Diagrama que detalle la conexión entre el canal de entrada (correo, portal, chat), el motor de IA (LLM o NLP) y las APIs de los sistemas de gestión de incidentes.

Diccionario de datos y mapeo de campos

Documento que defina cómo se traducen los datos extraídos por la IA (e.g. "Problem") a los campos específicos del sistema de gestión de incidentes (e.g. Ticket::Subject o Ticket::Body).

Motor de Inteligencia Artificial

Prompts Estructurados: El conjunto de instrucciones optimizadas para que la IA extraiga la información en formato JSON limpio, sin alucinaciones.

API de Integración: El código puente que recibe el texto del cliente, llama a la IA y formatea la respuesta para el sistema de gestión de incidentes.

Integración con apis de herramientas de gestión

Script de Consumo de API (Web Services): Configuración del proveedor de servicios web del sistema de gestión de incidentes para permitir la creación de tickets mediante REST/JSON.

Pruebas y Calidad

Minutas de las sesiones de revisión de calidad.

Matriz de Pruebas de Idiomas: Reporte de resultados que demuestre que el sistema funciona correctamente en alemán, inglés y español.

Informe de Precisión (Accuracy Report): Un documento que valide que la IA extrae la información clave con el margen de error aceptado en los requerimientos.

[Manuales y Código Fuente](#)

Repositorio de Gitlab con el código completo.

README: Instrucciones para instalar la solución, configurar las llaves de API y levantar el servicio. Descripción del proyecto. Estructura de las carpetas. Etc.

Manual de usuario/administrador: Guía rápida sobre cómo monitorear los tickets creados por la IA y qué hacer si un idioma no es reconocido.

Cronograma de Desarrollo

Fase	Actividad
I. Lanzamiento	Publicación de la ERS y selección de equipos por la empresa.
II. Análisis	Sesión de dudas (Kick-off) y definición de la arquitectura.
III. Desarrollo	Implementación del núcleo de la solución.
IV. Refinamiento	Pruebas y correcciones.
V. Entrega	Despliegue y documentación.

Calendario de mentoría y seguimiento

Sincronización técnica

- Reunión de Kick-off en la semana 1
- Juntas presenciales cada dos semanas para reforzar las mentorías.
- Juntas semanales para la revisión de avances.
- Sesiones de calidad al terminar cada componente de la aplicación

Checkpoints y entregas

A definir según el backlog que propongan los equipos.

Canal de comunicación

- Correo electrónico.
- Microsoft Teams (preferido).

Criterios de evaluación

Criterio	Descripción Refinada	Indicador de Éxito (KPI)
Robustez y Escalabilidad	Capacidad de la arquitectura para manejar de 10 a 100 tickets y gestión de errores.	Soporta concurrencia sin degradación de velocidad.

Criterio	Descripción Refinada	Indicador de Éxito (KPI)
		Levanta errores communes en logs.
Compleitud de la Solución	Grado de cumplimiento de los 4 requerimientos funcionales.	100% de los RF implementados y funcionales.
Calidad de la Ingeniería	Uso de Programación Orientada a Objetos (POO) , patrones de diseño, código limpio (SOLID) y modularidad para facilitar cambios de idiomas o APIs.	Código modular, DRY (Don't Repeat Yourself) y legible.
Precisión de la IA (Accuracy)	Porcentaje de acierto en la extracción de entidades clave en los 3 idiomas solicitados.	Confianza mínima del 90% (según RNF2).
Calidad de Integración y UX	Facilidad de uso de la solución y correcta comunicación con las APIs. Formato de ticket.	Integración fluida vía TLS/SSL sin pérdida de datos.
Documentación y Git	Calidad del README.md, diagramas de arquitectura, comentarios en código y un historial de commits organizado.	Documentación técnica completa y clara.

Criterio	Descripción Refinada	Indicador de Éxito (KPI)
Velocidad de Respuesta	Tiempo transcurrido desde la recepción del audio/texto hasta la creación del ticket en la plataforma.	Menor a 60 segundos (según RNF1).