

Tabla de Contenido

1. Descripción General del Proyecto

2. Tecnologías Utilizadas

3. Arquitectura del Sistema

4. Sistema de Autenticación (Auth0)

5. Base de Datos (Supabase)

6. Backend — Netlify Functions

7. Cómo Funciona la IA (Moffy)

8. Manual de Uso

1. Descripción General del Proyecto

KnowledgeTI es una aplicación web desarrollada para centralizar, organizar y consultar soluciones técnicas de TI de forma rápida, segura y eficiente. El proyecto nació de la necesidad de tener un repositorio único donde el equipo de soporte técnico pueda encontrar respuestas a problemas frecuentes sin depender de documentación dispersa.

La aplicación cuenta con un asistente de inteligencia artificial llamado **Moffy**, que permite a los usuarios hacer preguntas en lenguaje natural y recibir respuestas basadas directamente en la base de conocimiento de la empresa. Esto elimina la necesidad de buscar manualmente entre registros y acelera significativamente la resolución de problemas.

Objetivos principales:

- Centralizar el conocimiento técnico del equipo de TI en un solo lugar.
- Permitir búsqueda rápida de soluciones mediante IA conversacional.
- Mantener la seguridad de la información con autenticación obligatoria.
- Permitir el crecimiento controlado de la base de conocimiento mediante sugerencias.
- Ser una herramienta de consulta rápida para el equipo de soporte.

■ **En términos simples: es como una Wikipedia interna de TI, pero con un asistente inteligente que te responde directamente en vez de tener que leer artículos completos.**

2. Tecnologías Utilizadas

El proyecto fue construido usando tecnologías modernas, cada una elegida por su confiabilidad, costo (todas gratuitas o de bajo costo) y facilidad de despliegue.

Tecnología	Rol en el proyecto	Por qué se usó
React	Frontend / Interfaz de usuario	Librería más popular para apps web. Permite crear interfaces dinámicas y reactivas.
Supabase	Base de datos (PostgreSQL)	Base de datos gratuita en la nube. Fácil de usar, con panel visual y API automática.
Auth0	Autenticación de usuarios	Servicio especializado en login seguro. Maneja toda la seguridad de acceso.
Groq + LLaMA	Inteligencia Artificial (Moffy)	API de IA gratuita y muy rápida. Procesa las preguntas y genera respuestas instantáneas.
Netlify	Hosting + Backend (Functions)	
Vite	Herramienta de desarrollo	Compilador moderno que hace el desarrollo más rápido.

3. Arquitectura del Sistema

La arquitectura del proyecto sigue un patrón moderno de tres capas: frontend, backend serverless y base de datos en la nube. Esto significa que no hay un servidor tradicional; en su lugar, se usan funciones pequeñas que se ejecutan solo cuando se necesitan.

Diagrama de arquitectura:

CAPA	COMPONENTE	FUNCIÓN
Frontend	React + Vite (Netlify CDN)	Interfaz que ve el usuario. Cards, chat, filtros, formularios.
Autenticación	Auth0	Verifica quién es el usuario antes de dar acceso.
Backend	Netlify Functions (Node.js)	Funciones de servidor que manejan la lógica segura: consultar Supabase, llama
Base de Datos	Supabase (PostgreSQL)	Guarda todos los registros: knowledge, sugerencias.
Inteligencia Artificial	Groq API (LLaMA 3.1)	Procesa el contexto de la BD y genera respuestas en lenguaje natural.

Flujo general de una solicitud:

Usuario abre la app → Auth0 verifica identidad → React carga la interfaz → Netlify Function consulta Supabase → Los datos se muestran en cards.

Cuando el usuario usa el chat: pregunta → NetlifyFunction busca en Supabase → envía contexto a Groq IA → Groq responde → respuesta llega al chat.

4. Sistema de Autenticación (Auth0)

Auth0 es un servicio especializado en autenticación. En lugar de construir un sistema de login desde cero (lo cual es complejo y propenso a vulnerabilidades), KnowledgeTI delega toda esa responsabilidad a Auth0.

¿Cómo funciona?

1. El usuario entra a la app y ve la pantalla de login.
2. Al hacer clic en 'Entrar con Auth0', es redirigido al sistema de Auth0.
3. Auth0 verifica las credenciales del usuario.
4. Si son correctas, Auth0 genera un token de seguridad y regresa al usuario a la app.
5. La app usa ese token para confirmar que el usuario está autenticado.
6. Cada vez que se recarga la página, el proceso se repite (sesión no persistente por diseño).

Decisiones de diseño importantes:

Decisión	Razón
Los usuarios NO se crean desde la app	Mayor control: solo el administrador puede crear accesos desde el panel de Auth0.
Sesión obligatoria en cada recarga	Seguridad: evita que alguien deje una sesión abierta y otro acceda a la información.
No se pueden cambiar contraseñas desde la aplicación:	todo el manejo de usuarios vive en Auth0

5. Base de Datos (Supabase)

Supabase es la base de datos del proyecto. Usa PostgreSQL por debajo, que es uno de los motores de base de datos más robustos y confiables del mundo. Supabase agrega una capa visual y una API automática que facilita trabajar con los datos.

Tablas del proyecto:

Tabla: knowledge

Contiene todas las soluciones técnicas oficiales de la empresa.

COLUMNA	TIPO	Descripción
id	UUID	Identificador único generado automáticamente.
categoria	TEXT	Categoría del problema (ej: Citrix, VPN, Impresoras).
pregunta	TEXT	Descripción del problema o pregunta frecuente.
solucion	TEXT	Pasos o descripción de la solución al problema.

Tabla: sugerencias

Almacena las sugerencias enviadas por los usuarios. El administrador las revisa manualmente y decide si las incorpora a la tabla knowledge.

COLUMNA	TIPO	Descripción
id	UUID	Identificador único.
pregunta	TEXT	El problema que el usuario sugiere agregar.
respuesta	TEXT	La solución propuesta por el usuario.

6. Backend — Netlify Functions

Las Netlify Functions son pequeñas funciones de servidor que se ejecutan en la nube. Son el 'backend' del proyecto: manejan la lógica que no puede ir en el frontend porque requiere claves secretas o acceso seguro a la base de datos.

■ ¿Por qué no poner todo en el frontend? Porque si las claves de Supabase o de la IA van en el código del navegador, cualquiera puede verlas y usarlas. Las Functions las mantienen seguras.

Funciones del proyecto:

Función	Método	Qué hace
getKnowledge.js	GET	Identificador único.
crearSugerencia.js	POST	El problema que el usuario sugiere agregar.
aiChat.js	POST	La solución propuesta por el usuario.

Variables de entorno (secretos):

Las Functions usan variables de entorno para guardar información sensible. Estas variables se configuran en Netlify y nunca aparecen en el código:

Variable	Uso
SUPABASE_URL	Dirección de la base de datos en Supabase.
SUPABASE_SERVICE_ROLE_KEY	Clave de acceso completo a Supabase (solo backend).
GROQ_API_KEY	Clave de acceso a la API de inteligencia artificial Groq.

7. Cómo Funciona la IA (Moffy)

Moffy es el asistente de IA de KnowledgeTI. Es importante entender que la IA no fue 'entrenada' desde cero con los datos de la empresa. En cambio, se usa una técnica llamada RAG (Retrieval-Augmented Generation).

¿Qué es RAG?

RAG significa 'Generación Aumentada por Recuperación'. En términos simples: en vez de entrenar un modelo de IA (lo cual cuesta millones de dólares), simplemente le damos a la IA la información relevante justo antes de que responda. Es como si le pasaras los apuntes a alguien inteligente antes de un examen.

Paso a paso de cómo responde Moffy:

1. Recibe la pregunta	El usuario escribe su problema en el chat.
2. Divide en palabras clave	La función divide la pregunta en palabras importantes (más de 3 letras) e ignora palabras como 'el', 'de', 'con'.
3. Busca en Supabase	Busca en la tabla knowledge registros que contengan esas palabras en el problema, solución o categoría.
4. Construye el contexto	Toma los resultados encontrados y los convierte en un texto que la IA pueda entender.
5. Incluye el historial	Agrega los mensajes anteriores de la conversación para que la IA tenga memoria del contexto.
6. Llama a Groq	Envía todo a la API de Groq con instrucciones de cómo debe comportarse (system prompt).
7. Groq responde	El modelo LLaMA 3.1 procesa toda la información y genera una respuesta en español.
8. Muestra la respuesta	La respuesta llega al chat y el usuario la ve.

El System Prompt — instrucciones de comportamiento:

El system prompt son las instrucciones que le damos a la IA para definir cómo debe comportarse. Es lo que hace que Moffy sea un asistente de TI y no un asistente genérico:

Eres un asistente técnico de TI para KnowledgeTI llamado Moffy. Responde SOLO basandote en las soluciones de la base de conocimiento. Si la solución esta en la base de datos usala directamente. Si una solución ya se intentó y no funcionó, sugiere una alternativa. Responde en español de forma clara y paso a paso si es necesario.

8. Manual de Uso

8.1 Acceso a la aplicación

1. Abrir el navegador y dirigirse a la URL de KnowledgeTI.
2. Hacer clic en el botón 'Entrar con Auth0'.
3. Ingresar las credenciales proporcionadas por el administrador.
4. Una vez autenticado, se cargará la pantalla principal con las cards de conocimiento.

8.2 Consultar soluciones (Cards)

En la pantalla principal se muestran todos los registros de la base de conocimiento organizados en tarjetas. Cada tarjeta presenta la categoría, el problema y su solución. Además, se pueden filtrar por categoría utilizando la barra de navegación superior.

8.3 Usar el asistente Moffy

1. Hacer clic en el ícono de Moffy en la esquina de la pantalla.
2. Escribir el problema técnico en el campo de texto.
3. Presionar Enter o el botón Enviar.
4. Esperar la respuesta (los tres puntos indican que Moffy está procesando).
5. Si la solución no funciona, decírselo a Moffy: él buscará una alternativa.
6. Para cerrar el chat, hacer clic fuera del panel.

8.4 Enviar una sugerencia

Si un usuario encuentra una solución a un problema que no está en la base de conocimiento, puede sugerirla mediante el formulario de sugerencias. El administrador la revisará y, si es válida, la agregará oficialmente a la base de conocimiento.

8.5 Administración (solo admin)

- Gestión de usuarios: se hace directamente desde el panel de Auth0.
- Agregar nuevos registros: desde el Table Editor de Supabase, tabla knowledge.
- Revisar sugerencias: desde el Table Editor de Supabase, tabla sugerencias.
- Ver logs del sistema: desde el panel de Netlify, sección Functions.