

Reporte del Proyecto

Base de Datos

- **Archivo:** debug_supabase.py
Propósito: El código proporciona funciones para verificar y establecer la conexión con la plataforma Supabase, así como para probar la inserción de datos en las tablas "contactos" y "mensajes" de la base de datos. También incluye funciones auxiliares para mostrar el estado de las operaciones, como la conexión a Supabase y la inserción de datos en las tablas. Además, se cargan las variables de entorno necesarias utilizando el archivo `.env`.
Rutas: Ninguna
Archivos CSS: Ninguno
Archivos HTML: Ninguno
Tablas y columnas:
Archivos .py usados: supabase, dotenv, datetime, termcolor
- **Archivo:** handle_ai.py
Propósito: El propósito de este código es implementar funciones relacionadas con la interacción de un asistente de inteligencia artificial (IA) llamado Nora AI, que forma parte de Aura Marketing. La función `cargar_base_conocimiento()` se encarga de obtener un fragmento de contenido de una tabla llamada "base_conocimiento" desde Supabase, un servicio de base de datos y backend. La función `manejar_respuesta_ai(mensaje_usuario)`
Rutas: Ninguna
Archivos CSS: Ninguno
Archivos HTML: Ninguno
Tablas y columnas:
Archivos .py usados: dotenv, clientes.aura.utils.error_logger, utils.supabase_client
- **Archivo:** handle_keywords.py
Propósito: Este código define una función llamada `manejar_respuesta_keywords` que recibe un mensaje de usuario como entrada. La función obtiene una lista de respuestas desde la base de datos a través de la función `obtener_respuestas` y luego itera sobre cada respuesta. Para cada respuesta, verifica si la palabra clave asociada a esa respuesta está presente en el mensaje del usuario en minúsculas. Si encuentra una coincidencia, la función devuelve la respuesta asociada a
Rutas: Ninguna
Archivos CSS: Ninguno
Archivos HTML: Ninguno
Tablas y columnas:
Archivos .py usados: utils.db.bot_data

Otros

- **Archivo:** from clientes.aura.utils.py
Propósito: Este código define una función llamada `procesar_mensaje` que se encarga de procesar un mensaje recibido. En primer lugar, se normaliza el número del remitente y se limpia el mensaje recibido. Luego, se guarda el mensaje en un historial con información del usuario. A continuación, se intenta manejar la respuesta del mensaje utilizando diferentes funciones como manejar palabras clave, manejar archivos adjuntos y manejar inteligencia artificial. Si ninguna
Rutas: Ninguna
Archivos CSS: Ninguno
Archivos HTML: Ninguno
Tablas y columnas:
Archivos .py usados: clientes.aura.utils.normalize, clientes.aura.utils.history, clientes.aura.utils.twilio_sender, clientes.aura.handlers.handle_keywords, clientes.aura.handlers.handle_ai, clientes.aura.handlers.handle_files

- **Archivo:** process_message.py

Propósito: Este código se encarga de procesar mensajes recibidos de usuarios en un sistema de asistente virtual llamado Aura. Algunas de las acciones que realiza son normalizar números de teléfono, limpiar mensajes, guardar mensajes en un historial, enviar mensajes a través de Twilio, cargar configuraciones, y manejar respuestas automáticas, archivos adjuntos e inteligencia artificial. Una parte importante es la verificación de si un contacto tiene desactivada la inteligencia artificial,

Rutas: Ninguna

Archivos CSS: Ninguno

Archivos HTML: Ninguno

Tablas y columnas:

Archivos .py usados: clientes.aura.utils.normalizador, clientes.aura.utils.limpieza, clientes.aura.utils.history, clientes.aura.utils.twilio_sender, clientes.aura.utils.settings_loader, clientes.aura.handlers.handle_keywords, clientes.aura.handlers.handle_ai, clientes.aura.handlers.handle_files, clientes.aura.utils.comunicacion_contextual, utils.db.contactos