

Agno — Preguntas Frecuentes (FAQ)

1. Variables de Entorno

- **OPENAI_API_KEY**
Clave para acceder a los modelos y embeddings de OpenAI.
Debe estar definida para la mayoría de los flujos por defecto.
 - **OPENAI_API_BASE**
Permite cambiar el endpoint de la API de OpenAI (por ejemplo, para usar OpenRouter o proxies compatibles).
 - **GOOGLE_API_KEY**
Clave para usar Gemini o servicios de Google AI.
 - **PGVECTOR_URL**
URL de conexión para la base de datos vectorial (PgVector).
 - **AGNO_LOG_LEVEL**
Controla el nivel de logs de Agno (INFO, DEBUG, etc).
 - **AGNO_EMBEDDINGS_PROVIDER**
Permite forzar el proveedor de embeddings (openai, google, etc).
 - **AGNO_AGENT_MODEL**
Permite definir el modelo LLM por defecto.
-

2. Instalación y Requisitos

- **Python 3.8+** es requerido.
- Se recomienda el uso de entornos virtuales ([venv](#), conda).
- Instalación estándar:

```
pip install agno
```

- [Tutorial de instalación](#)
 - [Guía de configuración](#)
 - [Guía de uso](#)
 - [Guía de desarrollo](#)
 - Para bases vectoriales, asegúrate de tener PostgreSQL con la extensión **pgvector** habilitada.
-

3. Modelos y Proveedores

- **Modelos soportados:**
OpenAI, Google Gemini, OpenRouter y cualquier otro compatible con la API de OpenAI.
- **Selección de modelo:**
Puedes cambiar el modelo por variable de entorno o directamente en el código.
- **Modelos gratuitos:**
Usa los etiquetados como :free en OpenRouter para evitar costos durante el prototipado.

4. Embeddings

- **Proveedor por defecto:**
OpenAI, salvo que se especifique otro.
- **Cambio de proveedor:**
Se puede cambiar por variable de entorno o argumento en el código.
- **API Key:**
Es obligatorio definir la API key del proveedor seleccionado.
- **Errores comunes:**
Si no se especifica un embedder, Agno intentará usar OpenAI y requerirá su API key.

5. Vector Databases

- **Soporte:**
PgVector (PostgreSQL), Qdrant, Chroma, entre otros.
- **Requisitos:**
La base de datos debe estar corriendo y accesible.
- **Creación automática:**
Agno puede crear la tabla vectorial si no existe.
- **Configuración:**
Usa la variable de entorno PGVECTOR_URL o pásala directamente en el código.

6. Manejo de PDFs y Archivos

- **Indexación:**
Puedes indexar PDFs locales o remotos.
- **Extracción de texto:**
Agno utiliza PyMuPDF para extraer el texto automáticamente.
- **Archivos grandes:**
El procesamiento puede tardar más tiempo según el tamaño del archivo.

7. Troubleshooting (Solución de Problemas)

- **Errores de API Key:**
Verifica que la variable de entorno esté definida y sea válida.
- **Problemas de conexión a la base de datos:**
Revisa la URL, los permisos y que la base esté corriendo.
- **Problemas de embeddings:**
Asegúrate de tener la API key correcta y el proveedor configurado.
- **Errores de importación:**
Verifica la versión de Python y que todas las dependencias estén instaladas.

- **Logs:**
Usa AGNO_LOG_LEVEL=DEBUG para obtener más información en los logs.
-

8. Buenas Prácticas

- Usa variables de entorno para credenciales y configuración sensible.
 - Mantén tus dependencias actualizadas.
 - Usa Docker para bases de datos y servicios auxiliares.
 - Consulta la documentación oficial para ejemplos y casos de uso avanzados.
 - Realiza pruebas con modelos gratuitos antes de pasar a producción.
-

9. Recursos Útiles

- [Documentación oficial de Agno](#)
 - [Ejemplos de configuración de PgVector](#)
 - [OpenRouter: Modelos gratuitos y pagos](#)
 - [Google AI Studio para Gemini](#)
-

10. Ejemplo de Configuración de Variables de Entorno

```
OPENAI_API_KEY=sk-xxxxxx
OPENAI_API_BASE=https://openrouter.ai/api/v1
GOOGLE_API_KEY=AIzaSyxxxxxx
PGVECTOR_URL=postgresql://user:password@localhost:5432/dbname
AGNO_LOG_LEVEL=INFO
```

- -
 -
 -
-

11. Ejemplo de Docker Compose para PgVector y Adminer

```
services:
  db:
    image: agnohq/pgvector:16
    restart: always
    container_name: pg-db
    volumes:
      - ./pgvolume:/var/lib/postgresql/data/pgdata
    environment:
      POSTGRES_DB: ${POSTGRES_DB}
```

```
POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
POSTGRES_USER: ${POSTGRES_USER}
PGDATA: /var/lib/postgresql/data/pgdata
ports:
  - "5432:5432"
adminer:
  image: adminer
  restart: always
  ports:
    - 8080:8080
  •
  •
  •
  •
```

12. Contacto y Soporte

- Consulta la [sección de soporte](#) para dudas técnicas o problemas avanzados.
- Participa en la comunidad o foros recomendados por Agno para compartir experiencias y soluciones.