LLM Remote

Puente Cifrado Telegram ↔ IA Multi-Proveedor

REDEGAL - A SMART COMPANY

Version 2.1 · Febrero 2026 · Documento interno

Indice

1. Introduccion

LLM Remote es una herramienta interna que permite a los miembros del equipo de Redegal interactuar con multiples proveedores de IA (Claude Code, OpenAI, Gemini, Groq) directamente desde Telegram, con cifrado de nivel bancario.

Que puedes hacer?

- **Desarrollo remoto** — Ejecuta Claude Code en tu maquina desde cualquier lugar
- **Consultas rapidas** — Pregunta a GPT-4o, Gemini o Llama desde el movil
- **Notas de voz** — Envia un audio y recibe transcripcion + respuesta IA
- **Analisis visual** — Envia fotos y capturas para que la IA las analice
- **Archivos** — Envia codigo, CSV, PDF y la IA los procesa
- **Busqueda web** — Busca en internet y obtiene resumen con IA
- **Pipelines** — Encadena operaciones: paso1 → paso2 → paso3
- **Tareas programadas** — Ejecuta prompts periodicamente
- **SSH remoto** — Ejecuta comandos en servidores desde Telegram
- **Text-to-Speech** — Recibe respuestas como notas de voz
- **Grupos** — Usa el bot en grupos de Telegram
- **MCP** — Conecta herramientas externas via Model Context Protocol
- **Multi-proyecto** — Cambia entre proyectos con un comando
- **Seguro** — Todo cifrado, autenticado y con audit log

> **Requisitos previos**
>
> Node.js 20+, un bot de Telegram (de @BotFather), y opcionalmente Claude Code CLI.

2. Proveedores IA

LLM Remote soporta 5 proveedores. Cambia entre ellos con `/ia <nombre>` en Telegram.

Proveedor	Comando	Modo	Coste
Claude Code	`/ia claude`	**Agentic** Lee/escribe ficheros, ejecuta comandos	**Segun plan**
OpenAI GPT-4o	`/ia openai`	Chat API + Vision	**Pay-per-use**
Gemini 2.5 Flash	`/ia gemini`	Chat API + Vision	**Gratis** 20 req/dia
Groq Llama 3.3	`/ia groq`	Chat API (ultra-rapido) + Whisper TTS	**Gratis** 30 req/min
Anthropic Sonnet	`/ia anthropic`	Chat API + Vision (sin agentic)	**Pay-per-use**

> **Recomendacion para uso diario**
>
> Usa **Groq** para consultas rapidas (gratis, respuesta en <1s) y **Claude Code** cuando necesites que la IA modifique archivos o ejecute comandos en tu maquina.

Diferencia entre Claude Code y Anthropic API

- **Claude Code** (`/ia claude`) — Modo agentic completo. Puede leer ficheros, escribir codigo, ejecutar comandos, navegar tu proyecto. Usa el CLI de Claude Code.
- **Anthropic API** (`/ia anthropic`) — Solo chat. Responde preguntas pero no tiene acceso a tu sistema de archivos. Usa la API directamente.

3. Instalacion

3.1 Instalacion automatica (recomendado)

Un solo comando que verifica requisitos, descarga, instala y configura:

```
bash installer.sh
```

El instalador hace todo automaticamente:

1. Verifica Node.js 20+, npm, git, Claude Code CLI
2. Descarga o actualiza el proyecto
3. Instala dependencias npm
4. Lanza el configurador interactivo (6 pasos)
5. Opcionalmente crea un servicio de auto-arranque

3.2 Instalacion manual

```
# 1. Clonar repositorio
git clone <repo-url> ~/llm-remote
cd ~/llm-remote

# 2. Instalar dependencias
npm install

# 3. Configurar
npm run setup

# 4. Arrancar
npm start
```

3.3 Requisitos

Requisito	Version	Obligatorio	Notas
Node.js	20+	Si	`brew install node` o nodejs.org

npm	cualquiera	Si	Viene con Node.js
git	cualquiera	Si	`xcode-select --install` (macOS)
Claude Code CLI	cualquiera	No	`npm i -g @anthropic-ai/claude-code`

Sin Claude Code CLI

Si no instalas Claude Code CLI, el proveedor `/ia claude` no funcionara. Los demas proveedores (OpenAI, Gemini, Groq, Anthropic) si funcionaran sin problema.

4. Configuracion

4.1 Crear bot de Telegram

1. Abre **@BotFather** en Telegram
2. Envia `/newbot`
3. Elige un nombre (ej: *"Mi LLM Remote"*)
4. Elige un username (ej: *"mi_llm_remote_bot"*)
5. Copia el **token** que te da BotFather

4.2 Obtener tu Telegram User ID

1. Abre **@userinfobot** en Telegram
2. Envia `/myid`
3. Copia el numero (ej: *123456789*)

4.3 Obtener API keys (opcional)

Proveedor	URL	Coste
Gemini	https://aistudio.google.com/apikey	Gratis
Groq	https://console.groq.com/keys	Gratis
OpenAI	https://platform.openai.com/api-keys	De pago
Anthropic	https://console.anthropic.com/settings/keys	De pago

4.4 Wizard de configuracion

El wizard (`npm run setup`) te guia en 6 pasos:

```
╔═══════════════════════════════════════╗
║   LLM Remote — Configurador           ║
╚═══════════════════════════════════════╝

── 1/6  Telegram ──
```

```
  Token del bot: <pegar token>
  IDs autorizados: <tu user ID>

── 2/6  Seguridad ──
  PIN: <auto-generado o personalizado>
  Contrasena maestra: <auto-generada>

── 3/6  Sesion y Limites ──
  Timeout: 15 min
  Max comandos/min: 10

── 4/6  Claude Code CLI ──
  Binario: claude
  Directorio: /Users/tu-usuario

── 5/6  Proveedores IA ──
  OpenAI, Gemini, Anthropic, Groq (opcionales)

── 6/6  Logging ──
  Nivel: info
```

Reconfigurar en cualquier momento

Ejecuta `npm run setup` de nuevo. Los valores actuales aparecen como defaults.

5. Uso desde Telegram

5.1 Primer uso

1. Arranca el bot: `npm start` (o el servicio lo hace automaticamente)
2. Abre tu bot en Telegram
3. Envia `/start` para ver los comandos
4. Autenticate: `/auth <tu-PIN>`
5. El mensaje del PIN se borra automaticamente por seguridad
6. Escribe cualquier mensaje — va directamente al proveedor IA activo

5.2 Cambiar proveedor IA

```
# Ver proveedores disponibles
/ia

# Cambiar a Groq (gratis, ultra-rapido)
/ia groq

# Cambiar a Claude Code (agentic)
/ia claude

# Cambiar a Gemini (gratis)
/ia gemini
```

5.3 Trabajar con proyectos

```
# Ver directorio actual
/project

# Cambiar a otro proyecto
/project ~/mi-proyecto

# Ahora los comandos a Claude Code operan en ~/mi-proyecto
Refactoriza el archivo main.js
```

5.4 Contexto conversacional

LLM Remote mantiene un historial de hasta **20 mensajes** por usuario. La IA recuerda lo que hablaste antes.

```
# La IA recuerda mensajes anteriores
Explica que es un closure en JavaScript
# ... respuesta ...
Dame un ejemplo practico
# ... la IA sabe que hablas de closures ...

# Limpiar contexto cuando cambies de tema
/clear
```

5.5 Sesiones y seguridad

```
# Ver estado de la sesion
/status

# Bloquear manualmente
/lock

# Re-autenticarse
/auth <PIN>

# Ver historial (descifrado)
/history
```

6. Multimedia: Voz, Fotos, Archivos

6.1 Notas de voz

Envia un audio o nota de voz a Telegram. El bot:

1. Transcribe el audio con **Groq Whisper** (gratis) o OpenAI Whisper
2. Muestra la transcripcion
3. Envia el texto al proveedor IA activo
4. Devuelve la respuesta (y opcionalmente como nota de voz con TTS)

> **Perfecto para el movil**
>
> Graba un audio: "Explica como funciona async/await en JavaScript" y recibe la respuesta escrita. Activa `/voz` para recibirla tambien como nota de voz.

6.2 Text-to-Speech (TTS)

Activa el modo voz para recibir las respuestas como notas de voz ademas de texto:

```
# Activar/desactivar modo voz
/voz

# Ahora todas las respuestas incluyen nota de voz
Explica que es Docker en 2 frases
# Recibes texto + audio
```

Proveedores TTS soportados: **OpenAI TTS** (voz nova, formato opus) y **Groq TTS** (PlayAI). Se usa el primero disponible.

6.3 Analisis de fotos

Envia una foto o captura de pantalla. La IA la analiza con Vision:

- **OpenAI GPT-4o Vision** (prioridad)
- **Anthropic Claude Vision** (fallback)
- **Google Gemini Vision** (fallback)

```
# Envia una foto con caption opcional
[foto] Que error hay en este codigo?

# O envia sin caption para analisis general
[foto]
```

6.4 Procesamiento de archivos

Envia archivos de codigo, texto, datos o PDF. El bot extrae el contenido y lo envia a la IA.

Tipo	Extensiones
Codigo	.js, .ts, .py, .go, .rs, .java, .c, .cpp, .rb, .php, .swift, .kt, .sh, .bash
Datos	.csv, .json, .yaml, .yml, .xml, .toml
Texto	.md, .txt, .log, .html, .css, .sql, .env, .conf, .ini, .cfg
Documentos	.pdf (extraccion basica de texto)

Limite: **5 MB** por archivo. CSV muestra las primeras 50 filas como preview.

7. Herramientas Avanzadas

7.1 Busqueda web

Busca en internet y obtiene un resumen con IA. No necesita API key (usa DuckDuckGo).

```
# Buscar y resumir
/web novedades Node.js 22

# Buscar noticias
/web ultimas noticias inteligencia artificial 2026
```

7.2 Tareas programadas

Ejecuta prompts automaticamente a intervalos regulares.

```
# Crear tarea cada 24 horas
/schedule 24h Resume el estado de los repos

# Crear tarea cada hora
/schedule 1h Revisa si hay errores en los logs

# Intervalos soportados: 5m, 30m, 1h, 6h, 24h, 7d

# Listar tareas
/schedules

# Eliminar tarea
/unschedule 3
```

7.3 Pipelines

Encadena multiples operaciones donde la salida de un paso alimenta al siguiente:

```
# Buscar + resumir
/pipe busca tendencias React 2026 → resume en 3 puntos clave

# Analizar + sugerir + redactar
/pipe lee el README del proyecto → sugiere mejoras → redacta un PR
```

```
# Datos + estadisticas + formato
/pipe analiza este CSV → genera estadisticas → formatea como tabla markdown
```

Separadores validos: `→`, `|`, `>`

7.4 Servidores MCP

Conecta herramientas externas via **Model Context Protocol**:

```
# Anadir servidor MCP
/mcp add github npx -y @modelcontextprotocol/server-github

# Ver herramientas disponibles
/mcp tools

# Ejecutar herramienta
/mcp call github/list_repos {"owner": "redegal"}

# Ver servidores conectados
/mcp

# Eliminar servidor
/mcp remove github
```

8. SSH Remoto

Ejecuta comandos en servidores remotos directamente desde Telegram. Usa SSH nativo (sin dependencias adicionales).

8.1 Configurar servidores

```
# Anadir servidor
/ssh add prod root@37.27.92.122
/ssh add staging deploy@staging.example.com 2222

# Formato completo
/ssh add <nombre> <user@host> [puerto] [ruta-clave-ssh]

# Listar servidores
/ssh list

# Eliminar servidor
/ssh remove prod
```

8.2 Ejecutar comandos

```
# Espacio en disco
/ssh prod df -h

# Estado de contenedores Docker
/ssh prod docker ps

# Ultimas lineas del log
/ssh prod tail -20 /var/log/syslog

# Estado del sistema
/ssh prod uptime && free -h
```

8.3 Seguridad SSH

> **Comandos bloqueados**
>
> Los siguientes comandos estan bloqueados por seguridad:

- `rm -rf /` (borrado raiz)
- `mkfs` (formatear disco)
- `dd if=` (escritura directa a disco)
- `shutdown`, `reboot`, `init 0`

Otras medidas de seguridad:

- Timeout de **30 segundos** por comando
- Salida truncada a 4000 caracteres
- `BatchMode=yes` (sin prompts interactivos)
- Todos los comandos quedan en el audit log
- Configuracion persistente en `data/ssh-servers.json`

9. Grupos de Telegram

LLM Remote funciona en grupos y supergrupos de Telegram. En grupos, el bot solo responde a:

- **Comandos** — `/status`, `/ia groq`, etc.
- **Menciones** — `@mi_bot Explica esto`
- **Respuestas al bot** — Responde a un mensaje del bot
- **Media** — Fotos, audios y archivos enviados en el grupo

Los mensajes normales del grupo se ignoran para no saturar la conversacion.

> **Configurar grupo**
>
> 1. Anade el bot al grupo
> 2. Todos los usuarios autorizados (whitelist) pueden usarlo
> 3. Usuarios no autorizados son ignorados silenciosamente

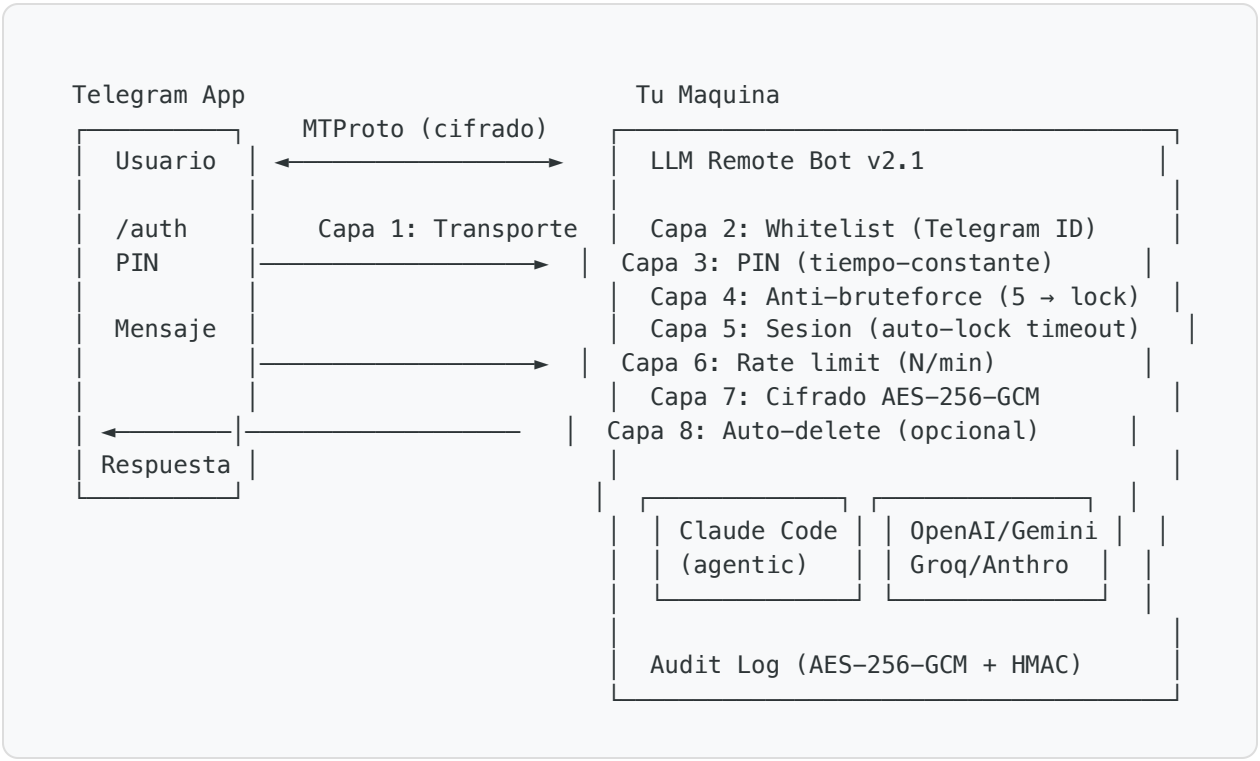
10. Referencia de Comandos

Comando	Descripcion	Ejemplo
`/start`	Muestra ayuda y proveedores disponibles	`/start`
`/auth <PIN>`	Autenticarse. El mensaje se borra automaticamente.	`/auth 1234`
`/ask <prompt>`	Enviar prompt explicito al proveedor activo	`/ask explica que hace este regex`
`/ia [nombre]`	Ver o cambiar proveedor IA	`/ia groq`
`/clear`	Limpiar contexto conversacional (20 msgs)	`/clear`
`/project [ruta]`	Ver o cambiar directorio de trabajo	`/project ~/app`
`/status`	Info de sesion, proveedor, TTS, SSH	`/status`
`/history`	Ultimos 15 comandos del audit log	`/history`
`/kill`	Matar proceso en ejecucion	`/kill`
`/lock`	Bloquear sesion inmediatamente	`/lock`
`/voz`	**Nuevo** Activar/desactivar respuestas por voz (TTS)	`/voz`
`/web <query>`	Busqueda web + resumen IA	`/web novedades React 2026`
`/schedule`	Crear tarea programada	`/schedule 24h Resumen diario`
`/schedules`	Listar tareas programadas	`/schedules`
`/unschedule <id>`	Eliminar tarea programada	`/unschedule 3`

`/pipe`	Ejecutar pipeline de pasos	`/pipe busca X → resume`
`/mcp`	Gestionar servidores MCP	`/mcp add github npx ...`
`/ssh`	**Nuevo** SSH remoto: add, list, remove, ejecutar	`/ssh prod df -h`
`/help`	Mostrar todos los comandos	`/help`
(texto libre)	Se envia directamente al proveedor activo	*Como optimizo esta query SQL?*
(audio)	Transcripcion + envio a IA	*[nota de voz]*
(foto)	Analisis visual con Vision	*[captura de pantalla]*
(archivo)	Extraccion + analisis con IA	*[archivo.csv]*

11. Arquitectura de Seguridad

11.1 Modelo de 8 capas



11.2 Cifrado

Componente	Especificacion
Algoritmo	AES-256-GCM (cifrado autenticado)
Derivacion de clave	PBKDF2 con 310.000 iteraciones + SHA-512
IV (vector de inicializacion)	16 bytes aleatorios por mensaje
Salt	32 bytes aleatorios por mensaje
Integridad	HMAC-SHA256 sobre todo el payload
Resultado	Cada cifrado es unico, incluso con el mismo texto

11.3 Autenticacion

- **Whitelist** — Solo los Telegram IDs configurados pueden interactuar. Usuarios no autorizados son ignorados silenciosamente.
- **PIN** — Comparacion en tiempo constante para prevenir ataques de timing.
- **Anti-bruteforce** — 5 intentos fallidos → bloqueo de 15 minutos.
- **Sesiones** — Auto-lock tras inactividad configurable (default: 15 min).
- **Auto-delete** — El mensaje con el PIN se borra automaticamente de Telegram.

> **Importante: Seguridad por instalacion**
>
> Cada instalacion genera su propia contrasena maestra de cifrado. Las claves no se comparten entre usuarios. Si pierdes tu `.env`, no hay forma de recuperar los datos del audit log.

12. Arquitectura Tecnica

12.1 Stack tecnologico

Componente	Tecnologia	Justificacion
Runtime	Node.js 20+	Async/await nativo, fetch nativo, sin dependencias nativas
Bot Telegram	grammY	Framework moderno, TypeScript-ready, middleware
Cifrado	node:crypto	Modulo nativo, AES-256-GCM, PBKDF2, HMAC
APIs IA	fetch nativo	Zero dependencias, HTTP estandar
Transcripcion	Groq Whisper / OpenAI Whisper	Groq gratis, OpenAI fallback
Vision	GPT-4o / Claude / Gemini	Triple fallback para maxima disponibilidad
TTS	OpenAI TTS / Groq TTS	Dual fallback
SSH	ssh nativo (child_process)	Zero dependencias, usa claves del sistema
Busqueda web	DuckDuckGo scraping	Sin API key, gratis
Configuracion	dotenv	Estandar de la industria
Audit log	NDJSON cifrado	Append-only, sin BD, portatil

12.2 Estructura del proyecto

```
llm-remote/
├── src/
│   ├── index.js              # Punto de entrada
│   ├── bot.js                # Bot Telegram + handlers
│   ├── setup.js              # Configurador interactivo
│   ├── auth/
│   │   ├── guard.js          # Whitelist + anti-bruteforce + grupos
```

```
│   │   └── session.js        # Sesiones + timeout
│   ├── crypto/
│   │   └── cipher.js         # AES-256-GCM + HMAC + PBKDF2
│   ├── providers/
│   │   ├── base.js           # Interfaz base
│   │   ├── manager.js        # Gestor multi-proveedor
│   │   ├── claude.js         # Claude Code CLI
│   │   ├── openai.js         # OpenAI GPT-4o
│   │   ├── gemini.js         # Google Gemini
│   │   ├── groq.js           # Groq Llama 3.3
│   │   └── anthropic.js      # Anthropic Sonnet
│   ├── context/
│   │   └── memory.js         # Memoria conversacional (20 msgs)
│   ├── media/
│   │   ├── voice.js          # Transcripcion (Whisper)
│   │   ├── vision.js         # Analisis de imagenes (Vision)
│   │   ├── files.js          # Procesamiento de archivos
│   │   └── tts.js            # Text-to-Speech
│   ├── search/
│   │   └── web.js            # Busqueda web (DuckDuckGo)
│   ├── scheduler/
│   │   └── scheduler.js      # Tareas programadas
│   ├── pipeline/
│   │   └── pipeline.js       # Motor de pipelines
│   ├── mcp/
│   │   └── client.js         # Cliente MCP (JSON-RPC stdio)
│   ├── remote/
│   │   └── ssh.js            # SSH remoto + seguridad
│   ├── claude/
│   │   └── formatter.js      # Chunking para Telegram
│   ├── security/
│   │   ├── ratelimit.js      # Rate limiting
│   │   └── audit.js          # Audit log cifrado
│   └── utils/
│       ├── config.js         # Configuracion
│       ├── logger.js         # Logger
│       └── keygen.js         # Generador de claves
├── tests/                    # 53 tests (crypto, memory, files, search, pipelin
├── docs/                     # Manuales ES/EN (HTML + PDF)
├── installer.sh              # Instalador corporativo
├── .env.example              # Plantilla de config
└── package.json
```

12.3 Dependencias

Solo **2 dependencias** de produccion (zero nativas):

Paquete	Version	Funcion
grammy	^1.31	Framework bot Telegram
dotenv	^16.4	Variables de entorno

13. Resolucion de Problemas

Problema	Causa	Solucion
El bot no responde	No esta corriendo o token incorrecto	Verificar `npm start` y el token en `.env`
"Sesion expirada"	Timeout de inactividad	Enviar `/auth <PIN>` de nuevo
"Bloqueado"	5 intentos de PIN fallidos	Esperar 15 minutos
"Rate limited"	Demasiados comandos por minuto	Esperar unos segundos
"Provider not configured"	Falta API key del proveedor	Ejecutar `npm run setup` y anadir la API key
Claude Code no funciona	CLI no instalado	`npm i -g @anthropic-ai/claude-code`
Gemini error 429	Limite diario alcanzado (20 req)	Cambiar a Groq: `/ia groq`
TTS no funciona	Sin API key de OpenAI ni Groq	Configurar al menos una API key de OpenAI o Groq
Voz no transcribe	Sin API key de Groq ni OpenAI	Groq es gratis: obtener key en console.groq.com
SSH timeout	Servidor no accesible o clave SSH incorrecta	Verificar conectividad y que las claves SSH estan configuradas
SSH "comando bloqueado"	Comando peligroso detectado	Proteccion de seguridad, no se puede desactivar
Bot no responde en grupo	Solo responde a comandos/menciones	Mencionar @bot o responder a un mensaje del bot
Error de cifrado	.env corrupto o cambiado	El audit log anterior no sera legible. Borrar `data/` y empezar de nuevo.

Logs

```
# Ver logs en tiempo real
tail -f ~/llm-remote/data/llm-remote.log

# Si usa servicio en macOS
tail -f ~/llm-remote/data/llm-remote.log

# Si usa servicio en Linux
journalctl --user -u llm-remote -f
```

Reconfigurar

```
cd ~/llm-remote && npm run setup
```

Tests

```
# Ejecutar los 53 tests
npm test
```

Desinstalar

```
bash ~/llm-remote/installer.sh --uninstall
```

14. Variables de Entorno

Variable	Req.	Default	Descripcion
`TELEGRAM_BOT_TOKEN`	Si	—	Token del bot de Telegram
`AUTHORIZED_USERS`	Si	—	IDs Telegram autorizados (coma)
`AUTH_PIN`	Si	—	PIN de autenticacion
`MASTER_PASSWORD`	Si	—	Contrasena maestra cifrado (16+ chars)
`SESSION_TIMEOUT_MIN`	No	15	Timeout sesion (minutos)
`RATE_LIMIT_PER_MIN`	No	10	Max comandos/minuto
`AUTO_DELETE_SEC`	No	0	Auto-borrar mensajes (0 = off)
`CLAUDE_BIN`	No	claude	Ruta binario Claude Code
`DEFAULT_WORK_DIR`	No	$HOME	Directorio trabajo por defecto
`MAX_CONCURRENT`	No	2	Procesos Claude simultaneos
`OPENAI_API_KEY`	No	—	API key OpenAI (chat + vision + TTS)
`OPENAI_MODEL`	No	gpt-4o	Modelo OpenAI
`GEMINI_API_KEY`	No	—	API key Google Gemini
`GEMINI_MODEL`	No	gemini-2.5-flash	Modelo Gemini
`ANTHROPIC_API_KEY`	No	—	API key Anthropic
`ANTHROPIC_MODEL`	No	claude-sonnet-4	Modelo Anthropic
`GROQ_API_KEY`	No	—	API key Groq (chat + whisper + TTS)
`GROQ_MODEL`	No	llama-3.3-70b	Modelo Groq
`LOG_LEVEL`	No	info	debug/info/warn/error