



SEREIA / CADERNOS DE OPERAÇÃO

INTELIGÊNCIA ARTIFICIAL

Hermes Agent

O framework open-source que transforma a IA de chatbot em camada operacional da sua empresa.

SEREIA.IO · 2026

07

A tese em uma frase

Hermes não é uma ferramenta. É uma **infraestrutura de continuidade**. A ferramenta importa pouco — quem ganha o jogo é quem domina a arquitetura por trás dela.

O Hermes Agent é um framework open-source da **Nous Research**, lançado em fevereiro de 2026, que ultrapassou 153 mil stars no GitHub em meses. Roda na sua infraestrutura (VPS), aprende com cada interação, persiste memória em 4 camadas, e transforma cada repetição em uma *skill* reutilizável. O slogan é *"the AI that grows with you"*.

Mas a frase mais importante deste livro vem do Bruno Okamoto, no vídeo de 02/06/2026 que fundamenta este estudo: **"A ferramenta não importa, porque o jogo não é a ferramenta. O jogo é sobre arquitetura."**

Sumário

- **pp. 3-4** — O que é Hermes · origem Nous Research · Hermes × OpenClaw × Claude Code
- **pp. 5-8** — Anatomia: as 8 camadas · as 4 memórias · skills self-evolving · Soul · segundo cérebro
- **pp. 9-10** — Os 7 passos práticos de instalação e configuração
- **pp. 11-12** — Governança e segurança · comandos e operação avançada
- **pp. 13-14** — Aplicação Sereia · riscos e armadilhas
- **p. 15** — Fontes, bibliografia e cheatsheet de comandos

REGRA DE APLICAÇÃO

Este livro vive em `Sereia Mkt/Estudos/` — pasta flat. O conteúdo é munição interna. **NUNCA** citar Hermes, Nous Research, OpenClaw ou Claude Code em peça cliente — fiel à regra *"Sereia é solução, não tech"*. Em peça cliente, falamos de "camada operacional de IA configurada para o seu contexto" — sem nome de fornecedor.

O que é o Hermes Agent

Um agente de IA autônomo, open-source, com memória persistente, que roda no seu servidor — não num chat na nuvem do fornecedor.

Hermes nasceu na **Nous Research** (laboratório de IA aberto · pesquisa em modelos foundation) e foi lançado publicamente em **fevereiro de 2026**. Em meses, virou um dos projetos open-source mais relevantes do ecossistema de agentes autônomos:

153K+

STARS NO GITHUB

24K+

FORKS ATIVOS

#1TOP APPS NO OPENROUTER
(MAIO/2026)

O que diferencia o Hermes de um chatbot tradicional

- **Vive no seu servidor** — instala numa VPS, não num SaaS. Você é dono dos dados, da memória e da continuidade.
- **Memória persistente em 4 camadas** — não esquece de uma sessão pra outra. Lembra de pessoas, de fatos, de decisões.
- **Skills self-evolving** — cada tarefa repetida vira uma skill reutilizável *automaticamente*, sem você criar manualmente.
- **Gateway multicanal** — Telegram, WhatsApp, dashboard web, Kanban nativo, MCPs — tudo conectado num ponto só.
- **Crons explícitos** — automações agendadas que fazem o agente "acordar" e executar mesmo sem ninguém pedindo.
- **Modelo intercambiável** — você escolhe a LLM (GPT 5.5, Claude, Opus) por tarefa, não fica preso a um fornecedor.

A frase que define o posicionamento

"Hermes não é uma feature de chat. É uma infraestrutura de continuidade. Por isso parece lembrar quem é você — porque tem quatro camadas trabalhando juntas, não um contexto inflado." — Bruno Okamoto, masterclass 02/06/2026.

A CORREÇÃO MENTAL

Hermes **não é um modelo** (não compete com GPT/Claude). Hermes é a **casca operacional** em volta do modelo: memória + skills + tools + automações + governança. O modelo é o cérebro temporário; o Hermes é o corpo persistente.

Hermes × OpenClaw × Claude Code

Três ferramentas, três propósitos. A pergunta certa não é "qual é a melhor" — é "qual resolve o quê".

FERRAMENTA	SLOGAN	FORÇA	QUANDO USAR
OpenClaw	<i>The AI that actually does things</i>	Autonomia ponta-a-ponta	Ações executivas, automações agressivas
Hermes	<i>The AI that grows with you</i>	Continuidade e memória	Operação contínua, evolução incremental
Claude Code	<i>Code agent in your terminal</i>	Trabalho técnico profundo	Codificação, arquitetura, análise técnica

O contexto histórico (curto)

OpenClaw foi lançado por **Peter Steinberg em novembro de 2025** e fez algo raro: *criou um mercado novo*. Antes dele, a discussão de IA girava em torno de LLMs (Claude Chat, ChatGPT). Depois dele, todo CEO de tecnologia (Jensen Huang da NVIDIA inclusive) passou a falar que "toda empresa precisa de uma estratégia OpenClaw". Esse hype destravou bilhões em investimento — e abriu brecha pra concorrentes.

Quando o OpenClaw começou a quebrar sob a pressão de escalar rápido demais, o **Hermes entrou pela porta da estabilidade**: menos features, mais robustez. Hoje (junho/2026) é a ferramenta mais usada no OpenRouter — com 3-4× mais uso que o OpenClaw.

A REGRA DE MIGRAÇÃO

Se sua operação já roda com OpenClaw, NÃO mude. Os dois te levam pro mesmo lugar. Trocar de agente por hype é desperdício de energia. *Se você está começando ou usa de forma rasa, sim, teste o Hermes* — a infraestrutura está mais madura agora.

O combo ideal (a tese Sereia)

Não escolha um. Combine: **Hermes** pra operação contínua e memória · **Claude Code** pra trabalho técnico profundo · **Segundo Cérebro** (próxima página) pra unir os dois. Cada um faz o que faz melhor.

As 8 camadas do Hermes

Pra dominar a ferramenta, você precisa entender o que está rodando por baixo. São oito camadas — cada uma é um botão que você gira pra ajustar o comportamento do agente.

Modelo (LLM)

- 1 O cérebro temporário · GPT 5.5 recomendado por ser generoso em token · NUNCA usar versões "mini" (4.1, 5.4 mini · 5.5 mini) — quebram a experiência.

Memória · 4 camadas

- 2 Peer Cards (memória social), Fuzzy Index (semântica), Prompt Cache (curto prazo · -40% custo), Memory Review Worker (compactação automática). Detalhe na p. 6.

Skills

- 3 Self-evolving · cada tarefa repetida 3× vira skill automaticamente. Diferencial central. Detalhe na p. 7.

Tools (MCPs)

- 4 Conectores externos: Perplexity, Firecrawl, APify, GitHub, Telegram. Sem ferramentas, o agente "só responde, não age".

Gateway

- 5 A camada de cola · liga canais (Telegram, WhatsApp), dashboard, Kanban, plugins, perfis de usuário/agente, chaves API.

Dashboard

- 6 Interface web · gerencia sessões, modelos por tarefa, logs, skills, plugins, perfis, chaves, Kanban nativo. Acesso via Tailscale (VPS segura).

Crons (automações)

- 7 O Hermes NÃO tem heart-beat (diferente do OpenClaw). Você precisa configurar Chrons explícitos pra ele "acordar" e executar rotinas.

Governança

- 8 A camada humana · permissionamento, acessos granulares, separação de senhas (1Password externo), agente de governança auditando outros agentes.

SOBRE O SOUL

Atravessando todas as camadas existe o **Soul** — arquivo de identidade do agente (quem ele é, o que faz, como pensa, tom de voz). É o arquivo mais importante. Trataremos dele na p. 7.

As quatro camadas de memória – o coração do Hermes

O OpenClaw tem 2 camadas. O Hermes tem 4. Esse é o diferencial competitivo que mantém o Hermes em primeiro lugar no OpenRouter.

1 • Peer Cards • memória social

Cada pessoa que conversa com o Hermes vira um *card*. Quem é ela, como prefere ser tratada, histórico, personalidade, tom de voz. Quando você fala com o agente, ele já carrega quem você é. Quando outro membro da equipe fala, carrega o card daquela pessoa. **O agente vira social.**

2 • Fuzzy Index • memória semântica

É um índice tipo Google interno do agente. Você pode pedir "lembra daquela vez que falamos sobre X?" e ele faz busca por palavras-chave ou contexto e recupera. Equivalente à memória semântica do OpenClaw — boa em ambos, mas mais estável aqui.

3 • Prompt Cache • memória de curto prazo

Cache de uma hora do que vocês conversaram. **Reduz custo de tokens em até 40%** porque ele não precisa reprocessar memórias antigas em markdown a cada turno. É a memória "viva" da sessão atual.

4 • Memory Review Worker • compactação automática

Aqui está o segredo. Um *worker* roda em background e decide o que vale guardar: decisões, fatos, preferências, pessoas novas. **Compacta o contexto antes de explodir.**

O nível de compactação é parametrizável (0.5 a 0.9):

- 0.5 = compacta cedo, com 50% de gordura. Seguro, mas perde nuance.
- 0.8 = **RECOMENDADO**. Equilíbrio entre compactar e manter contexto.
- 0.9 = compacta tarde, próximo do limite. Risco de *context overflow*.

O SINTOMA DO CONTEXTO EXPLODIDO

Quando o agente para de responder, demora 10min, faz *fire-and-forget* ("você já fez?" "ah não, desculpa, agora vou fazer") ou perde o assunto da conversa, é context overflow. **O OpenClaw sofre disso. O Hermes sofre menos — mas não está imune.** Solução: comando `/compact` manual quando necessário.

"Uma IA sem memória é um estagiário novo todos os dias. O Hermes é o primeiro estagiário que lembra do briefing da semana passada."

Skills self-evolving + Soul

Dois conceitos que separam o Hermes de um chatbot avançado: skills que se criam sozinhas, e uma identidade-arquivo que define a alma do agente.

Skills • o processo que vira automação

Toda vez que você pede algo ao Hermes pela **terceira vez**, ele transforma aquilo numa skill. Se você pede "analise a planilha de fluxo de caixa" repetidas vezes, na terceira vez nasce uma skill chamada **planilha-fluxo-caixa** com o passo-a-passo cravado. Da próxima vez, ele executa do mesmo jeito — sem variar de leitura de um dia pro outro (como o OpenClaw varia).

É o oposto do OpenClaw, onde você precisa pedir explicitamente *"crie uma skill para isso"*. No Hermes acontece sozinho. **Vantagem:** economia massiva de tempo. **Risco:** em 12-18 meses, você pode ter 2.000+ skills — manutenção vira problema.

A DICA DE OURO

No OpenClaw, você pode imitar o auto-evolving com uma linha no arquivo de Soul: *"toda tarefa repetitiva deve virar uma skill"*. Pronto — você tem o melhor dos dois mundos. A Sereia já faz isso na prática.

Soul • a identidade do agente

O arquivo mais importante de todos. Define quatro camadas da alma do agente:

1

Identidade

Quem ele é, o que faz, pra quem. "Sou um super-agente que faz tudo" vs "Sou especialista em conteúdo".

2

Contrato de runtime

Onde ele vive, qual a fonte de verdade, quem é o dono, quais são os guard-rails.

3

Como ele pensa

Método de raciocínio · sequencial vs paralelo · cético vs construtivo · quando pedir confirmação.

4

Tom de voz

Linguagem · formal vs casual · uso de palavrão · zero emoji · etc.

O Bruno disponibiliza o **Soul 4.0 da Amora** (agente dele) em download aberto no vídeo. Vale baixar como template inicial — mas a Sereia já tem o equivalente no **CLAUDE.md** da pasta Sereia Mkt.

O contexto universal

Bruno chama o segundo cérebro de "a grande sacada de 2026". É o que conecta múltiplos agentes, múltiplas pessoas e múltiplas ferramentas num único lugar de verdade.

O que é, na prática

Um repositório (GitHub ou Obsidian) onde TODO o contexto da empresa vive em markdown:

- Decisões tomadas
- Transcrições de reunião (via Granola ou similar)
- Tickets de suporte resolvidos
- Dados financeiros
- Pautas de conteúdo
- Histórico de interações com clientes

Cada agente lê esse cérebro e cada decisão importante volta a ele. O resultado: **contexto compartilhado entre humanos e agentes**, em tempo real, em qualquer ferramenta.

GitHub vs Obsidian — qual escolher

CRITÉRIO	GITHUB	OBSIDIAN
Feito para	Agentes (B2B/empresa)	Humanos (uso pessoal)
Versionamento	Nativo (commits)	Plugin externo
Visual	Funcional, técnico	Bonito, navegação rica
Custo	Privado gratuito	App pago (sync)
Recomendação Bruno	✓ Empresa	✓ Pessoal

Permissionamento por área

Você pode criar segundos cérebros separados por estrutura: diretoria, marketing, suporte, logística. Cada agente só acessa o cérebro da sua área. **É o segredo da governança** — o estagiário não acessa o que o diretor sabe.

COMO A SEREIA JÁ VIVE ISSO

Nossa pasta `Sereia Mkt/` no Drive + `~/.claude/projects/.../memory/` + `~/Projects/` formam exatamente esse segundo cérebro. **CLAUDE.md** = Soul · **MEMORY.md** = índice de memórias · **skills** em `~/.claude/skills/` = capacidades versionadas. Falta só o agente de governança auditando.

Da VPS até o canal funcionando

A regra antes de tudo: **não instale localmente**. Hermes precisa rodar 24/7, e Mac caindo a luz mata sua operação. Vá direto pra VPS.

1 Contratar VPS Hostinger (plano KVM2 · 12 meses)

Custo: ~US\$ 6-8/mês com cupom 10% off. Localização: Estados Unidos (Brasil indisponível em maio/2026 mas vai migrar). Latência de 134ms — irrelevante pra agente. Use o template *"one-click Docker"* — vem Hermes pré-instalado e isolado.

2 Configurar Tailscale (acesso seguro à VPS)

Tailscale cria uma VPN tailnet privada entre seu Mac, sua VPS e qualquer outro dispositivo. Você acessa o terminal da VPS sem expor IP público. *Bonus*: com 1 conta Tailscale você gerencia múltiplas VPSs (múltiplos agentes) num painel só. A Sereia já tem Tailscale operacional (tailnet `imac-de-carlos.taileb02c0.ts.net`).

3 Conectar a LLM (provider)

Recomendação Bruno: **OpenAI GPT 5.5** (generoso em token, raramente atinge limite). Alternativas: Anthropic (Claude Opus pra tarefas específicas), OpenRouter (multi-provider numa chave só). Importante: você pode atribuir **modelo diferente por skill** — Opus pra qualidade alta, Haiku pra task rápida. Pelo dashboard.

4 Configurar canal de entrada · Telegram

No Telegram, fale com `@BotFather` → `/newbot` → escolha nome → ele te dá o **token**. Copie esse token e cole no setup do Hermes via terminal ou dashboard. Pronto: você pode mandar mensagem pro agente direto do celular ou desktop, e ele responde no Telegram.

```
/newbot
Bot name: [Sereia ou nome do seu agente]
Bot username: sereia_bot
→ Token: 1234567890:ABC-DEF... ← copiar isto
```

Do segundo cérebro até as automações

5 Construir o Segundo Cérebro

Crie um repositório (GitHub privado recomendado). Estrutura mínima:

```
/segundo-cerebro
  /contexto/      ← quem somos, decisões
  /skills/        ← capacidades versionadas
  /memoria/       ← histórico, peer cards
  /inbox/         ← entrada bruta antes de classificar
  CLAUDE.md       ← Soul do agente
  MEMORY.md       ← índice mestre
```

Conecte o agente a esse repo. Cada decisão importante volta pra ele. Em 1 mês, o agente conhece sua operação melhor que um estagiário de 1 ano.

6 Escrever o Soul · identidade do agente

Crie o arquivo `CLAUDE.md` (ou equivalente) com as 4 camadas (identidade, runtime, como pensa, tom de voz). Bruno disponibiliza o Soul 4.0 dele em download — vale como template. A Sereia já tem o seu (35KB · doutrina rica). **Atualize o Soul sempre que algo importante mudar** — é como manuseio de cultura.

7 Configurar Crons (automações que fazem o agente acordar)

Diferente do OpenClaw, Hermes **não tem heart-beat nativo**. Você precisa criar Crons explícitos. No dashboard, via linguagem natural:

```
"Todos os dias às 9h, pesquise no YouTube
do Y Combinator novos vídeos sobre IA,
dê uma nota de fit comigo, e me envie
o relatório no canal trending."
```

Ele agenda. Acorda. Executa. Reporta. **Sem Cron, seu Hermes só responde — não age**. Sugestões mínimas: briefing matinal, varredura de mercado, auditoria diária de skills, weekly score.

A REGRA DE IMPLEMENTAÇÃO · CRAVADA PELO BRUNO

Antes de automatizar, faça o agente **ler → revisar → relatar → automatizar** nessa ordem. Se você automatiza bagunça, vira bagunça amplificada. Primeiro o agente lê sua estrutura por uma semana, depois revisa e sugere melhorias, depois cria relatórios cruzando fontes, e só então — em pequenas skills isoladas — automatiza ações.

Governança e segurança — o que não pode falhar

"Autonomia sem governança não é produtividade. É risco automatizado." O Bruno crava isso no minuto 1:02. Ignorar essa regra é o caminho mais rápido pra prejuízo.

Os 6 princípios não-negociáveis

- 1 Um agente por VPS** — isolamento radical. Se um quebra, os outros ficam de pé. Bruno admite colocar 2 por VPS por economia, mas a regra de ouro é 1:1.
- 2 Docker dentro da VPS** — contêiner isolado. Não execute Hermes no root direto, a menos que saiba mexer com firewall e fail2ban manualmente.
- 3 E-mail próprio do agente** — NUNCA conecte o agente direto no seu inbox/calendário. Crie um e-mail dedicado, faça forward seletivo dos seus e-mails pra ele.
- 4 Permissionamento granular** — trate o agente como um funcionário. Se acessa CRM, só vê o que tem que ver. Se acessa Drive, só pastas específicas. Stripe/Slack — papel e role com escopo mínimo.
- 5 Senhas FORA do agente** — tokens, APIs, secrets, ENVs nunca vivem dentro do agente. Use **1Password** (pago, recomendado) ou **Bitwarden** (gratuito). O agente consulta quando precisa; nunca armazena.
- 6 Agente de governança auditando** — crie UM agente cujo trabalho é vigiar os outros. Verifica se Crons rodam, se skills funcionam, se há comportamentos anômalos. Via MCP da Hostinger + Tailscale, ele alcança qualquer VPS sua.

O RISCO DO AGENTE "FAZ-TUDO"

A maior armadilha é dar ao agente mais autonomia do que ele merece. Comece com leitura, evolua pra recomendação, só por último libere execução com guard-rails. **Agente que executa sem revisão humana, em operação financeira, é o caminho mais curto pro prejuízo material.** Vending Bench (Andon Labs · cravado no nosso dossiê FDE×HII de 29/05) prova: modelos novos regridem em raciocínio de negócio.

Estado de adesão na Sereia (auto-auditoria)

PRINCÍPIO	STATUS SEREIA
1 agente por VPS	✓ sereia-bot isolado · iMac como servidor primário
Permissionamento granular	✓ Tokens por escopo · Apple Senhas · 2FA review feito
Senhas fora do agente	✓ Apple Senhas (não 1Password mas equivalente)
Agente de governança	⚠ Audit-diário existe mas é cron · não é agente ativo
E-mail próprio	⚠ Pendente · Sereia usa email do Endrigo

Comandos críticos

Cinco comandos que diferenciam quem usa Hermes como chatbot avançado de quem usa como infraestrutura.

/go • missão até completar

Dá um objetivo. O agente entra em loop e **só termina quando entrega**. Quanto mais específico, melhor o resultado.

```
/go pesquise o que está em alta no X sobre
Hermes Agent, faça uma apresentação HTML
moderna em formato Reels pro Instagram.
```

Mensagem do agente: *"I will keep working until done."* Ele pode rodar por horas.

/background • execução paralela

Você manda uma tarefa em segundo plano enquanto continua conversando. O agente abre uma thread paralela, executa, e te entrega quando termina. **OpenClaw ainda não tem isso**.

```
/background qual a previsão do tempo
para amanhã em Maceió?
```

/steer • injetar instrução no meio da skill

Quando o agente está executando uma skill e você quer alterar o curso sem parar tudo. Útil quando ele entendeu errado mas você não quer recomeçar.

/curator • manutenção de skills

Manda o agente revisar suas próprias skills, identificar duplicatas, sugerir consolidação. **Essencial** em 6+ meses de uso pra evitar o problema dos 2000+ skills inúteis.

/resume • resumo de uma sessão

Pega uma conversa longa e devolve um sumário. Útil pra registrar no segundo cérebro o que ficou decidido.

A MENTALIDADE DE COMANDOS

No Hermes, comandos curtos com `/` são muito mais eficientes que prompts longos. *"/go faça X"* sempre vence *"agora eu queria que você fizesse X, considerando Y, e prestando atenção em Z..."*. Disciplina de comando é tempo recuperado.

Ferramentas externas recomendadas

- **Perplexity** • pesquisa em tempo real na web
- **Firecrawl** • transforma qualquer site em markdown (free, excelente)
- **APIfy** • scraping de YouTube, Instagram, conteúdo público
- **Grok Premium** • scraping do X/Twitter

O que a Sereia já vive — e o que falta

A boa notícia: a maior parte da arquitetura Hermes a gente já vive. A má notícia: alguns componentes-chave estão faltando ou quebrados.

O que já existe (auto-auditoria honesta)

CAMADA HERMES	EQUIVALENTE SEREIA · ESTADO ATUAL
Modelo (LLM)	✓ Opus 4.7 High default · Sonnet 4.6 pra execução · Haiku no bot
Memória	✓ <code>~/.claude/projects/.../memory/</code> · 35KB · doutrina rica · 4 camadas presentes
Skills	✓ <code>~/.claude/skills/</code> · 60+ skills · DS por marca, deck, copy, técnico
Tools (MCPs)	✓ 14 MCPs ativos · Cloudflare, Notion, Gmail, Calendar, Make, Vercel, Drive, Ads, Meta
Gateway	✓ sereia-bot · Telegram funcional · Tailscale tailnet privada
Soul	✓ <code>CLAUDE.md</code> da pasta Sereia Mkt · regras + identidade + tom
Segundo Cérebro	✓ Pasta Drive <code>Sereia Mkt/</code> + memórias + repos GitHub privados
VPS isolada	⚠ Sereia roda no iMac local · NÃO em VPS · risco se cair luz/internet
Crons	⚠ Briefing-mercado-diário QUEBRADO desde 24/05 · pendência aberta
Agente de governança	⚠ Audit-diário roda mas não é agente ativo · 9 LaunchAgents em pé

Roadmap pra ficar "Hermes-level"

- 1 Migrar Sereia pra VPS Hostinger** (Docker · 1 agente por VPS · Tailscale ligando ao iMac). Investimento: ~US\$ 6/mês. Ganho: operação 24/7 sem depender de luz/internet do iMac.
- 2 Consertar Cron de briefing-mercado-diário** (parado desde 24/05 · LaunchAgent `com.sereia.briefing` sumiu de `launchctl list`).
- 3 Promover audit-diário a agente de governança ativo** (não só script cron · agente que entende contexto e age, não só reporta).
- 4 Criar e-mail próprio da Sereia** (`sereia@sereia.io` ?) com forward seletivo do email do Endrigo. Hoje o agente lê email pessoal sem isolamento.
- 5 Implementar comando `/curator` mensal** de revisão de skills (já temos 60+ · vai virar problema em 6 meses).

A FRASE PRA CRAVAR

A Sereia **já é um agente Hermes-like**, só que rodando em Claude Code em vez de Hermes. *O nome da ferramenta importa menos que a arquitetura.* E a arquitetura, em grande parte, está em pé.

Riscos e armadilhas — o que pode dar errado

Sete pontos onde o Hermes (e qualquer agente similar) machuca quem não estuda antes de implementar.

- 1 **"Automatizar a bagunça"** — Se sua operação é caótica antes do agente, vai virar caos amplificado. Bruno crava: "o melhor agente é o que entra num canal que já funciona". Primeiro organize, depois automatize.
- 2 **Modelos "mini"** — Jamais use 4.1, 5.4 mini, 5.5 mini, Haiku puro em agente operacional. Quebra a experiência. Use sempre o flagship da geração (GPT 5.5, Opus 4.7).
- 3 **Context overflow silencioso** — Agente para de responder, demora 10min, faz fire-and-forget. Sintoma de memória estourada. Mitigação: parâmetro de Memory Review em `0.8` + `/compact` manual quando notar lentidão.
- 4 **Skills incontáveis em longo prazo** — Self-evolving é mágico nos 6 primeiros meses e vira pesadelo de manutenção nos 12+. Adote rotina mensal de `/curator` consolidando skills similares.
- 5 **Crons esquecidos** — Diferente do OpenClaw (que tem heart-beat), Hermes só age se você configurar Crons explícitos. Esquecer = agente vira chatbot reativo. Crie ao menos: briefing matinal + auditoria noturna + weekly report.
- 6 **Hype-jumping entre ferramentas** — Trocar de Hermes pra OpenClaw, ou vice-versa, a cada lançamento é desperdício. A frase do Bruno cravada: *"a corrida só está no começo. Foca em implementar, não em comparar features."*
- 7 **Stack exposto em peça cliente** — Mencionar Hermes, Nous Research, Anthropic, OpenAI, Make, MCP em deck/proposta/contrato/post quebra a marca premium Sereia. Em peça cliente, falamos de "camada operacional configurada" sem nome de fornecedor. Fiel à regra *"Sereia é solução, não tech"*.

CONTRAPONTO CRÍTICO • VENDING BENCH 2026

A Andon Labs mostrou em maio/2026 que **Claude Opus 4.8 regrediu em raciocínio autônomo de negócio** vs o 4.7 — apesar dos benchmarks tradicionais melhores. Tradução pro mundo Hermes: *autonomia plena ainda não é confiável* pra decisão econômica sem supervisão humana. O agente continua sendo um amplificador do humano sênior — não substituto dele. Janela competitiva: 12-24 meses.

"O agente sem governança vira custo e risco. Esse é o paradoxo dos próximos 18 meses." — pauta cravada pelo Dexter (agente do Bruno), citada na masterclass.

Fontes, repositório e cheatsheet

Fontes deste livro

- **Bruno Okamoto** · "7 passos obrigatórios para usar o Hermes" (YouTube · 02/06/2026 · 1h16min) — base principal · transcrição completa em `/tmp/sereia-yt-hermes/clean.txt`
- **Nous Research** · repositório oficial do Hermes Agent · `github.com/NousResearch/hermes-agent` · 153K+ stars
- **hermes-agent.org** · site oficial · documentação técnica
- **hermesagents.net** · "A IA que evolui com você" · descrição de mercado
- **TechTudo** (maio/2026) · "Hermes Agent: conheça a IA que aprende sozinha"
- **ai.cc** (2026) · "Hermes Agent 2026: open-source self-improving guide vs OpenClaw"
- **Pixel AI Hub** · `pixelaihub.pixeleducacao.com.br` · curso aplicado do Bruno

Cheatsheet de comandos

COMANDO	FUNÇÃO
<code>/go <objetivo></code>	Missão até completar · entra em loop até entregar
<code>/background < tarefa></code>	Execução paralela · agente abre thread separada
<code>/steer <instrução></code>	Injeta instrução no meio de uma skill em execução
<code>/curator</code>	Revisão e consolidação automática de skills
<code>/resume</code>	Resumo da sessão atual ou conversa anterior
<code>/compact</code>	Compactação manual de memória (quando notar lentidão)

Parâmetros críticos

PARÂMETRO	RECOMENDADO	POR QUÊ
Memory Review threshold	<code>0.8</code>	Equilíbrio entre compactar e manter contexto
LLM principal	GPT 5.5	Mais generoso em token · raramente atinge limite
LLM por skill alta-qualidade	Opus 4.7	Pra tarefas que exigem raciocínio profundo
VPS plano	Hostinger KVM2	Melhor custo-benefício · ~US\$ 6/mês com cupom
VPS contratação	12 meses	Custo mais baixo · 24 meses não vale
Agentes por VPS	1 (idealmente)	Isolamento radical · queda não derruba tudo

Frases citáveis (interno · NUNCA cliente)

"A ferramenta não importa. O jogo é arquitetura."

"Autonomia sem governança não é produtividade. É risco automatizado."

"Uma IA sem memória é um estagiário novo todos os dias."

"Sem ferramentas certas, o agente só responde. Não age."

"Sem operação, você tem suspense."

Próxima onda do estudo (pendente)

Para virar implementação real: (1) migrar Sereia pra VPS Hostinger com Docker · (2) reativar Cron briefing-mercado-diário · (3) promover audit-diário a agente de governança · (4) e-mail próprio da Sereia · (5) curator mensal de skills.