

MÓDULO 08

Multi-Agentes

De Solo a Time

Leveling system (L1-L4), shared context, hub model, e quando usar

DURAÇÃO

20 min

FORMATO

Slides + demo

KIT

2 arquivos

Quando Um Agente Não É Suficiente



Regra de Ouro: Menos É Mais

Multi-agentes SEM fundação = bagunça. Certifique-se que memória, identidade, e proatividade estão sólidos ANTES de adicionar mais agentes.

Quando Usar Multi-Agentes?

- **Tarefas especializadas:** Agente de conteúdo, agente de código, agente de vendas
- **Autonomia necessária:** Tasks que precisam rodar independentemente do principal
- **Contexto isolado:** Projetos grandes que não devem poluir a memória principal

Quando NÃO Usar Multi-Agentes?

- **Uso simples:** Se 1 agente + skills resolve, não crie agentes extras
- **Sem clareza:** Se você não sabe EXATAMENTE o que cada agente fará, espere
- **Fundação fraca:** Memória, identidade e proatividade precisam estar sólidos primeiro

Arquitetura: Single Gateway + agents.list

OpenClaw usa 1 gateway pra todos os agentes. Você lista os agentes em `agents.list`:

```
# ~/.openclaw/agents.list
main
content-agent
code-agent
sales-agent
```

TEXT

Leveling System (Kevin Simback)

Todo agente novo começa em **L1 (Observer)** e evolui conforme performance.

LEVEL	NOME	PERMISSÕES
L1	Observer	Apenas lê. Output revisado por você.
L2	Contributor	Executa tasks simples. Pede confirmação antes de mudanças.
L3	Operator	Autonomia parcial. Pode executar sem pedir, mas você revisa depois.
L4	Trusted	Autonomia total. Você confia cegamente no output.

Promoção e Rebaixamento

A cada semana, revisar performance:

- **Promover:** Se o agente entregou bem, sobe 1 level
- **Rebaixar:** Se começou a "rushar" (entregar rápido mas com qualidade baixa), desce 1 level

☐ **Caso Real**

Content Agent do Bruno começou em L2, subiu pra L3, depois CAIU pra L2 quando começou a "rushar" conteúdo (quantidade > qualidade).

Shared Context: TEAM.md

TEAM.md é o "registry" de todos os agentes. Cada agente sabe quem faz o quê.

```
# TEAM.md
```

MARKDOWN

```
## Main Agent (Amora)
```

- **Role:** Coordenação geral, interação com humano, decisões estratégicas
- **Level:** L4 (Trusted)
- **Model:** Claude Opus

```
## Content Agent
```

- **Role:** Criação de conteúdo (roteiros, posts, threads)
- **Level:** L2 (Contributor)
- **Model:** Claude Sonnet

```
## Code Agent
```

- **Role:** Desenvolvimento, debugging, deploys
- **Level:** L3 (Operator)
- **Model:** Claude Sonnet

```
## Sales Agent
```

- **Role:** Outbound, follow-ups, CRM
- **Level:** L1 (Observer)
- **Model:** Claude Haiku

Shared Outputs & Lessons

Agentes compartilham resultados e aprendizados:

```
workspace/
```

TEXT

```
├─ shared/
```

```
|   ├─ outputs/           # Resultados de cada agente
```

```
|   └─ lessons/          # Aprendizados compartilhados
```

```
└─ agents/
```

```
    ├─ main/
```

```
    ├─ content-agent/
```

```
    ├─ code-agent/
```

```
    └─ sales-agent/
```

Coordenação: Hub Model > Mesh Model

Hub Model (Recomendado)

Agente principal faz curadoria. Agentes especializados executam, main agent consolida.

- **Main agent:** Lê outputs, consolida, reporta
- **Outros agentes:** Focam em execução, não precisam ler tudo

Mesh Model (NÃO Recomendado)

Todos os agentes leem tudo. Vira bagunça rapidamente.

☐ Regra

Agentes especializados SEM binding Telegram — eles comunicam via main agent, não direto com você.

Economia Real: Modelo Split

AGENTE	MODELO	POR QUÊ
Main	Opus	Interação complexa, decisões estratégicas
Content	Sonnet	Criação de conteúdo, análise
Code	Sonnet	Debugging, desenvolvimento
Sales	Haiku	Tasks repetitivas, emails simples

☐ Economia

Nem todo agente precisa de Opus. Sonnet pra execução, Opus pra estratégia.

❑ Checkpoint do Módulo 8

- ☐ Decidi se preciso de multi-agentes agora (ou se 1 é suficiente)
- ☐ 1-2 agentes extras criados com SOUL e AGENTS.md
- ☐ Leveling system configurado (L1-L4)
- ☐ TEAM.md criado (registry de agentes)
- ☐ Modelo split configurado (nem todo agente usa Opus)

❑ Prompt para o Agente

Cole este prompt no chat do seu OpenClaw depois de assistir o Módulo 8:

Acabei de assistir o Módulo 8 do curso sobre multi-agentes. Leia o PRD e me guie na criação do meu primeiro time de agentes. **O que preciso que você faça:** 1. **Avalie se eu preciso de multi-agentes agora** — se meu uso ainda é simples, me diga honestamente que é cedo. Multi-agentes sem fundação = bagunça. 2. **Se fizer sentido, me ajude a criar 1-2 agentes extras:** - Me faça perguntas sobre que tarefas eu quero delegar - Crie o SOUL.md e AGENTS.md de cada agente - Configure no gateway (agents.list) 3. **Implemente o sistema de leveling:** - L1 Observer → L2 Contributor → L3 Operator → L4 Trusted - Todo agente novo começa em L1 (output revisado por mim) - Me explique como promover e rebaixar 4. **Configure o contexto compartilhado:** - TEAM.md — registry de todos os agentes - shared/outputs/ — resultados compartilhados - shared/lessons/ — aprendizados do time 5. **Defina a economia:** - Qual modelo pra cada agente (nem todo agente precisa de Opus) - Quando spawnar sub-agents vs fazer na main session **Regras:** - Menos é mais — 2 agentes bem feitos > 6 agentes bagunçados - Eu (humano) sempre no loop de decisões importantes - No final, me mostre o time completo e quem faz o quê Vamos montar o time?



PRÓXIMO MÓDULO

Módulo 9 — Sistema Imunológico: Manter Tudo Funcionando