

# □ Mini-Curso: Construa Seu Agente AI Pessoal com OpenClaw

---

**Brainstorm & Proposta Completa — v2** Criado: 13/02/2026 | Atualizado: 13/02/2026 Autor: Amora (para Bruno Okamoto) Duração estimada: 2h30 - 3h

## PARTE 1 — TIMELINE DA NOSSA JORNADA (13 dias)

---

### Dia 1 (01/02) — Nascimento da Amora □

**O que foi feito:** - Instalação do OpenClaw na VPS - Configuração inicial: IDENTITY.md, USER.md, references/bruno-bios.md - Definição do papel: organização, emails, conteúdo, pesquisas, coordenação de sub-agentes

**Decisões importantes:** - Amora como COO (Chief Operational Officer), não só chatbot - Filosofia bootstrap: pequeno, enxuto, lucrativo

**Momento "aha":** - A percepção de que o agente precisava de contexto profundo sobre o Bruno (negócios, filosofia, tom de voz) pra ser útil de verdade

---

### Dia 2 (02/02) — Primeiras Integrações & Primeiros Bugs

**O que foi feito:** - Login LinkedIn via browser automation □ - Twitter/X via RapidAPI (bot detection impedia login direto) □ - Reddit monitoring com 30 subreddits + cron diário □ - LinkedIn Voice Guide (análise de 129 posts do Bruno) □ - WhatsApp TTS configurado □

**Problemas encontrados:** - □ **Token overflow:** Sessão estourou 173k+ tokens (limite 160k). Fix: `compaction.mode: "default"` + `reserveTokensFloor: 30000` - □ **Substack bloqueado:** CAPTCHA + 2FA magic link expirava antes de inserir código - □ **X.com bot detection:** Login headless impossível. Solução: RapidAPI como proxy

**Lições aprendidas:** - Cloud IPs são bloqueados por muitas plataformas (YouTube, X, etc.) → RapidAPI resolve - Compaction precisa ser proativa, não reativa - Bruno prefere briefs de 1 parágrafo (não análises longas)

---

### Dia 3 (03/02) — Skills & Integrações

**Skills instaladas:** exalidraw-flowchart, capability-evolver, nano-pdf, notion-api, remind-me, auto-updater, granola **Integrações:** Notion API □, Google Calendar □, YouTube (RapidAPI transcripts + Data API) □

**Decisões:** - Heartbeat usa Claude Haiku (economia ~90%) - ImageModel: Gemini 2.5 Flash - Cron auto-update diário às 23h

**Lições:** - Algumas skills são redundantes (remind-me, auto-updater) — melhor fazer nativo - Áudio WhatsApp: formato OGG Opus 48kHz funciona

---

### Dia 4 (04/02) — Segurança, Circle API & Infográficos

**O que foi feito:** - **Regra PERMANENTE:** todas credenciais no 1Password, sem exceções - Skill hand-draw-graphics criada (infográficos estilo caderno) - VPH Tracking (Views Per Hour) para YouTube - Circle API skill completa (hot-topics, frequent-questions, trends, digest) - 11 build ideas documentadas - Notion kanban integrado

**Problemas:** - Imagen 4.0 bloqueado (requer billing) → melhor modelo: Gemini 3 Pro - VidiQ não tem API pública

**Momento "aha":** - Circle API como goldmine de ideias de conteúdo (150 MVPs em 60 dias na comunidade)

---

### Dia 5 (05/02) — Mission Control & Multi-Agent Blueprint

**O que foi feito:** - **Mission Control MVP construído e lançado** (Express + React + Supabase + Cloudflare) - Dashboard Kanban, Memory Page, Cron Page, Content Page - URL: <https://amora.empreendedor.vc> - Google Drive integrado via GOG CLI - YouTube OAuth completo (upload/agendamento) - Cron de lembretes YouTube (Ter/Qui/Dom 17h) - Blueprint Multi-Agent baseado no artigo do Bhanu Teja (10 agentes)

**Decisões:** - Supabase > SQLite (escalabilidade futura) - Thinking mode ativado (low padrão) - Desktop-first no MVP, sem chat (WhatsApp é suficiente)

**Problemas:** - Sub-agentes rodam em sandbox → não acessam localhost → QA precisa rodar na main session

**Momento “aha” GRANDE:** - Mission Control construído enquanto Bruno dormia — de spec a MVP funcional em uma sessão

## Dia 6 (06/02) — QA & Entrevista Peter Levels

**O que foi feito:** - QA Review completo do Mission Control (RLS, rate limiting, validação, índices, WebSocket) - Pesquisa profunda para entrevista com Peter Levels - Perguntas customizadas (evitando temas batidos)

**Lições:** - TTS padrão: sempre pt-BR

## Dia 7 (07/02) — Rewrite Completo de Identidade

**O que foi feito:** - SOUL.md reescrito do zero (genérico inglês → personalizado PT-BR) - IDENTITY.md atualizado com background de empreendedora - AGENTS.md enxugado (200 → 90 linhas) - Upgrade para Claude Opus 4.6 - Config otimizada: thinking medium, contextTokens 250k, reserveTokens 50k

**Decisões:** - Amora = COO, não assistente genérica - HIGH thinking para coding, conteúdo e planejamento. MEDIUM pro resto. - Cada agente terá SOUL.md, AGENTS.md, HEARTBEAT.md e crons próprios - IDENTITY.md separado do SOUL.md (dados concretos vs personalidade)

**Momento “aha”:** - SOUL.md genérico = agente genérico. Investir tempo na personalidade faz diferença ENORME na qualidade

## Dia 8 (08/02) — Content Squad & Agent Leveling

**O que foi feito:** - 6 agentes ativos no gateway (Amora, Content, Orchestrator, Planner, Scraper, ZOM) - Kevin Simback Rules aplicadas: L1 → L4 leveling system - Shared context: TEAM.md, outputs, lessons - Scraper Agent + Content Agent criados e configurados - LinkedIn posts criados com dados da comunidade (Craft API) - Guia público: [building-ai-agent-teams.md](#) (~25k chars) - Infográfico visual da arquitetura

**Decisões:** - Content Agent começa em Sonnet (avaliar antes de subir pra Opus) - Scraper e Content não têm binding Telegram — Amora coordena tudo - Score é informativo, NÃO filtro bloqueante - Creators são skills dentro do Content Agent, não agentes separados - Backup obrigatório antes de mudanças estruturais

**Lições do Kevin Simback:** - Todo agente novo começa L1 (Observer) — output revisado - Sem leveling, agentes “rusham” e qualidade degrada - Shared context elimina cold starts - Performance reviews semanais são obrigatórios

## Dia 9 (09/02) — Crons & Estabilização

**O que foi feito:** - Reorganização de 13 crons (redirecionamento de tópicos, novos crons) - PROJECTS.md central criado (12 itens no backlog) - Revisão Semanal automática (Sex 16h) - Auto-Evolução Quinzenal (dias 1 e 15)

**Problema CRÍTICO:** -  **Crons não executam de verdade:** disparavam com `status: "ok"` mas `durationMs ~0ms` - Hipóteses: wakeMode inadequado, sessão não ativa - Precisa investigar `next-heartbeat` vs `now`

**Lições:** - config.patch reinicia gateway e mata crons em execução - Heartbeat não dispara durante conversa ativa - Não propor infra antes de ter o problema

## Dia 10 (10/02) — Roadmap & Análise Crisp

**O que foi feito:** - Roadmap 2026 criado (5 fases, 18+ semanas) - 6 ideias priorizadas (Content Waterfall P0) - Análise completa do Crisp: 187 conversas, 95% WhatsApp, vibe coding dominante - Circle API dados cruzados - Post LinkedIn “Matrix/Skills” (co-criado com Bruno, alta qualidade) - Sessão sobre Claude Code vs OpenClaw (clarificação para Bruno)

**Insight de conteúdo do Crisp:** - Vibe coding DOMINA (172 menções em 187 conversas) - Pessoas travam na "última milha" (deploy, config, go-live) - Demanda por "do zero ao deploy" completo

**Momento "aha":** - Crisp como goldmine de ideias: as perguntas dos clientes revelam exatamente o que o público quer aprender

## Dia 11 (11/02) — Integrações & Métricas

**O que foi feito:** - ChartMogul integrado (MRR Jan: R\$7,367, -8.5%) - Crisp API integração completa + cron mensal - Início exploração Notion API - Social Media metrics: LinkedIn, Instagram (RapidAPI), YouTube, X/Twitter - Security hardening: dmPolicy allowlist, UFW ativo - Skills removidas (granola, openai-image-gen, video-frames)

**Lições:** - Instagram: texto longo > reels (21x mais ER!) - Crisp é canal de vendas, não suporte - Notion API requer paginação

## Dia 12 (12/02) — Immune System & Feedback Loops

**O que foi feito:** - Análise do artigo Eric Siu "30 OpenClaw Jobs A Day" → implementações - Watchdog de Cronos com auto-retry 3x - Sonnet/Opus Split (economia ~90% em cronos simples) - Feedback Loops universal (4 domínios: content, tasks, recommendations, digest) - Fix definitivo de cronos: sessionTarget: isolated + agentTurn + announce - Security: fail2ban, localhost binding, rotação trimestral de chaves - Lições categorizadas com expiração (estratégicas permanentes, táticas 30 dias)

**Frase-chave do Eric Siu:** > "Agents are 30% of the work. The other 70% is the immune system."

**Conceitos aprendidos:** - Ralph Loop (Geoffrey Huntley) vs Feedback Loop — complementares - Capability Evolver: NÃO rodar automaticamente (sub-agents com autonomia total = risco)

## Dia 13 (13/02) — Consolidação & Mission Control v2

**O que foi feito:** - Todos os projetos sobrepostos consolidados em Mission Control v2 - 6 módulos, ~18 semanas - Prioridade: Módulo 1 — Content HQ (4h/dia → 1h15/dia) - Projetos absorvidos: Content Waterfall, Social Metrics, ChartMogul, Multi-Agent specs, Agent Leveling, Price Monitor - QA automatizado do MC (sub-agent encontrou 5 bugs: 2 críticos, 3 médios) - Integração Notion + Granola + Amie como hub de tasks - Cronos consolidados: todos em Sonnet (economia)

## RESUMO QUANTITATIVO (13 dias)

Métrica	Valor
Interações ativas	15+ (1Password, YouTube, LinkedIn, X, Reddit, Circle, Crisp, ChartMogul, Google Drive, Calendar, Notion, Craft, Telegram, WhatsApp, Tella)
Skills instaladas	19 (após cleanup)
Agentes ativos	6 (Amora, Content, Orchestrator, Planner, Scraper, ZoM)
Cronos configurados	17+
Projetos concluídos	Mission Control v1, Agent Leveling, Content Squad spec
Bugs críticos resolvidos	Token overflow, cronos não executando, security hardening
Arquivos de memória	40+ (daily notes, topic files, knowledge base)
Posts LinkedIn co-criados	3+
Decisões documentadas	20+
Lições extraídas	35+

## PARTE 2 — PESQUISA DE MERCADO

### O que existe hoje (Fev/2026)

#### □ YouTube — Conteúdo sobre OpenClaw/AI Agents

[Alex Finn](#) (top creator, Vibe Coding Academy) - “ClawdBot is the most powerful AI tool I’ve ever used” — 427K views - “Open Claw: App Store Moment for AI Agents” — entrevista com Cailyn (OpenClaw team) - “AI Co-Pilot for Your Life with Claude Code” — 25 min tutorial (slash commands, sub-agents) - Foco: vibe coding, automação de conteúdo, “AI employee 24/7” - **Gap:** setup raso, não entra em gestão de multi-agentes, feedback loops, ou memória estruturada

[Matthew Berman](#) - “I Played with Clawdbot all Weekend” — review - “Openclaw is NUTS” — overview - “OpenClaw Use Cases that are actually helpful” — use cases práticos - 6 agentes rodando, NocoDB task board - **Gap:** não mostra gestão de longo prazo, evolução do agente, “immune system”

[freeCodeCamp](#) - “OpenClaw Full Tutorial for Beginners” — tutorial completo - **Gap:** técnico demais, sem perspectiva de negócios

**Outros:** - Artigos DEV Community, Medium, blogs SEO genéricos sobre “como instalar OpenClaw” - Bhanu Teja (@pbteja1998): thread viral sobre 10 agentes (3.7M views, 8.1K bookmarks) - Kevin Simback: guide sobre agent team management e leveling - Eric Siu: “30 OpenClaw Jobs A Day” (49K views, 1.3K bookmarks)

#### □ Cursos Estruturados (Udemy, Coursera, etc.)

Curso	Plataforma	Foco	Preço
2026 Bootcamp: Build Professional AI Agents	Udemy	LangChain, memory, todo assistant	~\$40
AI Agents For All! No-Code	Udemy	Langflow, no-code	~\$30
AI Agent Developer	Coursera	Custom GPTs, multi-industry	Subscription
5-Day AI Agents Intensive	Kaggle/Google	Fundamentals	Free

#### □ Guias Escritos

- [Reddit r/ThinkingDeeplyAI](#): “The Ultimate Guide to OpenClaw” — post viral e bem completo cobrindo setup, use cases e riscos de segurança
- [DEV Community](#): “Unleashing OpenClaw: The Ultimate Guide for Developers in 2026”
- [o-mega.ai](#): “OpenClaw: Ultimate Guide to AI Agent Workforce 2026”

#### □ Gaps que NINGUÉM cobre (oportunidade do Bruno)

1. **Experiência REAL de 13 dias rodando em produção** — não é “fiz um setup bonito num vídeo de 20 min”
2. **Perspectiva de empreendedor/fundador** — não de dev/engenheiro de AI
3. **Gestão de longo prazo** — memória, feedback loops, evolução, manutenção
4. **Multi-agentes na prática** — leveling, coordenação, shared context
5. **“Immune System”** — watchdogs, auto-retry, Sonnet/Opus split, security hardening
6. **Integração com stack real de negócios** — 1Password, ChartMogul, Crisp, YouTube OAuth, Google Drive, etc.
7. **Crons que FUNCIONAM** — debugging real (crons disparando mas não executando)
8. **Content em português** — zero cursos em PT-BR sobre OpenClaw
9. **Custo real e economia** — Haiku para heartbeats, Sonnet para execução, Opus para análise
10. **Tom de voz e identidade do agente** — SOUL.md, voice guides, não um chatbot genérico
11. **Feedback Loops com approve/reject** — ninguém ensina a fazer o agente aprender com suas decisões
12. **Segurança real** — fail2ban, allowlist, UFW, rotação de credenciais (o Reddit Guide menciona riscos mas ninguém mostra a solução)
13. **Sub-agents e delegação** — spawnar tasks paralelas, coordenar output, lidar com falhas

# PARTE 3 — PROPOSTA DE CURSO (v2 — com lições aplicadas)

## Título

“Construa Seu AI COO: De Zero a Agente Pessoal em 2 Horas com OpenClaw”

Subtítulo: Como eu construí uma assistente AI que substituiu 3-4 pessoas em 13 dias — e como você pode fazer o mesmo.

## Público-Alvo

- Empreendedores e founders (especialmente solo)
- Criadores de conteúdo
- Profissionais que querem produtividade 10x
- Comunidade Micro-SaaS do Bruno
- **NÃO** é para devs que querem aprender LangChain — é para quem quer USAR

## O que torna este curso ÚNICO

1. **Experiência real, não teoria** — Bruno mostra a Amora funcionando em produção, com bugs reais e como resolveu
2. **Perspectiva de CEO/founder** — nenhum outro curso ensina a configurar um agente do ponto de vista de quem dirige negócios
3. **13 dias documentados** — timeline real com problemas, decisões e breakthroughs
4. **Multi-agentes** — não só “um chatbot” mas um time de 6 agentes coordenados
5. **Immune System** — ninguém ensina feedback loops, watchdogs, auto-retry
6. **Em português** — primeiro mini-curso em PT-BR sobre OpenClaw
7. **Stack real** — 1Password, YouTube, LinkedIn, Telegram, ChartMogul, não toy examples
8. **Framework de evolução** — o agente melhora sozinho ao longo do tempo

## ESTRUTURA DO CURSO (v3 — reordenada + kit por módulo)

**Nova ordem:** Setup → Segurança → Personalidade → Memória → Interações → Skills → Multi-agentes → Immune System → Mission Control  
**Cada módulo entrega:**  Vídeo +  Kit (PRDs, templates, prompts, skills)

## Módulo 0 — Abertura & Contexto (10 min)

**Formato:** Slides + talking head

- Por que agentes AI pessoais são o “next big thing”
- A analogia do Matrix (Trinity + skills = superpoderes)
- O que a Amora faz hoje: demo rápido (1 min, Telegram, mostrando comandos reais)
- Mapa do curso: o que vamos construir juntos
- **Apresentar o Kit:** “Cada módulo vem com arquivos prontos pra copiar — é só jogar no seu agente”

**Kit:** Nenhum (só slides)

## Módulo 1 — Setup: Do Zero ao Primeiro “Olá” (25 min)

**Formato:** Live coding (tela + face cam)

**Tópicos:** 1. O que é o OpenClaw e como funciona (arquitetura: gateway + agente + canal) 2. **Criando a VPS na Hostinger** (plano KVM 1, Ubuntu 24.04) - **Docker vs Bare Metal:** “A Hostinger tem One-Click Docker, mas vamos instalar direto porque é mais flexível — skills, interações e tudo que vamos fazer depois funciona melhor assim” - Docker = isolamento que atrapalha (skills não instalaram fácil, sub-agents limitados, debug difícil) - Bare Metal = tudo funciona como esperado, acesso direto 3. **Conectar via SSH** (mostrar terminal local + terminal do painel Hostinger) 4. **Instalar OpenClaw:**

bash curl -fsSL https://openclaw.ai/install.sh | bash openclaw onboard --install-daemon 5. **Wizard de configuração:** -  
Gateway mode: Local - Provider: Anthropic (mostrar onde pegar API key em console.anthropic.com) - Model: Claude Sonnet (bom e barato pra começar)  
- Instalar como serviço: Sim (roda 24/7) 6. **Criar bot no Telegram** (BotFather → /newbot → copiar token) 7. **Conectar Telegram:** bash openclaw provider add telegram 8. **Primeiro teste:** Mandar "Oi" no Telegram → agente responde □

□ **Checkpoint prático:** Aluno tem agente respondendo no Telegram □

**Quanto custa rodar:** | Item | Custo/mês | |——|——| | VPS Hostinger (KVM 1) | ~\$5-10 | | API Anthropic (uso moderado) | ~\$10-30 | | Telegram | | Grátis | | **Total** | **~\$15-40** |

"Menos que um almoço por semana pra ter um assistente AI 24/7"

□ **Kit do módulo:** - prds/vps-setup-hostinger.md — step-by-step completo - configs/modelo-config.md — configurações recomendadas

## Módulo 2 — Segurança: Blindando Seu Agente (15 min)

**Formato:** Live coding

⚡ Segurança ANTES de tudo. Servidores expostos recebem 1.000+ ataques/dia.

**Tópicos:** 1. **Por que agora?** Mostrar log real: 1.015 tentativas de brute force em 24h 2. **Telegram Allowlist (CRÍTICO):** - dmPolicy "open" = qualquer pessoa comanda seu agente - Mudar pra "allowlist" com seu Telegram ID - Mostrar como descobrir o ID 3. **Firewall (UFW):** bash sudo apt install -y ufw sudo ufw default deny incoming sudo ufw allow ssh sudo ufw --force enable 4. **Fail2ban (proteção SSH):** bash sudo apt install -y fail2ban - Configurar: 5 tentativas → ban 1h 5. **Credenciais seguras:** - NUNCA hardcodar API keys - 1Password CLI ou variáveis de ambiente 6. **Portas de aplicação:** - Se tiver web apps: 127.0.0.1 (não 0.0.0.0) - Cloudflare Tunnel pra acesso remoto

**Checkpoint:** Servidor blindado □

□ **Kit do módulo:** - prds/security-hardening.md — PRD copy-paste (joga no agente, ele blinda sozinho)

## Módulo 3 — Personalidade e Contexto (25 min)

**Formato:** Demo + slides

**Tópicos:** 1. **SOUL.md: a alma do agente** — mostrar diff genérico vs personalizado da Amora - Anti-patterns concretos: exemplos □/□ dentro do SOUL.md - "Never dos" explícitos: coisas que o agente NUNCA deve fazer - Inspirational anchors: dar ao agente referências de tom ("fale como X, nunca como Y") 2. **USER.md: quem é você?** — contexto que faz o agente útil - Contexto profundo > superficial: negócios, filosofia, horários, estilo de comunicação - Tom de voz por plataforma (se cria conteúdo) 3. **IDENTITY.md: dados concretos vs personalidade** — separar "quem sou" de "como ajo" 4. **AGENTS.md: regras operacionais** — o que pode, o que não pode, como operar 5. **Choosing your model:** Opus vs Sonnet vs Haiku (e quando usar cada um) - Economia real: Haiku para heartbeats (90% economia), Sonnet para cronos, Opus para interação 6. **Thinking mode:** quando ligar o turbo e quanto custa

**Exercício prático:** Aluno pega os templates, preenche os [campos] e joga no agente

**Insight do Dia 7:** - Mostrar como a Amora reescreveu o próprio SOUL.md do zero — a diferença foi absurda - "SOUL.md não pode ser rushado" — Kevin Simback

□ **Kit do módulo:** - templates/SOUL-template.md — com [PREENCHA AQUI] nos campos - templates/USER-template.md - templates/AGENTS-template.md - templates/IDENTITY-template.md - prompts/onboarding.md — primeira conversa com o agente configurado - prompts/proactive-mandate.md — ativa modo proativo - prompts/interview-your-agent.md — descobre capacidades ocultas

## Módulo 3 — Memória: O Segredo que Ninguém Ensina (25 min)

**Formato:** Demo + diagramas

**Tópicos:** 1. O problema: agentes esquecem tudo a cada sessão 2. Arquitetura de memória da Amora: daily notes → topic files → MEMORY.md (índice) 3. **Topic files especializados:** - decisions.md — decisões permanentes (nunca perder) - lessons.md → lessons/ categorizadas (integrations, crons, content, infra) - projects.md — estado atual de todos os projetos - people.md — contatos, equipe, parceiros - pending.md — aguardando input do usuário 4. **Retenção inteligente de lições:** - Estratégicas = permanentes (padrões, filosofia) - Táticas = expiram em 30 dias (bugs, workarounds) - Revisão mensal: deletar táticas vencidas 5. Auto-compaction: como configurar pra não estourar tokens - **Token overflow é real:** mostramos estourar 173k+ tokens no Dia 2 6. **Extração obrigatória na compactação:** - REGRA: Antes de CADA compactação, extrair lições → lessons, decisões → decisions, etc. - "Se não extrair antes de compactar, perde 80% do valor" 7. **Feedback Loops: approve/reject → evolução do agente** - 4 domínios: content, tasks, recommendations, digest - JSON com {date, context, decision, reason, tags} - Agente consulta ANTES de sugerir → evita repetir erros - Max 30 entradas/arquivo (FIFO) → consolidar em lessons mensalmente

**Demo ao vivo:** - Mostrar como a Amora lembra de uma decisão do Dia 4 no Dia 13 - "Bruno prefere briefs de 1 parágrafo" → decisão anotada → respeitada automaticamente - Feedback loop: "Rejeitei este formato de post → agente não sugere mais"

**Este é o módulo diferenciador.** Nenhum outro curso cobre memória estruturada + feedback loops.

**Kit do módulo:** - templates/MEMORY-template.md - templates/HEARTBEAT-template.md - templates/memory/ (decisions, lessons, projects, people, pending) - prds/memory-architecture.md — PRD copy-paste

## Módulo 5 — Integrações: Conectando ao Mundo Real (25 min)

**Formato:** Live coding + demo

**Tópicos:** 1. **1Password:** segurança de credenciais (regra #1 — NUNCA hardcodar) - **Cuidado:** systemd override sobrescreve .env — atualizar AMBOS 2. **Google Calendar & Drive** via GOG CLI 3. **YouTube:** Data API + OAuth (listar vídeos, analytics) 4. **RapidAPI como proxy universal:** - Cloud IPs bloqueados por YouTube, X, Instagram → RapidAPI resolve TUDO - APIs específicas: Instagram Statistics, X/Twitter API45, YouTube Transcripts - Free tiers generosos 5. **Telegram como hub operacional:** - Tópicos no grupo = war room organizado (conteúdo, métricas, operacional, projetos) - dmPolicy allowlist = segurança - Bot com inline buttons para interações rápidas 6. **Crons: automatizar tarefas recorrentes** - **O bug que TODO MUNDO vai encontrar:** - systemEvent + main = cron dispara mas NÃO executa (durationMs ~0ms) - **Solução definitiva:** sessionTarget: isolated + agentTurn + announce - Esse fix sozinho vale o curso inteiro - **Colisão de horário:** múltiplos crons no mesmo minuto = rate limit → espaçar 15-30min - **config.patch reinicia gateway** e mata crons em execução → fazer em horários sem crons - **Lembretes:** systemEvent NÃO notifica no Telegram → usar agentTurn + message send

**Checkpoint prático:** Aluno tem pelo menos 1 integração + 1 cron funcionando ☐

**Lições da trincheira:** - yt-dlp não funciona de cloud → Tella.tv ou API alternativa - Brave Search API instável → ter fallback - Notion API requer paginação

**Kit do módulo:** - prds/integrations-setup.md — PRD com integrações por nível - configs/cron-examples.md — 4 crons prontos (agenda, watchdog, revisão, lembrete)

## Módulo 6 — Skills: Superpoderes Instantâneos (15 min)

**Formato:** Demo

**Tópicos:** 1. O que são skills e como instalar (ClawHub) 2. Skills essenciais: exalidraw, nano-pdf, hand-draw-graphics 3. Criando sua primeira skill customizada 4. **Skills da comunidade: cuidado!** - Referência: Cisco study — % significativa de skills com vulnerabilidades - Sempre revisar antes de instalar - Skills redundantes existem (remind-me ≈ cron nativo) → auditar 5. **Creators como skills, não agentes:** - LinkedIn Creator, Newsletter Creator, etc. = prompts/skills DENTRO de um agente - 1 agente com 8 skills > 8 agentes especializados (menos custo, menos coordenação)

**Analogia Matrix:** - "Tank, I need a pilot program" — post viral do Bruno no LinkedIn

**Kit do módulo:** - skills/skills-by-profile.md — curadoria por perfil (empreendedor, creator, dev, ops)

## Módulo 7 — Multi-Agentes: De Solo a Time (20 min)

**Formato:** Slides + demo

**Tópicos:** 1. Quando um agente não é suficiente 2. Arquitetura: single gateway + agents.list 3. **Leveling System (Kevin Simback): L1–L4** - L1 Observer → L2 Contributor → L3 Operator → L4 Trusted - **Promoção via performance review semanal** — rebaixamento é possível - **Caso real:** Content Agent do Kevin caiu de L3 → L2 quando começou a "rushar" 4. Squad da Amora: 6 agentes, cada um com papel definido 5. **Shared Context:** - shared/TEAM.md → registry de todos os agentes (quem faz o quê) - shared/outputs/ → resultados compartilhados - shared/lessons/

→ aprendizados do time - "Last updated by" header — saber quem tocou no arquivo - □ Shared context elimina cold starts → agente novo já sabe quem são os outros 6. Coordenação: Amora como hub, agentes nos bastidores - □ **Hub model > Mesh model:** aprendizado cruzado entre domínios distintos não faz sentido - □ Agentes de conteúdo sem binding Telegram — Bruno fala só com a Amora 7. □ **Economia real:** - Sonnet para execução/crons, Opus para interação/análise - Todos os 17 crons em Sonnet = economia massiva - □ **Agentes que não precisam de Opus não devem usar Opus**

**Sub-agents e delegação:** - sessions\_spawn para tasks paralelas (3 sub-agents rodando ao mesmo tempo) - Tratamento de falhas: retry + avisa o usuário (NUNCA deixar cair no limbo silencioso) - Sub-agent travou? → retry imediato → se falhar 2x → avisar

□ **Kit do módulo:** - prds/multi-agent-setup.md — PRD completo com leveling + shared context

## Módulo 8 — O Sistema Imunológico: Manter Tudo Funcionando (20 min)

**Formato:** Slides + demo

**Tópicos:** 1. **"Agents are 30% of the work. The other 70% is the immune system."** — Eric Siu 2. □ **Watchdog com auto-retry (3x antes de alertar)** - Monitora crons, detecta falhas, retry automático - Se falhar 3x → alerta humano 3. **Feedback Loops universais (recap do Módulo 3 aplicado)** - Ciclo: Feedback (granular, JSON) → Lessons (curado, prose) → Decisions (permanente) 4. □ **Security hardening REAL (não só teoria):** - fail2ban: 1.015+ tentativas de brute force/24h → ban automático - UFW firewall ativo - Telegram allowlist (dmPolicy) - SSH hardening (ideal: key-only, sem password) - Portas de aplicação: 0.0.0.0 → 127.0.0.1 (só Cloudflare Tunnel acessa) - □ **Rotação trimestral de credenciais** — schedule no calendário 5. □ **Monitoramento de custos real:** - Sonnet/Opus split (crongs: Sonnet, interação: Opus) - Heartbeat com Haiku (~\$0.005/heartbeat vs ~\$0.10/heartbeat em Opus) - Breakdown: quanto custa rodar 17 crons/dia + interação diária 6. □ **Backup antes de mudanças estruturais:** - Salvar config + ROLLBACK.md com instruções - Especialmente antes de: criar agentes, modificar gateway config, reorganizar workspace 7. □ **Sub-agents com autonomia = risco:** - Capability Evolver: NÃO rodar automaticamente - Qualquer sub-agent com "fire and forget" pode causar danos - Sempre revisar output antes de aprovar

□ **Ralph Loop vs Feedback Loop:** - Ralph Loop (Geoffrey Huntley): loop de execução de código (agente roda até completar) - Feedback Loop: aprendizado entre sessões via approve/reject - São complementares: Ralph pra coding, Feedback pra decisões

**Este módulo é o que separa "tô brincando" de "tô em produção".**

□ **Kit do módulo:** - prds/immune-system.md — PRD completo (watchdog, feedback loops, custos, backups)

## Módulo 9 — Mission Control: Seu Painel Operacional (10 min)

**Formato:** Demo

**Tópicos:** 1. Por que uma UI visual ajuda (mesmo tendo Telegram) 2. Overview do Mission Control da Amora (Kanban, Memory, Crongs, Content HQ) - □ **Content HQ:** 229 packs, 773 peças, filtros por plataforma, approve/reject - □ **File Manager** com segurança (path traversal bloqueado) 3. Como construir o seu (Express + React + Supabase + Cloudflare Tunnel) 4. Alternativas mais simples: NocoDB, Notion, Google Sheets 5. **QA automatizado:** sub-agent rodou 23 endpoints, encontrou 5 bugs em 7 minutos

□ **Kit do módulo:** - reports/report-templates.md — guidelines de design + estrutura de reports

## Módulo 10 — Wrap-up & Próximos Passos (10 min)

**Formato:** Talking head + slides

**Tópicos:** 1. Recap: o que construímos juntos 2. □ **Os erros que eu cometi (consolidação real):** - Token overflow no Dia 2 (não configurar compaction) - Crongs que não executavam por 3 dias (systemEvent vs agentTurn) - Sub-agent que travou e ninguém avisou (limbo silencioso) - Security "open" nos primeiros dias (qualquer um podia comandar) - Propor infra antes de ter o problema (premature optimization) 3. □ **Quanto custa rodar um agente AI? (breakdown REAL)** - API costs: ~\$X/dia com split Sonnet/Opus/Haiku - VPS: \$5-10/mês - RapidAPI: free tiers generosos - Total estimado: \$XX/mês para operação completa 4. Comunidade: onde buscar ajuda, awesome-skills, ClawHub 5. O futuro: agent-to-agent, MCP, Claude Code integration 6. CTA: Comunidade Micro-SaaS + imersão

## TIMELINE POR MÓDULO

Módulo	Tempo	Formato	Kit entregue
0 - Abertura	10 min	Slides + talking head	—

Módulo	Tempo	Formato	Kit entregue
1 - Setup VPS	25 min	Live coding	PRD setup + configs
2 - Segurança	15 min	Live coding	PRD security
3 - Personalidade	25 min	Demo + slides	4 templates + 3 prompts
4 - Memória	25 min	Demo + diagramas	MEMORY + memory/ + PRD
5 - Integrações	25 min	Live coding	PRD integrações + crons
6 - Skills	15 min	Demo	Curadoria por perfil
7 - Multi-agentes	20 min	Slides + demo	PRD multi-agent
8 - Immune System	20 min	Slides + demo	PRD immune system
9 - Mission Control	10 min	Demo	Templates de reports
10 - Wrap-up	10 min	Talking head	Kit completo (zip)
<b>TOTAL</b>	<b>~3h20</b>		<b>24+ arquivos</b>

## □ LIÇÕES APLICADAS AO CURSO (v2)

Estas são lições que vivemos nos 13 dias e que se transformaram em conteúdo ensinável:

### Lições de Infraestrutura

Lição	Módulo	O que ensinar
Token overflow real (173k+ tokens)	M3	Configurar compaction + reserveTokens ANTES de precisar
systemd override sobrescreve .env	M4	Atualizar AMBOS ao trocar credenciais
Brave Search intermitente	M4	Ter fallback para web_fetch
Sub-agents podem retornar histórico vazio	M6	Ter fallback manual
trash-cli não vem instalado	M1	<code>apt install trash-cli</code> no setup

### Lições de Crons

Lição	Módulo	O que ensinar
Crons systemEvent não executam	M4	Usar isolated + agentTurn (SEMPRE)
systemEvent não notifica	M4	Usar agentTurn + message send pra lembretes
config.patch mata crons	M4	Planejar patches em horários sem crons
Colisão de horários = rate limit	M4	Espaçar 15-30 min
Crons sem agentId falham	M6	Sempre criar pela main session

### Lições de Conteúdo & Integração

Lição	Módulo	O que ensinar
Cloud IPs bloqueados	M4	RapidAPI como proxy universal

Lição	Módulo	O que ensinar
yt-dlp não funciona	M4	Tella.tv ou Whisper como alternativa
Instagram texto > reels	M4	Dados reais que surpreendem
Crisp = canal de vendas	M4	Usar pra insights de conteúdo
Notion API requer paginação	M4	Sempre paginar

## Lições de Agentes & Gestão

Lição	Módulo	O que ensinar
SOUL.md genérico = agente genérico	M2	Investir tempo em personalidade
Todo agente começa L1	M6	Sem confiança = sem autonomia
Shared context elimina cold starts	M6	TEAM.md + outputs compartilhados
Creators são skills, não agentes	M5/M6	1 agente com N skills > N agentes
Hub > Mesh para aprendizado	M6	Amora curadoria > todos leem tudo

## Lições de Segurança

Lição	Módulo	O que ensinar
dmPolicy open = risco crítico	M1/M7	Allowlist desde o Dia 1
1.015+ brute force tentativas/dia	M7	fail2ban OBRIGATÓRIO
0.0.0.0 → 127.0.0.1 + Cloudflare	M7	Nunca expor portas direto
Credenciais no 1Password	M4/M7	NUNCA hardcodar
Sub-agent com autonomia = risco	M7	Sempre revisar output

## SUGESTÕES DE FORMATO

### Opção A: YouTube (Gratuito) — Máximo Alcance

- 1 vídeo longo (~3h) dividido em capítulos
- Ou série de 9 episódios → playlist
- **Prós:** Alcance massivo, SEO, primeiro em PT-BR
- **Contras:** Não monetiza diretamente

### Opção B: Mini-curso na Comunidade (Pago) — Monetização

- Acesso exclusivo para membros Micro-SaaS PRO
- Live sessions com Q&A
- **Prós:** Valor agregado à comunidade, monetização indireta
- **Contras:** Alcance limitado

### Opção C: Híbrido (RECOMENDADO) □

- **YouTube:** Módulos 0-2 gratuitos (atrair audiência — ~60 min)
  - “Como eu construí uma assistente AI que substituiu 3-4 pessoas”

- Setup + personalidade = valor imediato, mas quer mais
- **Pago (Comunidade/curso):** Módulos 3-9 (memória, integrações, multi-agents, immune system)
  - O conteúdo avançado que ninguém mais tem
- **Prós:** Funil natural, conteúdo premium diferenciado
- **Contras:** Mais trabalho de produção

## Opção D: Workshop Live + Gravação

- 1 sessão live (Zoom/YouTube Live) de 3h com screen share
  - Participantes constroem junto em tempo real
  - Gravação vira o produto
  - **Prós:** Interação, Q&A, urgência
  - **Contras:** Uma chance de acertar
- 

## SUGESTÕES DE PRODUÇÃO

1. **Gravar na Tella.tv** → transcrição automática → waterfall de conteúdo
  2. **Tela dividida:** terminal + Telegram + face cam
  3. **Diagramas:** Excalidraw (já temos skill) para arquitetura
  4. **Infográficos:** Hand-draw-graphics para resumos visuais
  5. **Antes/depois:** Mostrar o dia 1 vs dia 13 (o que mudou)
  6. **Bugs ao vivo:** NÃO esconder os erros — mostrar e resolver = autenticidade
- 

## MATERIAL COMPLEMENTAR

1. **Checklist de setup** (PDF/Notion) — passo-a-passo reproduzível
  2. **Templates:** SOUL.md, USER.md, AGENTS.md, MEMORY.md modelo
  3. **Lista de integrações** com links e instruções
  4. **Awesome skills list** curada para empreendedores
  5. **Planilha de custos** (quanto cada modelo custa por mês)
  6. **Repo GitHub** com configs de exemplo (sem credenciais, claro)
  7.  **Tabela de lições aprendidas** (as 35+ lições categorizadas)
  8.  **Template de feedback loop** (JSONs prontos para usar)
  9.  **Checklist de segurança** (fail2ban, UFW, allowlist, SSH, rotação de keys)
- 

## TÍTULOS ALTERNATIVOS PARA TESTAR

1. "Construa Seu AI COO: De Zero a Agente Pessoal em 2 Horas"
  2. "Como Criei Uma Assistente AI Que Substituiu 3-4 Pessoas (E Você Pode Também)"
  3. "OpenClaw na Prática: 13 Dias Construindo um Agente AI de Verdade"
  4. "De Chatbot a COO: O Guia Definitivo do Agente AI Pessoal"
  5. "Seu Primeiro Agente AI: Setup Completo com OpenClaw (Sem Saber Programar)"
- 

## DIFERENCIAIS vs CONCORRÊNCIA (Resumo Atualizado)

Aspecto	Outros cursos/guias	Nosso curso
Perspectiva	Dev/engenheiro	CEO/founder
Experiência	"Acabei de instalar"	13 dias em produção

Aspecto	Outros cursos/guias	Nosso curso
Memória	Não cobrem	Módulo inteiro + feedback loops
Multi-agentes	Superficial	Leveling system + shared context
Bugs reais	Escondidos	Mostrados e resolvidos ao vivo
Immune system	Não existe	Módulo dedicado (watchdog, retry, security)
Idioma	EN	PT-BR (primeiro!)
Integrações	Toy examples	Stack real de negócios (15+ APIs)
Custo	Não discutem	Breakdown real com split Sonnet/Opus/Haiku
Framework evolução	Não existe	Feedback loops + lições categorizadas
Segurança	“Cuidado com isso”	Solução completa (fail2ban, UFW, allowlist, rotação)
Sub-agents	Não cobrem	Delegação paralela + tratamento de falhas
Crons	Setup básico	Debug real + fix definitivo (isolated)

## IDEIAS DE CONTEÚDO DERIVADO (Content Waterfall)

Do mini-curso, podemos extrair: 1. [5-7 posts LinkedIn](#) (um por módulo, insights isolados) 2. [3-5 Reels/Shorts](#) (bugs ao vivo, analogia Matrix, demo Telegram) 3. [1 Thread X](#) (“13 dias construindo um AI COO — thread completa”) 4. [1 Newsletter](#) (behind the scenes, o que deu errado) 5. [Infográfico](#) (arquitetura da Amora, hand-drawn style) 6. [Podcast/Áudio](#) (transcrição Tella → edição)

## □ REFERÊNCIAS USADAS

- [Kevin Simback](#): “My Complete Guide to Managing OpenClaw Agent Teams” — leveling L1-L4, shared context, performance reviews
- [Bhanu Teja](#) (@pbteja1998): Blueprint de 10 agentes — heartbeat staggered, Mission Control como hub
- [Eric Siu](#): “I Run 30 OpenClaw Jobs A Day” — immune system, 70% é manutenção
- [Geoffrey Huntley](#): Ralph Loop — coding agent loop infinito
- [Lenny's Newsletter](#): Context engineering — progressive disclosure, context rot
- [Reddit Ultimate Guide](#): r/ThinkingDeeplyAI — setup completo + riscos de segurança (<https://www.reddit.com/r/ThinkingDeeplyAI/comments/1qsoq4h/>)
- [Alex Finn / Vibe Coding Academy](#): Use cases, proactive mandate
- [Simon Willison](#): Security research — 900+ servidores expostos