

GUIA GRATUITO · MAIS COMPLETO

Construa seu Jarvis Pessoal com IA

Um assistente que conhece sua vida inteira — feito sob medida com Obsidian + Claude Code

- Sistema modular adaptável a qualquer rotina
- Prompt mestre que constrói o vault sozinho
- Memória persistente entre conversas
- Genérico — você customiza pro seu uso

POR

Erick Scur

Engenheiro de Software Sênior · @erickscur

1% Programador

Sumário

O que você vai construir	3
Por que esse sistema funciona	3
Pré-requisitos	4
Conceitos fundamentais	5
Setup passo a passo	7
Anatomia de um módulo	8
Módulos genéricos prontos	9
Criando módulos customizados	12
O Prompt Mestre	13
Manutenção e evolução	22
FAQ — perguntas que todo mundo faz	24
Próximos passos práticos	24
Quer ir mais fundo?	25

O que você vai construir

Imagina abrir o terminal de manhã, digitar `bom dia` e ler:

Exemplo de briefing

“Bom dia. Você tem 3 reuniões hoje, a primeira é com o time de produto às 10h. Última vez que você treinou foi há 3 dias — vai querer reforçar? Seu gasto de abril já passou da meta de lazer em 18%. E aquele projeto que você travou semana passada continua na pendência.”

Não é mágica. Não é um app caro. É um sistema que **você** monta, com duas ferramentas gratuitas (Obsidian + Claude Code), seguindo o modelo desse guia.

O sistema se chama **Nerso** — uma referência direta ao Jarvis do Tony Stark. A ideia é a mesma: uma IA que não é um chatbot genérico. Ela conhece seus projetos, sua rotina, sua saúde, suas finanças, seus objetivos. E age com base nisso.

Esse guia é genérico de propósito

Eu não quero te dar um vault pronto pra copiar. Sistema bom é o que reflete **sua** vida. Por isso esse guia ensina:

- O modelo conceitual (hub central + módulos especializados)
- Os blocos de construção (CLAUDE.md, frontmatter, logs, memória)
- O **Prompt Mestre** que faz o Claude Code te entrevistar e construir tudo do zero
- Templates de módulos genéricos pra você adaptar

Você responde algumas perguntas, escolhe quais áreas da sua vida quer organizar, e o Claude Code monta o vault inicial. Daí em diante, você usa, edita, evolui.

Atenção

Esse é o guia mais técnico que eu tenho gratuito. Se você nunca abriu um terminal na vida, não desista — eu explico cada passo. Mas saiba que vai precisar instalar duas coisas (Obsidian e Claude Code) e copiar/colar comandos.

Por que esse sistema funciona

Sistemas tradicionais de produtividade falham por dois motivos:

1. **Fragmentação:** notas em um app, tarefas em outro, finanças num terceiro. Nada se conversa.
2. **Genericidade:** apps tratam você igual a 10 milhões de outras pessoas. Não conhecem seu contexto.

O Nerso resolve isso com três princípios.

Princípio 1 — Hub central + spokes especializados

Um arquivo `CLAUDE.md` na raiz do vault funciona como **mapa mental** do sistema. Ele define quem você é, sua rotina, e aponta pra sub-`CLAUDE.md` em cada domínio (saúde, finanças, trabalho, etc.). Quando você pergunta algo de saúde, a IA primeiro lê o `CLAUDE.md` de saúde. Quando pergunta de trabalho, lê o de trabalho. Sempre com contexto certo, sem desperdiçar tokens com informação irrelevante.

Princípio 2 — Memória persistente entre sessões

Cada conversa importante gera um log atômico em `Nerso/Logs/`. A IA pode buscar “o que conversamos sobre X semana passada?” sem você precisar lembrar.

Memória de longo prazo de verdade — não são os 200k tokens da janela de contexto que ela esquece quando a sessão fecha.

Princípio 3 — IA com personalidade configurada

Você define o tom. Pai duro? Coach motivacional? Assistente neutro? A IA segue. Você define gatilhos: “quando eu falar X, faça Y”. A IA respeita. Você define o que NÃO fazer. Ela não faz.

O que esse sistema NÃO é

- **Não é um app** — é uma metodologia que você instala usando ferramentas que já existem.
- **Não é um agente autônomo** — ele te orienta e executa quando você pede, não age por conta própria.
- **Não substitui Notion, TickTick ou Calendar** — ele se conecta a esses (via MCP) quando faz sentido.

Pré-requisitos

Item	Onde pegar	Custo
Obsidian	obsidian.md	Grátis
Claude Code	claude.com/code	Free tier ou pago
Conta Anthropic	console.anthropic.com	Grátis pra começar
Markdown básico	10 min de leitura	—

Opcional mas recomendado:

- **Plugin Templater** (Obsidian) — pra templates dinâmicos de notas
- **Plugin Dataview** (Obsidian) — pra dashboards e queries em notas
- **MCPs do Claude Code** (TickTick, Notion, Calendar) — pra integrações depois que o sistema estiver rodando

Dica

Não se preocupe em instalar tudo de uma vez. Comece com o básico (Obsidian + Claude Code), monte o sistema, use por uma semana. Os plugins e MCPs você adiciona depois quando perceber que precisa.

Conceitos fundamentais

Quatro coisas que você precisa entender antes de construir. Sem isso, o resto não faz sentido.

O que é um CLAUDE.md

É um arquivo markdown que dá **instruções permanentes** pra IA. Sempre que você abre o Claude Code num diretório, ele lê automaticamente o `CLAUDE.md` daquele lugar.

Hierarquia de leitura:

1. `~/claude/CLAUDE.md` — instruções globais (todos os projetos)
2. `<vault>/CLAUDE.md` — instruções do vault inteiro (raiz)
3. `<vault>/<modulo>/CLAUDE.md` — instruções específicas daquele módulo

A IA lê de cima pra baixo. Instruções mais específicas sobrescrevem as genéricas. Se o CLAUDE.md raiz diz “responda de forma direta” e o de Saúde diz “use linguagem motivacional”, quando você pergunta algo de saúde, ela usa motivacional.

Hub e módulos (spokes)

```

vault/
├── CLAUDE.md           ← HUB (define quem você é, mapeia tudo)
├── Saude/
│   └── CLAUDE.md      ← módulo (instruções específicas de saúde)
├── Financeiro/
│   └── CLAUDE.md      ← módulo
├── Trabalho/
│   └── CLAUDE.md      ← módulo
└── Nerso/
    ├── CLAUDE.md      ← módulo (sistema de logs)
    └── Logs/           ← memória de sessões
    
```

O CLAUDE.md raiz é o “cérebro central”. Os módulos são especialistas. A IA navega entre eles conforme a pergunta.

Memory system (memória de longo prazo)

O Claude Code tem um sistema de memória nativo em `~/claude/projects/<hash-do-projeto>/memory/`. Ele guarda quatro tipos:

- **User memory** — quem você é (perfil, papel, contexto)
- **Feedback memory** — preferências de comportamento (“não faça X”, “sempre faça Y”)
- **Project memory** — estado de iniciativas em andamento
- **Reference memory** — ponteiros pra sistemas externos

A IA escreve nessa memória **automaticamente** conforme você conversa. Ela é carregada em todas as sessões futuras. Você não precisa fazer nada — só conversar e dar feedback.

Logs de sessão (memória de curto prazo estruturada)

Em `Nerso/Logs/`, cada sessão importante vira um arquivo `YYYY-MM-DD-HHmm.md` com frontmatter rico. Permite buscar conversas antigas com Grep sem ter que reler tudo.

```

---
data: 2026-04-25T14:32
dominio: [trabalho, saude]
    
```

```
acoes: [criou tarefa X, registrou treino Y]
decisoos: [decidiu adiar projeto Z para maio]
---
```

Bônus — frontmatter e backlinks

Frontmatter YAML (bloco no topo da nota): a IA lê pra extrair dados estruturados sem parsear o texto inteiro. Use em logs, registros, métricas.

Backlinks `[[Nota]]`: sintaxe nativa do Obsidian que cria link clicável. **A IA usa backlinks pra navegar entre notas.** Sempre que possível, refira a outras notas via backlink em vez de duplicar conteúdo.

Setup passo a passo

Passo 1 — Instalar e abrir o Obsidian

1. Baixe e instale do site oficial.
2. Crie um novo vault. Nome sugerido: `Life 0.S` ou `<seu-nome> 0.S`.
3. Escolha onde salvar. Recomendo iCloud Drive ou Dropbox pra sincronizar entre dispositivos.

Passo 2 — Instalar o Claude Code

1. Acesse `claude.com/code` e siga as instruções de instalação.
2. Faça login com sua conta Anthropic.
3. No terminal, navegue até o diretório do vault e rode `claude`.

```
cd ~/Documents/Life-0S
claude
```

Passo 3 — Rodar o Prompt Mestre

Vá pra seção “O Prompt Mestre” desse guia, copie o bloco entre as linhas `===`, e cole no Claude Code aberto na pasta do vault.

A IA vai te entrevistar (uma pergunta por vez, sem formulário gigante) e construir a estrutura inicial baseada nas suas respostas.

Passo 4 — Validar

Depois que o prompt rodar, você terá:

- `CLAUDE.md` na raiz (hub central com quem você é, tom, regras)
- Pastas dos módulos que você escolheu, cada uma com seu `CLAUDE.md`
- Pasta `Nerxo/Logs/` pronta pra receber logs de sessão
- Templates básicos em `Recursos/Templates/`

Teste com: `bom dia, me dá um briefing`. A IA deve responder com base no que você configurou.

Passo 5 — Integrações (opcional)

Se quiser conectar a TickTick, Notion, Google Calendar, Linear, etc., instale os MCPs correspondentes via `/mcp` no Claude Code. Cada integração expande o que a IA pode fazer (criar tarefas, ler eventos do calendário, registrar no Notion).

Dica

Não plugue tudo de uma vez. Use o sistema base por 1-2 semanas primeiro. Aí você vai perceber qual integração faz sentido pra **seu** fluxo.

Anatomia de um módulo

Todo módulo segue um padrão simples. Vamos pegar **Financeiro** como exemplo:

```

Financeiro/
├─ CLAUDE.md           ← instruções da IA pra esse módulo
├─ Dashboard.md       ← visão geral (queries Dataview opcional)
├─ Meses/
│   └─ 04-26.md       ← lançamentos de abril/2026
│       └─ 03-26.md
├─ Custos.md          ← lista de despesas fixas
└─ Metas.md           ← objetivos financeiros
    
```

O que vai dentro do CLAUDE.md de um módulo

```
# Financeiro – CLAUDE.md

## Contexto
Módulo de controle financeiro pessoal.
Renda mensal ~R$X, custo fixo ~R$Y.

## Estrutura
- `Meses/MM-AA.md` – lançamentos do mês
- `Custos.md` – despesas fixas recorrentes
- `Metas.md` – objetivos (reserva, investimento, compra)

## Como agir
Quando o usuário registrar gasto:
1. Identificar o mês corrente
2. Adicionar linha em `Meses/MM-AA.md` no formato definido
3. Atualizar total

## Regras
- Sempre converter valores para BRL
- Não criar nota nova se já existe a do mês
- Frontmatter obrigatório: total, categoria_principal
```

Padrão geral de qualquer módulo

Todo CLAUDE.md de módulo tem 4 blocos:

1. **Contexto** — pra que esse módulo existe na sua vida
2. **Estrutura** — quais arquivos/pastas a IA pode esperar
3. **Como agir** — passo a passo do que fazer quando você pedir algo
4. **Regras** — restrições, formatos obrigatórios, gatilhos

Mantenha cada CLAUDE.md de módulo **enxuto**. Máximo 100 linhas. Se passar, simplifique. Instrução demais confunde a IA tanto quanto confunde gente.

Módulos genéricos prontos

Os módulos abaixo cobrem 80% das necessidades de qualquer pessoa. Use como ponto de partida — o Prompt Mestre vai te perguntar quais você quer e cria só esses, sem inflar o sistema.

Financeiro

Pra que serve: rastrear receitas, despesas, metas e investimentos.

Estrutura sugerida:

```
Financeiro/  
├─ CLAUDE.md  
├─ Dashboard.md  
├─ Meses/MM-AA.md  
├─ Custos.md           (gastos fixos)  
└─ Metas.md           (reserva, investimentos)
```

Quando usar: sempre que precisar registrar entrada/saída de dinheiro, ver onde está gastando, planejar grandes compras.

Saúde

Pra que serve: consolidar treino, alimentação, sono, exames, peso.

Estrutura sugerida:

```
Saude/  
├─ CLAUDE.md  
├─ Dashboard.md  
├─ Treinos/YYYY-MM-DD.md  
├─ Alimentação/Logs/YYYY-MM-DD.md  
├─ Exames/  
└─ Métricas.md       (peso, sono, etc.)
```

Quando usar: registrar treino do dia, log de refeição, resultado de exame, evolução de peso.

Trabalho

Pra que serve: projetos profissionais, metas anuais, deadlines, reuniões.

Estrutura sugerida:

```
Trabalho/  
├─ CLAUDE.md  
├─ Projetos/  
│   └─ <nome-do-projeto>/  
├─ Metas <Ano>.md  
└─ Reuniões/YYYY-MM-DD.md
```

Quando usar: documentar reunião, registrar decisão técnica, preparar 1:1 com gestor, atualizar status de projeto.

Rotina e Hábitos

Pra que serve: definir o dia ideal, rastrear hábitos, integrar com app de tarefas.

Estrutura sugerida:

```

Rotina/
├─ CLAUDE.md      (rotina ideal, regras, integração com TickTick/ToDoist)
├─ Hábitos.md     (lista de hábitos rastreados)
└─ Protocolos/   (planos semanais, reviews)
  
```

Quando usar: “o que tenho hoje?”, “organiza minha semana”, “criar tarefa”, “lembrar de X”.

Journal

Pra que serve: daily notes, weekly reviews, registro pessoal.

Estrutura sugerida:

```

Journal/
├─ CLAUDE.md
├─ Daily/YYYY-MM-DD.md
└─ Weekly/YYYY-WWW.md
  
```

Quando usar: registrar como foi o dia, weekly review, tracking emocional.

Conhecimento

Pra que serve: notas de livros, cursos, artigos, ideias.

Estrutura sugerida (versão simplificada de Zettelkasten):

```

Conhecimento/
├─ CLAUDE.md
├─ Livros/
├─ Cursos/
├─ Artigos/
└─ Ideias/      (notas livres)
  
```

Quando usar: “criar nota sobre o livro X”, “salvar este insight”, “o que eu sei sobre Y?”.

Nerso (sistema de logs) — obrigatório

Pra que serve: memória entre sessões. **Esse módulo é o coração do sistema.** Sem ele, a IA esquece tudo entre conversas e perde contexto.

Estrutura:

```
Nerso/
├── CLAUDE.md          (regras do sistema de logs)
└── Logs/
    └── YYYY-MM-DD-HHmm.md
```

Cada log tem 200 tokens, frontmatter estruturado, e seções S/K/P/C (Summary, Key items, Pending, Context). O Prompt Mestre cria isso automaticamente.

Atenção

Independente de quais outros módulos você escolher, o módulo Nerso/Logs sempre vai ser criado. É a fundação do sistema.

Criando módulos customizados

Quando criar um módulo novo? Quando você tem **um domínio recorrente com regras próprias**. Exemplos do mundo real:

- **Idiomas** — se você estuda outra língua
- **Hobby** (perfumes, vinhos, filmes, livros) — se você coleciona ou registra
- **Negócios / Side project** — se você empreende
- **Relacionamentos** — se você quer rastrear contatos importantes
- **Viagens** — se você viaja com frequência
- **Conteúdo** — se você cria pra redes sociais e quer organizar ideias, roteiros, métricas

Receita pra criar um módulo

1. Crie a pasta: `<vault>/<NomeDoModulo>/`

2. Crie o `CLAUDE.md` seguindo esse template:

```
# <NomeDoModulo> - CLAUDE.md

## Contexto
[1-2 frases: pra que esse módulo existe na sua vida]
```

```

## Estrutura
[Lista de pastas/arquivos esperados]

## Como agir
[Quando o usuário disser X, faça Y]

## Regras
[Restrições, formatos obrigatórios, gatilhos]

## Templates (opcional)
[Estruturas de notas reutilizáveis]
    
```

3. Adicione referência no CLAUDE.md raiz na tabela “Como Navegar”:

```

| O usuário quer...      | Leia primeiro...      |
|-----|-----|
| Algo do módulo X      | [[NomeDoModulo/CLAUDE]] |
    
```

4. Teste: abra uma nova sessão e pergunte algo do domínio. A IA deve consultar o CLAUDE.md do módulo automaticamente.

Dica

Atalho: peça pro próprio Claude Code criar o módulo pra você. “Quero um módulo novo de Idiomas pra rastrear estudo de inglês. Crie a estrutura seguindo o padrão dos outros módulos.” Ele lê o que já existe, cria seguindo o mesmo modelo, e atualiza o CLAUDE.md raiz.

O Prompt Mestre

Esse é o coração do guia. Cole esse prompt no Claude Code aberto na pasta do seu vault novo (vazio). Ele vai te entrevistar e construir tudo.

Importante

Como usar: abra o terminal na pasta do vault, rode `claude`, e cole **todo** o bloco entre as linhas `===` (incluindo as linhas `===`). A IA vai conduzir o resto.

```

===
Você vai me ajudar a montar um sistema pessoal de
    
```

produtividade chamado Nerso, baseado em Obsidian + Claude Code. Sua missão tem 4 fases: ENTREVISTA, PLANEJAMENTO, CONSTRUÇÃO, ENTREGA.

REGRAS GERAIS:

- Pergunte UMA coisa por vez. Sem formulário gigante.
- Confirme entendimento depois de cada bloco.
- Se eu der resposta vaga, pergunte mais específico.
- Crie todos os arquivos no diretório atual.
- Idioma de saída: igual ao que eu estou usando.
- Antes de criar QUALQUER arquivo, valide que ele segue as Convenções e Templates abaixo.

=====
 CONVENÇÕES GLOBAIS (valem pra TODA criação)
 =====

A) NAMING DE ARQUIVOS

Tipo de nota	Padrão
Daily note	YYYY-MM-DD.md
Weekly review	YYYY-Www.md
Mês financeiro	MM-AA.md
Log de sessão Nerso	YYYY-MM-DD-HHmm.md
Treino do dia	YYYY-MM-DD.md
Refeição/dia	YYYY-MM-DD.md
Reunião	YYYY-MM-DD-slug.md
Projeto	<slug-kebab>.md
Livro/Curso/Artigo	<slug-kebab>.md

- CLAUDE.md sempre em UPPERCASE, na raiz da pasta correspondente.
- Slugs: minúsculas, kebab-case, sem acento.
- Pasta de arquivamento: _Arquivado/ (prefixo underscore pra cair no fim da listagem).

B) FRONTMATTER YAML (no topo de toda nota estruturada)

- Sempre incluir `data:` em ISO (YYYY-MM-DD).
- Sempre incluir `tags:` como array (mesmo vazio).
- Datas com hora: ISO completo (2026-04-25T14:32).
- Numéricos sem aspas (peso: 78.5, total: 1200).
- Não inventar campos fora do template do módulo.

C) BACKLINKS

- Sempre que referenciar outra nota, use

[[caminho/ate/nota]].

- Em vez de duplicar conteúdo, linke.

D) VOCABULÁRIO CONTROLADO (campo `dominio` em logs e tags)

- Use só estes termos: trabalho, saude, financeiro, rotina, journal, conhecimento, pessoal.
- Sem acento, sem variação de grafia.
- Domínio novo? Adicione à seção "Vocabulário" do CLAUDE.md raiz antes de usar.

=====
 TEMPLATES OBRIGATÓRIOS
 =====

[A] CLAUDE.md DE MÓDULO – formato fixo, 4 blocos:

```
# <NomeModulo> – CLAUDE.md

## Contexto
[1-3 frases: pra que esse módulo existe na vida do usuário]

## Estrutura
[árvore das pastas/arquivos esperados, com comentário curto em cada um]

## Como agir
Quando o usuário disser "<gatilho>":
1. Faça <ação concreta>
2. Atualize <arquivo>
3. Confirme com o usuário

(1 bloco "Quando..." pra cada interação típica do domínio – mínimo 3, máximo 7)

## Regras
- <restrição 1>
- <formato obrigatório>
- <gatilho de atualização>
```

Limite: 100 linhas. Simplifique se passar.

[B] CLAUDE.md RAIZ – formato fixo, 7 blocos:

```
# <NomeAssistente> – CLAUDE.md (raiz)

## Identidade
```

Você é <NomeAssistente>, assistente pessoal contextual estilo Jarvis. <regras de tom>.

Quem é o usuário
 [perfil – só dados que o usuário deu na entrevista, com placeholder "[a definir]" pra lacunas]

Tom e comunicação
 - <regra específica>
 - <comportamento a evitar>

Como Navegar
Intenção do usuário	Leia primeiro
<intenção>	[[Modulo/CLAUDE]]

Gatilhos de atualização
Gatilho	Arquivo a sugerir update
Mudei de emprego	[[Trabalho/CLAUDE]]

Protocolo de sessão
 - Início: ler último log em Nerse/Logs/
 - Fim: criar log seguindo TEMPLATE [C]

Vocabulário
 - Domínios: trabalho, saude, financeiro, rotina, journal, conhecimento, pessoal

[C] LOG DO NERSO (S/K/P/C) – formato fixo:

```

---
data: YYYY-MM-DDTHH:mm
dominio: [trabalho, saude]
acoes: [criou tarefa X, registrou treino Y]
decisoes: [adiar projeto Z para maio]
pendencias: [revisar custos fixos do mês]
tags: [log]
---
    
```

S – Summary
 [1-2 frases do que rolou na sessão]

K – Key items
 - bullet do que foi tratado (com [[backlink]] quando aplicável)

P – Pending
 - [] item pendente

```
## C – Context
- [[Modulo/Arquivo relevante]]
```

[D] DAILY NOTE (se Journal):

```
---
data: YYYY-MM-DD
humor: ?/10
tags: [daily]
---
```

```
## Manhã
## Tarde
## Noite
## Reflexão
```

[E] MÊS FINANCEIRO (se Financeiro):

```
---
mes: MM-AA
total_entrada: 0
total_saida: 0
saldo: 0
tags: [financeiro]
---
```

```
## Lançamentos
| Data | Descrição | Categoria | Valor |
|-----|-----|-----|-----|
```

[F] TREINO (se Saúde + treina):

```
---
data: YYYY-MM-DD
tipo: musculacao
duracao_min: 0
intensidade: moderada
tags: [saude, treino]
---
```

```
## Exercícios
- exercício: séries x reps @ carga
```

```
## Notas
```

```
=====
FASE 1: ENTREVISTA
=====
```

Pergunte na ordem, UMA por vez:

1. "Qual o seu primeiro nome?"
2. "Qual nome você quer dar pro assistente?
(sugestão: Nerso)"
3. "Em uma frase: o que você faz da vida?"
4. "Que idade você tem? (faixa serve)"
5. "Cidade/país (opcional)"
6. "Quais áreas você quer organizar?
[] Financeiro
[] Saúde (treino + alimentação)
[] Trabalho/Carreira
[] Rotina e Hábitos
[] Journal/Daily Notes
[] Conhecimento (livros, cursos, ideias)
[] Outros (descreva)"
7. "Qual tom você quer?
a) Direto e duro
b) Coach motivacional
c) Neutro/profissional
d) Customizado (descreva)"
8. "Algum comportamento que JÁ SABE que não quer?
(ex: sem emojis, sem linguagem corporativa)"
9. "Como é seu dia ideal? (horário, blocos)"
10. "Usa algum app de tarefas? (TickTick, Todoist, Notion, nenhum)"

Pra cada módulo marcado, 1-2 perguntas específicas:

- Financeiro: "Renda e custos fixos aproximados?
Rastrear investimentos?"
- Saúde: "Treina? Tipo? Acompanha macros ou só
intuição?"
- Trabalho: "CLT, PJ, freeela, empreendedor?
Projetos ativos pra registrar?"
- Rotina: "Hábitos? 3-5 mais importantes?"
- Journal: "Daily todo dia ou só weekly?"
- Conhecimento: "Leitura, cursos, ou ambos?
Quer flashcards?"
- Outros: "Descreva cada um em 1 frase"

=====

FASE 2: PLANEJAMENTO

=====

Apresente:

PLANO DE CONSTRUÇÃO

- Árvore completa de pastas/arquivos
- Quais templates ([A]-[F]) cada arquivo usa

- Resumo de 2 linhas do "Como agir" de cada módulo
- Lista de gatilhos no CLAUDE.md raiz

Pergunte: "Aprovado? (sim / mude X)"
 Itere até eu aprovar.

```
=====
FASE 3: CONSTRUÇÃO
=====
```

Crie NA ORDEM, validando contra Convenções e Templates:

1. CLAUDE.md raiz – usar TEMPLATE [B].
 Preencher cada bloco com base na entrevista.
2. Pra cada módulo escolhido (ver SPECS abaixo):
 - Criar pasta
 - CLAUDE.md usando TEMPLATE [A]
 - Estrutura interna conforme spec
 - Frontmatter conforme Convenção (B)
3. Nerso/ (SEMPRE criar, módulo obrigatório):
 - Nerso/CLAUDE.md (regras do sistema de logs, referência ao TEMPLATE [C])
 - Nerso/Logs/Índice Nerso.md (vazio, frontmatter `tags: [indice, log]`)
 - Nerso/Logs/<hoje>-0000.md (primeiro log usando TEMPLATE [C], com S = "Sistema criado", K = lista dos módulos criados)
4. Recursos/Templates/ – copiar TEMPLATE [D]/[E]/[F] conforme módulos escolhidos.
5. .gitignore na raiz:
 - .obsidian/workspace.json
 - .obsidian/workspace-mobile.json
 - .obsidian/workspace-mobile-vertical.json
 - .trash/
6. README.md curto:
 - Como usar (rodar `claude` na pasta)
 - Onde editar quando algo mudar (regra: edite o CLAUDE.md do módulo, não recrie)
 - Link pro guia original

```
=====
SPECS POR MÓDULO (use só os escolhidos)
=====
```

[FINANCEIRO]

- Pasta: Financeiro/
- Arquivos: CLAUDE.md, Dashboard.md, Custos.md, Metas.md, Meses/MM-AA.md (mês corrente, TEMPLATE [E])
- "Como agir" deve cobrir:
 - registrar gasto (anexa linha em Meses/MM-AA.md e atualiza totais do frontmatter)
 - registrar entrada
 - consultar saldo do mês
 - revisar custos fixos
 - ajustar meta de Metas.md

[SAÚDE]

- Pasta: Saude/
- Arquivos: CLAUDE.md, Dashboard.md, Métricas.md, Treinos/, Alimentação/Logs/, Exames/
- "Como agir" deve cobrir:
 - registrar treino (TEMPLATE [F] em Treinos/)
 - registrar refeição (Alimentação/Logs/)
 - registrar peso/medida (anexa em Métricas.md com frontmatter de série temporal)
 - cadastrar exame (anexa em Exames/)

[TRABALHO]

- Pasta: Trabalho/
- Arquivos: CLAUDE.md, Metas <Ano>.md, Projetos/, Reuniões/
- "Como agir" deve cobrir:
 - documentar reunião (Reuniões/<data>-<slug>.md com frontmatter participantes, decisoes, follow_ups)
 - criar projeto novo (Projetos/<slug>/index.md)
 - atualizar status de projeto
 - preparar 1:1 (gerar pauta)

[ROTINA]

- Pasta: Rotina/
- Arquivos: CLAUDE.md, Hábitos.md, Protocolos/
- "Como agir" deve cobrir:
 - briefing matinal (cruza com Trabalho, Saude, Journal)
 - planejamento semanal
 - registrar hábito do dia
 - integração com app de tarefas (se houver MCP configurado)

[JOURNAL]

- Pasta: Journal/

- Arquivos: CLAUDE.md, Daily/, Weekly/
- Criar Daily/<hoje>.md vazio (TEMPLATE [D])
- "Como agir" deve cobrir:
 - criar/atualizar daily note
 - rodar weekly review (cria Weekly/YYYY-Www.md)
 - tagging emocional via campo `humor`

[CONHECIMENTO]

- Pasta: Conhecimento/
- Arquivos: CLAUDE.md, Livros/, Cursos/, Artigos/, Ideias/
- "Como agir" deve cobrir:
 - salvar nota de livro (frontmatter: autor, status, rating)
 - salvar nota de curso/artigo
 - criar nota de ideia livre
 - linkar nota nova com [[backlinks]] de notas relacionadas existentes

=====

FASE 4: ENTREGA

=====

Mostre:

1. Tree completo dos arquivos criados
2. Caminho exato de cada CLAUDE.md
3. Verificação:
 - Cada CLAUDE.md de módulo tem os 4 blocos?
 - Frontmatter de exemplos bate com Convenção (B)?
 - Naming bate com Convenção (A)?
 Se algo não bater, corrija antes de entregar.
4. Próximos passos:
 - Abrir vault no Obsidian
 - Rodar `bom dia, me dá um briefing`
 - Considerar MCPs (TickTick, Notion, Calendar)
 - Ler seção de manutenção do guia

=====

REGRAS DE OURO

=====

- Não invente dados sobre mim (idade exata, salário, doenças). Se não perguntei e não está na resposta, use "[a definir]".
- CLAUDE.md de módulo: máximo 100 linhas.
- NÃO crie módulos que eu não pedi.
- Se eu pedir algo fora do escopo durante a construção, anote em "pendencias" do log e siga.
- Antes de gravar qualquer arquivo, releia o

TEMPLATE correspondente e confirme que está 100% conforme.

Comece agora pela pergunta 1.
===

Como rodar (resumo de 7 passos)

1. Crie pasta vazia que vai ser seu vault.
2. Abra terminal nela e rode `claude`.
3. Cole o prompt acima.
4. Responda às perguntas (uma por vez).
5. Aprove o plano.
6. Espere a IA construir.
7. Abra a pasta no Obsidian.

Dica

Demora uns 15-25 minutos pra responder a entrevista com calma. Vale o tempo. Quanto mais específico você for nas respostas, melhor o sistema sai.

Manutenção e evolução

Sistema vivo só funciona se você sabe como mantê-lo. Aqui as regras.

Edite, não recrie

Quando algo mudar (novo trabalho, nova rotina, novo módulo), **edite o CLAUDE.md correspondente**. Não rode o Prompt Mestre de novo — você perderia tudo.

A IA vai entender o estado novo na próxima sessão.

Gatilhos de atualização

Coloque no CLAUDE.md raiz uma seção assim:

Gatilhos de Atualização

Quando eu mencionar X, sugerir update em Y:

Gatilho	Arquivo
Mudei de emprego	Trabalho/CLAUDE.md
Comecei novo treino	Saude/CLAUDE.md
Mudou despesa fixa	Financeiro/CLAUDE.md
Feedback sobre comporta- mento da IA	Memory (feedback)

A IA vai aprender a sugerir atualizações sozinha conforme você conversa.

Memory system

O Claude Code escreve memórias automaticamente em `~/.claude/projects/<hash>/memory/`.
 Você pode:

- Pedir explicitamente: “salva como memory que eu prefiro X”
- Pedir esquecimento: “esquece a memory sobre Y”
- Inspeccionar: ver o arquivo `MEMORY.md` na pasta de memórias

Atenção

Não force escrita constante de memória. A IA decide o que vale virar memória de longo prazo. Se você marca tudo, ela perde sinal.

Logs do Nerso

Não precisa abrir manualmente. São gerados pela IA. Use pra:

- **Buscar conversa antiga:** “o que falamos sobre X semana passada?”
- **Ver pendências:** a IA lê automaticamente no início de cada sessão
- **Auditoria:** ver decisões importantes que foram tomadas

Quando refatorar o sistema

A cada 3 meses, abra o CLAUDE.md raiz e revise:

- Algum módulo virou ruído? Archive.
- Algum padrão de comportamento está consistente em 5+ logs? Promova pra memory ou pra regra no CLAUDE.md.

- Mudou de fase de vida (novo emprego, mudou pra outra cidade, casou, teve filho)? Atualize o “Quem sou”.

FAQ — perguntas que todo mundo faz

Preciso pagar Claude Code?

Free tier funciona pra começar. Pra uso intensivo (briefings diários, várias sessões/dia), o plano pago compensa rápido.

Funciona offline?

Obsidian sim. Claude Code não — precisa de internet pra falar com a API.

E se meu vault tiver dados sensíveis?

O vault Obsidian fica local. O Claude Code envia o conteúdo dos arquivos lidos pra API da Anthropic. Avalie o que faz sentido guardar e o que não. Pra dados super sensíveis (senhas, segredos médicos, etc.), use um gerenciador dedicado, não o vault.

Posso versionar com Git?

Sim. Recomendo. Faça commit dos CLAUDE.md sempre. Logs do Nerso podem ou não ir pro Git — depende se você quer histórico versionado.

Posso sincronizar entre celular e PC?

Sim, via iCloud Drive, Dropbox, OneDrive ou Obsidian Sync (pago). Claude Code só roda no PC, mas você consulta as notas no celular.

Quantos módulos no máximo?

Não tem limite técnico, mas mais de 10 vira bagunça. Pense em domínios distintos com regras próprias. Se duas coisas são parecidas, junte num módulo só.

A IA me corrige errado às vezes.

Dá feedback explícito: “isso aí não, prefiro X”. A IA salva como memory de feedback e aplica em sessões futuras.

Esqueci de configurar algo crítico no início.

Edita o CLAUDE.md correspondente quando perceber. Sistema é vivo, não estático.

Posso compartilhar meu sistema com outra pessoa?

A estrutura sim (pastas + CLAUDE.md genéricos). Não compartilhe dados pessoais nem memórias.

Próximos passos práticos

1. **Construa.** Rode o Prompt Mestre. Não pense demais — o sistema só faz sentido quando existe.
2. **Use diariamente por 7 dias.** Briefing matinal + 1 registro por dia (treino, gasto, ideia).
3. **Refleta no final da semana.** O que funcionou? O que você pediu que a IA não entregou? Ajusta.
4. **Adicione integrações depois.** TickTick, Notion, Calendar via MCP. Não comece com tudo — comece simples.
5. **Customize.** Crie seus próprios módulos. O sistema é seu, não meu.

A regra de ouro

Construído pra durar, não pra impressionar. Se em 6 meses você ainda usa, valeu. Se virou mais um app abandonado, simplifica. Sistema bom é sistema que sobrevive ao seu próprio entusiasmo inicial.

Quer ir mais fundo?

Esse guia te dá o esqueleto. Mas tem coisa que cabe num ebook de 250 páginas que não cabe num lead magnet:

- **Como dominar o Claude Code de verdade** — instalação, configuração, hooks, skills, MCP, multi-agent, debugging
- **Biblioteca de 51 prompts testados** — todos por categoria, prontos pra colar no seu sistema
- **8 workflows completos** — passo a passo de uso real
- **Templates avançados de CLAUDE.md** — pra freelance, conteúdo, estudo, fullstack
- **Troubleshooting** — quando o Claude responde mal, quando o sistema fica lento, quando você precisa refatorar

Tudo isso está no **Fluente em Claude**.

PRÓXIMO PASSO

Quer dominar o Claude de verdade?

Este guia é só uma amostra. No **Fluente em Claude** você encontra:

- ▶ Guia completo do Claude Desktop — cada feature explicada
- ▶ Claude Code do zero ao avançado — instalação, CLAUDE.md, skills, hooks, MCP
- ▶ 51 prompts testados organizados por categoria — prontos pra copiar
- ▶ 8 workflows completos passo a passo
- ▶ Templates de CLAUDE.md, Project Instructions e Skills
- ▶ Checklists, troubleshooting e FAQ completo

[Quero o guia completo →](#)

https://claude.umporcentoprogramador.com.br?utm_source=guia_gratuito_jarvis&utm_medium=organic&utm_campaign=lead_magnet