



O Segundo Cérebro da Empresa

As 4 camadas antes da IA, a estrutura de pastas e o arquivo de entrada prontos para copiar

por **Gabriel Madureira**

@ga.madureira

madureira.xyz

O Segundo Cérebro da Empresa

A pergunta na ordem errada

Quase toda empresa que me procura querendo "usar IA" está fazendo a pergunta na ordem errada. A pergunta que elas fazem é qual ferramenta. A pergunta que resolve é onde mora a informação da minha empresa.

Isso não é filosofia. É a diferença prática entre uma IA que responde genérico e uma que responde como alguém que trabalha ali há dois anos. O modelo é o mesmo nos dois casos. O que muda é o que você conseguiu colocar na frente dele.

Este kit tem duas partes. A primeira é a ordem que eu uso: quatro camadas, e cada uma só faz sentido se a anterior existir. A segunda são os modelos para copiar: a estrutura de pastas, o arquivo de entrada e as regras duras, do jeito que eu uso na minha operação, com os nomes trocados para você preencher.

Parte 1. As quatro camadas

Camada 1. O segundo cérebro: um lugar, não sete

O primeiro problema da sua empresa não é falta de IA. É que a informação está espalhada em sete lugares que não se falam: e-mail, WhatsApp, Drive, Notion abandonado, a cabeça de duas pessoas e uma planilha que só o financeiro abre.

Segundo cérebro é o nome bonito de uma coisa simples: um lugar só onde o conhecimento da empresa mora, em texto, e que você controla.

Três propriedades importam, e nenhuma é sobre a ferramenta:

Texto puro. Markdown, arquivo comum. Não é purismo de nerd. É que texto puro é a única coisa que qualquer ferramenta futura vai conseguir ler, incluindo as que ainda não existem. Base presa dentro de um app é base que você aluga.

Uma hierarquia rasa e óbvia. Se você precisa pensar onde salvar, a estrutura está errada. Três a quatro níveis resolvem quase toda empresa: cliente, projeto, tipo de documento. Pasta que precisa de manual não sobrevive à segunda semana.

Um ponto de entrada único. Um arquivo, na raiz, que diga o que a empresa é e onde está cada coisa. É o documento que a pessoa nova lê no primeiro dia e, mais tarde, é literalmente o que a IA vai ler antes de fazer qualquer coisa.

Na minha operação isso é um único diretório com três pastas: documentos, código, mídia. E um arquivo na raiz que descreve as frentes vivas do negócio e aponta para o resto. Toda vez que alguém, pessoa ou máquina, chega ali, começa pelo mesmo lugar.

Se você tem menos de vinte pessoas, isso cabe numa tarde. O que demora não é montar. É decidir o que fica de fora.

Camada 2. A estrutura no seu computador

O segundo cérebro precisa estar num disco que você abre, não só numa aba de navegador. Essa é a parte que quase todo mundo pula, e é a que destrava tudo que vem depois.

A razão é técnica: as ferramentas de IA que trabalham de verdade, as que editam arquivo, rodam comando e leem o histórico do projeto, trabalham em cima do sistema de arquivos. Se a sua informação só existe dentro de um SaaS, ela é visível pela IA no máximo por um plugin, um pedaço de cada vez. Se ela existe como pasta, a IA lê tudo.

O desenho que funciona:

- Uma pasta raiz para a empresa inteira. Tudo mora dentro dela.
- Separação por natureza, não por assunto: documento é documento, código é código, arquivo pesado é arquivo pesado. Vídeo de 3 GB no meio dos textos quebra busca, backup e sincronização.
- Sincronização escolhida com consciência. Documento em nuvem sincronizada, código em Git, mídia pesada onde não custa caro. São três problemas diferentes e uma solução só para os três sempre falha em um.

E o aviso que eu dou por experiência própria: decida o backup no dia um. É a coisa que ninguém faz porque nunca é urgente, até o dia em que é a única coisa que importa. Um diretório sem backup é um diretório que você vai perder. A única variável é quando.

Camada 3. A IA que trabalha dentro da estrutura

Com as duas primeiras camadas de pé, a IA deixa de ser um chat e vira um agente que trabalha no seu material. É aqui que entram ferramentas como o Claude Code ou o Codex: elas rodam no terminal, dentro da pasta, e enxergam o que está lá.

A mudança de natureza é essa: você para de descrever o contexto e passa a apontar para ele. Em vez de colar três parágrafos explicando como o cliente fala, você diz onde está o documento que descreve como o cliente fala.

Duas práticas fazem quase toda a diferença:

O arquivo de instrução na raiz. Um documento, sempre lido, que diz as regras da casa: como as coisas se chamam, o que nunca se faz, onde cada tipo de arquivo vive. É a diferença entre uma IA que te pergunta a mesma coisa toda semana e uma que já sabe.

Regras duras, escritas como regras. "Nunca publicar sem aprovação." "Número em material comercial só sai da fonte X." Curtas e absolutas, no arquivo de instrução. Regra que mora na cabeça de alguém não é regra, é sorte.

O ganho real não é velocidade de digitação. É que o contexto para de ser recontado. O trabalho que você fez uma vez explicando como a empresa funciona passa a valer para todas as conversas seguintes, inclusive as que outra pessoa vai ter.

Camada 4. A ponte com o sistema de tarefas

As três primeiras camadas resolvem conhecimento. Falta o que a empresa está fazendo agora, e isso mora no ClickUp, no Trello, no Linear, no que a sua empresa já usa.

Aqui eu recomendo começar por um MVP, e o mais burro possível. A tentação é desenhar a integração completa antes de ligar qualquer coisa. Ela sempre morre no meio.

1. **Uma direção só.** Da IA para o sistema de tarefas: ela cria a tarefa, com título decente, descrição e responsável. Ida e volta sincronizada é um projeto dez vezes maior e você ainda não sabe se vai usar.
2. **Um responsável padrão, definido no código.** Tarefa sem dono é tarefa que ninguém vê. Se você não decidir o padrão, o padrão vira "ninguém".
3. **Um campo de origem.** Uma etiqueta que marca o que foi criado pela automação. No dia em que der errado (e vai dar, uma vez), é o que te deixa achar e limpar em um minuto em vez de uma tarde.

4. Nada que publica, nada que apaga. No MVP, a automação só cria e só comenta. As ações destrutivas e as públicas continuam passando por gente. Essa fronteira eu mantenho até hoje, muito depois do MVP.

A ordem importa mais que as escolhas

Se você trocar Claude Code por outra ferramenta, o guia continua de pé. Se trocar ClickUp por Trello, idem. O que não dá para trocar é a ordem:

1. Um lugar só, em texto, que você controla.
2. Esse lugar num disco que a máquina abre.
3. A IA trabalhando dentro dele, com as regras escritas.
4. A ponte para o sistema de tarefas, começando pequeno.

Quase todo projeto de "IA na empresa" que eu vi fracassar começou pelo passo 3, sem os passos 1 e 2. O resultado é sempre o mesmo: uma ferramenta cara respondendo genérico, porque ninguém deu a ela nada específico para ler.

Parte 2. Os modelos para copiar

Modelo A. A estrutura de pastas

Esta é a estrutura que eu uso, com os nomes genéricos. Três pastas na raiz separadas por natureza, e dentro de documentos uma hierarquia rasa.

```

EMPRESA/
├── CLAUDE.md           ← o arquivo de entrada (Modelo B)
├── documentos/         ← tudo que é texto; sincroniza em nuvem
│   ├── 00-PAINEL/      ← diário, decisões, dashboard
│   │   ├── diario/     ← um arquivo por dia: AAAA-MM-DD.md
│   │   └── decisoes/   ← uma decisão por arquivo, datada
│   ├── 01-EMPRESA/     ← o negócio: oferta, preço, processos
│   │   └── CLIENTES/
│   │       └── <CLIENTE>/
│   │           ├── CLAUDE.md      ← cérebro do cliente (Modelo C)
│   │           ├── 01-ESTUDO/
│   │           ├── 02-CONTEUDO/
│   │           └── 03-DOCUMENTACAO/
│   ├── 02-PROJETOS/    ← o que não é cliente nem empresa
│   └── 99-SISTEMA/      ← regras, padrões, modelos, skills
├── code/               ← tudo que tem package.json ou .git; vive no Git
└── media/              ← vídeo, bruto, arquivo pesado; fora da
sincronização de texto

```

Regra de roteamento, em uma linha cada:

- Tem código ou repositório? Vai para `code/`.
- É texto ou asset leve? Vai para `documentos/`.
- É vídeo ou pesa mais de 50 MB? Vai para `media/`.
- Nunca arquivo solto na raiz. Nunca código dentro de documentos.

Modelo B. O arquivo de entrada (raiz)

Copie, troque o que está entre colchetes, apague o que não vale para você. O arquivo tem quatro blocos e cabe em uma tela. Se passar de duas, está virando manual, e manual ninguém lê.

```
# [NOME DA EMPRESA]
```

```
Workspace de [quem]. [Idioma] sempre.
```

```
## As frentes vivas. Só estas.
```

```
| Frente | O que é | Onde está tudo dela |  
|---|---|---|  
| 1. [Frente A] | [uma linha] | `documentos/01-EMPRESA/FRENTE-A.md` |  
| 2. [Frente B] | [uma linha] | `documentos/02-PROJETOS/FRENTE-B.md` |
```

Este arquivo é o ponto de partida único. Os índices acima dizem onde está cada coisa de cada frente. Nenhum outro documento se apresenta como entrada.

[Opcional] Frentes descontinuidades: [lista]. Nenhum trabalho novo nelas.

```
## Onde ficam as coisas
```

Regra: código em `code/`. Texto e asset leve em `documentos/`. Vídeo em `media/`.

Nunca código dentro de documentos. Nunca arquivo solto na raiz.

```
## Regras duras
```

1. Nunca publicar nem agendar sem OK explícito de [responsável]. Sobe como rascunho e para.
2. Número em material comercial só sai de [arquivo-fonte]. Se o número não está lá,
 parar e pedir apuração. "Não rastreado" é resposta correta.
3. Nunca apagar. Sempre mover para pasta datada.
4. Nunca tocar em `.env` ou `.secrets`. Nunca imprimir valor de segredo.
5. Antes de trabalhar num cliente, ler `documentos/01-EMPRESA/CLIENTES/<CLIENTE>/CLAUDE.md`.
6. [Sua regra 6]

```
## Diário
```

Toda tarefa pedida, iniciada e finalizada vai para `documentos/00-PAINEL/diario/AAAA-MM-DD.md`.

Um arquivo por dia. Uma linha por tarefa. Ao concluir, marcar `[x]` no lugar.

Seções: Feito · Lido · Notas · Próximos (teto de 3 linhas abertas).

Três decisões de desenho que valem explicar:

"Só estas." A tabela de frentes é curta de propósito. Quando eu tinha doze projetos listados, nenhum agente (nem pessoa) sabia o que era prioridade. Com três, sabe. O que saiu da tabela não desaparece: vai para uma pasta datada de parados, com a razão

escrita.

Regras duras numeradas e absolutas. Sem "de preferência", sem "quando possível". A regra 2 (número só da fonte) é a que mais me salvou: ela impede que um número sem lastro entre em proposta, post ou site, porque a IA tem onde conferir e a instrução de parar quando não achar.

Diário obrigatório. Parece burocracia. É memória. Quando o trabalho é feito por várias pessoas e vários agentes, o diário é a única coisa que responde "o que aconteceu terça" sem depender de quem estava lá.

Modelo C. O cérebro de cliente

Um por cliente, dentro da pasta dele. É lido antes de qualquer peça.

```
# [CLIENTE]: cérebro do cliente

## Identidade
- Quem é: [uma linha]
- Posicionamento: [uma linha]
- Público: [uma linha]
- Handles e canais: [tabela: rede, handle, idioma, cadência]

## Pilares de conteúdo
| Pilar | % | 0 que entra | 0 que nunca entra |

## Voz
- Palavras que a marca usa: [...]
- Palavras proibidas: [...]
- Dois exemplos de peça aprovada: [links ou caminhos]

## Regras do cliente (não negociáveis)
1. [ex.: nenhuma peça sai sem aprovação de X]
2. [ex.: nunca citar preço]
3. [...]

## Integrações e identificadores
| Sistema | Identificador |

## Histórico de decisões
- AAAA-MM-DD: [decisão, em uma linha]
```

Modelo D. A regra de número

Esta é uma regra só, mas merece arquivo próprio porque é a que mais protege a reputação. A ideia: todo número que a empresa afirma em material público (site, proposta, post) sai de um único arquivo, e cada item desse arquivo carrega a fonte e a data de apuração.

Prova da casa

Regra: nada entra aqui sem `fonte` (caminho ou link verificável) e `apurado` (data).

Nada aqui é média da empresa. É resultado de UM cliente, num período, e a copy diz isso.

– valor: [ex.: 33% de abertura]

o que mede: [ex.: abertura média da newsletter do cliente X, N assinantes]

cliente: [X]

fonte: [caminho do print ou export]

apurado: [AAAA-MM]

O que não tem fonte sai do ar. Eu já tirei oito métricas do meu próprio site por essa regra, e a prova qualitativa honesta que sobrou vende mais que número grande que não sobrevive a uma pergunta.

Como montar numa tarde

1. Crie a pasta raiz com `documentos/`, `code/` e `media/`.
2. Mova para `documentos/` tudo que é texto e está espalhado. Não organize ainda. Só junte.
3. Escreva o arquivo de entrada (Modelo B) com as frentes vivas e as 3 regras duras que você já sabe.
4. Abra a IA dentro da pasta e peça para ela ler o arquivo de entrada e dizer o que não entendeu. O que ela não entendeu, você reescreve.
5. Decida o backup hoje. Nuvem para documentos, Git para código, disco externo ou armazenamento barato para mídia.
6. Amanhã, escreva o primeiro diário.

A parte chata é a estrutura. É também a única que ainda vai estar valendo daqui a três anos, quando as ferramentas de hoje tiverem outro nome.

Sobre este material

O Segundo Cérebro da Empresa. Escrito por Gabriel Madureira, fundador de uma agência de marketing e criador de conteúdo sobre marketing, IA aplicada e operação. Versão de outubro de 2026.

Tudo aqui vem do que eu uso de verdade no meu trabalho: as regras, as medições e os exemplos saíram da minha operação e das peças que eu publiquei. Onde um número aparece, ele tem fonte. Onde não tinha fonte, o número não entrou.

Este material não promete resultado. Ele mostra o critério. O resultado depende do que você faz com ele.

@ga.madureira · madureira.xyz