APRESENTAÇÕES
FEITAS COM IA

COMPATÍVEL COM:

MIRA ANIMATOR

METÁFORAS INTELIGENTES RESPONSIVAS E ANIMADAS

Sandeco Macedo

Aos meus alunos dos mais de 10 livros que já lancei, vocês são fantáticos!


Meus Livros sobre
Inteligência Artificial
crewAI
Para iniciantes
Agentes
Inteligentes
vol. 1
Sandeco Macedo
Para iniciantes
com IA
Python para
Inteligência
Artificial
Sandeco Macedo
Tudo o que você precisa saber
do Python antes de criar suas IAs
Prompts em Ação
Engenharia de Prompts
para leigos
Sandeco Macedo
Resultado
das minhas
pesquisas em IA

Sumário

CAPÍTULO 1

O que é o Mira e como instalar

Imagine que você passou três semanas construindo um projeto. Escreveu o código, documentou as decisões, sofreu com os bugs, comemorou quando finalmente funcionou. Aí chega a sexta-feira e alguém te avisa: segunda tem apresentação. Você precisa transformar tudo aquilo, o repositório inteiro, num punhado de slides bonitos que caibam em vinte minutos de fala.

E você conhece esse filme. Vai abrir o programa de slides, copiar um trecho de código aqui, colar um diagrama ali, procurar um ícone que não seja feio, alinhar caixinhas de texto às duas da manhã. No fim, o que você entrega é um resumo empobrecido do que você fez. As setas não se movem, o fluxo não flui, a ideia que era viva no seu código vira uma foto parada numa parede.

Convenhamos: o problema nunca foi falta de conteúdo. O conteúdo já existe, está no seu repositório, no seu livro, no seu PDF, no artigo que você acabou de ler. O problema é a ponte entre o conteúdo que você já tem e a apresentação que você precisa mostrar. Essa ponte é lenta, manual e dolorosa. É exatamente essa ponte que o Mira constrói para você.

1.1 O QUE É O MIRA

O Mira é uma ferramenta que transforma conteúdo que já existe em apresentações animadas que rodam no navegador. Você aponta uma fonte (uma pasta de projeto, um livro, um PDF, um artigo) e, por meio de uma conversa com o seu assistente de código, vai saindo do outro lado um deck completo: slides com cards translúcidos, texto refinado e, o mais importante, animações que se mexem sozinhas.

O nome não é aleatório. **MIRA** é a sigla de **Metáforas Inteligentes Responsivas e Animadas**, e cada palavra dessa sigla é uma promessa que a ferramenta cumpre. Metáforas, porque a aposta central é ensinar por analogia, não por definição fria. Inteligentes, porque

quem lê a sua fonte e monta os slides é um time de agentes de inteligência artificial, não você no braço. Responsivas, porque o mesmo deck se adapta ao formato que você precisa, seja a tela larga da palestra, o quadrado do feed ou o vertical do celular. E Animadas, porque nada ali fica parado: o slide se comporta como um pequeno vídeo, não como uma fotografia.

Repare no que isso significa na prática. Você não abre o Mira e começa a escrever slides do zero. Você já tem o material. O que faltava era um tradutor, alguém que lesse aquilo que você produziu e reexpressasse em linguagem visual. O Mira é esse tradutor. Ele não inventa o seu assunto; ele veste o seu assunto.

E como todo bom tradutor, ele trabalha por etapas. Por baixo dos panos existe uma linha de montagem de agentes especializados: um lê a fonte, outro planeja quantos slides fazem sentido, outro cuida do texto, outro monta a tela, outro dá vida às animações e um último confere se está tudo certo. Você não precisa decorar esse pipeline agora (ele é o assunto dos próximos capítulos), mas guarde a imagem: não é um único robô fazendo tudo, é uma equipe, cada um bom numa coisa só.

Não é um editor de slides. É uma fábrica de apresentações a partir do que você já escreveu.

1.2 POR QUE UMA METÁFORA EXTRAI A ALMA DE UMA IDEIA

Antes de instalar qualquer coisa, precisamos conversar sobre a ideia que sustenta o Mira inteiro. Porque se você não comprar essa ideia, a ferramenta vira só mais um gerador de slides bonitos, e ela é muito mais do que isso.

A pergunta é simples e você já a viveu como aluno e como professor: por que algumas explicações grudam para sempre e outras escorregam no minuto seguinte? Você provavelmente não lembra a definição formal de corrente elétrica que decorou na escola. Mas se alguém te disse uma vez que a corrente é como a água correndo num cano, com a tensão sendo a pressão e a resistência sendo o estreitamento do tubo, isso ficou. Ficou porque não era uma definição. Era uma imagem.

A metáfora como atalho cognitivo

A mente humana é preguiçosa no melhor sentido da palavra: ela odeia começar do zero. Toda vez que algo novo aparece, a primeira coisa que o cérebro faz é procurar, no que já sabe, algo parecido para se agarrar. É um instinto de sobrevivência cognitiva. O desconhecido assusta; o familiar acalma.

Uma metáfora é exatamente esse agarrar-se. Quando eu digo que um agente de código

Figura 1.1: Uma boa metáfora funciona como uma ponte: ela ancora o conceito novo e abstrato naquilo que a mente já conhece do cotidiano, preservando a estrutura da ideia enquanto descarta o que é acessório.

é como um estagiário muito rápido e muito literal, eu não estou enfeitando a frase. Eu estou te entregando um atalho: você já sabe como um estagiário se comporta, então eu importei todo esse conhecimento de graça e apliquei ao conceito novo. Você não teve que construir o entendimento tijolo por tijolo. Você emprestou uma estrutura pronta.

O problema de ensinar por definição pura é que a definição não tem onde se agarrar. *"Um deck é um conjunto ordenado de artefatos de apresentação renderizados em HTML."* Está correto, e é inútil. Ninguém aprende nada com isso, porque não há gancho, não há âncora, não há nada familiar em que a ideia possa descansar. É informação flutuando no vácuo, e o que flutua no vácuo se perde.

Do abstrato ao concreto

Existe uma assimetria cruel na forma como aprendemos. O abstrato é onde mora o poder de uma ideia (a generalização, a regra que vale para mil casos), mas o abstrato é também onde a ideia é impossível de segurar. Já o concreto é limitado (é só um caso, um exemplo, uma cena), mas o concreto a gente pega com a mão.

O truque de todo bom professor é fazer o abstrato descer até o concreto sem perder a alma no caminho. Você não explica recursão dizendo que é uma função que se invoca sobre uma versão reduzida do próprio problema. Você diz: é como dois espelhos de frente um para o outro, cada reflexo contendo outro reflexo menor, até sumir. A recursão continua sendo abstrata; mas agora ela tem um corpo, uma cena, um lugar no mundo.

O cérebro não aprende por definição, ele aprende por analogia. A definição é o destino, não o caminho. Ninguém entende o abstrato indo direto ao abstrato; a gente sobe até lá

pisando em degraus concretos. A metáfora é o degrau. Tire o degrau e o aluno fica olhando para o topo sem saber como chegar.

A alma do conceito

Aqui está a parte sutil, e é a que dá nome a esta seção. Uma metáfora ruim decora; uma metáfora boa **preserva a estrutura**. Essa é a diferença entre enfeite e explicação.

Pense no que uma boa metáfora faz. Ela pega um conceito e pergunta: o que aqui é essencial e o que é acessório? A água no cano preserva o essencial da corrente elétrica (algo flui, há uma força que empurra, há uma resistência que atrapalha) e descarta o acessório (a água não tem carga, não tem elétron, não tem campo). E tudo bem descartar, porque o que foi descartado não era a alma da ideia. A alma era a estrutura do fluxo, e essa a metáfora guardou inteira.

É por isso que digo que uma boa metáfora extrai a alma de uma ideia. Ela não copia o conceito, isso seria impossível e nem útil. Ela faz uma coisa mais difícil e mais valiosa: separa o que a ideia *é* do que a ideia *tem*, guarda o ser e joga fora o ter. Uma metáfora ruim faz o contrário, se apega a um detalhe superficial (a cor, o formato, a palavra) e deixa a estrutura escapar. Por isso ela decora sem ensinar: enfeitou o acessório e perdeu a alma.

Traduzindo: escolher a metáfora certa é um ato de análise, não de decoração. É preciso entender o conceito fundo o bastante para saber o que nele pode ser sacrificado sem que ele deixe de ser ele mesmo. Quem escolhe a metáfora certa provou que entendeu o conceito de verdade.

Metáfora somada à animação

Agora chegamos ao pulo do gato do Mira. Uma metáfora concreta já é poderosa parada. Mas quase toda ideia que vale a pena ensinar não é sobre uma coisa, é sobre uma coisa *acontecendo*. Corrente é fluxo. Recursão é descida. Um algoritmo é uma sequência. Um pipeline é uma passagem de mão em mão. Tudo isso são processos, e processo tem tempo dentro.

Uma imagem parada de um processo é uma contradição. É como explicar uma dança mostrando uma única foto do dançarino no ar. Você vê a pose, mas perde o movimento, e o movimento era tudo. Quando você congela um processo numa figura estática, o leitor precisa fazer o trabalho mais pesado sozinho: reconstruir o movimento na cabeça, imaginar a seta saindo daqui e chegando ali, supor a ordem dos passos. A maioria não faz esse trabalho, e o que não se anima na tela não se anima na mente.

Ver o conceito se mover resolve isso de um jeito quase injusto de tão eficiente. Quando a

metáfora se movimenta diante dos seus olhos, o entendimento não precisa ser reconstruído, ele é entregue pronto. A água corre, o reflexo afunda no espelho, a mão passa o objeto para a mão seguinte. O tempo do conceito vira o tempo da tela, e a sua atenção segue o movimento sem esforço. É por isso que o Mira não faz slides parados. Um conceito que é movimento merece uma explicação que se move.

Onde isso vive no Mira

Toda essa filosofia não é discurso de abertura, ela é lei dentro da ferramenta. O Mira tem uma regra que atravessa tudo o que ele produz, e vamos chamá-la pelo nome que ela tem: a **Regra Zero**. Ela diz que nenhuma animação pode ser estática. Toda animação entra em cena com uma coreografia própria e, depois que entrou, continua se movendo em loop, para sempre, sozinha. Um slide parado, no mundo do Mira, não é uma opção simples nem um estilo minimalista: é considerado um defeito.

E há um comando que encarna literalmente esta seção inteira. Ele se chama `/mira-animated-metaphor`, e o nome não deixa dúvida sobre o que ele faz: pega o conceito de um slide, inventa uma analogia concreta do cotidiano para ele e anima essa analogia na tela. É a sigla MIRA virando ação. Você vai conhecê-lo de perto no próximo capítulo; por ora, basta saber que aquela conversa sobre extrair a alma de uma ideia tem, do lado de dentro do Mira, um botão com o seu nome.

Uma boa ideia não se explica. Ela se encena.

1.3 A FILOSOFIA REVERSA

Deixe eu te contar o pesadelo que o Mira decidiu nunca causar. Você tem um repositório de produção, código que roda de verdade, ou o material de um cliente que confiou em você. Aí você instala uma ferramenta nova bem no meio desse projeto para gerar uns slides, e a ferramenta começa a criar pastas, arquivos de configuração, coisas que você não pediu, espalhadas no meio do que era sagrado. No dia seguinte você não sabe mais o que é seu e o que é da ferramenta. É como convidar alguém para fotografar a sua casa e a pessoa começar a mudar os móveis de lugar para a foto ficar melhor.

O Mira faz o contrário, e chama isso de **filosofia Reversa**. Em vez de se instalar dentro do projeto que você quer apresentar, ele se instala numa pasta separada, isolada, só dele. Dessa pasta, ele *lê* as suas fontes, mas nunca escreve dentro delas. A regra é curta e absoluta: os agentes leem das fontes; eles escrevem somente em `decks/`. Suas fontes são somente leitura, e a pasta `decks/` é a única coisa que o Mira toca com a caneta.

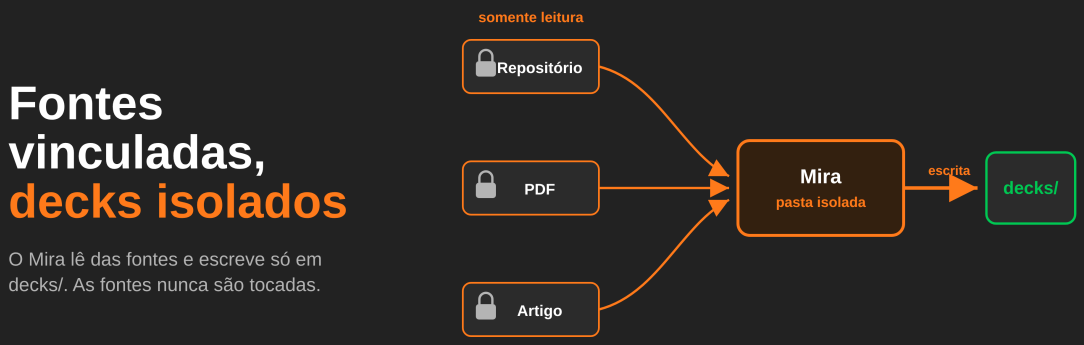

Figura 1.2: A filosofia Reversa: o Mira vive numa pasta de trabalho isolada, lê as fontes vinculadas em modo somente leitura e escreve exclusivamente na pasta de decks, sem jamais modificar os projetos originais.

Parece um detalhe técnico, mas é uma decisão de confiança. Por causa dela, você pode conectar ao Mira um repositório de produção, um contrato de cliente, um livro inteiro, sabendo com certeza absoluta que nada daquilo vai ser alterado, movido ou apagado. O pior que pode acontecer é o Mira ler algo e não usar bem. Ele jamais vai estragar a sua fonte, porque para ele a sua fonte é intocável por construção, não por promessa.

É a diferença entre um fotógrafo que mexe nos seus móveis e um que fotografa a casa exatamente como ela é. Suas fontes ficam onde estão, do jeito que estão. O Mira só olha e reconta.

1.4 O QUE TORNA O MIRA DIFERENTE

Se você tirar duas coisas deste capítulo, que sejam estas: como o Mira *parece* e como o Mira *se mexe*. As duas juntas são a assinatura da ferramenta, o motivo de um deck do Mira ser reconhecível de longe.

A aparência tem nome: **glassmorfismo**. É aquele visual de vidro fosco em que os cards da tela são translúcidos, com bordas suaves, um leve brilho nas quinas e um desfoque delicado por trás, como se o conteúdo flutuasse sobre uma superfície de vidro iluminado. Não é enfeite gratuito. Esse tratamento cria profundidade e hierarquia: o olho sabe na hora o que está na frente, o que é o herói do slide e o que é fundo. A informação respira.

O movimento tem nome também, e você já o conhece: a Regra Zero. Vale repetir porque é o coração da coisa. Nenhuma animação do Mira é estática. Cada uma entra com uma coreografia de abertura e, terminada a entrada, não para: entra num loop interno perpétuo, se

movimentando para sempre sem que ninguém precise clicar em nada. O objetivo declarado é que a apresentação se comporte como um vídeo, não como um álbum de slides. Você abre o deck e ele já está vivo.

A Regra Zero é o que separa o Mira de um gerador comum de slides: toda animação entra com coreografia e depois continua em loop interno, para sempre. No mundo do Mira, um slide parado não é minimalismo, é um bug. A apresentação tem que funcionar como vídeo, não como slideshow.

Junte as duas: cards de vidro que dão profundidade e animações que nunca param de se mover. É por isso que um deck do Mira não parece uma apresentação, parece uma vitrine viva. E o melhor é que tudo isso roda no navegador, num arquivo que você abre com dois cliques.

1.5 PRÉ-REQUISITOS

Antes de instalar, dois pré-requisitos, e nenhum deles é complicado.

O primeiro é o **Node.js** na versão 18.20.2 ou mais recente. É o que permite rodar o instalador do Mira pela linha de comando. Se você já mexe com projetos modernos, provavelmente já tem; se não, é uma instalação de poucos minutos no site oficial. Para conferir a versão que você tem, um comando basta:

```
node --version
```

O segundo é um **assistente de código com inteligência artificial** que saiba ler skills, ou seja, que consiga carregar e executar os agentes especializados que o Mira instala. O Mira foi feito sob medida para o Claude Code, o assistente de linha de comando da Anthropic. É com ele que você vai conversar para criar os decks: você digita instruções em português, ele aciona os agentes certos, e os slides vão nascendo dessa conversa.

Só isso. Node para instalar, um assistente de código para conversar. Com esses dois na mesa, você está pronto.

1.6 INSTALAÇÃO COM UM COMANDO

Aqui vem a parte que costuma assustar e não deveria. Instalar o Mira é um comando. Um. Você abre o terminal dentro de uma pasta dedicada aos seus slides (e sublinho: uma pasta *só* para isso, não o projeto que você quer apresentar, lembre-se da filosofia Reversa) e digita:

```
npx mira-animator install
```

Pronto. Não há assistente de vinte telas, não há caixinhas para marcar, não há reinício de máquina. O comando faz o trabalho e devolve o terminal para você com o ambiente montado.

Por que insistir tanto na pasta dedicada? Porque é o momento em que a filosofia Reversa deixa de ser teoria e vira prática. Se você rodar esse comando no lugar errado, dentro do projeto que quer apresentar, você mistura o que deveria ficar separado. Rodando numa pasta limpa e só sua, você garante que o Mira tem a casa dele, e os seus projetos-fonte continuam intocados lá onde sempre estiveram. Escolha bem a pasta antes de apertar Enter, e o resto se resolve sozinho.

Um comando, uma pasta limpa. O ambiente inteiro nasce daí.

1.7 O QUE É CRIADO

Quando o instalador termina, ele não deixou uma bagunça, deixou uma estrutura. Vale a pena olhar o que apareceu na sua pasta, porque cada peça tem um papel e conhecê-las agora poupa confusão depois.

- Uma pasta com os **agentes** do Mira, os tais assistentes especializados que o seu assistente de código vai acionar. São eles que fazem o trabalho pesado do pipeline.
- Uma pasta de **templates**, com os modelos de deck e os temas visuais de partida. É o guarda-roupa do Mira: os pontos de partida prontos que você vai vestir com o seu conteúdo.
- A pasta `decks/`, ainda vazia. Lembre-se: esta é a *única* pasta onde o Mira escreve. Cada apresentação que você criar vira uma subpasta aqui dentro.
- Um arquivo `mira.config.json`, o painel de controle da instalação. É nele que ficam registradas as suas fontes vinculadas, o tema padrão e os decks já criados.
- Um arquivo `CLAUDE.md`, que funciona como o manual de comportamento da pasta: ele orienta o seu assistente de código sobre as regras do Mira, para que ele aja como esperado.

Não precisa decorar nada disso. A ideia é só te dar o mapa da casa antes de você começar a morar nela. Com o tempo, essa estrutura vira paisagem, e você nem repara mais nela. Mas na primeira vez é bom saber que cada pasta tem uma razão de existir.

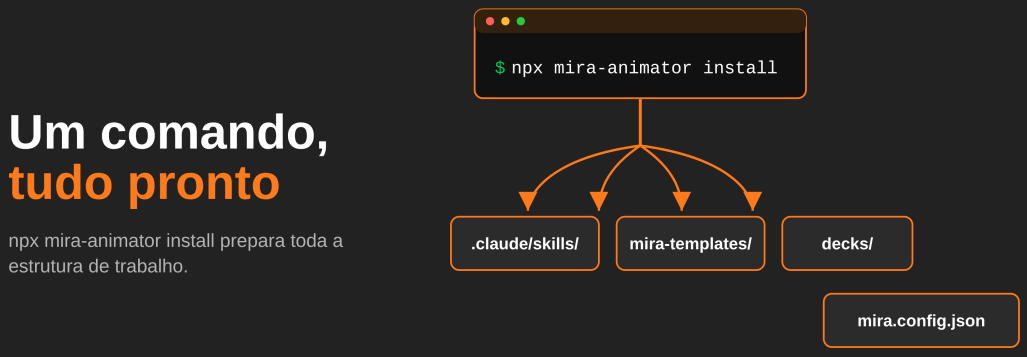

Figura 1.3: O resultado da instalação: a partir de um único comando, o Mira monta a estrutura de trabalho com os agentes, os templates, a pasta de decks (a única onde ele escreve), o arquivo de configuração e o manual de comportamento.

1.8 VINCULANDO SUAS FONTES

Ambiente montado, chegou a hora de dar comida ao Mira. E a comida do Mira são as suas fontes. Vincular uma fonte é dizer para a ferramenta: olhe aqui, é deste material que a minha apresentação vai sair. Você faz isso pela linha de comando, apontando um caminho:

```
npx mira-animator link C:/projects/reversa --name=reversa
```

O que acontece aí? Você entregou ao Mira um caminho (no exemplo, uma pasta de projeto) e deu a ela um apelido curto, `reversa`, com a opção `--name`. Esse apelido é o nome pelo qual você vai se referir a essa fonte daqui para frente, sem ter que digitar o caminho inteiro toda vez. É um cadastro, nada mais: o Mira anota onde a fonte está e como você a chama.

E não precisa ser uma pasta de código. O Mira aceita vários tipos de fonte e, na maioria das vezes, descobre sozinho com o que está lidando. Um PDF de um artigo, um livro em formato de texto, um documento estruturado: é só apontar.

```
npx mira-animator link ./inbox/paper.pdf
```

Depois de vincular algumas fontes, é natural querer conferir o que já está na mesa. Para isso existe um comando que lista tudo o que você conectou:

```
npx mira-animator sources
```

Vale reforçar, porque nunca é demais: vincular uma fonte é conceder *leitura*, não escrita. O Mira vai ler daquele PDF, daquela pasta, daquele livro, mas jamais vai alterar uma vírgula do original. Você pode conectar o que tem de mais precioso sem medo. Ligar uma fonte é abrir uma janela, não uma porta.

1.9 TEMAS E TEMPLATES DE PARTIDA

Você não começa do zero, e essa é uma bênção que muita gente subestima. O Mira vem com dois tipos de ponto de partida prontos: os **temas**, que definem a aparência, e os **templates de deck**, que definem a estrutura.

Os temas são as paletas visuais. O padrão deste projeto, e o que dá a cara do Mira, é o `mira-dark`: fundo preto com um laranja vibrante por cima, aquele contraste que faz a animação saltar da tela. Mas não é o único. Há também um tema claro e minimalista, um corporativo em tons de azul e um de verde neon, para quando o assunto ou a marca pedirem outro clima. A regra de ouro, aqui, é usar sempre as cores do tema escolhido, nunca uma cor solta na mão. É isso que mantém o deck coerente do primeiro ao último slide: troque o tema e o deck inteiro se reveste de novo, sem retoque manual.

Os templates de deck, por sua vez, são esqueletos de apresentação pensados para propósitos comuns. Existe um voltado para transformar o capítulo de um livro ou uma aula em slides; um pensado para o pitch de um projeto; e um feito para uma demonstração técnica. Cada um já traz um ritmo e uma organização adequados ao seu tipo de conteúdo, para você não ter que inventar a estrutura do nada toda vez.

Escolher tema e template é como escolher a roupa e o corte antes de vestir o conteúdo. O trabalho pesado da forma já está feito; você entra com a alma.

1.10 A LINHA DE COMANDO

Para fechar o capítulo, vale conhecer o pequeno painel de comandos que o Mira oferece na linha de comando. São poucos, e você já usou os principais sem perceber. Vale organizar o conjunto:

- `install`: monta o ambiente do Mira na pasta atual. É o comando com que tudo começa.
- `link`: vincula uma nova fonte para o Mira ler.

- `sources`: lista todas as fontes que você já vinculou.
- `status`: mostra o estado atual da instalação, para você saber onde está pisando.
- `update`: atualiza o Mira para a versão mais recente, trazendo agentes e melhorias novas.
- `uninstall`: remove o Mira da pasta, caso você queira recomeçar ou encerrar.

Repare no que *não* está nessa lista: não há um comando de linha para criar uma apresentação. E isso é proposital. Criar um deck no Mira não é um comando frio de terminal, é uma conversa. Você não digita parâmetros para gerar slides; você pede, em português, ao seu assistente de código, algo como criar uma apresentação com determinado nome e determinado template. Um exemplo do que essa conversa parece:

"/mira-new crie uma apresentação chamada 'minha-aula' com o template de aula"

A partir daí, o Mira acorda o time de agentes e a apresentação começa a nascer da conversa. Como essa conversa se desenrola, quais agentes entram em cena e em que ordem, é o assunto dos próximos capítulos. Por ora, você tem uma amostra do que sai do outro lado.

Figura 1.4: Um exemplo de slide gerado pelo Mira: card de vidro fosco sobre fundo escuro, com a animação viva no centro e o conteúdo organizado ao redor. É o tipo de resultado que nasce da conversa com o assistente de código.

A linha de comando prepara o terreno; a conversa faz a apresentação. Instalar o Mira leva um comando e um minuto. O que você vai fazer com ele é o que preenche o resto deste livro.