

DO CONCEITO AO CÓDIGO:

INTELIGÊNCIA ARTIFICIAL NO DESENVOLVIMENTO WEB

UM GUIA PRÁTICO

POR DANIEL HASAN DALIP E IZABELA A. ANDRADE



DO CONCEITO AO CÓDIGO:

INTELIGÊNCIA ARTIFICIAL NO DESENVOLVIMENTO WEB

UM GUIA PRÁTICO

“Nada na vida deve ser temido, somente compreendido.”

Marie Curie

**DO CONCEITO AO CÓDIGO:
INTELIGÊNCIA ARTIFICIAL
NO DESENVOLVIMENTO WEB
UM GUIA PRÁTICO**

Se você quer aprender a usar a inteligência artificial para tirar suas ideias do papel e construir projetos de forma mais rápida, criativa e eficiente, este guia é o ponto de partida certo.

Aqui, você vai descobrir como integrar a IA em diferentes etapas do desenvolvimento.

A proposta não é substituir o seu trabalho, mas potencializá-lo: unir raciocínio humano, criatividade e tecnologia para desenvolver soluções mais inteligentes, organizadas e escaláveis.

Com exemplos reais, dicas práticas e ferramentas que fazem a diferença, este e-book vai te ajudar a transformar seus processos e alcançar novos níveis de produtividade e inovação.

Vamos juntos explorar o poder da IA e criar projetos que unem técnica, propósito e impacto.

DO CONCEITO À TELA:

**DESENVOLVENDO
PROTÓTIPOS COM IA
UM GUIA PRÁTICO**

IDEALIZAÇÃO & ELABORAÇÃO



DANIEL HASAN DALIP

Daniel H. Dalip é professor no Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) e é coordenador de Inovação e Empreendedorismo na instituição. Ele possui doutorado (UFMG/2015), mestrado (UFMG/2009) e bacharelado (Uni-BH/2006) em Ciência da Computação. Com ampla experiência docente, Daniel leciona disciplinas como Programação Web, Algoritmos, Recuperação de Informação e Aprendizado de Máquina.



IZABELA A. M. ANDRADE

Cursou ensino médio técnico em Informática no CEFET-MG (2019-2021), onde participou dos projetos Bytes & Elas e Enxurrada de Bits, promovendo a inclusão de mulheres e jovens de escolas públicas na computação. Tem grande interesse em desenvolvimento web, especialmente front-end. Atualmente, é estudante de Ciências Biomédicas na USP e atua como voluntária no curso de Introdução a HTML, CSS, JavaScript e UX.

INFORMAÇÕES DO CURSO

Instagram - @expresso.dev.cefetmg

W E B . E X P R E S S O D E V . C O M . B R

1 Introdução:

Bem-vindos ao Futuro — a IA como aliada estratégica no desenvolvimento de código

```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown" data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```

De fóruns a prompts: a nova era do dev

Quando muitos aprendiam a programar, um bug trivial como um ponto e vírgula faltando ou uma indentação incorreta podia consumir horas preciosas.



Nesses momentos de frustração, estranhos que haviam enfrentado os mesmos erros e compartilhado soluções em fóruns como o Stack Overflow tornavam-se aliados improváveis.

A sensação de vitória, após horas de investigação digital para encontrar e corrigir o problema, era deliciosa; ainda assim, perceber que o erro era algo tão simples frequentemente deixava uma pontinha de frustração — e, em alguns casos, a sensação incômoda de incompetência.



Hoje, iniciantes no desenvolvimento web têm um aliado poderoso: a inteligência artificial (IA). Como um "amigo sabichão", ela aponta erros, sugere melhorias e até cria protótipos em instantes, acelerando a produtividade de novatos e veteranos.

Mas é preciso cautela: depender apenas de respostas prontas (nem sempre corretas ou seguras) pode atrapalhar a formação de profissionais críticos e independentes.

Este guia mostra como usar a IA como parceira no aprendizado de desenvolvimento web como uma parceira que apoia, mas não substitui.

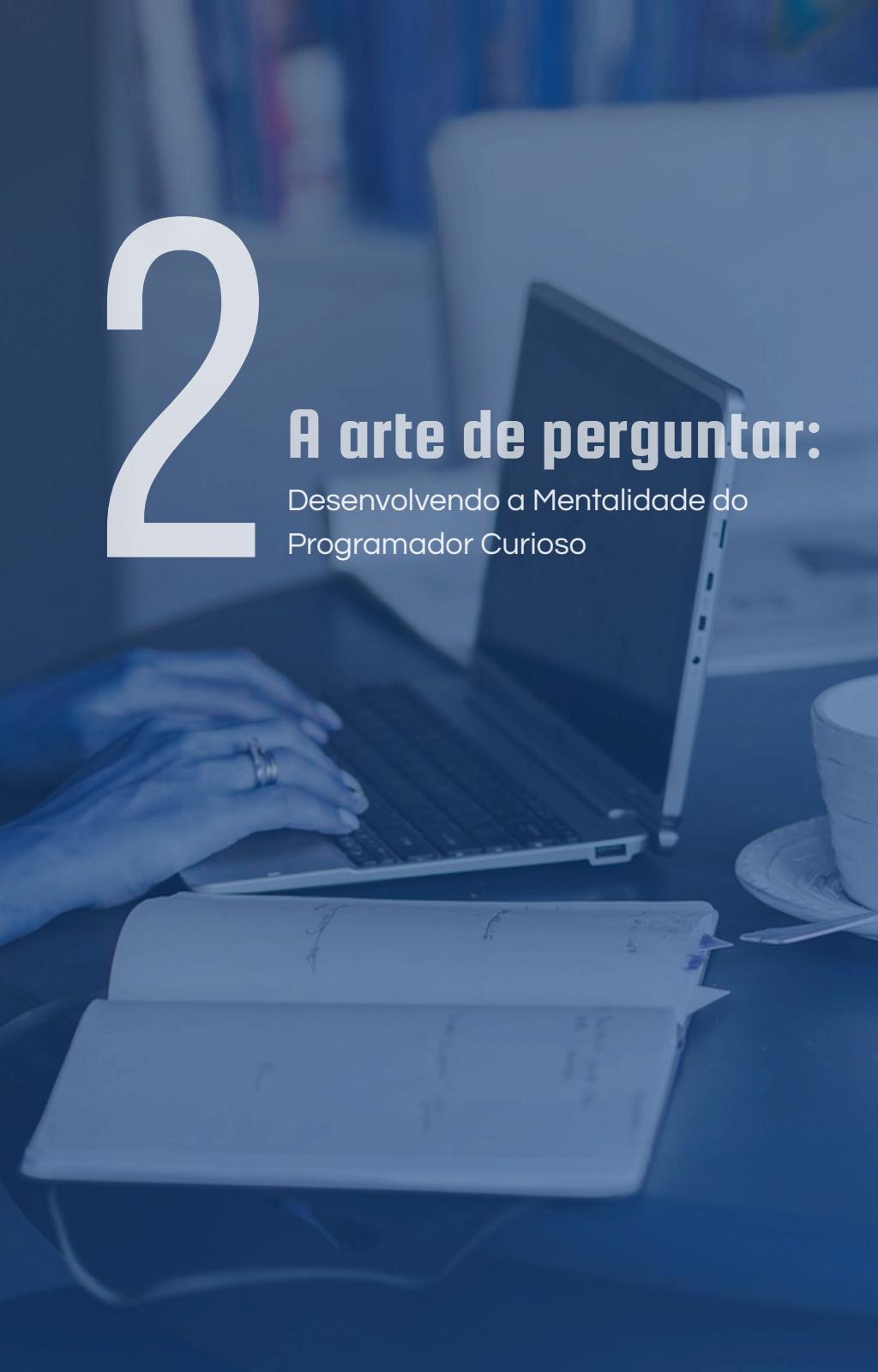
- Copiloto, não piloto: a IA sugere e acelera tarefas.
- Uso consciente: resultados dependem do programador.
- Aliada no aprendizado: potencializa, mas não dispensa pensar e testar.



2

A arte de perguntar:

Desenvolvendo a Mentalidade do
Programador Curioso



```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown" data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```



Entre a Pressa e o Pensar

Em uma era voltada à produtividade e métricas, em que os programadores estão constantemente lidando com demandas que parecem gritar "resolva isso agora!", é fácil recorrer ao "amigo sabichão" da inteligência artificial, quase um "estagiário premium" que entrega soluções em segundos para problemas que, no passado, poderiam levar horas de investigação e frustração.

Usar a IA sem reflexão pode parecer prático, mas traz riscos importantes para quem está aprendendo e programando no dia a dia.

- Dependência da IA:** respostas rápidas reduzem o exercício do raciocínio lógico.
- Perda de pensamento crítico:** entender o "porquê" e o "como" do código fica em segundo plano.
- Qualidade comprometida:** soluções podem ser funcionais, mas nem sempre eficientes ou seguras.

E os problemas disso? Ah, são diversos! Para não ficar só na teoria, é possível mostrar exemplos de como o uso não consciente da IA pode gerar soluções confusas, ineficientes ou até perigosas.

Armadilhas do Amigo Sabichão: Por Que Pensar Ainda É Necessário

Embora a IA pareça um gênio com respostas para tudo, confiar cegamente nela pode trazer algumas surpresas. Mesmo com acesso a uma quantidade enorme de dados e conhecimentos, nem sempre ela entrega soluções melhores que a própria lógica e experiência do programador.

Incorrências factuais

A IA às vezes se empolga se perde no personagem, e "inventando" funções, APIs ou bibliotecas que não existem.

Sugere uma função fetchDataAsyncPlus() que não existe em nenhuma biblioteca – parece válido, mas o código não compila.

```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown"
    data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```



Vulnerabilidades e Falhas de Segurança

Soluções geradas podem funcionar, mas nem sempre consideram usuários maliciosos.

Imagina que você pede para o usuário digitar a idade e simplesmente usa o valor sem conferir nada:

```
let idade = prompt("Digite sua idade:");
if (idade >= 18) {
    alert("Você pode entrar!");
} else {
    alert("Entrada não permitida!");
}
```

Pode parecer um cenário inofensivo, mas uma vez que o usuário pode digitar qualquer coisa ali, incluindo letras ou símbolos, não criar um código para melhor receber e filtrar esses dados que chegam pode dar brecha para comportamentos maliciosos que podem quebrar a lógica do seu programa.

Complexidade Desnecessária

Às vezes, a IA resolve o problema de forma exageradamente complexa ou lenta.

Você pede para checar se um valor existe no array, e ela monta uma busca binária complicada, quando um includes() já dava conta.



```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown" data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
    </div>
</div>
```

Falhas Lógicas

Respostas parecem corretas à primeira vista, mas podem falhar em casos de borda.

Um código que calcula descontos funciona para valores positivos, mas quebra completamente se o preço for zero ou negativo.

O ponto é: a IA é uma ferramenta poderosa, mas não substitui o pensamento crítico.

Analizar, testar e compreender o que ela gera mantém o programador no controle, transformando o "amigo sabichão" em um verdadeiro copiloto, e não em um substituto do próprio raciocínio.

De "Obtentor de Respostas" a "Formulador de Perguntas": Uma Mudança de Paradigma



Ok, já entendemos: não dá para simplesmente pedir a resposta pronta para a IA. Mas calma! A boa notícia é que isso não significa que a IA perdeu a graça.

Muito pelo contrário: podemos aproveitar toda a sua versatilidade e capacidade de simular diferentes "personagens" ou "contextos" para formular boas perguntas, estimular a reflexão e receber explicações mais didáticas ou lúdicas.

Ao invés disso:

Tente isso:

O que está errado aqui?

Mensagem de erro: TypeError: Assignment to constant variable.

Esse erro aparece quando tento mudar o valor de uma variável dentro do meu script.

Aja como um professor e me faça perguntas direcionadas para que eu mesma descubra o motivo do erro.

Observe que a mensagem foi destacada e o contexto do problema, explicado. Quanto mais detalhes sobre o erro você der, melhor será a resposta!

```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown"
    data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```

Essa abordagem transforma o aprendizado: você não recebe apenas a resposta, mas é guiado a pensar, descobrir e entender o que está acontecendo.

Sugestões de Prompts Legais/Úteis:

- "Você pode me explicar o conceito de ponteiros como se eu fosse uma criança de 5 anos?"
- "Me explique, usando uma analogia de jogos, como funcionam os laços de repetição."
- "Está dando o seguinte erro X, você poderia me fazer perguntas que me ajudem a encontrar problemas potenciais?"
- "Atue como um professor de programação e me explique linha a linha por que esse trecho não funciona."
- "Transforme este conceito em um pequeno desafio ou quiz interativo para testar meu entendimento."
- "Explique essa função usando um exemplo do dia a dia, como fazer compras ou organizar uma festa."

Sherlock dos bugs: A Arte de Interrogar Erros

Quem nunca ficou horas com cara de paisagem encarando uma mensagem de erro enigmática, que atire a primeira pedra.

Embora encorajemos sempre a reflexão, um bom uso de IA associado a ela pode te poupar bastante tempo (e estresse).

Mas a ideia não é só pedir a resposta pronta. Dá pra usar nosso estagiário premium como um parceiro de investigação, quase um detetive do seu lado. Aqui vão algumas dicas para isso.



```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown" data-wow-duration="500ms" data-wow-delay="200ms">
  <div class="counters-item">
    <span data-to="565">565</span>
```



Peça hipóteses (e como testá-las)

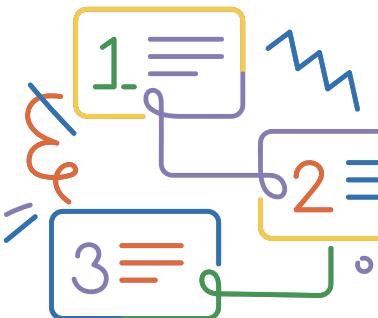
Trate o bug como um mistério a ser desvendado: apresente hipóteses e peça à IA possíveis explicações e também formas de confirmar ou descartar cada uma delas. Isso já dá meio caminho para descobrir o problema de verdade.

Chegue à solução junto

Em vez de “corrija para mim”, peça para a IA guiar você no raciocínio. Isso evita a dependência e ainda fortalece seu olhar crítico. Não é só consertar, é aprender com o erro.

Não pare na mensagem de erro

Em vez de pedir só uma explicação do erro, pergunte também quando ele costuma aparecer e quais as causas possíveis. Assim você entende o contexto, e não só a superfície.



3

Aprendizado x Produção

O papel da IA em cada fase





Tempo de Aprender & Tempo de Entregar

Embora muito úteis no processo de aprendizagem, as dicas descritas até aqui podem ser pouco realistas em um contexto de trabalho. Com prazos curtos e múltiplas demandas batendo à porta, nem sempre dá para investigar cada detalhe com calma.

Nessas horas, é natural recorrer à IA para obter uma solução direta ou até pedir que ela ajude a desenvolver pequenos trechos de código ou protótipos. Isso é totalmente válido no dia a dia de produção, quando já se tem base de programação e o objetivo é ganhar agilidade.

Já no início da jornada de aprendizado, a lógica é outra. Ter tempo para explorar, levantar hipóteses e “quebrar a cabeça” é essencial para consolidar o raciocínio e evitar pular etapas importantes.



Responsabilidade: Desenvolvimento Consciente

Em ambos os cenários, porém, há um ponto que nunca muda: **o desenvolvedor precisa ter pleno entendimento do código gerado**. Afinal, no fim das contas, a responsabilidade pelo que é colocado em um projeto é sempre sua.

A IA acelera, mas quem assina o código é você.



IA DETOX

As vezes vale dar um tempinho da IA e encarar o código sozinho, tipo “ficar off” pra lembrar que você ainda sabe pensar sem autocomplete! Depois disso, voltar pra IA é quase como reencontrar um amigo te apoia em seus trabalhos... mas agora você entende o que ele tá fazendo.

```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown"
    data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```

Desenvolvimento em Equipe: Manual da boa convivência

Ao trabalhar com desenvolvimento, em especial num contexto empresarial, você raramente irá programar sozinho. O sistema em construção provavelmente estará nas mãos de uma equipe considerável, na qual os membros atuarão em diferentes porções e branches.

Com isso em mente, é importante pensar na equipe e nos outros desenvolvedores que vão interagir com o código que você está criando. Algumas práticas ajudam a manter tudo organizado e seguro:

Ajustes de Padronização

a IA pode gerar códigos seguindo padrões diferentes dos usados pela sua equipe. Sempre revise e adapte o que ela sugere para que fique alinhado com as convenções do time.

Comentar o código gerado pela IA

Sempre que a IA criar um trecho de código, adicione comentários breves explicando o que ele faz e, se você ajustou algo, registre essas mudanças. Isso facilita para você e para outros desenvolvedores que venham trabalhar no mesmo projeto.

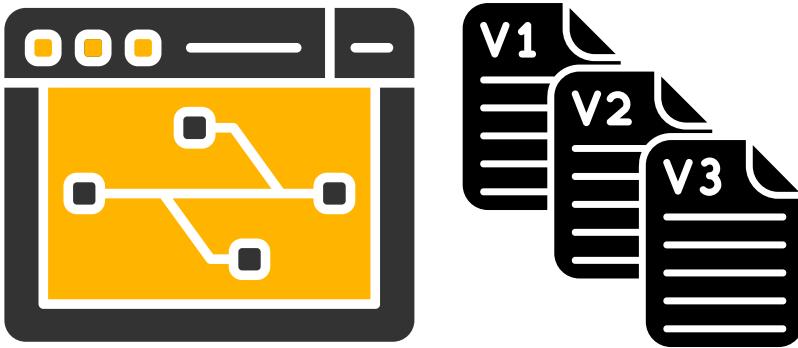


Não publicar sem testar

O ambiente de produção é como o palco principal: tudo que vai para lá precisa estar testado e aprovado. Não faça merge nem envie código da IA para produção sem uma revisão cuidadosa.

```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown"
    data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```

Outra prática fundamental é **manter um bom histórico de versões**. Ter registros claros (com branches, commits bem descritos e até tags quando necessário) permite voltar atrás caso algo dê errado e entender a evolução do projeto. Além disso, um histórico organizado facilita para que qualquer pessoa da equipe saiba o que foi testado, aprovado ou descartado, evitando retrabalho e garantindo mais segurança no processo de desenvolvimento.



4

IA na prática:

Conhecendo algumas ferramentas



```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown" data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```



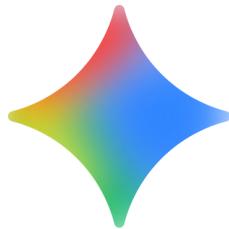
Quem é quem no mundo da IA?

Agora que já pegamos o jeito de como usar a Inteligência Artificial pra turbinar o aprendizado e o desenvolvimento, é hora de conhecer alguns dos "estagiários premium" que estão por ai!

Nesta seção, vamos trazer um pouco sobre desenvolvimento com IA na web e aninhado à IDE de desenvolvimento, destacando o que cada um faz de melhor e como podem te ajudar no dia a dia, sempre sem roubar o protagonismo do seu raciocínio, porque o cérebro humano ainda é o chefe por aqui.

Prompts na web: ideias do zero à tela em minutos

Utilizar aplicações web de IA para desenvolvimento é bastante comum. Algumas delas, como o Gemini, oferecem recursos específicos que facilitam o trabalho em colaboração com a IA, tornando o desenvolvimento mais ágil e produtivo.



The screenshot shows a web-based prototyping environment. On the left, there's a code editor window with the following CSS code:

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>FinomPéssimo - Protótipo</title>
7     <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800;900" rel="stylesheet">
8     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
9     <link href="https://fonts.googleapis.com/css2?family=Gstotic:ital@0,1" rel="stylesheet">
10    <style>
11        /* Palito de cores do cliente */
12        :root {
13            --dark: #1E1E1E;
14            --light: #F9F9F9;
15            --accent: #F7F7E6;
16        }
17        body {
18            font-family: 'Inter', sans-serif;
19            background-color: var(--light);
20            color: var(--dark);
21        }
22        .dark { background-color: var(--dark); }
23        .bg-light { background-color: var(--light); }
24        .bg-accent { background-color: var(--accent); }
25        .text-accent { color: var(--accent); }
26    </style>
```

On the right, there's a preview window showing a landing page with the title "Finance." and a main heading "Controle suas finanças com simplicidade e clareza." Below it, a section titled "Gráfico Financeiro" is partially visible.

Com o recurso "Canvas", dá pra trabalhar lado a lado com o seu estagiário premium, criando, ajustando e visualizando interfaces direto na própria ferramenta. É ótimo pra quem quer montar protótipos rápidos e de qualidade, testar variações ou fazer pequenos ajustes de layout sem precisar abrir mil abas ou ferramentas externas.

```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown" data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span data-to="565">565</span>
```

Recursos como esse podem ser extremamente úteis no contexto da elaboração rápida de protótipos sem gastar tanto tempo em desenvolvimento.

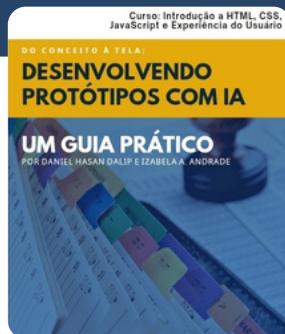
Para obter um resultado bastante próximo do desejado, salientamos novamente a importância de contextualizar bem o problema.

Dica de ouro:

Antes de permitir que a IA gere uma solução aos problemas, perguntamos se ela possui alguma dúvida. Isso é uma prática interessante para que porções que não tenham ficado claras sejam esclarecidas e evitem a utilização de inferências.

Para mais insights e dicas de prototipagem, recomendamos nosso guia prático "Prototipando com IA".

[ACESSE AQUI](#)



IA na web: algumas pedras no sapato

Mesmo super úteis, algumas coisas podem complicar a vida:

- **Límite de arquivos:** muitas plataformas permitem enviar poucos arquivos de cada vez, o que pode dificultar dar todo o contexto em projetos maiores.
- **Contexto parcial:** a IA nem sempre lembra tudo que você já compartilhou, então pode acabar escrevendo partes incompletas ou se confundindo em conversas longas.
- **Atualizações e memória:** cada "reset" ou atualização de chat pode fazer a IA esquecer detalhes importantes, exigindo que você repita informações.
- **Dependência de instruções claras:** quanto mais específico você for, melhor será o resultado. Deixar o pedido vago pode gerar respostas "meh".

```
</div>
<!-- end first count item -->
<!-- second count item -->
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown"
    data-wow-duration="500ms" data-wow-delay="200ms">
    <div class="counters-item">
        <span>565</span>
        <span data-to="565">565</span>
    </div>
</div>
```

IAs acopladas à IDE: Uma alternativa para melhor contextualização

Para projetos maiores, onde múltiplos arquivos são atualizados com frequência e o contexto completo precisa estar sempre em foco, vale a pena usar ferramentas de IA integradas ao ambiente de desenvolvimento, como o VS Code.

Essas IAs geralmente aparecem como extensões ou plug-ins e podem oferecer suporte direto enquanto você programa. Um exemplo famoso é o Copilot, que:

- Sugere alterações diretamente na plataforma, prontas para você aprovar;
- Em alguns casos, até testa o código sozinho para verificar se funciona.

Usar essas ferramentas pode tornar o trabalho mais ágil, mantendo você no controle e com menos risco de perder o contexto do projeto.