2ª EDIÇÃO — AMPLIADA



IA do Zero

Do **"nunca programei IA"** ao seu primeiro projeto — sem matemática assustadora.

---

23 capítulos · 6 módulos

exercícios com respostas comentadas
+ código de todos os projetos

Redes neurais
LLMs · RAG · Agentes
Projeto final na prática

Acompanha o curso em vídeo grátis do canal @CursoGratisdoBill no YouTube

MÓDULO 0 · CAPÍTULO 1

O que é IA, e por que todo mundo fala disso agora

Nesta aula você vai entender a diferença entre programar com regras e ensinar com exemplos, separar de vez IA, Machine Learning e Deep Learning, e explicar para qualquer pessoa por que a explosão da IA aconteceu agora e não nos anos 1950.

> 🍊 **ANALOGIA**
>
> Imagine ensinar uma criança pequena a reconhecer um cachorro. Você não entrega a ela uma lista de regras ("tem quatro patas, tem pelo, tem focinho comprido, late"). Você aponta na rua e diz: "olha, um cachorro". No outro passeio, aponta de novo. E de novo. Depois de ver o suficiente, ela reconhece sozinha até uma raça que nunca tinha visto, e não confunde cachorro com gato, mesmo que os dois tenham quatro patas e pelo. Ninguém programou uma regra na cabeça dela: o padrão saiu dos exemplos. **A IA moderna aprende exatamente assim: com exemplos, não com regras escritas à mão.**

Passo 1 · O jeito antigo: escrever as regras

Durante décadas, programar um computador significou uma coisa: escrever regras exatas. "Se isto acontecer, faça aquilo." Uma calculadora de imposto de renda é o exemplo perfeito. A lei diz que quem ganha até tanto paga tal alíquota, acima disso paga outra. O programador traduz a lei para código, linha por linha, e o computador obedece sem errar.

Esse jeito de programar funciona muito bem quando **as regras existem e cabem num papel**. Imposto, folha de pagamento, controle de estoque, o caixa do supermercado: tudo isso é regra.

Agora tente escrever as regras para outras tarefas:

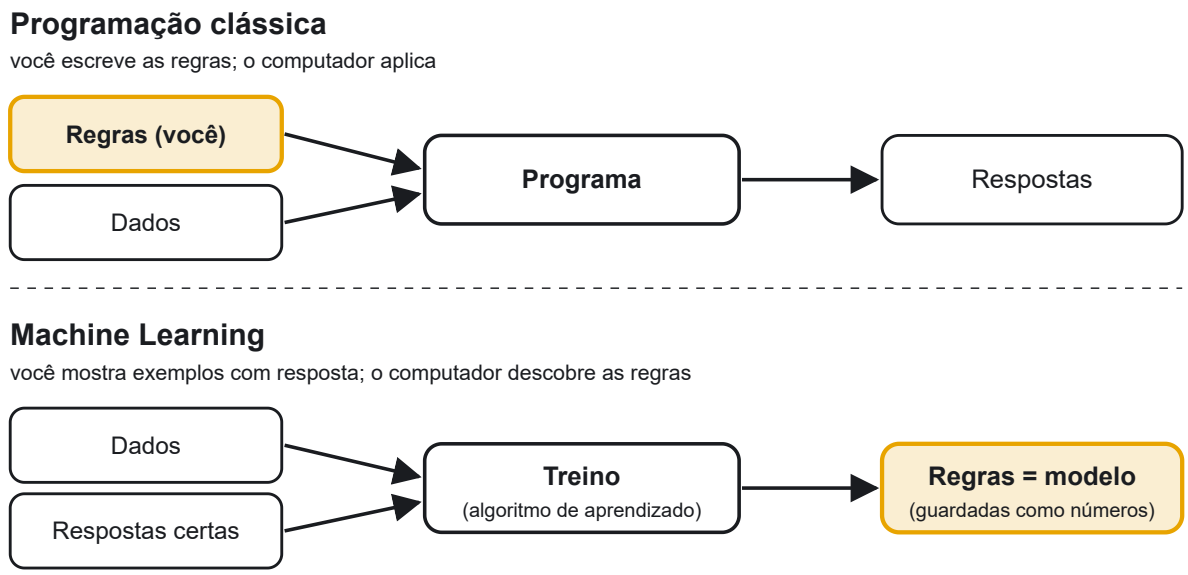
- **Reconhecer um rosto numa foto.** Qual é a regra? "Dois olhos acima de um nariz"? Funciona até aparecer alguém de perfil, de óculos escuros, com sombra no rosto, de chapéu.
- **Filtrar spam.** Você escreve "se o e-mail tiver GANHE DINHEIRO, é spam". Na semana seguinte os spammers escrevem "G4NHE D1NHEIRO". Você escreve outra regra. Eles mudam de novo. A brincadeira nunca acaba.
- **Entender ironia.** "Que ótimo, o ônibus atrasou de novo." As palavras são positivas; o sentido é negativo. Que regra pega isso?

Essas são tarefas que qualquer pessoa faz sem pensar e que ninguém consegue explicar em regras. O mundo real é bagunçado, ambíguo e cheio de exceções. Não existe papel grande o bastante.

Passo 2 · O jeito novo: mostrar exemplos

O **Machine Learning** (em português, *aprendizado de máquina*) virou o jogo de cabeça para baixo. Em vez de escrever a regra, você mostra ao computador muitos exemplos **já com a resposta certa**, e ele descobre a regra sozinho.

Quer um filtro de spam? Junte dez mil e-mails que pessoas já marcaram como "spam" ou "não é spam" e entregue ao algoritmo. Ele vai encontrar padrões que você nem imaginava: certas combinações de palavras, horários de envio, quantidade de links. Quando os spammers mudarem de tática, você não reescreve código: junta exemplos novos e treina de novo.



Programação clássica: regras e dados entram, respostas saem. Machine Learning: dados e respostas entram, regras saem

Repare no desenho. Na programação clássica, **você entrega as regras** e o computador produz as respostas. No Machine Learning, **você entrega as respostas** (nos exemplos) e o computador produz as regras. É a mesma máquina, trabalhando na direção oposta.

Passo 3 · O que é, afinal, um "modelo"

Você vai ouvir a palavra **modelo** o livro inteiro. Modelo é o que sai do treino: as regras que o computador descobriu, guardadas na forma de números. Não são regras escritas em português, como "acima de 120 gramas é laranja"; são centenas, milhões ou até bilhões de números ajustados de um jeito que, juntos, dão a resposta certa na maioria dos exemplos.

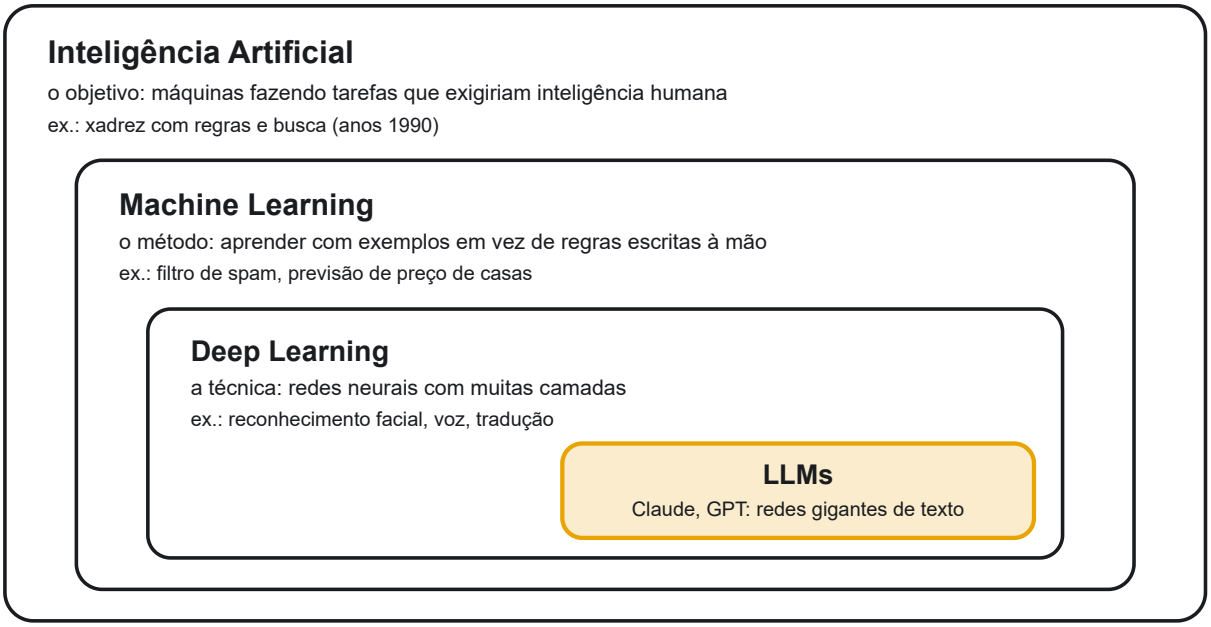
Duas palavras andam com ele:

- **Treino** (ou *treinamento*): a fase em que o modelo olha os exemplos e ajusta os seus números. É demorada e, nos modelos grandes, caríssima.
- **Uso** (os técnicos dizem *inferência*): a fase em que o modelo já pronto recebe um caso novo e dá a resposta. É rápida. Quando você conversa com o Claude, está na fase de uso: o treino aconteceu meses antes, nos computadores da empresa.

E a palavra mais importante de todas: **generalizar**. Um bom modelo não decora os exemplos; ele acerta casos que **nunca viu**, como a criança que reconhece uma raça nova de cachorro. Um modelo que só acerta o que já viu é como o aluno que decorou as respostas da prova do ano passado: tira dez no simulado e zero na prova de verdade. O Módulo 1 inteiro é sobre como medir isso.

Passo 4 · Três nomes, uma caixa dentro da outra

Três termos vão te acompanhar até o fim do livro, e muita gente usa os três como se fossem a mesma coisa. Não são. Pense em caixas, uma dentro da outra, como bonecas russas:



Caixas uma dentro da outra: Inteligência Artificial contém Machine Learning, que contém Deep Learning, que contém os LLMs

- **Inteligência Artificial (IA)** é a caixa maior: o objetivo de fazer máquinas realizarem tarefas que exigiriam inteligência humana. Cabe tudo aqui, inclusive os programas de xadrez dos anos 1990, que eram regras e buscas muito bem feitas, sem aprender nada com exemplos.
- **Machine Learning (ML)** é a caixa do meio: o *método* de chegar lá aprendendo com dados em vez de regras. É o que domina hoje.
- **Deep Learning** (*aprendizado profundo*) é a caixa menor: uma técnica de ML que usa **redes neurais** com muitas camadas, inspiradas de longe no cérebro. É a responsável pelos saltos impressionantes dos últimos anos: reconhecimento de voz, carros que leem placas, tradução automática. Você vai montar uma rede neural no Módulo 2.
- Lá dentro, uma caixinha que ficou famosa: os **LLMs** (*Large Language Models*, ou grandes modelos de linguagem), como o Claude e o GPT. São redes neurais gigantes treinadas com uma quantidade enorme de texto para prever a próxima palavra. É o assunto do Módulo 3.

Regra de bolso: **todo Deep Learning é Machine Learning, todo Machine Learning é IA, mas o contrário não vale.**

Passo 5 · Três jeitos de aprender

Dentro do Machine Learning existem três grandes famílias, e é útil saber o nome delas desde já:

1. **Aprendizado supervisionado:** os exemplos vêm com a resposta certa ("esta foto é de gato", "esta casa custou R$ 400 mil"). O modelo aprende a ir da pergunta à resposta. É o mais comum, e é o que você vai fazer no Mão na massa.
2. **Aprendizado não supervisionado:** os exemplos vêm sem resposta. O modelo procura grupos e padrões sozinho. Uma loja pode descobrir, por exemplo, que tem três tipos de cliente que ninguém tinha percebido.
3. **Aprendizado por reforço:** o modelo age, recebe uma recompensa ou uma punição, e vai aprendendo a estratégia que rende mais pontos. É como se ensina um computador a jogar videogame, e também uma parte de como os LLMs aprendem a ser úteis e educados.

Passo 6 · Por que a explosão veio agora

A teoria das redes neurais é dos anos 1950. Então por que a IA só virou assunto de almoço de domingo nos últimos anos? Porque três coisas, que avançavam separadas, se encontraram:

1. **Dados.** A internet produziu uma montanha de texto, foto, vídeo e áudio. Modelos aprendem com exemplos, e de repente havia exemplos aos bilhões.
2. **Poder de processamento.** As placas de vídeo (as **GPUs**), feitas para videogame, se mostraram perfeitas para a conta que as redes neurais fazem: milhões de multiplicações ao mesmo tempo. Ficaram absurdamente rápidas, e as empresas passaram a juntar dezenas de milhares delas num só lugar.
3. **Um algoritmo melhor.** Em 2017, pesquisadores do Google publicaram um artigo apresentando o **Transformer**, uma arquitetura de rede neural muito boa em entender texto e fácil de treinar em muitas GPUs ao mesmo tempo. Ele é o motor de todos os LLMs atuais. (O "T" de GPT é de Transformer.)

Quando os três se encontraram, modelos que eram brinquedo de laboratório viraram ferramentas que escrevem, programam, traduzem e conversam. Em novembro de 2022 o ChatGPT foi aberto ao público e ganhou 100 milhões de usuários em dois meses. A tecnologia já vinha amadurecendo havia anos; o que mudou foi que qualquer pessoa passou a poder conversar com ela.

Passo 7 · O que a IA faz bem e o que ela não faz

Entender IA também é saber os limites dela, e isso vai te poupar de muita decepção:

- **Ela gera, não busca.** Quando você pergunta algo ao Claude, ele não procura a resposta num banco de respostas prontas. Ele *gera* o texto, palavra por palavra, com base nos padrões que aprendeu. Por isso consegue escrever um poema inédito sobre o seu gato.
- **Por isso mesmo, ela pode errar com toda a confiança.** Se o padrão mais provável for uma informação errada, ela escreve a informação errada com a mesma segurança. Isso tem nome: **alucinação**. No Módulo 3 você vai ver como reduzir o problema dando à IA os seus próprios

documentos.

- **Ela não "pensa como gente".** Ela é ótima em padrões, mas não tem experiência do mundo, não sabe o que aconteceu depois do treino (a menos que alguém conte) e não tem intenção própria.
- **Nem tudo precisa de IA.** Se a regra existe e cabe num papel (o imposto, o cálculo do frete, a validação de um CPF), escreva a regra. É mais barato, mais rápido e acerta sempre. Use ML quando a regra não cabe no papel.

🛠️ Mão na massa

Antes do código, um experimento de dois minutos. Abra o Claude (claude.ai) ou o ChatGPT e peça:

> Explique o que é machine learning para uma criança de 10 anos.

Depois peça de novo, com as mesmas palavras, numa conversa nova. As respostas não vão ser idênticas. É a prova de que o texto não estava guardado em lugar nenhum: foi gerado na hora. Você acabou de usar um modelo de Deep Learning.

Agora o código. Você ainda não instalou nada (isso é o capítulo 2), então leia com calma e volte para rodar depois. Se já tiver Python, rode agora. O programa resolve um problema pequeno de dois jeitos: dizer se uma fruta é **laranja** ou **limão** olhando só o peso e o diâmetro. O primeiro jeito é uma regra que eu escrevi; o segundo **aprende com exemplos**, calculando a "fruta média" de cada tipo e respondendo com o tipo mais parecido.

```
"""Capítulo 1 - Regras escritas à mão x aprender com exemplos.

Problema: dizer se uma fruta é LARANJA ou LIMÃO olhando só dois números:
o peso (em gramas) e o diâmetro (em centímetros).

Rodar:  python regras_vs_exemplos.py
Não precisa instalar nada: só Python puro.
"""

# ------------------------------------------------------------
# Jeito 1: programação clássica. EU escrevo a regra.
# ------------------------------------------------------------
def classificar_por_regra(peso, diametro):
    """Regra que eu inventei olhando umas frutas na feira."""
    if peso > 120:
        return "laranja"
    return "limão"


# ------------------------------------------------------------
# Jeito 2: aprendizado de máquina. Os EXEMPLOS ensinam a regra.
# ------------------------------------------------------------
# Cada exemplo: (peso em g, diâmetro em cm, resposta certa).
# Esses são os "dados de treino": frutas que alguém já identificou.
EXEMPLOS = [
    (180, 8.0, "laranja"),
    (210, 8.5, "laranja"),
```

```
    (160, 7.5, "laranja"),
    (230, 9.0, "laranja"),
    (70, 5.0, "limão"),
    (90, 5.5, "limão"),
    (60, 4.8, "limão"),
    (100, 6.0, "limão"),
]


def treinar(exemplos):
    """'Aprende' calculando a fruta média de cada tipo.

    O resultado (o "modelo") é só um dicionário de médias:
    {"laranja": (peso_medio, diametro_medio), "limão": (...)}.
    Ninguém escreveu "acima de 120 g é laranja": as médias saem dos dados.
    """
    somas = {}  # rotulo -> [soma_peso, soma_diametro, quantidade]
    for peso, diametro, rotulo in exemplos:
        if rotulo not in somas:
            somas[rotulo] = [0.0, 0.0, 0]
        somas[rotulo][0] += peso
        somas[rotulo][1] += diametro
        somas[rotulo][2] += 1

    modelo = {}
    for rotulo, (soma_peso, soma_diam, qtd) in somas.items():
        modelo[rotulo] = (soma_peso / qtd, soma_diam / qtd)
    return modelo


def distancia(a, b):
    """Distância entre duas frutas, como se fossem pontos num mapa.

    Dividimos o peso por 30 para ele não 'pesar' mais que o diâmetro
    (gramas são números bem maiores que centímetros).
    """
    d_peso = (a[0] - b[0]) / 30
    d_diam = a[1] - b[1]
    return (d_peso**2 + d_diam**2) ** 0.5


def classificar_por_exemplos(modelo, peso, diametro):
    """Responde com o tipo cuja fruta média está mais perto."""
    fruta = (peso, diametro)
    return min(modelo, key=lambda rotulo: distancia(fruta, modelo[rotulo]))


# ---------------------------------------------------------------
# Comparando os dois jeitos
# ---------------------------------------------------------------
if __name__ == "__main__":
    modelo = treinar(EXEMPLOS)
    print("O que o modelo aprendeu (fruta média de cada tipo):")
    for rotulo, (peso, diam) in modelo.items():
        print(f"  {rotulo:8} -> {peso:.0f} g, {diam:.1f} cm")

    # Frutas novas, que nenhum dos dois jeitos viu antes.
```

```
    # As duas últimas são limões grandes; a última é um limão-siciliano.
    frutas_novas = [
        (200, 8.2, "laranja"),
        (80, 5.2, "limão"),
        (125, 6.1, "limão"),
        (150, 6.8, "limão"),
    ]

    print("\nFrutas novas:")
    for peso, diam, certo in frutas_novas:
        regra = classificar_por_regra(peso, diam)
        aprendido = classificar_por_exemplos(modelo, peso, diam)
        print(f"  {peso} g, {diam} cm (é {certo}): "
              f"regra diz {regra}, modelo diz {aprendido}")

    # O mundo mudou: a feira passou a vender limão-siciliano.
    # Na programação clássica eu teria de reescrever a regra à mão.
    # No aprendizado, eu só acrescento exemplos e treino de novo.
    sicilianos = [(140, 6.5, "limão"), (155, 6.9, "limão"),
                  (165, 7.0, "limão"), (150, 6.6, "limão")]
    modelo2 = treinar(EXEMPLOS + sicilianos)
    print("\nDepois de treinar de novo com 4 limões-sicilianos:")
    print(f"  150 g, 6.8 cm agora é: {classificar_por_exemplos(modelo2, 150, 6.8)}")
    print(f"  200 g, 8.2 cm continua: {classificar_por_exemplos(modelo2, 200, 8.2)}")
```

O arquivo está em `projetos/cap01/regras_vs_exemplos.py`. Rodando com `python regras_vs_exemplos.py`, a saída é:

```
O que o modelo aprendeu (fruta média de cada tipo):
  laranja  -> 195 g, 8.2 cm
  limão    -> 80 g, 5.3 cm

Frutas novas:
  200 g, 8.2 cm (é laranja): regra diz laranja, modelo diz laranja
  80 g, 5.2 cm (é limão): regra diz limão, modelo diz limão
  125 g, 6.1 cm (é limão): regra diz laranja, modelo diz limão
  150 g, 6.8 cm (é limão): regra diz laranja, modelo diz laranja

Depois de treinar de novo com 4 limões-sicilianos:
  150 g, 6.8 cm agora é: limão
  200 g, 8.2 cm continua: laranja
```

Leia a saída com calma, porque ela conta a história do capítulo inteiro:

- **O "modelo" é só um punhado de números** (duas médias). Ninguém digitou esses números: eles saíram dos exemplos.
- **No limão de 125 g, a regra erra e o modelo acerta.** A regra olhava só o peso; o modelo usa as duas medidas ao mesmo tempo, sem ninguém mandar.
- **No limão-siciliano, os dois erram.** Ele é diferente de tudo o que o modelo já viu. Modelos não fazem milagre: só sabem o que os exemplos ensinaram.

- **A correção não exigiu mexer em nenhuma linha da lógica.** Bastou acrescentar exemplos e treinar de novo. Esse é o superpoder do Machine Learning, e também a sua fraqueza: o modelo é tão bom quanto os dados que recebeu.

Os modelos de verdade usam milhares de medidas em vez de duas e bilhões de números em vez de quatro, mas a ideia é exatamente esta.

⚠️ Erros comuns

1. **Achar que IA é sinônimo de ChatGPT.** O ChatGPT e o Claude são um tipo de IA (LLMs), a caixinha mais de dentro. O filtro de spam do seu e-mail, a recomendação de filmes e o desbloqueio facial do celular também são IA. *Solução:* lembre das caixas: IA ⊃ ML ⊃ Deep Learning ⊃ LLMs.
2. **Achar que a IA "consulta" a resposta em algum lugar.** Ela gera o texto com base em padrões; não abre um banco de respostas. *Solução:* trate o que ela diz como um rascunho de alguém muito lido, não como uma enciclopédia. Confira dados, datas e números importantes.
3. **Acreditar que, se a resposta soa confiante, está certa.** A alucinação vem com o mesmo tom seguro de uma resposta correta. *Solução:* peça as fontes, compare com documentos seus e, quando o erro custar caro, verifique sempre.
4. **Usar IA onde uma regra simples resolve.** Validar CPF, calcular frete, somar uma nota fiscal: tudo isso tem regra exata. Um modelo vai ser mais lento, mais caro e às vezes errar. *Solução:* pergunte primeiro "consigo escrever a regra num papel?". Se sim, escreva a regra.
5. **Achar que mais dados sempre resolvem.** Dados ruins, repetidos ou tendenciosos ensinam coisas ruins. Se todos os seus exemplos de "limão" forem verdes e pequenos, o modelo nunca vai entender o siciliano. *Solução:* exemplos variados e corretos valem mais que exemplos em grande quantidade.

📌 Resumo

- Programação clássica: **você escreve as regras**. Machine Learning: **você mostra exemplos com a resposta** e o computador descobre as regras.
- **Modelo** é o resultado do treino: regras guardadas como números. **Treino** é quando ele aprende; **uso** (inferência) é quando ele responde.
- Um bom modelo **generaliza**: acerta casos que nunca viu, em vez de decorar os exemplos.
- **IA ⊃ Machine Learning ⊃ Deep Learning ⊃ LLMs**, uma caixa dentro da outra.
- Três famílias de aprendizado: **supervisionado** (com resposta), **não supervisionado** (sem resposta) e **por reforço** (com recompensa).
- A explosão veio do encontro de **dados + GPUs + Transformer (2017)**.
- LLMs **geram** texto, não buscam; por isso podem **alucinar** com confiança.
- Se a regra cabe num papel, escreva a regra. IA é para quando ela não cabe.

📝 Exercícios

1. **(Fácil)** Com as suas palavras, em no máximo cinco linhas, explique a diferença entre "programar com regras" e "aprender com exemplos". Use um exemplo que não esteja neste capítulo.
2. **(Fácil)** Para cada tarefa, diga se você usaria **regras** ou **Machine Learning**, e por quê: (a) calcular o troco no caixa; (b) sugerir a próxima música numa playlist; (c) verificar se uma data é válida; (d) identificar se um comentário numa loja online é elogio ou reclamação; (e) prever quanto uma padaria vai vender de pão amanhã.
3. **(Médio)** Classifique cada item na caixa mais de dentro que ele ocupa (IA, ML, Deep Learning ou LLM): (a) um programa de xadrez de 1997 que só usa regras e busca; (b) o Claude; (c) um modelo que prevê o preço de casas a partir de uma tabela, aprendendo médias; (d) o reconhecimento facial do celular, feito com uma rede neural de muitas camadas.
4. **(Médio)** Rode `regras_vs_exemplos.py` e mude a regra do Jeito 1 até ela acertar as quatro frutas novas. Depois responda: a sua regra nova acertaria um limão-siciliano de 170 g e 7,2 cm? O que isso diz sobre escrever regras à mão?
5. **(Desafio)** Acrescente uma terceira fruta ao programa: **tangerina** (exemplos sugeridos: 110 g e 6,5 cm; 120 g e 6,8 cm; 100 g e 6,2 cm; 130 g e 7,0 cm). Quantas linhas da função `treinar` você precisou mudar? E da função `classificar_por_exemplos`? Teste com uma tangerina nova de 115 g e 6,6 cm.

As respostas comentadas estão no fim do livro, em Respostas dos exercícios · Capítulo 01.

FIM DO CAPÍTULO 1

Você leu o capítulo 1. Faltam 114.

Esse foi só o começo. A Trilha IA reúne tudo o que vem depois, do mesmo jeito: passo a passo, com analogia antes de cada ideia.

5
cursos em ebook

115
capítulos

30
projetos com código

Os 5 cursos da Trilha

- IA do Zero (o capítulo que você acabou de ler)
- Engenharia de IA
- IA multimodal
- Produto de IA
- Agentes que trabalham por você

Tudo por R$ 37.

Quero a Trilha IA completa

iadozero.seenupdigital.com

No ebook completo, os exercícios deste capítulo têm respostas comentadas.