

// laboratório codex 03 · economia empresarial

python + ia + app

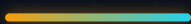
## >\_ Laboratório Codex 03

# App de precificação em Python, com IA

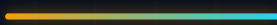
Envie o histórico de vendas e descubra o preço que dá mais lucro:  
um app com demanda estimada, curvas de lucro, ponto de equilíbrio,  
inflação real e relatório em Excel.

// apostila do aluno III · o projeto em etapas

01 Histórico de vendas



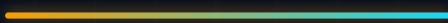
03 Curvas de lucro



05 Inflação e cenários



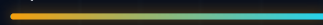
07 Elasticidade cruzada



02 Demanda e preço ótimo



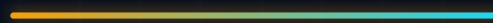
04 Upload e custos



06 Recomendação em Excel



08 Apresentação



**Prof. Caio Haffner**

Mestre em Desenvolvimento Econômico · Unicamp

# 1 Como usar esta apostila

## ✖ LEIA TUDO, NA ORDEM

Leia esta apostila **desde a primeira palavra** e faça tudo o que ela manda, na ordem. Não pule direto para os exercícios: cada página prepara a seguinte. Se ainda não fez a **Apostila I** (inclusive a página "PARE. Leia antes de tudo"), comece por ela.

Esta é a **Apostila III**, que continua a I (instalação e primeiros passos) e a II (exercícios no terminal). Agora você vai construir **um projeto completo**: um **app de precificação** que lê o histórico de vendas, estima a demanda, mostra a curva de lucro, o ponto de equilíbrio e o efeito da inflação, e gera um relatório em Excel.

### 1. Dados

Histórico de vendas em CSV

### 2. Cálculos

Demanda, preço e equilíbrio

### 3. Gráficos

Lucro e equilíbrio

### 4. App

Upload e custos na tela

### 5. Relatório

Inflação e Excel

## ★ O PROJETO: APP DE PRECIFICAÇÃO

O dono da empresa envia a planilha de vendas e informa os custos. O app estima a demanda, encontra o preço que maximiza o lucro, desenha as curvas de receita, custo e lucro, mostra o ponto de equilíbrio, simula a inflação com o IPCA do Banco Central e entrega uma recomendação em Excel.

## Ao final, você deverá conseguir

| Habilidade               | Como saber que consegui                               | Ok                       |
|--------------------------|-------------------------------------------------------|--------------------------|
| Preparar dados de vendas | Monto e leio o histórico em CSV.                      | <input type="checkbox"/> |
| Estimar a demanda        | Ajusto a reta e mostro o $R^2$ no app.                | <input type="checkbox"/> |
| Encontrar o preço ótimo  | Mostro a curva de lucro e o ponto máximo.             | <input type="checkbox"/> |
| Mostrar o equilíbrio     | Gráfico de receita e custo com o ponto de equilíbrio. | <input type="checkbox"/> |
| Usar dados reais         | Simulo custos com o IPCA do Banco Central.            | <input type="checkbox"/> |
| Recomendar               | O app escreve a recomendação e gera o Excel.          | <input type="checkbox"/> |

1

#### REGRA DE OURO

Você é responsável pelo código que entrega, mesmo quando a IA o escreveu. **Não entregue código que você não consegue explicar.** No projeto, vale mais uma: **a cada etapa, o app precisa continuar funcionando.** Teste antes de seguir.

## 2 Antes de começar

Você precisa do que montou nas Apostilas I e II: VS Code, Python e Codex funcionando. Se faltar algo, volte à **Apostila I, seções 3 a 7**. A novidade são três **bibliotecas** que dão ao Python gráficos, telas e planilhas.

### Instale as bibliotecas (uma vez só)

No terminal do VS Code, digite:

TERMINAL

```
python -m pip install streamlit matplotlib pandas openpyxl
```

| Biblioteca        | Para que serve                                               |
|-------------------|--------------------------------------------------------------|
| <b>pandas</b>     | Trabalhar com tabelas (como uma planilha dentro do Python).  |
| <b>matplotlib</b> | Fazer gráficos: linhas, barras, pizza, dispersão.            |
| <b>streamlit</b>  | Transformar o programa num aplicativo que abre no navegador. |
| <b>openpyxl</b>   | Salvar e ler arquivos do Excel (.xlsx).                      |

Para conferir, digite:

TERMINAL

```
python -c "import streamlit, matplotlib, pandas, openpyxl; print('tudo certo!')"
```



#### CHECKPOINT

O terminal mostrou `tudo certo`.



#### SE DER ERRO

Se aparecer "pip não é reconhecido" ou "python não é reconhecido", troque `python` por `py` (como na Apostila I). Se a instalação demorar, é normal: são bibliotecas grandes. Se travar, copie a mensagem e use o prompt universal de diagnóstico.

### Crie a pasta do projeto

Crie a pasta `Laboratorio_Codex_EE3` em Documentos e abra no VS Code. Ao longo das etapas, ela vai ficar assim:

#### TERMINAL

```
Laboratorio_Codex_EE3/  
├─ dados/           ← arquivos de dados (CSV)  
├─ calculos.py      ← as contas (funções)  
├─ graficos.py      ← os gráficos  
├─ app.py           ← a tela (Streamlit)  
└─ relatorio.xlsx   ← gerado pelo app
```

#### **i** GUARDE VERSÕES

Ao terminar cada etapa com o app funcionando, copie a pasta inteira com o nome da etapa (por exemplo, `Laboratorio_Codex_EE3_etapa2`). Se algo quebrar depois, você volta.

## **3** Seu primeiro app

Antes do projeto, um teste rápido das ferramentas novas. Crie um arquivo `teste_app.py` com o código abaixo. Você pode digitar ou pedir ao Codex.

```
import streamlit as st  
import matplotlib.pyplot as plt  
  
st.title("Meu primeiro app")  
valor = st.number_input("Valor aplicado (R$)", value=1000.0)  
taxa = st.slider("Taxa ao mês (%)", 0.0, 2.0, 1.0)  
meses = st.slider("Meses", 1, 60, 12)  
  
saldos = [valor * (1 + taxa / 100) ** m for m in range(meses + 1)]  
fig, ax = plt.subplots()  
ax.plot(saldos)  
ax.set_xlabel("Mês")  
ax.set_ylabel("R$")  
st.pyplot(fig)  
st.metric("Valor final", f"R$ {saldos[-1]:,.2f}")
```

Para abrir o app, digite no terminal:

#### TERMINAL

```
python -m streamlit run teste_app.py
```



### CHECKPOINT

O navegador abriu um app com o título "Meu primeiro app", dois controles deslizantes e um gráfico. Mexa nos controles: o gráfico muda na hora.

## Complete a tela

### TERMINAL

Valor aplicado: 1000 · Taxa ao mês: 1,00 · Meses: 12

Valor final: R\$ \_\_\_\_\_



### ARMADILHA: O FORMATO DO NÚMERO

O app mostra o valor no formato americano (vírgula no milhar e ponto nos centavos). Peça ao Codex: *"mostre os valores no formato brasileiro, com ponto no milhar e vírgula nos centavos"*. Você vai usar isso em todo o projeto.



### COMO O STREAMLIT FUNCIONA

Cada vez que você mexe num controle, o Streamlit roda o programa inteiro de novo, de cima para baixo. Por isso o gráfico se atualiza sozinho. Para fechar o app, volte ao terminal e aperte `Ctrl` + `C`.



### LEMBRETES DAS APOSTILAS ANTERIORES

Peça sempre um **plano** antes do código · teste com contas feitas por você · peça o **diagnóstico antes da correção** · aceite vírgula ou ponto · só aprove o que entender.

## 4

## Etapa 1 · O histórico de vendas

O app começa pelos dados que toda empresa tem: preços praticados e quantidades vendidas.

### PROMPT 1

Crie dados/vendas.csv com as colunas mes, preco, quantidade e 12 meses de vendas de caixas de brigadeiro da Doceria Bom Gosto, com preços entre R\$ 10 e R\$ 18 e quantidades coerentes com uma demanda que cai quando o preço sobe (com algum ruído). Inclua os 5 pontos da Apostila II (10→120, 12→110, 14→95, 16→85, 18→70). Em calculos.py, crie ler\_vendas(arquivo) com pandas, aceitando vírgula ou ponto e vírgula.



### CHECKPOINT

A função lê os 12 meses e mostra a tabela no terminal.

## 5 Etapa 2 · Demanda, preço e equilíbrio

As contas da Apostila II, agora como funções que recebem os dados da planilha.

### PROMPT 2

Em `calculos.py`, crie:

1. `estimar_demanda(df)`:  $a$  e  $b$  da reta  $Q = a + b \cdot P$  e o  $R^2$ ;
2. `lucro(preco, a, b, custo_variavel, custo_fixo)` e `preco_otimo(...)`: o preço que maximiza o lucro (fórmula) e uma conferência por busca entre o menor e o maior preço;
3. `equilibrio(custo_fixo, preco, custo_variavel)`: ponto de equilíbrio e margem de segurança.

Teste com os 5 pontos da Apostila II (resultado esperado:  $Q = 183,5 - 6,25 \cdot P$ ).

| Conferência                      | Esperado                   | Calculado |
|----------------------------------|----------------------------|-----------|
| Demanda (5 pontos)               | $Q = 183,5 - 6,25 \cdot P$ |           |
| $R^2$                            | 0,995                      |           |
| Preço ótimo (custo 10, fixo 200) | R\$ 19,68                  |           |



#### DE ONDE VEIO R\$ 19,68?

Com  $Q = 183,5 - 6,25 \cdot P$ ,  $A = 183,5$  e  $B = 6,25$ :  $P^* = (183,5 \div 6,25 + 10) \div 2 = 19,68$ . É diferente da Apostila II porque lá a demanda foi dada ( $200 - 8P$ ); aqui ela foi estimada pelos dados.

## Etapa 3 · As curvas de lucro e de equilíbrio

Dois gráficos que todo dono de empresa entende de primeira.

### PROMPT 3

Crie graficos.py com:

1. grafico\_demanda: os pontos do histórico e a reta estimada (preço no eixo vertical);
2. grafico\_lucro: o lucro para cada preço, com o preço ótimo marcado e o preço atual destacado;
3. grafico\_equilibrio: as retas de receita total e custo total pela quantidade, com o ponto de equilíbrio marcado e as áreas de prejuízo e lucro coloridas.



### CHECKPOINT

O gráfico de lucro tem forma de "montanha", com o pico no preço ótimo.

## 7

## Etapa 4 · O app com upload e custos na tela

O usuário envia a planilha, informa os custos e vê tudo mudar na hora.

### PROMPT 4

Crie app.py com Streamlit:

- st.file\_uploader para o CSV de vendas (com o arquivo de exemplo como padrão);
- barra lateral: custo variável, custo fixo e preço atual;
- no topo: st.metric com a demanda estimada ( $R^2$ ), o preço ótimo, o lucro no ótimo e o lucro no preço atual (com a diferença);
- abaixo, os três gráficos.

### TERMINAL

```
python -m streamlit run app.py
```

| Teste no app          | Preço ótimo | Lucro no ótimo |
|-----------------------|-------------|----------------|
| Custo variável R\$ 10 |             |                |
| Custo variável R\$ 12 |             |                |
| Custo fixo R\$ 500    |             |                |



### PARA PENSAR

Por que o custo fixo muda o lucro, mas não muda o preço ótimo?

## 8 Etapa 5 · Inflação e cenários

Os custos sobem com a inflação. O app busca o IPCA no Banco Central e mostra o novo preço ótimo.

### PROMPT 5

Acrescente uma aba "Inflação": busque o IPCA de 12 meses no Banco Central (série 13522, com plano B de 4,22%), aplique-o ao custo variável (e um percentual extra informado pelo usuário) e mostre o novo preço ótimo, o reajuste necessário e o efeito no lucro. Compare o reajuste com o IPCA: o preço sobe mais ou menos que a inflação?



### CHECKPOINT

A aba mostra o IPCA com a data e a fonte, e o novo preço ótimo.

## Etapa 6 · A recomendação em Excel

O app termina com uma recomendação escrita e um Excel para o dono guardar.

### PROMPT 6

Acrescente um botão "Baixar recomendação em Excel" com as abas:  
"Recomendação" (texto de 4 a 6 frases escrito pelo app, com o preço sugerido, o lucro esperado, o ponto de equilíbrio e o efeito da inflação), "Demanda" (histórico e reta),  
"Lucro por preço" (tabela) e "Premissas".



### CHECKPOINT

A recomendação cita os números do app e o Excel tem 4 abas.

## 10 Desafio extra · O concorrente baixou o preço

Na vida real, a demanda também depende do preço do concorrente: é a **elasticidade cruzada**.

### PROMPT 7

desafio

Acrescente ao CSV uma coluna `preco_concorrente` e estime uma demanda com dois preços ( $Q = a + b \cdot P + c \cdot P_{\text{concorrente}}$ ) por mínimos quadrados (`numpy.linalg.lstsq`).  
Mostre a elasticidade cruzada e simule: se o concorrente baixar o preço em 10%, qual o novo preço ótimo? Explique bens substitutos antes de escrever o código.

## 11 Use o Codex como revisor do projeto

### REVISAR A ORGANIZAÇÃO

Revise o meu projeto como um programador experiente revisaria o de um iniciante. Não faça alterações. Aponte: funções muito longas, código repetido, nomes pouco claros e cálculos misturados com a tela. Separe o que é obrigatório do que é opcional.

### REVISAR O CONTEÚDO

Revise os cálculos do meu projeto como um professor de Economia Empresarial. Não faça alterações. Confira a regressão, a fórmula do preço ótimo, o ponto de equilíbrio e o uso do IPCA e liste o que pode estar errado.



### GRÁFICO BONITO NÃO É GRÁFICO CERTO

Confira pelo menos um número de cada gráfico com a calculadora ou com a Apostila II. A IA erra eixos, unidades e somas com frequência.

## Entrega: apresente o seu projeto

Um projeto só existe quando alguém o vê funcionando. Prepare uma **apresentação de 3 minutos** e um registro para o seu portfólio.

### Roteiro de 3 minutos

| Tempo      | O que mostrar                                                                                   |
|------------|-------------------------------------------------------------------------------------------------|
| 30 s       | O problema que o app resolve e para quem.                                                       |
| 1 min 30 s | O app funcionando: mexa nos controles e mostre a curva de lucro mudando quando os custos sobem. |
| 30 s       | Um erro que a IA cometeu e como você descobriu.                                                 |
| 30 s       | O que você faria a seguir, se tivesse mais tempo.                                               |

### O que entregar

- ☐ A pasta do projeto, com todos os arquivos .py e os dados
- ☐ O relatório em Excel gerado pelo app
- ☐ Três prints do app (telas principais)
- ☐ Um arquivo LEIAME.md explicando o que o app faz e como rodar (peça um rascunho ao Codex e revise)
- ☐ O diário de prompts: pelo menos 5 prompts importantes e o que aconteceu



#### PARA O SEU PORTFÓLIO

Um post no LinkedIn com um vídeo curto do app funcionando ("Construí em Python, com ajuda de IA, um app de precificação para pequenas empresas") mostra a recrutadores algo que poucos estudantes têm: **uma ferramenta real feita por você.**

### Reflexão individual

1. Qual etapa foi a mais difícil, e como você a resolveu?

---



---

2. Em que momento a IA errou, e como você percebeu?

---



---

3. O que você aprendeu de Economia Empresarial construindo o app que não tinha aprendido só com a teoria?

## 13 Como você será avaliado

| Critério                                                  | 0                        | 1                    | 2                    | 3                                   |
|-----------------------------------------------------------|--------------------------|----------------------|----------------------|-------------------------------------|
| <b>O app funciona</b>                                     | Não abre                 | Abre com erros       | Funciona             | Funciona e trata entradas inválidas |
| <b>Conteúdo</b>                                           | Cálculos errados         | Alguns certos        | Quase todos certos   | Todos certos e conferidos           |
| <b>Gráficos</b>                                           | Nenhum                   | Sem títulos ou eixos | Corretos             | Corretos e fáceis de ler            |
| <b>Relatório</b>                                          | Não gera                 | Gera incompleto      | Gera completo        | Completo e organizado em abas       |
| <b>Uso do Codex</b>                                       | Não usou                 | Só copiou            | Usou com compreensão | Usou de forma crítica               |
| <b>Apresentação</b>                                       | Não apresentou           | Confusa              | Clara                | Clara, no tempo e com o erro da IA  |
| Registro da auditoria humana (opcional, vale ponto extra) | <input type="checkbox"/> |                      |                      |                                     |

Total máximo: 18 pontos, convertidos para a escala da disciplina.

## 14 Consulta rápida

### Comandos

#### TERMINAL

```
python -m pip install streamlit  
matplotlib pandas openpyxl  
python -m streamlit run app.py    # abre  
o app  
Ctrl + C                          #  
fecha o app
```

### Streamlit

|                                              |                           |
|----------------------------------------------|---------------------------|
| <code>st.title · st.header</code>            | Títulos                   |
| <code>st.number_input · st.slider</code>     | Entradas de número        |
| <code>st.selectbox · st.file_uploader</code> | Escolha e upload          |
| <code>st.metric · st.dataframe</code>        | Indicador e tabela        |
| <code>st.pyplot(fig)</code>                  | Mostrar um gráfico        |
| <code>st.tabs · st.sidebar</code>            | Abas e barra lateral      |
| <code>st.download_button</code>              | Botão para baixar o Excel |

### Glossário

#### Biblioteca

Código pronto que você instala para ganhar recursos (gráficos, telas).

#### pip

O instalador de bibliotecas do Python.

#### DataFrame

Tabela do pandas, com linhas e colunas.

#### App web

Programa que roda no navegador.

#### Precificação

Definição do preço de venda.

#### Curva de lucro

Lucro para cada preço possível.

#### Ponto de equilíbrio

Vendas em que o lucro é zero.

1

#### LEMBRE-SE

A IA é uma ferramenta. Conhecimento, julgamento, validação e responsabilidade continuam sendo de quem programa. Nada nesta apostila é recomendação profissional: é aprendizado.

**Todo material produzido com IA precisa de auditoria humana. Sempre.** A IA escreve rápido e com segurança, mas pode errar uma conta, inventar uma regra ou usar um dado desatualizado. E ela escreve o erro com a mesma confiança do acerto.

Esta apostila foi escrita com ajuda de IA e revisada pelo professor, conta por conta. Mesmo assim, algum erro pode ter escapado. Agora o auditor é você: encontre, comprove e relate.



### REGRA DO LABORATÓRIO: IA SEMPRE PASSA POR AUDITORIA HUMANA

A IA não sabe quando errou: o texto errado sai tão convincente quanto o certo.

Quem assina o resultado é uma pessoa: você no trabalho, o analista no relatório, o professor na apostila.

Auditar não é desconfiar da ferramenta. É a etapa que transforma um rascunho da IA num material confiável.

Vale para tudo: código, contas, textos, gráficos e esta apostila.

## O que procurar

| Tipo de erro        | Exemplo                                               | Como conferir                                                  |
|---------------------|-------------------------------------------------------|----------------------------------------------------------------|
| Conta ou número     | Um total que não fecha, uma porcentagem errada        | Refaça na calculadora                                          |
| Regra da disciplina | Fórmula trocada, conceito mal explicado               | Compare com o livro ou o material da aula                      |
| Referência          | "Veja a seção 9", quando o assunto está na 11         | Vá até a seção indicada                                        |
| Prompt              | Um prompt que faz o Codex errar sempre do mesmo jeito | Rode o prompt duas vezes                                       |
| Dado desatualizado  | Selic, IPCA, regra de imposto                         | Confira no site oficial (Banco Central, IBGE, Receita Federal) |
| Texto               | Frase confusa, erro de português                      | Leia o trecho em voz alta                                      |

## Como auditar com o Codex

### PROMPT DE AUDITORIA

use com cuidado

Vou colar um trecho da minha apostila de [DISCIPLINA].

Atue como revisor. Não reescreva o texto.

1. refaça cada conta e mostre o cálculo;
2. aponte afirmações que podem estar erradas ou desatualizadas;
3. para cada problema, diga como eu posso confirmar sozinho.

Trecho: [COLE AQUI]



### A IA TAMBÉM ERRA NA AUDITORIA

O Codex pode apontar erros que não existem. Só relate o que você confirmou com uma conta, uma fonte oficial ou um teste.

## Registro da auditoria

| # | Seção e página | O que está escrito | O que deveria ser | Como confirmei |
|---|----------------|--------------------|-------------------|----------------|
| 1 |                |                    |                   |                |
| 2 |                |                    |                   |                |
| 3 |                |                    |                   |                |

## Como enviar

- 1 Preencha a tabela acima.
- 2 Envie cada erro pelo formulário: **laboratorio-codex.pages.dev/auditoria**
- 3 Aguarde a confirmação do professor.



### O QUE VOCÊ GANHA

Cada erro confirmado vale **+1 ponto extra** na avaliação (até 3 pontos), e o seu nome entra na **lista de revisores** da próxima versão da apostila. Se duas pessoas relatarem o mesmo erro, vale o primeiro envio.