

// laboratório codex 03 · economia · micro e macro

python + ia + app

>_ Laboratório Codex 03

Modelos econômicos em Python, com IA

As curvas da Economia ganham vida: mexa nos parâmetros e veja o equilíbrio, os custos e o multiplicador se moverem num app com dados reais do Banco Central.

// apostila do aluno III · o projeto em etapas

01 Modelos em funções

03 Abas de micro

05 O app que explica

07 Imposto e peso morto

02 Curvas no gráfico

04 Multiplicador e inflação

06 Relatório 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 **laboratório de modelos econômicos**: um app em que o usuário mexe nos parâmetros e vê as curvas de oferta, demanda, custos e multiplicador se moverem, com dados reais da economia.

1. Modelos

Funções dos modelos

2. Gráficos

Curvas e equilíbrio

3. Micro

Abas de microeconomia

4. Macro

Multiplicador e inflação real

5. Relatório

Explicações e Excel



★ O PROJETO: LABORATÓRIO DE MODELOS ECONÔMICOS

Cada aba do app é um modelo da Apostila II, agora em gráfico: oferta e demanda com tabelamento, elasticidade e receita, custos e lucro máximo, multiplicador e inflação real do Banco Central. O app explica o que acontece em cada mudança e gera um relatório em Excel.

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Programar modelos	Escrevo cada modelo como função testada.	☐
Desenhar curvas	Mostro oferta, demanda e equilíbrio num gráfico.	☐
Simular políticas	Mostro o efeito de um tabelamento ou de um gasto do governo.	☐
Usar dados reais	Trago o IPCA do Banco Central para um gráfico.	☐
Construir um app	Cada modelo numa aba, com controles.	☐
Explicar	O app escreve o que aconteceu em linguagem simples.	☐

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
pandas	Trabalhar com tabelas (como uma planilha dentro do Python).
matplotlib	Fazer gráficos: linhas, barras, pizza, dispersão.
streamlit	Transformar o programa num aplicativo que abre no navegador.
openpyxl	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_EC03** em Documentos e abra no VS Code. Ao longo das etapas, ela vai ficar assim:

TERMINAL

```
Laboratorio_Codex_EC03/  
├─ 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_EC03_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 · Os modelos em funções

Primeiro, os modelos da Apostila II viram **funções** num só arquivo, testadas com os números que você já conhece.

PROMPT 1

Crie calculos.py com funções para:

- equilibrio(a, b, c, d): preço e quantidade de equilíbrio de $Q_d = a - bP$ e $Q_s = c + dP$;
- controle_preco(a, b, c, d, preco): Q_d , Q_s e escassez ou excedente;
- elasticidade(p1, q1, p2, q2): pelo ponto médio, com a classificação;
- custos(cf, c1, c2, preco, qmax): tabela de CT, CMe, CMg, receita e lucro;
- multiplicador(pmgc, delta_g, rodadas): a renda rodada a rodada.

Escreva testes simples (assert) com os números da Apostila II.

Teste	Apostila II	A função devolveu
Equilíbrio ($Q_d = 100 - 2P$, $Q_s = 20 + 2P$)	$P = 20$, $Q = 60$	
Elasticidade (10→12, 100→80)	-1,22	
Lucro máximo ($CT = 100 + 10Q + Q^2$, $P = 50$)	$Q = 20$, lucro 300	
Multiplicador (PMgC 0,8)	$k = 5$	

5 Etapa 2 · As curvas no gráfico

O gráfico de oferta e demanda é o mais famoso da Economia. Atenção: por tradição, o **preço** fica no eixo vertical.

PROMPT 2

Crie graficos.py com:

1. grafico_mercado: as curvas de oferta e demanda (preço no eixo vertical, quantidade no horizontal), o ponto de equilíbrio marcado e, se houver preço tabelado, uma linha horizontal e a escassez destacada;
2. grafico_custos: CMe e CMg em função de Q, com a linha do preço e o ponto de lucro máximo;
3. grafico_multiplicador: barras da renda gerada em cada rodada e a linha do total acumulado se aproximando de ΔY .



CHECKPOINT

No gráfico de mercado, o equilíbrio está em $P = 20$ e $Q = 60$, com o preço no eixo vertical.



CONFIRA A IA

É muito comum a IA colocar o preço no eixo horizontal (o jeito "matemático"). Na Economia, o preço vai no eixo vertical.

Etapa 3 · As abas de microeconomia

Agora o app. Cada modelo de micro ganha uma aba com controles e o gráfico que se move.

PROMPT 3

Crie app.py com Streamlit e abas (st.tabs):

1. "Oferta e demanda": sliders para a, b, c, d e um preço tabelado opcional; mostra o gráfico e uma frase explicando o resultado;
2. "Elasticidade": preços e quantidades antes e depois; mostra E, a classificação e a receita antes e depois, com um gráfico de barras da receita;
3. "Custos": custo fixo, coeficientes e preço; mostra o gráfico e o lucro máximo.

TERMINAL

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



CHECKPOINT

Mexendo no intercepto da demanda (a), a curva se desloca e o equilíbrio muda na hora.

Teste no app

Novo equilíbrio

Demanda aumenta (a de 100 para 120)

Oferta diminui (c de 20 para 0)

Etapa 4 • Macro: multiplicador e inflação real

Duas abas de macroeconomia: o multiplicador, com os parâmetros na tela, e a inflação de verdade, buscada no Banco Central.

PROMPT 4

Acrescente ao app:

1. aba "Multiplicador": PMgC e ΔG em sliders, o gráfico das rodadas e o ΔY final;
2. aba "Inflação real": busque no SGS do Banco Central o IPCA de 12 meses (série 13522) dos últimos 36 meses e faça um gráfico de linha com a faixa da meta (de 1,5% a 4,5%) sombreada; com plano B se não houver internet.



CHECKPOINT

A aba Inflação mostra 36 meses de IPCA, com a faixa da meta sombreada e a data do último dado.

8 Etapa 5 · O app que explica

Um bom laboratório não só mostra: **explica**. Cada aba vai escrever, em linguagem simples, o que mudou e por quê.

PROMPT 5

Em cada aba, acrescente uma caixa de explicação (st.info) que mude com os controles. Exemplos: "Com o preço tabelado em R\$ 15, faltam 20 unidades: os consumidores querem 70, mas os produtores só oferecem 50." ou "A demanda é elástica: o aumento de preço reduziu a receita de R\$ 1.000 para R\$ 960." As frases devem usar os números calculados, não textos fixos.



CHECKPOINT

Mudando os controles, as frases mudam junto, com os números certos.

Etapa 6 · O relatório em Excel

O professor pediu um relatório dos seus experimentos. O app gera um Excel com o que está na tela.

PROMPT 6

Acrescente um botão "Baixar relatório" que gere um Excel com uma aba por modelo: os parâmetros usados, a tabela de valores (por exemplo, a tabela de custos de $Q = 0$ a 40) e a frase de explicação. Use pandas com ExcelWriter.



CHECKPOINT

O Excel tem uma aba por modelo, com os parâmetros que estavam na tela.

10 Desafio extra · Imposto e peso morto

O governo cria um imposto por unidade. Quem paga: o consumidor ou o produtor? E quanto a sociedade perde?

PROMPT 7

desafio

Acrescente à aba Oferta e demanda um imposto por unidade (slider).
Mostre o novo equilíbrio, o preço pago pelo consumidor e o recebido pelo produtor, a arrecadação e o triângulo do peso morto sombreado no gráfico.
Explique incidência tributária e peso morto 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. Não faça alterações. Confira as fórmulas dos modelos, o sinal da elasticidade, as unidades dos eixos e as hipóteses 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 o tabelamento criando escassez e o multiplicador em ação.
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 modelos econômicos interativos") 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 construindo o app que não tinha aprendido só com a teoria?

13 Como você será avaliado

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

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</code> · <code>st.header</code>	Títulos
<code>st.number_input</code> · <code>st.slider</code>	Entradas de número
<code>st.selectbox</code> · <code>st.file_uploader</code>	Escolha e upload
<code>st.metric</code> · <code>st.dataframe</code>	Indicador e tabela
<code>st.pyplot(fig)</code>	Mostrar um gráfico
<code>st.tabs</code> · <code>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.

Parâmetro

Número que define a curva (inclinação, intercepto).

Estática comparativa

Comparar o equilíbrio antes e depois de uma mudança.

Série temporal

Dados ao longo do tempo, como o IPCA mês a mês.

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.