

// Laboratório codex 02 · economia · micro e macro

python + ia

>_ Laboratório Codex 02

Economia em Python, com IA: micro e macro

Oferta e demanda, elasticidade, custos, PIB, inflação, multiplicador e juros: os modelos da Economia viram programas, e no fim você usa dados reais do Banco Central.

// apostila do aluno II · trilha de exercícios

01 Oferta, demanda e equilíbrio

02 Elasticidade-preço

03 Custos e lucro máximo

04 PIB nominal e real

05 Inflação acumulada

06 Multiplicador keynesiano

07 Regra de Taylor

08 Painel do Banco Central



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 **continuação da Apostila I** (Laboratório Codex 01). Lá você instalou as ferramentas e aprendeu a conversar com o Codex. Aqui você vai **praticar**: cada exercício transforma um **modelo econômico** em programa. Os três primeiros são de **microeconomia**; os três seguintes, de **macroeconomia**. No fim, você usa dados reais do Banco Central.

1. Revisar

Checklist e lembretes da Apostila I

2. Micro

Mercado, elasticidade, custos e lucro

3. Macro

PIB, inflação e multiplicador

4. Política

Juros e a Regra de Taylor

5. Dados reais

Painel com dados do Banco Central

Os sinais desta apostila

Quando aparecer	O que significa
Cartão roxo · PROMPT	Texto para copiar e colar no Codex. Leia antes de enviar.
Quadrado tracejado · CHECKPOINT	O que você deve estar vendo. Se não estiver, pare e peça ajuda.
Caixa azul · DICA	Atalho, pergunta para pensar ou jeito mais fácil de fazer.
Caixa laranja · ATENÇÃO	Ponto em que muita gente erra (e a IA também).

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Achar o equilíbrio	Calculo preço e quantidade de equilíbrio e o efeito de um tabelamento.	☐
Medir a elasticidade	Calculo a elasticidade e digo o que acontece com a receita.	☐
Decidir a produção	Encontro a quantidade de lucro máximo a partir dos custos.	☐
Ler as contas nacionais	Calculo o PIB, o PIB real e o crescimento.	☐
Medir a inflação	Acumulo taxas mensais e calculo a perda de poder de compra.	☐
Usar dados reais	Busco indicadores no Banco Central e calculo o juro real.	☐

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.** E aqui vale mais uma: **todo modelo simplifica a realidade**; saiba dizer o que ele deixa de fora.

2 Antes de começar

A instalação está na **Apostila I** (Laboratório Codex 01, seções 3 a 7). Se alguma coisa abaixo não estiver pronta, volte lá antes de seguir.

O que precisa estar pronto	Onde está na Apostila I	Ok
VS Code instalado e abrindo	Seção 4	☐
Python funcionando no terminal (<code>python --version</code> ou <code>py --version</code>)	Seção 5	☐
Extensão do Codex instalada e com login feito	Seção 6	☐
Você sabe criar uma pasta de projeto e abrir no VS Code	Seção 7	☐

Crie a pasta desta apostila

- 1 Na pasta Documentos, crie uma pasta chamada `Laboratorio_Codex_ECO`.

- 2 No VS Code, use **Arquivo** → **Abrir Pasta** e escolha essa pasta.
- 3 Abra o painel do Codex e confira que ele está conectado à sua conta.



CHECKPOINT

O Explorer mostra a pasta `Laboratorio_Codex_EC0` e o Codex responde a um "olá".



NÃO TEM A APOSTILA I?

Peça ao professor o PDF do **Laboratório Codex 01 · Apostila do aluno**. As seções 3 a 7 levam cerca de 30 minutos.

3

Lembretes da Apostila I



O CICLO DE TRABALHO COM A IA

Pedir um **plano** → ler → aprovar → **executar** → **testar** com contas feitas por você → **explicar** o código com suas palavras.



MÉTODO DE DEPURAÇÃO

Diagnóstico → explicação → solução → alteração → teste. **Peça o diagnóstico antes da correção.**



A VÍRGULA BRASILEIRA

No Python, o decimal é com **ponto** (`13.75`). Em todos os programas desta apostila, peça ao Codex para **aceitar vírgula ou ponto**. E digite os milhares **sem ponto** (`1000` , e não `1.000`): um programa que aceita vírgula ou ponto pode ler 1.000 como 1.



SEGURANÇA

Só aprove o que você entende. Trabalhe só na pasta do projeto. Nunca coloque senhas ou chaves no código.

PROMPT UNIVERSAL DE DIAGNÓSTICO

serve para qualquer erro

Estou com este problema no meu projeto.

Erro/mensagem: [COLE A MENSAGEM]

Arquivo envolvido: [NOME DO ARQUIVO]

Antes de alterar qualquer coisa:

1. explique a causa provável;
2. diga como confirmar a causa;
3. proponha a menor correção;
4. aguarde minha autorização para alterar o arquivo.

4

Exercício 1 · Oferta, demanda e equilíbrio

O preço de mercado é onde a quantidade que os consumidores querem comprar **encontra** a que os produtores querem vender. Fora desse ponto, sobra ou falta produto.



demanda: $Q_d = a - b \cdot P$ **oferta:** $Q_s = c + d \cdot P$

equilíbrio: $Q_d = Q_s \rightarrow P^* = (a - c) \div (b + d)$

Exemplo do mercado de um produto: $Q_d = 100 - 2P$ e $Q_s = 20 + 2P$

PROMPT 1

Crie um programa em Python chamado mercado.py.

O usuário informa os coeficientes das curvas lineares de demanda ($Q_d = a - b \cdot P$) e de oferta ($Q_s = c + d \cdot P$). O programa deve:

1. calcular o preço e a quantidade de equilíbrio;
2. simular um preço máximo (tabelamento) ou mínimo informado pelo usuário e mostrar a quantidade demandada, a ofertada e se há escassez ou excedente;
3. mostrar uma pequena tabela de preços (de 0 até o preço em que $Q_d = 0$) com Q_d e Q_s , marcando o equilíbrio.

Antes de criar o arquivo, apresente um plano curto.

Situação ($Q_d = 100 - 2P$ · $Q_s = 20 + 2P$)	Qd	Qs	Resultado
Equilíbrio			
Preço tabelado em R\$ 15			
Preço mínimo de R\$ 25			



PARA PENSAR

Um governo tabela o preço abaixo do equilíbrio para "ajudar o consumidor". O que o seu programa mostra que acontece? Quem fica sem o produto?

5

Exercício 2 · Elasticidade-preço da demanda

Se a empresa aumentar o preço, a receita sobe ou cai? Depende de quanto os consumidores reagem: a **elasticidade-preço**.



$$E = (\text{variação \% da quantidade}) \div (\text{variação \% do preço})$$

Pelo método do ponto médio: $\text{variação \%} = \text{diferença} \div \text{média dos dois valores} \cdot |E| > 1$
 elástica · $|E| < 1$ inelástica

PROMPT 2

Crie um programa em Python chamado elasticidade.py.

O usuário informa o preço e a quantidade antes e depois de uma mudança de preço.

O programa deve:

1. calcular a elasticidade-preço da demanda pelo método do ponto médio;
2. classificar em elástica, inelástica ou unitária;
3. calcular a receita total antes e depois (preço × quantidade);
4. explicar em uma frase o que aconteceu com a receita e por quê.

Antes de escrever o código, apresente um plano curto.

Mudança	Elasticidade	Classificação	Receita antes → depois
Preço 10 → 12 · quantidade 100 → 80			
Preço 10 → 12 · quantidade 100 → 95			



PARA PENSAR

Que tipo de produto costuma ter demanda inelástica? Dê dois exemplos e diga por que as empresas desses setores conseguem repassar aumentos de preço.

6

Exercício 3 · Custos e lucro máximo

Quanto uma empresa deve produzir? Em concorrência perfeita, ela produz até o ponto em que o **custo de produzir mais uma unidade** (custo marginal) iguala o preço.



$$CT(Q) = 100 + 10Q + Q^2 \quad \cdot \quad CMe = CT \div Q \quad \cdot \quad CMg = CT(Q) - CT(Q-1)$$

$$lucro = P \cdot Q - CT(Q)$$

Exemplo: preço de mercado $P = R\$ 50$

PROMPT 3

Crie um programa em Python chamado custos.py.

O usuário informa o custo fixo, os coeficientes do custo variável ($CV = c1 \cdot Q + c2 \cdot Q^2$) e o preço de mercado. O programa deve:

1. montar uma tabela de $Q = 0$ até 40 com custo total, custo médio, custo marginal, receita e lucro;
2. encontrar a quantidade que dá o maior lucro e mostrar esse lucro;
3. mostrar o preço mínimo para a empresa não ter prejuízo (o menor custo médio da tabela).

Antes de escrever o código, apresente um plano curto.

Quantidade	Custo total	Custo médio	Lucro
10			
20			
30			

Quantidade de lucro máximo: _____ · Lucro máximo: R\$ _____ · Nesse ponto, o custo marginal é igual a quê?

7

Exercício 4 · PIB nominal, real e crescimento

Agora a **macroeconomia**. O PIB mede tudo o que o país produziu. Mas, se os preços sobem, o PIB nominal cresce mesmo sem produção a mais. Por isso existe o **PIB real**.



$$\text{PIB} = C + I + G + (X - M)$$

$$\text{PIB real} = \text{PIB nominal} \div (\text{deflador} \div 100)$$

C consumo · I investimento · G governo · X exportações · M importações

PROMPT 4

Crie um programa em Python chamado `pib.py`.

O programa deve:

1. pedir consumo, investimento, gastos do governo, exportações e importações (em bilhões) e calcular o PIB pela ótica da despesa;
2. mostrar a participação de cada componente no PIB (%);
3. pedir o deflador do PIB (base 100) e calcular o PIB real;
4. pedir o PIB real do ano anterior e calcular o crescimento real (%).

Antes de escrever o código, apresente um plano curto.

Dados (bilhões): C 600 · I 200 · G 250 · X 120 · M 100 · deflador 107 · PIB real do ano anterior 970.

Resultado	Você espera	O programa mostrou
PIB nominal		
PIB real		
Crescimento real		



PARA PENSAR

O PIB nominal cresceu muito mais que o real. O que explica a diferença? O que o deflador está medindo?

8

Exercício 5 · Inflação acumulada e poder de compra

A inflação é divulgada **mês a mês** (IPCA, pelo IBGE). Para saber quanto os preços subiram no período, não se soma: **multiplica-se**, como nos juros compostos.



$$\text{acumulada} = (1 + i_1) \times (1 + i_2) \times \dots \times (1 + i_n) - 1$$

$$\text{poder de compra} = \text{valor} \div (1 + \text{inflação acumulada})$$

Taxas em decimal: 0,5% = 0,005

PROMPT 5

Crie um programa em Python chamado inflacao.py.

O usuário informa as taxas mensais de inflação (%), uma por linha, até digitar uma linha vazia. O programa deve:

1. calcular a inflação acumulada no período (por multiplicação);
2. mostrar a soma simples das taxas, para comparar;
3. anualizar a inflação do período (como se o ritmo continuasse por 12 meses);
4. calcular quanto vale hoje, em poder de compra, um salário informado que não foi reajustado.

Antes de escrever o código, apresente um plano curto.

Taxas mensais: 0,5 · 0,4 · 0,3 · 0,6 · 0,2 · 0,4 (%). Salário: R\$ 1.000.

Resultado	Você espera	O programa mostrou
Inflação acumulada (6 meses)		
Soma simples		
Anualizada		
Poder de compra de R\$ 1.000		

9

Exercício 6 · O multiplicador keynesiano

Quando o governo gasta R\$ 1, a renda do país pode crescer **mais** do que R\$ 1: quem recebe gasta parte, que vira renda de outro, que gasta parte de novo. É o **multiplicador**.



$$k = 1 \div (1 - PMgC) \quad \cdot \quad \Delta Y = k \times \Delta G$$

PMgC: propensão marginal a consumir (quanto de cada R\$ 1 a mais de renda vira consumo)

PROMPT 6

Crie um programa em Python chamado multiplicador.py.

O programa deve:

1. pedir a propensão marginal a consumir (entre 0 e 1) e o aumento dos gastos do governo;
 2. calcular o multiplicador e o aumento da renda;
 3. mostrar, rodada por rodada (até 10 rodadas), como o gasto inicial vira renda e consumo, e o total acumulado se aproximando de ΔY ;
 4. recusar PMgC igual a 1 ou fora do intervalo, explicando por quê.
- Antes de escrever o código, apresente um plano curto.

PMgC	Aumento do gasto	Multiplicador	Aumento da renda
0,80	R\$ 10 bi		
0,75	R\$ 10 bi		



CONFIRA A IA

O modelo simples ignora impostos, importações e juros. Peça ao Codex para explicar por que, na vida real, o multiplicador costuma ser bem menor.

10 Desafio extra · O Copom deveria subir os juros?

A **Regra de Taylor** é uma fórmula usada para avaliar se os juros básicos estão altos ou baixos, olhando a inflação e o ritmo da economia.

★
$$\text{juros} = \text{juro real neutro} + \text{inflação} + 0,5 \times (\text{inflação} - \text{meta}) + 0,5 \times \text{hiato do produto}$$

Hiato positivo: a economia está acima do seu potencial (aquecida)

PROMPT 7

desafio

Crie um programa em Python chamado `taylor.py`.

O usuário informa o juro real neutro, a inflação esperada, a meta de inflação, o hiato do produto (%) e a Selic atual. O programa deve calcular a taxa indicada pela Regra de Taylor, compará-la com a Selic e dizer se a política monetária está mais apertada ou mais frouxa que a regra. Explique a regra para um estudante de Economia antes de escrever o código.

Juro neutro	Inflação	Meta	Hiato	Taxa pela regra
5%	4,22%	3%	0,5%	

Com a Selic em 13,75%, a política está mais apertada ou mais frouxa que a regra? _____

11 Use o Codex como professor

Escolha um dos seus programas e use os três prompts abaixo. Anote no diário de prompts o que aprendeu.

EXPLICAR

Explique este código linha por linha para um estudante de Economia que conhece o conteúdo, mas nunca programou em Python. Não reescreva o código.

REVISAR O CONTEÚDO

Revise meu programa como um professor de Economia. Não faça alterações. Confira se os conceitos e as contas estão corretos (fórmulas, unidades, sinais, hipóteses do modelo) e liste os riscos de entrada inválida.

TESTAR

Crie uma tabela de casos de teste para este programa, com o resultado esperado de cada caso: um caso normal, um caso-limite, valores com centavos ou decimais e uma entrada inválida.



CONFIRA A TABELA DE TESTES

Os "resultados esperados" que a IA escreve também podem estar errados. Confira pelo menos dois na calculadora.

Chega de dados inventados. Seu programa vai buscar os números **direto do Banco Central**, pela internet, e montar um painel da economia com a Selic, a inflação e o dólar.



DE ONDE VÊM OS DADOS

O Banco Central publica as séries do **SGS** (Sistema Gerenciador de Séries Temporais) num endereço aberto: <https://api.bcb.gov.br/dados/serie/bcdata.sgs.{codigo}/dados/ultimos/{N}?formato=json>. Códigos: **432** meta da Selic · **433** IPCA mensal · **13522** IPCA acumulado em 12 meses · **1** dólar (venda).

Requisitos

- ☐ Buscar Selic, IPCA de 12 meses, IPCA mensal e dólar no SGS
- ☐ Usar só a biblioteca padrão do Python (urllib e json)
- ☐ Calcular o juro real (Selic e IPCA de 12 meses, pela fórmula de Fisher)
- ☐ Mostrar os últimos 12 IPCAs mensais e a inflação acumulada
- ☐ Avisar com uma mensagem clara se estiver sem internet
- ☐ Mostrar a data de cada dado e a fonte



JURO REAL (FISHER)

$(1 + \text{Selic}) \div (1 + \text{IPCA 12 meses}) - 1$.

Compare com o resultado do Desafio: os juros do Brasil estão altos ou baixos?

PROMPT A

planejamento

Quero criar em Python um painel da economia brasileira com dados do Banco Central.

Fonte: API do SGS do Banco Central, endereço

<https://api.bcb.gov.br/dados/serie/bcdata.sgs.{codigo}/dados/ultimos/{N}?formato=json>

Séries: 432 (meta Selic), 433 (IPCA mensal), 13522 (IPCA 12 meses), 1 (dólar venda).

Antes de escrever código:

1. explique o que é uma API e o formato JSON que ela devolve;
2. proponha uma função para buscar uma série e uma para cada indicador;
3. diga como tratar erro de internet e dado faltando.

Use só a biblioteca padrão (urllib e json). Não escreva o código ainda.

PROMPT B

implementação

Agora implemente o painel em um arquivo chamado `painel.py`.

Mostre: Selic, IPCA 12 meses, juro real, dólar e os últimos 12 IPCAs mensais com a inflação acumulada (reaproveite o Exercício 5). Indique a data e a fonte.

Indicador	Valor	Data
Meta da Selic		
IPCA 12 meses		
Juro real		
Dólar (venda)		



CONFIRA COM A FONTE

Abra o site do Banco Central ou do IBGE e confira pelo menos dois números do seu painel. A IA pode errar o código da série ou a ordem das datas.

13 Entrega e reflexão

Diário de prompts

#	O que pedi (resumo)	O que a IA fez	Precisei corrigir?
1			
2			
3			

O que entregar

Item	Entregue
Pasta do projeto com os arquivos .py dos exercícios e da atividade final	☐
As tabelas de teste desta apostila preenchidas	☐
Um erro de conteúdo que a IA cometeu (ou que você encontrou) e como foi corrigido	☐
O diário de prompts (acima) e a reflexão final (abaixo)	☐
Registro da auditoria humana (opcional, vale ponto extra)	☐

Reflexão individual

1. Qual modelo econômico ficou mais claro depois que você o programou?

2. Em que momento a IA errou um conceito de Economia, e como você percebeu?

3. O que o painel com dados reais mostrou sobre a economia brasileira hoje?

4. Escolha um modelo da apostila e diga o que ele deixa de fora da realidade.

14 Como você será avaliado

Critério	0	1	2	3
Conteúdo da disciplina	Conceitos errados	Acertou alguns	Acertou quase todos	Todos corretos e conferidos
Uso do Codex	Não conseguiu	Só copiou	Usou com alguma compreensão	Usou de forma crítica
Programação	Não funciona	Funciona em parte	Funciona	Funciona e está bem organizado
Testes	Não testou	Um teste	Vários testes	Testes e casos-limite
Depuração	Não identificou	Precisou da solução pronta	Identificou com ajuda	Diagnosticou e corrigiu
Explicação	Não explica	Explica pouco	Explica a lógica	Explica a lógica e o significado econômico

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

As contas desta apostila

Equilíbrio	$P^* = (a-c)/(b+d)$
Elasticidade	$\Delta\%Q / \Delta\%P$ (ponto médio)
Custo marginal	$CT(Q) - CT(Q-1)$
PIB	$C + I + G + (X - M)$
PIB real	$\text{nominal} / (\text{deflator}/100)$
Inflação acumulada	$\text{produto}(1+i) - 1$
Multiplicador	$1 / (1 - PMgC)$
Taylor	$r^* + \pi + 0,5(\pi - \text{meta}) + 0,5 \cdot \text{hiato}$

Comandos

TERMINAL

```
python arquivo.py    # executa um
programa
Ctrl + C             # interrompe um
programa travado
```

Glossário

Equilíbrio de mercado

Preço em que a quantidade demandada iguala a ofertada.

Elasticidade-preço

Sensibilidade da quantidade demandada a uma mudança de preço.

Custo marginal

Custo de produzir uma unidade a mais.

PIB

Valor de tudo o que o país produz num período.

Deflator

Índice de preços usado para tirar a inflação do PIB.

IPCA

Índice oficial de inflação do Brasil (IBGE).

Multiplicador

Quanto a renda cresce para cada R\$ 1 de gasto autônomo.

Selic e Copom

A taxa básica de juros e o comitê do Banco Central que a define.

API

Endereço na internet de onde um programa busca dados.

1

LEMBRE-SE

A IA é uma ferramenta. Conhecimento, julgamento, validação e responsabilidade continuam sendo de quem programa, e de quem usa os modelos para entender a economia.

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.