

// laboratório codex 03 · panorama econômico

python + ia + app

>_ Laboratório Codex 03

Painel da economia em Python, com IA

Anos de dados do Banco Central num app: juros, inflação, câmbio, emprego, dívida e comércio exterior em gráficos, com filtros e um boletim automático em Excel.

// apostila do aluno III · o projeto em etapas

01 Séries históricas

02 Indicadores calculados

03 Gráficos das séries

04 Painel com filtros

05 Boletim automático

06 Boletim em Excel

07 Expectativas Focus

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 **painel da economia brasileira**, com séries históricas do Banco Central, gráficos, filtros de período e um boletim em Excel.

1. Dados

Séries históricas
do BCB

2. Cálculos

Juro real e
indicadores

3. Gráficos

Séries no tempo

4. App

Painel com filtros

5. Boletim

Texto automático
e Excel

★ O PROJETO: PAINEL DA ECONOMIA BRASILEIRA

O app busca no Banco Central vários anos de Selic, IPCA, dólar, desemprego, dívida pública e balança comercial; guarda os dados para funcionar sem internet; mostra gráficos com filtros de período e escreve um boletim automático, que pode ser baixado em Excel.

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Coletar séries	Baixo anos de dados do Banco Central e guardo em CSV.	☐
Tratar dados	Converto datas e valores e junto séries diferentes.	☐
Fazer gráficos de séries	Mostro a evolução com a meta e os destaques.	☐
Construir um painel	O usuário escolhe o período e os indicadores.	☐
Escrever um boletim	O app escreve o resumo do mês com os números.	☐
Apresentar	Explico a conjuntura usando o meu painel.	☐

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_PE3` em Documentos e abra no VS Code. Ao longo das etapas, ela vai ficar assim:

TERMINAL

```
Laboratorio_Codex_PE3/  
├─ 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_PE3_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 · As séries históricas

Desta vez, não só o último número: o app vai buscar **anos de dados** e guardá-los numa pasta, para funcionar mesmo sem internet.



AS SÉRIES

<https://api.bcb.gov.br/dados/serie/bcdata.sgs.{código}/dados?formato=json&dataInicial=01/01/2021&dataFinal=31/12/2026> · 432 Selic · 13522 IPCA 12 meses · 433 IPCA mensal · 1 dólar (diário) · 24369 desocupação · 13762 dívida bruta (% PIB) · 22707 balança comercial (US\$ milhões).

PROMPT 1

Crie `calculos.py` com uma função `baixar_serie(codigo, data_inicial, data_final)` que busque a série no SGS e devolva um `DataFrame` com as colunas `data` (tipo `data`) e `valor` (número). Depois, crie `atualizar_dados()` que baixe as 7 séries de 2021 até hoje e salve cada uma em `dados/serie_CODIGO.csv`. Se não houver internet, use os CSVs já salvos e avise a data da última atualização. Atenção: a série 1 (dólar) é diária; transforme-a em média mensal.



CHECKPOINT

A pasta `dados/` tem 7 arquivos CSV. Sem internet, o programa usa os arquivos e avisa.

Série	Primeira data	Última data	Último valor
Selic (432)			
IPCA 12 meses (13522)			
Dólar (1, média mensal)			

5

Etapa 2 · Os indicadores calculados

Com as séries em tabelas, dá para calcular indicadores que o Banco Central não entrega prontos.

PROMPT 2

Em `calculos.py`, crie funções que juntem as séries por mês e calculem:

1. o juro real mensal: $(1 + \text{Selic}) \div (1 + \text{IPCA 12 meses}) - 1$;
2. a balança comercial acumulada em 12 meses (soma móvel), em US\$ bilhões;
3. a variação do dólar em 12 meses (%);
4. um quadro com o último valor, o valor de 12 meses antes e a variação de cada indicador.

Indicador	Último valor	12 meses antes	Variação
Selic			
IPCA 12 meses			
Juro real			
Dólar			
Desocupação			
Dívida bruta			

Etapa 3 · Os gráficos das séries

Gráfico de série temporal tem regras: datas legíveis no eixo, unidade clara e destaque para o que importa.

PROMPT 3

Crie graficos.py com:

1. grafico_juros_inflacao: Selic, IPCA 12 meses e juro real no mesmo gráfico, com a faixa da meta de inflação (1,5% a 4,5%) sombreada;
2. grafico_cambio: dólar médio mensal, marcando a máxima e a mínima do período;
3. grafico_trabalho_divida: desocupação e dívida bruta em dois gráficos lado a lado;
4. grafico_balanca: barras da balança mensal e a linha da soma de 12 meses.



CHECKPOINT

Os gráficos têm datas legíveis no eixo horizontal e a unidade no eixo vertical.

7 Etapa 4 · O painel com filtros

O usuário escolhe o período e o que quer ver. O painel se ajusta.

PROMPT 4

Crie app.py com Streamlit:

- barra lateral: período (st.slider com datas ou dois st.date_input), botão "Atualizar dados do Banco Central" e a data da última atualização;
- no topo: st.metric com o último valor de cada indicador e a variação em 12 meses (setinha para cima ou para baixo);
- abaixo, abas: "Juros e inflação", "Câmbio", "Trabalho e dívida", "Comércio exterior".

TERMINAL

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



CHECKPOINT

Mudando o período para só 2025 e 2026, todos os gráficos se ajustam.

Etapa 5 · O boletim automático

Analistas escrevem todo mês o mesmo tipo de texto. O app pode fazer o rascunho, com os números certos.

PROMPT 5

Crie a função `escrever_boletim(quadro)` que gere um texto de 4 a 6 frases, usando os números do quadro da etapa 2. Exemplo: "Em agosto de 2026, a Selic estava em 13,75% ao ano e o IPCA de 12 meses em 4,22%, dentro da meta. O juro real ficou em 9,14%..." Use regras simples (se/senão) para escolher as palavras (subiu, caiu, dentro, acima). Mostre o boletim numa aba "Boletim".



CHECKPOINT

O boletim muda quando os dados mudam e cita a data e a fonte.



O TEXTO É UM RASCUNHO

O programa escreve, mas quem assina é você. Leia o boletim e confira se as palavras (subiu, caiu, dentro da meta) estão certas para os números.

Etapa 6 • O boletim em Excel

O boletim vai para a diretoria ou para os clientes. O app gera um Excel com o texto, o quadro e as séries.

PROMPT 6

Acrescente um botão "Baixar boletim em Excel" com as abas: "Boletim" (o texto e a data), "Quadro" (últimos valores e variações) e uma aba para cada série, filtrada pelo período escolhido.



CHECKPOINT

O Excel abre com o boletim na primeira aba e as séries nas seguintes.

10 Desafio extra · O que o mercado espera

Acrescente as expectativas do **Boletim Focus** ao painel, pela API Olinda do Banco Central.

PROMPT 7

desafio

Pesquise e use a API Olinda de expectativas do Banco Central (ExpectativasMercadoAnuais) para buscar a mediana do IPCA e da Selic esperados para este ano e o próximo. Mostre numa aba "Expectativas" e calcule o juro real ex-ante (Apostila II). Explique o endereço da API e os filtros 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 Panorama Econômico. Não faça alterações. Confira os códigos das séries, as unidades (% , R\$, US\$ milhões), a conversão de datas e o cálculo do juro real 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 como juros, inflação e câmbio se moveram juntos nos últimos anos.
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 painel da economia brasileira") 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 Panorama Econômico 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 · 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.

Série histórica

Valores de um indicador ao longo do tempo.

Cache

Cópia local dos dados para não depender da internet.

Período

Intervalo de datas escolhido para análise.

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.