

// laboratório codex 03 · gerenciamento orçamentário

python + ia + app

>_ Laboratório Codex 03

O app do orçamento em Python, com IA

Mude o preço ou as vendas e veja na hora a DRE e o caixa mudarem: você transforma o orçamento da Mesas Ipê num app com gráficos, cenários e Excel.

// apostila do aluno III · o projeto em etapas

01 Premissas em planilha

03 Gráficos de gestão

05 Cenários e tabelas

07 Orçado × realizado

02 Orçamentos em cadeia

04 Premissas na tela

06 Orçamento 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: o app de orçamento da Mesas Ipê**, em que o usuário muda as premissas e vê na hora a DRE, o caixa e os gráficos mudarem.

1. Dados

Premissas numa planilha

2. Cálculos

Os orçamentos em cadeia

3. Gráficos

Receita, custos e caixa

4. App

Premissas na tela

5. Relatório

Cenários e Excel

★ O PROJETO: APP DO ORÇAMENTO DA MESAS IPÊ

As premissas (vendas, preço, consumo, custos, estoques e prazos) ficam numa planilha. O app monta todos os orçamentos da Apostila II, mostra os gráficos de receita, custos, lucro e caixa, compara cenários e entrega o orçamento completo em Excel, uma aba por orçamento.

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Organizar premissas	Guardo as premissas numa planilha e o app as lê.	☐
Encadear orçamentos	Vendas → produção → insumos → DRE → caixa, em funções.	☐
Fazer gráficos de gestão	Mostro receita, custos, lucro e caixa trimestre a trimestre.	☐
Construir um app	Mudo uma premissa e tudo se recalcula.	☐
Comparar cenários	Mostro base, otimista e pessimista lado a lado.	☐
Gerar o orçamento	Entrego um Excel com uma aba por orçamento.	☐

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_G03` em Documentos e abra no VS Code. Ao longo das etapas, ela vai ficar assim:

TERMINAL

```
Laboratorio_Codex_G03/  
├─ 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_G03_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 premissas numa planilha

Num orçamento de verdade, ninguém mexe no código para mudar o preço: as **premissas** ficam numa planilha que o gestor edita.

PROMPT 1

```
Crie o arquivo dados/premissas.csv com as premissas da Mesas Ipê  
(uma linha por premissa: nome, valor, unidade, descrição):  
vendas de T1 a T4 (1.000, 1.200, 1.500, 1.300) e do T1 seguinte (1.100),  
preço (400), estoque inicial de mesas (200), estoque final (20% das vendas  
do trimestre seguinte), madeira por mesa (3 m²), preço do m² (50), estoque  
inicial de madeira (300), estoque final de madeira (10% do consumo seguinte;  
350 no T4), horas por mesa (2), custo da hora (25), encargos (80%), CIF por  
trimestre (60.000, com 10.000 de depreciação), comissão (5%), despesas fixas  
(50.000), prazos de recebimento (60/40) e de pagamento (50/50), saldos e a  
máquina de 120.000 no T2.  
Depois, em calculos.py, crie ler_premissas() que leia o CSV com pandas.
```



CHECKPOINT

A função leu o CSV e mostrou as premissas no terminal, com vírgula ou ponto funcionando.

5

Etapa 2 · Os orçamentos em cadeia

Cada orçamento usa o resultado do anterior. Em funções, a cadeia fica clara e qualquer mudança se propaga.

PROMPT 2

Em `calculos.py`, crie uma função para cada orçamento, cada uma devolvendo um `DataFrame` por trimestre (T1 a T4 e total):
orcamento_vendas → orcamento_producao → orcamento_materia_prima →
orcamento_mao_de_obra → custo_unitario_e_dre → orcamento_caixa.
Crie também `orcamento_completo(premissas)`, que chama todas na ordem e devolve um dicionário com as tabelas. Reaproveite a lógica da Apostila II.

Conferência (cenário base)	Apostila II	O app calculou
Produção do ano	5.020 mesas	
Custo por mesa	R\$ 287,81	
Lucro operacional	R\$ 260.956,18	
Empréstimo no T2	R\$ 68.900	

! BATEU?

Se algum número não bater com a Apostila II, não siga para os gráficos: descubra antes qual função está errada.

Etapa 3 · Os gráficos de gestão

O dono da Mesas Ipê não lê tabelas: ele quer ver. Três gráficos contam a história do ano.

PROMPT 3

Crie graficos.py com:

1. grafico_receita_custos: barras por trimestre com receita, custos de produção e despesas, lado a lado;
2. grafico_caixa: linha do saldo de caixa por trimestre, com uma linha horizontal tracejada no saldo mínimo e destaque no trimestre com empréstimo;
3. grafico_dre: barras horizontais da DRE (receita, CPV, lucro bruto, despesas, lucro), como uma "cascata" simples.

Valores em reais no formato brasileiro. Mostre como testar salvando em png.



CHECKPOINT

Os três gráficos foram salvos. No gráfico de caixa, dá para ver o problema do T2.

Etapa 4 • O app com as premissas na tela

Agora o gestor muda as premissas na tela, sem tocar no código, e vê o efeito na hora.

PROMPT 4

Crie app.py com Streamlit:

- barra lateral com as premissas principais (preço, vendas de cada trimestre, custo da hora, preço do m², saldo mínimo e o valor da máquina), começando pelos valores do CSV;
- no centro: indicadores com st.metric (receita, lucro operacional, margem, maior empréstimo) e os três gráficos;
- avise se em algum trimestre o caixa precisar de empréstimo.

TERMINAL

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



CHECKPOINT

Mudando o preço de 400 para 380, o lucro e o caixa mudaram na hora.

Teste no app	Lucro operacional	Empréstimo no T2
Cenário base		
Preço R\$ 380		
Máquina adiada (R\$ 0 no T2)		

8 Etapa 5 · Cenários e tabelas

Orçamento bom vem com **cenários**: o que acontece se as vendas caírem 10%? E se subirem?

PROMPT 5

Melhore o app com abas (st.tabs):

1. "Painel": o que já existe;
2. "Orçamentos": uma tabela para cada orçamento (st.dataframe);
3. "Cenários": base, otimista (+10% de vendas) e pessimista (-10%), com um gráfico comparando o lucro operacional e uma frase explicando por que o lucro varia mais que as vendas (alavancagem operacional).

Cenário	Lucro operacional	Variação contra a base
Pessimista (-10%)		
Base		
Otimista (+10%)		

Etapa 6 • O orçamento em Excel

O orçamento aprovado vai para a diretoria em Excel, uma aba para cada peça.

PROMPT 6

Acrescente um botão "Baixar orçamento em Excel" que gere um .xlsx com as abas: Premissas, Vendas, Produção, Matéria-prima, Mão de obra, DRE, Caixa e Cenários. Use pandas com ExcelWriter (openpyxl), números com 2 casas e títulos claros.



CHECKPOINT

O Excel tem 8 abas e os números batem com o app.

10 Desafio extra · Orçado x realizado no app

Feche o ciclo de gestão: compare o orçamento com o que aconteceu.

PROMPT 7

desafio

Acrescente uma aba "Orçado x realizado": o usuário envia um CSV com vendas e preços reais por trimestre (st.file_uploader), e o app mostra as variações de quantidade e de preço (Apostila II), com um gráfico de barras das variações favoráveis e desfavoráveis.

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 Gerenciamento Orçamentário. Não faça alterações. Confira estoques inicial e final, prazos de recebimento e pagamento, depreciação fora do caixa e a sequência dos orçamentos 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 que acontece com o caixa do T2 quando as vendas caem.
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 orçamento empresarial") 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 Gerenciamento Orçamentário 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.

Premissa

Hipótese usada para orçar.

Cenário

Conjunto de premissas (base, otimista, pessimista).

Alavancagem operacional

O lucro varia mais que as vendas por causa dos custos fixos.

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.