

// laboratório codex 03 · logística 4.0

python + ia + app

>_ Laboratório Codex 03

Gestão da transportadora em Python, com IA

Monte o sistema de gestão de uma transportadora: custo e margem por viagem, ocupação da frota, calculadora de frete, frota própria x terceiros e relatório em Excel.

// apostila do aluno III · o projeto em etapas

01 Frota e viagens

02 Custo por viagem

03 Gráficos da operação

04 Calculadora de frete

05 Dimensionar a frota

06 Relatório do mês

07 Roteirização

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 gestão de transportes** para uma transportadora rodoviária: custo e margem de cada viagem, ocupação dos caminhões, calculadora de frete, dimensionamento da frota e relatório em Excel.

1. Dados

Frota e viagens do mês

2. Custos

Viagem, km e margem

3. Gráficos

Rotas e caminhões

4. App

Frete e diesel na tela

5. Relatório

Frota e Excel

★ O PROJETO: GESTÃO DA TRANSPORTADORA

Você vai abrir (no computador) a Transportadora Estrada Boa, uma transportadora rodoviária fictícia de Campinas. O app lê a frota e as viagens do mês, calcula o custo de cada viagem (diesel, pedágio, motorista e manutenção), a margem, a ocupação dos caminhões e o custo por km; tem uma calculadora de frete, compara frota própria e terceirizada e gera o relatório do mês em Excel.

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Organizar os dados da operação	Monto os arquivos da frota e das viagens.	☐
Custear uma viagem	Calculo diesel, pedágio, motorista e manutenção.	☐
Medir a frota	Calculo ocupação, custo por km e margem por rota.	☐
Formar o frete	Calculo o frete mínimo para a margem desejada.	☐
Dimensionar a frota	Comparo frota própria, terceirizada e mista.	☐
Relatar	Gero o relatório do mês em Excel.	☐

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_L063` em Documentos e abra no VS Code. Ao longo das etapas, ela vai ficar assim:

TERMINAL

```
Laboratorio_Codex_LOG3/  
├─ 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_LOG3_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 · A frota e as viagens

Toda gestão de transportes começa por dois cadastros: os caminhões e as viagens.

PROMPT 1

Crie a pasta dados com dois arquivos CSV da Transportadora Estrada Boa (fictícia, de Campinas):

- frota.csv: placa, tipo (toco, truck, carreta), capacidade_t, consumo_km_l, custo_fixo_mes
(5 caminhões, com consumos entre 2,2 e 4,0 km/l);
- viagens.csv: data, caminhao (placa), origem, destino, km, peso_t, pedagio, frete
(30 viagens no mês, saindo de Campinas para cidades de SP e MG, incluindo Campinas → Santos, 180 km, pedágio R\$ 120, frete R\$ 1.500, num caminhão de 3 km/l).

Em calculos.py, crie ler_dados() com pandas, que devolve os dois arquivos.



CHECKPOINT

A função lê 5 caminhões e 30 viagens e mostra as tabelas no terminal.

5

Etapa 2 · O custo e a margem de cada viagem

O coração do sistema: quanto custou cada viagem e quanto sobrou.

PROMPT 2

Em `calculos.py`, crie `custo_viagem(viagem, caminhao, premissas)`, com as premissas `diesel = R$ 6,00 por litro`, `motorista = R$ 0,80 por km` e `manutenção = R$ 0,50 por km`:
`diesel = km ÷ consumo × preço do litro`; `custo variável = diesel + pedágio + motorista + manutenção`;
`margem = frete - custo variável (em R$ e em %)`; `ocupação = peso ÷ capacidade`.
 Crie também `resumo_frota()`: custo por km, ocupação média e margem por caminhão.
 Teste com a viagem Campinas → Santos.

Campinas → Santos (180 km, 3 km/l, pedágio R\$ 120, frete R\$ 1.500)	Esperado	Calculado
Diesel	R\$ 360,00	
Custo variável	R\$ 714,00	
Margem	R\$ 786,00 (52,4%)	

Etapa 3 · Os gráficos da operação

Três gráficos que mostram onde a transportadora ganha e perde dinheiro.

PROMPT 3

Crie graficos.py com:

1. `grafico_margem_rotas`: barras horizontais com a margem (%) de cada destino, vermelho abaixo de 15%;
2. `grafico_ocupacao`: ocupação média de cada caminhão, com uma linha em 80% (meta);
3. `grafico_custo_km`: custo por km de cada caminhão, comparando com a média da frota.



CHECKPOINT

As rotas com margem abaixo de 15% aparecem em vermelho.

7

Etapa 4 · O app com a calculadora de frete

O dono muda o preço do diesel e vê na hora o efeito nas margens. E calcula o frete de uma viagem nova.

PROMPT 4

Crie app.py com Streamlit:

- barra lateral: preço do diesel, custo do motorista por km, manutenção por km e margem desejada (%);
- no topo: st.metric com faturamento do mês, custo variável, margem total e ocupação média;
- os três gráficos;
- uma "Calculadora de frete": escolha o caminhão, informe km e pedágio, e o app mostra o custo
e o frete mínimo para a margem desejada: $\text{frete} = \text{custo} \div (1 - \text{margem})$.

TERMINAL

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

Teste no app	Resultado
Diesel de R\$ 6,00 para R\$ 7,00: margem total	
Frete mínimo Campinas → Santos com margem de 30%	
Rotas que ficam abaixo de 15%	

Etapa 5 · Frota própria, terceirizada ou mista?

Os slides do Tema 4 mostram o risco de dimensionar a frota pelo pico. Agora o app faz a conta.

PROMPT 5

Acrescente a aba "Dimensionar frota": a partir da quantidade de viagens por dia (de viagens.csv, ou digitada), do número de viagens por caminhão por dia, do custo diário de um caminhão próprio e do preço de uma viagem terceirizada, calcule o custo total para frotas de 1 a 10 caminhões, mostre um gráfico do custo total pela frota e destaque a melhor. Use os números do Desafio da Apostila II para conferir.



CHECKPOINT

Com os números do Desafio da Apostila II, a melhor frota é de 5 caminhões (R\$ 37.200).

Etapa 6 · O relatório do mês em Excel

Fechamento do mês para o dono e para o contador.

PROMPT 6

Acrescente um botão "Baixar relatório do mês" com as abas:
"Resumo" (texto de 4 a 6 frases escrito pelo app: faturamento, margem, melhor e pior rota, ocupação, caminhão com maior custo por km e recomendação de frota), "Viagens" (com custo e margem),
"Frota" e "Premissas" (diesel, motorista, manutenção).



CHECKPOINT

O resumo cita os números do app e o Excel tem 4 abas.

10

Desafio extra · Roteirização: a entrega na última milha

Na última milha, a ordem das entregas muda o total rodado. Comece pela regra do **vizinho mais próximo**.

PROMPT 7

desafio

Acrescente uma aba "Roteiro": o usuário informa o depósito e 6 a 10 endereços de entrega com coordenadas (x, y em km), e o app monta a rota pela regra do vizinho mais próximo, mostra a ordem, a distância total e um gráfico do trajeto. Compare com a ordem digitada. Explique por que essa regra é boa, mas nem sempre ótima.

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 Logística. Não faça alterações. Confira o custo da viagem (unidades: km, litros, R\$), a margem, a ocupação e as premissas de diesel e manutenção 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 preço do diesel subindo e as rotas que deixam de dar lucro.
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 gestão de transportes e logística") 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 Logística 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

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.

Frete

Valor cobrado pelo transporte da carga.

Ocupação

Peso transportado ÷ capacidade do caminhão.

Custo por km

Custo total ÷ quilômetros rodados.

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.