



// laboratório codex 05 · logística 4.0

python + ia + app

>_ Laboratório Codex 05

A transportadora inteligente, com IA

Banco de dados, previsão de demanda, rotas otimizadas, mapa, alertas e relatório executivo: o app da transportadora que ajuda a decidir.

// apostila do aluno V · o projeto em etapas

01 Banco de dados



03 Rotas otimizadas



05 Alertas



07 Rotas com capacidade



02 Previsão de demanda



04 Mapa



06 Relatório executivo



08 Apresentação



Prof. Caio Haffner

Mestre em Desenvolvimento Econômico · Unicamp

1 Como usar esta apostila

Esta é a **Apostila V de Logística**, a etapa avançada da trilha: ela continua a **Apostila IV** (o app da transportadora no ar). Agora você vai dar **inteligência** ao app da Transportadora Estrada Boa: um banco de dados com o histórico, a previsão das viagens dos próximos meses, rotas otimizadas no mapa, alertas automáticos e um relatório executivo para o dono.

1. Guardar

Banco de dados

2. Prever

Demanda de viagens

3. Otimizar

Rotas e mapa

4. Vigiar

Alertas

5. Decidir

Relatório executivo



O PROJETO: A TRANSPORTADORA INTELIGENTE

Nas Apostilas III e IV, o app mostra o que aconteceu. Agora ele passa a ajudar a decidir o que fazer: prevê quantas viagens virão, monta a melhor rota de entregas, mostra tudo no mapa, avisa quando algo sai do padrão e escreve o relatório do mês.

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Usar um banco de dados	O histórico fica num banco SQLite e o app consulta com SQL.	☐
Prever a demanda	Comparo média móvel e tendência e meço o erro.	☐
Otimizar rotas	Comparo o vizinho mais próximo com a rota ótima.	☐
Mostrar no mapa	As rotas aparecem num mapa interativo.	☐
Criar alertas	O app avisa quando margem, ocupação ou custo saem do padrão.	☐
Relatar	O app gera o relatório executivo do mês.	☐

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.** Aqui vale mais uma: **previsão e otimização são apoio à decisão.** Quem decide continua sendo uma pessoa, que entende os números.

2

Antes de começar



LEIA TUDO, NA ORDEM

Leia esta apostila **desde a primeira palavra** e faça tudo o que ela manda, na ordem. Cada etapa usa o que a anterior construiu.

A Apostila V começa onde a IV termina: **o app da transportadora no GitHub, com testes, dados reais e publicado na internet**. Se ainda não está assim, termine a IV antes.

Instale as bibliotecas novas (uma vez só)

TERMINAL

```
python -m pip install ortools folium streamlit-folium fpdf2
```

Biblioteca	Para que serve
sqlite3	Banco de dados num arquivo (já vem com o Python).
ortools	Otimização: a melhor rota e a divisão das entregas entre caminhões.
folium e streamlit-folium	Mapas interativos dentro do app.
fpdf2	Gerar o relatório executivo em PDF.

Ao longo das etapas, a pasta do projeto vai ganhar:

TERMINAL

```
Laboratorio_Codex_L0G3/  
├─ dados/viagens.db   ← novo: banco de dados  
├─ calculos.py  
├─ previsao.py        ← novo  
├─ rotas.py           ← novo  
├─ alertas.py         ← novo  
├─ app.py  
└─ tests/
```



LEMBRETES DAS APOSTILAS ANTERIORES

Peça sempre um **plano** antes do código · teste com contas feitas por você · peça o **diagnóstico antes da correção** · nada vai para a internet sem passar nos testes · nunca coloque senhas no código.

3

Etapa 1 · Guardar: o banco de dados da operação

Planilhas soltas não aguentam anos de viagens. Um **banco de dados** guarda o histórico inteiro e responde perguntas em segundos.

PROMPT 1

Crie dados/viagens.db (SQLite, que já vem com o Python) com as tabelas frota e viagens, importando os CSV das Apostilas III e IV. Crie banco.py com funções que usem SQL para:

1. o faturamento, o custo e a margem por mês;
2. as 3 rotas com maior margem e as 3 com menor;
3. a ocupação média por caminhão.

Antes de criar, explique em 5 linhas o que são tabela, linha, coluna e consulta SQL.



CHECKPOINT

As três consultas mostram números iguais aos da Apostila III para o mesmo mês.



LEIA O SQL

Peça ao Codex: "explique esta consulta SQL linha por linha". SELECT escolhe as colunas, WHERE filtra as linhas, GROUP BY agrupa e ORDER BY ordena.

4

Etapa 2 · Prever: quantas viagens virão?

Para decidir a frota do próximo mês, a transportadora precisa de uma **previsão**. Duas formas simples e honestas: média móvel e tendência.



média móvel de 3 meses = média dos 3 últimos · **tendência:**
viagens = $a + b \times \text{mês}$ (regressão)

Erro absoluto médio (EAM) = média de $|\text{real} - \text{previsto}|$ nos meses em que dá para testar



VIAGENS POR MÊS (12 MESES)

120 · 130 · 125 · 140 · 150 · 145 · 160 · 170 · 165 · 180 · 190 · 185

PROMPT 2

Crie previsao.py com:

1. media_movel(serie, n=3) e a previsão do mês 13;
2. tendencia(serie): a e b da reta por mínimos quadrados e a previsão dos meses 13 e 14;
3. o erro absoluto médio da média móvel nos meses 4 a 12.

No app, mostre um gráfico com o histórico, as duas previsões e uma frase dizendo qual erra menos.

Resultado	Esperado	Calculado
Média móvel (mês 13)	185,0	
Tendência: a e b	$a \approx 113,18$ · $b \approx 6,43$	
Tendência (mês 13 e 14)	196,8 e 203,3	
Erro absoluto médio da média móvel	13,33	



PARA PENSAR

A média móvel ficou para trás de uma série que cresce. Por quê? Quando ela seria melhor que a tendência?

5

Etapa 3 · Otimizar: a melhor rota de entregas

Na Apostila III, o desafio usou o **vizinho mais próximo**. Agora você compara com a **rota ótima**, calculada pelo OR-Tools.



AS ENTREGAS DO DIA (COORDENADAS EM KM, DEPÓSITO EM 0,0)

A (2, 0) · B (9, 1) · C (3, 4) · D (8, 6) · E (1, 7) · F (-2, 3). O caminhão sai do depósito, passa por todos e volta.

PROMPT 3

Instale o OR-Tools (python -m pip install ortools) e crie rotas.py com:

1. vizinho_mais_proximo(pontos): a ordem e a distância total;
2. rota_otima(pontos): a melhor ordem com o OR-Tools (problema do caixeiro-viajante);
3. uma conferência por força bruta (todas as ordens possíveis) para até 7 pontos.

Mostre a economia em km e em % da rota ótima sobre o vizinho mais próximo.

Método	Ordem	Distância total
Vizinho mais próximo	A-C-E-F-D-B	39,32 km
Rota ótima	A-B-D-C-E-F (ou o inverso)	31,77 km

Economia da rota ótima: _____ km (_____ %).

Etapa 4 • Mostrar: as rotas no mapa

Um mapa vale mais que uma tabela de coordenadas. O dono vê as rotas, os clientes e onde a margem é baixa.

PROMPT 4

```
Instale folium e streamlit-folium (python -m pip install folium streamlit-folium).  
Acrescente ao app a aba "Mapa": as cidades das viagens (Campinas, Santos, São Paulo,  
Ribeirão Preto,  
Sorocaba, Belo Horizonte...) com latitude e longitude, uma linha de Campinas a cada  
destino com a cor  
da margem (verde acima de 30%, amarela de 15% a 30%, vermelha abaixo de 15%) e um  
balão com o  
destino, o número de viagens e a margem média.
```



CHECKPOINT

O mapa abre no app, dá zoom e as rotas de margem baixa aparecem em vermelho.

Etapa 5 · Vigiar: alertas automáticos

Ninguém olha todos os números todos os dias. Os **alertas** mostram só o que pede atenção.

PROMPT 5

Crie `alertas.py` com regras que o dono pode ajustar na barra lateral:

1. viagem com margem abaixo de 15%;
2. caminhão com ocupação média abaixo de 60%;
3. caminhão com custo por km 20% acima da média da frota;
4. previsão do próximo mês acima da capacidade da frota (viagens por caminhão × caminhões).

No topo do app, mostre os alertas em vermelho (crítico) e amarelo (atenção), do mais grave para o menos grave.

Alerta	Quantos apareceram	Faz sentido?
Margem abaixo de 15%		
Ocupação abaixo de 60%		
Custo por km 20% acima		
Previsão acima da capacidade		



ALERTA DEMAIS É ALERTA NENHUM

Se aparecerem 30 alertas, o dono para de olhar. Ajuste as regras para mostrar só o que exige decisão.

Etapa 6 · Decidir: o relatório executivo

O dono não quer tabelas: quer saber o que aconteceu, o que vem aí e o que fazer.

PROMPT 6

Instale fpdf2 (python -m pip install fpdf2) e acrescente o botão "Relatório executivo (PDF)",
de uma página: o resultado do mês (faturamento, margem, ocupação), a previsão dos próximos 2 meses,
a economia possível com rotas otimizadas, os 3 alertas mais graves e 3 recomendações escritas pelo app,
cada uma com o número que a justifica. Atualize os testes e publique a nova versão (como na Apostila IV).



CHECKPOINT

O PDF tem uma página, os números batem com o app e a nova versão está no ar pelo link da Apostila IV.

Desafio extra · Rotas com capacidade: dois caminhões

Na vida real, o caminhão enche. Com **capacidade**, o problema vira roteirização de veículos (VRP).

PROMPT 7

desafio

Use o OR-Tools para dividir as 6 entregas entre 2 caminhões de capacidade 10 toneladas (cargas: A 3 t, B 4 t, C 2 t, D 5 t, E 3 t, F 2 t), saindo e voltando ao depósito, com a menor distância total. Mostre a rota de cada caminhão, a carga e a distância, e compare com um caminhão só sem limite de capacidade. Explique o problema antes de escrever o código.



CHECKPOINT

Cada caminhão leva no máximo 10 t e a soma das cargas é 19 t.

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 SQL, a previsão (e o erro medido), a distância das rotas e as regras dos alertas 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 a rota ótima no mapa e a previsão dos próximos meses.
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 logística com otimização e previsão") 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 usando previsão e otimização que não tinha aprendido nas apostilas anteriores?

12 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.

13 Consulta rápida

Comandos

TERMINAL

```
python -m pip install ortools folium
streamlit-folium fpdf2
python -m pytest # roda
os testes
python -m streamlit run app.py # abre
o app
git add . ; git commit -m "..."; git
push # publica
```

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.

SQLite

Banco de dados num arquivo só, sem servidor.

SQL

Linguagem para consultar tabelas (SELECT, WHERE, GROUP BY).

Média móvel

Média dos últimos períodos, usada para prever o próximo.

Erro absoluto médio

Quanto a previsão erra, em média, para mais ou para menos.

Roteirização

Escolha da ordem das entregas.

OR-Tools

Biblioteca gratuita do Google para otimização.

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.