

// laboratório codex 05 · gestão 4.0

python + ia + app

>_ Laboratório Codex 05

O gestor inteligente, com IA

Banco de dados, previsão do movimento, escala ótima da equipe, painel interativo, alertas e relatório executivo: o gestor que decide com dados.

// apostila do aluno V · o projeto em etapas

01 Banco de dados



03 Escala ótima



05 Alertas



07 Escala de menor custo



02 Previsão do movimento



04 Painel do dono



06 Relatório executivo



08 Apresentação



Prof. Caio Haffner

Mestre em Desenvolvimento Econômico · Unicamp

1 Como usar esta apostila

Esta é a **Apostila V de Princípios de Gestão**, a etapa avançada da trilha: ela continua a **Apostila IV** (o Painel do Gestor no ar (Apostila IV)). Agora você vai dar **inteligência** ao Painel do Gestor do Café Bom Dia: um banco de dados com os indicadores, a previsão do movimento, a escala ótima da equipe, um painel interativo, alertas automáticos e um relatório executivo para o dono.

1. Guardar Banco de dados	2. Prever Clientes por dia	3. Otimizar Escala da equipe	4. Vigiar Alertas	5. Decidir Relatório executivo
-------------------------------------	--------------------------------------	--	-----------------------------	--



O PROJETO: O GESTOR INTELIGENTE

Nas Apostilas III e IV, o painel mostra os indicadores. Agora ele passa a ajudar a decidir: prevê quantos clientes virão, monta a escala com o menor número de atendentes, 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	Comparo média móvel e tendência e meço o erro.	☐
Otimizar a escala	Monto a escala com o menor número de atendentes.	☐
Mostrar num painel	Os resultados aparecem num painel interativo.	☐
Criar alertas	O app avisa quando um número sai do padrão.	☐
Relatar	O app gera o relatório executivo do mês.	☐



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, simulação 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 Painel do Gestor no ar, no GitHub, com testes e dados reais**. Se ainda não está assim, termine a IV antes.

Instale as bibliotecas novas (uma vez só)

TERMINAL

```
python -m pip install numpy scipy plotly fpdf2
```

Biblioteca	Para que serve
sqlite3	Banco de dados num arquivo (já vem com o Python).
numpy e scipy	Contas, simulação e otimização (a melhor decisão dentro dos limites).
plotly	Gráficos interativos dentro do app: zoom, filtro e valor ao passar o mouse.
fpdf2	Gerar o relatório executivo em PDF.

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

TERMINAL

```
Laboratorio_Codex_PG3/  
├─ dados/historico.db ← novo: banco de dados  
├─ calculos.py  
├─ previsao.py        ← novo  
├─ otimizacao.py      ← novo  
├─ painel.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

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

PROMPT 1

Crie dados/historico.db (SQLite, que já vem com o Python) com as tabelas indicadores (mês, nome, meta, real), vendas (dia, clientes, ticket) e equipe (nome, turno), importando os dados das Apostilas III e IV. Crie banco.py com funções que usem SQL para:

1. os clientes por dia da semana (média);
2. as reclamações por mês, contra a meta;
3. o ticket médio de cada mês.

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



CHECKPOINT

Os indicadores do banco são iguais aos do painel 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: o que vem por aí?

Para decidir o próximo mês, é preciso uma **previsão**. Duas formas simples e honestas: média móvel e tendência.



média móvel de 3 períodos = média dos 3 últimos ·
tendência: valor = $a + b \times \text{período}$ (regressão)

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



CLIENTES POR DIA (MÉDIA DO MÊS, 12 MESES)

180 · 185 · 182 · 190 · 196 · 193 · 200 · 206 · 203 · 210 · 215 · 212

PROMPT 2

Crie `previsao.py` com:

1. `media_movel(serie, n=3)` e a previsão do período 13;
2. `tendencia(serie)`: a e b da reta por mínimos quadrados e a previsão dos períodos 13 e 14;
3. o erro absoluto médio da média móvel nos períodos 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 (período 13)	212,3	
Tendência: a e b	$a \approx 176,8$ · $b \approx 3,22$	
Tendência (períodos 13 e 14)	218,6 e 221,8	
Erro absoluto médio da média móvel	6,7	



PARA PENSAR

O movimento cresce ≈ 3 clientes por dia a cada mês. Quando o café vai precisar de mais um atendente? Que outros dados ajudariam a prever (férias, chuva, feriados)?

5

Etapa 3 · Otimizar: a escala da equipe

Quantos atendentes o Café Bom Dia precisa, no mínimo, para atender bem o dia inteiro? Como pessoas não se dividem, é um problema de **programação inteira**.



O DIA DO CAFÉ

Atendentes necessários: 6h–10h: 2 · 10h–14h: 5 · 14h–18h: 4 · 18h–22h: 2. Turnos de 8 horas: 6h–14h, 10h–18h e 14h–22h.

PROMPT 3

Crie `otimizacao.py` com `scipy.optimize.milp` (programação inteira):

1. quantos atendentes em cada turno, com o menor total, cobrindo a necessidade de cada faixa;
2. quantos atendentes ficam em cada faixa e onde sobra gente;
3. uma conferência por força bruta (todas as combinações de 0 a 6 em cada turno).

Antes do código, escreva o problema em português: o que decidir, o que minimizar e os limites.

Resultado	Esperado	Calculado
Total mínimo de atendentes	7	
Uma escala ótima (6h, 10h, 14h)	2 · 3 · 2 (há outras)	
Na escala 2 · 3 · 2, a faixa em que sobra gente	14h–18h (1 a mais)	



PARA PENSAR

Há mais de uma escala com 7 atendentes. Que critério de gestão (folgas, preferências, custo) ajuda a escolher entre elas?

Etapa 4 · Mostrar: o painel do dono

O dono quer ver, numa tela, se o café vai bem e se a equipe está bem dimensionada.

PROMPT 4

Acrescente ao app a aba "Dono" com plotly: os indicadores do mês com o semáforo (meta × real), o movimento com a previsão dos próximos 2 meses tracejada, e a escala do dia em barras (necessário × escalado por faixa). Controles deslizantes mudam a necessidade de cada faixa e refazem a escala.



CHECKPOINT

Com a necessidade 2 · 5 · 4 · 2 o painel mostra 7 atendentes, e a escala muda quando um controle muda.

7

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 usuário pode ajustar na barra lateral:

1. reclamações acima da meta;
2. ticket médio 10% abaixo do mês anterior;
3. equipe escalada abaixo do mínimo em alguma faixa;
4. previsão do movimento acima de 215 clientes por dia.

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?
Reclamações acima da meta		
Ticket médio caiu 10%		
Equipe abaixo do mínimo		
Previsão acima de 215 clientes		



ALERTA DEMAIS É ALERTA NENHUM

Se aparecerem 30 alertas, ninguém olha mais. Ajuste as regras para mostrar só o que exige decisão.

Etapa 6 · Decidir: o relatório executivo

Quem decide 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 (indicadores com o semáforo), a previsão do movimento dos próximos 2 meses,
a escala ótima da equipe, 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.

9

Desafio extra · A escala de menor custo

O turno da noite custa mais (adicional). A escala de menos pessoas é também a mais barata?

PROMPT 7**desafio**

Cada atendente custa R\$ 100 por dia nos turnos 6h-14h e 10h-18h e R\$ 120 no turno 14h-22h.

Refaça a otimização minimizando o custo total e compare com a Etapa 3. Depois, aumente a necessidade das 18h-22h para 3 e veja o que muda. Explique o resultado em 3 linhas antes de mostrar o código.

**CHECKPOINT**

Custo mínimo R\$ 740 por dia, com 7 atendentes e só 2 no turno da noite; com 3 às 18h-22h, o total sobe para 8.

10

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 Princípios de Gestão. Não faça alterações. Confira o SQL, a previsão (e o erro medido), os limites da escala 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 escala ótima mudando com a necessidade de cada faixa.
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 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 Princípios de Gestão 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 numpy scipy
plotly 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.

Programação inteira

Otimização em que a resposta precisa ser número inteiro (pessoas).

Turno

Período de trabalho de um atendente (aqui, 8 horas).

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.