

// laboratório codex 05 : projetos 4.0

python + ia + app

>_ Laboratório Codex 05

O projeto inteligente, com IA

Banco de dados, previsão de horas, prazo com Monte Carlo, painel interativo, alertas e relatório executivo: o projeto que ajuda a decidir.

// apostila do aluno V · o projeto em etapas

01 Banco de dados

03 PERT e Monte Carlo

05 Alertas

07 Prazo seguro

02 Previsão de horas

04 Painel do patrocinador

06 Relatório executivo

08 Apresentação



Prof. Caio Haffner

Mestre em Desenvolvimento Econômico · Unicamp

1 Como usar esta apostila

Esta é a **Apostila V de Gerenciamento de Projetos**, a etapa avançada da trilha: ela continua a **Apostila IV** (o App de Gestão do Projeto no ar (Apostila IV)). Agora você vai dar **inteligência** ao App de Gestão do Projeto: um banco de dados com os apontamentos, a previsão das horas, a simulação do prazo com incerteza, um painel interativo, alertas automáticos e um relatório executivo para o patrocinador.

1. Guardar

Banco de dados

2. Prever

Horas trabalhadas
por semana

3. Simular

Prazo com
incerteza

4. Vigiar

Alertas

5. Decidir

Relatório
executivo



O PROJETO: O PROJETO INTELIGENTE

Nas Apostilas III e IV, o app acompanha o projeto. Agora ele passa a ajudar a decidir: prevê as horas da equipe, simula o prazo com incerteza para dizer a chance de entregar em dia, avisa quando algo sai do padrão e escreve o relatório da semana.

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.	☐
Simular o prazo	Uso PERT e Monte Carlo para dar a chance de entregar em dia.	☐
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.	☐

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, 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 App de Gestão do Projeto 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_GP3/  
├─ dados/historico.db  ← novo: banco de dados  
├─ calculos.py  
├─ previsao.py        ← novo  
├─ simulacao.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 tarefas (código, nome, duração, predecessoras), apontamentos (semana, tarefa, horas) e custos (semana, planejado, real), importando os dados das Apostilas III e IV. Crie banco.py com funções que usem SQL para:

1. as horas trabalhadas por semana;
2. o custo real e o planejado de cada tarefa;
3. as tarefas que terminaram depois do previsto.

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



CHECKPOINT

O custo real e o planejado batem com a curva S do app da Apostila III para a mesma semana.



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



HORAS TRABALHADAS POR SEMANA (12 SEMANAS)

30 · 34 · 32 · 36 · 40 · 38 · 42 · 45 · 43 · 47 · 50 · 48

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)	48,3	
Tendência: a e b	$a \approx 29,0$ · $b \approx 1,75$	
Tendência (períodos 13 e 14)	51,8 e 53,6	
Erro absoluto médio da média móvel	3,6	



PARA PENSAR

As horas crescem $\approx 1,75$ por semana. Uma equipe de 2 pessoas com 25 horas cada aguenta a previsão? Quando é hora de pedir reforço?

5

Etapa 3 · Simular: a chance de entregar em dia

Uma duração única esconde o risco. Com três estimativas por tarefa (**PERT**) e milhares de simulações (**Monte Carlo**), o app diz a **chance** de cumprir o prazo.



O CAMINHO CRÍTICO (DIAS: OTIMISTA · MAIS PROVÁVEL · PESSIMISTA)

A: 1 · 2 · 4 | B: 2 · 3 · 6 | D: 3 · 4 · 7 | F: 2 · 3 · 7. Prazo combinado com o cliente: 14 dias.

PROMPT 3

Crie simulacao.py com:

1. `pert(o, m, p)`: a duração esperada $(o + 4m + p) \div 6$ e o desvio padrão $(p - o) \div 6$ de cada tarefa;
2. a duração esperada do caminho, o desvio padrão (raiz da soma das variâncias) e a chance de terminar em até 14 dias pela distribuição normal (`scipy.stats.norm`);
3. uma simulação de Monte Carlo com 10.000 projetos (`numpy`, distribuição triangular de cada tarefa) e o histograma da duração no `plotly`.

Explique antes, em 5 linhas, por que somar as durações mais prováveis (12 dias) engana.

Resultado	Esperado (PERT)	Calculado
Duração esperada do caminho	≈ 13,33 dias	
Desvio padrão	≈ 1,35 dia	
Chance de terminar em até 14 dias	≈ 69%	



PARA PENSAR

A soma das durações mais prováveis dá 12 dias, mas a chance de cumprir 14 dias é só ≈ 69%. Por que as tarefas atrasam mais do que adiantam?

6

Etapa 4 · Mostrar: o painel do patrocinador

O patrocinador quer saber três coisas: está no prazo, está no custo e qual a chance de entregar em dia.

PROMPT 4

Acrescente ao app a aba "Patrocinador" com plotly: o Gantt com o caminho crítico em vermelho, a curva S (planejado × real), o histograma do Monte Carlo com o prazo marcado e um controle deslizante do prazo (12 a 18 dias) que mostra a chance de cumprir. Ao passar o mouse, mostre os valores.



CHECKPOINT

Com prazo de 14 dias o painel mostra $\approx 69\%$ (PERT) e um valor próximo pelo Monte Carlo.

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. IDC (custo) abaixo de 0,9;
2. IDP (prazo) abaixo de 0,9;
3. tarefa do caminho crítico atrasada;
4. horas previstas acima da capacidade da equipe.

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?
IDC abaixo de 0,9		
IDP abaixo de 0,9		
Tarefa crítica atrasada		
Horas previstas acima da capacidade		



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 status da semana (IDC, IDP, farol e tarefas atrasadas), a previsão de horas das próximas 2 semanas,
a chance de entregar no prazo, 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 · Que prazo prometer?

Se o cliente quer 90% de segurança, que prazo prometer? A simulação responde.

PROMPT 7

desafio

Use a Etapa 3 para achar o prazo com 90% de chance de cumprimento de duas formas: pela normal ($\text{média} + 1,28 \times \text{desvio padrão}$) e pelo percentil 90 da simulação (`numpy.percentile`). Mostre também o prazo para 50%, 80% e 95%, numa tabela e no histograma. Explique o resultado em 3 linhas antes de mostrar o código.



CHECKPOINT

Pela normal, o prazo de 90% é $\approx 15,1$ dias; a simulação dá um valor próximo, e 95% pede mais dias que 90%.

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 Gerenciamento de Projetos. Não faça alterações. Confira o SQL, a previsão (e o erro medido), as fórmulas do PERT 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 chance de entregar em dia mudando com o prazo.
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 projetos com simulação 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 Gerenciamento de Projetos usando previsão e simulação 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.

PERT

Estimativa com três durações: otimista, mais provável e pessimista.

Monte Carlo

Simular milhares de projetos possíveis para medir o risco.

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.