

// laboratório codex 03 · gerenciamento de projetos

python + ia + app

>_ Laboratório Codex 03

Gestão de projetos em Python, com IA

Cronograma, Gantt, curva S, riscos e farol num app: você constrói a ferramenta de um gerente de projetos e gera o relatório de status em um clique.

// apostila do aluno III · o projeto em etapas

01 Dados do projeto



03 Gantt, curva S e riscos



05 Simulações



07 Nivelamento



02 Caminho crítico e EVM



04 Cronograma editável



06 Relatório de status



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 projetos** com cronograma editável, gráfico de Gantt, curva S do valor agregado, matriz de riscos e relatório de status.

1. Dados

Atividades e riscos em CSV

2. Cálculos

CPM, valor agregado, riscos

3. Gráficos

Gantt, curva S, matriz

4. App

Cronograma editável

5. Status

Simulação e Excel



O PROJETO: APP DE GESTÃO DO PROJETO

As atividades e os riscos ficam em planilhas. O app calcula o caminho crítico, desenha o Gantt, acompanha o valor agregado semana a semana com a curva S, mostra a matriz de riscos, simula atrasos e gera o relatório de status com farol em Excel.

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Estruturar o projeto	Guardo atividades, avanço e riscos em planilhas.	☐
Calcular o cronograma	Caminho crítico e folgas a partir dos dados.	☐
Desenhar o Gantt	Mostro as atividades no tempo, com o caminho crítico em destaque.	☐
Acompanhar o valor agregado	Curva S com planejado, agregado e real.	☐
Priorizar riscos	Matriz de probabilidade e impacto em gráfico.	☐
Reportar	Relatório de status com farol, 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_GP3** em Documentos e abra no VS Code. Ao longo das etapas, ela vai ficar assim:

TERMINAL

```
Laboratorio_Codex_GP3/  
├─ 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_GP3_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 · Os dados do projeto

O projeto do novo site da Clínica Vida (Apostila II) vira planilhas: atividades, avanço semanal e riscos.

PROMPT 1

Crie três arquivos em dados/:

1. `atividades.csv`: `codigo`, `descricao`, `duracao_dias`, `predecessoras`, `custo_orcado`
(A Levantamento 3 dias R\$ 9.000; B Layout 4, depois de A, R\$ 16.000;
C Conteúdo 2, depois de A, R\$ 6.000; D Programação 5, depois de B, R\$ 45.000;
E Fotos 3, depois de C, R\$ 9.000; F Testes 2, depois de D e E, R\$ 15.000);
 2. `avanco.csv`: `semana`, `valor_planejado`, `valor_agregado`, `custo_real` (acumulados);
use semanas 1 a 3 com: 25.000/20.000/22.000 · 50.000/40.000/45.000 ·
75.000/62.000/70.000;
 3. `riscos.csv`: `descricao`, `probabilidade`, `impacto` (os 4 riscos da Apostila II).
- Em `calculos.py`, crie funções para ler os três arquivos e conferir os dados
(predecessoras que existem, probabilidades entre 0 e 1).



CHECKPOINT

As funções leem os três CSVs e avisam se algum dado estiver errado (teste colocando uma predecessora "Z").

5 Etapa 2 · Os cálculos

O caminho crítico, o valor agregado e os riscos da Apostila II, agora lendo as planilhas.

PROMPT 2

Em `calculos.py`, crie:

1. `cronograma(atividades)`: início e término cedo e tarde, folga e se é crítica;
 2. `valor_agregado(avanco, orcamento_total)`: para cada semana, IDC, IDP, estimativa no término e o farol (verde $\geq 0,95$; amarelo $\geq 0,85$; vermelho);
 3. `riscos_priorizados(riscos)`: VME e classificação, do maior para o menor.
- O orçamento total é a soma dos custos das atividades.

Conferência	Esperado	Calculado
Duração do projeto	14 dias	
Caminho crítico	A-B-D-F	
Orçamento total	R\$ 100.000	
IDC e IDP na semana 2	0,89 e 0,80	
Farol na semana 3		

Etapa 3 · Gantt, curva S e matriz

Os três gráficos que aparecem em toda reunião de projeto.

PROMPT 3

Crie graficos.py com:

1. `grafico_gantt`: barras horizontais (matplotlib `barh`) de cada atividade do início ao término cedo, com as críticas em vermelho e a folga desenhada mais clara;
2. `grafico_curva_s`: linhas acumuladas do valor planejado, agregado e custo real por semana, com o orçamento total como linha horizontal;
3. `grafico_riscos`: dispersão com probabilidade no eixo horizontal e impacto no vertical, tamanho do ponto pelo VME e o nome de cada risco ao lado.



CHECKPOINT

No Gantt, A, B, D e F estão em vermelho; C e E mostram 4 dias de folga.

7

Etapa 4 · O app com cronograma editável

O gerente atualiza o cronograma na própria tela, como numa planilha, e tudo se recalcula.

PROMPT 4

Crie app.py com Streamlit:

- aba "Cronograma": a tabela de atividades editável (st.data_editor); ao editar uma duração, o caminho crítico e o Gantt se recalculam;
- aba "Custos": a curva S e os indicadores da última semana com st.metric (IDC, IDP, estimativa no término) e o farol colorido;
- aba "Riscos": a tabela priorizada, a matriz e a reserva de contingência.

TERMINAL

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

Teste no app	Nova duração	Caminho crítico
D com 8 dias		
E com 6 dias		
E com 8 dias		

O patrocinador pergunta: "e se?". O app responde na hora.

PROMPT 5

Acrescente uma aba "Simulações":

1. atraso: escolher uma atividade e os dias de atraso; mostrar a nova duração, se o caminho crítico mudou e o Gantt antes e depois;
2. PERT: para as atividades críticas, receber O, M e P e mostrar a duração esperada, o desvio e a probabilidade de terminar até uma data, com um gráfico da curva normal.



CHECKPOINT

Atrasando E em 5 dias, o app mostra que o caminho crítico mudou para A-C-E-F.

Etapa 6 • O relatório de status

Toda semana, o relatório de status para o patrocinador, em um clique.

PROMPT 6

Acrescente um botão "Baixar relatório de status" que gere um Excel com as abas: "Status" (farol, duração, IDC, IDP, estimativa no término, os 3 maiores riscos e um resumo de 3 linhas escrito pelo app), "Cronograma", "Valor agregado" e "Riscos".



CHECKPOINT

A aba Status mostra o farol e o resumo; os números batem com o app.

10 Desafio extra · Nivelamento de recursos

O Gantt supõe gente sobrando. E se só houver um designer para Layout e Fotos?

PROMPT 7

desafio

Acrescente às atividades uma coluna "recurso" (quem faz). Se duas atividades do mesmo recurso se sobrepõem, adie a que tem mais folga até o recurso ficar livre. Mostre o Gantt antes e depois do nivelamento e a nova duração. Explique o que é nivelamento de recursos antes de escrever o código.

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 de Projetos. Não faça alterações. Confira as dependências, o cálculo de ida e volta, as fórmulas de IDC e IDP e a regra do farol 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 Gantt com o caminho crítico e o farol mudando quando uma atividade atrasa.
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") 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 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</code> · <code>st.header</code>	Títulos
<code>st.number_input</code> · <code>st.slider</code>	Entradas de número
<code>st.selectbox</code> · <code>st.file_uploader</code>	Escolha e upload
<code>st.metric</code> · <code>st.dataframe</code>	Indicador e tabela
<code>st.pyplot(fig)</code>	Mostrar um gráfico
<code>st.tabs</code> · <code>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.

Gantt

Gráfico de barras das atividades no tempo.

Curva S

Valor acumulado do projeto ao longo do tempo (planejado, agregado, real).

Linha de base

O plano aprovado, usado para comparar.

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.