

// laboratório codex 02 · gerenciamento de projetos

python + ia

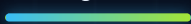
>_ Laboratório Codex 02

Gestão de projetos em Python, com IA

Cronograma e caminho crítico, PERT, valor agregado, riscos, burndown e viabilidade: você constrói as ferramentas de um gerente de projetos e gera um relatório de status automático.

// apostila do aluno II · trilha de exercícios

01 Cronograma e caminho crítico



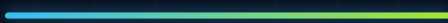
03 Valor agregado



05 Sprint e burndown



07 E se atrasar?



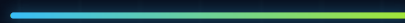
02 Estimativas com PERT



04 Matriz de riscos



06 O projeto vale a pena?



08 Relatório de status



Prof. Caio Haffner

Mestre em Desenvolvimento Econômico · Unicamp

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 **continuação da Apostila I** (Laboratório Codex 01). Lá você instalou as ferramentas e aprendeu a conversar com o Codex. Aqui você vai **praticar**: cada exercício é uma **ferramenta do gerente de projetos**, do cronograma à viabilidade. Vários usam o mesmo projeto fictício: o novo site da Clínica Vida.

1. Revisar

Checklist e lembretes da Apostila I

2. Tempo

Caminho crítico e PERT

3. Custo

Valor agregado

4. Riscos e ágil

Matriz de riscos e burndown

5. Status

Relatório de status automático

Os sinais desta apostila

Quando aparecer	O que significa
Cartão roxo · PROMPT	Texto para copiar e colar no Codex. Leia antes de enviar.
Quadrado tracejado · CHECKPOINT	O que você deve estar vendo. Se não estiver, pare e peça ajuda.
Caixa azul · DICA	Atalho, pergunta para pensar ou jeito mais fácil de fazer.
Caixa laranja · ATENÇÃO	Ponto em que muita gente erra (e a IA também).

Ao final, você deverá conseguir

Habilidade	Como saber que conseguiu	Ok
Montar o cronograma	Calculo folgas, caminho crítico e prazo do projeto.	☐
Estimar com incerteza	Uso o PERT para dar prazos com probabilidade.	☐
Controlar custo e prazo	Calculo e interpreto IDC, IDP e a estimativa no término.	☐
Gerenciar riscos	Priorizo riscos pelo valor esperado e defino a reserva.	☐
Acompanhar o ágil	Leio um burndown e projeto o fim da sprint.	☐
Justificar o projeto	Calculo VPL, TIR e payback.	☐

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.** E aqui vale mais uma: **o plano vai mudar**; o valor das ferramentas está em mostrar cedo o desvio para corrigir a rota.

2 Antes de começar

A instalação está na **Apostila I** (Laboratório Codex 01, seções 3 a 7). Se alguma coisa abaixo não estiver pronta, volte lá antes de seguir.

O que precisa estar pronto	Onde está na Apostila I	Ok
VS Code instalado e abrindo	Seção 4	☐
Python funcionando no terminal (<code>python --version</code> ou <code>py --version</code>)	Seção 5	☐
Extensão do Codex instalada e com login feito	Seção 6	☐
Você sabe criar uma pasta de projeto e abrir no VS Code	Seção 7	☐

Crie a pasta desta apostila

- 1 Na pasta Documentos, crie uma pasta chamada `Laboratorio_Codex_GP`.
- 2 No VS Code, use **Arquivo** → **Abrir Pasta** e escolha essa pasta.
- 3 Abra o painel do Codex e confira que ele está conectado à sua conta.

**CHECKPOINT**

O Explorer mostra a pasta `Laboratorio_Codex_GP` e o Codex responde a um "olá".

**NÃO TEM A APOSTILA I?**

Peça ao professor o PDF do **Laboratório Codex 01 · Apostila do aluno**. As seções 3 a 7 levam cerca de 30 minutos.



O CICLO DE TRABALHO COM A IA

Pedir um **plano** → ler → aprovar → **executar** → **testar** com contas feitas por você → **explicar** o código com suas palavras.



MÉTODO DE DEPURAÇÃO

Diagnóstico → explicação → solução → alteração → teste. **Peça o diagnóstico antes da correção.**



A VÍRGULA BRASILEIRA

No Python, o decimal é com **ponto** (**13.75**). Em todos os programas desta apostila, peça ao Codex para **aceitar vírgula ou ponto**. E digite os milhares **sem ponto** (**1000** , e não **1.000**): um programa que aceita vírgula ou ponto pode ler 1.000 como 1.



SEGURANÇA

Só aprove o que você entende.
Trabalhe só na pasta do projeto. Nunca coloque senhas ou chaves no código.

PROMPT UNIVERSAL DE DIAGNÓSTICO

serve para qualquer erro

Estou com este problema no meu projeto.

Erro/mensagem: [COLE A MENSAGEM]

Arquivo envolvido: [NOME DO ARQUIVO]

Antes de alterar qualquer coisa:

1. explique a causa provável;
2. diga como confirmar a causa;
3. proponha a menor correção;
4. aguarde minha autorização para alterar o arquivo.

4

Exercício 1 · Cronograma e caminho crítico

Num projeto, algumas atividades dependem de outras. O **caminho crítico** é a sequência que determina o prazo: qualquer atraso nele atrasa o projeto inteiro.



início cedo = maior término cedo das predecessoras · término cedo = início cedo + duração
folga = início tarde - início cedo · folga zero = caminho crítico

Método do Caminho Crítico (CPM): ida para os cedos, volta para os tardes

PROMPT 1

Crie um programa em Python chamado `cronograma.py`.

O usuário informa as atividades do projeto: código, descrição, duração (dias) e predecessoras (códigos separados por vírgula). O programa deve:

1. calcular início e término cedo (ida) e início e término tarde (volta);
2. calcular a folga de cada atividade;
3. mostrar o caminho crítico e a duração total do projeto;
4. desenhar um gráfico de Gantt simples no terminal (com `█` por dia).

Antes de criar o arquivo, apresente um plano curto.

Projeto **Novo site da Clínica Vida** (fictício): A Levantamento (3 dias) · B Layout (4, depois de A) · C Conteúdo (2, depois de A) · D Programação (5, depois de B) · E Fotos (3, depois de C) · F Testes e publicação (2, depois de D e E).

Atividade	Início cedo	Término cedo	Folga	Crítica?
A				
B				
C				
D				
E				
F				

Duração do projeto: _____ dias · Caminho crítico: _____

5

Exercício 2 · Estimativas com incerteza (PERT)

Ninguém sabe exatamente quanto uma atividade vai durar. O **PERT** usa três estimativas (otimista, mais provável e pessimista) para chegar à duração esperada e à incerteza.



$$\text{duração esperada} = (O + 4M + P) \div 6 \quad \cdot \quad \text{desvio-padrão} = (P - O) \div 6$$

No caminho crítico, somam-se as médias e as variâncias (desvio²)

PROMPT 2

Crie um programa em Python chamado `pert.py`.

Para cada atividade do caminho crítico, o usuário informa O, M e P (dias).

O programa deve:

1. calcular a duração esperada e o desvio-padrão de cada atividade;
2. somar as durações e as variâncias do caminho e calcular o desvio total;
3. calcular a probabilidade de terminar até um prazo informado, usando a distribuição normal (`math.erf` da biblioteca padrão);
4. dizer qual prazo dá 90% de chance de cumprimento.

Antes de escrever o código, apresente um plano curto.

Atividade	O	M	P	Esperada	Desvio
Layout	2	3	6		
Programação	3	4	8		
Testes	4	5	9		

Duração esperada do caminho: _____ dias · Chance de terminar em até 14 dias: _____ %



PARA PENSAR

O cliente quer uma data. Você daria a duração esperada ou a de 90% de chance? Por quê?

6

Exercício 3 · Valor agregado (EVM)

No meio do projeto, o patrocinador pergunta: "**estamos no prazo e no orçamento?**". O gerenciamento do valor agregado responde com números.

★ $VP = \text{planejado até hoje} \cdot VA = \text{valor do trabalho feito} \cdot CR = \text{custo real}$

$$IDC = VA \div CR \cdot IDP = VA \div VP \cdot ENT \text{ (estimativa no término)} = ONT \div IDC$$

ONT = orçamento no término · IDC/IDP abaixo de 1 = acima do custo / atrasado

PROMPT 3

Crie um programa em Python chamado `valor_agregado.py`.

O usuário informa o orçamento total (ONT), o valor planejado até hoje (VP), o valor agregado (VA) e o custo real (CR). O programa deve calcular: variação de custo ($VA - CR$), variação de prazo ($VA - VP$), IDC, IDP, estimativa no término ($ONT \div IDC$), estimativa para terminar ($ENT - CR$) e variação no término ($ONT - ENT$). No fim, deve dar um diagnóstico em linguagem simples: atrasado? acima do orçamento? quanto vai custar no total? Antes de escrever o código, apresente um plano curto.

Projeto com orçamento de **R\$ 100.000**. Até hoje: planejado R\$ 50.000 · feito (valor agregado) R\$ 40.000 · gasto R\$ 45.000.

Indicador	Você espera	O programa mostrou
Variação de custo		
Variação de prazo		
IDC		
IDP		
Estimativa no término		
Variação no término		

7

Exercício 4 · Matriz de riscos

Risco é um evento incerto que, se acontecer, afeta o projeto. A **matriz de probabilidade e impacto** mostra quais riscos merecem atenção primeiro.



valor monetário esperado (VME) = probabilidade × impacto

A soma dos VMEs ajuda a dimensionar a reserva de contingência

PROMPT 4

Crie um programa em Python chamado `riscos.py`.

Para cada risco, o usuário informa: descrição, probabilidade (0 a 1) e impacto em reais. O programa deve:

1. calcular o valor monetário esperado de cada risco;
2. classificar a probabilidade e o impacto em baixa, média ou alta (defina e explique os limites que você usou);
3. ordenar os riscos do maior para o menor VME;
4. sugerir a reserva de contingência (soma dos VMEs) e uma resposta (evitar, mitigar, transferir ou aceitar) para os dois maiores.

Antes de escrever o código, apresente um plano curto.

Risco	Probabilidade	Impacto (R\$)	VME (R\$)
Atraso do fornecedor	50%	40.000	
Saída de membro-chave	20%	60.000	
Mudança de escopo	60%	25.000	
Falha técnica	10%	100.000	

Reserva de contingência sugerida: R\$ _____

8

Exercício 5 · Sprint e gráfico de burndown

Nos projetos ágeis, o trabalho é medido em **pontos**. O **burndown** compara, dia a dia, os pontos que faltam com a linha ideal até o fim da sprint.

★
$$\text{linha ideal no dia } d = \text{pontos totais} - (\text{pontos totais} \div \text{dias}) \times d$$

$$\text{velocidade média} = \text{pontos entregues} \div \text{dias decorridos}$$

Sprint de 10 dias com 40 pontos

PROMPT 5

Crie um programa em Python chamado burndown.py.

O usuário informa os pontos da sprint, o número de dias e os pontos restantes ao fim de cada dia. O programa deve:

1. calcular a linha ideal de cada dia;
2. mostrar dia a dia: ideal, real e se o time está adiantado ou atrasado;
3. desenhar o burndown no terminal (dois caracteres diferentes para ideal e real);
4. com a velocidade média, estimar quantos dias a mais seriam necessários.

Antes de escrever o código, apresente um plano curto.

Pontos restantes ao fim de cada dia (do dia 0 ao 10): 40 · 38 · 35 · 35 · 30 · 26 · 22 · 20 · 14 · 8 · 3

Pergunta	Sua resposta	O programa mostrou
Ideal no dia 5		
Velocidade média		
A sprint terminou?		
Dias a mais necessários		

9

Exercício 6 · O projeto vale a pena?

Antes de começar, o projeto precisa de um **caso de negócio**. O dinheiro investido hoje volta? Em quanto tempo? Rende mais que deixar o dinheiro aplicado?



$VPL = \sum \text{fluxo do ano } t \div (1 + \text{taxa})^t$ · TIR = taxa em que o VPL é zero

Payback = ano em que o fluxo acumulado fica positivo

PROMPT 6

Crie um programa em Python chamado `viabilidade.py`.

O usuário informa o investimento inicial, os fluxos de caixa de cada ano e a taxa mínima de atratividade (%). O programa deve calcular:

1. o VPL;
2. a TIR (por tentativa e erro ou bisseção; explique o método);
3. o payback simples (fluxo acumulado);
4. a recomendação: aprovar se $VPL > 0$.

Antes de escrever o código, apresente um plano curto.

Investimento: R\$ 100.000 · fluxos: ano 1 R\$ 30.000 · ano 2 R\$ 35.000 · ano 3 R\$ 40.000 · ano 4 R\$ 40.000.

Indicador	Você espera	O programa mostrou
VPL a 10% ao ano		
VPL a 15% ao ano		
TIR		
Payback		

10 Desafio extra · E se atrasar?

O fornecedor avisou: a atividade **D (Programação)** vai atrasar **3 dias**. E se fosse a atividade **E (Fotos)**? O caminho crítico responde na hora.

PROMPT 7

desafio

Altere o `cronograma.py` para permitir simular atrasos: o usuário escolhe uma atividade e quantos dias ela atrasa, e o programa recalcula tudo e mostra: a nova duração do projeto, se o caminho crítico mudou e quais atividades perderam folga. Explique por que atrasar uma atividade com folga nem sempre atrasa o projeto.

Simulação	Nova duração	Caminho crítico mudou?
D atrasa 3 dias		
E atrasa 3 dias		
E atrasa 5 dias		

11 Use o Codex como professor

Escolha um dos seus programas e use os três prompts abaixo. Anote no diário de prompts o que aprendeu.

EXPLICAR

Explique este código linha por linha para um estudante de Gerenciamento de Projetos que conhece o conteúdo, mas nunca programou em Python. Não reescreva o código.

REVISAR O CONTEÚDO

Revise meu programa como um professor de Gerenciamento de Projetos. Não faça alterações. Confira se os conceitos e as contas estão corretos (dependências, cálculos de ida e volta, fórmulas de valor agregado, unidades) e liste os riscos de entrada inválida.

TESTAR

Crie uma tabela de casos de teste para este programa, com o resultado esperado de cada caso: um caso normal, um caso-limite, valores com centavos ou decimais e uma entrada inválida.

**CONFIRA A TABELA DE TESTES**

Os "resultados esperados" que a IA escreve também podem estar errados. Confira pelo menos dois na calculadora.

Atividade final · Relatório de status do projeto

Toda semana, o gerente de projetos apresenta um **relatório de status**: prazo, custo, riscos e próximos passos, com um farol verde, amarelo ou vermelho. Seu programa vai gerá-lo sozinho.

Requisitos

- ☐ Ler as atividades e calcular o cronograma (Exercício 1)
- ☐ Calcular o valor agregado e o IDC/IDP (Exercício 3)
- ☐ Listar os 3 maiores riscos (Exercício 4)
- ☐ Dar um farol: verde ($IDC \text{ e } IDP \geq 0,95$), amarelo ($\geq 0,85$) ou vermelho
- ☐ Gerar o relatório num arquivo de texto (status.txt)
- ☐ Escrever um resumo em 3 linhas para o patrocinador



ORGANIZE OS DADOS

Guarde as atividades e os riscos em arquivos CSV (**atividades.csv** e **riscos.csv**). Assim, na próxima semana, é só atualizar a planilha e rodar de novo.

PROMPT A

planejamento

Quero criar em Python um gerador de relatório de status de projeto que junte: cronograma com caminho crítico, valor agregado e matriz de riscos.

Antes de escrever código:

1. proponha o formato dos arquivos atividades.csv e riscos.csv;
2. proponha as funções, reaproveitando os exercícios;
3. defina a regra do farol (verde, amarelo, vermelho).

Não escreva o código ainda.

PROMPT B

implementação

Agora implemente o gerador em status.py. Ele deve ler os CSVs, mostrar o relatório no terminal e salvá-lo em status.txt, com o farol e um resumo de 3 linhas para o patrocinador.

Com os dados desta apostila	Resultado
Duração prevista	
IDC / IDP	
Farol	
Maior risco	

13 Entrega e reflexão

Diário de prompts

#	O que pedi (resumo)	O que a IA fez	Precisei corrigir?
1			
2			
3			

O que entregar

Item	Entregue
Pasta do projeto com os arquivos .py dos exercícios e da atividade final	☐
As tabelas de teste desta apostila preenchidas	☐
Um erro de conteúdo que a IA cometeu (ou que você encontrou) e como foi corrigido	☐
O diário de prompts (acima) e a reflexão final (abaixo)	☐
Registro da auditoria humana (opcional, vale ponto extra)	☐

Reflexão individual

1. Qual ferramenta de projetos ficou mais clara depois que você a programou?

2. Em que momento a IA errou um cálculo de projeto, e como você percebeu?

3. O seu relatório de status deu qual farol? O que você faria para melhorar?

4. Por que é melhor descobrir um atraso pelo IDP na semana 3 do que no último dia?

14 Como você será avaliado

Critério	0	1	2	3
Conteúdo da disciplina	Conceitos errados	Acertou alguns	Acertou quase todos	Todos corretos e conferidos
Uso do Codex	Não conseguiu	Só copiou	Usou com alguma compreensão	Usou de forma crítica
Programação	Não funciona	Funciona em parte	Funciona	Funciona e está bem organizado
Testes	Não testou	Um teste	Vários testes	Testes e casos-limite
Depuração	Não identificou	Precisou da solução pronta	Identificou com ajuda	Diagnosticou e corrigiu
Explicação	Não explica	Explica pouco	Explica a lógica	Explica a lógica e o significado para o projeto

Total máximo: 18 pontos, convertidos para a escala da disciplina.

As contas desta apostila

Término cedo	$\text{início cedo} + \text{duração}$
Folga	$\text{início tarde} - \text{início cedo}$
PERT	$(O + 4M + P) / 6$
IDC · IDP	$VA/CR \cdot VA/VP$
ENT	ONT / IDC
VME	$\text{probabilidade} \times \text{impacto}$
VPL	$\sum \text{fluxo} / (1 + \text{taxa})^t$

Comandos

TERMINAL

```
python arquivo.py    # executa um
programa
Ctrl + C              # interrompe um
programa travado
```

Glossário

Caminho crítico

Sequência de atividades sem folga; define o prazo.

Folga

Quanto uma atividade pode atrasar sem atrasar o projeto.

PERT

Estimativa com três cenários: otimista, provável e pessimista.

Valor agregado

Valor orçado do trabalho realmente feito.

IDC / IDP

Índices de desempenho de custo e de prazo.

Reserva de contingência

Dinheiro guardado para os riscos identificados.

Sprint

Ciclo curto de trabalho nos métodos ágeis.

Burndown

Gráfico do trabalho restante ao longo da sprint.

VPL e TIR

Indicadores de viabilidade financeira do projeto.

1

LEMBRE-SE

A IA é uma ferramenta. Conhecimento, julgamento, validação e responsabilidade continuam sendo de quem programa, e de quem lidera o projeto.

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.