

// laboratório codex 02 · logística 4.0

python + ia

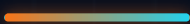
>_ Laboratório Codex 02

Logística 4.0 em Python, com IA

Curva ABC, lote econômico, ponto de pedido, nível de serviço, centralização, efeito chicote, modais e frota: a Logística 4.0 em programas que você cria com IA.

// apostila do aluno II · trilha de exercícios

01 Curva ABC



03 Nível de serviço



05 Efeito chicote



07 Dimensionar a frota



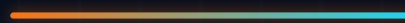
02 Lote econômico



04 Centralizar estoques



06 Modais de transporte



08 Plano de estoques



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 **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 **decisão logística** dos temas da disciplina: armazenagem e estoques, cadeia de suprimentos e transportes. Quase todos usam a Distribuidora Rota Certa e a Transportadora Estrada Boa, empresas fictícias.

1. Revisar

Checklist e lembretes da Apostila I

2. Armazenar

Curva ABC e lote econômico

3. Servir

Nível de serviço e centralização

4. Transportar

Efeito chicote e modais

5. Integrar

O plano de estoques

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
Classificar estoques	Faço a curva ABC e digo onde pôr mais controle.	☐
Decidir compras	Calculo lote econômico, estoque de segurança e ponto de pedido.	☐
Escolher o nível de serviço	Equilíbrio custo da falta e custo do excesso.	☐
Desenhar a rede	Comparo estoques centralizados e descentralizados.	☐
Ler a cadeia	Mostro o efeito chicote com números.	☐
Escolher o modal	Calculo custos e a distância de empate.	☐

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: **em logística, o ótimo de uma parte não é o ótimo do todo** (trade-off); olhe sempre o custo total.

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

- Na pasta Documentos, crie uma pasta chamada `Laboratorio_Codex_LOG`.
- 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_L06** 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.

3

Lembretes da Apostila I



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**.



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 · Armazenagem: a curva ABC

Num armazém, poucos itens concentram a maior parte do valor. A **curva ABC** mostra onde o gestor deve pôr mais atenção (e mais controle).



participação = valor anual do item ÷ valor total · ordene do maior para o menor e acumule

Classe A: até 80% do valor acumulado · Classe B: de 80% a 95% · Classe C: o restante

PROMPT 1

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

O usuário informa os itens do estoque com o nome e o valor consumido no ano (R\$).

O programa deve:

1. ordenar do maior para o menor valor;
2. calcular a participação (%) e a participação acumulada de cada item;
3. classificar cada item em A, B ou C;
4. mostrar quantos itens e quanto do valor há em cada classe.

Antes de criar o arquivo, apresente um plano curto.



O ESTOQUE DA DISTRIBUIDORA ROTA CERTA (EMPRESA FICTÍCIA), VALOR ANUAL EM R\$

Óleo 15W40: 90.000 · Pneu 295: 60.000 · Bateria: 25.000 · Filtro de ar: 10.000 · Lona de freio: 6.000 · Lâmpada: 4.000 · Palheta: 2.000 · Fusível: 1.500 · Abraçadeira: 1.000 · Etiqueta: 500.

Classe	Itens	Participação no valor
A		
B		
C		



PARA PENSAR

Que tipo de controle você faria nos itens A que não faria nos itens C?

5

Exercício 2 · Estoques: lote econômico e ponto de pedido

Quanto comprar de cada vez e quando comprar? O **lote econômico** equilibra o custo de pedir com o custo de manter, e o **ponto de pedido** protege contra a incerteza do lead time.



$$LEC = \sqrt{(2 \cdot D \cdot S \div H)} \quad \cdot \quad ES = z \cdot \sigma \cdot \sqrt{LT} \quad \cdot \quad PP = d \cdot LT + ES$$

D: demanda anual · S: custo por pedido · H: custo anual de manter 1 unidade · d: demanda diária · σ : desvio diário · LT: lead time (dias) · $z = 1,65$ para 95%

PROMPT 2

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

O usuário informa a demanda anual, o custo de cada pedido, o custo anual de manter uma unidade, o lead time (dias), o desvio padrão da demanda diária e o nível de serviço (z).

O programa deve calcular o lote econômico, o número de pedidos por ano, o custo total (pedir + manter), o estoque de segurança e o ponto de pedido (ano de 360 dias).

Mostre também o custo total para lotes 20% menores e 20% maiores, para comparar.

Antes de escrever o código, apresente um plano curto.

D = 12.000 · S = R\$ 150 · H = R\$ 3 · LT = 6 dias · $\sigma = 8$ · $z = 1,65$

Você espera

O programa mostrou

Lote econômico

Pedidos por ano

Custo total no lote econômico

Estoque de segurança

Ponto de pedido

6

Exercício 3 · Custo da falta x custo do excesso

Faltar produto custa a venda e a imagem; sobrar custa dinheiro parado. O **nível de serviço ótimo** equilibra os dois (slides do Tema 2).

★
$$\begin{aligned} \text{custo da falta} &= \text{preço} - \text{custo} & \text{custo do excesso} &= \text{custo} - \text{valor de recuperação} \\ \text{nível de serviço ótimo} &= \text{falta} \div (\text{falta} + \text{excesso}) & Q^* &= \text{média} + z \cdot \sigma \end{aligned}$$

O z vem da tabela normal para o nível de serviço ótimo (use `statistics.NormalDist().inv_cdf`)

PROMPT 3

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

Para um produto sazonal, o usuário informa o preço de venda, o custo, o valor pelo qual a sobra é vendida depois da estação e a demanda prevista (média e desvio padrão, distribuição normal).

O programa deve calcular o custo da falta, o custo do excesso, o nível de serviço ótimo, o z

correspondente e a quantidade a comprar. Use `statistics.NormalDist` do Python.

Antes de escrever o código, apresente um plano curto.

Preço R\$ 100 · custo R\$ 60 · sobra a R\$ 30 · demanda média 500, σ 100

Você espera

O programa mostrou

Custo da falta

Custo do excesso

Nível de serviço ótimo

Quantidade a comprar



PARA PENSAR

Se a sobra não valesse nada (R\$ 0), a empresa compraria mais ou menos? Teste no programa.

7

Exercício 4 · Centralizar ou descentralizar estoques?

Com vários centros de distribuição, cada um guarda o seu estoque de segurança. Centralizando, as variações se compensam: é a **variabilidade agregada** dos slides.



descentralizado: $ES = n \cdot z \cdot \sigma \cdot \sqrt{LT}$ • centralizado: $ES = z \cdot \sigma \cdot \sqrt{n \cdot LT}$

Vale para demandas iguais e independentes entre as regiões. Redução = $1 - 1 \div \sqrt{n}$

PROMPT 4

Crie um programa em Python chamado centralizar.py.

O usuário informa o número de centros de distribuição, o desvio padrão da demanda diária de cada um, o lead time e o z. O programa deve calcular o estoque de segurança total descentralizado e centralizado, a redução (%) e mostrar uma tabela para 1, 2, 4, 9 e 16 centros.

Explique, num comentário, por que o transporte fica mais caro quando se centraliza.

$\sigma = 50$ por dia · $LT = 4$ dias · $z = 1,65$	Descentralizado	Centralizado	Redução
4 centros			
9 centros			

8

Exercício 5 · Cadeia de suprimentos: o efeito chicote

Uma pequena variação no consumo vira uma grande variação nos pedidos lá atrás na cadeia. É o **efeito chicote** (bullwhip) dos slides do Tema 3.



regra de cada elo: pedido = demanda recebida + (demanda recebida - demanda anterior)

O elo "corrige" a variação com exagero. Amplitude = maior valor – menor valor da série

PROMPT 5

Crie um programa em Python chamado efeito_chicote.py.

Dada a demanda semanal do consumidor, o programa calcula os pedidos do varejo e, a partir deles, os pedidos do atacado, usando a regra: $\text{pedido} = \text{demanda} + (\text{demanda} - \text{demanda da semana anterior})$, nunca menor que zero (na primeira semana, $\text{pedido} = \text{demanda}$). Mostre as três séries numa tabela, a amplitude de cada uma e quantas vezes a amplitude do atacado é maior que a do consumidor.



DEMANDA DO CONSUMIDOR (8 SEMANAS)

100 · 100 · 100 · 110 · 100 · 100 · 100 · 100

Série	Pedidos semana a semana	Amplitude
Consumidor		
Varejo		
Atacado		



PARA PENSAR

Nos slides, o compartilhamento de dados em tempo real (Indústria 4.0) reduz o efeito chicote. Por quê?

9

Exercício 6 · Transportes: qual modal é mais barato?

O rodoviário é mais barato nas distâncias curtas; o ferroviário compensa nas longas, porque o custo de terminal e transbordo se dilui. Onde fica a virada?



custo = custo fixo da viagem + custo por km × distância

Rodoviário: fixo R\$ 300 e R\$ 5,00/km · Ferroviário: fixo R\$ 1.200 (terminais e transbordo) e R\$ 2,00/km

PROMPT 6

Crie um programa em Python chamado modais.py.

O usuário informa o custo fixo e o custo por km de dois modais e uma lista de distâncias.

O programa deve calcular o custo de cada modal em cada distância, indicar o mais barato

e calcular a distância de empate (onde os dois custam o mesmo).

Antes de escrever o código, apresente um plano curto.

Distância	Rodoviário	Ferroviário	Mais barato
150 km			
300 km			
600 km			
1.000 km			

Distância de empate: _____ km. Confere com os slides do Tema 4?

10 Desafio extra · Dimensionar a frota

Frota pelo pico fica parada na maior parte do tempo; frota pequena demais paga muito frete terceirizado. A **frota mista** busca o equilíbrio.

PROMPT 7

desafio

Crie um programa em Python chamado frota.py.

Uma transportadora tem a demanda de viagens de 10 dias: 8, 10, 12, 9, 20, 11, 10, 9, 13, 8.

Cada caminhão próprio faz 2 viagens por dia e custa R\$ 600 por dia (parado ou não).

A viagem que a frota própria não cobre vai para um terceiro, a R\$ 450.

Calcule o custo total dos 10 dias para frotas de 3 a 10 caminhões e indique a melhor.

Compare com a frota dimensionada pelo pico. Explique o resultado.

Frota	Custo próprio	Custo de terceiros	Custo total
4 caminhões			
5 caminhões			
6 caminhões			
10 (pelo pico)			

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 Logística que conhece o conteúdo, mas nunca programou em Python. Não reescreva o código.

REVISAR O CONTEÚDO

Revise meu programa como um professor de Logística.

Não faça alterações. Confira se os conceitos e as contas estão corretos

(as fórmulas (LEC, estoque de segurança, nível de serviço), as unidades (dias, anos, km) e as hipóteses de cada modelo) 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.

12

Atividade final · O plano de estoques do armazém

Junte os exercícios num programa que o gestor do armazém usaria toda semana: ele lê os itens, faz a **curva ABC**, calcula o **lote econômico**, o **estoque de segurança** e o **ponto de pedido**, e diz o que comprar agora.

Requisitos

- ☐ Ler itens.csv (item, valor_anual, demanda_anual, custo_pedido, custo_manter, lead_time, desvio_diario, estoque_atual)
- ☐ Classificar em A, B e C
- ☐ Usar $z = 1,65$ para itens A, $1,28$ para B e $1,04$ para C
- ☐ Calcular lote econômico, estoque de segurança e ponto de pedido de cada item
- ☐ Listar os itens com estoque atual abaixo do ponto de pedido: "comprar agora", com a quantidade (o lote)
- ☐ Escrever um resumo para o gestor, com os números



POR QUE Z DIFERENTE?

Itens A valem muito: faltar custa caro, então se protege mais. Itens C valem pouco: um nível de serviço menor libera espaço e dinheiro. É a estratégia de estoques dos slides.

PROMPT A

planejamento

Quero criar em Python um "plano de estoques" para um armazém. Ele lê itens.csv, classifica os itens na curva ABC, calcula o lote econômico, o estoque de segurança (z por classe: A $1,65$, B $1,28$, C $1,04$) e o ponto de pedido, e lista o que comprar agora. Antes de escrever código, proponha as funções (reaproveitando os exercícios 1 e 2), um itens.csv de exemplo com os 10 itens do Exercício 1 e os testes. Não escreva o código ainda.

PROMPT B

implementação

Agora implemente em plano_estoques.py e termine com o resumo escrito pelo programa.

Com os seus dados

Resultado

Itens A / B / C

Itens para comprar agora

Estoque de segurança total (unidades)

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)	☐

Reflexão individual

1. Qual decisão logística ficou mais clara depois que você a programou?

2. Em que momento a IA errou uma fórmula ou uma unidade (dia × ano), e como você percebeu?

3. Por que centralizar estoques reduz o estoque de segurança, mas encarece o transporte?

4. Que dado real faltou ao seu plano de estoques?

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 logístico

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

As contas desta apostila

Curva ABC	80% · 95% · 100%
Lote econômico	$\sqrt{(2DS/H)}$
Estoque de segurança	$z \cdot \sigma \cdot \sqrt{LT}$
Ponto de pedido	$d \cdot LT + ES$
Nível de serviço ótimo	$falta / (falta + excesso)$
Modais	$fixo + km \times custo/km$

Comandos

TERMINAL

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

Glossário

Curva ABC

Classificação dos itens pelo valor que representam.

Lote econômico

Quantidade por pedido que minimiza o custo total.

Lead time

Tempo entre fazer o pedido e receber.

Estoque de segurança

Proteção contra a variação da demanda e do lead time.

Nível de serviço

Chance de atender a demanda sem faltar.

Efeito chicote

Amplificação das variações ao subir na cadeia.

Modal

Tipo de transporte: rodoviário, ferroviário, aquaviário, aéreo, dutoviário.

Frota mista

Frota própria para a base da demanda e terceiros para os picos.

1

LEMBRE-SE

A IA é uma ferramenta. Conhecimento, julgamento, validação e responsabilidade continuam sendo de quem programa, e de quem decide a logística da empresa.