

// laboratório codex 03 · marketing digital

python + ia + app

>_ Laboratório Codex 03

Painel de marketing em Python, com IA

Envie a planilha das campanhas e veja o painel nascer: funil, métricas, gráficos por canal, teste A/B e o relatório do mês em Excel, num app feito por você.

// apostila do aluno III · o projeto em etapas

01 Dados das campanhas



03 Funil e canais



05 Ferramentas de decisão



07 Previsão de vendas



02 Métricas



04 Upload e filtros



06 Relatório do mês



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 **painel de marketing digital** que recebe a planilha exportada das campanhas, calcula as métricas, mostra gráficos com filtros e gera o relatório do mês.

1. Dados

O CSV das campanhas

2. Métricas

Funções das métricas

3. Gráficos

Funil e canais

4. App

Upload e filtros

5. Relatório

Ferramentas e Excel



O PROJETO: PAINEL DE MARKETING DIGITAL

O usuário envia o CSV das campanhas. O app calcula CTR, CPC, CPL, CPA, ROAS e ROI, mostra o funil, compara canais, tem uma calculadora de teste A/B e de LTV, sugere a divisão da verba e gera o relatório mensal em Excel.

Ao final, você deverá conseguir

Habilidade	Como saber que consegui	Ok
Preparar dados	Monto e limpo o CSV das campanhas.	☐
Calcular métricas	Funções de CTR a ROI, sem divisão por zero.	☐
Fazer gráficos de marketing	Mostro funil, ROI por canal e evolução.	☐
Construir um painel	Upload, filtros e indicadores no topo.	☐
Decidir com dados	Teste A/B, LTV e sugestão de verba no app.	☐
Gerar o relatório	Excel do mês, pronto para o gestor.	☐

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_MKT3** em Documentos e abra no VS Code. Ao longo das etapas, ela vai ficar assim:

TERMINAL

```
Laboratorio_Codex_MKT3/  
├─ dados/           ← arquivos de dados (CSV)  
├─ calculos.py      ← as contas (funções)  
├─ graficos.py      ← os gráficos  
├─ app.py           ← a tela (Streamlit)  
└─ relatorio.xlsx   ← gerado pelo app
```



GUARDE VERSÕES

Ao terminar cada etapa com o app funcionando, copie a pasta inteira com o nome da etapa (por exemplo, `Laboratorio_Codex_MKT3_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 das campanhas

Toda análise começa por dados limpos. Você vai montar o CSV das campanhas de três meses da Loja Bem-Estar.

PROMPT 1

Crie dados/campanhas.csv com as colunas: mes, campanha, canal, gasto, impressoes, cliques, leads, vendas, receita. Coloque 3 meses (julho a setembro de 2026), 3 canais (Google Ads, Meta Ads, Influenciador) e 2 campanhas por canal por mês (18 linhas), com números realistas e coerentes (cliques menores que impressões, vendas menores que leads). Em calculos.py, crie ler_campanhas(arquivo) que leia o CSV com vírgula ou ponto e vírgula e confira se as colunas existem.



CHECKPOINT

O CSV tem 18 linhas e a função o lê sem erro, nos dois formatos de separador.

5 Etapa 2 · As métricas

As fórmulas da Apostila II, agora aplicadas a tabelas inteiras, de uma vez.

PROMPT 2

Em `calculos.py`, crie `calcular_metricas(df, margem)` que acrescente ao DataFrame as colunas CTR, CPC, CPL, CPA, ROAS e ROI, sem dividir por zero (use vazio quando não der para calcular). Crie também `resumo_por(df, coluna)`, que agrupe por canal ou por mês somando gasto, cliques, leads, vendas e receita e recalculando as métricas (atenção: não é a média das métricas!).



A ARMADILHA DA MÉDIA

O CPA de um canal é o gasto total ÷ as vendas totais, e **não** a média dos CPAs das campanhas. Teste com duas campanhas bem diferentes e veja se o Codex acertou.

Teste com a campanha da Apostila II	Esperado	Calculado
CTR (200.000 impressões, 4.000 cliques)	2,00%	
CPA (gasto 8.000, 40 vendas)	R\$ 200,00	
ROI (receita 15.000, gasto 5.000, margem 40%)	20%	

Etapa 3 · Os gráficos

Três gráficos que todo gestor de marketing pede.

PROMPT 3

Crie graficos.py com:

1. grafico_funil: barras horizontais decrescentes (impressões, cliques, leads, vendas) com a taxa de conversão de cada etapa escrita ao lado;
2. grafico_roi_canal: barras do ROI por canal, verdes para positivo e vermelhas para negativo, com uma linha no zero;
3. grafico_evolucao: linhas de gasto e receita por mês.



CHECKPOINT

O funil mostra as taxas de conversão e o ROI negativo aparece em vermelho.

7

Etapa 4 • O painel com upload e filtros

O gestor envia a planilha do mês e o painel aparece.

PROMPT 4

Crie app.py com Streamlit:

- st.file_uploader para enviar o CSV (e um botão para usar o arquivo de exemplo);
- barra lateral: margem de contribuição e filtros de mês e de canal (st.multiselect);
- no topo: st.metric com gasto, receita, vendas, CPA e ROI do período filtrado;
- abaixo: os três gráficos e a tabela das campanhas, ordenada pelo ROI.

TERMINAL

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



CHECKPOINT

Filtrando só o canal Influenciador, todos os números e gráficos mudam.

O painel vira uma caixa de ferramentas: as calculadoras da Apostila II, cada uma numa aba.

PROMPT 5

Acrescente abas ao app:

1. "Teste A/B": visitantes e conversões de A e B; mostra conversões, lift, valor z e se é significativo, com uma frase explicando;
2. "LTV": ticket, frequência, anos e margem; mostra LTV, LTV/CAC (usando o CPA do painel) e o payback;
3. "Verba": com o CPA de cada canal (vindo dos dados) e um limite de vendas, sugere a divisão de uma verba informada (desafio da Apostila II).

Teste na aba A/B	z	Significativo?
5.000/150 × 5.000/190		
500/15 × 500/19		

9 Etapa 6 • O relatório do mês

Todo início de mês, o relatório das campanhas. O app gera em um clique.

PROMPT 6

Acrescente um botão "Baixar relatório do mês" que gere um Excel com as abas: "Resumo" (KPIs do período e três destaques escritos pelo app: melhor campanha, pior campanha e canal com menor CPA), "Por canal", "Por mês" e "Campanhas". Formate percentuais e reais.



CHECKPOINT

O relatório tem 4 abas e os destaques batem com o painel.



LGPD

Se um dia usar dados reais, o relatório não pode ter nomes, e-mails ou telefones de clientes: só números agregados.

10 Desafio extra · Previsão de vendas

Com três meses de dados, dá para arriscar uma previsão: se eu gastar mais, quanto vendo?

PROMPT 7

desafio

Acrescente uma aba "Previsão": para cada canal, ajuste uma reta simples de vendas em função do gasto (mínimos quadrados, sem bibliotecas extras ou com numpy, que já vem com o pandas). Mostre o gráfico de dispersão com a reta e uma previsão para um gasto informado. Explique as limitações (poucos dados, saturação do canal) 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 Marketing Digital. Não faça alterações. Confira as fórmulas das métricas, o CPM por mil, o ROI com a margem e o teste A/B 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 funil do mês e qual canal merece mais verba.
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 painel de marketing digital") 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 Marketing Digital 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 · 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.

Upload

Enviar um arquivo para o app.

Filtro

Escolha que limita os dados mostrados (canal, mês).

KPI

Indicador-chave de desempenho.

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.