

// laboratório codex 01 · programação com ia

python + ia

>_ Laboratório Codex 01

Programação assistida por IA VS Code + ChatGPT + Codex

Instale as ferramentas, converse com um agente de programação e aprenda a criar, executar, testar, explicar e corrigir programas em Python.

// apostila do aluno I · trilha do laboratório

01 Instalar o VS Code e o Python

03 Criar o primeiro projeto

05 Provocar um erro de propósito

07 Exercício 3 · Valor futuro

02 Instalar e entrar no Codex

04 Exercício 1 · Nome e idade

06 Exercício 2 · Média de notas

08 Mini sistema financeiro



Prof. Caio Haffner

Mestre em Desenvolvimento Econômico · Unicamp



Leia antes de tudo.

Esta apostila funciona como uma receita: **cada passo prepara o seguinte**. Quem pula direto para a instalação quase sempre instala errado, não entende o que está fazendo e perde a aula.

- 1 Leia desde a primeira letra da primeira palavra**, na ordem, sem pular nada. Inclusive esta página.
- 2 Faça tudo o que a apostila manda, na hora em que ela manda**. Se ela diz "abra", abra. Se diz "copie", copie. Se diz "marque o checkpoint", marque.
- 3 Não vá direto para o VS Code e o Python**. As seções 1, 2 e 3 explicam o que você vai instalar, por que e como trabalhar com a IA.
- 4 Só avance quando o checkpoint estiver certo**. Se não estiver, pare e peça ajuda ao professor ou a um colega.
- 5 Não entendeu um parágrafo?** Releia. Se continuar sem entender, pergunte antes de seguir.

x PULAR CUSTA CARO

Pular estas páginas economiza 5 minutos e custa a aula inteira: a maior parte dos problemas de instalação acontece com quem não leu o começo.

PALAVRA-CHAVE

Quem leu esta página sabe a palavra-chave da Apostila I: **FAROL**. Escreva **FAROL** na primeira linha do seu **diário de prompts** (seção 16). O professor vai conferir.

☐ Li esta página inteira e vou seguir a apostila **na ordem, fazendo tudo o que ela manda** .

1 Como usar esta apostila

Você vai trabalhar com um **editor de código** (VS Code) e com um **agente de programação** (Codex). A meta não é só gerar código: é entender o problema, conversar com a IA, analisar a solução, executar, testar e corrigir.

1. Instalar VS Code, Python e Codex	2. Criar O primeiro programa com a IA	3. Depurar Provocar um erro e entender a causa	4. Aplicar Notas e valor futuro (juros compostos)	5. Entregar Mini sistema financeiro e reflexão
---	---	--	---	--

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. Marque o quadrado e siga. Se não estiver vendo, pare e peça ajuda.
Caixa azul · DICA	Atalho ou jeito mais fácil de fazer.
Caixa laranja · ATENÇÃO	Ponto em que muita gente erra.
Caixa vermelha · NÃO FAÇA	Algo que pode causar problema de segurança.

Ao final, você deverá conseguir

Habilidade	Como saber que conseguiu	Ok
Instalar o VS Code	Abro o editor e crio uma pasta de projeto.	☐
Instalar o Codex	Encontro a extensão oficial e abro o painel do Codex.	☐
Conectar o ChatGPT	Entro com a minha conta ChatGPT pelo Codex.	☐
Criar um programa	Peço ao Codex para criar ou alterar arquivos.	☐
Executar	Rodo o programa no terminal integrado.	☐
Depurar	Mostro um erro ao Codex e peço o diagnóstico antes da correção.	☐
Explicar	Explico o que o código faz sem depender da IA.	☐

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.**

2 O que é o Codex

O **Codex** é o agente de programação da OpenAI. Ele lê os arquivos do seu projeto, escreve e altera código, executa comandos e explica o que fez. No VS Code, ele aparece como um painel dentro do editor.

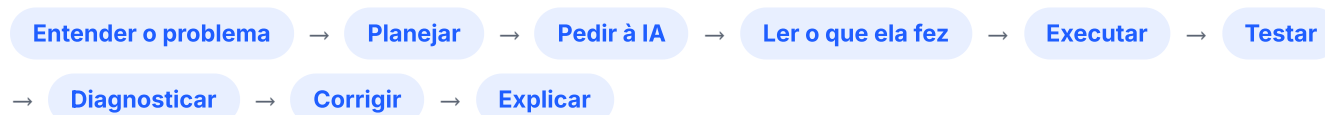
Segundo a documentação da OpenAI, o Codex está disponível nos planos do ChatGPT, inclusive o gratuito, com **limites de uso** que variam conforme o plano. Recursos como o Codex na nuvem podem ficar de fora dos planos gratuitos.

! TELAS MUDAM

Nomes de botões, limites e telas mudam com frequência. A apostila ensina o **jeito de trabalhar**. Se uma tela estiver diferente, siga o que a própria extensão mostra e confira a documentação oficial (seção 18).

Como vamos trabalhar com a IA

Em todos os exercícios, siga este ciclo. Ele vale para qualquer ferramenta de IA, hoje e no futuro:



★ A IA É BOA EM

Escrever rapidamente um primeiro rascunho, explicar mensagens de erro, sugerir testes e revisar código.

! A IA ERRA EM

Entender o que você quis dizer sem que você diga, conferir contas, e ela às vezes muda mais do que foi pedido. Por isso você sempre confere.

3

O que você precisa

Item	Necessário?	Observação
Computador Windows, macOS ou Linux	Sim	Os passos foram escritos pensando no Windows.
Conta ChatGPT	Sim	A conta gratuita funciona, dentro dos limites do plano.
Internet	Sim	Para instalar e para entrar com a conta.
VS Code	Sim	Editor de código gratuito (seção 4).
Python	Sim	Linguagem dos exercícios (seção 5).

Checklist antes da aula

- ☐ Sei o e-mail e a senha da minha conta ChatGPT.
- ☐ Tenho permissão para instalar programas neste computador.
- ☐ Sei onde vou guardar meus projetos (por exemplo, em Documentos).
- ☐ O computador está conectado à internet.



NÃO FAÇA

Nunca compartilhe a senha do ChatGPT, tokens, chaves de API ou credenciais com colegas, nem cole esses dados em arquivos do projeto. Nesta atividade **não é preciso nenhuma chave de API**.

4

Instalar o Visual Studio Code



CHEGOU AQUI PULANDO?

Volte para a página **"PARE. Leia antes de tudo"**, logo depois da capa, e comece pela seção 1. Quem pula as seções 1 a 3 costuma errar a instalação.

- 1 Acesse code.visualstudio.com e baixe a versão para o seu sistema. No Windows, escolha o instalador **User Setup**: ele instala só para o seu usuário e não pede permissão de administrador.
- 2 Execute o arquivo baixado e aceite os termos.
- 3 Na tela de opções, deixe marcada **"Adicionar a PATH"** (já vem marcada). Se quiser, marque também as opções de **"Abrir com Code"** no menu do Explorador de Arquivos.
- 4 Conclua a instalação e abra o VS Code.



CHECKPOINT

O VS Code abriu e mostra uma tela de boas-vindas.

Conheça a tela

Área	Para que serve	Atalho
Explorer	Mostra as pastas e os arquivos do projeto.	Ctrl + Shift + E
Editor	Onde você lê e altera o código.	—
Terminal	Onde você executa comandos e roda o programa.	Ctrl + '
Extensões	Onde você instala recursos extras, como o Codex.	Ctrl + Shift + X
Paleta de comandos	Procura qualquer comando do VS Code pelo nome.	Ctrl + Shift + P
Codex	Painel para conversar com o agente de programação.	ícone na barra lateral



DICA

Não precisa decorar tudo agora. Você vai usar cada área ao longo da aula. No teclado brasileiro, o atalho do terminal costuma ser **Ctrl** + **'** (aspas simples); se não funcionar, use o menu **Terminal** → **Novo Terminal**.

5

Instalar o Python

O VS Code é o **editor**; o Python é a **linguagem** que executa os programas. Os dois são instalados separadamente.

5.1 Verifique se o Python já está instalado

No VS Code, abra o terminal (**Terminal** → **Novo Terminal**) e digite:

TERMINAL

```
python --version
```

Se aparecer algo como **Python 3.12.x**, pule para a seção 5.3.

5.2 Se não estiver instalado

- 1 Baixe o instalador em python.org/downloads.
- 2 Na primeira tela do instalador, marque a caixa "Add python.exe to PATH". Ela fica embaixo, e é fácil passar sem ver.

- 3 Clique em **Install Now** e aguarde.
- 4 **Feche e abra de novo o VS Code**, para o terminal reconhecer o Python.



O ERRO MAIS COMUM DA AULA

Se o terminal disser que `python` não é reconhecido, ou abrir a Microsoft Store, quase sempre a caixa **"Add python.exe to PATH"** ficou desmarcada. Tente o comando `py --version`, que costuma funcionar no Windows. Se funcionar, use `py` no lugar de `python` em toda a apostila.

5.3 Teste

TERMINAL

```
python --version
python -c "print('Python funcionando!')"
```

5.4 Instale a extensão Python no VS Code

Em **Extensões** (`Ctrl` + `Shift` + `X`), procure **Python** e instale a extensão da **Microsoft**. Ela colore o código, aponta erros de digitação e adiciona um botão ► para executar.



CHECKPOINT

O terminal mostrou "Python funcionando!" e a extensão Python aparece como instalada.



SE DER ERRO

Não tente adivinhar. Copie a mensagem do terminal e, na seção 6, peça ao Codex para **explicar** o erro antes de mudar qualquer coisa.

6

Instalar e entrar no Codex

6.1 Instale a extensão oficial

- 1 Abra as **Extensões** (`Ctrl` + `Shift` + `X`) e pesquise **Codex**.
- 2 Confira se o publicador é a **OpenAI**. Existem extensões com nomes parecidos que não são oficiais.
- 3 Clique em **Install**. Um ícone do Codex aparece na barra lateral.

6.2 Entre com a sua conta ChatGPT

- 1 Clique no ícone do Codex para abrir o painel.

- 2 Escolha a opção de entrar com a conta **ChatGPT**. O navegador abre a página de login.
- 3 Entre com a mesma conta que você usa no ChatGPT e autorize o acesso.
- 4 Volte ao VS Code e espere o painel do Codex ficar pronto para receber mensagens.



NÃO FAÇA

A senha do ChatGPT é digitada **somente na página de login do navegador**. Nunca a cole no terminal, no código, num arquivo do projeto ou no chat do Codex.



CHECKPOINT

O painel do Codex está aberto e aceita uma mensagem. Teste escrevendo: *"Olá! Em uma frase, o que você consegue fazer neste projeto?"*



SE A TELA ESTIVER DIFERENTE

O VS Code e o Codex recebem atualizações com frequência. Siga o fluxo de login que a própria extensão mostrar e, na dúvida, consulte a documentação oficial (seção 18).

7

Criar o primeiro projeto

- 1 Na pasta Documentos, crie uma pasta chamada `Laboratorio_Codex_01`.
- 2 No VS Code, use **Arquivo** → **Abrir Pasta** e escolha essa pasta. Se aparecer a pergunta "Você confia nos autores desta pasta?", responda **Sim**: a pasta é sua.
- 3 Confira que o nome da pasta aparece no Explorer.



CONCEITO

A **pasta** é o projeto. Os **arquivos .py** ficam dentro dela. O **terminal** executa os programas. O Codex trabalha **dentro da pasta que você abriu**, por isso nunca abra a pasta inteira de Documentos ou do computador.



CHECKPOINT

O Explorer mostra a pasta `Laboratorio_Codex_01`, ainda vazia.

8

Exercício 1 • Nome e idade

Problema: queremos um programa que peça o nome e a idade de uma pessoa e mostre uma mensagem com essas informações.

PROMPT 1

copie e cole no Codex

Crie um programa em Python chamado main.py.

O programa deve:

1. pedir o nome do usuário;
2. pedir a idade;
3. mostrar uma saudação;
4. mostrar a idade informada.

Antes de alterar o arquivo, explique brevemente o que pretende fazer.

Depois crie o arquivo.

Não adicione funcionalidades que não foram pedidas.

**LEIA ANTES DE APROVAR**

O Codex pode pedir autorização para criar o arquivo ou executar um comando. Leia o que ele quer fazer e só aprove se fizer sentido. Mais sobre isso na seção 13.

O que observar na resposta

- A IA entendeu o pedido? Criou o arquivo com o nome certo?
- O código é simples o bastante para você explicar?
- Tem alguma palavra ou comando que você não conhece? Anote para perguntar.

Um código possível

```
nome = input("Digite seu nome: ")
idade = int(input("Digite sua idade: "))

print(f"Olá, {nome}!")
print(f"Você tem {idade} anos.")
```

**ENTRADA • PROCESSAMENTO • SAÍDA**

`input()` recebe dados do usuário (entrada). `int()` transforma o texto digitado em número (processamento). `print()` mostra o resultado (saída). Quase todo programa segue esse caminho.

Execute

Abra o terminal (`Ctrl` + ```) e digite:

TERMINAL

```
python main.py
```

Teste e registre

#	Entrada	Resultado obtido	Funcionou?
1	João / 20		☐ Sim
2	Maria / 35		☐ Sim
3	Pedro / 18		☐ Sim



CHECKPOINT

Os três testes mostraram a saudação e a idade corretas.

9

Provocando um erro de propósito

Errar faz parte da programação. A habilidade que você vai treinar agora se chama **depuração**: descobrir a causa de um erro antes de corrigi-lo.

9.1 Provoque o erro

Execute o programa de novo e, quando ele pedir a idade, digite a palavra `vinte`. O programa vai parar com uma mensagem de erro em inglês. **Não se assuste: leia a última linha da mensagem.**



COMO LER UMA MENSAGEM DE ERRO

A **última linha** diz o tipo do erro (por exemplo, `ValueError`) e o motivo. As linhas de cima mostram **onde** ele aconteceu: o arquivo e o número da linha.

9.2 Peça o diagnóstico, não a correção

PROMPT 2 · DIAGNÓSTICO

não deixe a IA alterar nada ainda

Analise o erro que ocorreu no programa.

Não altere o código ainda.

Explique:

1. qual linha causou o problema;
2. qual é a causa do erro;
3. por que o Python não conseguiu executar essa linha;
4. qual seria uma estratégia para corrigir.

Anote, com suas palavras, a causa do erro:

9.3 Agora peça a correção

PROMPT 3 · CORREÇÃO

Agora corrija o programa para que uma idade inválida não encerre o programa com um erro não tratado.

Faça a menor alteração necessária e explique o que mudou.

Teste de novo com `vinte` e depois com `20`. Os dois casos devem funcionar sem o programa travar.



MÉTODO DE DEPURAÇÃO

Diagnóstico → explicação → solução → alteração → teste. Pedir a correção direto funciona às vezes, mas você não aprende nada e a IA pode mudar coisas que não precisavam mudar.



CHECKPOINT

Com "vinte", o programa avisa que a idade é inválida em vez de travar. Com "20", continua funcionando.



PROMPT FRACO

"Faça um sistema de notas."

Que notas? Quantas? Qual a regra de aprovação? A IA vai adivinhar, e cada um da turma vai receber uma coisa diferente.



PROMPT MELHOR

Diz o contexto, os dados, as regras, a saída esperada e como a IA deve trabalhar (planejar antes, não inventar funcionalidades).

A estrutura de um bom prompt

Elemento	Pergunta que você responde	Exemplo
Contexto	O que estou construindo, e para quem?	Programa para um professor
Objetivo	O que o programa precisa fazer?	Calcular a média de um aluno
Dados	Quais entradas o programa recebe?	Três notas de 0 a 10
Regras	Quais fórmulas ou condições existem?	Aprovado se média $\geq 6,0$
Saída	O que deve aparecer na tela?	Média com 2 casas e a situação
Restrições	O que a IA não deve fazer?	Não criar telas gráficas
Validação	Como saberemos que funciona?	Testes com notas conhecidas

Exercício 2 · Média de notas

PROMPT 4

Crie um programa em Python para um professor calcular a média de três notas de um aluno.

Requisitos:

- receber três notas;
- calcular a média aritmética;
- informar a média com duas casas decimais;
- informar "Aprovado" se a média for $\geq 6,0$;
- informar "Reprovado" se a média for $< 6,0$.

Antes de escrever o código, apresente um plano curto.

Depois implemente a solução em um arquivo chamado notas.py.

Notas	Média que você espera	Situação esperada	O programa mostrou
7, 8, 9			
5, 6, 7			
5,5 · 6 · 6 ← digite com vírgula			



A VÍRGULA BRASILEIRA

No Python, o separador decimal é o **ponto**: `5.5`. Se você digitar `5,5`, o programa provavelmente vai dar erro. Use o método da seção 9: peça o diagnóstico e depois peça para o programa **aceitar vírgula ou ponto**.

11 Exercício 3 · Valor futuro

Vamos transformar uma fórmula de matemática financeira em programa: o valor futuro de uma aplicação com juros compostos.



$$VF = VP \times (1 + i)^n$$

VP = valor presente · i = taxa por período (em decimal) · n = número de períodos · VF = valor futuro

PROMPT 5

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

O programa deve:

1. solicitar o valor presente (VP);
2. solicitar a taxa de juros por período, em percentual;
3. solicitar o número de períodos;
4. converter a taxa percentual para decimal;
5. calcular $VF = VP \times (1 + i) ** n$;
6. mostrar o valor futuro com duas casas decimais;
7. mostrar os juros acumulados ($VF - VP$).

Use nomes de variáveis claros.

Valide entradas inválidas de forma simples.

Antes de implementar, explique a lógica.

Rode o programa e complete a tela

No terminal, execute o programa e digite os valores destacados. Depois, escreva nas linhas o que o programa mostrou.

TERMINAL

```
python valor_futuro.py
Valor presente: 10000
Taxa ao mês (%): 1
Períodos: 12
Valor futuro: R$ _____
Juros acumulados: R$ _____
```

Confira o programa com a sua calculadora

Calcule primeiro **na calculadora (ou na HP 12C)** e só depois rode o programa. Se os resultados forem diferentes, alguém errou: descubra quem.

VP	Taxa	Períodos	VF pela calculadora	VF pelo programa	Juros
10.000,00	1% ao mês	12			
5.000,00	0,5% ao mês	24			
1.000,00	0%	10			



ARMADILHA: "10.000,00"

Digite o valor presente do jeito brasileiro, **10.000,00**, e veja o que acontece. O Python lê o ponto como separador decimal, então **10.000** pode virar **dez**, e não dez mil. Peça ao Codex para tratar o formato brasileiro e teste de novo.



REFLEXÃO

A fórmula financeira virou um algoritmo. O código é só outra forma de escrever um raciocínio que você já conhece.

12

Use o Codex como professor, não como copiador

A IA não serve só para escrever código. Ela pode explicar, revisar e criar testes. Use estes quatro prompts no programa de valor futuro.

PROMPT 6 · EXPLICAR

Explique este código linha por linha para um estudante de Administração que conhece a fórmula financeira, mas nunca programou em Python.
Não reescreva o código; apenas explique.

PROMPT 7 · REVISAR

Revise meu programa como um professor de programação.
Não faça alterações ainda.
Liste: erros; riscos de entrada inválida; pontos de clareza; melhorias opcionais.
Separe problemas obrigatórios de melhorias opcionais.

PROMPT 8 · TESTAR

Crie uma tabela de casos de teste para este programa.
Inclua: um caso normal; um caso com zero; um caso com valor decimal;
um caso de entrada inválida; e o resultado esperado de cada caso.

PROMPT 9 · MELHORAR SÓ UMA COISA

Implemente somente a validação de entrada.
Não altere a fórmula nem a interface do programa.
Depois mostre um resumo das mudanças realizadas.



CONFIRA A TABELA DE TESTES

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

13 Segurança e permissões

Agentes de programação podem **ler** arquivos, **editar** arquivos e **executar comandos** no seu computador, conforme as permissões. O Codex trabalha em um ambiente protegido (sandbox) e pede autorização antes de ações mais arriscadas. A decisão final é sua.

Antes de clicar em "Aprovar", pergunte-se

- ☐ Eu sei o que este comando ou esta alteração faz?
- ☐ Ele mexe só nos arquivos da pasta do projeto?
- ☐ Ele apaga, instala ou baixa alguma coisa? Se sim, eu pedi isso?
- ☐ Se der errado, consigo desfazer?

Boas práticas

- Trabalhe sempre numa pasta criada só para a atividade.
- Não coloque senhas nem chaves secretas dentro do projeto.
- Não dê acesso a pastas pessoais (fotos, documentos, downloads).
- Guarde uma cópia do projeto quando ele estiver funcionando.



REGRA DE SEGURANÇA

Se você não sabe o que um comando vai fazer, **não o execute só porque a IA recomendou**. Pare e pergunte: à IA ("o que este comando faz, exatamente?") e ao professor.

14 Problemas comuns

Problema	Primeira ação
O Codex não aparece	Confira se a extensão é a oficial da OpenAI e reinicie o VS Code.
Não consigo entrar na conta	Confira a conta ChatGPT, o navegador e a internet. Tente sair e entrar de novo.
<code>python</code> não é reconhecido	Tente <code>py</code> . Se não der, reinstale o Python marcando "Add python.exe to PATH" e reabra o VS Code.
O programa dá erro	Leia a última linha da mensagem, copie o erro e peça primeiro o diagnóstico.
O Codex muda coisas demais	Diga: "faça somente a alteração X e não modifique os demais arquivos".
Limite de uso atingido	Continue testando e analisando sem a IA. Os limites variam por plano e se renovam com o tempo.

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.

Com a ajuda do Codex, crie um pequeno programa para uma microempresa registrar o mês.

Requisitos

- ☐ Cadastrar receitas do mês
- ☐ Cadastrar despesas do mês
- ☐ Calcular o total de receitas
- ☐ Calcular o total de despesas
- ☐ Resultado = receitas – despesas
- ☐ Informar superávit ou déficit
- ☐ Aceitar pelo menos três lançamentos
- ☐ Mostrar valores com duas casas decimais

Teste com estes dados

Lançamento	Tipo	Valor (R\$)
Vendas	Receita	8.000,00
Serviços	Receita	2.500,00
Aluguel	Despesa	3.000,00
Salários	Despesa	5.200,00
Energia	Despesa	650,00

Resultado esperado (calcule você): **R\$** _____ .
superávit ou déficit? _____

PROMPT A • PLANEJAMENTO

não deixe escrever código ainda

Quero desenvolver um mini sistema financeiro em Python para uma microempresa.

Antes de escrever código:

1. transforme os requisitos em uma lista de funcionalidades;
2. proponha uma estrutura simples de arquivos;
3. sugira como representar receitas e despesas;
4. apresente casos de teste.

Não escreva o código ainda.

PROMPT B • IMPLEMENTAÇÃO

Agora implemente a versão mínima do projeto.

Crie os arquivos necessários.

Mantenha a solução simples para um estudante iniciante.

Inclua comentários apenas onde forem realmente úteis.

Depois explique como executar o projeto.

PROMPT C • TESTES

Execute e revise o projeto e verifique:

- soma das receitas; soma das despesas; resultado;
- cenário de superávit; cenário de déficit; entradas inválidas.

Corrija somente problemas comprovados e explique as mudanças.

Diário de prompts

Palavra-chave da página "PARE": _____

Registre pelo menos dois prompts que você usou e o que aconteceu.

#	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	☐
Print ou registro de pelo menos um teste	☐
Explicação do funcionamento, com suas palavras	☐
Um erro encontrado e como foi corrigido	☐
O diário de prompts (acima)	☐
A reflexão final (abaixo)	☐
Registro da auditoria humana (opcional, vale ponto extra)	☐

Reflexão individual

1. O que o Codex fez bem no seu projeto?

2. Em que momento você precisou tomar uma decisão que a IA não poderia tomar por você?

3. Qual erro apareceu e o que você aprendeu com ele?

4. Você consegue explicar o código principal sem consultar a IA? Explique uma parte.

5. Que melhoria você faria no sistema se tivesse mais tempo?

17

Como você será avaliado

Critério	0	1	2	3
Instalação	Não realizou	Com muita ajuda	Com pouca ajuda	Sozinho
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 de erro
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 as decisões

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

Atalhos

Ctrl + Shift + X	Extensões
Ctrl + Shift + P	Paleta de comandos
Ctrl + `	Terminal
Ctrl + S	Salvar arquivo
Ctrl + Z	Desfazer

Comandos

TERMINAL

```
python --version      # ou: py --version
python main.py        # executa o
programa
Ctrl + C              # interrompe um
programa travado
```

Glossário

Prompt

O pedido que você escreve para a IA.

Terminal

Janela onde você digita comandos para o computador.

Extensão

Recurso extra instalado no VS Code (como o Codex).

Depurar

Encontrar e corrigir a causa de um erro (debug).

PATH

Lista de lugares onde o Windows procura programas como o Python.

Sandbox

Ambiente protegido que limita o que o agente pode fazer.

ValueError

Erro de valor: o dado tem o tipo certo, mas um conteúdo que não serve (como "vinte" para int).

1

LEMBRE-SE

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

Referências oficiais

- OpenAI. Usando o Codex com seu plano ChatGPT: help.openai.com/pt-br/articles/11369540
- OpenAI Developers. Codex: developers.openai.com/learn/codex
- OpenAI. Sandbox: developers.openai.com/pt-BR/docs/sandboxing
- Visual Studio Code. Instalação no Windows: code.visualstudio.com/docs/setup/windows
- Visual Studio Code. Primeiros passos: code.visualstudio.com/docs/getstarted/overview
- Python. Downloads: python.org/downloads

Auditoria humana · ajude a melhorar esta apostila

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.