



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 06 – CHATGPT II

O QUE IREMOS APRENDER

01

REVISÃO

02

ENGENHARIA DE PROMPT

03

IAS PARA APRENDIZADO

04

DESAFIOS FINAIS

Introdução ao ChatGPT

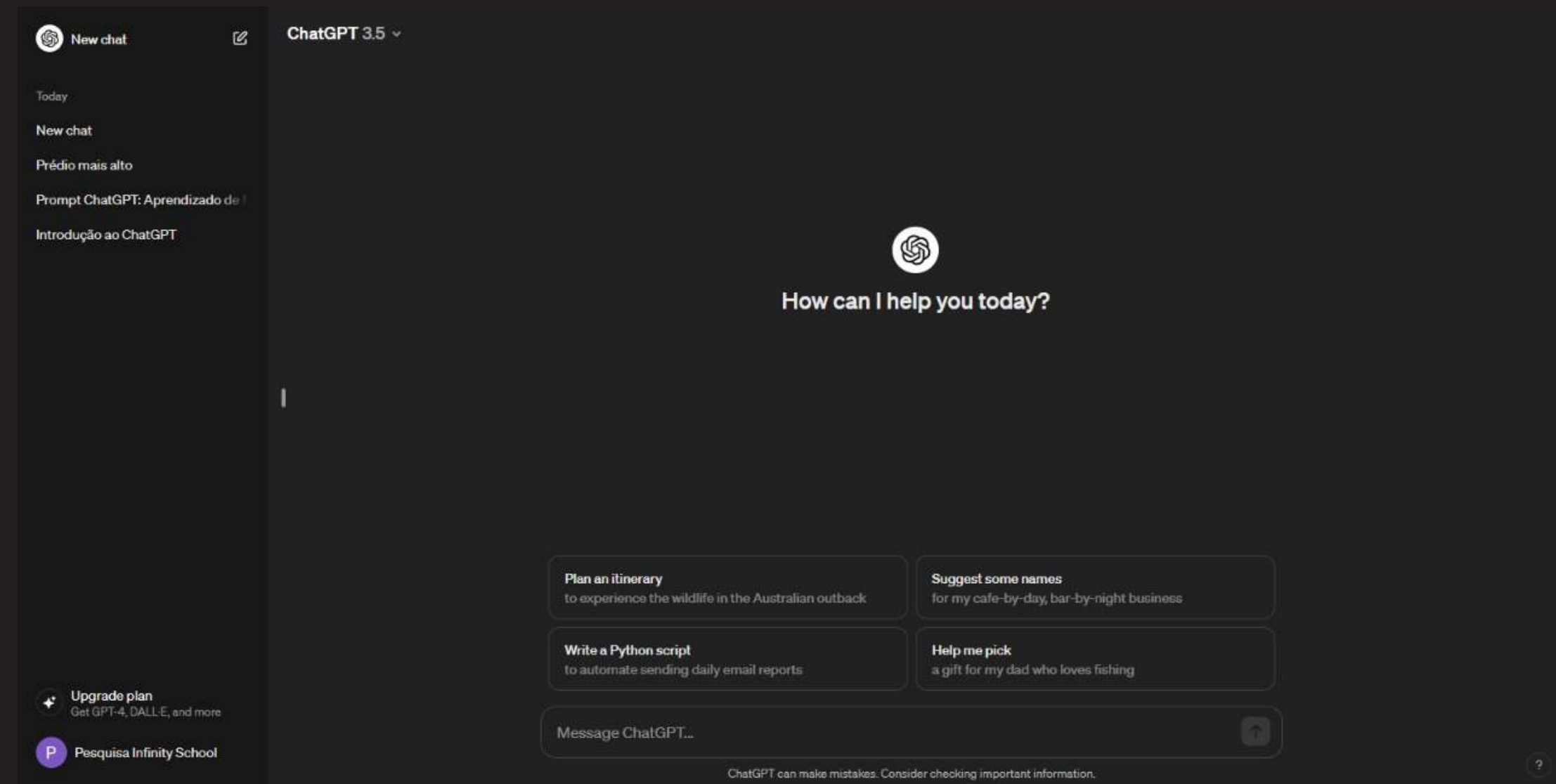


O **ChatGPT** é uma inteligência artificial desenvolvida pela **Open AI**, sendo uma inteligência artificial feita para **processamento de linguagem natural** que foi treinada com o conteúdo da internet. O seu funcionamento se baseia na criação de **chats**, onde nós informamos um determinado **prompt** na linguagem de sua preferência, e ele irá te responder baseado nas informações que ele foi treinado.

Introdução ao ChatGPT

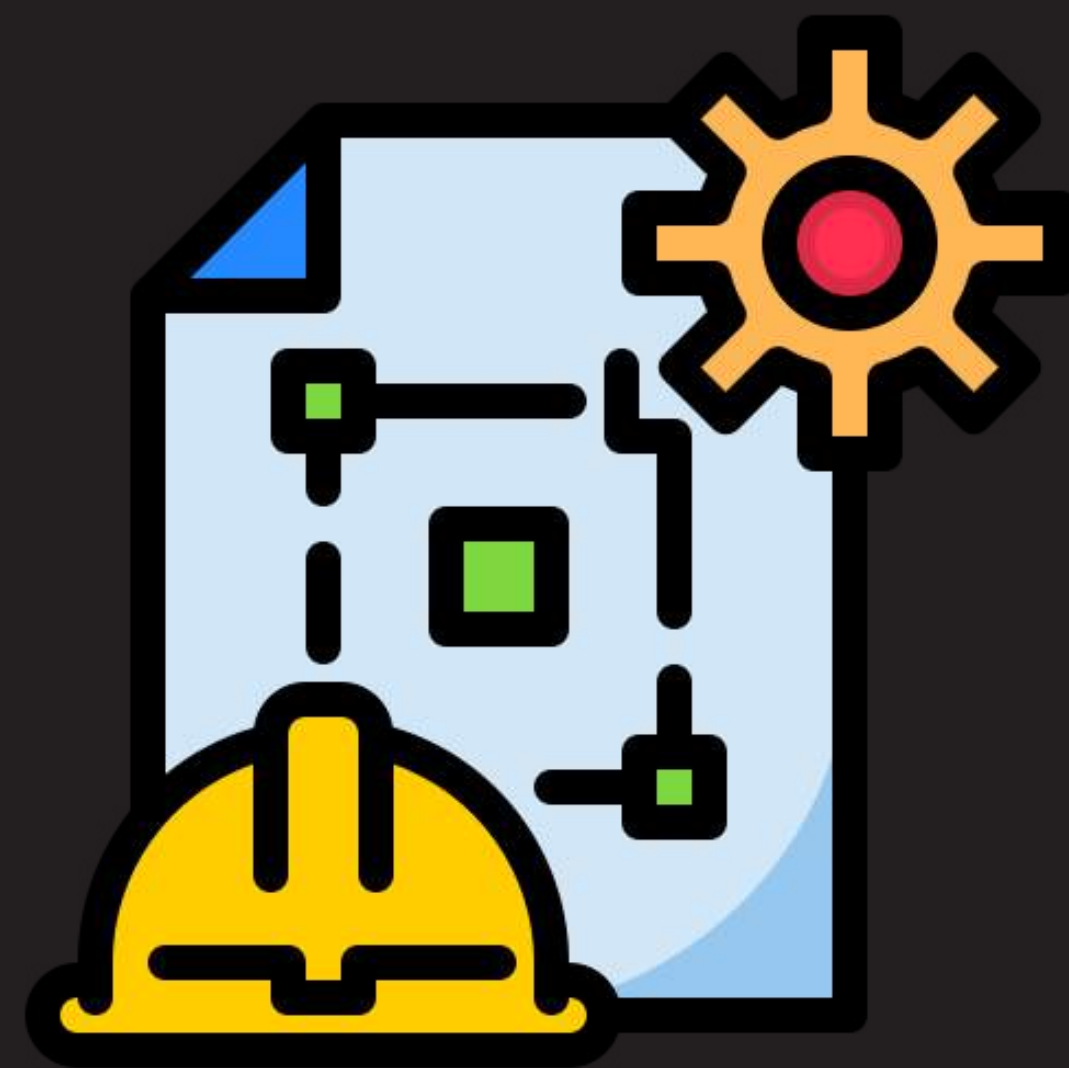
A direita temos a tela inicial do ChatGPT. Aqui onde podemos criar os **chats**, para conversar com o chatgpt.

Na entrada principal, nós podemos escrever o **prompt** para conversar com o chatgpt



Engenharia de Prompt

A **engenharia de prompt** refere-se ao processo de **otimizar** e criar instruções específicas para modelos de linguagem, como o GPT, a fim de obter respostas desejadas ou melhorar o desempenho em tarefas específicas. Isso envolve a **formulação cuidadosa de comandos (prompts)** para obter resultados mais úteis e precisos.



Formulando Prompts Otimizados

PROMPT

Os prompts são mensagens ou símbolos utilizados para solicitar informações de um usuário em sistemas que utilizam modelos de linguagem.

PERSONA

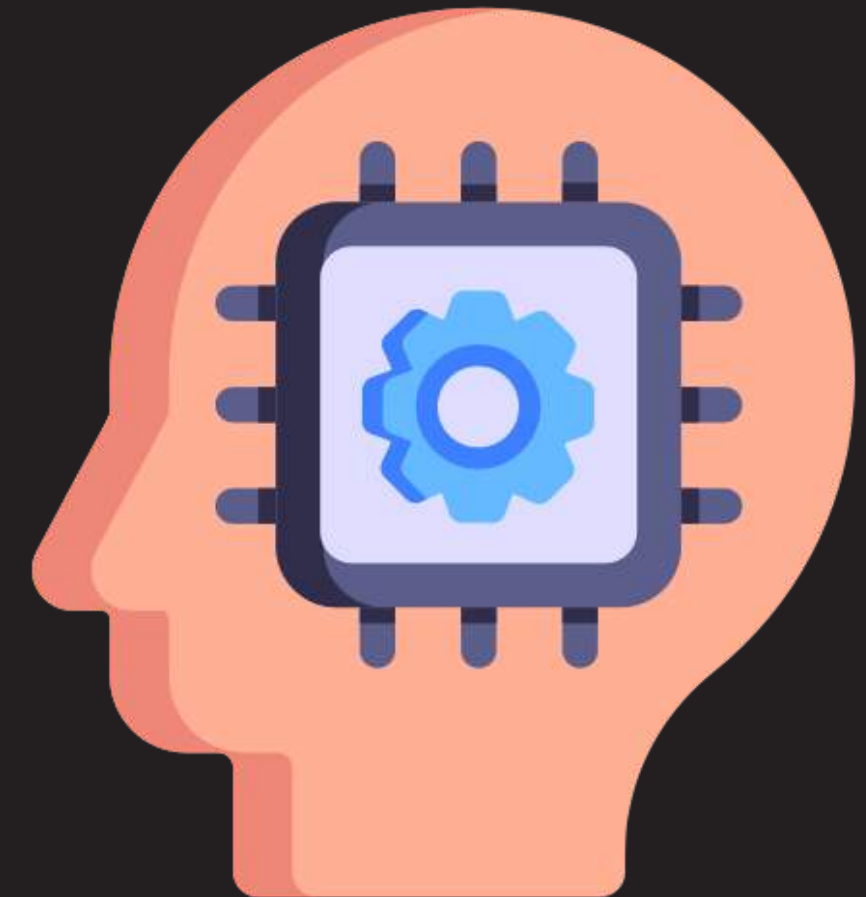
Uma técnica utilizada para fazer o ChatGPT e outras IAs generativas interpretarem um “personagem” que auxilia na compreensão da linguagem acerca do assunto, melhorando os resultados.

OBJETIVIDADE

Seja claro e descreva o mais detalhadamente possível o resultado que você quer. Existem algumas palavras/frases chaves que também melhoram os resultados como “Utilize sua imaginação”.

Exemplo

{IA}, você irá agir como {descrição da persona que resolveria seu problema}, e preciso que você ajude com {objetivo desejado}.



Outras Inteligências Artificiais



PHIND

Search engine que utiliza inteligência artificial para otimizar seus resultados e ainda propõe soluções interessantes para solucionar bugs.



PERPLEXITY

Uma IA que tem acesso a internet, interessante para buscar conceitos e ler docs.



CHATGPT

Inteligência artificial muito interessante para trabalhos de criatividade e orientação.



PHIND

phind

Ask anything. What are you stuck on?

Phind Model ▾



What is llama.cpp?

How to create a link with query params using React router

Async/await syntax in JavaScript

This week's AI news recap





PERPLEXITY



New Thread **Ctrl I**

🔍 Home

🔍 Discover

📖 Library

🔑 Sign in

Sign Up

Try Pro

Upgrade for image upload, smarter AI, and more Pro Search.

➔ Learn More

📄 Download



Where knowledge begins

Ask anything...

☰ Focus ⌕ Attach

● Pro ➔

➔ Try searching The history of film The history of matryoshka dolls
 How do cacti survive in the desert? The best dog breed for apartments



Sign in or create an account

Use Pro Search, upload files, and save your threads.

Continue with Google

Continue with Apple

henry@example.com

Continue with Email

Pro Careers Playground Blog · English (English) ▾

ATIVIDADE PRÁTICA

Atividade 01

Utilize o perplexity para listar e explicar as possíveis áreas que você pode seguir na programação.

Atividade 02

Escolha uma das áreas listadas pelo perplexity e peça para o phind fazer um roadmap dos assuntos utilizando o **princípio de pareto**.

Loop: For

No caso, o **FOR** do Python nos permite realizar essas repetições porque **ele percorre coleções** (strings, listas, dicionários etc.), **tudo que for iterável**. Nada nos impede de utilizar as outras estruturas que vimos anteriormente dentro do **FOR**. Exemplo:

Código:

```
1 nome = 'Davi'
2
3 for letra in nome:
4     if letra != 'a':
5         print(letra)
```

Saída:

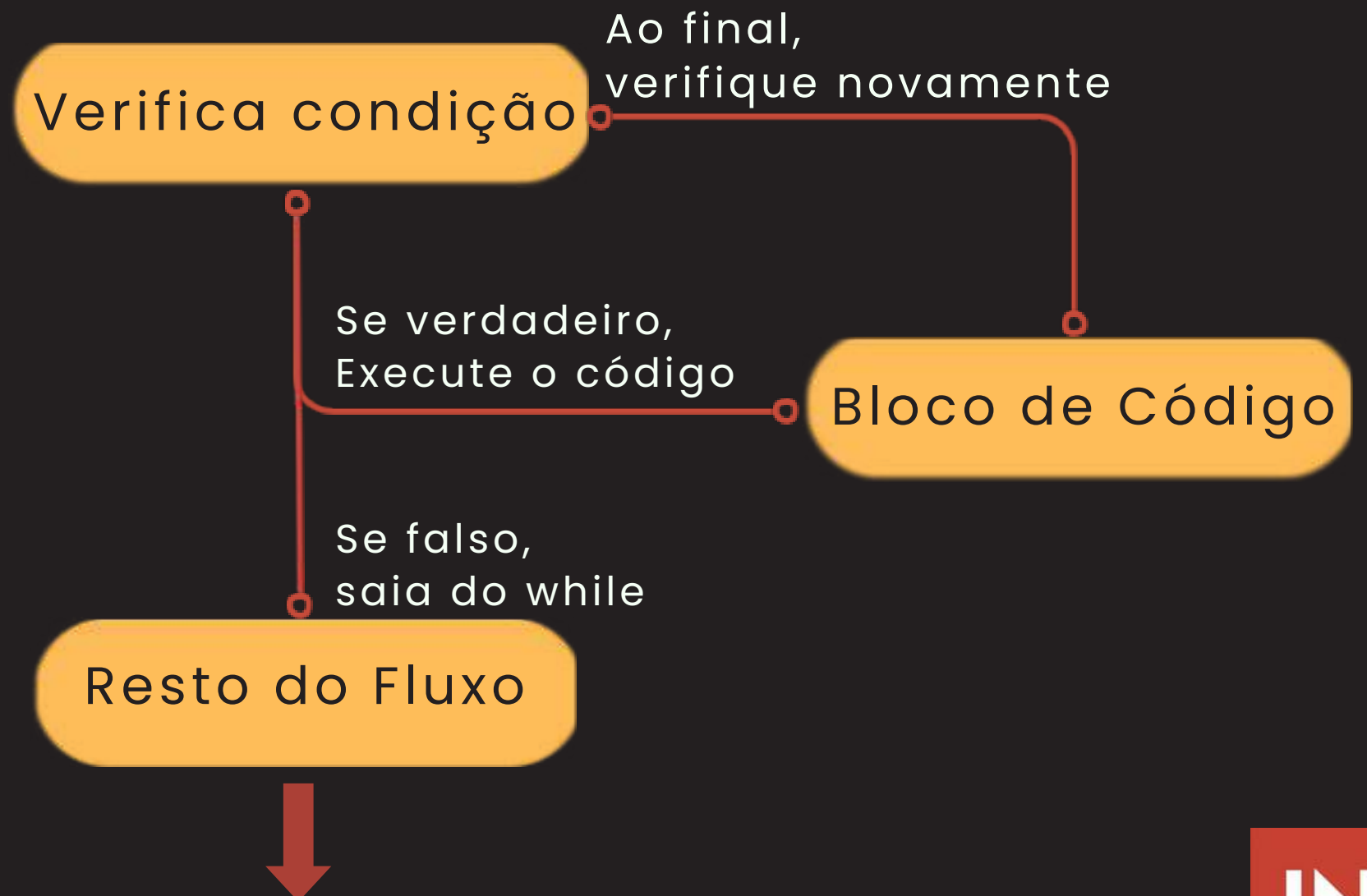
```
/exemplo.py
D
v
i
```

Loop: While

A estrutura de repetição **WHILE** nos permite repetir um determinado bloco de código **enquanto uma condição for verdadeira**. Exemplo:

ENQUANTO <condição>
FAÇA: fazendo algo

```
1 contador = 1
2 while contador < 11:
3     print(contador)
4     contador += 1
```



Variáveis Acumuladoras

Uma **variável acumuladora** é uma variável que é utilizada em **conjunto com estruturas de repetição** para **salvar/acumular** valores durante a execução do loop.

Em um loop while, já estamos utilizando uma variável acumuladora (**contador**) para contar, quantas repetições foram executadas.

```
1 contador = 1
2 while contador < 11:
3     print(contador)
4     contador += 1
```

Variáveis Acumuladoras

Outro exemplo é quando queremos salvar um determinado valor durante um loop. Exemplo: digamos que queremos saber o maior número de uma determinada quantidade digitada.

```
1 maior = None
2 for _ in range(3):
3     numero = int(input('Digite um valor: '))
4     if maior is None or numero > maior:
5         maior = numero
6
7 print(f'O maior numero inserido foi: {maior}')
```

Ao final, temos certeza de qual foi o maior número digitado.

Loop: Break e Continue

A instrução "**break**" é uma instrução que é utilizada em estruturas de repetição para **sair do laço no momento que é executada**.

```
while True:
    resp = input('Deseja continuar?')
    if resp.upper() == 'N':
        # Quebra o laço imediatamente
        break
```

A instrução "**continue**" é uma instrução que é utilizada em estruturas de repetição para **ir para a próxima repetição**, ignorando as linhas abaixo.

```
for num in range(1, 11):
    if num % 2 == 0:
        # Pula para a proxima repetição
        continue
    print(num)
```


Prompt para encontrar Bugs

“Execute testes abrangentes no código Python abaixo, identificando potenciais bugs, erros de lógica e problemas de desempenho:

{CÓDIGO PYTHON AQUI}

Avalie a manipulação correta de exceções, verifique a validação adequada de entradas, e assegure-se de que as saídas estejam em conformidade com as expectativas. Considere casos extremos e situações de borda para garantir robustez. Forneça feedback detalhado sobre qualquer anomalia encontrada, incluindo mensagens de erro, se aplicável. Priorize a clareza do código e a eficiência algorítmica, destacando áreas que podem ser otimizadas ou simplificadas.”

LISTA DE EXERCÍCIOS

1. Faça um programa que peça 4 notas de um aluno (**utilizando estruturas de repetição**) e mostre a média delas.
2. Faça um programa que receba vários números e, quando for digitado um valor negativo, mostre a média dos números digitados (**não considere o negativo**).
3. Faça um programa que receba um número e calcule o fatorial desse número.
4. Faça um programa que receba um número e diga se esse número é primo ou não.
5. Faça um programa que receba um número "n" e mostre a sequência de *Fibonacci* até o **enésimo elemento**.

OBS. Lembre-se dos conceitos que aprendeu nesta aula para montar os prompts

DESAFIO

Faça um Programa que registrará as vendas de uma loja. Cada venda conterá:

Nome do Comprador
Preço da Compra

A cada venda o programa deve perguntar: *"Deseja registrar mais uma venda?"*

O usuário deve responder com "S" ou "N" para a pergunta. Essa resposta deve ser tratada, não permitindo números ou palavras com mais uma letra. Além disso, o programa deve aceitar a resposta independentemente dela ser maiúscula ou minúscula.

Ao final do programa, ele deve mostrar o faturamento total do dia e qual foi o comprador que gastou mais na loja.

SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

PYTHON – AULA 01
LISTAS E TUPLAS

The logo consists of the letters 'IN' in a white, bold, sans-serif font, centered within a solid red square.

INFINITY SCHOOL
VISUAL ART CREATIVE CENTER



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 06 – CHATGPT II